**Test Input and Output Organization**

There is a folder called “test\_case\_files” that contains all the test input and output files, and the are going to be grouped based on the transactions/commands. This folder also contains a current bank accounts file named “current\_bank\_accounts\_file.txt”. The transactions are login, logout, create, delete, disable, enable, changeplan, deposit, withdrawal, transfer, paybill, help, and quit. In each of these folders will the test cases divded into sub-groups based on what those tests are testing. Inside the sub-groups, there will be an input folder, an output folder, and a bank account transaction file folder.

Each input test file is named with the transaction followed by a number followed by the extension of “in”. For example “login01.in” for the login transaction for test input 1. The output test is named similar as the input test files except it is followed by the extension of “out”. For example “login01.out” for the login transaction for test output file 1. The bank account transaction file outputs are named the name as the input and output files except they have the extension of “tra” For example “login01.tra” for the login transaction for bank account transaction file 1.

**Test Run Plan**

The test run plan is that the program will the ran with the input files within each sub-group. The output files and the bank account transaction files that resulted from the input files being ran in the program will be compared with the expected output files and expected bank account transaction files.

If the outputs do not match, or if the bank account transaction files do not match, then the test will show false. When false is shown, there is a known problem within the program, and the program is not producing the correct results. If output matches, and the bank account transaction files match, then the test will show true. When true is shown, there program is producing the correct results.

All the tests in the sub-group will be run with a script, which will show all the tests within the sub-group with the result of true or false. This can be used to fix the sub-group problems and the scope of the problem can be narrowed down. When all the tests in the sub-group come out with all true results, the sub-group is producing all the correct results. The script will be run with every sub-group. When all tests from every sub-group come out with all true results. The whole program is producing the correct results.

The order in which the sub-groups are run matters as some tests from other sub-groups depend on the working conditions of the other sub-groups. The first two sub-groups that will be tested will be quit and help. These two are tested first as the program needs a way to exit gracefully, and if a user is confused, they could always refer to the help menu. The next two sub-groups that will be tested will be login and logout. These two should be tested next as login and logout will be used in every test case. Next will be followed by create and delete. Create and delete as the following transaction test cases need to test if an account has been just created. Following create and delete will be disable and enable. These two are tested as changeplan, deposit, withdrawl, transfer, and paybill all have test cases that depend on whether an account is disabled or active. The next transaction to be tested would be changeplan as deposit, withdrawl, transfer, and paybill all have fees that depend on the plan of the user.

Next test would be for deposit. This is tested next as withdrawal, transfer, and paybill all need deposit to test if the account can be used for those transactions. Finally, withdrawal, transfer and paybill will be tested. These are tested last as these 3 sub-groups depend on the sub-groups tested before them.