**https://www.grubhub.com/**

**Adam Darr, Michael Stoltz**

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

**https://github.com/adamdarr/deliverable-3**

**Summary**

For our project we decided to test GrubHub.com. We chose to test this website because we wanted to see how a popular website used by college students would hold up under system tests. Also we were hungry.

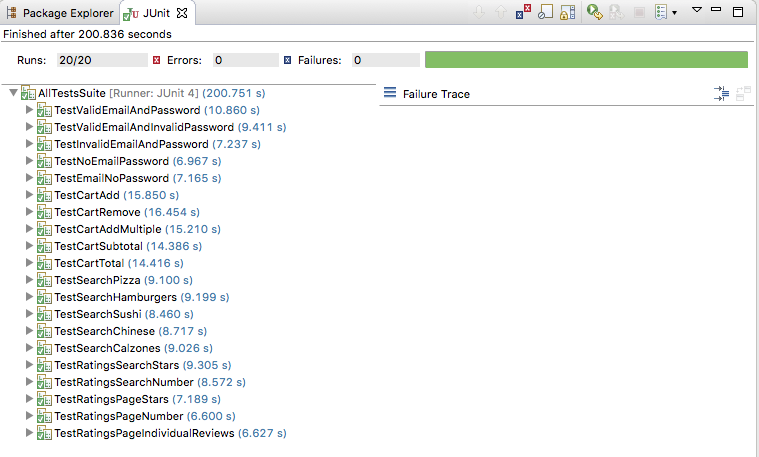
In order to test GrubHub, we wanted to pick user stories and scenarios that would be likely to occur. Our first user story involves testing GrubHub’s login system. For this story, we wanted to ensure that a user could only gain access to a profile if they had the correct login information. Additionally, we wanted to test that there was proper error handling of invalid email addresses and passwords. For our second user story, we decided to test shopping cart modifications. For this user story, we added scenarios that ensured that adding and removing items from the cart would appear on-screen and the appropriate count of items and subtotals changed.

In our third and fourth tests, we attempted to ensure that searches for food categories as well as the ratings system worked properly. To test food category searches we tested multiple categories and made sure that the results matched the category. For the ratings system scenarios, we assured that searches as well as individual pages listed ratings and a count of the number of ratings. Additionally, we ensured that individual reviews showed up on the page for restaurants.

**Testing Concerns**

We faced a variety of issues when writing tests and using the Selenium IDE. At first, it was difficult for us to write user stories and scenarios since we did not have a firm understanding of how they related to one and other. However, after looking at course slides, any discrepancies were cleared immediately. Furthermore, there was a learning curve with the Selenium IDE. It proved difficult to figure out some of the assertions and other commands since there was such a large library. The built-in reference, however, was able to resolve the issue. Additionally, dealing with JavaScript and using Selenium proved difficult due to the dynamic nature of JavaScript.

Another of the major problems we ran into was testing for counts and comparing values on the page. To do this, we discovered that we could store values using the variety of store commands provided by the Selenium IDE. We then used these stored values to compare with other values using ternary statements with the assertEval command. Our final issue came about when converting our Selenium tests to JUnit with WebDriver. While setting up the environment in Eclipse was fairly straightforward, many of our tests failed at first due to differences between the converted JUnit tests and the tests we ran through the Selenium IDE in our browser. We were eventually able to get our tests to pass, but only after realizing the subtle differences between running the tests as JUnits with WebDriver and running them directly through the Selenium IDE in Firefox.

**Screenshot of Test Results**

**User Stories & Scenarios**

**1.** As a user I want to log in so that I don’t have to enter my information every time I make an order

* Given a correct email address and password when logging in then I should be see that I am now logged in in the top bar
* Given a correct email address and incorrect password when logging in then I should see a message telling me I entered the wrong information
* Given an invalid email address and password when logging in then I should see a message telling me I entered the wrong information
* Given no email is entered when logging in then I should be prompted to enter an email
* Given no password is entered when logging in then I should be prompted to enter a password

**2.** As a user I want to have a shopping cart so that I can manage my order

* Given an empty shopping cart when I add an item to it then it should appear in the cart
* Given a shopping cart with 1 item in it when I remove that item then it should no longer be in the cart
* Given an empty shopping cart when I add multiple of one item then that quantity should appear in the cart
* Given an empty shopping cart when I add multiple of one item then the cart subtotal should be the same as the total price of the multiple items.
* Given a cart that is not empty when the subtotal, delivery fee, and sales tax are added together then they should equal the total price

**3.** As a user I want to search for food categories so that I can narrow my selection

* Given I type in a search for “pizza” when I press the search button then I’m given a results page with pizza restaurants
* Given I type in a search for “hamburgers” when I press the search button then I’m given a results page with hamburger restaurants
* Given I type in a search for “sushi” when I press the search button then I’m given a results page with sushi restaurants
* Given I type in a search for “Chinese” when I press the search button then I’m given a results page with Chinese restaurants
* wor

**4.** As a user I want to be able see ratings of restaurants so that I can make a more informed decision

* Given I enter a search for restaurants in my area (15213) when I see the results then I should be able to see the ratings (out of five stars) for each of the results
* Given I enter a search for restaurants in my area (15213) when I see the results then I should be able to see the number of ratings for each of the results
* Given I click on a restaurant page when the page loads then I should be able to see the rating (out of five stars)
* Given I click on a restaurant page when the page loads then I should be able to see the number of ratings for the restaurant
* Given I click on a restaurant page and the restaurant has reviews when the page loads then I should be able to see the individual reviews submitted by users