Team Blue
Deliverable #4
Amara
11-15-2016

Our task was to complete the design and implementation of your testing framework as specified in Deliverable #3. You are now ready to test your project in earnest. You will create 25 test cases that your framework will automatically use to test your H/FOSS project.

Process:

We have been able to write 25 test cases that don't involve us using a false database or Django. We have been very limited in methods we can use because they all have database dependencies. The first method we found we can test was the LoggingTimer because it was one of the only classes in the entire repository that does not have database dependencies. We created 10 different test cases for this method testing it against a sleep timer to make sure the method produced the correct results. We also tested this method to make sure that we could induce a fault and we could break this method.

The next method we found that had no database dependencies was the log(msg, *args, **kwargs) method what created a log of the time. Again we were very limited in the methods we could test because we were not able to get all the background dependencies to run. We tested to make sure this method was producing the correct results we also were able to make this method fail our tests by forcing it to produce incorrect results.

Additionally, we tested log and log_nostar with multiple arguments in *args, in order to ensure that it could handle multi-argument calls.

We then tested the cd_to_root_project function in order to see if it retrieves the correct root directory. We did this by seeing if the path generated in the function matched the directory for the function caller, which produced the correct results.

Finally, we checked out if the get_docker_hosts function returned the correct error message, as we could not set up the database. It informed us that the DOCKER_HOSTS ENV variable was not set, which was the expected output.

The last method that we could that did not require a database or Django dependencies was the log_nostar(msg, *args, **kwargs). This method simply took the log methods output and stripped the star from it in order to make the results from the log method look more uniform.

Results:
After analyzing our results we were able to determine that our test cases were working as intended. Some of the issues we ran into were when we wanted to produce an incorrect result it would cause the method to crash. Our results showed that our driver was not able to handle exceptions very well.

Problems:

Amara has proved to be nothing but problems because of dependencies. When we first started writing our test cases we planned on testing items against a false database. This proved to be very complicated because we could not get their database to work because both of us had no experience with Docker. Docker was just the first problem. Once we got Docker running we ran into the problem that would ultimately make it so we couldn't use the Database and that was Django. Both of us spent many hours researching how to make Django work. We were able to get Django installed but could never use it to build our false database. Because of this we had to dig through the Amara repository and find methods that did not use a Django dependency which turned out to be pretty much all of them. Another problem we kept running into was python pathing and how to run methods from the command line with our driver.