

Sprint 1:

- US2 --
 - Start App
 - Click on Sign Up; type;
 - email= <fred@fred.com>
 - Password = <hahaha>
 - Password confirmation = <hahaha>
 - Click box next to terms and conditions
 - Click Register button
 -
- US3 --
 - Start App
 - Click on Login; type;
 - email= <fred@fred.com>
 - Password = <hahaha>
 - Click Login
 - Click the pencil button on the sidebar; type;
 - “Hi this is my first post”
 - Click Post button
 - User will see the Post

Sprint 2:

- Function for Add friend button
 - Login
 - Click on

Each team member picks a module or module and lists the equivalence classes and the test cases selected to cover all equivalence classes.

Sprint 1:

(3)As a user, I want to be able to post information about myself and post my thoughts so that others will know who I am and what I am thinking

I worked on grabbing a logged in user's posts from the database so I tested how the posts were received. Since there is only one option, receiving a post the equivalence class was trivial.

Equivalence class Grab a Post:

Test Case 1: Grab only one post.

Test Case 2: Grab all posts.

When I finished this equivalence class I grabbed all of the logged in user's posts.

Sprint 2:

(7) As a user I want to be able to post comments on someone else's post

I worked on the functionality of getting the comment text stored in the database.

Equivalence class: Storing a comment

I typed in a comment on a post and then hit the submit button and then I checked the firebase dashboard to see if the comment object was present

(8) As a user I want to like another person's post

I worked on the functionality of storing a like object in the database

Equivalence class: Storing a like

I the like button on a post and then I checked the firebase dashboard to see if the like object was present

Sprint 3:

(3) As a user, I want my posts to be deleted after a certain amount of time so that way my past won't come back to haunt me.

- I worked on the functionality of scheduling firebase to run a certain task on a certain time interval:

Equivalence class: Wrote a function that printed a simple message and had my scheduling function call it.

If I saw the message show up in my console I knew that the scheduler function works.

- I worked on the functionality to delete a function.

Equivalence class: delete a function manually

For this class I created a button on the home page that when pressed the delete function would be called.

If I no longer saw the post in the firebase database after I pressed the button on the home page that means that my function worked correctly

Equivalence class: Have my scheduler function call the delete post function

For this class I reduced the time interval of the scheduler function to 5 seconds and called the delete post function inside the scheduler function

If I no longer saw the post in the firebase database after 10 seconds then that means the function worked correctly

(4) As a user, I want to be able to message my friends so that way I can talk to them privately.

- I worked on writing a function to query for messages from a user's friends:

Equivalence class: Wrote a function that queries database for all messages sent to this user:

For this class I created a button on the user's message page that would call the query function and just iterate through the returned object printing out each element.

If I saw all the messages that a user received on their message page after I clicked the button then I knew that the function worked.

Equivalence class: Wrote a function that queries database for all messages sent to this user from a specific friend:

For this class I created a button on the user's message page that would call the query function and just iterate through the returned object printing out each element along with the friend who sent it.

If I saw all the messages that a user received from a certain friend on their message page after I clicked the button then I knew that the function worked.

Sprint 4:

(1)As a user I want an accurate representation of notifications so that way I am not clicking on past notifications.

- I worked on the functionality of showing the profile icon and embolden the name of the actioner:

Equivalence class: Wrote a function to get a user's icon and their name from a notification object.

If I saw the user's icon and their emboldened name in the notification listing I knew the function worked.