Team Polandball

Banking System Phase I - Test Plan

Course Code: CSCI 3060/SOFE 3980U - Software Quality Assurance
Course Instructor: Dr. Jeremy Bradbury

Members:
Dominick Mancini - **100517944**
Scott McLean - **100379538**
Janahan Nirmalan - **100459567**

Tutorial CRN: **73386 - Tuesday, 2:10-3:30**
Report Due Date: **February 14th, 2016**

Items and Order

The testing on the bank account frontend will make use of extensive testing off all areas of the 11 features of the bank account. These tests are intended to observe how the program handles input that is valid or invalid. The 11 features are to be tested in the following order:

1. login - logging into a session
2. create - creating a new account
3. deposit - depositing to an account
4. paybill - paying a bill from a specified account
5. withdrawal - withdrawing money from an account
6. transfer - transferring money from one account to another
7. changeplan - changing a bank account plan
8. disable - disabling an account
9. enable - enabling an account
10. delete - deleting an account
11. logout - logging out/ending a session

Testing Organization

The tests are organized into folders by feature. Each of these feature folders contains two subfolders, standard and admin, further subdividing the testing into tests conducted in the admin state, and tests conducted in the standard state. Each of these folders is in turn divided into three subfolders; test input, expected output, and actual output. Test input contains the files that hold the test input to the program Expected output contains files that hold what the program is expected to output corresponding the what was input. The output file and the input file have the same name to avoid ambiguity. The actual output folder will contain the program's actual output to the test case, and this output will be compared against the expected output. This hierarchical organization can be visualized using Figure I. Figure II represents the entire tree directory for the tests.

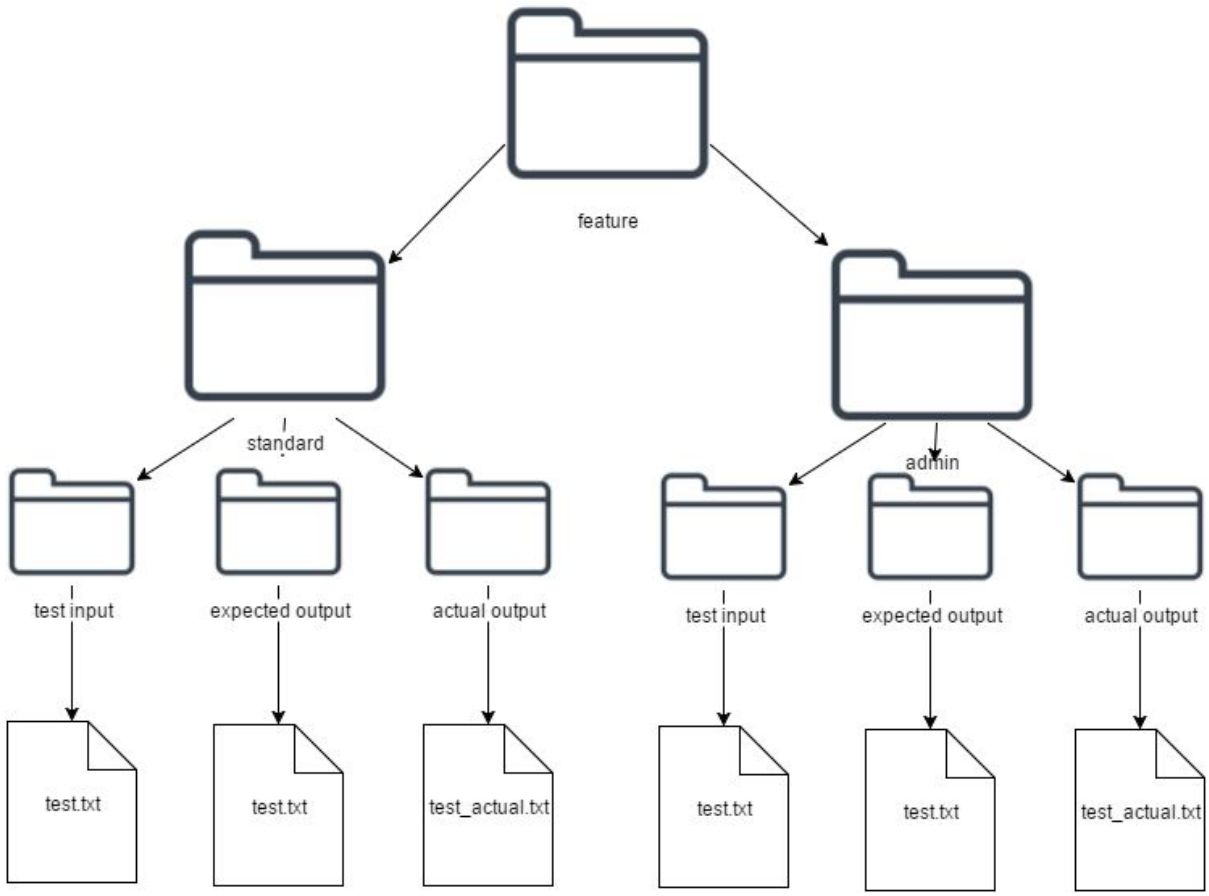

FIGURE I.

```
+---changeplan
|   +---admin
|   |   +---actual output
|   |   +---expected output
|   |   \---test input
|   \---standard
|       +---actual output
|       +---expected output
|       \---test input
+---create
|   +---admin
|   |   +---actual output
|   |   +---expected output
|   |   \---test input
|   \---standard
|       +---actual output
|       +---expected output
|       \---test input
+---delete
|   +---admin
|   |   +---actual output
|   |   +---expected output
|   |   \---test input
|   \---standard
|       +---actual output
|       +---expected output
|       \---test input
+---deposit
|   +---admin
|   |   +---actual output
|   |   +---expected output
|   |   \---test input
|   \---standard
|       +---actual output
|       +---expected output
|       \---test input
+---disable
|   +---admin
|   |   +---actual output
|   |   +---expected output
|   |   \---test input
|   \---standard
|       +---actual output
|       +---expected output
|       \---test input
+---enable
|   +---admin
|   |   +---actual output
```

```
|   |   +---expected output
|   |   \---test input
|   \---standard
|       +---actual output
|       +---expected output
|       \---test input
+---login
|   +---admin
|   |   +---actual output
|   |   +---expected output
|   |   \---test input
|   \---standard
|       +---actual output
|       +---expected output
|       \---test input
+---logout
|   +---admin
|   |   +---actual output
|   |   +---expected output
|   |   \---test input
|   \---standard
|       +---actual output
|       +---expected output
|       \---test input
+---paybill
|   +---admin
|   |   +---actual output
|   |   +---expected output
|   |   \---test input
|   \---standard
|       +---actual output
|       +---expected output
|       \---test input
+---transfer
|   +---admin
|   |   +---actual output
|   |   +---expected output
|   |   \---test input
|   \---standard
|       +---actual output
|       +---expected output
|       \---test input
\---withdrawal
    +---admin
    |   +---actual output
    |   +---expected output
    |   \---test input
    \---standard
```

```
        +---actual output
        +---expected output
        \---test input
```

FIGURE II

Testing Method and Environment

The testing method for the frontend will consist of a series of Shell scripts written for Linux Bash, and will pipe the input files into the program, and then store the program output into a the actual output folder for the appropriate feature and standard/admin state. The testing operating system will be either Ubuntu 15.10 or Kubuntu 15.10, using Linux Kernel version 4.2.0 The shell script will then use the “diff” command to differentiate between the expected and actual output. Ideally, if there is a difference between the expected and actual output, that will indicate that the program has failed the test case, as the actual output would not equal the expected output.