Prevent, Mitigate, and Recover (PMR) Insight Collective Knowledge System (PICK)

Test plan

2.2

05/05/2020

**Document Control**

**Approval**

The Guidance Team and the customer shall approve this document.

**Document Change Control**

|  |  |
| --- | --- |
| Initial Release: | 1.0 |
| Current Release: | 2.2 |
| Indicator of Last Page in Document: | $$ |
| Date of Last Review: | 04/28/20 |
| Date of Next Review: | 04/28/20 |
| Target Date for Next Update: | 04/28/20 |

**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. Roach, Jake Lasley

Customer: Dr. Oscar Perez, Vincent Fonseca, Herandy Denisse Vazquez, Baltazar Santaella, Florencia Larsen, Erick De Nava

Software Team Members: Ian Hudson, Diego Garcia, Wan Koo, Ricardo Pineda, Jesus Yael Ramos Favela

**Change Summary**

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1.1 | 04/15/20 | Wan Koo | Worked in Sections 1 & 2 |
| 1.2 | 04/15/20 | Ricardo Pineda | Worked in Sections 1 & 3 |
| 1.3 | 04/15/20 | Jesus Yael Ramos Favela | Worked in Sections 1 & 4 |
| 1.4 | 04/15/20 | Ian Hudson | Worked in Sections 1 & 3 |
| 1.5 | 04/15/20 | Diego Garcia | Worked in Sections 1 & 4 |
| 1.6 | 04/28/20 | Wan Koo | Edited Sections 2 & 4 |
| 1.7 | 04/28/20 | Ricardo Pineda | Edited Sections 2 & 4 |
| 1.8 | 04/28/20 | Jesus Yael Ramos Favela | Edited Sections 2 & 4 |
| 1.9 | 04/28/20 | Ian Hudson | Edited Sections 2 & 4 |
| 2.0 | 04/28/20 | Diego Garcia | Edited Sections 2 & 4 |
| 2.1 | 05/05/20 | Ricardo Pineda | Added Graph Test plans to Section 4 |
| 2.2 | 05/05/20 | Jesus Yael Ramos Favela | Added more test plans to section 4 |

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**](#_heading=h.1fob9te)

[**Approval ii**](#_heading=h.3znysh7)

[**Document Change Control ii**](#_heading=h.2et92p0)

[**Distribution List ii**](#_heading=h.tyjcwt)

[**Change Summary ii**](#_heading=h.3dy6vkm)

[**1.**](#_heading=h.1t3h5sf) **Introduction 1**

[**1.1.**](#_heading=h.4d34og8) **Purpose 1**

[**1.2.**](#_heading=h.2xcytpi) **Scope 1**

[**1.3.**](#_heading=h.2s8eyo1) **System Overview 1**

[**1.4.**](#_heading=h.17dp8vu) **Suspension and Exit Criteria 1**

[**1.5.**](#_heading=h.3rdcrjn) **Document Overview 1**

[**1.6.**](#_heading=h.26in1rg) **References 1**

[**2.**](#_heading=h.lnxbz9) **Test Items and Features 2**

[**3.**](#_heading=h.35nkun2) **Testing Approach 3**

[**4.**](#_heading=h.44sinio) **Test XX 4**

[**4.1.**](#_heading=h.2jxsxqh) **Test <<test id>> 4**

[**5.**](#_heading=h.z337ya) **User Interface Testing 5**

[**6.**](#_heading=h.3j2qqm3) **Test Schedule 6**

[**7.**](#_heading=h.1y810tw) **Other Sections 7**

[**8.**](#_heading=h.4i7ojhp) **Appendix 8**

# Introduction

The overview of the project and test plan is explained in this section, with more in depth details explained in sections two through six. This section comprises the following sub-sections: Purpose, Scope, System Overview, Suspension and Exit Criteria, Document Overview and References.

## Purpose

The purpose of the Test Plan document is to provide detailed information on the testing approach and schedule conducted for the Prevent, Mitigate, and Recover (PMR) Insight Collective Knowledge System (PICK) system. The purpose of this document is to verify the functionality of the PICK system according to the requirements specified by the client, using a system test plan that describes the system from the customer’s point of view.

## System Overview

The project software version encompassed by the test plan will be PICK version 4.0, which represents the latest released version.

