<dbEdit.jar>

Test plan

Version <1.0>

<4/11/2020>

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

|  |  |
| --- | --- |
| Initial Release: | 1.0 |
| Current Release: | 1.0 |
| Indicator of Last Page in Document: | %$ |
| Date of Last Review: | 4/11 |
| Date of Next Review: | 4/13 |
| Target Date for Next Update: | 4/13 |

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. Steve Roach

Customer:

Dr. Steve Roach

Software Team Members:

Ethan Hardin (author)

Denise Castro (reviewer)

Alex Ortega (reviewer)

Change Summary

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1.0 | 4/11 | Ethan Hardin | Initial creation |
|  |  |  |  |
|  |  |  |  |

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](#_Toc22915465)

[Approval ii](#_Toc22915466)

[Document Change Control ii](#_Toc22915467)

[Distribution List ii](#_Toc22915468)

[Change Summary ii](#_Toc22915469)

[1. Introduction 1](#_Toc22915470)

[1.1. Purpose 1](#_Toc22915471)

[1.2. Scope 1](#_Toc22915472)

[1.3. System Overview 1](#_Toc22915473)

[1.4. Suspension and Exit Criteria 1](#_Toc22915474)

[1.5. Document Overview 1](#_Toc22915475)

[1.6. References 1](#_Toc22915476)

[2. Test Items and Features 2](#_Toc22915477)

[3. Testing Approach 3](#_Toc22915478)

[4. Test XX 4](#_Toc22915479)

[4.1. Test <<test id>> 4](#_Toc22915480)

[5. User Interface Testing 5](#_Toc22915481)

[6. Test Schedule 6](#_Toc22915482)

[7. Other Sections 7](#_Toc22915483)

[8. Appendix 8](#_Toc22915484)

# Introduction

<< This section gives introductory information regarding the project, the system to be tested, and the testing approach.>>

## Purpose

Purpose of the project dbEdit is to “allow access to view and modify the data. The editor does not allow modification of the database schema.” This test plan provides a means of organizing the testing process by providing a list of items to be tested in addition to providing relevant details on the process.

<< Identify the project and stipulate the test plan purpose by indicating what the document contains (e.g., organizational responsibilities, test approach, test schedule. There are generally four different types of test plans: project test plan that describes the overall strategy for testing; the system test plan that describes the system from the customer’s point of view; integration test plan that describes integration of units and subsystems; unit test plan that describes modules or classes. This section needs to identify which of these this document is.>>

## Scope

dbEdit v.1. filter and search functions in the edit menu. dbEdit taken from initial commit from testPlangroup1-team1 where Dr. Roach provided the initial commit on github CS5387.

<<Specify the project software releases/versions encompassed by the plan. >>

## System Overview

As described by the specification:

“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. 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.”

<<Describe the system to be exercised by the testing approach specified in the plan. This overview serves to identify aspects of the system operation that will be the focus of the plan’s testing approach. This should align with the systems overview of other documents in the project.>>

## Suspension and Exit Criteria

Suspension criteria: 40% test case failure

Reason: Testing requirements provided by the guidance team described this test plan to consist of the 5 most significant tests. Therefore, if 2 of these “significant” tests are to fail – continue development to address these.

Exit Criteria: 100% pass rate.

Reason: Search and filtering are one of the most important features of a database viewer and editor. If these features are not correct, the program is not satisfactory.

<< “suspension criteria” describes when we suspend testing, to be resumed at a later time. For example, if 40% of the test cases fail, or if any of the critical test cases fail. If there are no suspension criteria, indicate that all tests cases will be executed. “Exit criteria” indicates when testing stops. This could be based on run rate (number of test cases run divided by number of test cases specified) or pass rate (number of test cases passed divided by number of test cases run, or test cases passed divided by number of test cases specified). Nominally, we expect to run all of the specified tests. We want the pass rate to be high. We might specify that all critical tests must pass, and 90% of the non-critical must pass. In general, we want this to be high. >>

## Document Overview

The remainder of the document will contain the test items themselves and the testing approach.

