Test Plan Document

CSCI 3060U

Test Organization

Tests will be organized into a testing folder. Within the testing folder we will have a directory that holds all the cases, named cases, and then a directory that contains completed tests. The cases directory will act as a reference for all test cases. The automated system will grab input and output files from this directory each time a test is requested to run. The system will then created a new folder in the *tests* directory named after the date and time the test was run (there will never be two folders containing the exact same time because multiple tests can be selected to run through the interface). Inside each test date stamped folder there will be subdirectories of the action types that were selected to test. This allows tests to be compared from different dates for different reporting and analysis measures.

The naming conventions for tests are as follows: each test has a test name and category. The category is the 'Action Type' in the table, which refers to the transaction or background process that is being tested. The name is a specific descriptor of what is being tested. For example, below we use 'create' as the action type and 'create_fs_user' as the name of the test.

Expand on the results file.

```
└── testing
    ├── cases
    │   └── create
    │       └── create_fs_user
    │           ├── input.txt
    │           └── output.txt
    └── tests
        └── 2024-01-30:11:54
            ├── buy
            │   └── buy_ss_fail
            │       ├── input.txt
            │       ├── output.txt
            │       └── results.txt
            └── create
                └── create_fs_user
```

```
├── input.txt
├── output.txt
└── results.txt
```

Test Automation

There will be a series of shell scripts that will automate the testing process. Using makefiles, we will have commands that run the tests based on the input, output files and writes the results to a new text file. The scripts will also take care of creating the file structure for the testing suite. Organizing each case that is run into the aforementioned outline.

Example commands:

- make test -all
- make test -create -delete

The option will exist to run all tests or run tests by categories. These parameters will be flags input with the command.

Test Table

The testing table serves as an outline for test cases. The table outlines each test category, name, explanation of the operation, required input, and expected output. In the initial half of the table each test is specifically laid out in the different user types. To save time later tests were only outlined with a note that all users will be tested for the specific action type and operation.

Accessibility Standards

Accessibility standards were not specified in the requirements document. Standards will have to be defined by the internal team and expressed in the development of the front end.

Confirmations

Currently, the system does not include confirmation in testing. As it was not asked in the requirements by the client this system does not ask to confirm actions. In the future, the system can be expanded to confirm on any write, or business intensive action.