## Suspension and Exit Criteria

The suspension and exit criteria implemented for the test plan will be applied as follows:

Suspension criteria:

* All test cases will be executed

Exit criteria:

* Critical tests must pass – 100% passing rate
* Non-critical tests – at least 90% must pass

## Document Overview

The rest of this document is to be able to recognize the test approach that will be used in each of the tests. In addition, we will be able to check the name of the title, objectives, status, purpose, steps, and comments in each of the tests of each of the sections specified above. We are going to describe each of the User Interface Testing which is interaction between the user and the system. Finally, we will be able to specify the schedule for testing activities to keep an order of when the task was assigned, who is going to do it and the description of the task.

## References

[1] E. Tai Ramirez., “Prevent, Mitigate, and Recover (PMR) Insight Collective Knowledge System (PICK)”, SRS, El Paso, TX USA, February 2020.

# Test Items and Features

This section supplies the following test cases for artifacts such as user interface components, objects, classes, functions, methods, features, and requirements that were accommodated to be tested and acknowledged in order to make sure that the system performs what the clients requested and outputs desired results appropriately. These test cases make sure that the above provisions produce correct behavior and that there are no errors that cause the system to crash or return inaccurate results and achieve the client's business requirements. The test cases are used to determine if the system works correctly, completed, and ready to be shipped to the clients. The following lists are test case that were conducted by the software team to check for proper functionality for all requirements as well as user interface components and their features to be tested:

|  |  |
| --- | --- |
| **Test Item** | **Features** |
| Vector | Know log entries |
| Vector | Know graph |
| Log Entry | Retrieving log entries |
| Log Entry | Updating log entries |
| Log Entry | Delete log entries |
| Node | Change visibility |
| Node | Know node information |
| Graph | Export graph |
| Graph | Position node |
| Graph | Know connections |
| Icon | Know icons |
| Icon | Add icons |
| Icon | Delete icon |
| Icon | Edit icon |
| Event Configuration | Stores event configuration information |
| Enforcement Action Report | Stores EAR |
| Vector DB | Commit changes |
| Ingestion | Cleanse log files |

# Testing Approach

The testing approach that will be implemented is Black-box testing. This approach will allow a to examine the functionality of the PICK system without focusing on its internal structure. This method of test will be applied primarily to the acceptance level of software testing, also referred to as user acceptance testing.

|  |  |
| --- | --- |
| **Criticality** | **Description** |
| **Non-critical** | The effects in the system is not going to affect anything in case there is an error. |
| **Critical** | The effects in the system can affect the entire system in case there is an error. |

**Table 1: Test Plan**

|  |  |  |
| --- | --- | --- |
| **Project Test Vector** | | |
| **Description of Test Suite** | **This test will perform tests related to the Vector.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| V1 | Know description of vector | **Critical** |
| V2 | Know all log entries associated with its Vector | **Critical** |
| V3 | Know which graph is used to represent this vector | **Critical** |
| V4 | Handle all CRUD operations with vector description information | **Critical** |
| V5 | Handle all CRUD operations with Log Entries associated to this Vector | **Critical** |

**Table 2: Test Plan**

|  |  |  |
| --- | --- | --- |
| **Project Test Suite Log Entry** | | |
| **Description of Test Suite** | **This test will perform tests related to the Log Entry.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| LE1 | Knowing descriptive data about the log entry | **Critical** |
| LE2 | Handle CRUD operations on all descriptive data about a log entry | **Critical** |
| LE3 | Handle associations to a vector for a log entry, collaborates with the Vector Class | **Critical** |
| LE4 | Handle marking this log entry as significant | **Critical** |

**Table 3: Test Plan**

|  |  |  |
| --- | --- | --- |
| **Project Test Suite Node** | | |
| **Description of Test Suite** | **This test will perform tests related to the Node.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| N1 | Graph created with Nodes | **Critical** |
| N2 | Associating of at least one graph | **Critical** |
| N3 | Handle Node attributes information | **Critical** |
| N4 | Relationship between Nodes | **Critical** |

**Table 4: Test Plan**

|  |  |  |
| --- | --- | --- |
| **Project Test Suite Graph** | | |
| **Description of Test Suite** | **This test will perform tests related to the graph.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| G1 | Handle positions of Nodes | **Non-Critical** |
| G2 | Handle connection of Nodes | **Critical** |
| G3 | Comprise of at least one Node | **Critical** |
| G4 | Knows which vector is associated with | **Critical** |
| G5 | Export table | **Critical** |

