Database Table Editor

Test plan

Version <1.0>

04/08/2020

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

|  |  |
| --- | --- |
| Initial Release: | 04/07/2020 |
| Current Release: | 04/07/2020 |
| Indicator of Last Page in Document: | \* |
| Date of Last Review: | 04/07/2020 |
| Date of Next Review: | 04/08/2020 |
| Target Date for Next Update: | 04/14/2020 |

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:

Steve Roach

Customer:

Software Team Members:

Stephanie Medina

Raquel B. Gonzalez

Jonathan Roman

Change Summary

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 0.1 | 04/06/2020 | Stephanie Medina | Initial information. Worked on all sections |
| 0.2 | 04/06/2020 | Stephanie Medina | Section 2 |
| 0.3 | 04/06/2020 | Stephanie Medina | Section 3 & 4 – Tests |
| 0.4 | 04/06/2020 | Stephanie Medina | Appendix – for test output |
| 0.5 | 04/07/2020 | Stephanie Medina | Formatting whole document |
| 0.6 | 04/07/2020 | Stephanie Medina | Add to section 4 |
| 0.7 | 04/08/2020 | Stephanie Medina | Formatting whole document, proofreading |
| 1.0 | 04/08/2020 | Stephanie Medina | First deliverable for team review |
| 1.1 | 04/13/2020 | Stephanie Medina | Reviewing comments left by team |
| 1.2 | 04/13/2020 | Stephanie Medina | Fixing formatting |
| 1.3 | 04/14/2020 | Stephanie Medina | Adjusted steps for tests – Section 4 |
| 1.4 | 04/14/2020 | Stephanie Medina | Final Formatting |
| 2.0 | 04/15/2020 | Stephanie Medina | Second deliverable/version for team review |

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](#_Toc37254758)

[Approval ii](#_Toc37254759)

[Document Change Control ii](#_Toc37254760)

[Distribution List ii](#_Toc37254761)

[Change Summary ii](#_Toc37254762)

[1. Introduction 5](#_Toc37254763)

[1.1. Purpose 5](#_Toc37254764)

[1.2. Scope 5](#_Toc37254765)

[1.3. System Overview 5](#_Toc37254766)

[1.4. Suspension and Exit Criteria 5](#_Toc37254767)

[1.5. Document Overview 5](#_Toc37254768)

[1.6. References 5](#_Toc37254769)

[2. Test Items and Features 6](#_Toc37254770)

[2.1. Edit Menu 6](#_Toc37254771)

[2.1.1. Insert 6](#_Toc37254772)

[2.1.2. Delete 6](#_Toc37254773)

[2.1.3. Copy 6](#_Toc37254774)

[2.1.4. Paste 6](#_Toc37254775)

[2.1.5. Drag and Drop 6](#_Toc37254776)

[2.1.6. Undo 6](#_Toc37254777)

[2.1.7. Highlighting 6](#_Toc37254778)

[3. Testing Approach 7](#_Toc37254779)

[4. Test XX 8](#_Toc37254780)

[4.1. Test Case 1 8](#_Toc37254781)

[4.2. Test Case 2 9](#_Toc37254782)

[4.3. Test Plan 3 9](#_Toc37254783)

[4.4. Test Case 4 10](#_Toc37254784)

[4.5. Test Case 5 11](#_Toc37254785)

[4.6. Test Case 6 12](#_Toc37254786)

[4.7. Test Case 7 13](#_Toc37254787)

[4.8. Test Case 8 14](#_Toc37254788)

[5. User Interface Testing 14](#_Toc37254789)

[6. Test Schedule 15](#_Toc37254790)

[7. Other Sections 16](#_Toc37254791)

[8. Appendix 17](#_Toc37254792)

# Introduction

The project is a database editor that provides the user access to view and modify the data of the stored XML tables. However, the tool does not allow for any modification of the database schema. The testing strategy and the test plan for this editor tool will be broken down in this document. The initial test strategy is to target all the edit menu operations using Black Box testing.

## Purpose

The purpose of this document is to break down a test plan for the database table editor. Due to the source code not being provided, the test plan will be focused on Black Box Testing. The components of this tool have been broken down and assigned to 4 different groups and broken down into 13 teams. This particular test plan will focus on the edit menu operations.

## Scope

The target date for this test plan is April 21st, 2020. Each week starting April 8th, the team members are to upload their current versions of their test plan and provide feedback so that the following week all authors can provide the next updated version.

## System Overview

In this system, configuration data is stored in XML tables. We've provided a simple editor to allow access to view and modify the data. The editor does not allow modification of the database schema.

A database is a set of flat files organized as rows and columns. A collection of related files are contained in a database. The database is organized in XML documents. There are two kinds of XML documents: a database description and the data tables. The description file contains a description of the data tables: the table name and columns for each table. The format is

<DATABASE name='databasename'>

<TABLE name='tablename'>

<COLUMN name='columnname' type='columntype'/>

...

</TABLE>

...

