# Usecases and White-box tests

### **Name – Open System**

1. Actors: system manager.
2. Precondition: the system available.
3. Input parameters: user id and password of the system manager.
4. Actions scenario:
   1. *The user signs in to the system.*
   2. *The system checks the rules of integrity.*
   3. *The system manager opens the system.*

**Test 1 – Happy Scenario**

1. The system manager enters his id and password.
2. All the rules of integrity are held.
3. The system opened.

**Test 2 – Happy Scenario**

1. The system manager enters his id and password.
2. All the rules of integrity are not held.
3. The system was not open.

**Test 3 – Sad Scenario**

1. The system manager enters his id and password.
2. All the rules of integrity are not held.
3. The system opened anyway.

**Test 4 – Sad Scenario**

1. The system manager enters the wrong id and password.
2. All the rules of integrity are held.
3. The system opened anyway.

### **Name – Add External Services**

1. Actors: system manager.
2. Precondition: the system available and the system manager is signed in.
3. Input parameters: The information on the external services.
4. Actions scenario:
   1. *The user signs in to the system.*
   2. *The system manager enters the information on the external service.*
   3. *The system adds a new external service.*

**Test 1 – Happy Scenario**

1. The system manager enters his id and password.
2. The system manager enters the information on the new external service.
3. The new external service is added to the system.

**Test 2 – Sad Scenario**

1. The system manager enters his id and password.
2. The system manager enters the information on the new external service.
3. The system alerts that the system is already connected to this service.

**Test 3 – Sad Scenario**

1. The system manager enters his id and password.
2. The system manager enters the information on an existing external service.
3. The system adds the service one more time to the system.

### **Name – Change External Services**

1. Actors: system manager.
2. Precondition: the system available and the system manager is signed in.
3. Input parameters: The information on the external services – the changes param.
4. Actions scenario:
   1. *User sign in to the system.*
   2. *The system manager enters the information that needs to be changed in the external service.*
   3. *The system changes to information of the external service.*

**Test 1 – Happy Scenario**

1. The system manager enters his id and password.
2. The system manager enters the information that he wants to change in the external service.
3. The external service param changes to the system.

**Test 2 – Sad Scenario**

1. The system manager enters his id and password.
2. The system manager enters the information that he wants to change in the external service.
3. The system adds the service one more time to the system.

**Test 3 – Sad Scenario**

1. The system manager enters his id and password.
2. The system manager enters information on a non-existing external service in the system.
3. The system adds the service one more time to the system.

### **Name – Payment**

1. Actors: user.
2. Precondition: the system available and the user is signed in. Also, there is an available and legal product to pay for.
3. Input parameters: The payment information.
4. Actions scenario:
   1. *The user signed in to the system.*
   2. *The user tries to pay on an available item.*
   3. *The user enters his payment information.*
   4. *The payment system confirms the transaction.*
   5. *The payment system sends a receipt to confirm the suppling.*

**Test 1 – Happy Scenario**

1. The user enters his id and password to log in to the system.
2. The user enters his legal payment methods for an item.
3. The payment system confirms the transaction and sends a receipt to confirm the suppling.

**Test 2 – Sad Scenario**

1. The user enters his id and password to log in to the system.
2. The user enters his illegal payment methods for an item.
3. The payment system confirms the transaction and sends a receipt to confirm the suppling.

**Test 3 – Sad Scenario**

1. The user enters his id and password to log in to the system.
2. The user enters his legal payment methods for an item.
3. The payment system confirms the transaction but does not send a receipt to confirm the suppling.

**Test 4 – Sad Scenario**

1. The user enters his id and password to log in to the system.
2. The user enters his legal payment methods for an item.
3. The payment system does not confirm the transaction but sends a receipt to confirm the suppling.

### **Name – Customer Buys From Store Alert**

1. Actors: store manager and a customer.
2. Precondition: the store manager is signed in the system as the store manager- where the customer buys items. Also, I customer sign in to the system and legal to buy item in this store.
3. Input parameters: none.
4. Actions scenario:
   1. *The user signs in to the system.*
   2. *The user buys an item from a store.*
   3. *The store manager gets a live alert about the transaction in his store.*

**Test 1 – Happy Scenario**

1. The user enters his id and password.
2. The user buys an item from a store.
3. The store manager gets a live alert about the transaction.

**Test 2 – Sad Scenario**

1. The user enters his id and password.
2. The user buys an item from a store.
3. The store manager does not get a live alert about the transaction.

**Test 3 – Sad Scenario**

1. The user enters his id and password.
2. The user buys an item from a store.
3. The store manager gets a live alert about something different from the transaction.

### **Name – Subscription Removal Alert**

1. Actors: system manager and store manager.
2. Precondition: the system available and the system manager is signed in.
3. Input parameters: The information of the store manager.
4. Actions scenario:
   1. *The system manager signs in to the system.*
   2. *The system manager enters the information of the store manager he would like to remove his subscription.*
   3. *The system removes the subscription of this store manager.*
   4. *The store manager gets a live alert about his subscription removal.*

**Test 1 – Happy Scenario**

1. The system manager enters his id and password.
2. The system manager enters the information of the store manager he would live to remove his subscription from the system.
3. The store manager subscription is removed from the system.
4. The store manager gets a live alert about the change.