**Table 5: Test Plan**

|  |  |  |
| --- | --- | --- |
| **Project Test Suite Connector** | | |
| **Description of Test Suite** | **This test will perform tests related to the Connector.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| C1 | Store the relationship between two Nodes | **Critical** |

**Table 6: Test Plan**

|  |  |  |
| --- | --- | --- |
| **Project Test Suite Icon** | | |
| **Description of Test Suite** | **This test will perform tests related to the Icon.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| I1 | Hold the name of the Icon Name | **Non-Critical** |
| I2 | Hold the File Path Information | **Non-Critical** |

**Table 7: Test Plan**

|  |  |  |
| --- | --- | --- |
| **Project Test Suite Event Configuration** | | |
| **Description of Test Suite** | **This test will perform tests related to the Event Configuration.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| EC1 | Store event configuration information | **Critical** |
| EC2 | Checks for the proper 3 folders in root directory | **Critical** |

**Table 8: Test Plan**

|  |  |  |
| --- | --- | --- |
| **Project Test Suite Enforcement Action Report** | | |
| **Description of Test Suite** | **This test will perform tests related to Enforcement Action Report.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| EAR1 | Approving or ignoring of enforcement action report when ingesting logs | **Non-Critical** |
| EAR2 | Handles position of Nodes | **Non-Critical** |

**Table 9: Test Plan**

|  |  |  |
| --- | --- | --- |
| **Project Test Suite Vector DB** | | |
| **Description of Test Suite** | **This test will perform tests related to the Vector Database.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| VDB1 | Store Vector data | **Critical** |
| VDB2 | Push VDB Changes | **Critical** |
| VDB3 | Pull VDB Changes | **Critical** |
| VDB4 | Approve VDB Changes | **Critical** |
| VDB5 | Commit Local Changes | **Critical** |
| VDB6 | Store changes made in the vector DB | **Critical** |

**Table 10: Test Plan**

|  |  |  |
| --- | --- | --- |
| **Project Test Suite Log File Ingestion** | | |
| **Description of Test Suite** | **This test will perform tests related to the File Log Ingestion.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| LFI1 | Validate Log Files | **Critical** |
| LFI2 | Cleansed Log Files | **Critical** |
| LFI3 | Ingestion of Log Files | **Critical** |

**Table 11: Test Plan**

|  |  |  |
| --- | --- | --- |
| **Project Test Suite Event Config Tab** | | |
| **Description of Test Suite** | **This test will perform tests related to the Event Config Tab.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| ECT1 | Displays lead analyst | **Non-Critical** |
| ECT2 | Displays event configuration information | **Critical** |
| ECT3 | Collects event configuration information | **Critical** |

**Table 12: Test Plan**

|  |  |  |
| --- | --- | --- |
| **Project Test Suite Log Ingestion Tab** | | |
| **Description of Test Suite** | **This test will perform tests related to the Log Ingestion Tab.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| LIT1 | Import the logs from the directories through use of a button | **Critical** |
| LIT2 | Validation of imported logs showing current imported logs and a progress bar | **Critical** |

**Table 13: Test Plan**

|  |  |  |
| --- | --- | --- |
| **Project Test Suite Log Entry Tab** | | |
| **Description of Test Suite** | **This test will perform tests related to the Log Entry Tab.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| LET1 | Display all information about Log Entries | **Critical** |
| LET2 | Send association updates to Vectors/Log Entries | **Critical** |
| LET3 | Filter/Sort/Search through Log entries | **Critical** |
| LET4 | Send Marked as significant updates to Log entries | **Critical** |

**Table 14: Test Plan**

|  |  |  |
| --- | --- | --- |
| **Project Test Suite Vector View Tab** | | |
| **Description of Test Suite** | **This test will perform tests related to the Vector View Tab.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| VVT1 | Display Vector Table view | **Critical** |
| VVT2 | Display Vector Graph view | **Critical** |
| VVT3 | Display Vector Graph relationship table | **Critical** |
| VVT4 | Collect Vector Table information | **Critical** |
| VVT5 | Collect Vector graph information | **Critical** |
| VVT6 | Collect new node relationship information | **Critical** |

**Table 15: Test Plan**

|  |  |  |
| --- | --- | --- |
| **Project Test Suite History Tab** | | |
| **Description of Test Suite** | **This test will perform tests related to the History Tab.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| CHT1 | Display Changes made in the workspace | **Non-Critical** |

