02/17/20
CS5387

# Test Plan #***8818*** Review

The test plan introduces the plan with a brief overview and purpose of what the TableSorter class is. However, due to an unclear explanation, the introduction could result confusing to follow for a person with no prior knowledge of the problem or no expertise in object oriented programming.

The test plan continues by describing the behavior and responsibilities of the Table class, by listing its method signatures. Following this, it states the requirements for the TableSorter class. It provides the method signatures of the two methods to be tested. The last part of the test plan is 8 test cases, each with a brief explanation of what they need to accomplish. These test cases do not come with a testing approach or any information of how to use with the code provided which will be explained in later sections of this review. Throughout the whole document there is no mention of JUnit, or the exact testing approach. It seems the test plan is based on coding the appropriate tables then looking at the output to verify it.

---

Notes on the **testing plan**:

- The exit criteria for the test plan is never stated, a tester might not know when the program has passed the testing.
- The testing plan doesn't account for errors that might arise or any other problem a tester might come across, such as an outdated import or missing classes within the testing files.
- The testing approach is unclear, and there is little to no information on how to actually test the cases. It doesn't provide a number of test cases to run, it only provides a general idea of what the test case should look like. This leaves room for interpretation on how many tests to run, therefore making its results hard to compare when tested by different testers.
- The test plan doesn't explicitly specify which type of testing techniques will be used. However an experienced tester could come to the conclusion that this is a black box testing plan. Also, one of the techniques it uses is: boundary value testing.
- ○ In summary, this test plan did not contain enough information. The test cases provided were not specific enough, and just provided a general idea on how to set up a valid test case with its expected output.

---

In conclusion the test plan was not well structured and lacked important elements such as an exit criteria, user interface testing, and a clear testing approach. The code was incompatible with this particular test plan. However, it would be possible to conduct the test cases in the test plan if most of the code (8827) provided was replaced. The code hard coded JUnit tests to matrix objects, and some of the tests were not necessary/included in the test plan.