**Test 2 – Sad Scenario**

1. The system manager enters this id and password.
2. The system manager enters the information of the store manager he would live to remove his subscription from the system, but the store manager is not in the system.
3. No exception is received.

**Test 3 – Sad Scenario**

1. The system manager enters his id and password.
2. The system manager enters the information of the store manager he would live to remove his subscription from the system.
3. The store manager subscription is not removed from the system.
4. The store manager does not get a live alert about the change.

### **Name – Received Message Alert**

1. Actors: user connected to the system. And a user that’s not connected to the system.
2. Precondition: the system available and the first user is signed in.
3. Input parameters: the information of the second user.
4. Actions scenario:
   1. *The first user signs in to the system.*
   2. *The first user searches for the second user in the system.*
   3. *The first user sends a message to the second user.*
   4. *The second user gets a live alert about the message the first user sends him.*

**Test 1 – Happy Scenario**

1. The first user enters his id and password.
2. The first user searched and found the second user in the system.
3. The first user sends a message to the second user.
4. The second user gets a live alert about this message.

**Test 2 – Sad Scenario**

1. The first user enters his id and password.
2. The first user searched and did not find the second user in the system.

**Test 3 – Sad Scenario**

1. The first user enters his id and password.
2. The first user searched and found the second user in the system.
3. The first user sends a message to the second user.
4. The second user does not get a live alert about this message.

### **Name – Store Manager Permissions**

1. Actors: store manager.
2. Precondition: the system available. Consider for example that the store manager can add items to the stock but cannot remove items from the store.
3. Input parameters: None.
4. Actions scenario:
   1. *The store manager logs in to the system.*
   2. *The store manager tries to add an item to his store.*
   3. *The item is added to his store.*

**Test 1 – Happy Scenario**

1. The store manager enters his id and password.
2. The system manager chooses the option of adding items to his store.
3. The store manager gets permission to do this functionality.
4. The item is added to his store.

**Test 2 – Sad Scenario**

1. The store manager enters his id and password.
2. The system manager chooses the option of adding items to his store.
3. The store manager does not get permission to do this functionality.
4. The item is not added to his store.

**Test 3 – Sad Scenario**

1. The store manager enters his id and password.
2. The system manager chooses the option of removing items from his store.
3. The store manager gets permission to do this functionality.
4. The item is removed from his store.

### **Name – Close Store Function**

1. Actors: system manager.
2. Precondition: the system available and the system manager is signed in.
3. Input parameters: The information about the store that will be closed.
4. Actions scenario:
5. *The system manager logs in to the system.*
6. *The system manager chooses the option of closing the store.*
7. *The system manager enters the information about the store that will be closed.*
8. *The store is deleting from the system.*

**Test 1 – Happy Scenario**

1. The system manager enters his id and password.
2. The system manager enters the information about the store that will be closed.
3. The store is deleting from the system.

**Test 2 – Sad Scenario**

1. The system manager enters his id and password.
2. The system manager enters wrong information about the store that will be closed, like wrong store id.
3. The system is not alerting on a problem.

**Test 3 – Sad Scenario**

1. The system manager enters this id and password.
2. The system manager enters the information about the store that will be closed.
3. The store is not deleted from the system.

### **Name – Close Store Alert**

1. Actors: user sign up to the store (manager or customer of the store), and the system manager.
2. Precondition: the system available, and the user we are checking, is connected to this store.
3. Input parameters: None.
4. Actions scenario:
5. *The system manager logs in to the system.*
6. *The system manager deletes a store from the system.*
7. *The user gets an alert that the store is being removed from the system.*

**Test 1 – Happy Scenario**

1. The system manager enters his id and password.
2. The system manager deletes the store from the system.
3. The user that is connected to that store gets an alert about the closing of the store.

**Test 2 – Sad Scenario**

1. The system manager enters his id and password.
2. The system manager deletes the store from the system.
3. The user that is connected to that store does not get an alert about the closing of the store.

**Test 3 – Sad Scenario**

1. The system manager enters his id and password.
2. The system manager deletes the store from the system.
3. The user that is connected to that store gets an alert about something else- not the closing of the store.

### **Name – Subscription Removal**

1. Actors: system manager.
2. Precondition: the system available and the system manager is signed in.
3. Input parameters: The information on the subscription that will be removed.
4. Actions scenario:
   1. *The system manager logs in to the system.*
   2. *The system manager enters the information about the subscription that will be removed.*
   3. *The system removes the subscription from the system.*

**Test 1 – Happy Scenario**

1. The system manager enters his id and password.
2. The system manager enters the information on the subscription that will be removed.
3. The subscription is removed from the system.

**Test 2 – Sad Scenario**

1. The system manager enters his id and password.
2. The system manager enters the information on the subscription that will be removed.
3. The subscription does not remove from the system.

**Test 3 – Sad Scenario**

1. The system manager enters his id and password.
2. The system manager enters non-existent information on the subscription that will be removed.
3. The system not alerting q doing something else that not expected.

### **Name – Purchase History in The System (Store)**

1. Actors: system manager.
2. Precondition: the system available and the system manager is signed in.
3. Input parameters: The information about the store.
4. Actions scenario:
   1. *The system manager logs in to the system.*
   2. *The system manager enters the information of the store.*
   3. *The system manager selects the option to see the whole purchase history of this store.*
   4. *The system prints to the system all the historical purchases of this store.*

**Test 1 – Happy Scenario**

1. The system manager enters his id and password.
2. The system manager enters the information of the store.
3. The system prints the purchase history of this store.