**Table 16: Test Plan**

|  |  |  |
| --- | --- | --- |
| **Project Test Suite Lead Tab** | | |
| **Description of Test Suite** | **This test will perform tests related to the Lead Tab.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| CLT1 | Approve changes made in vector DB | **Critical** |
| CLT2 | Deny changes made in vector DB | **Critical** |
| CLT3 | Display commits made my analyst waiting to be approve/deny | **Critical** |

# Test Project Test Suite

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

## 4.1 Test EC1

**Objective:** Create an event configuration.

**Notes:** PICK System running is needed for testing.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: EC1 | | | | Current Status: Pass | | |
| Test title: Event Configuration | | | | | | |
| Testing approach: Creating a new event configuration, using information such as root path, event date, event time and vector. | | | | | | |
| **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.** | | | **EXPECTED 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** |
| STEP  1 | OPERATOR ACTION  In order, the start test operator clicks on the “New Project” button. | PURPOSE  The purpose is to fill-in the information such as root path, event date, event time and add a Vector. | | | EXPECTED RESULTS  Root path chosen.  Dates information will be filled in.  Time event information will be filled in.  Add vector will be added. | COMMENTS  “Invalid Root Directory” error message will appear if the Root Directory is not valid. |
| 2 | Operator clicks on the “Accept” button. | The purpose is having the options of:  Making changes on log files.  Log entry and Vector data. | | | Availability to make changes on log files.  Availability to enter log entry and Vector data. |  |
| Concluding Remarks: Users will have the ability to create an event configuration. | | | | | | |
| Testing Team:  Team 2 | | | Date Completed:  05/01/2020 | | | |

## 4.2 Test LFI1

**Objective:** Validate log files.

**Notes:** PICK System running is needed for testing.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: LFI1 | | | | Current Status: Pass | | |
| Test title: Validation of log files | | | | | | |
| Testing approach: Checking the validation of a log file containing timestamps that are bounded by the start data, end date, start time, and end time specified in the event configuration. Timestamp should be in the correct range.  Lower limit of the range: (Average of the start and end timestamps in the CSV file) minus 23 hours and 59 mins.  Upper limit of the range: (Average of the start and end timestamps in the CSV file) plus 23 hours and 59 mins. | | | | | | |
| STEP  1 | OPERATOR ACTION  After Log Files went through the cleansing process, the operator clicks on the “Import Logs” button | PURPOSE  The purpose is to valid Log File looking for specific attributes such as the start data, end date, start time, and end time that are specified in the event configuration.  It will also check if the timestamp is in the correct range. | | | EXPECTED RESULTS  The validation status of the Log File shall display “PASS” | COMMENTS  If the data validation operation is incomplete, the system shall generate an enforcement action report and set the validation status of the Log Entry to “FAIL” |
| 2 | Operator clicks on “Next File” button until he decides how many Log Files he needs/wants to validate. | The purpose is to have validated all the Log Files the operator needs/wants so he can do the ingestion process. | | | All Log Files will display the “PASS” status and the “Ingestion” button should be available for the operator. |  |
| Concluding Remarks:Users will be able to know if logs are validated.. | | | | | | |
| Testing Team:  Team 2 | | | Date Completed:  05/01/2020 | | | |

## 4.3 Test LFI2

**Objective:** Clean log files.

**Notes:** PICK System running is needed for testing.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: LFI2 | | | | Current Status: Pass | | |
| Test title: Cleansing of log files | | | | | | |
| Testing approach: Checking that log files are cleansed by not having unwanted characters and/or spaces. | | | | | | |
| STEP  1 | OPERATOR ACTION  Operator selects all types of log files that are ready to be cleansed. | PURPOSE  The purpose is to put this log files in the directory. | | | EXPECTED RESULTS  Log files will gain unwanted characters or unwanted blank spaces. | COMMENTS |
| 2 | Operator starts cleansing process. | The purpose is to remove blank lines, unwanted characters.  Remove blank rows and/or columns that CSV and Excel files might have. | | | Log files should contain no blank lines or unwanted characters.  The cleansing status of the log file should display “CLEANSED” | COMMENTS |
| Concluding Remarks: Different types of files with empty rows/columns and unwanted characters will be removed after the cleansing process.. | | | | | | |
| Testing Team:  Team 2 | | | Date Completed:  05/01/2020 | | | |

