CS 1632 – Deliverable 3: Web testing with bdd

Testing Stack Overflow (http://stackoverflow.com)

Created by Philip Ni

CS 1632 Software Testing

University of Pittsburgh

Project Git Repository: https://github.com/phil-nye/CS1632/tree/master/Deliverable3

# **Summary**

For this project, I tested Stack Overflow (<http://stackoverflow.com>) using the BDD testing method, in order to ensure the login, search, tag, and user features were functional.

Stack Overflow contained a “Tour” link and “Advanced Search Tips” list that helped me determine which features should be tested and how they should behave. In addition, many links contained helpful tooltips when hovering over interactive objects. Specific requirements were derived by looking at the HTML source code generated by using the website during exploratory testing.

A user story was created for Login because much of the functionality of the website requires a user account. Asking questions, answering questions, responding to posts, generating reputation, and many other features require a valid user account. As such, I wanted to make sure that a user could login to a valid account to access these features, and invalid account credentials could not access user-only features (e.g. “Ask Question”).

The Search feature also had a user story to verify that advanced search methods worked as defined in the “Advanced Search Tips.” These search tools would allow users to filter out results based on criteria defined by the keywords specified in the tips.

The Tags and Users features were tested to ensure that all users could search by topics or reputable users. The tests in the Tags user story focuses primarily of the ability for the user to search and filter for tags that are interesting to them or the most popular. On the other hand, the Users user story focuses on features that allow users to find the most reputable, newest, up-and-coming users (e.g. user accounts generated in the past month with the highest vote scores), and to find moderators and editors. The Users feature is important in finding the most helpful users.

# **Difficulties**

Many issues were encountered in the process of testing Stack Overflow. Issues encountered included synchronicity problems, dynamic content problems, and issues with using Selenium.

The first problem I encountered was with “HtmlUnitDriver()” in Selenium. This web-driver seemed to be running faster than new webpages could load. I found that the “FirefoxDriver()” ran better by forcing the tests to wait for the pages to load during execution. However, this created a new problem.

Using “FirefoxDriver()” instead of “HtmlUnitDriver()” means that my tests execute slower because the Selenium WebDriver object now has to create a new browser instance and load each page to the screen. It also means that the Mozilla Firefox web browser IS REQUIRED to run the tests.

Furthermore, even with the “FirefoxDriver(),” some lines of the test code would execute too early because the element that was being found depended on further JavaScript code to fully load. I found that adding a Thread object to force the application helped ameliorate these problems. However, the wait time is a constant value, and a page’s load time is dependent on the user’s network connection and the availability of their PC’s system resources. As a result, the wait times may not let the tests pass if you run the tests on a slower PC or network. In all, it takes between 4.5 and 7 mins to run (depending on whether the sites are cached and my network speed) on my system.

A unique issue arose when testing question and answer submissions on Stack Overflow. Because of the stringent submission criteria, I could submit questions or answers to test. All submissions require a certain length and specificity. Additionally, moderators are very quick in removing questions that have already been asked, or that do not fit in with the expectations of a Stack Overflow user. Thus, I could not fully test this feature with the dummy account I created.

Finally, a problem that I have not yet encountered, but expect to encounter, is the issue of changing content. As time passes, the most popular tags/users, search results, etc. will change. I have hopefully fixed this problem by searching for HTML tags and objects that are not specific to the content. Most of the test criteria ensure that only behavior is tested, and not the content. For example, when searching for “selenium” tags, I make sure that a result contains “selenium” in it. Also, when testing for Users filters, I check that information that should be shown to the user is present (e.g. account creation date is shown when filtering newest accounts). Unfortunately, if the site is redesigned, these tests will, likely, no longer function.

*All code, tests, test reports, and resources can be found on:*

<https://github.com/phil-nye/CS1632/tree/master/Deliverable3>

*The site tested is:*

[*http://stackoverflow.com*](http://stackoverflow.com)