**Test 2 – Sad Scenario**

1. The system manager enters his id and password.
2. The system manager enters the information of the store.
3. The system prints nothing.

**Test 3 – Sad Scenario**

1. The system manager enters his id and password.
2. The system manager enters wrong / non existing information of a store.
3. The system not alerting on a problem / print different store historical purchases.

### **Name – Purchase History in The System (Customer)**

1. Actors: system manager.
2. Precondition: the system available and the system manager is signed in.
3. Input parameters: The information about the customer.
4. Actions scenario:
   1. *The system manager logs in to the system.*
   2. *The system manager enters the information of the customer.*
   3. *The system manager selects the option to see the whole purchase history of this customer.*
   4. *The system prints to the system all the historical purchases of this customer.*

**Test 1 – Happy Scenario**

1. The system manager enters his id and password.
2. The system manager enters the information of the customer.
3. The system prints the purchase history of this customer.

**Test 2 – Sad Scenario**

1. The system manager enters his id and password.
2. The system manager enters the information of the customer.
3. The system prints nothing.

**Test 3 – Sad Scenario**

1. The system manager enters his id and password.
2. The system manager enters wrong / non existing information of a customer.
3. The system not alerting on a problem / print different customer historical purchase

### **Name – Shopping cart (Visitor)**

1. Actors: visitor.
2. Precondition: Visitor accessing the system.
3. Input parameters:
4. Actions scenario:
   1. Visitor accesses the system.
   2. System assigns a shopping cart to the visitor.
   3. Visitor browses products and adds items to the shopping cart.

Test 1 – Happy Scenario

1. Visitor successfully adds items to the shopping cart.

2. System accurately tracks the items in the shopping cart.

Test 2 – Sad Scenario

1. Visitor adds products to the shopping cart but encounters technical issues during the process.
2. When the visitor tries to view the cart, none of the previously added products are displayed.

### **Name – Exit - Leaving The System (Visitor)**

1. Actors: Visitor,
2. Precondition: The Visitor accessing the system and defined as a visitor object in the system.
3. Input parameters:
4. Actions scenario:
5. The visitor decides to leave the system.
6. The system clears the visitor's shopping cart.
7. The system clears the visitor from the visitor’s list.
8. The visitor exits the system.

Test 1 – Happy Scenario

1. Visitor successfully exits the system.
2. The shopping cart is cleared without any errors.
3. The visitor is no longer on the visitors’ list.

Test 2 – Sad Scenario

1. The visitor successfully exits the system.
2. Shopping cart is not cleared.

Test 3 – Sad Scenario

1. The visitor successfully exits the system.
2. The visitor is still on the visitors’ list.

### **Name – Register – Becoming a Registered User (Visitor)**

1. Actors: Visitor
2. Precondition: The Visitor accessing the system and defined as a visitor object in the system.
3. Input parameters: Visitor's registration information ( user name, password).
4. Actions scenario:
   1. The visitor registers to the system.
   2. Visitor provides required information (e.g., user name, password).
   3. The system verifies and stores the registration information.
   4. The visitor becomes a registered user of the system.

Test 1 – Happy Scenario

1. The visitor completes the registration process.
2. The system accurately stores the registration information.
3. The visitor becomes a registered user of the system.

Test 2 – Sad Scenario

1. A Visitor register with a username that is already taken by another user.
2. The system accepts the registration request without alerting the visitor about the existing username.
3. As a result, the visitor's registration is processed successfully, but their chosen username is not unique.

### **Name – Login – Becoming a Subscriber/Member (Visitor)** Actors: Visitor Precondition: Visitor already registered with the system. Input parameters: Login credentials. Actions scenario:

1. The visitor provides login credentials (e.g., user name, password).
2. The system verifies the credentials and authenticates the user.
3. The visitor is now defined in the system as a member.

Test 1 – Happy Scenario

1. The registered user successfully logs in with the correct credentials.
2. The visitor is now defined in the system as a member.

Test 2 – Sad Scenario

1. Registered user enters incorrect login credentials.
2. System is not alerting that the credentials are wrong.

### **Name – Get Information About Stores And Products (Visitor)** Actors: Visitor Precondition: The Visitor accessing the system and defined as a visitor object in the system. Input parameters: Actions scenario:

1. Visitor navigates to the section for accessing information about stores and products.
2. The system retrieves and displays a list of available stores and products based on the member's preferences or recent activity.
3. Visitor browses through the list of stores and products, viewing details such as prices, descriptions, and availability.

**Name - Search Products Without Specific Store (Visitor)**

Actors: Visitor  
Precondition: The Visitor accessing the system and defined as a visitor object in the system.

Input parameters:

1. Search By - Product name, category, keywords, or price range
2. Filter By - price range, category, etc…

Actions scenario:

1. Visitor navigates to the search section for products.
2. The visitor enters search criteria such as product name, category, or keywords into the search bar.
3. The system retrieves and displays a list of products matching the search criteria across all available stores.
4. Visitor applies additional filters such as price range, category, etc., to narrow down the search results.
5. The visitor browses through the filtered list of products, viewing details and comparing options.
6. Visitor selects a product of interest to view more detailed information or make a purchase.

