INF SCI 2955 – Final DELIVERABLE:

Web Testing with Behavior Driven Test for Yelp

Weichuan Hong

July 2014

Contents

[**1. Summary** 3](#_Toc393566602)

[**2. Description** 4](#_Toc393566603)

[2.1 Issues 4](#_Toc393566604)

[2.2 Additional tests 5](#_Toc393566605)

[2.3 Assessment 6](#_Toc393566606)

[**3. Test Results (Screen shot)** 7](#_Toc393566607)

**1. Summary**

Yelp's website, Yelp.com, is a crowd-sourced local business review and social networking site. [The Register](http://en.wikipedia.org/wiki/The_Register) described it as an online city guide. The site has pages devoted to individual locations, such as restaurants or schools, where anyone can submit a review on them.  It uses a one to five star rating system. In addition to writing reviews, users can react to reviews, plan events or discuss their personal lives. According to Sterling Market Intelligence, Yelp is "one of the most important sites on the Internet.” Yelp is primarily active in major metropolitan regions. It has 132 million monthly visitors and 57 million reviews. As of 2010, there were local Yelp pages in 33 cities.

The reason I want to test Yelp is not because Yelp has any specific functions that I am interested in testing. Instead, Yelp possesses so much general functions which are quite similar to the other website, which is why I want to test it. Since I have no experience in Web testing, even Web developing, I choose Yelp to try different general tests on functions, such as searching, logging, writing a review etc. Of course, the most important reason I want to do the Web testing is because it’s so amazing, even just watching example in the class.

In this deliverable, I choose BDD because I have no idea about the code behind the pages. BDD can help me confirm what I am trying to test and it’s also unambiguous for the other to understand.

**2. Description**

2.1 Issues

2.1.1 Build Testing Environment

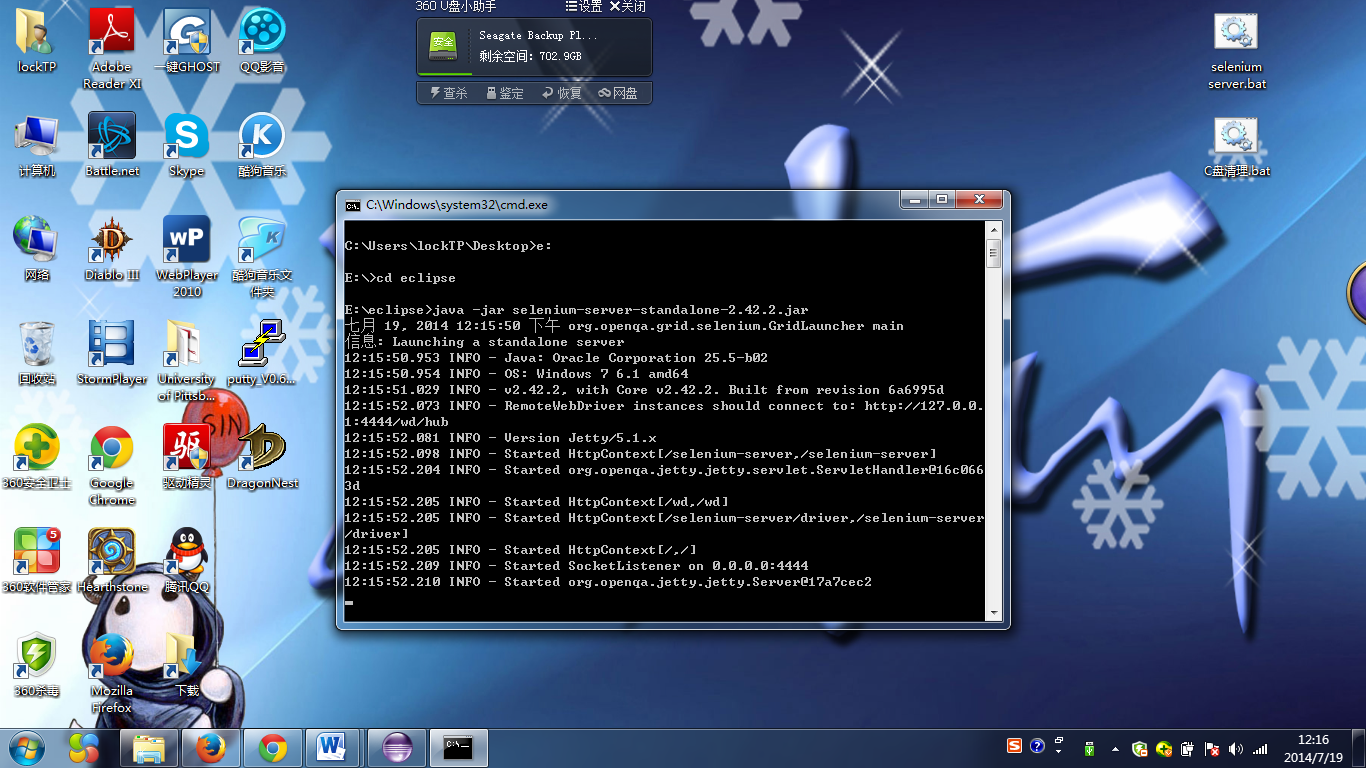
It’s always confused when first tried to do something new. After downloading the Firefox and set up the Selenuim IDE, which all go well, I had no idea how to transfer the code to Java and create local Selenium WebDriver Script.