</DATABASE>

A data table is named "databasename\_\_tablename.xml". For example, if the database description is in 'TEST\_DB.XML', the <DATABASE> tag inside that document should have as its name attribute 'TEST\_DB'. If that file has a <TABLE> tag with an attribute 'BIG\_TABLE', there should be a file 'TEST\_DB\_\_BIG\_TABLE.XML' that contains the data for the BIG\_TABLE database table. This file must be in the same directory as the database file.

The format of a data table XML file is

<DATABASE name='databasename'>

<TABLE name='tablename'>

<ROW>

<columnname> value </columnname>

...

</ROW>

...

</TABLE>

</DATABASE>

Any xml file in the database can have a history section, usually at the bottom of the file. The history section is

<HISTORY>

<ITEM date='mm/dd/yyyy' author='authorname' comment='note' />

</HISTORY>

When started, the editor opens a database edit window. This window displays the tables in a database. Initially, no database is loaded. There is a menu with File and Search. File has the options to open, close, and print tables. A user can open either a database description file or a data file. If a data file is opened, the corresponding database description is also opened. The database description window displays all the database tables. Selecting a table from this display opens the data table display in a new window. It is possible to open more than one database at a time. Opening a database table file results in the database description also being opened. It is possible to open more than one database at a time. All the windows associated with a specific database have the same color outline. Opening an already-opened table should result in the table editor window for that table being raised to the top of the screen and being visible.

The Options/History option displays the history data section. On file save, the user is prompted to enter a new history entry.

If the file has been modified and the user attempts to close the file, a save prompt should be displayed.

Search opens a file search dialog that allows a user to search specified directories for files that contain a given text string. When using this option, files containing the text string are listed. Selecting the file from the display will open the database table.

In the data table display, data is displayed and is editable. The menu has File, Edit, Options, and Help. The Help option describes the editor features. The Edit menu option allows the user to insert, delete, copy, paste, and undo. The create and delete options are row operations: entire rows are affected. Undo is any operation that affects the data. It is also possible to modify rows from the table. It is not allowed to change the column headers. It is possible to re-order columns by dragging them to a new location. Duplicate rows are highlighted in yellow. Fields violating field constraints specified in the database description file are highlighted in red.

In data table display, the column names are displayed across the top row. Just below this is a blank row. This row is the display filter row. Entering data in this row will result in only the matching data rows to be displayed. In the File menu, there are options for OR FILTER or AND FILTER. If AND FILTER is selected, the display is restricted to rows that match all of the filters entered. If the OR FILTER is selected, the display should contain rows that match any of the filters entered. The filters allow a limited type of regular expression matching described on the Help pages.

The File/Compare option allows a user to compare two versions of a table. This compare can be restricted to selected columns by using the Options/Column Selection feature. The output is rather crude and is not intended for large change lists.

File/Check for Duplicate Rows should list rows that have identical contents. This check is restricted to the columns selected in Options/Select Columns.

Constraints

A constraint on a column is specified in the database table description inside the column description. For example:

<COLUMN name='FIELD\_DP' type='DOUBLE\_PRECISION' >

<CONSTRAINT name='NN\_0' type='NotNull'/>

</COLUMN>

The constraint name must be unique in the table. Types of constraints are

NotNull

Unique

Check

For NotNull, the entry in the data table must have some value. For Unique, the value must not be duplicated anywhere else in the table (e.g., the table's key value). For Check, the value must match one of the OPTION fields. For example:

<CONSTRAINT name='EN\_1' type='Check'>

<OPTION>OPT01</OPTION>

<OPTION>OPT02</OPTION>

</CONSTRAINT>

Constraint violations are highlighted in red in the data table display, and hovering the mouse cursor over the highlighted column should display a message describing the violation. Multiple constraints are allowed for each column.

## Suspension and Exit Criteria

## In this particular project, there is no suspension criteria. This is due to the fact that there are time constraints and there’s a small amount of test that need to be conducted. The basis of the exit criteria is that if 80% of the High criticality have passed, the test plan is successful. Document Overview

1. *Introduction:* general information of what the document contains
2. *Test Items and Features:* describes the test items (e.g., components, classes, functions or methods) and the features to be tested
3. *Testing Approach:* describes the approach to be used to the test the system
4. *Test “x”:* information about what each test consists of such as materials, data, expected input and output, etc.
5. *User interface testing:* not applicable due to the focus being the edit menu operations
6. *Test Schedule:* this is where the schedule for testing activities are specified
7. *Other Sections:* not applicable for this scenario
8. Appendix: further details for expected output and/or visuals needed for the sections in this document

## References

CS5387. (2020, April 5). CS5387/testplangroup3-team4. Retrieved from <https://github.com/CS5387/testplangroup3-team4>

# Test Items and Features

## Edit Menu

This option allows the user to insert, delete, copy, paste, and undo. The edit option is one of the 4 options available in the menu for the tool. The specific section where this option is available is in the data table display. Since the data that is being displayed is editable, that is why this option appears here.

### Insert

This option from the edit menu is a row operation where all the rows are affected.

### Delete

This option from the edit menu is a row operation where all the rows are affected.