## 4.4 Test LFI3

**Objective:** Ingest log files.

**Notes:** PICK System running is needed for testing.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: LFI3 | | | | Current Status: Pass | | |
| Test title: Ingestion of log ingestion. | | | | | | |
| Testing approach: This test is to check the ingestion process is good by following the structural check completion. | | | | | | |
| STEP  1 | OPERATOR ACTION  When the validation operation is completed, the ingestion process can begin. | PURPOSE  The purpose is to check if the ingestion process can be performed. | | | EXPECTED RESULTS  The ingestion process and operator should be able to see the “Ingest” button. | COMMENTS |
| 2 | Operator will begin the ingestion process by clicking the “Ingestion” button. | The purpose is to do start the ingestion process. | | | The ingestion status should display the word “PASS” | Not ingested refers to the process of ingestion has not begun. |
| Concluding Remarks: Logs will be ingested after the log Ingestion process. | | | | | | |
| Testing Team:  Team 2 | | | Date Completed:  05/01/2020 | | | |

## 4.5 Test LE1

**Objective:** Edit a log entry.

**Notes:** PICK System running is needed for testing.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: LE1 | | | | Current Status: Pass | | |
| Test title: Editing a log entry. | | | | | | |
| Testing approach: This test is to check if a log entry can be edited. | | | | | | |
| STEP  1 | OPERATOR ACTION  The operator clicks on the “Search” blank box. | PURPOSE  The purpose is for the operator to be able to use the keyboard and search for a specific log entry. | | | EXPECTED RESULTS  The “Searching” text box should record everything that the operator typed. | COMMENTS |
| 2 | Operator clicks on “Search” button. | The purpose is to have all the log entries displayed. | | | The log entries should appear.  A scroll bar with up & down buttons should also be displayed to facilitate the operator’s navigation through the different log entries. | A scroll bar will only appear if not all log entries fit on the log entries window. |
| 3 | Operator clicks on the “Edit” button that is next to the log entry. | The purpose is to edit the log entry content, host, source, sourcetype, timestamp. | | | Changes should be visible below the column header edited (e.g. timestamp label). |  |
| Concluding Remarks: Users will have the ability to edit a log entry. | | | | | | |
| Testing Team:  Team 2 | | | Date Completed:  05/01/2020 | | | |

## 4.6 Test G5

**Objective:** Export graph.

**Notes:** PICK System running is needed for testing.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: G5 | | | | Current Status: Fail | | |
| Test title: Export graph. | | | | | | |
| Testing approach: This test is to check if the graph can be exported to jpeg or png. | | | | | | |
| STEP  1 | OPERATOR ACTION  Operator clicks on “Graph” button | PURPOSE  The purpose is to open the graphing section to create a graph with nodes and connectors. | | | EXPECTED RESULTS  A graph section will be displayed with the following buttons:  “Add node”  “Add connector”  “Delete node”  “Delete connector”  “Edit node”  “Edit Connector” | COMMENTS |
| 2 | Operator uses the following buttons:  “Add node”  “Add connector”  “Delete node”  “Delete connector”  “Edit node”  “Edit Connector” | The purpose is to create a graph. | | | Graph will be created with the help of nodes and connectors. |  |
| 3 | Operator clicks on “Export graph”. | The purpose is to have the option of exporting the graph as a JPEG or as a PNG file. | | | A small window pop-out with the following buttons:  “Export graph as JPEG”  “Export graph as PNG” |  |
| 4 | Operator uses the following buttons:  “Export graph as JPG”  “Export graph as PNG” | The purThe purpose is to export the graph. | | | The graph will be exported as a PNG or JPG format. |  |
| Concluding Remarks: After trying to export graphs, we failed to export as jpeg or png and rather could only export as json in the code we created since we couldn't figure out how to directly turn the graph to an image. | | | | | | |
| Testing Team:  Team 2 | | | Date Completed:  05/02/2020 | | | |

