Test Plan Assignment

Test plan

Version 4.0

4/24/2020

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

|  |  |
| --- | --- |
| Initial Release: | 1.0 |
| Current Release: | 4.0 |
| Indicator of Last Page in Document: | END |
| Date of Last Review: | N/A |
| Date of Next Review: | Open |
| Target Date for Next Update: | 4/23/2020 |

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: Emmanuel Jacquez, Mathew Gabel, Sebastian Quinones

Customer: Dr. Roach

Software Team Members: Dr. Roach

Change Summary

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1.0 | 4/7/2020 | Emmanuel Jacquez | Started test plan for db |
| 2.0 | 4/19/2020 | Emmanuel Jacquez | Updated to reflect feedback |
| 3.0 | 4/21/2020 | Emmanuel Jacquez | 3rd update |
| 4.0 | 4/24/2020 | Emmanuel Jacquez | Final Update |

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

[Document Control ii](#_Toc38444054)

[Approval ii](#_Toc38444055)

[Document Change Control ii](#_Toc38444056)

[Distribution List ii](#_Toc38444057)

[Change Summary ii](#_Toc38444058)

[1. Introduction 1](#_Toc38444059)

[1.1. Purpose 1](#_Toc38444060)

[1.2. Scope 1](#_Toc38444061)

[1.3. System Overview 1](#_Toc38444062)

[1.4. Suspension and Exit Criteria 1](#_Toc38444063)

[1.5. Document Overview 1](#_Toc38444064)

[1.6. References 1](#_Toc38444065)

[2. Test Items and Features 2](#_Toc38444066)

[3. Testing Approach 3](#_Toc38444067)

[4. Test Suite 1 4](#_Toc38444068)

[4.1. Test 01 4](#_Toc38444069)

[4.2. Test 02 5](#_Toc38444070)

[4.3. Test 03 6](#_Toc38444071)

[4.4. Test 04 7](#_Toc38444072)

[4.5. Test 05 8](#_Toc38444073)

[5. User Interface Testing 9](#_Toc38444074)

[6. Test Schedule 10](#_Toc38444075)

[7. Other Sections 11](#_Toc38444076)

[8. Appendix 12](#_Toc38444077)

# Introduction

This document contains a test plan designed for the DbEdit.jar program. The test plan will focus on the filter and search functions in the edit menu of the program.

## Purpose

DBEdit program was designed and implemented by Steven Roach at an earlier part in his career and is now having us design a proper test plan for the filter and the search functions found in the edit menu. We want to be able to produce a test plan that is both easy to follow and is able to properly test a program with the goal of verifying and validating its design and purpose.

## Scope

Scope will focus on the current version of the DBEdit program provided by the instructor. As part of Team 2, the scope is narrowed down to the current filter and search functions found in the edit menu.

## System Overview

DBEdit contains functions designed for searching and filtering elements that exist within the tables in the database. After opening an existing table in the database, a user can filter and search through its elements by expanding the Edit menu option available and scrolling down to the search function. A user can then type in and theoretically search for existing elements within the database. Other functionalities of DBEdit that are going to be tested are the “And” and “Or” Filter functions responsible for filtering out elements within columns depending on different search criteria.

## Suspension and Exit Criteria

There is no suspension criteria for this test plan. All of the test that are being performed by this test plan should run. All tests will need to be completed and passed at the conclusion of this test plan .

## Document Overview

The remainder of the document contains a test plan that is to be followed via black box testing by a tester. Section 2 contains a description of the elements that are going to be tested. The main focus of Section 3is the approach that is to be taken during the testing phase in order to verify and validate the current implementation of a function of the DbEdit program.