**Name - Search Products in a Specific Store (Visitor)**

Actors: Visitor  
Precondition: The Visitor accessing the system and defined as a visitor object in the system.  
Input parameters:

1. Store Name
2. Search By - Product name, category, keywords, or price range
3. Filter By - price range, category, etc…

Actions scenario:

1. Visitor navigates to the search section for products within a specific store.
2. Visitor selects the desired store from a list of available options or enter the store's name directly.
3. The visitor enters search criteria such as product name, category, or keywords into the search bar.
4. The system retrieves and displays a list of products matching the search criteria within the selected store.
5. Visitor applies additional filters such as price range, category, etc., to narrow down the search results further.
6. The visitor browses through the filtered list of products, viewing details and comparing options.
7. Visitor selects a product of interest to view more detailed information or make a purchase within the chosen store.

**Name - Buy The cart (Visitor)**

Actors: Visitor  
Precondition: The Visitor accessing the system and defined as a visitor object in the system.  
cart.  
Input parameters: None  
Actions scenario:

1. Visitor navigates to their shopping cart to review the items selected for purchase.
2. The visitor ensures that all desired items are present in the cart and reviews the quantities.
3. The visitor may adjust quantities or remove items if necessary.
4. The visitor proceeds to the checkout process to finalize the purchase.
5. The system prompts visitor to provide shipping and payment information.

### **Name – Get Information About Stores And Products (Member)** Actors: Member Precondition: The member is already logged in to the system. Input parameters: Actions scenario:

1. Member navigates to the section for accessing information about stores and products.
2. The system retrieves and displays a list of available stores and products based on the member's preferences or recent activity.
3. Member browses through the list of stores and products, viewing details such as prices, descriptions, and availability.

### 

**Name - Search Products Without Specific Store (Member)**

### Actors: Member Precondition: The member is already logged in to the system

Input parameters:

1. Search By - Product name, category, keywords, or price range
2. Filter By - price range, category, etc…

Actions scenario:

1. Member navigates to the search section for products.
2. The member enters search criteria such as product name, category, or keywords into the search bar.
3. The system retrieves and displays a list of products matching the search criteria across all available stores.
4. Member applies additional filters such as price range, category, etc., to narrow down the search results.
5. The member browses through the filtered list of products, viewing details and comparing options.
6. Member selects a product of interest to view more detailed information or make a purchase.

**Name - Search Products in a Specific Store (Member)**

Actors: Member  
Precondition: The member is already logged in to the system  
Input parameters:

1. Store Name
2. Search By - Product name, category, keywords, or price range
3. Filter By - price range, category, etc…

Actions scenario:

1. Member navigates to the search section for products within a specific store.
2. Member selects the desired store from a list of available options or enter the store's name directly.
3. The member enters search criteria such as product name, category, or keywords into the search bar.
4. The system retrieves and displays a list of products matching the search criteria within the selected store.
5. Member applies additional filters such as price range, category, etc., to narrow down the search results further.
6. The member browses through the filtered list of products, viewing details and comparing options.
7. Member selects a product of interest to view more detailed information or make a purchase within the chosen store.

**Name - Buy The cart (Member)**

Actors: Member  
Precondition: The member is already logged in to the system  
Input parameters: None  
Actions scenario:

1. Member navigates to their shopping cart to review the items selected for purchase.
2. The member ensures that all desired items are present in the cart and reviews the quantities.
3. The member may adjust quantities or remove items if necessary.
4. The member proceeds to the checkout process to finalize the purchase.
5. The system prompts member to provide shipping and payment information.

# Name - Logout (Member)

# Actors: Member Precondition: The member is logged in to the system. Input parameters: None Actions scenario:

1. Member navigates to the logout option in the system interface.
2. Member selects the logout option to end the current session.
3. Member is redirected to the visitor mode, returning to the initial state as a visitor.

# Name - Open a Store (Member)

# Actors: Member Precondition: The member is registered and logged in to the system. Input parameters: Store name, store details, product listings, pricing, etc. Actions scenario:

1. Member navigates to the section for opening a new store within the system interface.
2. Member selects the option to create a new store and enters the required information, including store name, description, and contact details.
3. Member can manages inventory and sets stock levels for each product.
4. Member becomes the owner of the store.

# Name - Managing Stock Of A Store (Store owner)

# Actors: Store Owner Precondition: Store owner is logged in and owns the store. Input parameters: Product details (name, description, quantity, etc.), Actions scenario:

1. Store owner navigates to the management section for the owned store within the system interface.
2. Store owner selects the option to manage stock or products within the store.
3. Store owner views a list of products currently available in the store, including their details and stock levels.
4. Store owner can add new products to the store by entering their details such as name, description, price, and quantity.
5. Store owner can remove existing products from the store if they are discontinued or no longer available.
6. Store owner can modify details of existing products such as price, description, or stock levels.
7. Store owner can adjust stock levels of products based on inventory updates, sales, or new arrivals.

