Erika Boquist

COSC 603

3.5.2013

Project 2

**Application URL:** http://3.boquiststutterwebapp.appspot.com/

**Google Repository:** cosc603-boquist-project2

**What I learned:**

I learned a lot in this project. The biggest thing I learned was the usefulness of repositories and source control. I mistakenly neglected to read all the directions for this project and just started coding and when I was almost done, read that I had to use Google Code again. Upon attempting to commit to the new repository, I accidentally deleted my local repo and lost all my work. ☹ Needless to say, I learned how important it is to follow directions and always ALWAYS use source control.

I also learned just how difficult it is working with code that isn’t modular. My last project I failed to split my project into more methods. I was tempted to use the professor’s provided project 1 code, but then thought I might learn more using mine, and knowing where my failures were. That being said, I most definitely learned that when your code isn’t modular, it’s awful trying to make it web-enabled! To the contrary, using the sample web project was very simple because their code was well documented and their methods were clearly defined.

Another thing I learned in this project was that it is very easy to make mistakes in Eclipse. I had set up my Google Code repo for this project and proceeded with my initial commit, upon continuing to make frequent check-ins, I finally noticed that I was no longer committing to my remote repo, rather my local repo! So that would also explain why I do not seem to have many commits. I’ve also provided a screen shot of my Git Staging Area with the reflogs to show that I was really trying to commit!

**Hardest Part:**

The most difficult part of this project was setting up the sample project with GWT. I’ve used GWT before for a few other classes, but it was mostly using their designer, not actual coding. When I had finished setting up the sample project, I was a little lost on how to actually run it because I had already deployed it to the App Engine, so there were two web applications available to run in Eclipse.

The second thing that was difficult was sifting through all the sample code that was provided. I’ve only ever programmed client-side so having to work with both client and server side was entirely new to me. Seeing how everything interacted was great though, and honestly, all the coding I do at work where I get thrown into a project that I’m not familiar has provided me with an ability to de-cypher something like the sample code.

**Code Modifications:**

There were a few changes I had to make to my code for this project. The first one was to fix the fact that in project 1 I was missing any output in the event that there are no stuttered words. A user would appreciate having some kind of feedback other than an empty string. The second change I had to make was regarding input and output. Instead of accepting a filepath and reading a file line by line, I had to accept raw text, and put it into an array of lines which could then be passed into the rest of my code. Regarding output, instead of printing out each stuttered word as it was found, I had to store the words so I could return all the stuttered words to the client.

The modifications to the sample code were also minimal, I just had to change the textbox to the text area, change the validation of user input, and return different data generated by my stutter method. This code modification was the easiest part of the entire project.

**Application Testing:**

The first round of testing my application to make sure it met the requirements was simply running the application. Verifying that it behaved correctly in the expected state(i.e. text with stuttered words) as well as the exceptions(i.e. no text, no stuttered words, invalid text).

The second round of testing was in code inspection, I had to verify that all the specific code requirements(i.e. text area instead of text box, checking user input validity).

**UI Improvements:**

I think the interface for this project is very clean and to the point. It serves its purpose efficiently. An improvement could be changing the color scheme or maybe incorporating some images since user respond to graphics more than texts. Also, I think it would be good to have their text be evaluated in real time. Essentially letting the user know they are stuttering words as they enter them.