## References

None

# Test Items and Features

The functions that are going to be tested in the current test plan are:

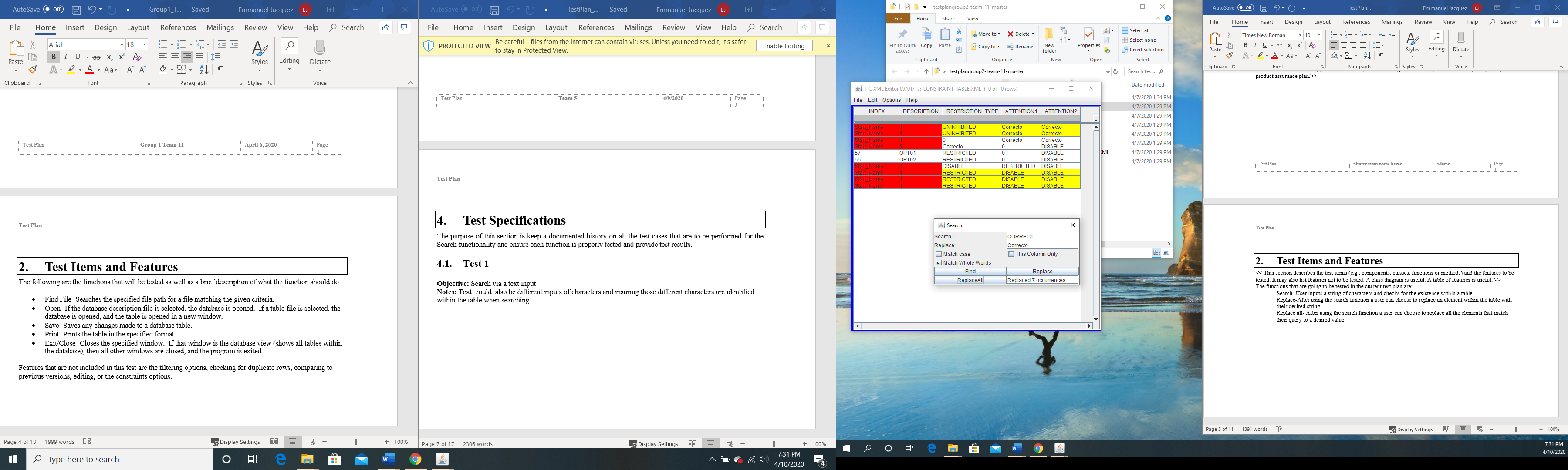
**Search**

**Find**- User inputs a string of characters and checks for the existence within a table **(Figure 1)**

**Replace**-After using the search function a user can choose to replace an element within the table with

their desired string **(Figure 1)**

**Replace all**- After using the search function a user can choose to replace all the elements that match their search. **(Figure 1)**