# Name - Manage Buying Methods and Discount Policies (Store Owner)

**Actors:** Store Owner

**Precondition:** Store owner is logged in and owns the store.

**Input parameters:** Buying methods (e.g., Auction buy, Bid buy), discount types (e.g., percentage, fixed amount), discount details (e.g., discount rate, validity period)

**Actions scenario:**

1. Store owner navigates to the store management section within the system interface.
2. Store owner selects the option to manage buying methods and discount policies.
3. Store owner chooses to add a new buying method by specifying and any relevant conditions.
4. Store owner can modify existing buying methods by updating the details or conditions associated with each method.
5. Store owner adds a new discount policy by specifying the discount type, discount value, and validity period.
6. Store owner assigns the discount policy to specific products or categories within the store.
7. Store owner can modify existing discount policies by updating the discount details or validity period.

**Name: Appoint Subscriber as New Store Owner (Current Store Owner)**

**Actors:** Current Store Owner, Subscriber

**Precondition:**

* Current store owner is logged in and owns the store.
* Subscriber is registered in the system but not a current store owner.

**Input parameters:** Subscriber ID, store details

**Actions scenario:**

1. Current store owner navigates to the store management section within the system interface.
2. Current store owner selects the option to appoint a new store owner.
3. Store owner enters the subscriber ID of the desired subscriber.
4. Store owner submits the appointment request.
5. The system verifies that the subscriber is not already a store owner.
6. The system sends a notification to the subscriber about the appointment.
7. Subscriber logs in and navigates to their notifications section.
8. Subscriber reviews the appointment request details.
9. Subscriber chooses to accept or reject the appointment
10. Upon acceptance, the system updates the subscriber's role to store owner, granting rights to manage store policies and management.

**Name: Appoint Subscriber as Store Manager (Store Owner)**

**Actors:** Store Owner, Subscriber

**Precondition:**

* Store owner is logged in and owns the store.
* Subscriber is registered in the system but not currently a manager or owner of any store.

**Input parameters:** Subscriber ID, management privileges (viewing only, editing inventory, managing purchase and discount policies, etc.)

**Actions scenario:**

1. Store owner navigates to the store management section within the system interface.
2. Store owner selects the option to appoint a new store manager.
3. Store owner enters the subscriber ID of the desired subscriber.
4. Store owner selects the specific management privileges to be granted to the store manager (e.g., viewing only, editing inventory, managing purchase and discount policies).
5. Store owner submits the appointment request.
6. The system verifies that the subscriber is not already a store manager or owner.
7. The system sends a notification to the subscriber about the appointment.
8. Subscriber reviews the appointment request details, including the specified management privileges.
9. Subscriber chooses to accept or reject the appointment:
10. Upon acceptance, the system updates the subscriber's role to store manager and assigns the specified privileges.
11. The system designates one of the store owners as the manager's delegate.

**Name: Modify Management Options for Store Manager (Store Owner)**

**Actors:** Store Owner

**Precondition:**

* Store owner is logged in and owns the store.
* Store manager is already appointed.
* Store manager was appointed by the this store owner.

**Input parameters:** Manager ID, management privileges (viewing only, editing inventory, managing purchase and discount policies, etc.)

**Actions scenario:**

1. Store owner navigates to the store management section within the system interface.
2. Store owner selects the option to manage the store manager's permissions.
3. Store owner views a list of currently appointed store managers.
4. Store owner selects the manager whose permissions are to be determined or changed.
5. Store owner updates the management privileges by selecting or deselecting options.
6. The system updates the store manager's permissions based on the store owner's selections.
7. The store manager receives a notification detailing the changes to their management privileges.
8. Store owner receives confirmation of the updated management options.

**Name: Close Store (Store Founder)**

**Actors:** Store Founder

**Precondition:**

* Store founder is logged in and owns the store to be closed.

**Input parameters:** Store ID

**Actions scenario:**

1. Store founder navigates to the store management section within the system interface.
2. Store founder selects the option to close the store.
3. Store founder confirms the decision to close the store.
4. The system processes the closure and marks the store as inactive.
5. The system sends notifications to all store owners and store managers about the store's closure.
6. The system restricts access to the closed store’s information, ensuring it is not available to users who are not store owners or administrators.
7. The system updates the market information, ensuring that products from the closed store do not appear in search results or any product listings.
8. Store owners and store managers retain their roles but cannot manage or access the closed store in the same capacity as when it was active.

**Name: View Store Officials and Privileges (Store Owner)**

**Actors:** Store Owner

**Precondition:** Store owner is logged in and owns the store.

**Input parameters:** None

**Actions scenario:**

1. Store owner navigates to the store management section within the system interface.
2. Store owner selects the option to view information about store officials.
3. The system retrieves and displays a list of all officials appointed to the store.
4. Store owner selects an official to view detailed information.
5. The system displays the selected official's information, including their name, role, and assigned management privileges.
6. Store owner can review the privileges of the store officials.

**Name: View Purchase History (Store Owner)**

**Actors:** Store Owner

**Precondition:** Store owner is logged in and owns the store.

**Input parameters:** None

**Actions scenario:**

1. Store owner navigates to the store management section within the system interface.
2. Store owner selects the option to view purchase history.
3. The system retrieves and displays a comprehensive list of all past purchases made within the store.
4. The purchase history displayed remains unaffected by changes made to the market, such as removing a product, changing product descriptions, or adjusting prices.

**Name: Store Manager Operations**

**Actors:** Store Manager, Store Owner

**Precondition:**

* Visitor-subscriber is appointed as a store manager by the store owner.
* Store owner has granted specific management permissions to the store manager.

