Katlyn Murphy & Benjamin Smith

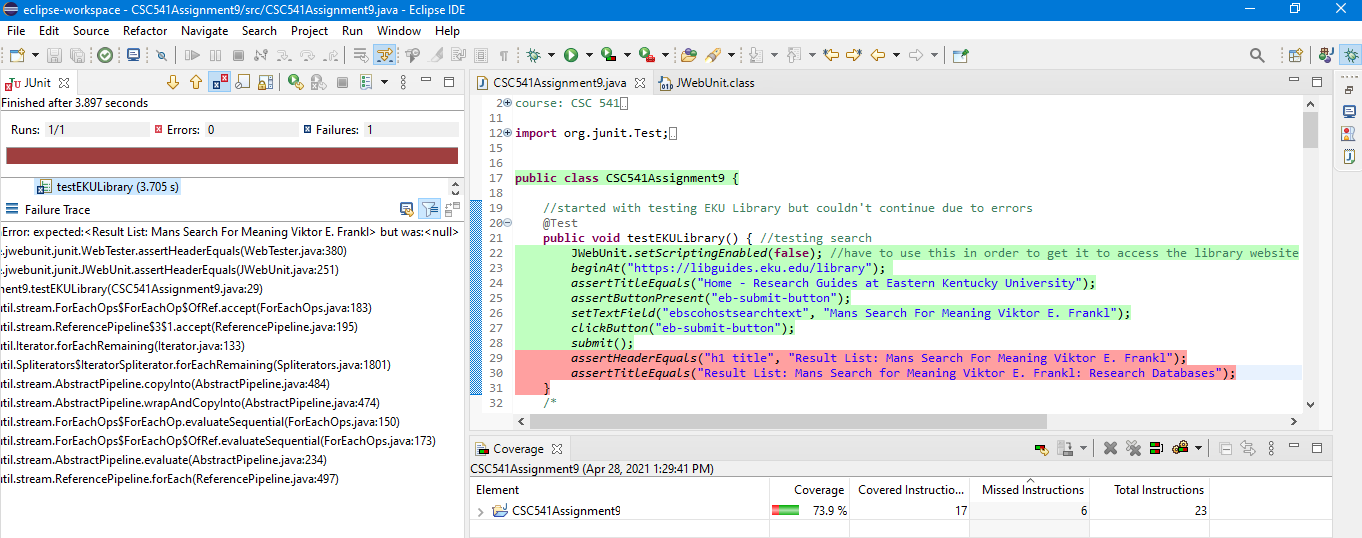
04/25/2021

CSC 541

Assignment 9/10

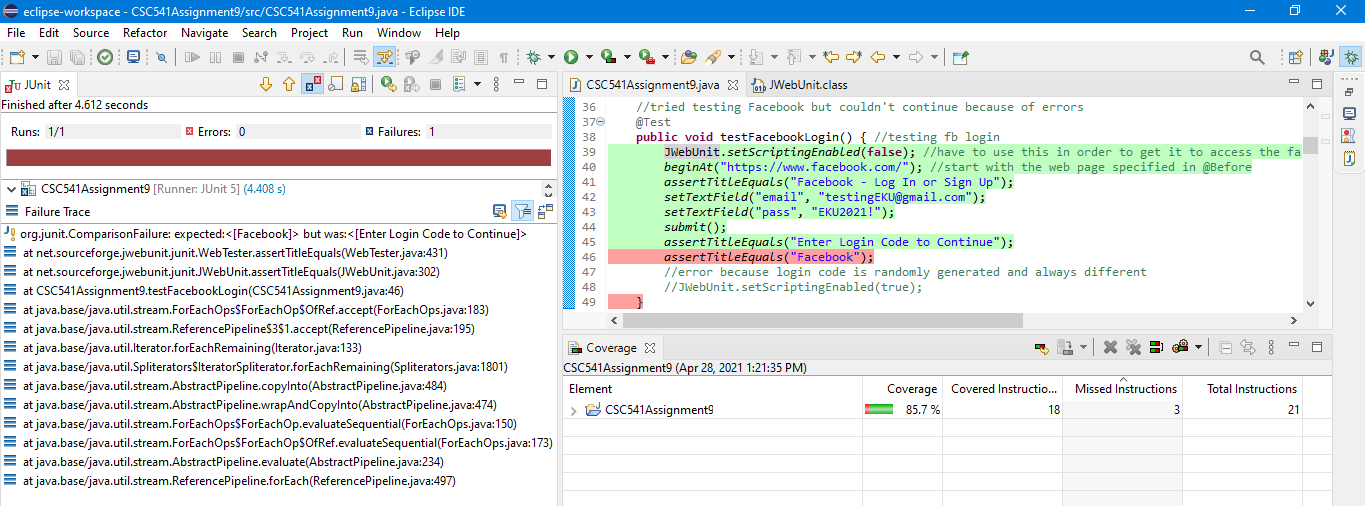
Test Case 1: Testing EKU Library search

This is the first website we started testing and it was unsuccessful. However, we found many errors with this website due to JavaScript errors being thrown with JWebUnit but not in the browser. So, we tried temporarily disabling JavaScript in our test but still ended up with errors, so we search for other websites to test. We tried multiple assertEquals and following elements by xPath but still got errors no matter anything we tried.