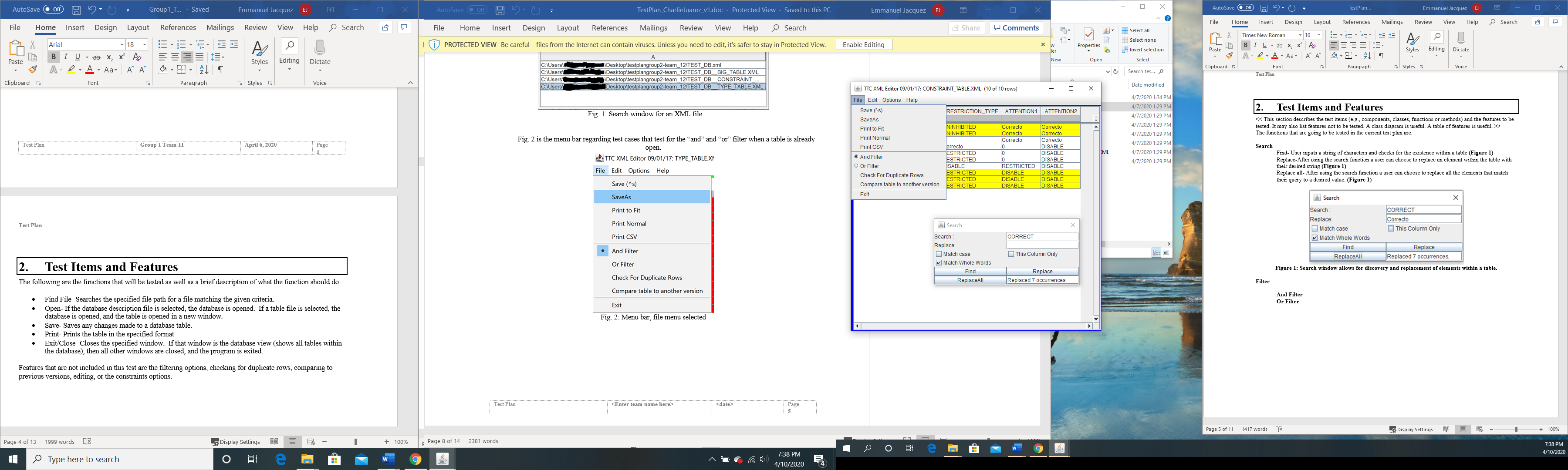


**Figure 1: Search window allows for discovery and replacement of elements within a table.**

**Filter**

**And Filter**-The AND filter (the default) only displays rows that match the selection criteria from every cell in the filter **(Figure 2)**

**Or Filter**-The OR filter displays rows that match the selection criteria of any cell in the filter**. (Figure 2)**



**Figure 2: And/Or Filter options in File menu**

# Testing Approach

The following test plan is to be performed as a black box. Testers should be able to follow and duplicate the tests using the following test plan. Scale ranges from 1-5, with one being the least critical state.

Table 1: Test Plan

|  |  |  |
| --- | --- | --- |
| **TEST SUITE 01** | | |
| **Description of Test Suite** |  | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| 01 | **Search for existing element/s within table via text input** | **5** |
| 02 | **Search and replace an existing element within table using the Replace button** | **5** |
| 03 | **Search and replace all existing elements within table using the ReplaceAll button** | **4** |
| 04 | **Test functionality of And Filter** | **4** |
| 05 | **Test Functionality of Or Filter** | **4** |

# Test Suite 1

## Test 01

**Objective: Search for existing element/s within XML table via text input**

**Notes**: User launches dbEdit.jar. Selects the file dropdown menu. Clicks open and selects the folder Test 1 and then double click on TEST\_DB\_\_CONSTRAINT\_TABLE. Once the file is opened, click on the Edit menu item and select the Search option. The search window will open Figure 1.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 01 | | | | Current Status: Pending | | |
| Test title: Find existing element within xml file via user input | | | | | | |
| Testing approach: Tester will have to have the dbEdit.jar program running and follow the procedure provided in the Notes for this test. (Figure 1) | | | | | | |
| STEP  1 | OPERATOR ACTION  User will enter the text “0” into the textbox found to the right of “Search:”. | PURPOSE  User needs to prepare a valid input in order to search through table. | | | EXEPCTED RESULTS  “0” Shall appear on the edit text box found next to Search: | COMMENTS  None |
| 2 | Tester will click on Find. | User needs to activate find event in order to search table. Every click of find will return the next element that is found in the table. | | | If the element has been found, the program will highlight it with a turquoise color. And display the item location in the matrix using the following notation. Each click will result in the following separate outputs: Found at (2,2), Found at (3,3), Found at (4,3) Found at (5,3) | Once last found element is shown, the program will return the first element found again after clicking Find. |
| Concluding Remarks: Close program at the conclusion of test | | | | | | |
| Testing Team: N/A | | | Date Completed:N/A | | | |

## Test 02

**Objective: Search and replace an existing element within table using the Replace button**

**Notes**: User launches dbEdit.jar. Selects the file dropdown menu Clicks open and selects the folder Test 2 and then double click on TEST\_DB\_\_CONSTRAINT\_TABLE. Once the file is opened, click on the Edit menu item and select the Search option. The search window will open **Figure 1**.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 02 | | | | Current Status: Pending | | |
| Test title: Find and replace an existing element within xml file | | | | | | |
| Testing approach: Tester will have to have the dbEdit.jar program running and follow the procedure provided in the Test 02 Notes. | | | | | | |
| STEP  1 | OPERATOR ACTION  Tester will enter the text “0” into the textbox found to the right of “Search:”. | PURPOSE  User needs to prepare a valid input in order to search and replace through table. | | | EXEPCTED RESULTS  “0” Shall appear on the edit text box found next to Search: after input | COMMENTS  None |
| 2 | Tester will enter the text “Test” to the blank box below the find edit text box. Tester will then click on Replace Button. In order to replace all the “0” values currently in the table, tester will click replace a total of 4 times | This is the text that will replace the element that was input in step 1. Every click to “Replace:” replaces the next item in line with the input text | | | “Test” shall appear in the edit text box found next to “Replace” after input. If the element has been found, the program will replace the “0” with “Test” a total of 4 times. The text “Test” will replace “0” at the locations in the matrix (Found at (2,2),Found at (3,3), Found at (4,3) Found at (5,3) | Each replaced element will be shown highlighted in a turquoise-like color |
| Concluding Remarks: Close program at the conclusion of test | | | | | | |
| Testing Team:  N/A | | | Date Completed: N/A | | | |