### Copy

This option from the edit menu will allow the user to copy any column or row they select in order to change the position of that column or row.

### Paste

This option from the edit menu will allow the user to paste whatever they copied beforehand to wherever they selected.

### Drag and Drop

An additional functionality where if the user does not want to copy and paste, they can drag and drop whatever column they want.

### Undo

This option from the edit menu is any operation that affects the data. If a certain action affects the data, the user will be able to select undo in order to revert back.

### Highlighting

For this functionality, duplicate rows will be highlighted in yellow and any fields that are violating field constraints that were specified in the database description file will be highlighted in red.

# Testing Approach

The following tests are Black Box tests that focuses on the edit menu operations. This section’s purpose is to list all the test that are going to be performed. Table 1 will display the plan for testing. Each test listed will be categorized in criticality of High, Medium, or Low. The test cases will be based on what was described in section 2 and further explained in section 4.

Table 1: Test Plan

|  |  |  |
| --- | --- | --- |
| **TEST SUITE 1** | | |
| **Description of Test Suite** | **This test suite will test the principle functionalities of the Edit Menu options that are displayed in the database editor tool.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| Test Case 1 | **Test the functionality of Insert and observe how it affects all rows and columns** | **High** |
| Test Case 2 | **Test the functionality of Delete and observe how it affects all rows and columns** | **High** |
| Test Case 3 | **Test if the copy and paste options to confirm that they work together** | **High** |
| Test Case 4 | **Test the drag and drop functionality to make sure data is not affected** | **Low** |
| Test Case 5 | **Test a scenario where the column header is trying to be modified** | **Low** |
| ~Test Case 6 | **Test a scenario where data is affected, and the Undo option reverts those changes** | **Medium** |
| ~Test Case 7 | **Test a scenario where data is duplicated and observe if it is highlighted yellow** | **Low** |
| Test Case 8 | **Test a scenario where field constraints are being violated and observe if is highlighted red** | **High** |

~ NOTE: This Test Case 6 is recommended to test after either one of the following test cases:

Test Case 1

Test Case 2

Test Case 3

Test Case 4

~ NOTE: This Test Case 7 is recommended to test after Test Case 3 due to the copy and paste function being able to assist with this.

# Test XX

The purpose of this section is to:

* document test input, specific test procedures, and outcomes.
* establish test methods,
* explain the nature and extent of each test

For each test case, complete the following:

## Test Case 1

**Objective:** Test the functionality of Insert and observe how it affects all rows and columns

**Notes:** Since the whole column and row is affected, we need to make sure that the data isn’t affected as well.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 1 | | | | Current Status: Fail | | |
| Test title: Test Insert | | | | | | |
| Testing approach: Black Box testing approach. The detailed code for function is unknown and therefore the tester shall follow the steps described in this table. No specific data is needed aside from the already given testing data in order to fill up the database. See appendix section for details. | | | | | | |
| STEP  1  2  3  4  5  6 | OPERATOR ACTION  On startup window, select “File” then select the “Open” option  Select the file named “TYPE\_TABLE”  Click anywhere in the table and select the Edit Menu option.  Select the Insert option  Observe where the row was inserted and check if there were any changes to the data    Repeat steps 2-5 if needed, to further observe | PURPOSE  Make sure to open a file to begin populating table  Select an XML file to populate data  This is to make sure that regardless of where you select, it’ll still insert  This is the main function we are testing  Testing the functionality of the insert option  If by inserting once, it wasn’t clear if row was inserted, try it again | | | EXEPCTED RESULTS  The file browser popup shall come up to select file to open  The other populated tables shall appear as a list on the main window and a separate window shall appear with the table  The spot that was selected is highlighted blue the edit menu option pops up  A new row shall be created  A new row was created where the user selected, and no data was modified  Multiple rows have been inserted and data was not changed | COMMENTS  This is where you can make sure you’re in the correct directory in order to choose the file  If it stays in the main window, make sure to check if there are any other windows that opened  Look for insert for the next step  There are times when something is inserted and it deletes/modifies data or the insert doesn’t happen where one wanted it to  [OPTIONAL] |
| Concluding Remarks:  The rows were inserted at the bottom of the file. | | | | | | |
| Testing Team:  Pending | | | Date Completed:  04/20/2020 | | | |