**Input parameters:** None

**Actions scenario:**

1. store manager logs into the system using their credentials.
2. The system verifies the store manager's role and permissions based on the appointment by the store owner.
3. Store manager navigates to the store management section within the system interface.
4. Store manager views available management options and features based on the permissions granted by the store owner.
5. Store manager performs management operations in the store according to the permissions granted

**Name: View Purchase History Information (System Manager)**

**Actors:** System Manager

**Precondition:**

* System Manager is logged in and has the role of manager of the system.

**Input parameters:** None

**Actions scenario:**

1. system manager logs into the system using their credentials.
2. The system verifies the manager role and permissions of the auditor-subscriber.
3. System Manager navigates to the administrative section within the system interface.
4. System Manager selects the option to access purchase history information.
5. The system retrieves and displays comprehensive purchase history data.

# Acceptance tests

**Scenario: Open System**

**Description:** Verify that the system manager can successfully add a new external service when providing valid information.

**Test Steps:**

1. Provide the user ID and password of the system manager.
2. Verify that all rules of integrity are checked by the system.
3. Attempt to open the system.

**Expected Result:** The system should allow access to the system manager and all relevant features should be accessible.

**Acceptance Criteria:**

1. Upon entering valid credentials, the system should authenticate the system manager.
2. All rules of integrity should be enforced before granting access to the system.
3. The system should open only if the user is authenticated.

**Scenario: Add External Services**

**Description:** Verify that the system manager can successfully add a new external service when providing valid information.

**Test Steps:**

1. The system manager signs in to the system.
2. The system manager enters the information on the new external service.
3. Attempt to add the new external service.

**Expected Result:** The system should allow the system manager to enter information on the new external service, and the new external service should be successfully added to the system.

**Acceptance Criteria:**

1. The system should authenticate the system manager.
2. The system should provide a form or interface for the system manager to input information on the new external service.
3. After submitting the information, the system should add the new external service.

**Scenario: Change External Services**

**Description:** Verify that the system manager can successfully change the parameters of an external service when providing valid information.

**Test Steps:**

1. The system manager signs in to the system.
2. The system manager enters the information that needs to be changed in the external service.
3. Attempt to change the external service parameters.

**Expected Result:** The system should allow the system manager to enter information for the changes to the external service, and the specified parameters of the external service should be successfully changed in the system.

**Acceptance Criteria:**

1. The system should authenticate the system manager.
2. The system should provide a form or interface for the system manager to specify the changes to the external service.
3. After submitting the changes, the system should update the specified parameters of the external service.

**Scenario: Payment**

**Description:** Verify that the user can successfully complete a payment transaction for an available item when providing valid payment information.

**Test Steps:**

1. The user signs in to the system using their ID and password.
2. The user selects an available item to purchase.
3. The user enters their payment information.
4. The user submits the payment.
5. Verify that the payment system confirms the transaction and sends a receipt to confirm the supplying.

**Expected Result:** The payment system should accept the user's payment information, and after submitting the payment, the payment system should confirm the transaction.

**Acceptance Criteria:**

1. Upon entering valid credentials, the user should be authenticated and signed in to the system.
2. The system should provide a list of available items for the user to choose from.
3. The user should be able to enter valid payment information for the selected item.
4. After submitting the payment, the system should confirm the transaction.
5. The system should generate and send a receipt to the user's provided contact information to confirm the supplying.

**Scenario: Customer Buys From Store Alert**

**Description:** Verify that the store manager receives a live alert when a customer successfully purchases an item from their store.

**Test Steps:**

1. The customer signs in to the system using their ID and password.
2. The customer selects and purchases an item from the store.
3. Verify that the store manager receives a live alert about the transaction.

**Expected Result:** The store manager should receive a live alert about the transaction.

**Acceptance Criteria:**

1. Upon entering valid credentials, the customer should be authenticated and signed in to the system.
2. The system should allow the customer to browse and purchase items from the store.
3. After a successful transaction, the system should send a live alert to the store manager about the transaction in their store.

**Scenario: Subscription Removal Alert**

**Description:** Verify that the system manager can successfully remove the subscription of a store manager from the system, and the store manager receives a live alert about the removal.

**Test Steps:**

1. The system manager signs in to the system using their ID and password.
2. The system manager enters the information of the store manager whose subscription they want to remove.
3. Verify that the store manager's subscription is removed from the system.
4. Verify that the store manager receives a live alert about the subscription removal.

**Expected Result:** The store manager's subscription should be successfully removed from the system. And the store manager should receive a live alert about the subscription removal.

**Acceptance Criteria:**

1. The system should allow the system manager to enter the information of the store manager whose subscription they want to remove.
2. After removing the subscription, the system should confirm the removal and update the database accordingly.
3. The system should send a live alert to the store manager about the subscription removal.

**Scenario: Received Message Alert**

**Description:** Verify that the second user receives a live alert when the first user sends them a message.

**Test Steps:**

1. The first user signs in to the system using their ID and password.
2. The first user searches for and finds the second user in the system.
3. The first user sends a message to the second user.
4. Verify that the second user receives a live alert about the message.

**Expected Result:**

1. The first user should be able to search for and find the second user in the system.
2. The first user should be able to send a message to the second user.
3. The second user should receive a live alert about the message sent by the first user.