After searching on the Internet, I successfully overcome this problem. Also, I wrote a .bat profile to execute the Selenium Sever in order to execute to local script.

e:

cd eclipse

java -jar selenium-server-standalone-2.42.2.jar



2.1.2 AssertText, AssertPresent issue

During the test, I found hard to assert whether a text contain a String or not. For example, after searching for Pittsburgh, the website is supposed to show a text: ‘Browsing Pittsburgh, PA Businesses’. What we search is ‘Pittsburgh’, however, instead of showing ‘Browsing Pittsburgh Businesses’, it add PA automatically after the Pittsburgh, which is good but hard to write testing code. In the code, are we supposed to check which state Pittsburgh is located? How to assert this text?

After checking the API, assertText need to match the exact what we need to assert and assertPresent checks the whole page.

Therefore, Instead of just use the Selenuim IDE to record and export as java profile, I tried to write the assertion in java directly. Finally, I found a way to assert this text by using the regular expressions.

Pattern p = **null**;

*assertTrue*(p.*matches*(".\*"+place+".\*", selenium.getText("css=h1")));

This method can assert whether a text contain a String or not.

2.1.3 Wait the page to load issue

During the test, after every waitForPageToLoad method has been executed, the program still fails when I assert the text. However, they should be passed because I certify it manually. I am curious why. After I add another sentence, the assertion passed.

selenium.waitForPageToLoad("30000"); //What the original code contains.

Thread.*sleep*(3000); // What I add.

I think it's because at the moment the page changes, the previous page is still there. When we assert it, the assertion is not true.

2.1.4 Regular expressions

Regular expressions are hard to memorize, I can still only use some simple regular expressions in the step definition. Therefore, lots of the step definitions are not universal and I need to write lots of @Given, @When, @Then to meet the scenarios.

2.1.4 Scenarios relations

With given conditions become more and more complex, the code that wrote in the @Given become more and more miscellaneous, which are hard to execute during testing. For example, ‘given the member is logged in’, I need to execute the whole log in process in @Given part. It takes too much time and resources. I think there should be any other way to make this @Given part.

2.1.5 Difficult to assert status

Because I haven’t learned any web development before, I have no idea how to assert status. For example, how to certify that a member has been logged in? Of course, during the manual testing it’s easy to find out whether we are logged in or not. However, it’s hard to certify in code, especially for me, a totally new learner. Therefore, I just assert the text on the page that show that I have been logged in.

2.2 Additional tests

2.2.1 More professional testing skills

This deliverable is supposed to hand up a professional test report. But actually, I am not testing the website, I am learning how to use the Web testing. Or, in other words, I am using to website to test my Web testing code. So there must be more professional test suite to test these functions in Yelp.

2.2.2 More test on the other functions

There are only 10 scenarios in the test suite, which is not enough at all even to test a simple student’s homework website. As for yelp.com, there must be more tests on its tons of functions, including member management, advertisement management, picture management etc.

2.3 Assessment

2.3.1 Failed tests and problem areas

I do fail on lots of tests, but they are all because I am still not familiar with the Web testing. After changing the code, most of them have passed.

2.3.2 Red/yellow/green template

Data base: green

The database is steady and reliable. It never failed down during testing.

Front end: green

The front end is friendly. It possesses all requirements that meet customer’s need.

Reliability: green

The system never breaks down during testing.