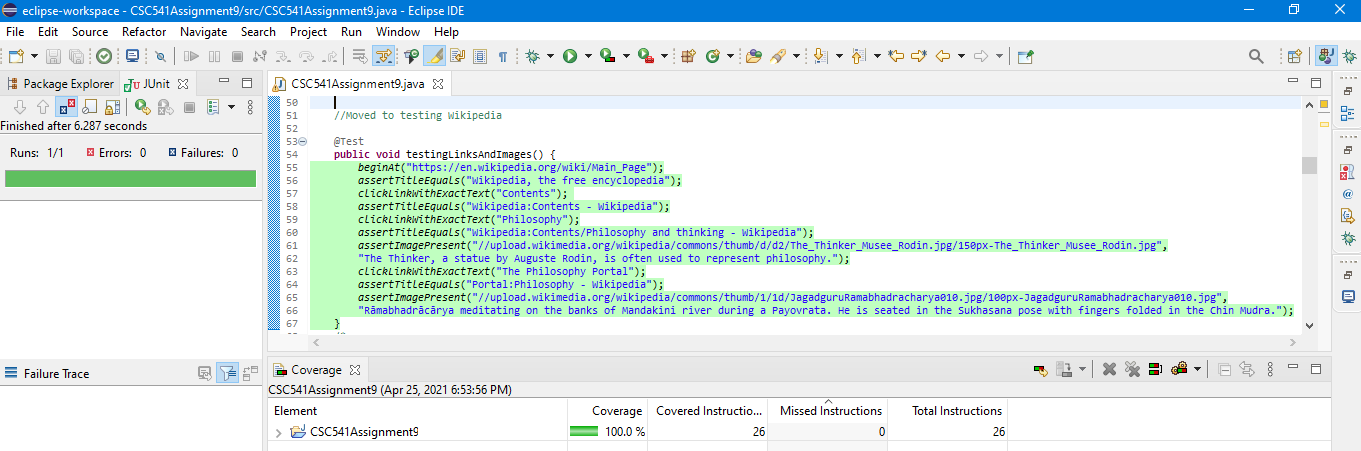
Test Case 2: Testing Facebook login

This is the second unsuccessful website we tried to test. However, this also had JavaScript errors even after disabling so we had to search for other websites again. There was also an error after logging in due to a privacy setting being enabled that requires a login code to be presented, which is similar to Captcha. Thus, we were unable to bypass it but this means the login code is successfully doing it’s job protecting the user.