## Test Case 2

**Objective:** Test the functionality of Delete and observe how it affects all rows and columns

**Notes:** Since the whole column and row is affected, we need to make sure that the data isn’t affected as well.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 2 | | | | Current Status: Pass | | |
| Test title: Test Delete | | | | | | |
| Testing approach: Black Box testing approach. The detailed code for function is unknown and therefore the tester shall follow the steps described in this table. No specific data is needed aside from the already given testing data in order to fill up the database. See appendix section for details. | | | | | | |
| STEP  1  2  3  4  5  6 | OPERATOR ACTION  On startup window, select “File” then select the “Open” option  Select the file named “TYPE\_TABLE”  Click anywhere in the table and select the Edit Menu option.  Select the Delete option  Observe where the row was deleted and check if there were any changes to the data  Repeat steps 2-5 if needed, to further observe | PURPOSE  Make sure to open a file to begin populating table  Select an XML file to populate data  This is to make sure that regardless of where you select, it’ll still delete  This is the main function we are testing  Testing the functionality of the delete option  If by deleting once, the results weren’t clear. Repeat it until the changes are noticeable | | | EXEPCTED RESULTS  The file browser popup shall come up to select file to open  The other populated tables shall appear as a list on the main window and a separate window shall appear with the table  The spot that was selected is highlighted blue the edit menu option pops up  The row that was selected will be deleted  The desired row that was selected was deleted without any changes to the other data  Multiple rows have been deleted without affecting the surrounding data | COMMENTS  This is where you can make sure you’re in the correct directory in order to choose the file  If it stays in the main window, make sure to check if there are any other windows that opened  Look for delete for the next step  There are times when something is deleted and it either doesn’t delete it completely or it deletes the wrong thing  [OPTIONAL] |
| Concluding Remarks:  The row that was selected was deleted. | | | | | | |
| Testing Team:  Pending | | | Date Completed:  04/20/2020 | | | |

## Test Plan 3

**Objective:** Test if the copy and paste options to confirm that they work together

**Notes:** Since copy and paste go hand in hand, we need to test them together cause if we test individually, we won’t know for sure if they work.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 3 | | | | Current Status: Pass | | |
| Test title: Test Copy & Paste | | | | | | |
| Testing approach: Black Box testing approach. The detailed code for function is unknown and therefore the tester shall follow the steps described in this table. No specific data is needed aside from the already given testing data in order to fill up the database. See appendix section for details. | | | | | | |
| STEP  1  2  3  4  5  6 | OPERATOR ACTION  On startup window, select “File” then select the “Open” option  Select the file named “TYPE\_TABLE”  Click anywhere in the table and select the Edit Menu option.  Select the Copy option  Select the Paste option  Repeat steps 2-5 if needed, to further observe | PURPOSE  Make sure to open a file to begin populating table  Select an XML file to populate data  This is to make sure that regardless of where you select, it’ll still copied  This is the main function we are testing  This is typically the next option to select after choosing to copy  If performing these steps doesn’t give clear result, try multiple times until it is clear | | | EXEPCTED RESULTS  The file browser popup shall come up to select file to open  The other populated tables shall appear as a list on the main window and a separate window shall appear with the table  The spot that was selected is highlighted blue the edit menu option pops up  The row that was selected will be copied to clipboard  The selected item that was copied is pasted without any changes to what was originally copied  Multiple rows were copied and pasted without affecting the data that was copied as well as the surrounding data. | COMMENTS  This is where you can make sure you’re in the correct directory in order to choose the file  If it stays in the main window, make sure to check if there are any other windows that opened  Look for copy for the next step  The expected results will not be noticeable until the next step has been performed  There are times where the data that was copied isn’t pasted or the item wasn’t copied in the first place resulting in either pasting the wrong thing or something blank  [OPTIONAL] |
| Concluding Remarks:  The copied row was pasted at the bottom of the file. | | | | | | |
| Testing Team:  Pending | | | Date Completed:  04/20/2020 | | | |

## Test Case 4

**Objective:** Test the drag and drop functionality to make sure data is not affected

**Notes:** Since the whole column and row is affected, we need to make sure that the data isn’t affected as well.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 4 | | | | Current Status: Pass | | |
| Test title: Test Drag & Drop | | | | | | |
| Testing approach: Black Box testing approach. The detailed code for function is unknown and therefore the tester shall follow the steps described in this table. No specific data is needed aside from the already given testing data in order to fill up the database. See appendix section for details. | | | | | | |
| STEP  1  2  3  4 | OPERATOR ACTION  On startup window, select “File” then select the “Open” option  Select the file named “TYPE\_TABLE”  Click and hold a column to re-order  Drag and drop the selected column | PURPOSE  Make sure to open a file to begin populating table  Select an XML file to populate data  This is the main function we are testing  Testing the functionality drag and drop | | | EXEPCTED RESULTS  The file browser popup shall come up to select file to open  The other populated tables shall appear as a list on the main window and a separate window shall appear with the table  The system shall detect the row that you want to drag  The data of desired column that was dragged and dropped was not affected nor was the other data surrounding it. The columns were just rearranged and not replaced. | COMMENTS  This is where you can make sure you’re in the correct directory in order to choose the file  If it stays in the main window, make sure to check if there are any other windows that opened  This is determined once the mouse is moved, if the column that begins to moves with mouse, that means it was detected  There are times when something is dropped somewhere that already contains data and ends up deleting/replacing what was already there |
| Concluding Remarks:  Worked as expected | | | | | | |
| Testing Team:  Pending | | | Date Completed:  04/20/2020 | | | |

## Test Case 5

**Objective:** Test a scenario where the column header is trying to be modified

**Notes:** The actual data inside the column cannot be modified. This is a specific constraint that was specified where the header cannot be edited