<<Describe the remainder of the document.>>

## References

* Spec.docx by Dr. Roach

<<List all the references applicable to the test plan. Generally, this includes project standards, SRS, SDD, and a product assurance plan.>>

# Test Items and Features

<< This section describes the test items (e.g., components, classes, functions or methods) and the features to be tested. It may also list features not to be tested. A class diagram is useful. A table of features is useful. >>

As described by the application the :

* Search function

“The EDIT/SEARCH menu option opens a search dialog. The search dialog allows search and replace for the entire table. A search string is entered in the text field labeled "Search". Replacement text is entered in the Replace text field. Find searches for the row with the next occurrence of the search string. The search is not case sensitive by default, but can be made so by selecting the Match Case option. The option Match Whole Words, matches a cell value if the search string matches the cell value in its entirety. If Match Whole Words is not selected, then cells that contain the search string will be found. The search can be restricted to the currently selected column by selecting the This Column Only option. Search treats all cells as text strings. Thus, (unlike filters), a cell with contents 1 is not found by the search criteria 1.0.”

* Filter function

The File/AND FILTER and File/OR FILTER menu options provide filtering for displayed data. This allows the user to select the elements of the table that are displayed and edited. The gray highlighted row at the top of the table is the filter row. The user enters selection criteria in the cells in this row. The AND filter (the default) only displays rows that match the selection criteria from every cell in the filter. The OR filter displays rows that match the selection criteria of any cell in the filter.

A simple selection criterion is just a value typed in the filter cell. For text columns, a value matches the selection text if it contains the selection text. For example, a cell with the value "Field Data" matches the selection criterion "Data". The matching is case sensitive. For columns with numeric data, the fields are converted to double precision and compared. Thus a cell containing "1.0E+01" will match the selection criterion "10".

Filters support the relational operators =, !, >, and <. A selection criterion ">5" in a numeric column will select all cells with numeric values greater than 5.0. A selection criterion of "!DATA" will select all cells that do not contain the string "DATA".

Compound selection criteria are of the form "( <criterion> <logical operator> <criterion>)". The logical operators are || for logical or and && for logical and.

# Testing Approach

<<Describe the approach to be used to the test the system. This description includes specifying the types of tests to be performed, e.g., tests designed to exercise system functions one by one; tests designed to exercise sequences of functions that approximate operational use of the system; tests designed to stress the system to its design and requirements limits. The description lists the specific tests to be performed, but does not give the test steps. For each of these tests, give it a name and specify its objective. Label the criticality of the test cases. >>

Table 1: Test Plan

|  |  |  |
| --- | --- | --- |
| **TEST SUITE <Search>** | | |
| **Description of Test Suite** | **Testing the search feature in the edit menu.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| Search1 | **Search words with match case** | **High** |
| Search2 | **Search for words that “match whole words”** | **High** |
| Search3 | **Search items in column only** | **High** |
| Search4 | **Search items in all cells** | **High** |

|  |  |  |
| --- | --- | --- |
| **TEST SUITE <Filter>** | | |
| **Description of Test Suite** | **Testing the filter feature in the edit menu.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| Filter1 | **Numeric (greater, less, equal)** |  |
| Filter2 | **String** |  |
| Filter3 | **Color** |  |
| Filter4 | **Logical** |  |

# 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 <<Search1>>

**Objective:** Search words with match case

**Notes:** The match case option narrows the search results to only those that match the same case as the search string.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: << Search1>> | | | | Current Status: << Passed >> | | |
| Test title: Searching words with the “Match case” option enabled on the search dialog. (Similar to figure 8.3) | | | | | | |
| Testing approach: Ensure proper results from manual testing by following the GUI. This is a high priority.  Precondition: Operator has run the dbEdit.jar file, opened the database file (TEST\_DB.xml), and selected the constraint Table to view. | | | | | | |
| STEP  <<1>>  <<2>>  <<3>>  <<4>>  <<5>> | OPERATOR ACTION  Click on Edit at the top left  Click on Search  Ensure the match case option is enabled (check marked)  Type Start\_Name in the search box  Press on “Find” button once | PURPOSE  Initial condition  .  Initial condition  Initial condition  String to be searched  Execute search | | | EXEPCTED RESULTS  Drop down is displayed  New dialog appears with Search and Replace fields  Start\_Name remains in search box  “Found at (1,0)” displayed bottom right and the cell it is found in should be highlighted blue | COMMENTS  Pressing Find multiple times will jump to next location this string is found. |
| Concluding Remarks: Returned correct location but the cell was not highlighted blue. Also, successfully finding the first column, first row entry may serve to find off by one errors. Figure 8.1 | | | | | | |
| Testing Team:  Ethan Hardin | | | Date Completed:  4/11/20 | | | |

## Test <<Search2>>

**Objective:** Search for words with “match whole words” enabled

**Notes:** Find cells that contain the whole search string rather than the search string being a substring within the cell