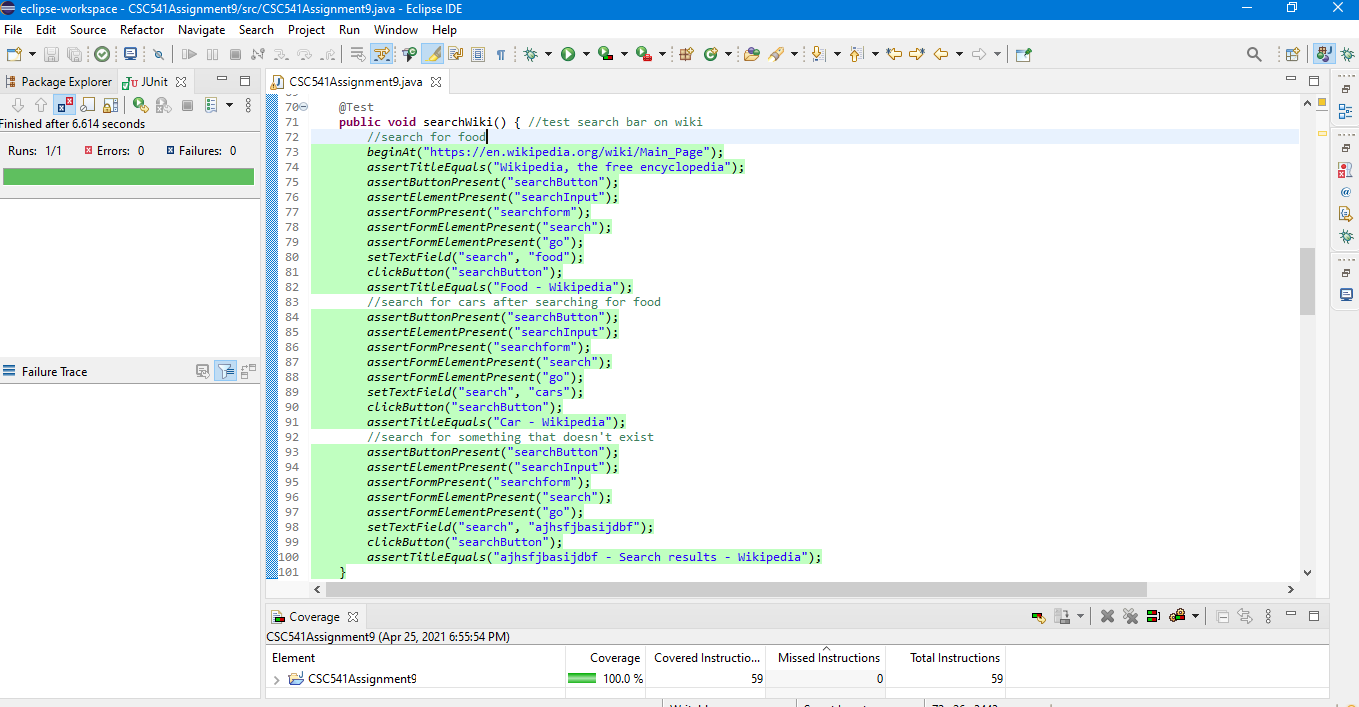
Test Case 3: Testing Wikipedia navigation links and images

Finally, after testing multiple other websites such as (Target, Papa Johns, Pizza Hut, and others not shown), we were successful with testing multiple navigation links and images present through Wikipedia.



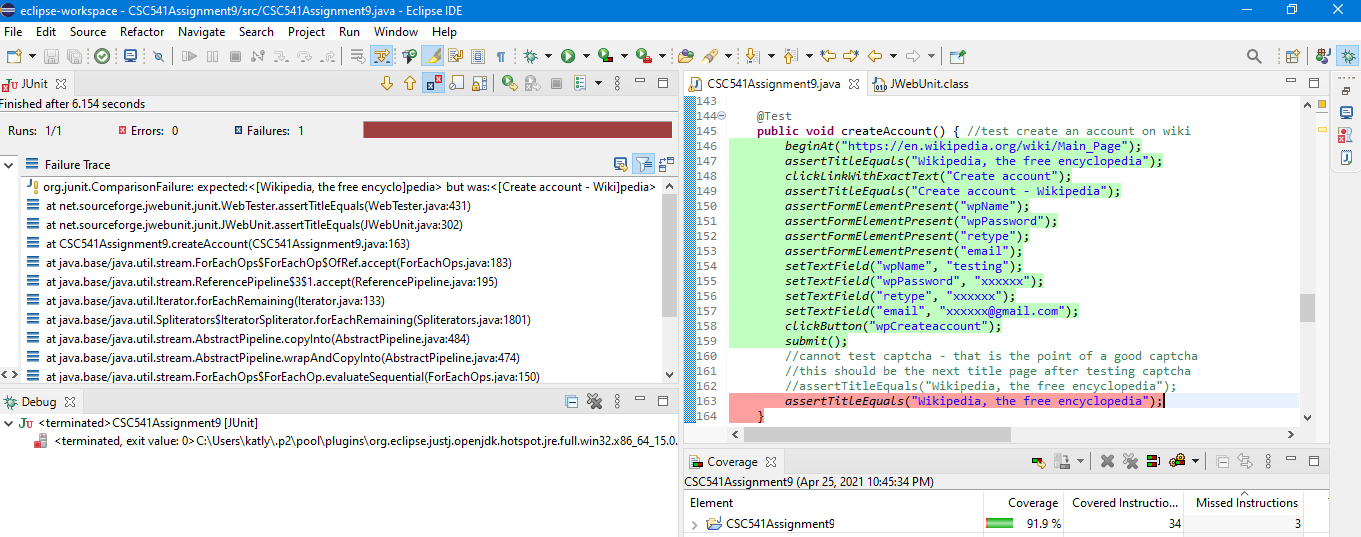
Test Case 4: Testing search on Wikipedia.

We successfully tested multiple search inputs varying from ones that gave us a successful returned Wikipedia webpage on our search input and ones that didn’t return a Wikipedia webpage because it wasn’t present or understood. We tested the search from not only the home page but after already testing a search as well.



