Red = broken functionality Yellow = visual bugs

|  |  |  |
| --- | --- | --- |
| **Use Case:** Create account | | |
| **Scenario 1**: A new user creates an account and provides a username, password (1), and email. After the create account process is done, the user then logs into their profile with their username and password (9). Once successful, the program takes the user to the profile page. | Tester: **Jackson** | Output: Functionality works correctly. |
| **Scenario 2**: A new user creates an account, and provides a username, password, and email. |  |  |
| **2a**: The password is not strong enough (19) | Tester: **Jackson** | Output: displays password is not valid. |
| **2b**: The email is invalid | Tester:  **David** | Output: Displays email is invalid. |
| **2c**: The username is already taken | Tester: **Jackson, David** | Output: displays username is already taken. |
| **Scenario 3:** A new user creates an account and provides a username, password, and email. After the create account process is done, the user attempts to log into their profile with a username and password. |  |  |
| **3a**: The username does not exist | Tester: **Jackson** | Output: The user cannot log in, and a message saying so is displayed. |
| **3b**: The password does not match | Tester: **Jackson, David** | Output: The user cannot log in, and a message saying so is displayed. |
| **Scenario 4:** A present user attempts to click ‘login’, but accidently presses ‘create account’ and needs to return to the previous page. | Tester: **Jackson,**  **David** | Output: The user returns to the login screen. |
| **Use Case:** Create Group | | |
| **Scenario 1:** A user goes to the create group page to create a new group. The user enters a group name and selects at least one and up to four tags that describe the group. The user then enters a description and clicks the ‘create group’ button, and is redirected to the search for groups tab. | Tester: David | Output: prints new group added to db on console.  Returns to search for group tab. |
| **1a:** The user does not provide a group description and clicks the button. | Tester: David | Output:prints new group added to db on console.  Returns to search for group tab. |
| **Scenario 2:** A user creates a group on the create group tab |  |  |
| **2a**: The user does not provide a group name | Tester: David | Output: error msg is displayed on screen. |
| **2b**: The user does not provide any group tags | Tester: David | Output: error msg displays |
| **2c:** The provided group name already exists | Tester: David | Output: error msg displays |
| **Use Case:** Add meetings | | |
| **Scenario 1:** A group leader goes to the edit group page to add a meeting. The group leader chooses the group to add the meeting to. The leader then inputs the location, date, and time for the new meeting and submits the information. The fields are cleared and the screen displays a ‘success’ confirmation message. The meeting is then added to the table of meetings on the same tab. | Tester: David | Output: Meeting is added to group. |
| **Scenario 2:** A group leader goes to the edit group page to add a meeting |  |  |
| **2a:** The leader does not select a group | Tester: David | Output: error msg displayed |
| **2b:** One of the meeting details were left empty | Tester: David | Output: error msg displayed |
| **Use Case:** Edit meetings | | |
| **Scenario 1:** A group leader goes to the edit group page to edit an existing meeting. The leader first selects the group and the selection fills a table of existing meetings. The leader highlights and double-clicks an entry in the meeting table. A new dialogue appears and all the details of the meeting that exist. Once the leader makes the necessary modifications, the leader clicks the save button and returns to the primary screen. The meeting table will be reloaded with the updated meeting. | Tester: David | Output: test passed |
| **Scenario 2:** A group leader goes to the edit group page to edit an existing meeting. |  |  |
| **2a:** The leader does not select a group | Tester: David | Output: error msg displayed |
| **2b:** One of the required fields in the dialogue is missing: Date, Time, or Location fields | Tester: David | Output: error msg displayed |
| **Scenario 3**: The group leader selects a group and opens the edit meeting dialogue. The leader then cancels making any edits to the meeting and returns to the primary screen. | Tester: David | Output: test passed  returns to profile page |
| **Use Case:** Edit account | | |
| **Scenario 1:** The user clicks the edit profile button and is taken to a new scene. The user edits account information in accordance to the SRS document. The user selects submit and returns to the primary screen | Tester:  Nicholas H | Output: The user is returned back to the profile tab of the primary scene’s tabpane. If the username was edited, then the new username should be displaying in the edit profile tab |
| **Scenario 2:** The user attempts to edit account information |  |  |
| **2a:** the user has left one of the required fields blank: Username, old password, confirm password, new password, or email. | Tester:  Nicholas H | Output: If all fields are left blank, the user is shown a warning message, and the proper fields are highlighted in red to show they need to be filled. |
| **2b:** The confirm fields do not match | Tester:  Nicholas H | Output: The user will now see a proper warning message, and fields will become highlighted. |