## 4.7 Test G2

**Objective:** Connecting Nodes

**Notes:** PICK System running is needed for testing.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: G2 | | | | Current Status: Pass | | |
| Test title: Connecting Nodes | | | | | | |
| Testing approach: This test is to check if a graph can create a connection between at least two nodes. | | | | | | |
| STEP  1 | OPERATOR ACTION  Operator clicks on “Graph” button | PURPOSE  The purpose is to open the graphing section to create a graph with nodes and connectors. | | | EXPECTED RESULTS  A graph section will be displayed with the following buttons:  “Add node”  “Add connector”  “Delete node”  “Delete connector”  “Edit node”  “Edit Connector” | COMMENTS |
| 2 | Operator uses the following buttons:  “Add node”  “Add connector”  “Delete node”  “Delete connector” | The purpose is to create a graph with multiple nodes | | | Graph will be created with the help of nodes and create a connection between them. |  |
| 3 | Operator clicks on “Add node” | The purpose is to create at least two or more nodes. | | | Nodes will be created with a certain icon and a label linked to that node. | The label was not editable. |
| 4 | Operator uses the following buttons:  “Add connector”  “Delete connector” | The puThe purpose is to create a connection between at least two nodes. | | | The graph will represent a line connecting two nodes. | The connector did not have a label. Connector is removable. Connector did not have an arrow for direction. |
| Concluding Remarks: A connection between two nodes was successful. Some smaller attributes were not implemented, such as a label for a connector. A node could be connected to more than one node if needed. | | | | | | |
| Testing Team:  Team 2 | | | Date Completed:  05/02/2020 | | | |

## 4.8 Test G3

**Objective:** Comprise of at least one node

**Notes:** PICK System running is needed for testing.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: G3 | | | | Current Status: Pass | | |
| Test title: Comprise of a node | | | | | | |
| Testing approach: This test is to check if the graph can create at least one node with label and icon image referencing the team it is coming from. | | | | | | |
| STEP  1 | OPERATOR ACTION  Operator clicks on “Graph” button | PURPOSE  The purpose is to open the graphing section to create a graph with nodes and connectors. | | | EXPECTED RESULTS  A graph section will be displayed with the following buttons:  “Add node”  “Add connector”  “Delete node”  “Delete connector”  “Edit node”  “Edit Connector” | COMMENTS |
| 2 | Operator uses the following buttons:  “Add node” | The purpose is to create and remove a node within a graph. | | | Graph will create a node including an icon that was selected and label. | There are options to which icon can be chosen. Label is not editable. |
| 3 | Operator clicks on “Delete node” | The purpose is to have the option to remove the node that was created. | | | Graph will have no nodes or one less node displayed. |  |
| Concluding Remarks: A node is able to be created within the graph view and shows a label directly linked to that node with a chosen icon as the background of the image. Node is also removable, bringing the graph back to an empty view. | | | | | | |
| Testing Team:  Team 2 | | | Date Completed:  05/02/2020 | | | |

## 4.9 Test LF13

**Objective:** Conversion of png files to text files

**Notes:** PICK System running is needed for testing.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: LF13 | | | | Current Status: Pass | | |
| Test title: OCR | | | | | | |
| Testing approach: This test is to convert all the png files into text files so they can go through the cleansing process. | | | | | | |
| STEP  1 | OPERATOR ACTION  Operator imports a png file by pressing the “Import Logs” button and selecting png files from the respective directory. | PURPOSE  The purpose is for the file to be processed into the OCR. | | | EXPECTED RESULTS  OCR will extract every character from the png file. | COMMENTS |
| 2 | Operator clicks on the “Ingest” button | The purpose is to store all the extracted characters and store them into a new text file. | | | A new text file will be created using the same name of the imported file.  Example:  File.png -> File.txt |  |
| Concluding Remarks:A new text file is created with the png content, after the OCR process is done. | | | | | | |
| Testing Team:  Team 2 | | | Date Completed:  05/02/2020 | | | |

## 4.9 Test LF13

**Objective:** Conversion of audio files to text files

**Notes:** PICK System running is needed for testing.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: LF13 | | | | Current Status: Pass | | |
| Test title: Transcription | | | | | | |
| Testing approach: This test is to convert all the audio(mp3, mp4, wav format)files into text files so they can go through the cleansing process. | | | | | | |
| STEP  1 | OPERATOR ACTION  Operator imports a png file by pressing the “Import Logs” button and selecting audio(e.g. mp3, mp4, wav) files from the respective directory. | PURPOSE  The purpose is for the file to be processed into the transcription process. | | | EXPECTED RESULTS  Transcriber will recognize and extract the speech from the audio files. | COMMENTS |
| 2 | Operator clicks on the “Ingest” button | The purpose is to store all the speech that was recognized and store them into a new text file. | | | A new text file will be created using the same name of the imported file.  Example:  File.mp3 -> File.txt |  |
| Concluding Remarks: A new text file is created with the audio content, after the transcription process is done. | | | | | | |
| Testing Team:  Team 2 | | | Date Completed:  05/02/2020 | | | |

# 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 users 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. >>

$$