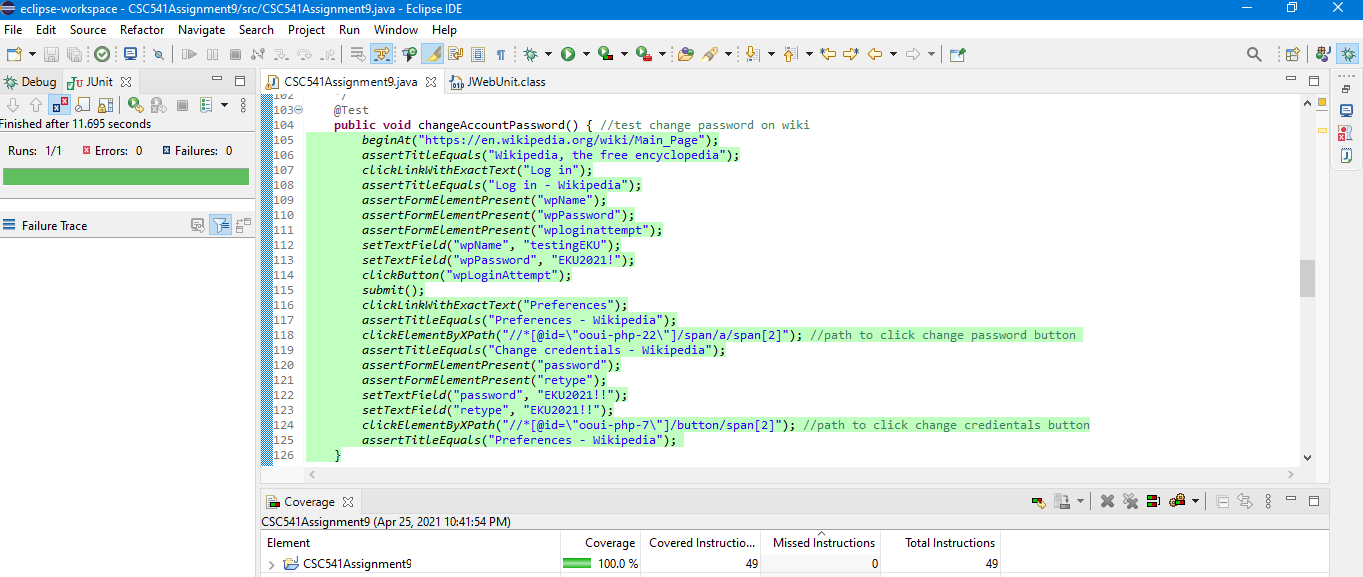
Test Case 5: Testing create account on Wikipedia

This was unsuccessful due to there being a Captcha when creating an account. We tried various ways to test the Captcha, but nothing worked. However, this is good because this means that the Captcha is doing its job at protecting the website from spam and abuse via automated scripts.



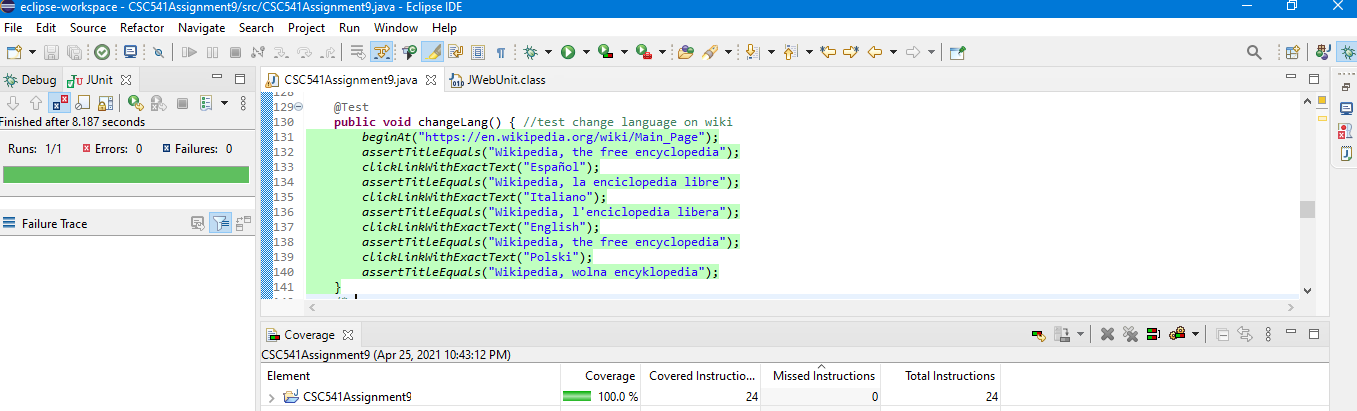
Test Case 6: Testing login and change account password on Wikipedia.