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 5 | | | | Current Status: Pass | | |
| Test title: Test Column Header | | | | | | |
| Testing approach: Black Box testing approach. The detailed code for function is unknown and therefore the tester shall follow the steps described in this table. No specific data is needed aside from the already given testing data in order to fill up the database. See appendix section for details. | | | | | | |
| STEP  1  2  3  4 | OPERATOR ACTION  On startup window, select “File” then select the “Open” option  Select the file named “TYPE\_TABLE”  Select a column header to edit  Attempt to modify the header name | PURPOSE  Make sure to open a file to begin populating table  Select an XML file to populate data  This is to select what we want to modify  This is the main constraint we’re trying to test | | | EXEPCTED RESULTS  The file browser popup shall come up to select file to open  The other populated tables shall appear as a list on the main window and a separate window shall appear with the table  The system shall detect that you are trying to select the column header  The system shall not allow for any modification. | COMMENTS  This is where you can make sure you’re in the correct directory in order to choose the file  If it stays in the main window, make sure to check if there are any other windows that opened  This can be determined by observing the behavior. Is there a prompt? Does a typing cursor appear on the title? Does nothing happen? |
| Concluding Remarks:  Unable to edit column name. | | | | | | |
| Testing Team:  Pending | | | Date Completed:  04/20/2020 | | | |

## Test Case 6

**Objective:** Test a scenario where data is affected, and the Undo option reverts those changes

**Notes:** There’s an assumption that this option reverts back to the step before any action has been performed. This test can be performed immediately after any other test case has been performed such as, Test Case 1-4.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 6 | | | | Current Status: Fail | | |
| Test title: Test Undo | | | | | | |
| Testing approach: Black Box testing approach. The detailed code for function is unknown and therefore the tester shall follow the steps described in this table. No specific data is needed aside from the already given testing data in order to fill up the database. See appendix section for details. | | | | | | |
| STEP  1  2  3  4  5  6 | OPERATOR ACTION  On startup window, select “File” then select the “Open” option  Select the file named “TYPE\_TABLE”  If action hasn’t been performed, follow any of 1-4 test cases in order to perform an action  Select the Undo option  Analyze data  Repeat steps 2-5 if needed, to further observe | PURPOSE  Make sure to open a file to begin populating table  Select an XML file to populate data  Follow steps to perform an action in order to make sure there is something to undo  This is the main function we are testing  Make sure if original state was recovered  If undo action wasn’t noticeable repeat multiple times to make sure it works | | | EXEPCTED RESULTS  The file browser popup shall come up to select file to open  The other populated tables shall appear as a list on the main window and a separate window shall appear with the table  At least one of the actions were performed  The system shall detect the last action that was performed and revert those changes  Changes were reverted correctly, and the table appears to be like how it was before performing an action  All the previous actions that were performed were reverted | COMMENTS  This is where you can make sure you’re in the correct directory in order to choose the file  If it stays in the main window, make sure to check if there are any other windows that opened  This is to prepare for the next steps.  There’s a possibility that the system doesn’t “save” the last state before performing an action  [OPTIONAL] |
| Concluding Remarks:  Only undoes the last thing that was done. | | | | | | |
| Testing Team:  Pending | | | Date Completed:  04/20/2020 | | | |

## Test Case 7

**Objective:** Test a scenario where data is duplicated and observe if it is highlighted yellow

**Notes:** The copy and paste functionality can assist with this test so it is recommended to this right after test case 3.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 7 | | | | Current Status: Pass | | |
| Test title: Test Duplicate | | | | | | |
| Testing approach: Black Box testing approach. The detailed code for function is unknown and therefore the tester shall follow the steps described in this table. No specific data is needed aside from the already given testing data in order to fill up the database. See appendix section for details. | | | | | | |
| STEP  1  2  3 | OPERATOR ACTION  On startup window, select “File” then select the “Open” option  Select the file named “CONSTRAINT\_TABLE”  Copy and Paste a column that isn’t highlighted (if any) | PURPOSE  Make sure to open a file to begin populating table  Select an XML file to populate data  This is to create a duplicate | | | EXEPCTED RESULTS  The file browser popup shall come up to select file to open  The other populated tables shall appear as a list on the main window and a separate window shall appear with the table  The system shall detect that there’s a duplicate and highlight the duplicate yellow | COMMENTS  This is where you can make sure you’re in the correct directory in order to choose the file  If it stays in the main window, make sure to check if there are any other windows that opened  There’s a possibility that system won’t detect a duplicate and not highlight it |
| Concluding Remarks:  Pastes and highlight the row that has been duplicated. | | | | | | |
| Testing Team:  Pending | | | Date Completed:  04/20/2020 | | | |