**Acceptance Criteria:** The live alert should include relevant information about the message, such as the sender's username and the message content.

**Scenario: Store Manager Permissions**

**Description:** Verify that the store manager can successfully add an item to their store, as permitted by their permissions.

**Test Steps:**

1. The store manager logs in to the system using their ID and password.
2. The store manager selects the option to add an item to their store.
3. Verify that the store manager is granted permission to perform this functionality.
4. The store manager adds the item to their store.

**Expected Result:**

1. The system should provide the store manager with the option to add items to their store.
2. The system should grant permission to the store manager to perform this functionality.
3. The item selected by the store manager should be successfully added to their store.

**Acceptance Criteria:**

1. The system should provide a menu or interface that allows the store manager to select the option to add items to their store.
2. After selecting the option, the system should grant permission to the store manager based on their assigned permissions.
3. The system should ensure that the store manager can only perform actions permitted by their permissions, such as adding items but not removing them.

**Scenario: Close Store Function**

**Description:** Verify that the system manager can successfully close a store by providing the necessary information.

**Test Steps:**

1. The system manager selects the option to close a store.
2. The system manager enters the information about the store that will be closed.
3. Verify that the store is successfully deleted from the system.

**Expected Result:** The system should provide the system manager with the option to close a store, and the system manager should be able to enter the necessary information about the store to be closed.

**Acceptance Criteria:**

1. The system should provide a menu or interface that allows the system manager to select the option to close a store.
2. After selecting the option, the system should prompt the system manager to enter the information about the store to be closed.
3. The system should ensure that the specified store is deleted from the system's database.

**Scenario: Close Store Alert**

**Description:** Verify that users connected to a store receive an alert when the system manager deletes the store from the system.

**Test Steps:**

1. The system manager logs in to the system using their ID and password.
2. The system manager deletes the store from the system.
3. Verify that users connected to the store receive an alert about the store closure.

**Expected Result:** Users connected to the store should receive an alert about the store closure.

**Acceptance Criteria:**

1. After deleting the store, the system should send an alert to all users connected to the store about the closure.
2. The alert should contain relevant information about the store closure, such as the reason for closure and any actions users need to take.

**Scenario: Subscription Removal**

**Description:** Verify that the system manager can successfully remove a subscription from the system by providing the necessary information.

**Test Steps:**

1. The system manager logs in to the system using their ID and password.
2. The system manager enters the information about the subscription that will be removed.
3. Verify that the subscription is successfully removed from the system.

**Expected Result:** The subscription specified by the system manager should be successfully removed from the system.

**Acceptance Criteria:**

1. The system manager should be able to enter the necessary information about the subscription to be removed.
2. The subscription specified by the system manager should be successfully removed from the system.

**Scenario: Purchase History in The System (Store)**

**Description:** Verify that the system manager can successfully view the purchase history of a store by providing the necessary information.

**Test Steps:**

1. The system manager logs in to the system using their ID and password.
2. The system manager enters the information of the store.
3. The system manager selects the option to view the purchase history of the store.
4. Verify that the system prints the purchase history of the store.

**Expected Result: T**he system manager should be able to select the option to view the purchase history of the store. And the system should print the purchase history of the store.

**Acceptance Criteria:**

1. The system should allow the system manager to specify the store for which to view the purchase history.
2. After selecting the option, the system should retrieve and print the purchase history of the specified store.

**Scenario: Purchase History in The System (Customer)**

**Description:** Verify that the system manager can successfully view the purchase history of a customer by providing the necessary information.

**Test Steps:**

1. The system manager logs in to the system using their ID and password.
2. The system manager enters the information of the customer.
3. The system manager selects the option to view the purchase history of the customer.
4. Verify that the system prints the purchase history of the customer.

**Expected Result:** The system manager should be able to select the option to view the purchase history of the customer. And the system should print the purchase history of the customer.

**Acceptance Criteria:**

1. The system should allow the system manager to specify the customer for which to view the purchase history.
2. After selecting the option, the system should retrieve and print the purchase history of the specified customer.

**Scenario: Log In to the System**

**Description:** Verify that a user can log in to the system using his credentials.

**Test Steps:**

1. The user navigates to the system login page.
2. The user enters their username and password.
3. Verify that the system redirects the user to their dashboard with all his information.

**Expected Result:** The user should be able to enter their username and password into the login form. After submitting the form, the system should authenticate the user's credentials. Upon successful authentication, the system should redirect the user to their dashboard or the main page.

**Scenario: Add Product to Cart as Guest User**

**Description:** A guest user visits the website and adds a product to their cart without being logged in.

**Test Steps:**

1. The guest user navigates to the website's homepage.
2. The guest user browses the product catalog to find the desired item.
3. The guest user chooses the desired quantity.
4. The guest user clicks on the "Add to Cart" button.
5. The system adds the selected product to the guest user's cart.
6. The system updates the cart total and displays a confirmation message.

**Expected Result:** The guest user should be able to add the selected product to their cart successfully.

**Scenario: Delete Guest User's Cart Upon Leaving Website**

**Description:** When a guest user leaves the website without completing a purchase, their cart is automatically deleted to maintain privacy and data security.

**Test Steps:**

1. The guest user adds at least one item to their cart.
2. The guest user decides to leave the website without completing the purchase.
3. The guest user closes the website.
4. The system automatically deletes the guest user's cart and associated items from the database.