We had to test the log in first before we could change the account password. As you can see from the results, we were successful at that because the link to update our preferences was available and found. We were successful at changing the password in JWebUnit and it also changed our actual password on Wikipedia, which was cool to see.



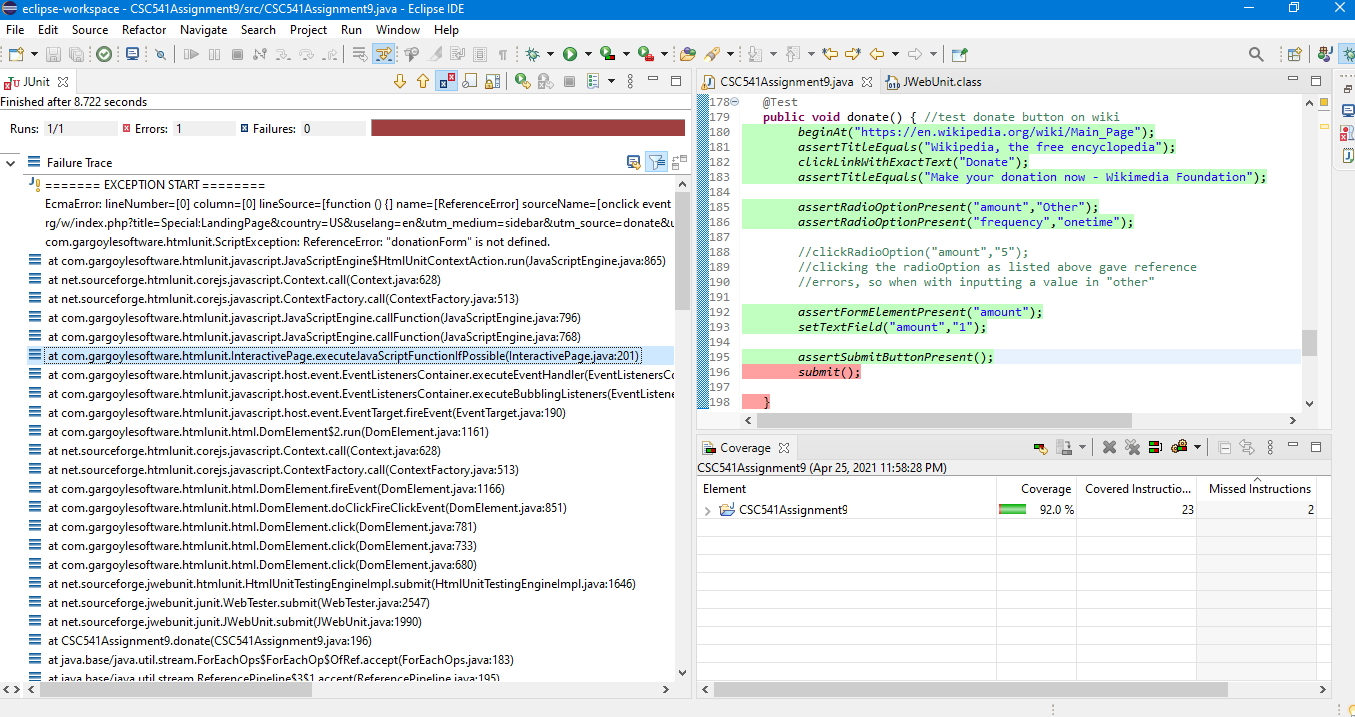
Test Case 7: Testing change language on Wikipedia

We successfully tested changing multiple languages by clicking the links to the different languages and seeing if the title was now in that language.



Test Case 8: Testing donate on Wikipedia

We tried testing the donate multiple ways by even specially clicking “Other ways to give” but we were still unsuccessful. Even after looking at the specific functions that were being called and making sure each one of those elements were given a value, it was impossible for us to figure out why it wouldn’t successfully submit.



Our opinions: We enjoyed working as a team together again and completing this assignment. It made things easier for both people involved and was nice to get various styles of website testing from each person. We think JWebUnit was very straight forward in testing websites and left room for little confusion ONLY if the website didn’t consist of many errors, like Wikipedia. However, it was difficult to find a website that didn’t cause much errors which was very tedious. After doing research, which we found out there was little to none on, we think JWebUnit would be successful in testing your own developed web applications due to being more familiar with the functionality and variables of the web application. For example, the Captcha issue that we encountered we would have been able to disable/workaround this but only in the testing application.