## Test Case 8

**Objective:** Test a scenario where field constraints are being violated and observe if is highlighted red

**Notes:** This test is assuming that one has access to see what field constraints were specified in the database description file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 8 | | | | Current Status: Fail | | |
| Test title: Test Drag & Drop | | | | | | |
| Testing approach: Black Box testing approach. The detailed code for function is unknown and therefore the tester shall follow the steps described in this table. No specific data is needed aside from the already given testing data in order to fill up the database. See appendix section for details. | | | | | | |
| STEP  1  2  3 | OPERATOR ACTION  On startup window, select “File” then select the “Open” option  Select the file named “CONSTRAINT\_TABLE”  Modify a field where it violates the constraints based on the database description file | PURPOSE  Make sure to open a file to begin populating table  Select an XML file to populate data  This is the main function we are testing | | | EXEPCTED RESULTS  The file browser popup shall come up to select file to open  The other populated tables shall appear as a list on the main window and a separate window shall appear with the table  The system shall detect a violation and highlight that component red | COMMENTS  Begin login for the database description file as well  There’s a possibility that system won’t detect a violation and not highlight it |
| Concluding Remarks:  Wasn’t sure what violates the constraints. | | | | | | |
| Testing Team:  Pending | | | Date Completed:  04/20/2020 | | | |

# User Interface Testing

This section focuses on the interaction between the user and the system. However, the goal for this test plan does not target the user interface so this section does not apply.

# Test Schedule

This section is where the schedule for testing activities is specified.

|  |  |  |
| --- | --- | --- |
| **Task and date** | **People** | **Description** |
| Task 1 – 04/08/2020 | Stephanie Medina | Generate a test plan for this specific section of the tool |
| Task 2 – 04/09/2020 ~ 04/14/2020 | Whole Team | Review and comment on each other’s test plan |
| Task 3 –  04/09/2020 ~ 04/21/2020 | Whole Team | Aside from review, attempt to perform all tests and determine if it was a good test or not |
| Task 4 – 04/14/2020 | Stephanie Medina | Review comments given by other team members and modify test plan |
| Task 5 – 04/21/2020 | Whole Team | Deliver final test plan |

# Other Sections

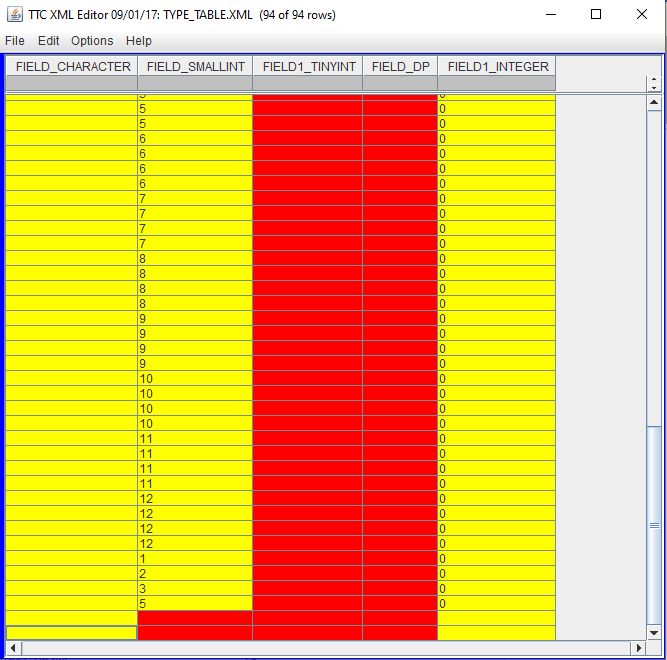
Other sections that may appear in a test plan (but not required for this course). Therefore, this section does not apply.

# Appendix

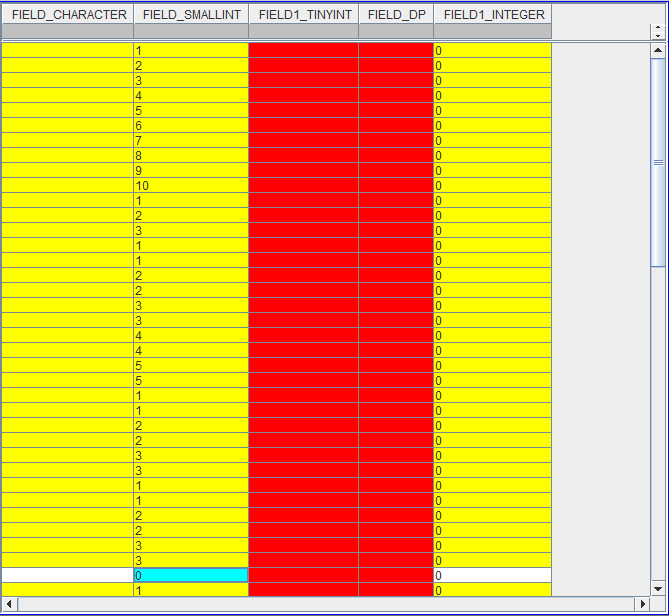
Please refer to GitHub Repository under references where there is already existing data to input for the tool. This is mainly to get the database populated. However, we are not specifically checking the data in this test plan, we are only focused on the functionalities of the operations in the Edit Menu as well as whether the data has been affected or not.

