**CS1699 – DELIVERABLE 3: Web Testing with BDD**

**Website:** Wikipedia (https://en.wikipedia.org/)

**Authors:** Sheridan Zivanovich & Ryan McDonald

As eager as we were to jump into the third phase of our software testing adventures, the amount of freedom we were granted in terms of what we website we could test via BDD was rather daunting. Initially, we struggled to decide on a website that would provide sufficient functionality to test, while simultaneously avoiding complicating requirements. We discussed testing the cs.pitt.edu website; however, the functionality of this website is quite limited and, as such, there would be very little for us to test.

We then moved to the opposite side of the spectrum, considering Twitter as our next candidate. Nonetheless, we quickly realized that many of Twitter’s features required significant interaction between users possessing accounts. As such, it would have been necessary to make several fake Twitter accounts with corresponding fake e-mails for these accounts; when considering the difficulty of both making and managing these accounts, we concluded Twitter was not an appealing option.

Although our valiant effort to attempt to test Twitter was all for naught, since the Internet is a vast domain, our simple lack of direction was our only issue. Due to the over-cumbersome material of a completely different class, one of us had many more Wikipedia articles open than any one human being probably should. With that, we eventually made the decision to test Wikipedia. We found Wikipedia to have significant functionality for non-users, and unlike Twitter, it is very easy to create a dummy Wikipedia account which we could use for testing.

Despite the attractiveness of Wikipedia as a test subject, we came across several difficulties in developing scenarios that would be testable without access to the codebase or the underlying infrastructure. For example, we were unable to test features such as article creation because the Wikipedia community generally frowns upon the creation of frivolous articles, and our test account and/or IP would be quickly blocked from creating further articles. Because of such issues, rather than actually saving pages when editing, we simply verified that certain required items were present when making edits (e.g., the “Save page” button).

To add to our difficulties, as the formatting of Wikipedia is fairly regular across the entire site, it can be difficult to determine whether or not you are simply testing the same functionality multiple times. For example, testing the ability to edit a user page is very similar to testing the ability to edit a regular article. However, since these two scenarios apply to two different user stories, we decided to keep overlapping tests in cases such as this, despite the possibility of redundancy. Similar or not, the functionality of these pages acting independently would certainly be important for users who are browsing the site.

Going forward, we expect that it would eventually become necessary to test certain scenarios on a non-production server. This would allow for the testing of article creation, modification, deletion, and other similar features without having to worry about affecting the integrity of pre-existing data or having our IP blocked. Without the ability to conduct this type of testing, we certainly felt limited in our scenario and feature choice but did our best to get as close as we could to verifying such features. With such capabilities, we could ensure that Wikipedia would be more robust than an M1A2 SEP battle tank through our BDD testing.

Code can be found at **https://github.com/sdzivanovich/cs1699\_deliverable3**.

