University of Pittsburgh

CS 1699 – FINAL DELIVERABLE

Test of the Coursera Website

Rinat Gatyatullin, Daniel Mahoney

CS 1699

Bill Laboon

14 April 2015

**Summary:**

For the final deliverable we choose to test a website called Coursera, we created web tests in Junit to accomplish this. We utilized Selenium web browser automation to create scripts for our tests.

Coursera provides college level courses on line, some of which are free. The majority of the courses offered are technology centered; this attracted us to the site. Users can participate in courses that are taught at established institutions around the country. Coursera users can watch every lecture, complete assignments, and participate in exams that the students enrolled in the course complete. The University of Pittsburgh has courses offered on the site.

We decided to test four separate features of the Coursera: user log in/out, account settings/profile changes, searching for a course, and adding/dropping courses.

**Issues:**

One of the major issues we faced while created tests classes was capturing input information on forms. Coursera auto saves user input on their forms, and does not have a button to submit the form. This made capturing and confirming input changes more difficult. Ultimately we used a "CSS selector" method to capture input data fields in a form. However this methodology did not work for every field, in particular drop down lists would not return text values, so we could not verify the selected value.

Coursera also used many modal pop up frames on their site. Initially this made it hard to retrieve/input information into the pop frame. We determine that we needed to change the focus of our web driver using methods we found in the Selenium Web Driver API.

**Additional Tests:**

Since Coursera is on the internet it will be subject to malicious users, and potentially high volumes of network traffic. We would like to outsource penetration testing to some crafty hacker, to expose any security vulnerabilities. Also we would like to conduct performance testing on Coursera, to see how it handles large amounts of traffic.

**Assessment of Quality:**

Coursera did not fail any of the test cases we created. We believe that it is a very good product, and has working functionality. We believe Coursera is ready to be released into a live environment.

We would still recommend penetration and performance testing, but the lack of this form of testing should not push back the Coursera deadline. There is no sensitive information about users contained in Coursera' s database, and if the web site goes down users will have to wait to continue learning their free class.

Our code is located at: https://github.com/rtg8/Coursera-Web-Testing