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: << Search2>> | | | | Current Status: << Passed >> | | |
| Test title: Search for strings with the “match whole words” enabled to find results that have the whole word as it is given in the search string rather than finding substring occasions. (As shown in figure 8.3) | | | | | | |
| Testing approach: Ensure proper results from manual testing by following the GUI. This is a high priority.  Precondition: Operator has run the dbEdit.jar file, opened the database file (TEST\_DB.xml), and selected the constraint Table to view. Table is sorted by Attention2 with the DISABLE at the top. As shown in appendix figure… | | | | | | |
| STEP  <<1>>  <<2>>  <<3>>  <<4>>  <<5>> | OPERATOR ACTION  Click on Edit at the top left  Click on Search  Ensure the “match whole words” option is enabled (check marked)  Type Correct in the search box  Press on “Find” button seven times | PURPOSE  Initial condition  Initial condition  Initial condition  Search string to find  Execute search | | | EXEPCTED RESULTS  Drop down is displayed  New dialog appears with Search and Replace fields  Correct remains in search box  “Found at (9,4)” displayed bottom right and the cell it is found in should be highlighted blue | COMMENTS  Press 7 times to determine if the search finds all 7 occurrences of the string “CORRECT” in addition to finding the occurrences at the last column at the last row – which may find off by one errors. |
| Concluding Remarks: Passed | | | | | | |
| Testing Team:  Ethan Hardin | | | Date Completed:  4/11/2020 | | | |

## Test <<Search3>>

**Objective** Search for words with “this column only” enabled

**Notes:** Find cells that contain the search string within the current column only.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: << Search3 >> | | | | Current Status: << Passed>> | | |
| Test title: Search items in current column only | | | | | | |
| Testing approach: Ensure proper results from manual testing by following the GUI. This is a high priority.  Precondition: Operator has run the dbEdit.jar file, opened the database file (TEST\_DB.xml), and selected the TYPE Table to view. (Similar to figure 8.3) | | | | | | |
| STEP  <<1>>  <<2>>  <<3>>  <<4>>  <<5>>  <<6>>  <<7>> | OPERATOR ACTION  Add a single 2 to the Field1\_Integer column by replacing one of the zeroes.  Click on the top “1” in the FIELD\_SMALLINT column.  Click on Edit at the top left  Click on Search  Ensure the “This column only” option is enabled (check marked)  Type 2 in the search box  Press on “Find” button 13 times | PURPOSE  Add other occurrences of 2 in a different column.  Initial condition  Initial condition  Initial condition  Initial condition  Search string to find  Execute search | | | EXEPCTED RESULTS  2 remains in the cell  The “1” becomes white.  Drop down is displayed  New dialog appears with Search and Replace fields  Remains enabled  2 remains in search box  “Found at (46,1)” displayed bottom right and the cell it is found in should | COMMENTS  There are 92 rows in this table. If search can find the middle row (46) then this helps indicate there are no off by one errors. |
| Concluding Remarks: Not highlighted in blue. Passed | | | | | | |
| Testing Team:  Ethan Hardin | | | Date Completed:  4/11/2020 | | | |

## Test <<Search4>>

**Objective:** Search for strings

**Notes:** Find cells that contain the search string with no constraints.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: <<Search4>> | | | | Current Status: << Passed >> | | |
| Test title: Search strings regardless of substring status in cells, column, or case sensitivity. | | | | | | |
| Testing approach: Ensure proper results from manual testing by following the GUI. This is a high priority.  Precondition: Operator has run the dbEdit.jar file, opened the database file (TEST\_DB.xml), and selected the BIG Table to view. (As shown in figure 8.2) | | | | | | |
| STEP  <<1>>  <<2>>  <<3>>  <<4>>  <<5>> | OPERATOR ACTION  Click on Edit at the top left  Click on Search  Ensure none of the 3 search constraints are enabled  Type V in the search box  Press on “Find” button once | PURPOSE  Initial condition  Initial condition  Initial condition  Search string to find  Execute search | | | EXEPCTED RESULTS  Drop down is displayed  New dialog appears with Search and Replace fields  V remains in search box  “Found at (0,42)” displayed bottom right and the cell it is found in should be highlighted blue | COMMENTS |
| Concluding Remarks: Not blue. Passed | | | | | | |
| Testing Team:  Ethan Hardin | | | Date Completed:  4/11/2020 | | | |