| **Scenario 3:** The user clicks the cancel button on the edit profile screen and returns to the primary screen without anything changing. | Tester:  Nicholas H | Output: The user will now be returned to the profile tab of the primary screen tabpane, and the username displayed should remain the same as it was before going to the edit profile screen. |
| **Use Case:** Join Group | | |
| **Scenario 1:** The user selects a group to join and is displayed a confirm message. The user can view the group joined in the profile tab and can now view the meetings for that group in the view meetings tab. | Tester:  Cameron | Output: A confirm message is displayed for a user, the user can view the group in their profile tab, and they can view meetings for that group in the meetings tab |
| **Scenario 2:** The user does not select a group to join and presses the ‘join’ button. | Tester:  Cameron | Output: The user does not join any group. A message is displayed for the user |
| **Scenario 3:** The user selects a group the user is already a member of and presses the ‘join’ button. | Tester:  Cameron | Output: The user sees a message informing them that they are already in that group, they do not join the group a second time. |
| **Use Case:** Search Group | | |
| **Scenario 1:** The user goes to the search tab and selects at least one tag to search with using any permutation of the four drop down boxes. The user hits the ‘search’ button, and a result of groups will be displayed according to SRS statements. | Tester:  Cameron | Output: The same groups are displayed regardless of what order the tag boxes are selected in or which boxes are left blank |
| **Scenario 2:** The user goes to the search tab | Cameron | They see the search tab, the results of their last search are displayed |
| **2a:** The user attempts to search for groups without entering any group tags. | Tester:  Cameron,  David | Output:  All groups are displayed. |
| **Scenario 3:** The user selects a group tag to search for. The user then wishes to remove that selection from the search queue. | Tester: David | Output: User can deselect tag. |
| **Use Case:** Search meeting | | |
| **Scenario 1:** The user goes to the search meeting tab. The user then enters input into the fields (according to the SRS document) that the user wishes to search by and clicks the ‘search meeting’ button. The table is populated with the meeting results. | Tester:  Cameron | Output: When searching for meetings only meetings fitting all specified criteria are returned. |
| **Scenario 2a:** The user enters an invalid date. | Tester:  Cameron | Output: The date field is cleared, the meetings displayed do not change |
| **Scenario 3:** The user does not enter any search criteria fields and clicks the button to search for meetings. | Tester:  Cameron,  David | Output: only display those the user has joined |
| **Use Case:** View meeting | | |
| **Scenario 1:** The user selects a meeting from the list of meetings they may view, and a meeting details dialogue appears. The user reads all the meeting information. The user closes the dialogue and returns to the primary scene. | Tester: **Jackson** | Output: All functionality works correctly. |
| **Scenario 2:** The user selects ‘view details’ without selecting a meeting from the below table. | Tester:  David | Output: null pointer is handled and program simply does nothing. |
| **Use Case:** Edit Group | | |
| **Scenario 1:** The user goes to the edit group tab. The user selects a group from a selection of owned groups. The user makes edits to the group’s tags or description. The user clicks the save changes button once finished. A confirmation message is displayed. | Tester: **Jackson** | Output: The functionality works correctly. |
| **Scenario 2:** The user attempts to edit a group. |  |  |
| **2a:** The user does not select a group to edit | Tester: David | Output: error msg displayed |
| **2b:** The user leaves all group tag selections empty. | Tester: **Jackson,**  **David** | Output: error msg displayed |
| **Use Case:** Sign-up for meeting | | |
| **Scenario 1:** The user selects a meeting from the list of meetings they may view, and a meeting details dialogue appears. The user reads all the meeting information and clicks attend meeting. A message appears confirming attendance | Tester: **Jackson** | Output: Functionality works correctly. |
| **Scenario 2:** The user selects a meeting from the list of meetings they may view, and a meeting details dialogue appears. The user reads all the meeting information and clicks attend meeting. |  |  |
| **2a:** The user attempts to sign-up for a meeting the user is already attending. | Tester: **Jackson,**  **David** | Output: The FXML Window says “You are already attending.” |
| **Scenario 3:** The user cancels attendant from a meeting | Tester: **Jackson** | Output: The FXML Window says “No Longer Attending.” |
| **Scenario 4:** The user attempts to cancel attendance from a meeting the user never signed-up for. | Tester: **Jackson, David** | Output: The FXML Window says “You Are Not On The Roster.” |
| **Other functionality** | | |
| **Scenario 1:** The user logs out of the program from every tab that prompts the log-off option. | Tester:  **Nicholas H** | Output: The user is returned to the login screen after logging out from any tab. The account that has been created still exists and allows the user to re log in with the same account they created before. |
| **Scenario 2:** All messages displayed to the user should be cleared after a successful action. | Tester: David, et.al. | Output: Messages displayed to user fade away. |