**Test Plan: User Management Feature**

**1. Introduction**

The User Management feature allows users to add, update, and delete user records. Added users are displayed in a table sorted by their last name, and users can be deleted individually.

**2. Objectives**

* Ensure that users can be added, updated, and deleted successfully.
* Verify that the users are displayed in a table sorted by last name.
* Confirm that each user can be deleted individually.

**3. Scope**

This test plan covers the following functionalities:

* Adding a new user
* Updating an existing user
* Deleting a user
* Verifying the sorting of users by last name in the table
* Ensuring data integrity after each operation

**4. Test Items**

* User Management interface
* User table (displayed users)
* Add User functionality
* Update User functionality
* Delete User functionality
* Sorting functionality

**5. Assumptions**

* The application has a user management interface.
* The application sorts users by their last names after each add, update, or delete operation.
* The user table refreshes automatically after each operation.

**6. Test Environment**

* Web browser (latest version)
* Desktop and mobile devices
* Stable internet connection
* Test database with dummy user records

**7. Test Cases**

**7.1. Add User**

* **Test Case 1: Add User with Valid Data**
  + **Objective:** Verify that a user can be added with valid data.
  + **Steps:**
    1. Navigate to the "Add User" form.
    2. Enter valid data in all required fields.
    3. Submit the form.
    4. Verify that the user appears in the table.
    5. Verify that the user is sorted correctly by last name.
  + **Expected Result:** User is added, displayed in the table, and sorted by last name.
* **Test Case 2: Add User with Missing Required Fields**
  + **Objective:** Ensure that the system does not allow adding a user with missing required fields.
  + **Steps:**
    1. Navigate to the "Add User" form.
    2. Leave one or more required fields empty.
    3. Attempt to submit the form.
    4. Verify that an error message is displayed.
  + **Expected Result:** User is not added, and an appropriate error message is displayed.
* **Test Case 3: Add User with Duplicate Email**
  + **Objective:** Check that the system prevents adding users with duplicate email addresses.
  + **Steps:**
    1. Add a user with a specific email address.
    2. Attempt to add another user with the same email address.
    3. Verify that an error message is displayed.
  + **Expected Result:** User is not added, and an error message indicating the email is already in use is displayed.

**7.2. Update User**

* **Test Case 4: Update User with Valid Data**
  + **Objective:** Verify that an existing user can be updated with valid data.
  + **Steps:**
    1. Select an existing user from the table.
    2. Modify one or more fields in the user’s information.
    3. Submit the update.
    4. Verify that the changes are reflected in the table.
    5. Verify that the user remains correctly sorted by last name.
  + **Expected Result:** User’s information is updated, and the table reflects the changes.
* **Test Case 5: Update User with Invalid Data**
  + **Objective:** Ensure that the system does not allow updating a user with invalid data.
  + **Steps:**
    1. Select an existing user from the table.
    2. Modify one or more fields with invalid data.
    3. Attempt to submit the update.
    4. Verify that an error message is displayed.
  + **Expected Result:** User is not updated, and an appropriate error message is displayed.

**7.3. Delete User**

* **Test Case 6: Delete User**
  + **Objective:** Verify that a user can be deleted individually.
  + **Steps:**
    1. Select a user from the table.
    2. Click the delete button next to the user.
    3. Confirm the deletion if prompted.
    4. Verify that the user is removed from the table.
  + **Expected Result:** User is deleted, and the table updates to reflect the deletion.
* **Test Case 7: Delete Multiple Users One by One**
  + **Objective:** Confirm that multiple users can be deleted one by one.
  + **Steps:**
    1. Delete several users, one after the other.
    2. Verify that each user is removed from the table after deletion.
  + **Expected Result:** Each user is successfully deleted, and the table updates accordingly.

**7.4. Sorting Verification**

* **Test Case 8: Verify Sorting of Users by Last Name**
  + **Objective:** Ensure that users are always sorted by last name in the table.
  + **Steps:**
    1. Add several users with different last names.
    2. Verify that the users appear in the correct alphabetical order by last name in the table.
    3. Update a user’s last name to change their position in the sort order.
    4. Verify that the table updates the sort order correctly.
  + **Expected Result:** Users are correctly sorted by last name, even after updates.

**7.5. Data Integrity**

* **Test Case 9: Verify Data Integrity After Multiple Operations**
  + **Objective:** Ensure that the data remains consistent after multiple add, update, and delete operations.
  + **Steps:**
    1. Perform a series of add, update, and delete operations.
    2. Verify that the data displayed in the table is accurate and consistent.
  + **Expected Result:** Data remains accurate and consistent in the table after multiple operations.

**8. Test Execution**

* **Testers:** Assigned QA team members
* **Test Schedule:** To be defined by the project timeline
* **Test Environment:** Development, Staging, and Production environments

**9. Risks**

* Possible delays in test execution due to dependency on development.
* Potential integration issues with other features.

**10. Reporting**

* **Test Results Documentation:** Detailed report of all test cases executed, including pass/fail status, issues found, and steps to reproduce.
* **Bug Reporting:** All defects found will be reported in the issue tracking system with high-priority issues flagged.

**11. Conclusion**

This test plan aims to thoroughly test the add, update, and delete functionality of the User Management feature to ensure that it meets the specified requirements and maintains data integrity throughout the process.