**CS 1632 - DELIVERABLE 3: Web Testing with BDD**

Stack Overflow BDD Web Testing

David Neiman and Luke Nosbisch

https://github.com/lmn33/CS1632\_Assignment3

**Summary**

We chose to test stackoverflow.com for this project, as it provides a lot of testable functionality to users such as logging in and out and navigating between pages. We began testing with the Selenium IDE and found that testing the site was relatively straightforward. As a question and answer platform, Stack Overflow has many static elements and is simple to use.

We began by testing the login and logout features, which are required to actively participate on the site. These test cases included logging in with and without valid credentials as well as logging out. While we tested the success case, we made sure to cover numerous scenarios in which the user should not be able to log in, such as when entering an invalid password. We then tested for navigation from page to page, including cases to and from the main page. Our remaining tests looked at functionalities users would implement within the pages, such as searching in the jobs tab.

In drafting our user stories and scenarios, we focused on the major functionalities that users would need to use the site. We also used exploratory testing to gain familiarity with the site, so that we could understand how to test it better. These steps allowed us to identify important features to test and guided us in writing our test plan.

**Difficulties During the Testing Process**

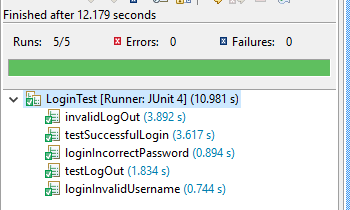
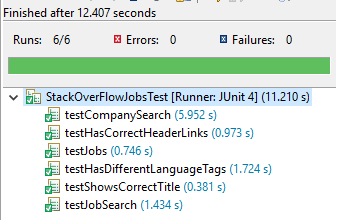
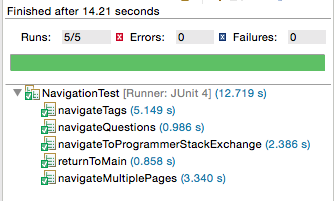
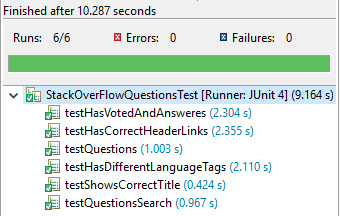
One issue we faced while writing the tests was finding resources on how to use Selenium. While SeleniumHQ has documentation available online, it was not as accessible and easy to use as other resources like the Java API. However, the sample code provided by the instructor was helpful in getting acclimated to Selenium and in showing us possible features to test. Additionally, Eclipse was useful in listing and autocompleting functions. Getting used to the commands was difficult, and it often took multiple attempts to properly write a test case.

Choosing a suitable web site to test was more difficult than we expected. We began by looking at sites such as pittbeer.com and hackthissite.org, but we decided that the features available, dynamic content, and the way in which the sites were coded would make them unfeasible to test for this assignment. It was especially important that the sites be written with identifiers like class names, IDs, and names that would allow individual elements to be effectively searched and located within the code. We decided that Stack Overflow would be an appropriate choice with a range of functionalities to test.

A major challenge was dealing with dynamic content during our testing process. Stack Overflow has a number of dynamic elements that we had adapt our tests to. For example, when entering an incorrect password, a notification is dynamically generated; since this didn't show up in the HTML, we adapted our test to check for remaining on the login page. Dynamic content also prevented us from testing certain features like registration, which weren't particularly amenable to testing using Selenium.

Based on our experiences with this assignment, we would expect testing dynamic content to be a recurring problem in the future. Many web sites use dynamic content extensively for a variety of purposes that make it hard to test. Additionally, even small changes in the site's source code could cause tests to fail; while small changes such as changing the link text could be fixed easily, significant changes to a site could easily lead to having to rewrite existing tests.

Our code is located at: https://github.com/lmn33/CS1632\_Assignment3

**Screenshots**