**Expected Result:** When a guest user leaves the website without completing a purchase, their cart and associated items should be deleted.

**Scenario: Register to the System**

**Description:** A user registers to the system to create an account.

**Test Steps:**

1. The user fills in the registration form with their personal details and submit the form.
2. The system validates the user's input data.
3. If the input data is valid, the system creates a new user account.
4. The user is redirected to the login page and prompted to log in with their newly created credentials.

**Expected Result:** The user should be able to successfully complete the registration process. And the system should validate the user's input data and create a new account if the data is valid.

**Scenario: Modify Cart Contents**

**Description:** A user should be able to make changes to the items in their cart, such as updating quantities, removing items, or adding new items.

**Test Steps:**

1. The user navigates to their cart page.
2. The user reviews the items currently in their cart.
3. The user makes changes in the cart, such as increasing or decreasing the quantity of an item or removing an item entirely.
4. The system updates the cart contents according to the user's changes.
5. The system recalculates the total price of the cart based on the updated contents.

**Expected Result:** The user should be able to successfully make changes to their cart, such as updating quantities or removing items. The system should accurately reflect the user's changes in the cart contents and total price.

**Scenario: Search for a Product**

**Description:** A user should be able to search for a product using the search functionality provided by the system.

**Test Steps:**

1. The user navigates to the search bar and enters a search query.
2. The system displays search results matching the user's query.
3. The user reviews the search results to find the desired product.
4. If the desired product is not found, the system should print a relevant message.

**Expected Result:** The user should be able to enter search queries or keywords into the search bar. And the system should provide relevant search results matching the user's query.

**Scenario: Buy a cart as a guest user**

**Description:** A guest user should be able to buy the entire contents of their cart without needing to create an account.

**Test Steps:**

1. The guest user adds one or more items to their cart.
2. The guest user navigates to their cart page and confirm that he wants to buy the cart.
3. The system redirects the guest user to the checkout page.
4. The guest user enters the required information for the purchase, such as shipping address and payment details.
5. The system processes the payment and completes the purchase.
6. The system displays a confirmation message to the guest user.

**Expected Result:** The guest user should be able to purchase the entire contents of their cart without needing to create an account.

**Scenario: Buy a cart as a Logged-in User**

**Description:** A logged-in user should be able to buy the contents of their cart, with the option to buy only selected items rather than the entire cart.

**Test Steps:**

1. The logged-in user adds one or more items to their cart.
2. The user navigates to their cart page to review the items.
3. The user decides to proceed with the purchase and clicks on the checkout button.
4. The system redirects the user to the checkout page.
5. The user reviews the items in their cart and decides which items he want to buy before proceeding.
6. The system prompts the user to enter shipping and payment information.
7. The user enters the required information.
8. The system processes the payment and completes the purchase.
9. The system displays a confirmation message to the user.
10. The user cart updated as needed.

**Expected Result:** The logged-in user should be able to purchase the contents of their cart, with the option to modify or remove items before completing the purchase.

**Scenario: Logout and Return as Guest**

**Description:** A logged-in user should be able to log out from their account and return to the website as a guest user.

**Test Steps:**

1. The logged-in user navigates to the account settings and log out.
2. The system logs out the user and redirects them to the homepage or login page.
3. The user navigates to a product page or adds an item to the cart as a guest user.

**Expected Result:** The logged-in user should be able to successfully log out from their account. After logging out, the system should redirect the user to homepage.

**Scenario: Write a Product Review as a User**

**Description:** A user who has purchased a product should be able to write a review for that product to share their experience with other users.

**Test Steps:**

1. The user logs in to their account on the website.
2. The user navigates to the product page of the item they want to review.
3. The user clicks on the "Write a Review" button.
4. The system opens a form for the user to input their review details.
5. The user fills in the review form, including a star rating, title, and written review.
6. The user submits the review by clicking on the "Submit" or "Post Review" button.
7. The system processes the review submission and displays the user's review on the product page.

**Expected Result:** The user should be able to successfully write and submit a review for the product they have purchased. And the system should display the user's review.

**Scenario: Open a Store as a User**

**Description:** A user should be able to open a store on the website, allowing them to sell products to other users.

**Test Steps:**

1. The user logs in to their account on the website.
2. The user navigates to the account settings and open a new store.
3. The system prompts the user to provide information about their store, such as store name, description, and contact details.
4. The user fills in the required information and agrees to any terms and conditions and submit.
5. The system processes the store opening request and creates a new store for the user.
6. The user receives a confirmation message indicating that their store has been successfully opened.
7. The user information updated to a store owner

**Expected Result:** The user should be able to successfully open a store on the website. The system should save all necessary information and update the user as this store owner.

**Scenario: Send Message or Questions to a Store**

**Description:** A user should be able to communicate with a store by sending messages or asking questions about products or services.

**Test Steps:**

1. The user logs in to their account on the website.
2. The user navigates to the store page of the desired store and choose “Contact”.
3. The system opens a form for the user to input their message.
4. The user fills in the required information and submit.
5. The system processes the message or question and sends it to the store owner.
6. The store owner receives the message and may respond to the user accordingly.

**Expected Result:** The user should be able to successfully send a message or ask a question to the desired store. And the owner should be able to receive the message and a notification about it.