System and Unit Test Report

- Product Name: **ClassNeeds**
- Team Name: **ClassNeeds**
- Date: **July 21, 2020**

Unit Test for Sprint1

A. User story 1.1 from sprint 1: As a student, I want to be able to find the class that I am taking or thinking of taking.

B. User story 1.2 from sprint 1: As a student or professor, I want to see files that have already been uploaded and upload my own.

C. User Story 1.3 from sprint 1: As a student, I want to make sure that the website is secure and only accessible by my peers.

Scenario:

1. Navigate to 'Sign In/Sign Up' tab; select 'New Account'; type
    - Email = <jotywong@ucsc.edu>
    - Password = <abcd>
    - Click Sign Up button or press Enter key
    - Users should see verification messages that they've signed up successfully.
    - Sign in with the previous credentials
2. Navigate to the 'Classes' tab.
3. Users should see a list of classes.
4. Click on CSE 101.
5. Users should see a list of files to download and also be able to upload.
    - Click 'Choose File' and select dummy.txt
    - Select notes button
    - Click 'Submit' button
6. Navigate back to CSE 101

7. Users should see dummy.txt under the notes category.

Unit Test for Sprint2

D. User story 2.1 from sprint 2: As a student, I want to rate and see past ratings for a class.

E. User story 2.1 from sprint 2: As a student, I want to voice my opinion about certain classes and professors. (commenting)

Scenario:

1. Navigate to ‘Sign In/Sign Up’ tab; select ‘Sign In; type
    - Email = <jotywong@ucsc.edu>
    - Password = <abcd>
    - Users should see verification messages that they’ve signed in successfully.
    - Click Ratings tab on Navigation bar
    - Users should see the Rating pages showing Highest Rated Classes and Lowest Rated Classes, also the upvote and downvote buttons for every class.
2. Click any class names
3. It will go to the specific page of the Classes page.
4. Users should be able to see the form for Comments
5. Type any comments to the class and click submit
6. If successful, it will go to Classes page
7. Navigate to the specific class page, users should be able to see the updated comments.
8. Navigate to ‘Ratings’
9. Click upvote/downvote for any class
10. The page should show the updated highest Rated Classes and Lowest Rated Classes after calculating the voting.

Unit Test for Sprint3

F. User story 3.1 from sprint 3: As a student, I would like to see who made each comment (names) and I would like to see a clean site overall.

G. User Story 3.2 from sprint 3: As any user of the site, I would like to have a profile page so I can see what classes I've favorited, rated, and change my password.

H. User Story 3.3 from sprint 3: As a CSE student, I would like to see CSE degree flowcharts so that I can better understand my pathway through my degree.

Scenario:

1. Navigate to 'Sign In/Sign Up' tab; select 'Sign In; type
   - Email = <jotywong@ucsc.edu>
   - Password = <abcd>
   - Users should see verification messages that they've signed in successfully.
   - Click Profile tab on Navigation bar
   - Users should see Toast 'Welcome,jotywong@ucsc.edu'
   - And the form 'Adding your favorite classes'
   - The bottom button Resetting password
2. Click the Resetting password link
3. Users should see two forms for inputting old password and new password, and a 'update' button to confirm.
4. If the old password is not matched with the current one, a flash message will pop up.
5. Otherwise Users can input the new password, and click 'update' will replace the old password to the new one.
6. Navigate back to Profile.
7. Select a favorite from the option list and click submit

8. Users should see the classname that he or she selected showing below the form.

9. Click the favorite class names, it will go to the corresponding pages of the Classes page.

10. Navigate to Curriculum.

11. Users could download the pdf files for the academic schedule

Unit Tests

In addition to the scenarios described above, we do have unit tests located at ClassNeeds/Testing/test_ClassNeeds.py. It can be run from the top directory using ‘nose2 -v’ if you have nose2 pip-ed and set your local db in the script.

At the beginning of every test, the app’s database is set to the global variable LOCAL_DB_URI. The tests seem to run in alphabetical order, but here are descriptions in order of the python file:

# ClassNeeds (main page) ------------------------------------

def test_main_page() - Tests that the main page returns a 200 status code.

# SignUp ------------------------------------

def test_SignUp_invalid_same_email() - Tests for signing up an account with the same email.

def test_SignUp_valid() - Tests for signing up an account.

# SignIn -------------------------------------

def test_SignIn_invalid_twice() - Tests for signing in when already signed in.

def test_SignIn_invalid_email() - Tests for an invalid email.

def test_SignIn_invalid_pw() - Tests for an invalid password.

def test_SignIn_valid() - Tests for signing in successfully.

# SignOut ------------------------------------

def test_SignOut_invalid_not_signed_in() - Tests for signing out when not already signed in.

def test_SignOut_valid() - Tests for signing out successfully.

# Classes ------------------------------------

def test_Classes_invalid_anon() - Tests that an anonymous user cannot see classes.

def test_Classes_valid_POST_CSE_101() - Tests that a good http response is returned for CSE 101. (POST)

def test_Classes_invalid_POST_unknown_class() - Tests for correct response from an unknown class. (POST)

def test_Classes_valid_GET() - Tests that a good http response is returned. (GET)

# ClassDetails ------------------------------------

# Ratings ------------------------------------

def test_Ratings_invalid_anon() - Tests that an anonymous user cannot see ratings.

def test_Ratings_valid_POST() - Tests that a good http response is returned for an upvote to CSE 101. (POST)

def test_Ratings_valid_GET() - Tests that a good http response is returned. (GET)

# Curriculum Charts ------------------------------------

# About ------------------------------------

# Profile ------------------------------------

Unfortunately, we didn’t get to finishing unit tests for every function, but the groundwork is there.