## Test 03

**Objective: Search and replace all existing elements within table using the ReplaceAll button**

**Notes**: User launches dbEdit.jar. Selects the file dropdown menu. Clicks open and selects the folder Test 3 and then double click on TEST\_DB\_\_CONSTRAINT\_TABLE. Once the file is opened, click on the Edit menu item and select the Search option. The search window will open Figure 1.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 03 | | | | Current Status: Pending | | |
| Test title: Find and replace an existing element/s within xml file at once using ReplaceAll | | | | | | |
| Testing approach: Tester will have to have the dbEdit.jar program running and follow the procedure provided in the Test 03 Notes. | | | | | | |
| STEP  1 | OPERATOR ACTION  Tester will enter the text “0” into the textbox found to the right of “Search:”. | PURPOSE  User needs to prepare a valid input in order to search and replace through table. | | | EXEPCTED RESULTS  “0” Shall appear on the edit text box found next to Search: after input | COMMENTS  None |
| 2 | Tester will enter the text “Test” to the blank box below the find edit text box. | This is the text that will replace the element that was input in step 1. | | | “Test” shall appear in the edit text box found next to “Replace” after input | None |
| 3 | Tester will click on ReplaceAll Button. In order to replace all the “0” values currently in the table. User only needs to click button once. | ReplaceAll replaces all items matching the Search: input with the input in Replace: | | | If the element has been found, the program will replace the “0” with “Test” a total of 4 times. The text “Replaced 4 occurrences” will appear in the text box on the bottom right corner | Only a single element will be highlighted after all of the elements have been replaced by the test. |
| Concluding Remarks: Close program at the conclusion of test | | | | | | |
| Testing Team:  N/A | | | Date Completed: N/A | | | |

## Test 04

**Objective: Test functionality of And Filter**

**Notes**: User launches dbEdit.jar. Selects the file dropdown menu. Clicks open and selects the folder Test 4 and 5 and then double click on TEST\_DB\_\_CONSTRAINT\_TABLE. Once the file is opened, click on the File menu item and select the And Filter

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No.: 04 | | | Current Status: Pending | | |
| Test title: Functionality of And Filter | | | | | |
| Testing approach: Tester will have to have the dbEdit.jar program running and follow the procedure provided in the Notes for this test | | | | | |
| STEP  1 | OPERATOR ACTION  Tester will enter the text “OPT01” in the grey box below the “Description” column title. Tester will then press enter once they are finished entering the specified text | PURPOSE  User needs to prepare a valid input to filter through the Description Column. Enter activates the function and filters out the data that matches the user input text. | | EXEPCTED RESULTS  “OPT01” Shall appear on the filter box below “Description”. “OPT01” is found and the rest if the table is eliminated/filtered out because the Description column value of OPT01 only occurs once in table. **See Appendix Expected Output 1** | COMMENTS  None |
| Concluding Remarks: Tester does not need to close window or program in order to continue with test 5 | | | | | |
| |  |  | | --- | --- | | Testing Team:  N/A | Date Completed:N/A | | | | | | |

## Test 05

**Objective: Test functionality of Or Filter**

**Notes** User launches dbEdit.jar. Selects the file dropdown menu. Clicks open and selects the folder Test 4 and 5 and then double click on TEST\_DB\_\_CONSTRAINT\_TABLE. Once the file is opened, click on the File menu item and select the Or Filter

