Team Deliverable #3

Overview

The script for our testing automation is complete and our driver is near completion in its entirety. Over all the automation is complete and the future focus of our team is the refining and cleaning up of our code and presentation. The next step is to complete the driver fully. Then to code a few failures into the Sakai source code to ensure that the testing is comparing returned results with the oracles for the test cases. Finally we will clean up how the results are displayed to a user of our testing framework.

File Structure

/Commonly-Named

```
/TestAutomation

      /docs (contains all relevant documents, including README.txt, final report,
presentation, and poster
           README.txt

      /oracles (unused)

      /project (contains all necessary files from Sakai needed)
           /bin
           /src
                StringUtil.java

      /reports (unused)
```

```
    /scripts (contains all scripts used by the project)
        runAllScripts.sh

    /temp (holds temporary files, such as the method outputs and the test report)
        Report.html
        Test01.txt
        Test02.txt
        ...

    /testCaseExecutables (contains all drivers used by the project)
        ContainsDriver.java
        ToStringCourseDriver.java
        ToStringStudentConstructorOne.java
        ToStringStudentConstructorTwoDriver.java
        ToStringUserDriver.java
        TrimLowerDriver.java

    /testCases (contains all test cases as .txt files)
        Test01.txt
        Test02.txt
        ...
        Test25.txt
```

The Driver

Summary

Each driver consists of one class file. Each class being tested has its own driver which will pass the parameters to the method being tested and save its output to a .txt file in the /temp directory.

Difficulties

The hardest issue we faced was the vast labyrinth of dependencies that many of the classes within Sakai posses. We have expended a large amount of effort to ensure that the methods and classes being tested could be tested independent of the project itself. Many of the dependencies were from within the project itself, some are dependencies with networking and server connections, but few are only dependent on Java packages. These dependencies would require the installation of the entire Sakai program, Apache Tomcat, and Maven before the method could instantiated and tested.

We also ran into issues with our standardization of our test case files. The original template for our test cases did not provide an easy method of parsing and readability for others to examine or create new test cases.

The Script

The script begins by deleting all files in the /temp directory. It then proceeds to compile all .java files in both the /project/src and /testCaseExecutables directories, ensuring that all relevant java files are executable. A template for a html table is formatted into a new file called Report.html. Next the script should read each .txt file in the /testCases directory line by line using a for loop. As an individual test case file is read, each line is placed into an element of a string array. The script should now gather the relevant information to run the driver, such as the driver's file name and the parameters to be passed. Once the driver has been executed, the script checks the /temp directory for a .txt with the same name as the test case. The file is read into the next empty element of the array and compared to the expected output defined by the test case. The result of this comparison is stored in the final element of the array. Finally, each array element is formatted into a html table row and added to

Report.html located in the /temp directory. After the script has looped through each test case, it completes the Report.html and opens it in the default web browser.

Sample Test Cases

Test Case Template

test case #

requirement being tested

CLASS.METHOD beimg tested

java driver file

expected outcome

inputs to the method (DELIMETED BY ",")

Test01

Search the list set to null for null element

StringUtil.contains

ContainsDriver.java

false

null:null

Test02

Test Empty least for element test.

StringUtil.contains

ContainsDriver.java

false

:test

Test03

Valid list containing one element which is being searched for.

StringUtil.contains

ContainsDriver.java

true

test:test

Test04

Valid list containing elements

StringUtil.contains

ContainsDriver.java

false

test1:test2

Test05

Valid list containing elements

StringUtil.contains

ContainsDriver.java

false

test7:test1,test2,test3,test4

Test06

Valid list containing elements

StringUtil.contains

ContainsDriver.java

true

test3:test1,test2,test3,test4