2.3.3 Overall assessment

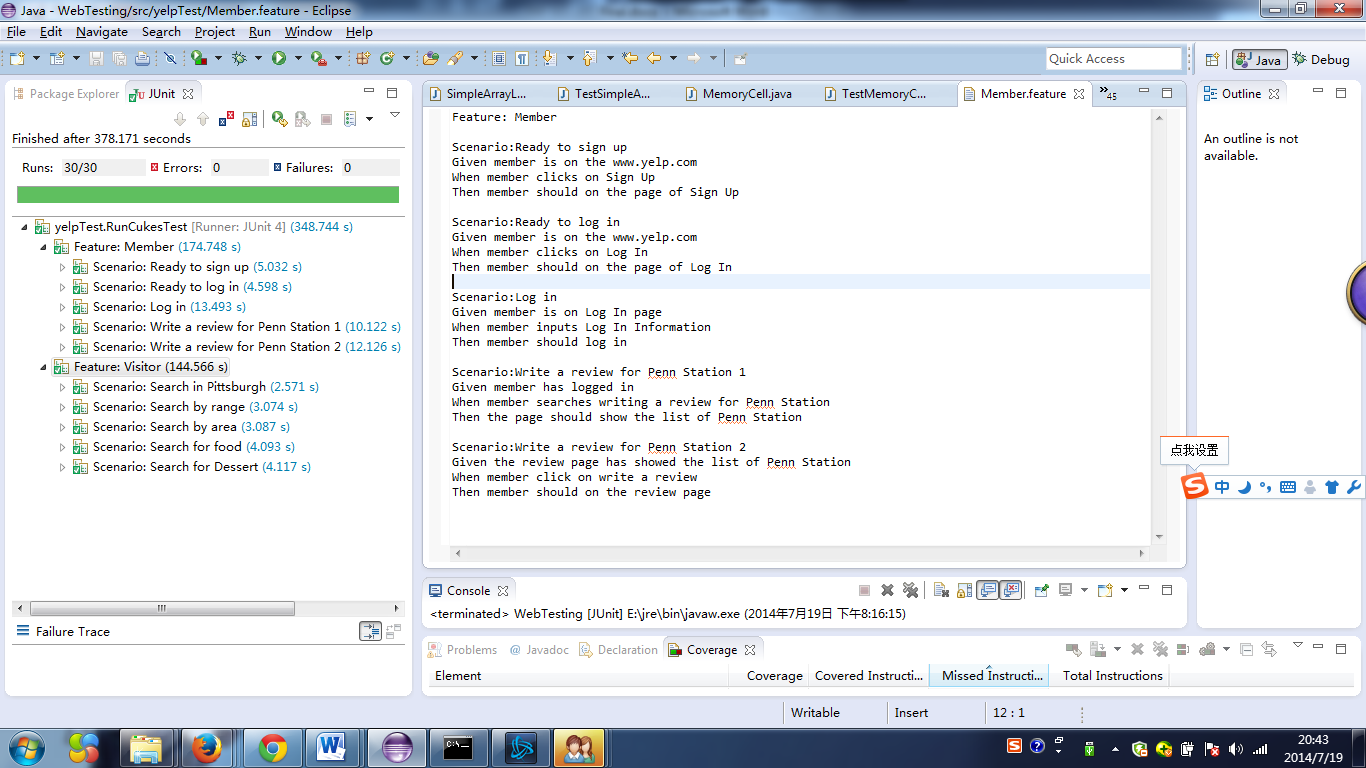
There is no doubt that yelp.com is mature enough to meet the customer’s need, it possesses adequate and reliable functions which can totally satisfy all the visitors to accomplish their needs.

2.3.4 Recommendation

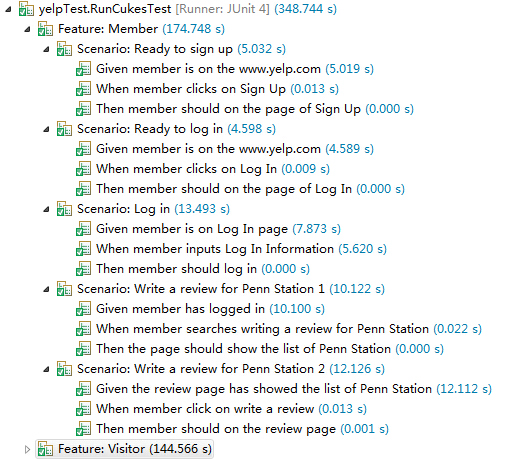
This product is sufficient enough to be released. Actually, it has been successfully released for a long time.

**3. Test Results (Screen shot)**

Screen shot:



Feature: Member



Feature: Visitor

