LO4 - Evaluate the limitations of a given testing process, using statistical methods where appropriate, and summarise outcomes

This document will look to analyse the limitations of the tests that were carried out on the software.

Limitations and potential improvements:

Despite covering a wide range of appropriate techniques well I will still bound by testing only a select few key components of the software.

Limitations of testing framework: Despite originally using Mockito and it still being included as a dependency in the source code pom.xml file it was removed from the final testing suite. This was due to my unfamiliarity with the framework making it impractical to work with within the time constraints of the task. Mockito would've been used to create mock data that could more rigorously test the software.

No test data automation: Currently all the test data is hand-crafted. This is a major limitation we would be able to have more confidence in our results.

Poor Commenting: When I finished the tests, I realised I lacked a thorough standard of documentation. At first this wasn't a problem but as the test suite increased in scale, and while coming back to work on it at a later date it the codebase was less clear. This reduced my efficiency and would also hinder adding more developers to help work on the project.

Creating targets for the test performances:

- All tests must pass. This is obligatory to meet the requirements laid out in LO1 Supporting Document (functional testing).
- Total code coverage of all components testing is required. Every line of code has to be executed to ensure that there are no errors (Structural Testing).
- We must have 100% control flow coverage, which means all input combinations must be covered (Combinatorial Testing).