## Test <<Filter1>>

**Objective:** Filter numerical comparisons

**Notes:** Find cells that match the AND filter criteria.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: <<Filter1>> | | | | Current Status: << Passed >> | | |
| Test title: Search cells based off of numerical comparisons such as greater, equal, or less than. | | | | | | |
| Testing approach: Ensure proper results from manual testing by following the GUI. This is a high priority.  Precondition: Operator has run the dbEdit.jar file, opened the database file (TEST\_DB.xml), and selected the Type Table to view. | | | | | | |
| STEP  <<N>> | OPERATOR ACTION  Click on the index filter cell (cell directly beneath FIELD\_SMALLINT)  Type 10 into the cell  Click on the index filter cell (cell directly beneath FIELD1\_INTEGER)  Type 0 into the cell  Press tab key | PURPOSE  Initial condition  Initial condition  Initial condition  String to filter  Execute filter | | | EXEPCTED RESULTS  10 remains in cell  0 remains in cell  5 results are displayed with 10 in the column operator specified 10 as the filter and 0 for the other column | COMMENTS |
| Concluding Remarks: Passed | | | | | | |
| Testing Team:  Ethan Hardin | | | Date Completed:  4/15/2020 | | | |

## Test <<Filter2>>

**Objective:** Filter strings

**Notes:** Find cells that pass the OR filter given criteria

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: <<Filter2>> | | | | Current Status: << Passed >> | | |
| Test title: Filter for strings regardless of substring status in cells, column, or case sensitivity using OR filter. | | | | | | |
| Testing approach: Ensure proper results from manual testing by following the GUI. This is a high priority.  Precondition: Operator has run the dbEdit.jar file, opened the database file (TEST\_DB.xml), and selected the Constraint Table to view. OR Filter enabled | | | | | | |
| STEP  <<N>> | OPERATOR ACTION  Click on the RESTRICTION\_TYPE filter cell (cell directly beneath RESTRICTION\_TYPE)  Type RESTRICTED into the cell  Click on the ATTENTION1 filter cell (cell directly beneath ATTENTION1)  Type CORRECT into the cell  Press tab key | PURPOSE  Initial condition  Initial condition  Initial condition  Search string to find  Execute search | | | EXEPCTED RESULTS  RESTRICTED remains in cell  CORRECT remains in cell  8 results are shown that match the column filter criteria | COMMENTS |
| Concluding Remarks: Passed. Figure 8.4 | | | | | | |
| Testing Team:  Ethan Hardin | | | Date Completed:  4/11/2020 | | | |

## Test <<Filter3>>

**Objective:** Search based off of cell COLOR

**Notes:** Find cells that contain the filter color.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: <<Filter3>> | | | | Current Status: << Failed >> | | |
| Test title: Search for cells of a given color. | | | | | | |
| Testing approach: Ensure proper results from manual testing by following the GUI. This is a high priority.  Precondition: Operator has run the dbEdit.jar file, opened the database file (TEST\_DB.xml), and selected the CONSTRAINT Table to view. AND filter is enabled | | | | | | |
| STEP  <<N>> | OPERATOR ACTION  Click on the DESCRIPTION filter cell (cell directly beneath DESCRIPTION)  Type RED into the cell  Click on the ATTENTION1 filter cell (cell directly beneath ATTENTION1)  Type Yellow into the cell  Press tab key | PURPOSE  Initial condition  Filter color to find  Initial condition  Filter color to find  Execute search | | | EXEPCTED RESULTS  RED remains in cell  YELLOW remains in cell  Should return results that have red cells in the DESCRIPTION column and yellow cells in the ATTENTION1 column | COMMENTS |
| Concluding Remarks: Failed. Figure 8.5 | | | | | | |
| Testing Team:  Ethan Hardin | | | Date Completed:  4/11/2020 | | | |

## Test <<Filter4>>

**Objective:** Filter based on compound logic operators

**Notes:** Find cells that match combined logic operators.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: <<Filter4>> | | | | Current Status: << Passed >> | | |
| Test title: Filter based on compounded logic operators. | | | | | | |
| Testing approach: Ensure proper results from manual testing by following the GUI. This is a high priority.  Precondition: Operator has run the dbEdit.jar file, opened the database file (TEST\_DB.xml), and selected the TYPE Table to view. | | | | | | |
| STEP  <<N>> | OPERATOR ACTION  Click on the FIELD\_SMALLINT filter cell (cell directly beneath FIELD\_SMALLINT)  Type ((>3 && <10) || 1) into the cell  Press tab key | PURPOSE  Initial condition  Initial condition  Initial condition  Filter criteria to find  Execute search | | | EXEPCTED RESULTS  All entries that are greater than 3 and less than 10, or 1  50 results that match the criteria | COMMENTS |
| Concluding Remarks: Passed. Figure 8.6 | | | | | | |
| Testing Team:  Ethan Hardin | | | Date Completed:  4/11/2020 | | | |

