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.

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

## System Overview

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.

<<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. >>

# 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** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **TEST SUITE <Filter>** | | |
| **Description of Test Suite** | **Testing the filter feature in the edit menu.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
|  |  |  |
|  |  |  |

# 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 <<test id>>

**Objective: <**< Define the objective of Test XX.Y. >>

**Notes:** <<This area provides general notes concerning the test procedure. Such notes might include comments on how to execute the test procedure, an estimate of the test duration, the requirements of the procedure tests, or a statement of resources needed for this test.>>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: << Unique test ID >> | | | | Current Status: << Passed / Failed / Pending >> | | |
| Test title: <<This line contains the long title of the test procedure.>> | | | | | | |
| Testing approach: <<Included in this section is a description of test harnesses, testing frameworks, environmental requirements, test tools and test automation that will be employed to achieve testing. Include naming conventions for tests and test scripts if appropriate. Provide requirements traceability and test priority.  >> | | | | | | |
| STEP  <<N>> | OPERATOR ACTION  Describe the actions taken by the person executing the test procedure. Include the test suite, or the name of the test file (in this case, the contents of the file should be given in the appendix). | PURPOSE  Describe the reason for the step. | | | EXEPCTED RESULTS  Describe the expected response of the system being tested to the action specified under OPERATOR ACTION. This should be derived from the SRS and SDD. Clearly indicate how we determine whether the step passes. | COMMENTS |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

# 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

<< Specify the schedule for testing activities. A table with the order and completion dates of the tests is useful. The table below might be useful.>>

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

# Other Sections

<< Other sections that may appear in a test plan (but not required for this course) are:

* Test Management Requirements: how testing is to be managed; a delineation of responsibilities of each project organization involved with testing
* Staffing and training needs: delineate the responsibilities of those individuals who are to perform the testing, level of skill required, and training to be provided
* Environmental Requirements: describe the hardware (including communication and network equipment) needed to support testing; describe configuration of hardware components on which software and database to be tested are to operate.
* Software Requirements: describe the software needed to support testing; include the software code and databases that are object of the testing. Also include software tools such as compilers, CASE instruments and simulators that are needed to model the user’s operational environment.
* Risk and contingencies
* Cost: include an estimate of costs.
* Approvals
* Test Deliverables

>>

# 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. >>