Test review on code #6858

Version <1.0>

<February 19, 2020>

**Document Control**

**Approval**

The Guidance Team and the customer shall approve this document.

**Document Change Control**

|  |  |
| --- | --- |
| Initial Release: | 1.0 |
| Current Release: |  |
| Indicator of Last Page in Document: |  |
| Date of Last Review: |  |
| Date of Next Review: |  |
| Target Date for Next Update: |  |

**Distribution List**

This following list of people shall receive a copy of this document every time a new version of this document becomes available:

Guidance Team Members:

Dr. Roach

Software Team Members:

**Change Summary**

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1.0 | 02/15/2020 |  |  |
|  |  |  |  |
|  |  |  |  |

Note: The template presented in this document was taken from:

Donaldson, S., and S. Siegel, *Successful Software Development*. Upper Saddle River, NJ: Prentice Hall, 2001, pp. 321-323.

Note: The template presented in this document was taken from: Donaldson, S., and S. Siegel, *Successful Software Development*. Upper Saddle River, NJ: Prentice Hall, 2001, pp. 321-323 and modified by Humberto Mendoza and Steve Roach.

Supplementary information is from:

Pfleeger, S. *Software Engineering, Theory and Practice*. Upper Saddle River, NJ: Prentice Hall, 1998, p. 365.

**Table of Contents**

[1. Review 4](#_Toc33017729)

[1.1. Code documentation 4](#_Toc33017730)

[1.2. Code 4](#_Toc33017731)

[1.3. Applying the test plan from 9914b 4](#_Toc33017732)

[2. Concluding Remarks 4](#_Toc33017733)

# Review

This section reviews the code

## Code documentation

The code is well documented and it is easy to tell what each piece of code is supposed to be doing. The code is easy to understand, there are some comments which may not be necessary like “Get size of table” since it’s easy to know what the code is doing. The grammar on the documentation is correct, and it is very easy to know what the documentation is trying to tell you.

## Code

The code is well modularized in many different methods. When running the code, the code prints the table checks if sorted, if not then sorts it and prints the process of doing that.

There is a try-catch exception wrapped through the code in main method, I would advise this against this as any exception will be giving you the same error of there being a problem with input of number of elements.

The method notSortedVector confused me and I wasn’t sure what the piece of code was supposed to do, that method needs a little more documentation.

I would make the methods notSortedVector and sortingVector private methods because it’s good practice and since they are not being used by other classes currently.

## Applying the test plan from 9914b

When applying the test plan from 9914b, we can tell that checking if the code is sorted or unsorted is working. All of tests given by the test plan passed.

We also know that testing for positive and negative values work, including the number 0.

I’m unsure if the sortable code works because there was no test in the test plan which checks for that. 2 by 2 and 3 by 3 tables work. Other table sizes also needed to be tested to make sure they work, like a table size of 0 or 1.

# Concluding Remarks

There are two public methods and methods helping the public methods and running the code will automatically use a test table to check if the code works. The code is well written so that it’s easy to understand for the most part, though there were a few redundant comments not needed and it’d be nice to have a few more comments at some parts.

Running the test plan tells us that the method isSortable works, but it is not entirely convincing that every edge case works. There were still some edge cases that needed to be tested. The test plan did not test the sortable code.