Group 8 - Bahaar Grewal, Newton Vu, Ese Okabiah, Darius Satterwhite, Danial Sharif

Use your sequence diagrams, functional requirements to generate 8 test cases for your application. Remember Chapter 8 is full of examples of different types of testing. Test cases are written and they demonstrate some features and must pass or fail. They are simple atomic operations directly related to features.

Test 1: Unique User ID when Registering

Input:

1. The user input credentials for their account

Tests:

1. Test should ensure that the login userID is unique
2. The user will not be able to register if the userID is already taken

Output:

1. The system will reject any user ID that already exists and the system will prompt the user to input another user ID that is not in the database.
2. If the userID has not been used and all of the boxes are filled, the system will accept the user credentials and put it into the database.
3. Test will pass if user inputs a valid and unique user ID for their registration for their account.
4. Test will fail if the user is able to input an existing user ID that is already in the database

Test 2: Login Validation

Input:

1. The user enters their login credentials

Tests:

1. Test should ensure that the login credentials are valid and match the database
2. The credentials should match with what the user created when they registered in the registration form

Output:

1. If valid credentials, the user will be logged into the application
2. If the credentials are not valid, the user will need to re-enter their credentials
3. Test will pass if the system blocks the user from access if credentials are invalid
4. Test will fail if the system allows the user into any account with incorrect credentials

Test 3: Checking & Deleting off Task

Input:

1.The user enters a task.

Tests:

1. Test to make sure that the user is able to check off tasks when completed.

2. Test to make sure that the user is able to delete tasks off the list.

Output:

1. Test passes when the task is checked off with a line through text or the task is deleted off the list. When a task is finished, the user is able to check off a task which is indicated by a line through the task description. When the user is ready to delete a task, the user should be able to click the ‘x’ next to the task to successfully delete the chosen task off the list.
2. Test fails when the task is not checked off or not deleted off the list. Therefore when a task is clicked a line doesn’t go through the task indicating that it's not checked off. Also if the user clicks the ‘x’ near the task, the task isn’t deleted off the list.

Test 4: Adding task to the list

Input:

1. User enters a task.

Tests:

1. Test to make sure that the application adds tasks to the list when entered.
2. Test to see if user is able to view all tasks

Output:

1. Test passes when the tasks are added to the list and the user is able to see all tasks added.When user clicks add task or presses enter key, the task should be added to the list of tasks.

2. Test fails when the tasks are not added to the list and the user is unable to see all tasks. If the user clicks add task, and the task is not added when clicked or when they press enter key.

Test 5: Notification sounds

Input:

1. The user sets a notification for a task.

Tests:

1. Test to ensure that the notification sound is played when the notification is triggered. Test to verify that the user can choose different notification sounds from the settings.

Output:

1. The notification sound plays when the notification is triggered meaning the test is passed.Test is failed when no sound is played
2. The user can select different notification sounds from the settings. Test is passed when the user is able to choose different notification sounds. Test fails when the user is unable to choose sounds.

Test 6: Editing tasks

Input:

1. The user edits the task.

Tests:

1. Test to see if user is able to edit the name or description of the task.
2. Test should update in the list to show the edited task.

Output:

User is able to edit the task name and it shall update on the list itself. Test passes when the user can edit the task name by clicking the editing icon. Test fails when edit icon fails to edit the task name.

Test 7: Notification timing

Input:

1. The user sets a notification timing.

Tests:

1. Test to see if the user can set the timing amongst the 4 options. (30m, 1h, 2h, 3h)
2. Test to see if the user is notified within the selected time.

Output:

The notification is set to the selected time and the user is notified by the selected time. Test passes when notification happens according to the selected time. Test fails when notification doesn't happen according to selected time.

Test 8: Saved tasks per userID

Input:

1. The user's task information will still be stored under their login credentials.

Tests:

1. Test to ensure tasks are saved under the user's unique credentials
2. Test to see whether the task is still present after a page refresh

3. Test to see whether other UserIDs will be able to access a different user's task while not logged in as them. (Other users should not be able to edit or check any task that is not theirs, or linked to their userID)

Output:

1. The tasks entered will be available under the user's login information. Test passes when user logs back in and still sees tasks saved.
2. Test fails when the user logs back in and doesn't see the tasks saved. The entered tasks will not be saved and text box for each task will be empty