

CSCI 3060U - Software Quality and Assurance

Project Phase I - Front-End Requirements (Test Plan)

Wenbo Zhang, Xuan Zheng, Neel samirkumar Shah, Dev rajivkumar Thaker

Test Plan

1.] Tests Directory Organization:

The tests will be organized in the following directory structure (generalized):

tests/

```
├── login/
│   ├── login01/
│   │   ├── input_data/
│   │   ├── expected_outputs/
│   │   ├── login01.sh
│   │   └── login01_output.txt (generated after the test is run)
│   ├── login02/
│   │   ├── input_data/
│   │   ├── expected_outputs/
│   │   ├── login02.sh
│   │   └── login02_output.txt (generated after the test is run)
│   └── run_login_tests.sh
├── run_tests.sh (driver shell script)
└── test.log
```

The test cases will be stored in the tests/ directory, with each transaction type (e.g., login) having its dedicated subdirectory. Each test case will be organized into its own directory, containing input data, expected output data, and a shell script to execute the test. Once the test is run, the output will

be stored in a text file.

Additionally, a driver shell script (run_tests.sh) will be implemented inside the tests folder which will be responsible for running the shell scripts for each transaction folders (ex. run_login_tests.sh). Each transaction shell script will be responsible to run all the test cases for the given transaction(ex. login01.sh, login02.sh, etc).

2.] How the Tests will be executed:

Each Test will have a corresponding shell script(ex. login01.sh) that will be responsible for the following:

- Preparing the environment (if necessary).
- Running the program or feature being tested.
- Comparing the actual output with the expected output.

3.] How the output will be stored and organized:

The output of each test will be stored in a text file called 'test_output.txt' inside each test case directory.

The difference between the expected output and the actual output will be captured in a file named comparison_result.txt. If there are no differences, it will indicate that the test passed.

Additionally, a final test report will be logged on the command line as well as stored in test.log file, The final test report will include the following:

- A summary of whether the test passed or failed.
- Details on any failures or mismatches in the output.
- Any additional logs or debug information for investigation.

Requirements Problems

- Transfer/Pay bill:
 - Since there is no given standard for this type, the industry standard is used in this project: the transfer/pay bill process with zero or negative amount and missing/invalid information will be rejected. Also, for bill payment, the user only needs to use the company name abbreviation (EC, CQ and FI) to finish the process instead of the company account number
- Change plan:
 - The requirements do not clearly state whether a user can switch both ways between Student Plan (SP) and Non-Student Plan (NP). Initially, our test cases assumed that a user could only switch from SP to NP, but later when the client clarified about that problem we added test cases to allow for both directions.

- The transaction log format was unclear on how to represent a plan change. We assumed that the last two characters (MM) in the transaction log could store the updated plan (SP or NP).