University of Texas at El Paso

**Test Plan**

Test Plan Report

**Course**

Software Integration and V&V

<<author name redacted>>

**Supervisor**

Dr. Steven Roach

Feb 2020

Design the Test Strategy:

In this test, we want to develop a test set for the **TableSort** class. This class is responsible to check if a given table is sort or not and if it is not sorted, it must sort the table.

In this project, we use another class. Its name is Table. This class was written by the instructor and we don’t have any plan to test that class in this test.

Define Test Objectives

The objectives of this test are finding any error and bugs in this class by comparing the given table and check the result with the expected one.

We will try to focus on some specific patterns that have more chances to break the program or cause any mismatching between the given table and the expected one.

Define Test Criteria:

In this test, we use JUnit to automate the test and be sure about the accuracy of the results. So, to run this test, the user must be sure he/she already has this package added to the java library path.

User can find the package here if he/she doesn’t have it:

<https://junit.org/>

In this test program, we try to test TableSorter by different inputs. The inputs chose in a way that covers all different failure scenarios. You can see the correct inputs and wrong inputs below:

private int[] userInput\_1 = new int[] { 16, 66, 87, 92, 123, 125, 182, 190, 201 }; private int[] userInput\_2 = new int[] { 16, 16, 17, 92, 92, 125, 182, 201, 201 }; private int[] WrongUserInput\_1 = new int[] { 67, 66, 87, 92, 91, 125, 182, 190, 201 }; private int[] WrongUserInput\_2 = new int[] { 16, 66, 87, 15, 123, 125, 182, 190, 180 }; private int[] WrongUserInput\_3 = new int[] { 16, 66, 87, 12, 123, 125, 12, 190, 201 };

Plan Test Environment:

User can run this code on any platform as far as the Java was installed there.

Code:

package assignment1;

import static org.junit.jupiter.api.Assertions.assertTrue; import static org.junit.jupiter.api.Assertions.assertFalse;

import org.junit.jupiter.api.Test;

class TableSorterTest {

private int[] userInput\_1 = new int[] { 16, 66, 87, 92, 123, 125, 182, 190, 201 }; private int[] userInput\_2 = new int[] { 16, 16, 17, 92, 92, 125, 182, 201, 201 }; private int[] WrongUserInput\_1 = new int[] { 67, 66, 87, 92, 91, 125, 182, 190, 201 }; private int[] WrongUserInput\_2 = new int[] { 16, 66, 87, 15, 123, 125, 182, 190, 180 }; private int[] WrongUserInput\_3 = new int[] { 16, 66, 87, 12, 123, 125, 12, 190, 201 };

@Test void testIsSortedNotSorted() throws Exception {

assertTrue(TableSorter.isSorted(new Table(9, userInput\_1))); assertTrue(TableSorter.isSorted(new Table(9, userInput\_2))); assertFalse(TableSorter.isSorted(new Table(9, WrongUserInput\_1))); assertFalse(TableSorter.isSorted(new Table(9, WrongUserInput\_2))); assertFalse(TableSorter.isSorted(new Table(9, WrongUserInput\_3)));

}

@Test void testIsSorted() throws Exception {

assertTrue(TableSorter.isSorted(new Table(9, userInput\_1))); assertTrue(TableSorter.isSorted(new Table(9, userInput\_2))); assertFalse(TableSorter.isSorted(new Table(9, WrongUserInput\_1))); assertFalse(TableSorter.isSorted(new Table(9, WrongUserInput\_2))); assertFalse(TableSorter.isSorted(new Table(9, WrongUserInput\_3)));

}

}