

Test Cases for <Use-Case 5: Manage User Account>:

Test Case 1 - Successful Account Creation:

Description: Validate successful account creation when all input fields are correctly filled.

Pre-conditions:

- 1) User navigates to the account creation page.
- 2) Input fields for username, email, password, and confirm password are available.

Post-conditions: After clicking "Create Account" with all fields correctly filled, an account should be successfully created for the user.

Data required: Valid username, email, password, and matching confirm password.

Test Case 2 - Missing @ in Email Address:

Description: Test if an error message is displayed when the email lacks the "@" symbol.

Pre-conditions:

- 1) User navigates to the account creation page.
- 2) Email field is filled without the "@" symbol.

Post-conditions: After clicking the "Create Account" button, an error message should be displayed indicating that the email format is incorrect.

Data required: Email address without "@".

Test Case 3 - Password Mismatch:

Description: Check if the app gives an error message when password and confirm password fields do not match.

Pre-conditions:

- 1) User navigates to the account creation page.

Samwise App Test Cases

2) Password and confirm password fields have different values.

Post-conditions: After clicking the "Create Account" button, an error message "Passwords do not match" should be displayed.

Data required: Different passwords in the password and confirm password fields.

Test Case 4 - Missing Information During Account Creation:

Description: Check if the app gives an error message when one of the fields during account creation is not filled.

Pre-conditions:

1) User navigates to the account creation page.

2) User does not fill one of the fields.

Post-conditions: After clicking the "Create Account" button, an error message "Please fill this field" should be displayed.

Data required: Missing one of the fields of username, email, password or confirm password.

Test Case 5 - Verify Account Creation Button Functionality:

Description: Ensure that the "Create Account" button functions correctly.

Pre-conditions: User fills in all required fields (username, email, password, confirm password) correctly.

Post-conditions: After clicking the "Create Account" button, the account creation process should initiate without errors, and the user should be directed to a confirmation or home page.

Data required: Valid inputs for username, email, password, and matching confirm password.

Test Case 6 - Successful Login:

Description: Validate successful login when all input fields are correctly filled.

Pre-conditions:

1) User navigates to the login page.

2) Input fields for username and password are available.

Samwise App Test Cases

Post-conditions: After clicking "Login" with all fields correctly filled, the user can successfully login to her account.

Data required: Valid username and password.

Test Case 7 - Username-Password Mismatch During Login:

Description: Check if the app gives an error message when username and password are not compatible.

Pre-conditions:

- 1) User navigates to the login page.
- 2) User enters the wrong password for the username.

Post-conditions: After clicking the "Login" button, it gives the error message "Invalid username or password" indicating the username and password do not match.

Data required: Non-matching username and password.

Test Cases for <Use-Case 9: Manage Service>:

Test Case 8 - Successful Service Creation:

Description: Verify successful creation of a new service.

Pre-conditions:

- 1) User navigates to the service creation page.
- 2) Required fields for Service Name, Service Type, and Available Times are available.

Post-conditions: After clicking "Create Service" button with all fields correctly filled,

- 1) A new service should be created.
- 2) The system generates and displays a Service UUID code for the newly created service.

Data required: Valid Service Name, Service Type, and Available Times.

Test Case 9 - Missing Information During Service Creation:

Samwise App Test Cases

Description: Check if the app gives an error message when one of the fields during service creation is not filled.

Pre-conditions:

- 1) User navigates to the service creation page.
- 2) User does not fill one of the fields.

Post-conditions: After clicking the "Create Service" button, it gives an error message "Please fill this field".

Data required: Missing one of the fields Service Name, Service Type, and Available Times.

Test Case 10 - Successful Edit of Service Information:

Description: Ensure successful editing of service information.

Pre-conditions:

- 1) User navigates to the Edit Service section.
- 2) User enters a valid Service UUID to retrieve service information.
- 3) Modifies Service Name, Service Type, or Available Times fields.

Post-conditions: After clicking the "Edit Service" button, the service information (Service Name, Service Type, Available Times) should be updated successfully.

Data required: Valid Service UUID and modified Service Name, Service Type, or Available Times.

Test Case 11 - Edit Service with Invalid Service UUID:

Description: Check if it gives an error message if wrong Service UUID is entered.

Pre-conditions:

- 1) User navigates to the Edit Service section.
- 2) User enters an invalid Service UUID to retrieve service information.
- 3) Modifies Service Name, Service Type, or Available Times fields.

Samwise App Test Cases

Post-conditions: After clicking the "Edit Service" button, the error message "Invalid Service UUID" is displayed and service information is not updated.

Data required: Invalid Service UUID.

Test Case 12 - Successful Delete of Service Information:

Description: Ensure successful deleting of service information.

Pre-conditions:

- 1) User navigates to the Delete Service section.
- 2) User enters a valid Service UUID of the service to be deleted.

Post-conditions: After clicking the "Delete Service" button, the service corresponding to the entered Service UUID should be deleted successfully.

Data required: Valid Service UUID of the service to be deleted.

Test Case 13 - Delete Service with Invalid Service UUID:

Description: Test the system's response when attempting to delete a service with an invalid Service UUID.

Pre-conditions:

- 1) User navigates to the Delete Service section.
- 2) User enters an invalid Service UUID to delete service information.

Post-conditions: After clicking the "Delete Service" button,

- 1) The system should handle the invalid Service UUID and display an error message "Invalid Service UUID" indicating that the service does not exist or the UUID is incorrect.
- 2) No service deletion should occur.

Data required: Invalid Service UUID.