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No.: 05 | | | Current Status: Pending | | |
| Test title: Test functionality of Or Filter | | | | | |
| Testing approach: Tester will have to have the dbEdit.jar program running and follow the procedure provided in the Notes for this test | | | | | |
| STEP  1 | OPERATOR ACTION  Tester will enter the text “Start\_Name” in the grey box below the “Index” column title. Tester will then press enter once. | PURPOSE  User needs to prepare a valid input to filter through the Index Column. Most of the current values in the current constraints table have “Start\_Name” as their index. Enter activates the function and filters out the data that matches the user input text from step 1. | | EXEPCTED RESULTS  “Start\_Name” Shall appear on the filter box below “Index”. All elements with “Start Name” are found, a total of 8. | COMMENTS  None |
| 2 | Tester will then enter the text “OPT01” in the grey box below the “Description” column title. Tester will then press enter once they are finished entering text | User needs to prepare a valid input to filter through the Description Column. Enter activates the function and adds data that meets the Or criteria. In this case, the data that also contains OPT01 is added to the current filtered list. | | “OPT01” Shall appear on the filter box below “Description”. Current list that only contained 9 elements that were filtered through only the index value is expanded to include the elements that have OPT01 as their Description value **See Appendix Expected Output 2** | None |
| Concluding Remarks: Close program at the conclusion of test | | | | | |
| |  |  | | --- | --- | | Testing Team: N/A | Date Completed:N/A | | | | | | |

# User Interface Testing

Not Applicable

# Test Schedule

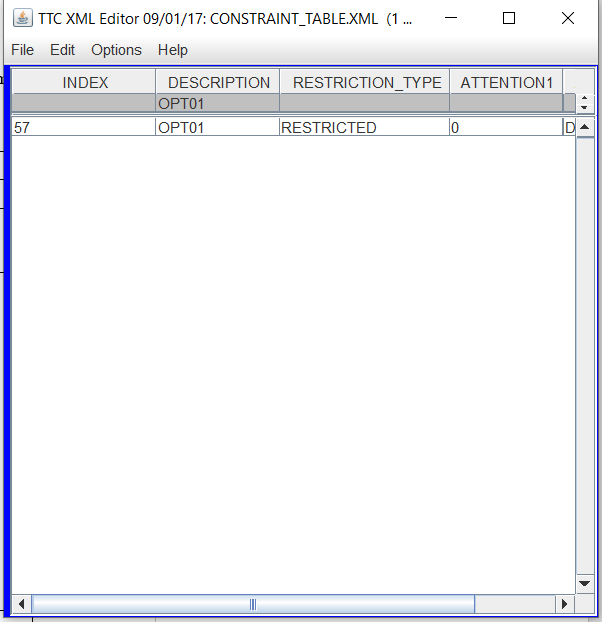
Not Applicable

# Other Sections

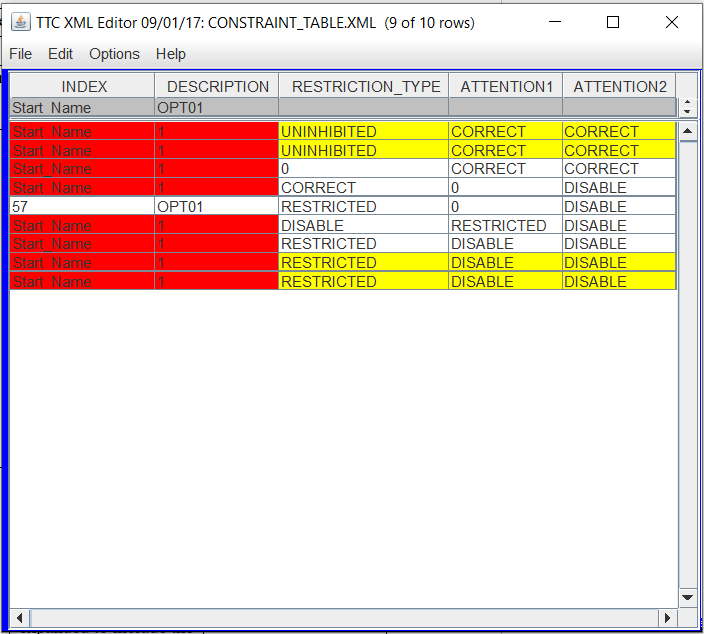
Not Applicable

# Appendix

Expected Outputs Figures:



Expected Output 1: Test 4 output



Expected Output 2:Test 5 output

**END**