# User Interface Testing

<<This section focuses on the interaction between the user and the system. For testing the user interface, consider the following traits:

* Consistent terminology, shortcut keys, menu selections, and presentation
* Correct language, spelling, and grammar.
* Flexibility in navigation between windows and interface elements.
* Error handling that will inform user of critical operations.
* Follows standards and guidelines such as placement of scroll bars, windows, and menu items.

This section could be integrated into Section 4.

>>

# Test Schedule

The test schedule below consists of the task to be done and the date at which it should be completed. The team involved and the description of the task will be included.

TBD

|  |  |  |
| --- | --- | --- |
| **Task and date** | **People** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

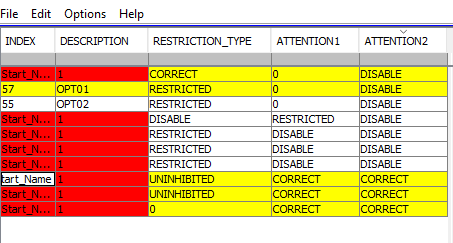
# Other Sections

Not applicable.

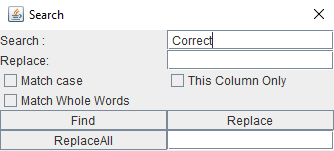
# Appendix

<< possibly more readable to put the expected output here and refer to it in the previous sections. Might also provide explicit directions for analysis of output, if it’s easier to read as an appendix or if analysis is post execution. >>

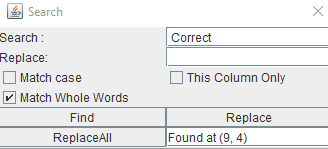
## Figure 8.1



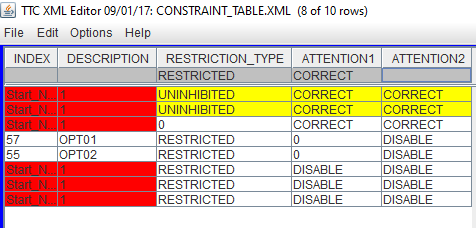
## Figure 8.2



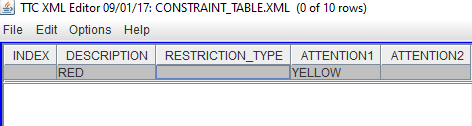
## Figure 8.3



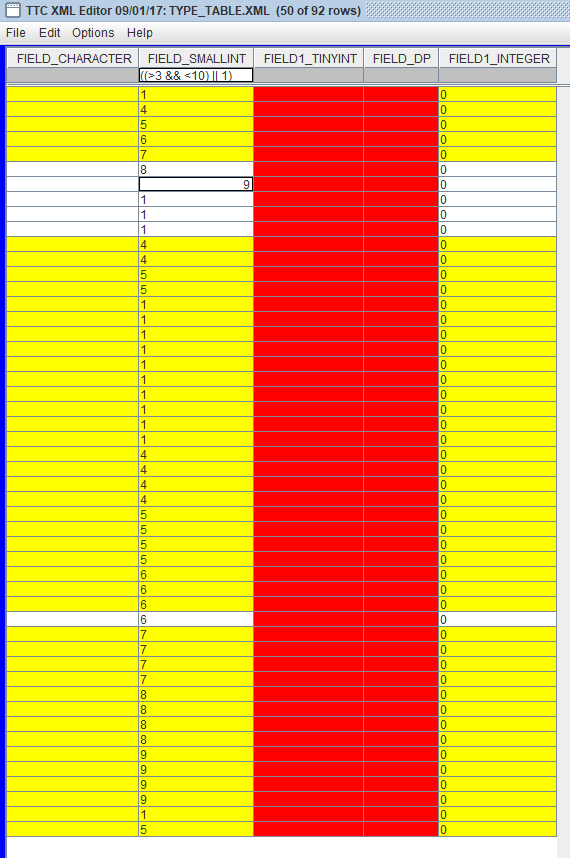
## Figure 8.4



## Figure 8.5



## Figure 8.6



%$