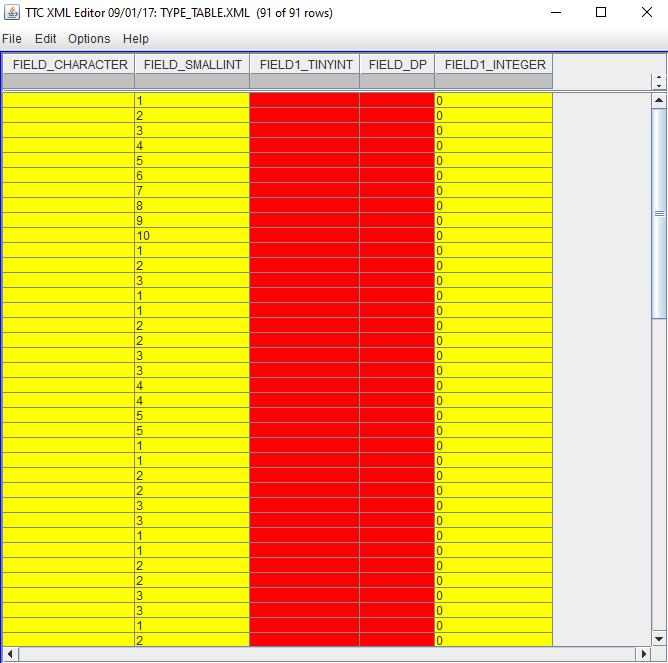
More data is to be determined for this section as the project moves along.

Test Case 1 Result:

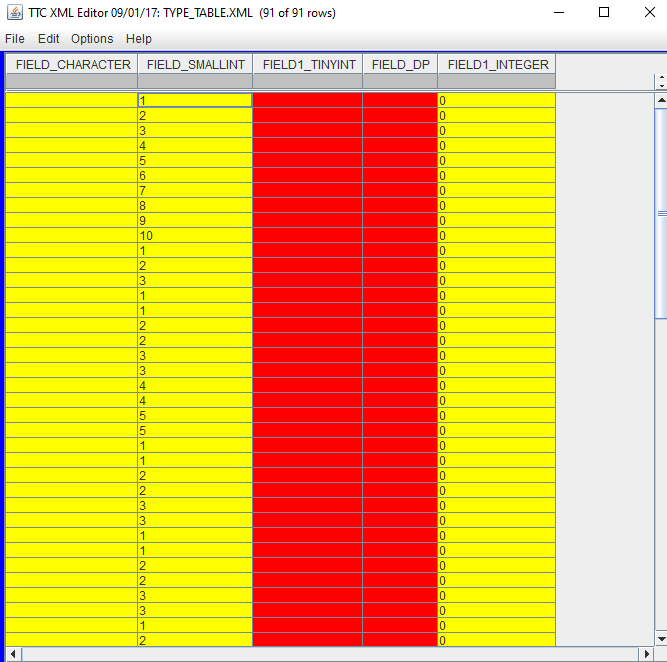


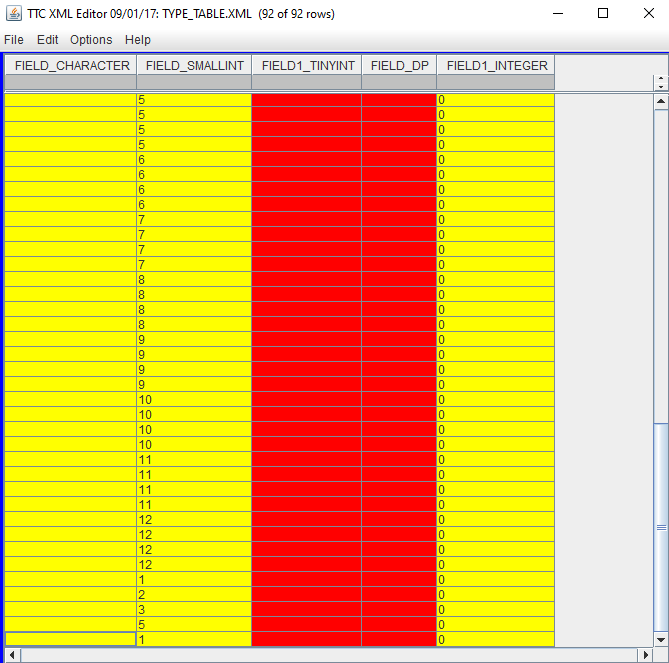
Test Case 2 Result:



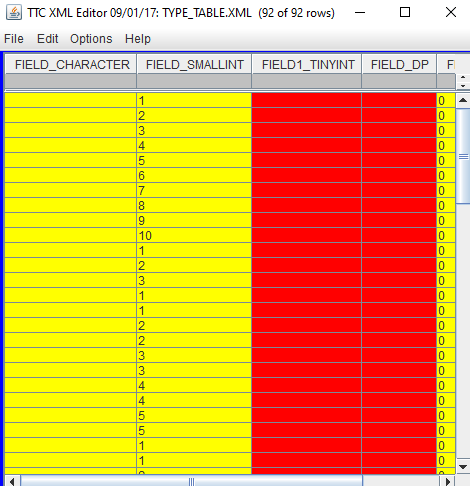


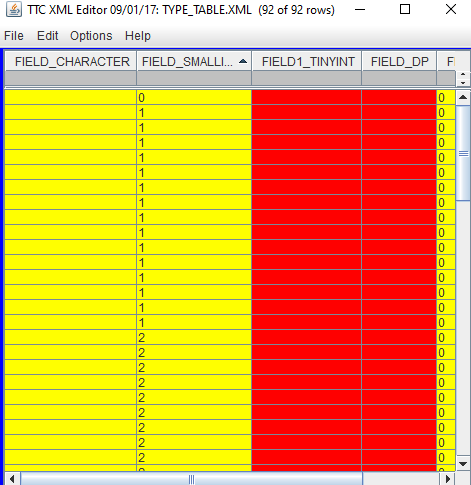
Test Case 3 Results:



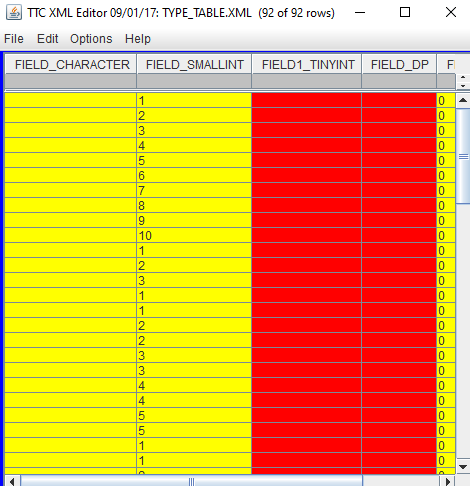


Test Case 4 Result:

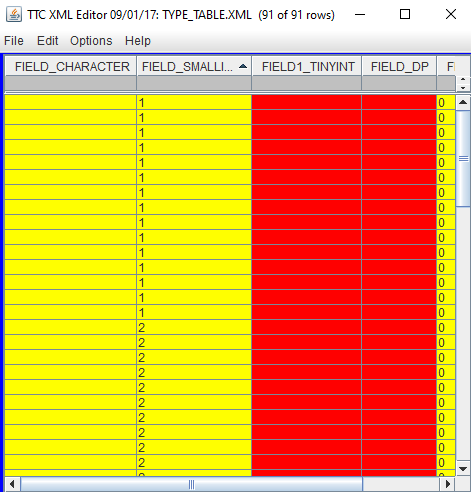


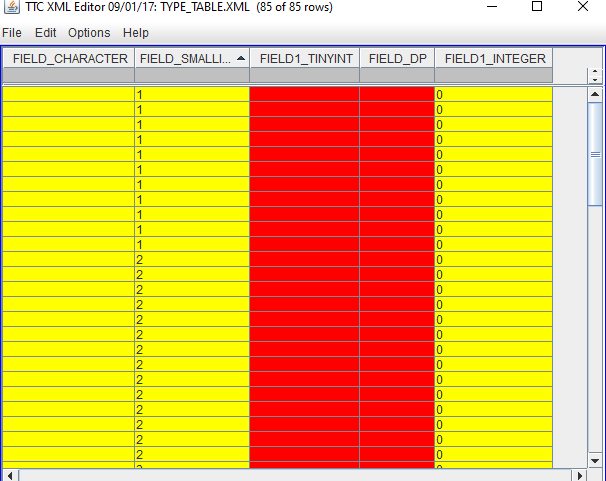


Test Case 5 Results:

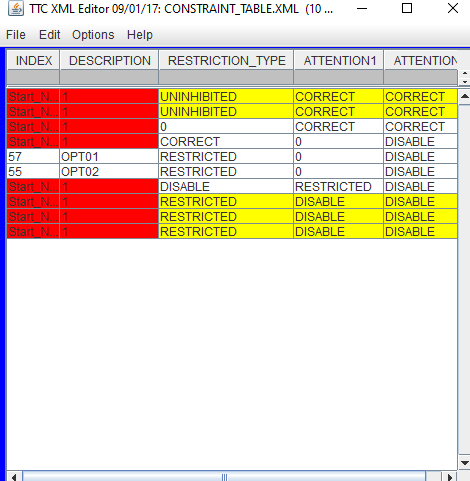


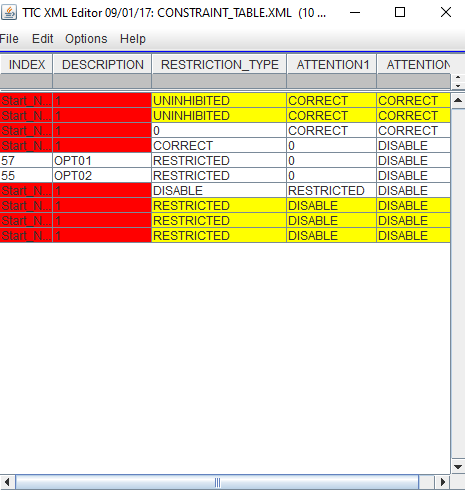
Test Case 6 Results:





Test Case 7 Results:





Test Case 8 Results:

