Team 6: Team404

Prevent, Mitigate, and Recover (PMR) Insight Collective Knowledge System (PICK)
Test plan
Version 2.0
04/28/2020

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

Initial Release:	04/14/2020
Current Release:	04/28/2020
Indicator of Last Page in Document:	**$**
Date of Last Review:	28 April 2020
Date of Next Review:	01 May 2020
Target Date for Next Update:	02/02/2020

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:
- Dr. Steven Roach
- Jake Lasley

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

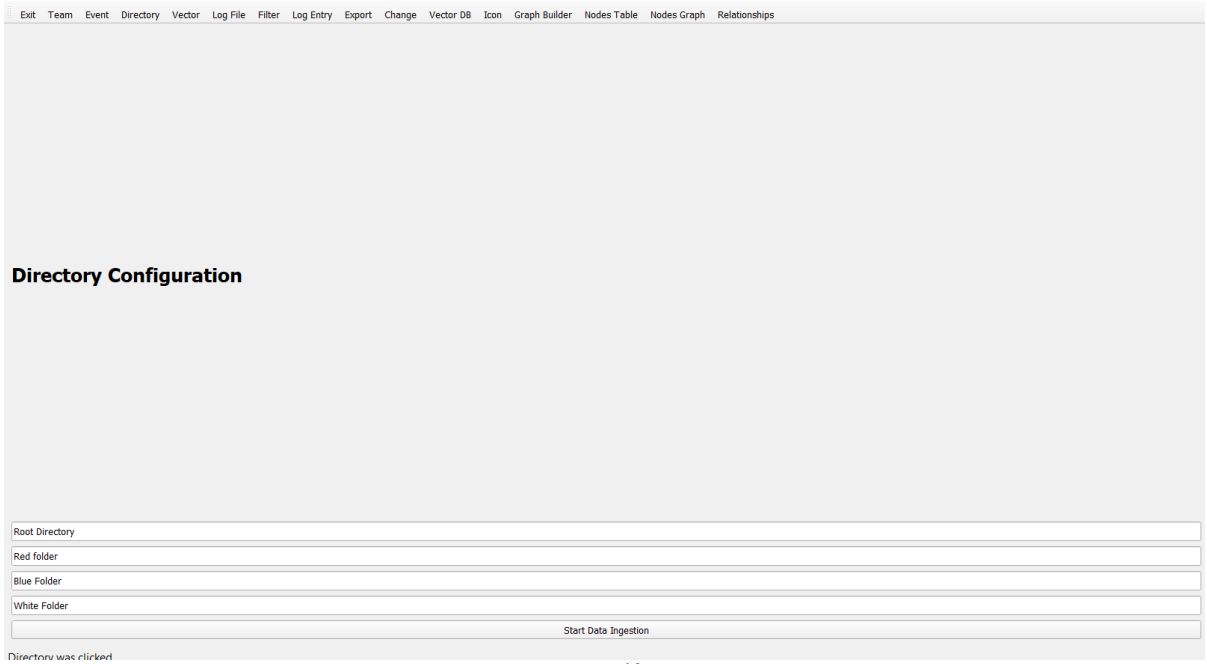
Software Team Members:
- Jacob Torres
- Eddy Todd
- Jorge Felix
- Matt Montoya
- Alejandro Zamora

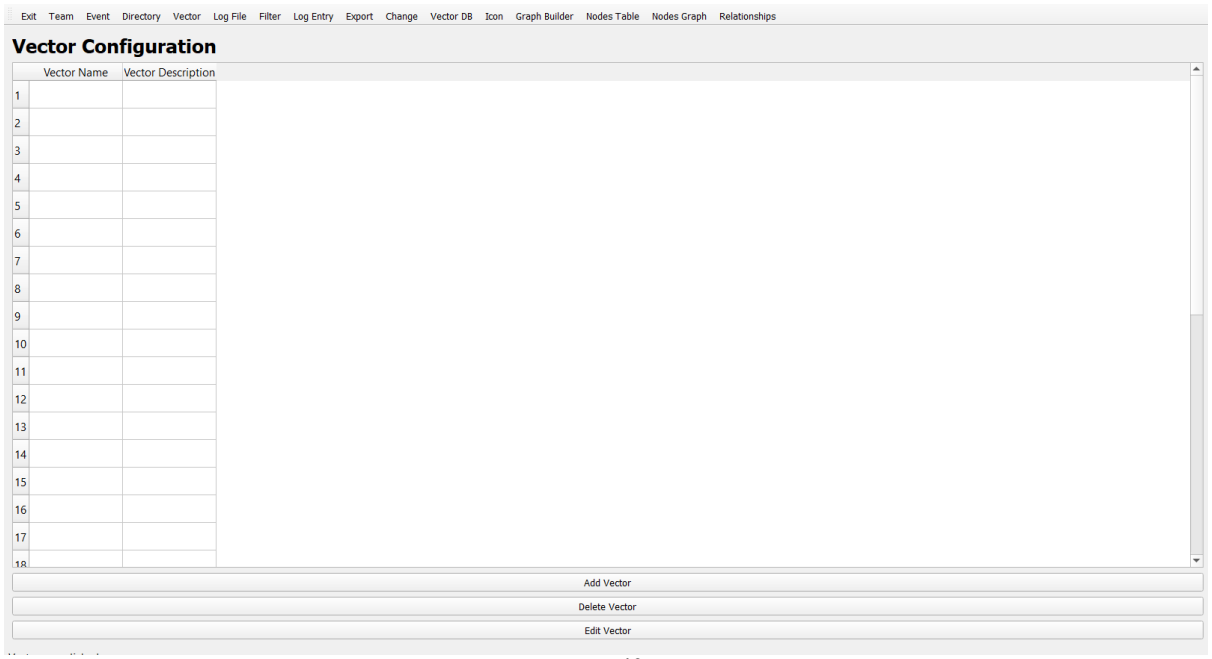
Change Summary

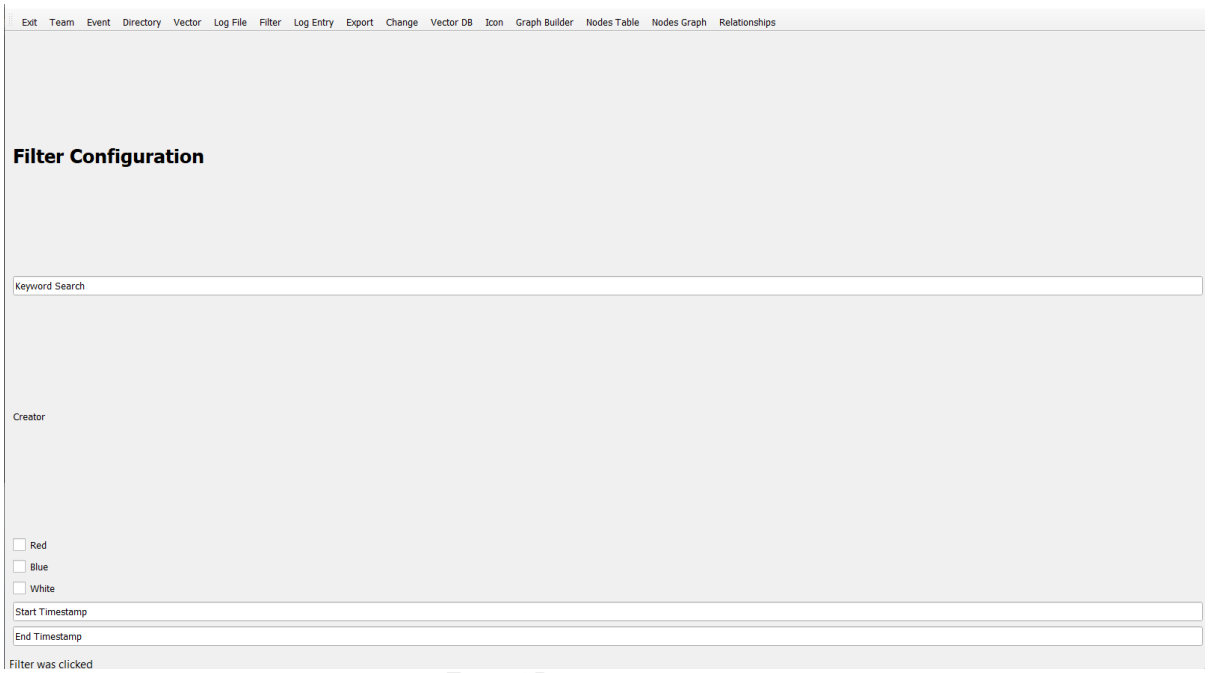
The following table details changes made between versions of this document

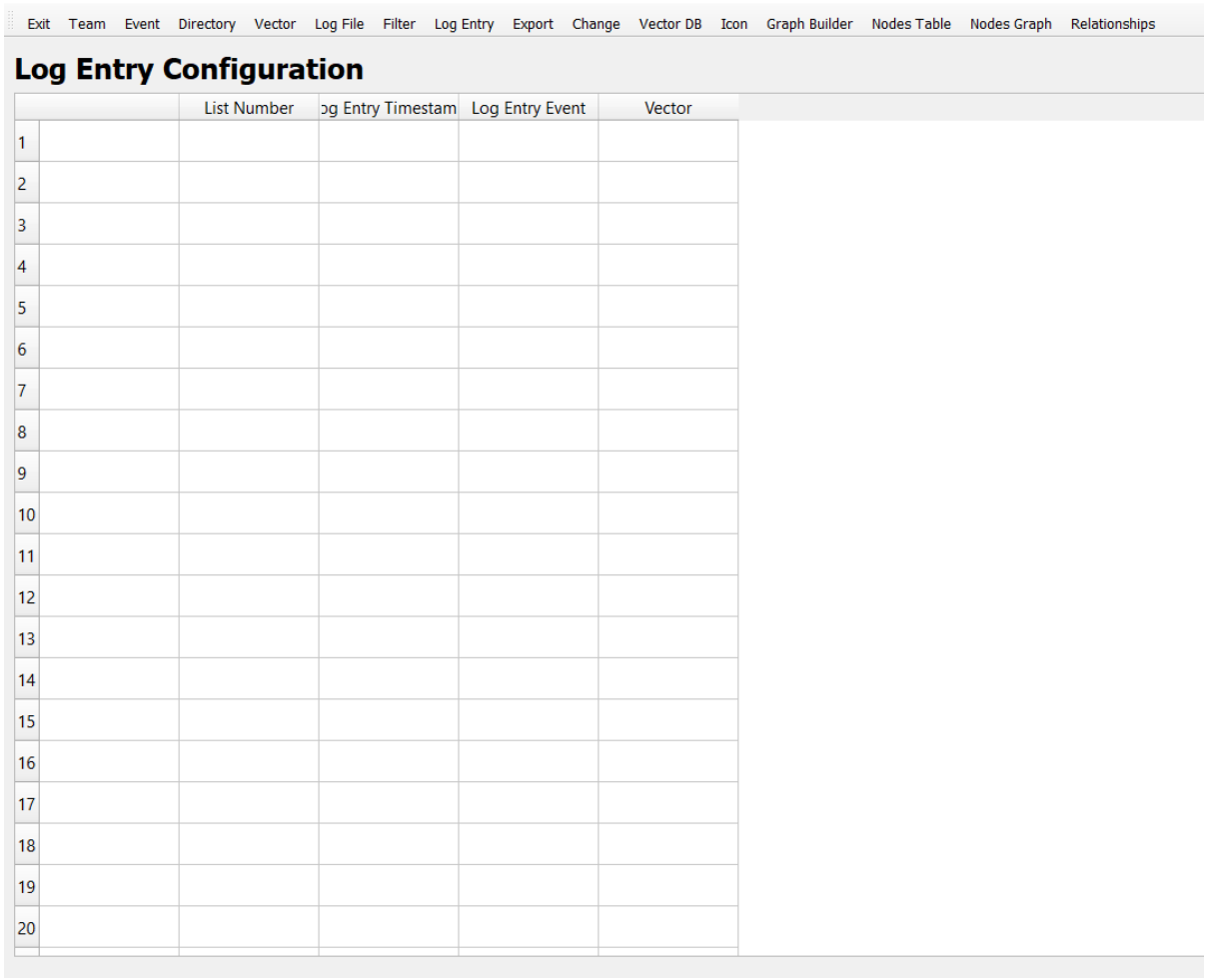
Version	Date	Modifier	Description
1.0	04/14/2020	Alex Z & Matt M	Initialized baseline
1.1	04/15/2020	Matt Montoya	Updated Section 1
1.2	04/24/2020	Alejandro Zamora	Updated Sections 2 & 3
1.3	04/24/2020	Alejandro Zamora	Updated appendix; added Section 4
1.4	04/25/2020	Alejandro Zamora	Added more tests to section 3 and 4
1.5	04/26/2020	Matt Montoya	Fixed grammar & formatting issues
1.6	04/26/2020	Matt Montoya	Reviewed current test cases
1.7	04/26/2020	Alejandro Zamora	Added test cases to section 4
2.0	04/27/2020	Matt Montoya	Fixed grammar & formatting issues

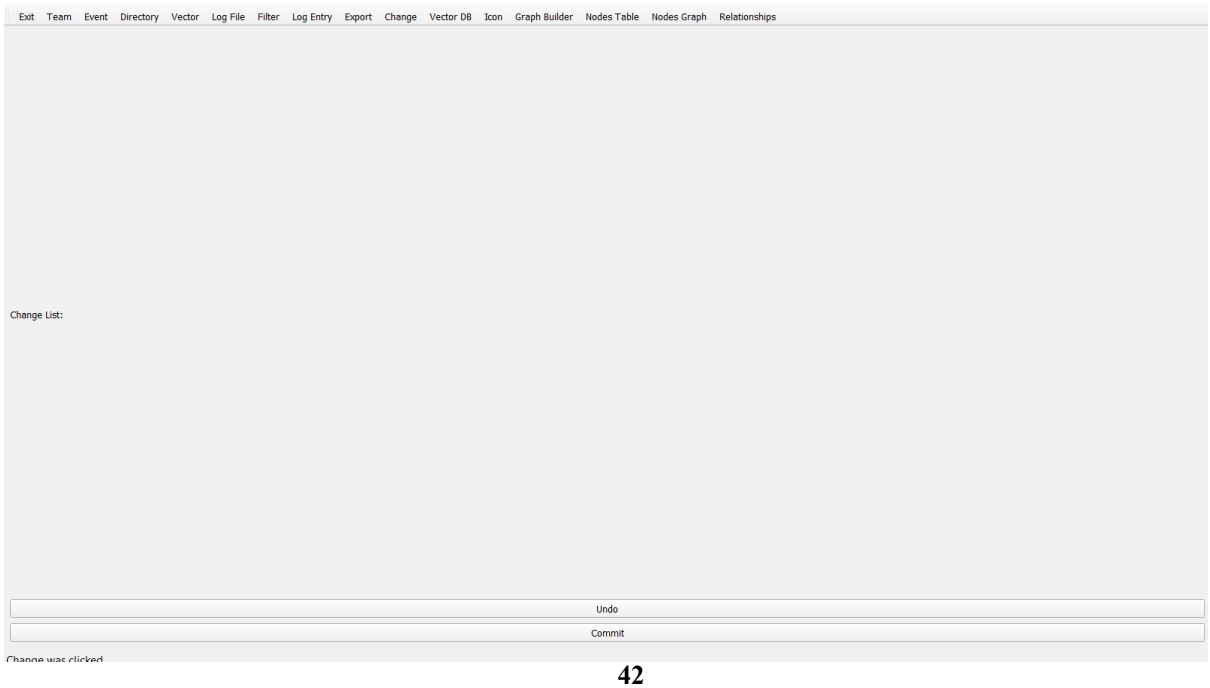
TABLE OF CONTENTS

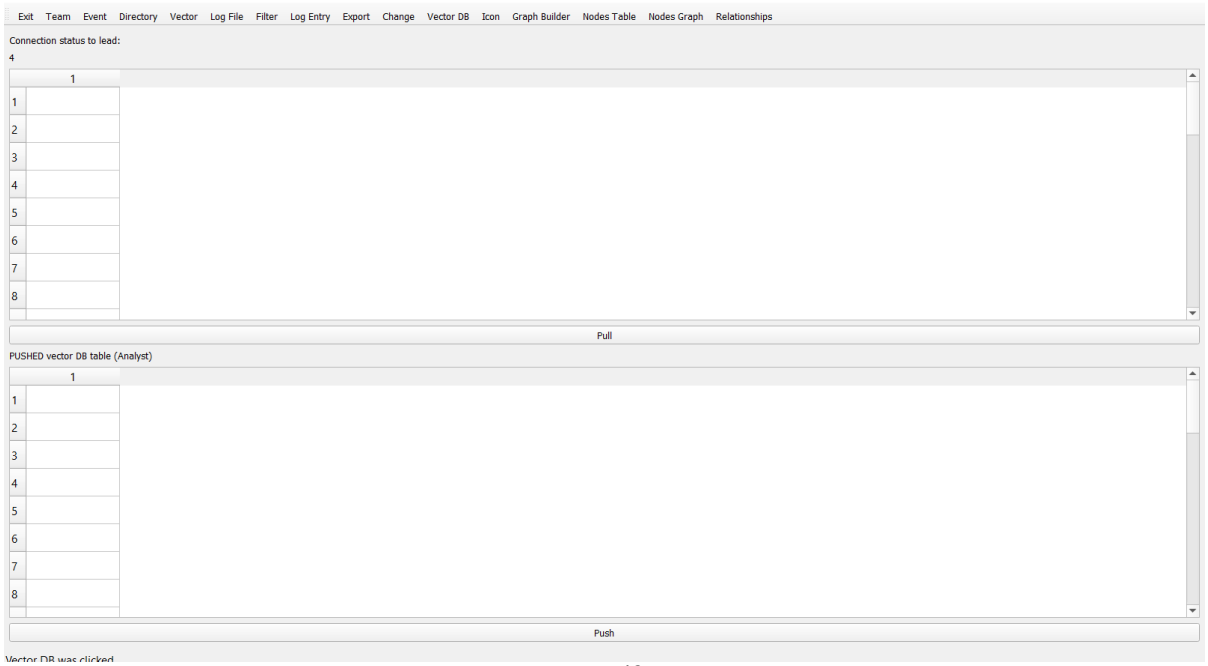


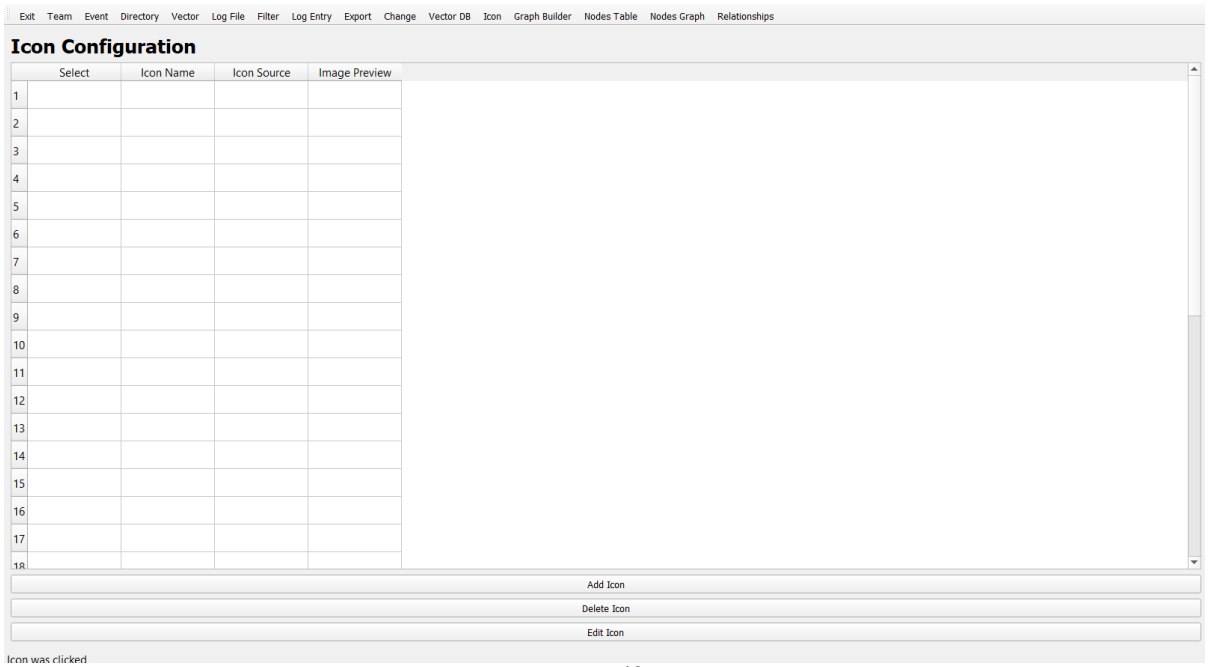


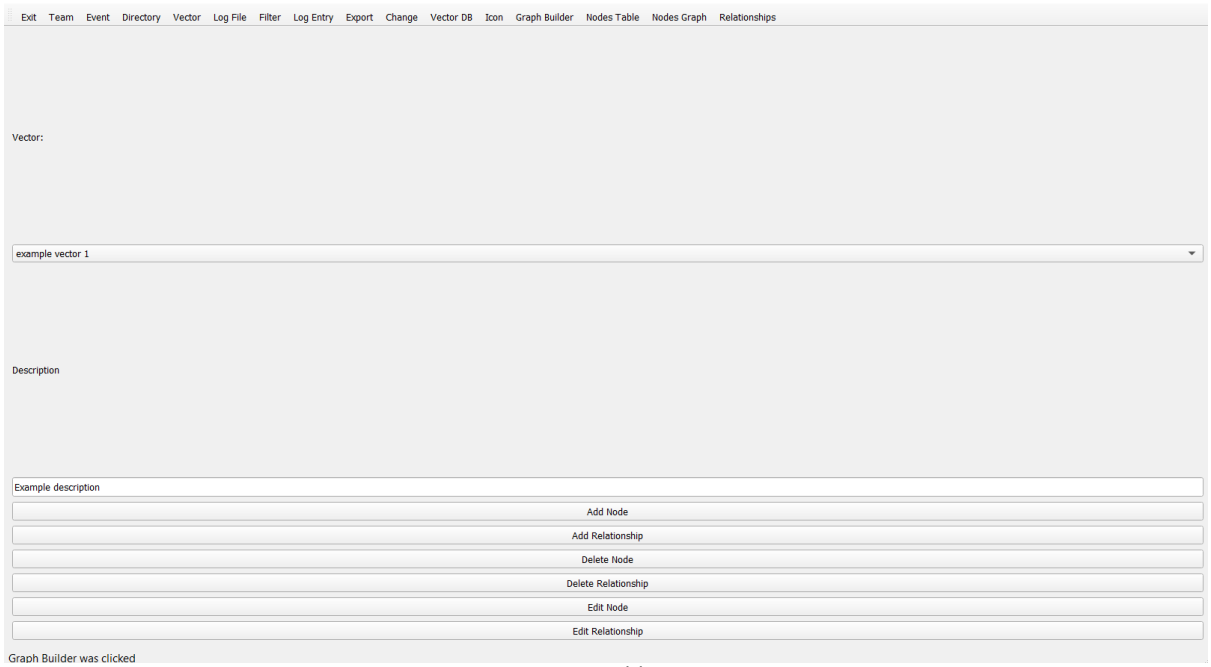


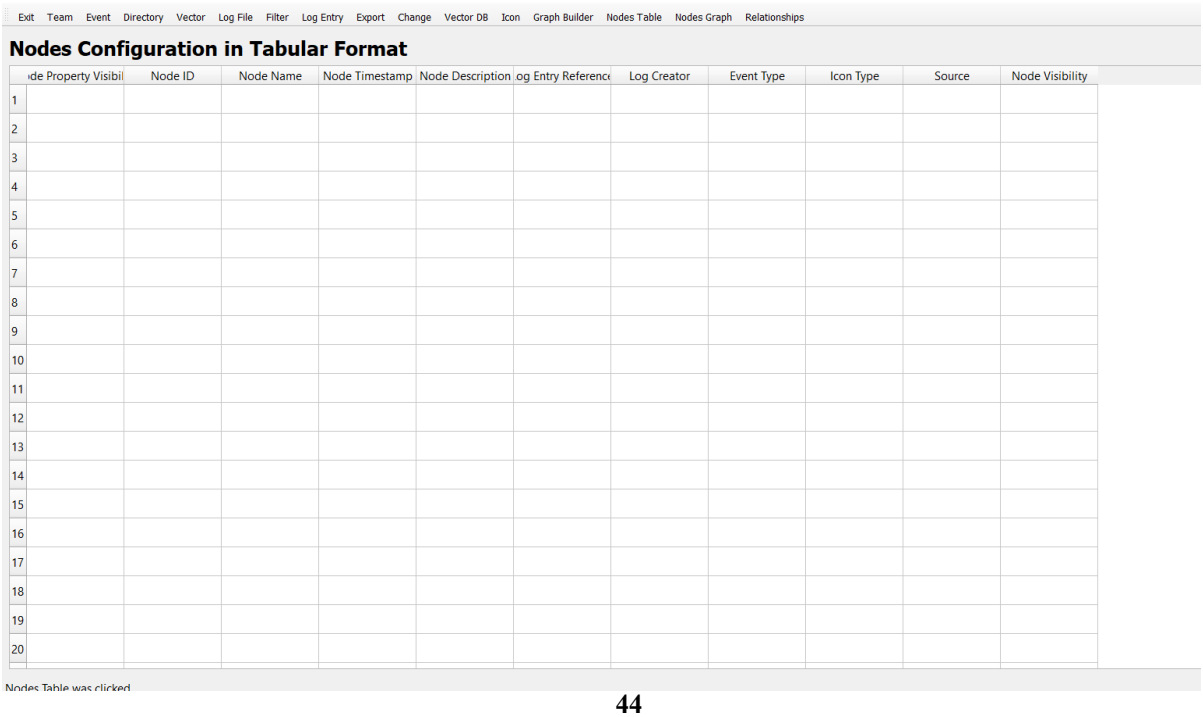


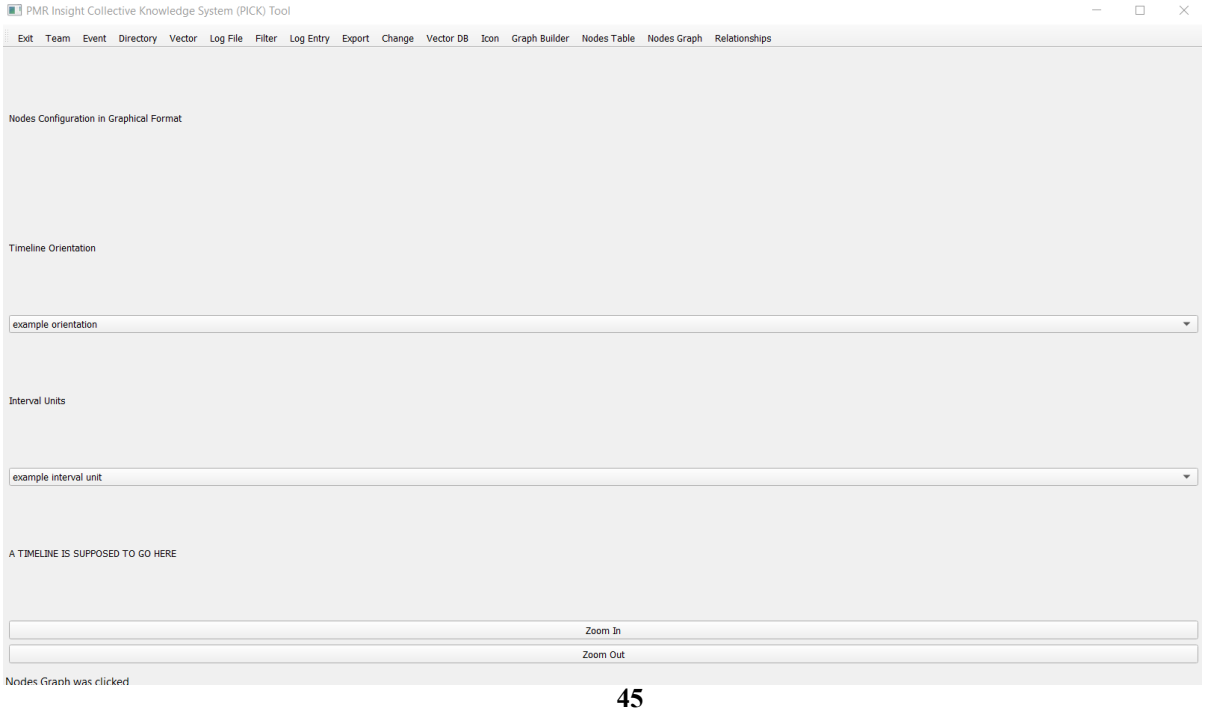


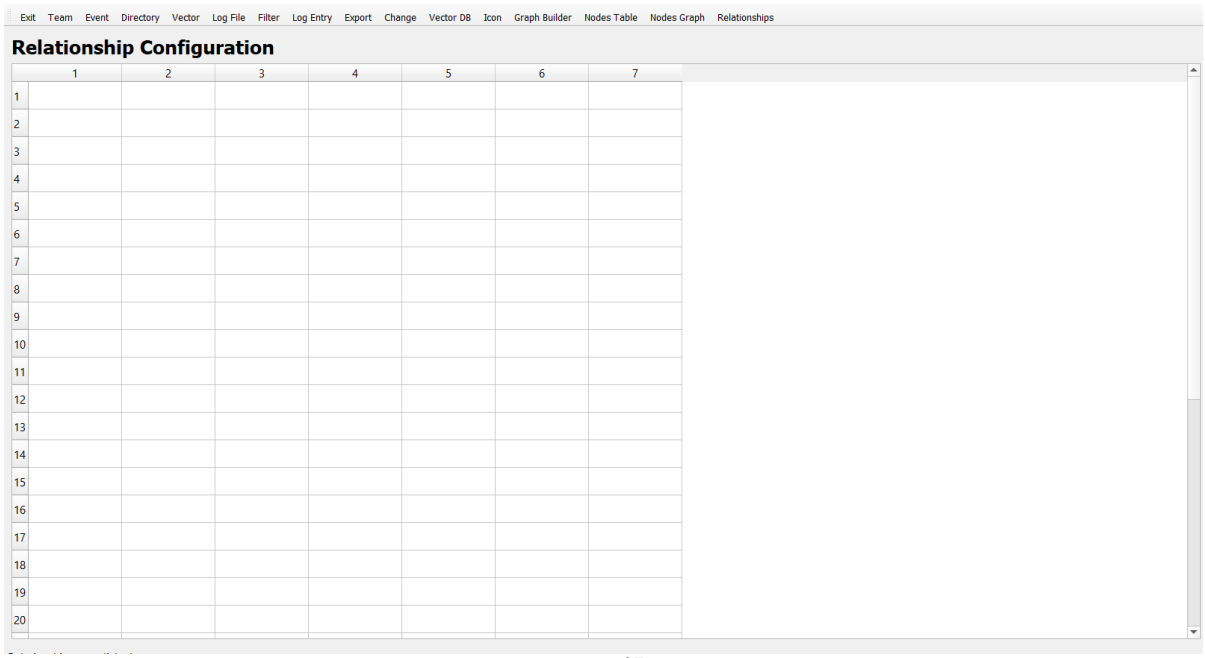


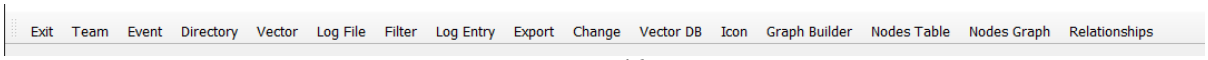


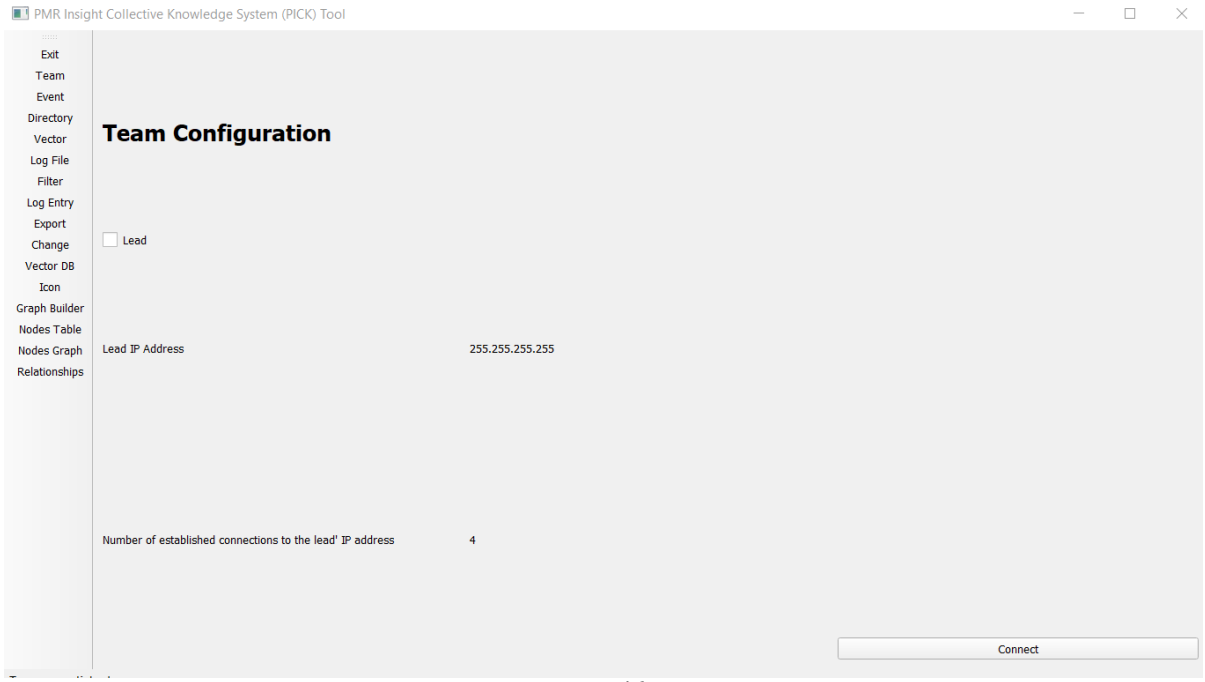


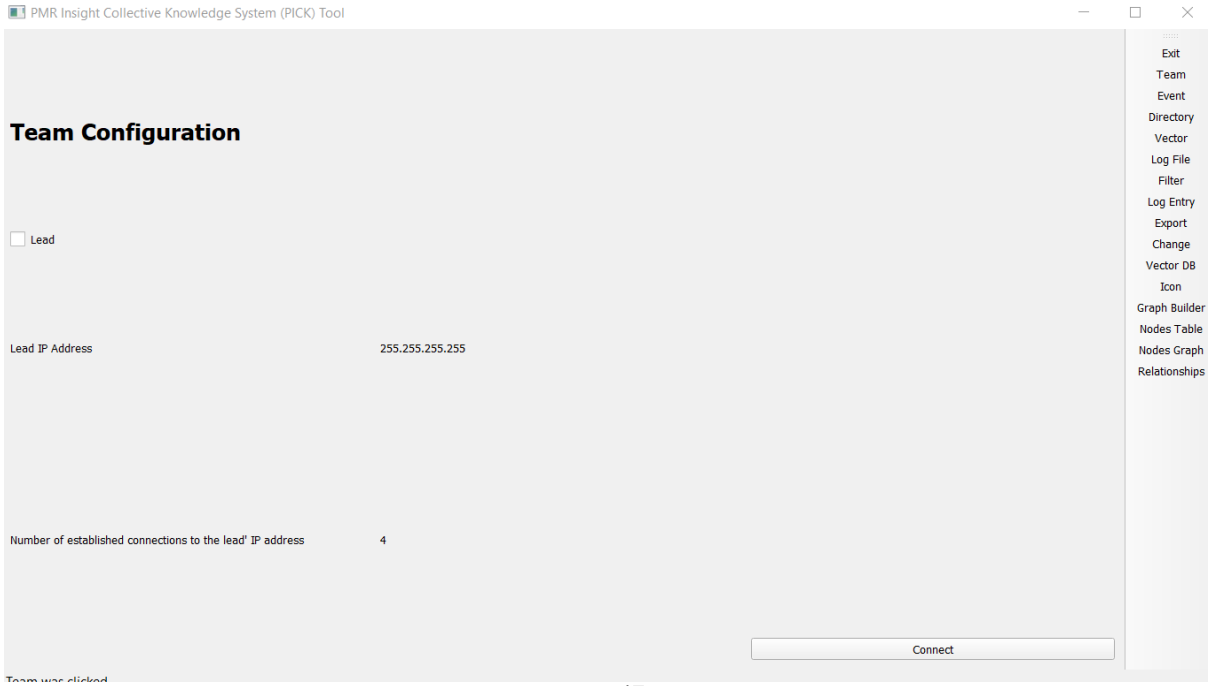


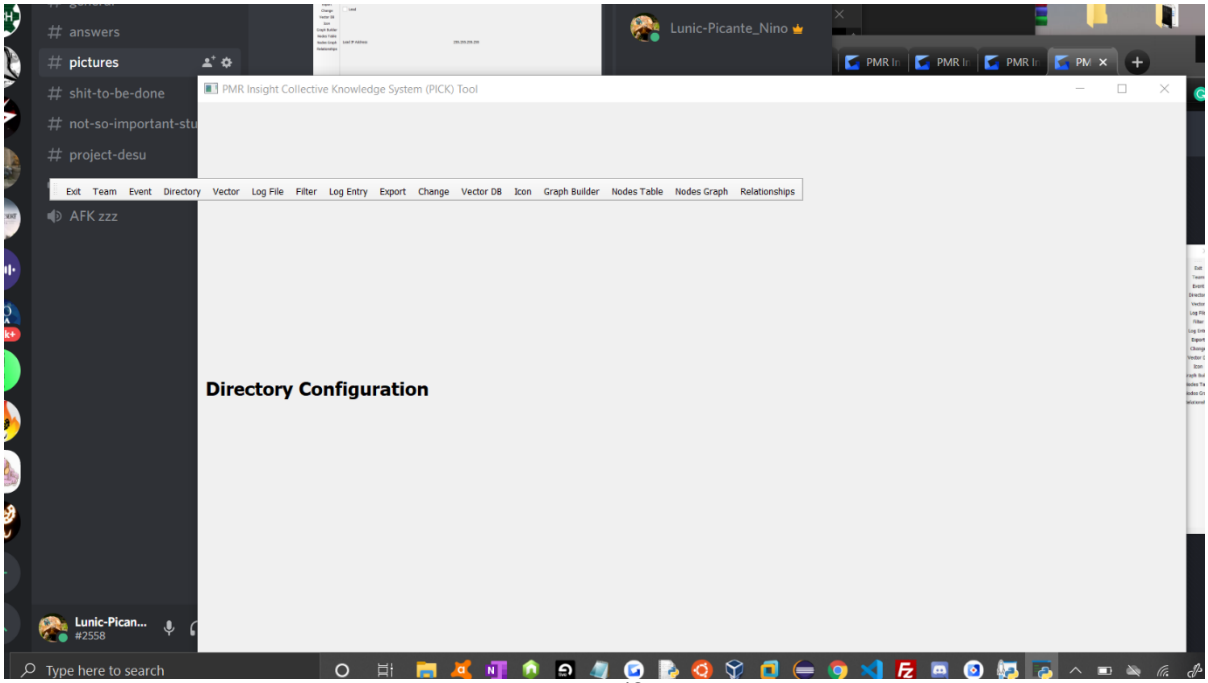












1. Introduction

Section 1 introduces the PICK Tool Test Plan. This introduction includes the purpose and scope of the document, as well as an overview of the PICK system, and establishes the conditions that shall be met to suspend or exit a test. All references applicable to the test plan, including the SRS and SDD, can be found in this section.

1.1. Purpose

The purpose of this Project Test Plan is to formally define the kinds of tests to be run, the precise tests to be run on the PICK Tool system, and the approach to running these tests. Testing of PICK Tool is an element of producing and ensuring quality software. These tests should identify any errors in the software deliverable and serve as a basis for removing defects in the system. The end goal is to ensure that the program is correct; that is, the program satisfies the specification(s) set forth by the customer.

1.2. Scope

This software release (AKA *Software Version*) encompassed by this test plan includes version 1.0. This version is the version that will be presented to the customer on the final presentation date.

1.3. System Overview

PICK Tool is a system that allows for adversarial assessment when it comes to cyber-attacks. The system will be able to allow analysts to examine a red team's actions and blue team’s actions through this assessment. The PICK Tool will allow the user to add vectors, relations, and ingest logs into the system while also showing a visual representation which would be a graphing function.

1.4. Suspension and Exit Criteria

Team404 defines Suspension and Exit Criteria as follows—

Suspension of testing may occur if there is a hardware malfunction where data is unrecoverable or if a highly critical test fails; that is, it has less than a 100% pass rate.

Exit of testing may occur if a highly critical test has a 100% pass rate, if a low critical test has at least 90% pass rate, and if the overall system has at least a 97% pass rate.

1.5. Document Overview

1.5.1. Introduction

Section 1 introduces the PICK Tool Test Plan. This introduction includes the purpose and scope of the document, as well as an overview of the PICK system, and establishes the conditions that shall be met to suspend or exit a test. All references applicable to the test plan, including the SRS and SDD, can be found in this section.

1.5.2. Test Items & Features

Section 2 describes the test items (e.g., components, classes, functions or methods) and the features to be tested.

1.5.3. Testing Approach

Section 3 details the testing approach Team404 has selected. 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.

1.5.4. GUI Functionality Test Suite

Section 4 documents test input, specific test procedures, and outcomes, as well as establish test methods, and explains the nature and extent of each test, as they relate to the GUI of PICK Tool.

1.5.5. User Interface Testing

Section 5 focuses on the interaction between the user and the system. This testing includes the following traits: Consistent terminology, menu selections, and presentation, grammar, and error handling that will inform user of critical operations.

1.5.6. Test Schedule

Section 6 specifies the schedule for testing activities as they pertain to PICK Tool.

1.5.7. Other Sections

Section 7 contains other sections. These requirements come from the SRS Document, written by the guidance team, and the code, written by Team404.

1.5.8. Appendix

Section 8 contains an appendix of figures (or images) depicting the GUI. These figures are referenced throughout the PICK Tool Test Plan.

1.6. References

1.6.1. Document Template

[1] Donaldson, S., and S. Siegel, *Successful Software Development*. Upper Saddle River, NJ: Prentice Hall, 2001, pp. 321-323.

[2] 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.

[3] Supplementary information is from:
Pfleeger, S. *Software Engineering, Theory and Practice*. Upper Saddle River, NJ: Prentice Hall, 1998, p. 365.

1.6.2. PICK Tool SRS

[4] E. Tai-Ramirez & S. Roach, SRS_v7. Internet: https://github.com/CS4311-spring-2020/pick-tool-team06-team-404/blob/master/doc/SRSv7.pdf, 2020 (Jan. 30, 2020).

1.6.3. PICK Tool SDD

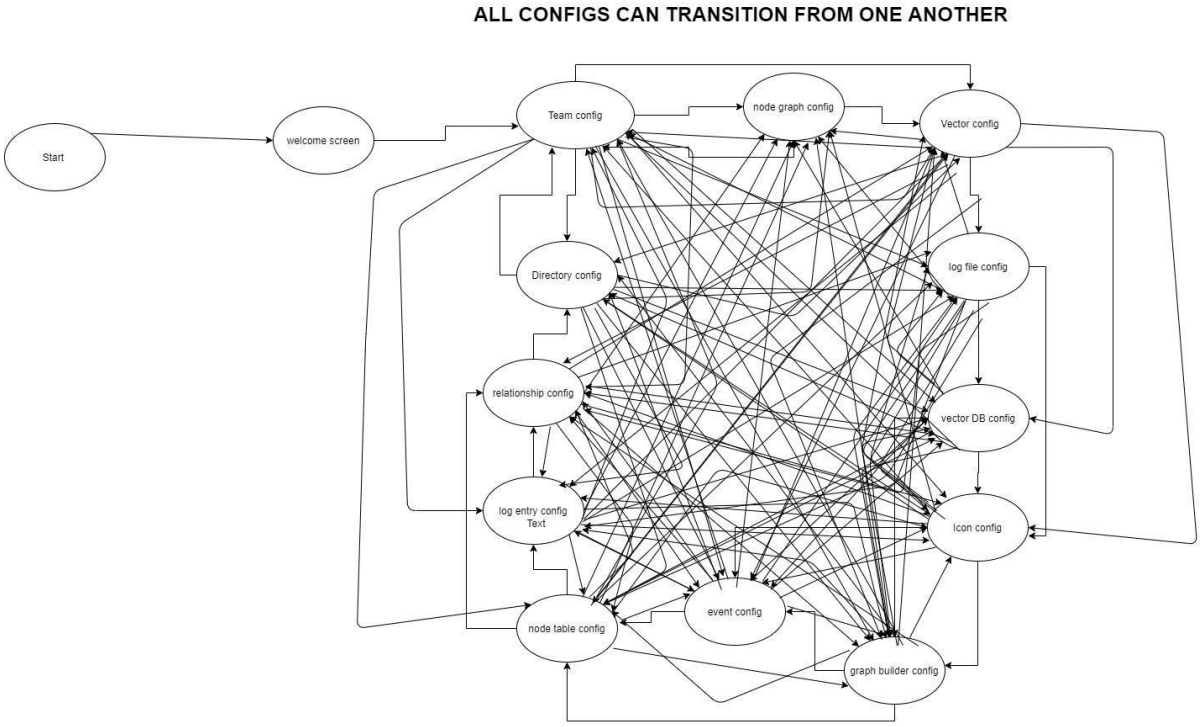
[5] A. Zamora, E. J. Todd, J. N. Torres, J. I. Felix, and M. S. Montoya, “Prevent, Mitigate, and Recover (PMR) Insight Collective Knowledge System (PICK) Tool Software Design Document,” 31-

Mar-2020. [Online]. Available: https://github.com/CS4311-spring-2020/pick-tool-team06-team-404/blob/master/doc/sdd/Team6Team404SDD.pdf. [Accessed: 14-Apr-2020].

2. Test Items and Features

The test items down below describe the system. These items given below give more specific information on the picktool.py.

2.1 Configurations Diagram



3. Testing Approach

Section 3 details the testing approach Team404 has selected. 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.

3.1. Approach & Plan

The testing approach selected by Team404 to test PICK Tool is Black Box testing. This approach examines the functionality of PICK TOOL without looking at the internal structure (or code) of the system. The types of tests to be performed are designed to exercise individual system functions. Table 1 (*Test Plan* [shown below]) describes the tests to be performed as well as their criticality, or level of importance.

GUI Functionality Test Suite		
Description of Test Suite	This will test the GUI interactions with the user	
Test Case Identifier	**Objective**	**Criticality**
GF1	Traversability between views to another view	High
GF2	Traversability of moving menu bar	Low
GF3	Persistent Information held across tabs	High
GF4	Resize of the PICK Tool window	Low
GF5	Can run Multiple instances of PICK Tool	Low
GF6	Menu bar functionality when PICK Tool window is set to minimal size	Low
GF7	Can handle large paragraphs of text as input without messing up GUI formatting	Low
GF8	Check to see if windows with tables are fully functional when PICK Tool window size is set to minimal	Low
GF9	Can recover menu bar once it has been disabled for any reason	High
GF10	When inputting information does undo and redo options in text fields work	Low

System Functionality Test Suite		
Description of Test Suite	This will test the System functionality	
Test Case Identifier	**Objective**	**Criticality**

SF1	Connect to the lead IP address	High
SF2	Creation of event	High
SF3	Start data ingestion	High
SF4	Add, edit, and delete vector	High
SF5	Filter for specific key word	Low
SF6	Export a project	High
SF7	Push and pull a project	High
SF8	Add, delete and edit Icon	High
SF9	Build a graph	High

4. GUI Functionality Test Suite

Section 4 documents test input, specific test procedures, and outcomes, as well as establish test methods, and explains the nature and extent of each test, as they relate to the GUI of PICK Tool. The purpose of is to show the step by step process on how tests are performed while also listing the expected outcomes.

4.1. Test GF1

Objective: The objective of test *GF1* is to ensure views are traversable; that is, going between views is allowed from any view to another view
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: GF1	**Current Status:** Passed
Test title: Traversability between views to another view	
Testing Approach: This test approach will see if you can navigate to the team configuration view to any other view and back to the team configuration view.	

STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the PICK Tool application	Starts PICK Tool session	The system displays the GUI	Starts the system
2	The user clicks “accept and continue”	Starts PICK Tool session	The system displays the views of a project	Starts a PICK Tool session
3	The user clicks the tab “event” located in the top center of the window	allows transition to event configuration view	The system displays the view of event configuration	Figure 2
4	The user clicks the tab “team” located in the top center of the window	allows transition to team configuration view	The system displays the view of team configuration	Figure 1
5	The user clicks the tab “directory” located in the top center of the window	allows transition to directory configuration view	The system displays the view of directory configuration	Figure 3
6	The user clicks the tab “team” located in the top center of the window	allows transition to team configuration view	The system displays the view of team configuration	Figure 1
7	The user clicks the tab “vector” located in the top center of the window	allows transition to vector configuration view	The system displays the view of vector configuration	Figure 4
8	The user clicks the tab “team” located in the top center of the window	allows transition to team configuration view	The system displays the view of team configuration	Figure 1
9	The user clicks the tab “log file” located in the top center of the window	allows transition to log file configuration view	The system displays the view of log file configuration	Figure 5
10	The user clicks the tab “team” located in the top center of the window	allows transition to team configuration view	The system displays the view of team configuration	Figure 1
11	The user clicks the tab “filter” located in the top center of the window	allows transition to filter configuration view	The system displays the view of filter configuration	Figure 6

12	The user clicks the tab “team” located in the top center of the window	allows transition to team configuration view	The system displays the view of team configuration	Figure 1
13	The user clicks the tab “log entry” located in the top center of the window	allows transition to log entry configuration view	The system displays the view of log entry configuration	Figure 7
14	The user clicks the tab “team” located in the top center of the window	allows transition to team configuration view	The system displays the view of team configuration	Figure 1
15	The user clicks the tab “export” located in the top center of the window	allows transition to export configuration view	The system displays the view of export configuration	Figure 8
16	The user clicks the tab “team” located in the top center of the window	allows transition to team configuration view	The system displays the view of team configuration	Figure 1
17	The user clicks the tab “change” located in the top center of the window	allows transition to change configuration view	The system displays the view of change configuration	Figure 9
18	The user clicks the tab “team” located in the top center of the window	allows transition to team configuration view	The system displays the view of team configuration	Figure 1
19	The user clicks the tab “vector DB” located in the top center of the window	allows transition to vector DB configuration view	The system displays the view of vector DB configuration	Figure 10
20	The user clicks the tab “team” located in the top center of the window	allows transition to team configuration view	The system displays the view of team configuration	Figure 1
21	The user clicks the tab “icon” located in the top center of the window	allows transition to icon configuration view	The system displays the view of icon configuration	Figure 11
22	The user clicks the tab “team” located in the top center of the window	allows transition to team configuration view	The system displays the view of team configuration	Figure 1
23	The user clicks the tab “graph builder” located in the top center of the window	allows transition to graph builder configuration view	The system displays the view of graph builder configuration	Figure 12
24	The user clicks the tab “team” located in the top center of the window	allows transition to team configuration view	The system displays the view of team configuration	Figure 1
25	The user clicks the tab “nodes table” located in the top center of the window	allows transition to node configuration view	The system displays the view of node configuration	Figure 13
26	The user clicks the tab “team” located in the top center of the window	allows transition to team configuration view	The system displays the view of team configuration	Figure 1
27	The user clicks the tab “nodes graph” located in the top center of the window	allows transition to node graph configuration view	The system displays the view of node graph configuration	Figure 14

28	The user clicks the tab “team” located in the top center of the window	allows transition to team configuration view	The system displays the view of team configuration	Figure 1
29	The user clicks the tab “relationship” located in the top center of the window	allows transition to relationship configuration view	The system displays the view of relationship configuration	Figure 15
30	The user clicks the tab “team” located in the top center of the window	allows transition to team configuration view	The system displays the view of team configuration	Figure 1
Concluding Remarks: This test is simple to check yet repetitive to check all of the cases.				
Testing Team: **Lead:** Alejandro Zamora **Members:** Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya			**Date Completed:** 04/14/2020	

4.2. Test GF2

Objective: The objective of this test is to check to see if the menu bar is functional when moved from its default position.
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: GF2			**Current Status**: Passed	
Test Title: Traversability of moving menu bar				
Testing Approach: This test approach will see if changing the menu bar inside the application and outside the application still yield the same functionality.				
STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the PICK Tool application	Starts PICK Tool session	The system displays the GUI	Starts the system
2	The user clicks “accept and continue”	Starts PICK Tool session	The system displays the views of a project	Starts a PICK Tool session
3	In the upper left-hand corner, you can see the menu bar. Next to the option “Exit” on the left side labeled with two columns of dots. Click and drag on those dots and to the left side of the PICK Tool window until the system displays a blue area where then you can drop the menu bar in.	Moves menu bar to left hand side of the application	The system will highlight the area where you can drag and drop the menu bar and will place the menu bar on the left side of the PICK Tool window	Figure 17
4	Looking at the menu bar click the option” Icon”	Checks to see if the menu bar still works at different x, y position	The system will transition to the Icon configuration tab	

5	Click and drag the menu bar into the bottom of the PICK Tool window until the system displays a blue area where you can then drop the menu bar in	Moves menu bar to the bottom side of the application	The system will highlight the area where you can drag and drop the menu bar and will place the menu bar on the bottom side of the PICK Tool window	Figure 18
6	Looking at the menu bar click the option" Filter"	Checks to see if the menu bar still works at different x, y position	The system will transition to the Filter configuration tab	
7	Click and drag the menu bar into the right of the PICK Tool window until the system displays a blue area where you can then drop the menu bar in	Moves menu bar to the right side of the application	The system will highlight the area where you can drag and drop the menu bar and will place the menu bar on the right side of the PICK Tool window	Figure 19
8	Looking at the menu bar click the option" Directory"	Checks to see if the menu bar still works at different x, y position	The system will transition to the Directory configuration tab	
9	Click and drag the menu bar outside of the PICK Tool window and drop the menu bar outside of the pick tool window	Moves menu bar outside of the application	The system will place the menu bar wherever the user places it	Figure20
10	Looking at the menu bar click the option" Log File"	Checks to see if the menu bar still works at different x, y position	The system will transition to the Log File configuration tab	
Concluding Remarks: The dots to move the menu bar a hard to see. When you mouse hovers over the dots the cursor changes to the 4-vector cursor indicating that relocation of menu bar is available				
Testing Team: **Lead:** Alejandro Zamora Members: Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya			**Date Completed:** 04/26/2020	

4.3. Test GF3

Objective: The objective of this test is to check to see if persistent information is held across tabs.
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: GF3			**Current Status**: Passed	
Test Title: Persistent Information held across tabs				
Testing Approach: This test approach checks if information that is input into a tab is held when going to different tabs				
STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the PICK Tool application	Starts PICK Tool session	The system displays the	Starts the system

			GUI	
2	The user clicks “accept and continue”	Starts PICK Tool session	The system displays the views of a project	Starts a PICK Tool session
3	Looking at the menu bar click the option” Vector”	Move to a window for user input	The system displays the vector configuration window	
4	Click on row 1, vector name column and populate it with “test1”	Populates the 1,1 with input	The system displays the input of the user	
5	Click on row 1, vector Description column and populate it with “test of test”	Populates the 1,2 with input	The system displays the input of the user	
6	Looking at the menu bar click the option” relationships”	Move to a window for user input	The system displays the Relationship configuration window	
7	Populate the 1st row with “1”	Populate entire first row with 1’s	The system displays the input of the user	
8	Looking at the menu bar click the option” Vector”	Check to see if information is held	The system displays the vector configuration window	
6	Looking at the menu bar click the option” relationships”	Check to see if information is held	The system displays the Relationship configuration window	
Concluding Remarks:				
Testing Team: **Lead:** Alejandro Zamora Members: Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya		**Date Completed:** 04/26/2020		

4.4. Test GF4

Objective: The objective of this test is to check to see if the system is still functional when window size is change as well as the formatting of tables.
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: GF4			**Current Status**: Passed	
Test Title: Resize of the PICK Tool window				
Testing Approach: This test approach checks if the formatting of the system along with its functionality is still able work.				
STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the PICK Tool application	Starts PICK Tool session	The system displays the GUI	Starts the system
2	The user clicks “accept and continue”	Starts PICK Tool session	The system displays the views of a project	Starts a PICK Tool session
3	The user hovers the	Resizes the PICK	The system displays the	

	mouse over the bottom right hand side corner of the PICK Tool window and clicks and drag the window to the center of the screen	Tool window	updated window to fit the new window size	
4	The user moves the mouse to the top of the window in the middle. Then clicks and drags the window to the to of their desktop screen	Resizes the PICK Tool window	The system displays the updated window to fit the new window size	
5	The user clicks the "Log File" option in the menu bar	Displays the log file configuration	The system displays the log file configuration window	Check to see if there are no formatting issues use Figure 5 for reference
Concluding Remarks:				
Testing Team: **Lead:** Alejandro Zamora Members: Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya			**Date Completed:** 04/27/2020	

4.5. Test GF5

Objective: The objective of this test is to check to see if the system can run more than one instance without those instances interfering with one another.
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: GF5			**Current Status**: Passed	
Test Title: Can run two instances of PICK Tool				
Testing Approach: This test approach checks if the system can run multiple instances of the PICK Tool without those instances interfering with one another				
STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the PICK Tool application	Starts PICK Tool session	The system displays the GUI	Starts the system
2	The user clicks "accept and continue"	Starts PICK Tool session	The system displays the views of a project	Starts a PICK Tool session
3	The user opens the directory where the picktool.py is located and executed the PICK Tool application	Starts another instance of PICK Tool	The system displays the GUI	Starts the system
3	The user clicks "accept and continue"	Starts PICK Tool session	The system displays the views of a project	
4	The User clicks on the first instance of PICK Tool and clicks the log	Displays log entry configuration page	The system displays the log entry configuration window	

	entry option in the menu bar			
5	In row 1 column 1 the user types ABC inside	Populates 1,1 with ABC	The system populates the users input in desired location	
6	The User clicks on the second instance of PICK Tool and clicks the log entry	Displays log entry configuration page	The system displays the log entry configuration window without any information input from the first instance of pick tool	
7	In row 1 column 1 the user types 123 inside	Populates 1,1 with 123	The system populates the users input in desired location	
Concluding Remarks: if the second instance sees any information of the first instance such as user input the test is a failure				
Testing Team: **Lead:** Alejandro Zamora Members: Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya			**Date Completed:** 04/27/2020	

4.6. Test GF6

Objective: The objective of this test is to check to see if the Menu bar can run and have no functionality / formatting errors when window is set to minimal
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: GF6			**Current Status**: Passed	
Test Title: Menu bar functionality when PICK Tool window is set to minimal				
Testing Approach: This test approach checks if the menu bar can display all options correctly when window of system is set to minimal size				
STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the PICK Tool application	Starts PICK Tool session	The system displays the GUI	Starts the system
2	The user clicks “accept and continue”	Starts PICK Tool session	The system displays the views of a project	Starts a PICK Tool session
3	Using the bottom right hand corner of the window the user changes the size of the window to the smallest it can be.	Resizes the PICK Tool window to the smallest it can be	The system displays the updated window to fit the new window size	
4	The user clicks the option labled “>>”	This shows the rest of the menu bar options	The system will display more menu options that were to large to fit onto the window	Figure 21
5	The user clicks on the option “Relationships”	Opens the relationships configuration table to check to see if the expanded menu	The system will display Relationship configuration window	

		options are working		
Concluding Remarks:				
Testing Team: **Lead:** Alejandro Zamora Members: Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya			**Date Completed:** 04/27/2020	

4.7. Test GF7

Objective: The objective of this test is to check to see if the system can handle large inputs of text without messing up the formatting of the window and keeping the information input without cutting out information.
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: GF7			**Current Status**: Passed	
Test Title: Can handle large paragraphs of text as input without messing up GUI formatting				
Testing Approach: This test approach checks if the system can handle large inputs of text keeping the integrity of the input and the formatting of the window. This test you will copy and paste lyrics then ascii art checking to see if the system has truncated any information from those inputs.				
STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the PICK Tool application	Starts PICK Tool session	The system displays the GUI	Starts the system
2	The user clicks “accept and continue”	Starts PICK Tool session	The system displays the views of a project	Starts a PICK Tool session
3	The user clicks log entry option in the menu bar	Move to a window that accepts input in small little boxes	The system displays the Log entry configuration	
4	The user paste text from section 8.22 Text1 into row 1 column 1	Puts a large text inside a small text box	The system displays the user input in 1,1 with a … at the end of the input indicating there is more text to be viewed	
5	The user paste text from section 8.23 Text2 into row 2 column 1	Puts a large text inside a small text box	The system displays the user input in 1,1 with a … at the end of the input indicating there is more text to be viewed	
6	The user clicks row 1 column 1	Check to see if 1,1 has all the information that was pasted into it	The system will display all text that was pasted into it with no truncations	
7	The user clicks row 2 column 1	Check to see if 2,1 has all the information that was pasted into it	The system will display all text that was pasted into it with no truncations	
Concluding Remarks: Another good way to check too see if the text is the same is to copy the text 1,1 and 2,1 into a note pad and compare but checking inside the application window is fine as well.				

Testing Team: **Lead:** Alejandro Zamora Members: Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya	**Date Completed:** 04/27/2020

4.8. Test GF8

Objective: The objective of this test is to check to see if the windows that tables are able to be navigated to view the whole table when menu size is set to minimal size.
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: GF8			**Current Status**: Passed	
Test Title: Check to see if windows with tables are fully functional when PICK Tool window size is set to minimal				
Testing Approach: This test approach checks if the windows that contain tables are able to be traversed to see the all the information of the tables in those windows when window size is set to minimal size.				
STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the PICK Tool application	Starts PICK Tool session	The system displays the GUI	Starts the system
2	The user clicks “accept and continue”	Starts PICK Tool session	The system displays the views of a project	Starts a PICK Tool session
3	The user clicks log entry option in the menu bar	Move to a window that contains a table	The system displays the Log entry configuration	
4	Using the bottom right hand corner of the window the user changes the size of the window to the smallest it can be.	Resizes the PICK Tool window to the smallest it can be	The system displays the updated window to fit the new window size	
5	Using the scroll bars on the window the user moves those scroll bars to see the entirety of the table	Displays information that is hidden due to small window size	The system displays the updated window allowing the user to scroll to view the rest of the table	
Concluding Remarks:				

Testing Team: **Lead:** Alejandro Zamora Members: Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya	**Date Completed:** 04/27/2020

4.9. Test GF9

Objective: The objective of this test is to check to see if the system allows ways to recover the menu bar once it has been disabled.
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: GF9	**Current Status**: Fail
Test Title: Can recover menu bar once it has been disabled for any reason	

Testing Approach: This test approach checks if the system allows the user to recover the menu bar once it has been purposefully disabled.				
STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the PICK Tool application	Starts PICK Tool session	The system displays the GUI	Starts the system
2	The user clicks “accept and continue”	Starts PICK Tool session	The system displays the views of a project	Starts a PICK Tool session
3	User right clicks on menu bar	View menu bar options	The system displays a blank tab with a check mark inside a box	
4	User left clicks the check mark inside the box	This disables the menu bar	The system removes the menu bar from the window	Failure at this point since there is not option to return the menu bar
Concluding Remarks: Going through the system there is no way to recover the menu bar once disabled making the whole system unable to be used except for the current window opened**.**				
Testing Team: **Lead:** Alejandro Zamora Members: Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya		**Date Completed:** 04/27/2020		

4.10. Test GF10

Objective: The objective of this test is to check to see if the system allows for undo and redo when inputting text into a table.
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: GF10		**Current Status**: Pass		
Test Title: When inputting information does undo and redo options in text fields work				
Testing Approach: This test approach checks if the system allows the undo and redo of text when being input into tables				
STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the PICK Tool application	Starts PICK Tool session	The system displays the GUI	Starts the system
2	The user clicks “accept and continue”	Starts PICK Tool session	The system displays the views of a project	Starts a PICK Tool session
3	The user clicks on the log entry option on the menu bar	This moves into a place where we can input text	The system displays the Log entry configuration window	
4	The user paste text from section 8.24 Text3 into row 1 column 1. A total of 4 times	Populates 1,1 with input 4 separate times	The system displays input in 1,1	
5	The user presses Ctrl+Z on their keyboard once	This undoes the most recent paste	The system undo’s the most recent paste from the user in 1,1	

6	The user presses Ctrl+Y on their keyboard once	This puts back the string to original state	The system returns the most recent change back to its original state	
Concluding Remarks:				
Testing Team: **Lead:** Alejandro Zamora Members: Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya			**Date Completed:** 04/27/2020	

4.11. Test SF1

Objective: The objective of this test is to check to see if the system can connect to a lead IP address
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: SF1			**Current Status**: Fail	
Test Title: Connect to the lead IP address				
Testing Approach: This test approach checks if the system allows the user to connect to a lead IP				
STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the PICK Tool application	Starts PICK Tool session	The system displays the GUI	Starts the system
2	The user clicks “accept and continue”	Starts PICK Tool session	The system displays the Team configuration window	Starts a PICK Tool session Figure 1
3	The user clicks the connect button	The only action available to do in this page	The system should respond with more options to allow user to connect to Lead IP	Test fails right here since the connect button has no functionality
Concluding Remarks: The test fails since the team configuration page has no functionality.				
Testing Team: **Lead:** Alejandro Zamora Members: Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya			**Date Completed:** 04/27/2020	

4.12. Test SF2

Objective: The objective of this test is to check to see if the system can create an event
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: SF2			**Current Status**: Fail	
Test Title: Creation of event				
Testing Approach: This test approach checks if the system allows the user can create an event				
STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the	Starts PICK Tool		

	PICK Tool application	session	The system displays the GUI	Starts the system
2	The user clicks “accept and continue”	Starts PICK Tool session	The system displays the Team configuration window	Starts a PICK Tool session Figure 1
3	The user clicks the event option on the menu bar	Open the window to create an event	The system displays the Event configuration window	Figure 2
4	The user fills out the textbox parameters with Event Name = test1 Event description = SF2 Start timestamp = 04/27/2020 End timestamp = 04/27/2021	Populate an event with necessary information	The system displays current input back to user	After this is done there is no apparent way to create an event once information has been provided
Concluding Remarks: The test fails since the Event configuration page has no functionality other than allowing the user to provide input into a text box.				
Testing Team: **Lead:** Alejandro Zamora Members: Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya			**Date Completed:** 04/27/2020	

4.13. Test SF3

Objective: The objective of this test is to check to see if the system can ingest data
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: SF3			**Current Status**: Fail	
Test Title: Start data ingestion				
Testing Approach: This test approach checks if the system allows the user can start data ingestion				
STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the PICK Tool application	Starts PICK Tool session	The system displays the GUI	Starts the system
2	The user clicks “accept and continue”	Starts PICK Tool session	The system displays the Team configuration window	Starts a PICK Tool session Figure 1
3	The user clicks the Directory option on the menu bar	Open the window to Directory configuration	The system displays the Directory configuration window	Figure 3
4	The user manually types out a path to a text file on their system in the red team text box and clicks the option start data ingestion	Provides a path to a file so the system can ingest	The system will go to the path provided and ingest data in specified location	The test fails right here since there is no functionality to the window
Concluding Remarks: The test fails since the Directory configuration page has no functionality other than allowing the user to provide input into a text box.				

Testing Team: **Lead:** Alejandro Zamora Members: Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya	**Date Completed:** 04/27/2020

4.14. Test SF4

Objective: The objective of this test is to check to see if the system can add, edit and delete vectors
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: SF4			**Current Status**: Fail	
Test Title: can add, edit and delete vectors				
Testing Approach: This test approach checks if the system allows the user can add, edit and delete vectors				
STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the PICK Tool application	Starts PICK Tool session	The system displays the GUI	Starts the system
2	The user clicks “accept and continue”	Starts PICK Tool session	The system displays the Team configuration window	Starts a PICK Tool session Figure 1
3	The user clicks the Vector option on the menu bar	Open the window to Vector configuration	The system displays the Vector configuration window	Figure 4
4	The user clicks the add vector option	Adds a vector to table	The system adds a vector the system	The test fails here turns out all buttons on this window have no functionality
Concluding Remarks: The test fails since the Vector configuration page has no functionality other than allowing the user to provide input into a text box.				
Testing Team: **Lead:** Alejandro Zamora Members: Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya			**Date Completed:** 04/27/2020	

4.15. Test SF5

Objective: The objective of this test is to check to see if the system can filter for specific key word
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: SF5			**Current Status**: Fail	
Test Title: Filter for specific key word				
Testing Approach: This test approach checks if the system allows the user can filter for specific key word				
STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the PICK Tool application	Starts PICK Tool session	The system displays the GUI	Starts the system

2	The user clicks “accept and continue”	Starts PICK Tool session	The system displays the Team configuration window	Starts a PICK Tool session Figure 1
3	The user clicks the filter option on the menu bar	Open the window to filter configuration	The system displays the Filter configuration window	Figure 6
4	The user examines the window to see if there is any apparent way to filter a key word	Check for functionality	The system displays the Filter configuration window	From examining figure 6 you can determine this window has no functionality
Concluding Remarks: The test fails since the Filter configuration page has no functionality other than allowing the user to provide input into a text box.				
Testing Team: **Lead:** Alejandro Zamora Members: Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya			**Date Completed:** 04/27/2020	

4.16. Test SF6

Objective: The objective of this test is to check to see if the system can Export a project
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: SF6			**Current Status**: Fail	
Test Title: Export a project				
Testing Approach: This test approach checks if the system allows the user can export a project				
STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the PICK Tool application	Starts PICK Tool session	The system displays the GUI	Starts the system
2	The user clicks “accept and continue”	Starts PICK Tool session	The system displays the Team configuration window	Starts a PICK Tool session Figure 1
3	The user clicks the Export option on the menu bar	Open the window to export	The system displays the export window	Figure 8
4	The user clicks the option export located at the bottom of the window	Continues the process to export	The system will export the project	Test fails right here since there is no functionality
Concluding Remarks: The test fails since the Export page has no functionality.				
Testing Team: **Lead:** Alejandro Zamora Members: Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya			**Date Completed:** 04/27/2020	

4.17. Test SF7

Objective: The objective of this test is to check to see if the system can push and pull a project
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: SF7			**Current Status**: Fail	
Test Title: Push and pull a project				
Testing Approach: This test approach checks if the system allows the user can push and pull a project				
STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the PICK Tool application	Starts PICK Tool session	The system displays the GUI	Starts the system
2	The user clicks “accept and continue”	Starts PICK Tool session	The system displays the Team configuration window	Starts a PICK Tool session Figure 1
3	The user clicks the vector DB option on the menu bar	Open the window to Vector DB	The system displays the Vector DB	Figure 10
4	The user clicks the option Push located at the bottom of the window	Continues the process to push a project	The system will push the project	Test fails right here since there is no functionality
Concluding Remarks: The test fails since the Vector DB page has no functionality.				
Testing Team: **Lead:** Alejandro Zamora Members: Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya			**Date Completed:** 04/27/2020	

4.18. Test SF8

Objective: The objective of this test is to check to see if the system can Add, delete and edit Icon
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: SF8			**Current Status**: Fail	
Test Title: Add, delete and edit Icon				
Testing Approach: This test approach checks if the system allows the user can Add, delete and edit Icon				
STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the PICK Tool application	Starts PICK Tool session	The system displays the GUI	Starts the system
2	The user clicks “accept and continue”	Starts PICK Tool session	The system displays the Team configuration window	Starts a PICK Tool session Figure 1
3	The user clicks the Icon option on the menu bar	Open the window to Icon configuration	The system displays the Icon configuration	Figure 11
4	The user clicks the option Add Icon located at the bottom of the window	Add and Icon	The system will update and add an Icon to the system	Test fails right here since there is no functionality
Concluding Remarks: The test fails since the Icon configuration page has no functionality.				

Testing Team: **Lead:** Alejandro Zamora Members: Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya	**Date Completed:** 04/27/2020

4.19. Test SF9

Objective: The objective of this test is to check to see if the system can Build a graph
Precondition: The PICK Tool (picktool.py) is visible inside its desired folder.

Test No.: SF9			**Current Status**: Fail	
Test Title: Build a graph				
Testing Approach: This test approach checks if the system allows the user can build a graph				
STEP	**OPERATOR ACTION**	**PURPOSE**	**EXPECTED RESULTS**	**COMMENTS**
1	The user executes the PICK Tool application	Starts PICK Tool session	The system displays the GUI	Starts the system
2	The user clicks “accept and continue”	Starts PICK Tool session	The system displays the Team configuration window	Starts a PICK Tool session Figure 1
3	The user clicks the Graph Builder option on the menu bar	Open the window to Graph Builder	The system displays the Graph Builder	Figure 12
4	The user clicks the option add node located at the bottom of the window	Add a node to the graph	The system will update and add a node to the system	Test fails right here since there is no functionality
Concluding Remarks: The test fails since the Graph Builder page has no functionality.				

Testing Team: **Lead:** Alejandro Zamora Members: Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya	**Date Completed:** 04/27/2020

5. User Interface Testing

Section 5 focuses on the interaction between the user and the system. This testing includes the following traits: Consistent terminology, menu selections, and presentation, grammar, and error handling that will inform user of critical operations.

5.1. Testing Disclaimer

Section 4 (GUI Functionality Test Suite) encompasses the totality of Section 5 in its tests.

6. Test Schedule

Section 6 specifies the schedule for testing activities as they pertain to PICK Tool.

6.1. Test Table

The table below shows the test schedule that Team404 will follow for the test plan process. The start date for testing is scheduled for April 22, 2020 and the target date to complete the tests is the same. A team member will be responsible for conducting one of the tests as described in each test suite.

Date	Task	People	Description
04/14/2020	GF1	Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya, Alejandro Zamora	Traversability between views to another view
04/26/2020	GF2	Alejandro Zamora, Jorge Felix	Test to see if menu bar works on different x/y coordinates
04/26/2020	GF3	Alejandro Zamora, Jorge Felix	Persistent Information held across tabs
04/27/2020	GF4	Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya, Alejandro Zamora	Resize of the PICK Tool window
04/27/2020	GF5	Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya, Alejandro Zamora	Can run multiple instances of PICK Tool
04/27/2020	GF6	Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya, Alejandro Zamora	Menu bar functionality when PICK Tool window is set to minimal
04/27/2020	GF7	Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya, Alejandro Zamora	Can handle large paragraphs of text as input without messing up GUI formatting
04/27/2020	GF8	Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya, Alejandro Zamora	Check to see if windows with tables are fully functional when PICK Tool window size is set to minimal

04/27/2020	GF9	Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya, Alejandro Zamora	Can recover menu bar once it has been disabled for any reason
04/27/2020	GF10	Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya, Alejandro Zamora	When inputting information does undo and redo options in text fields work
04/27/2020	SF1	Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya, Alejandro Zamora	Connect to the lead IP address
04/27/2020	SF2	Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya, Alejandro Zamora	Creation of event
04/27/2020	SF3	Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya, Alejandro Zamora	Start data ingestion
04/27/2020	SF4	Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya, Alejandro Zamora	Add, edit, and delete vector
04/27/2020	SF5	Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya, Alejandro Zamora	Filter for specific key word
04/27/2020	SF6	Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya, Alejandro Zamora	Export a project
04/27/2020	SF7	Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya, Alejandro Zamora	Push and pull a project
04/27/2020	SF8	Jacob Torres, Eddy	Add, delete and edit Icon

		Todd, Jorge Felix, Matt Montoya, Alejandro Zamora	
04/27/2020	SF9	Jacob Torres, Eddy Todd, Jorge Felix, Matt Montoya, Alejandro Zamora	Build a graph

7. Other Sections

Section 7 contains other sections. These requirements come from the SRS Document, written by the guidance team, and the code, written by Team404.

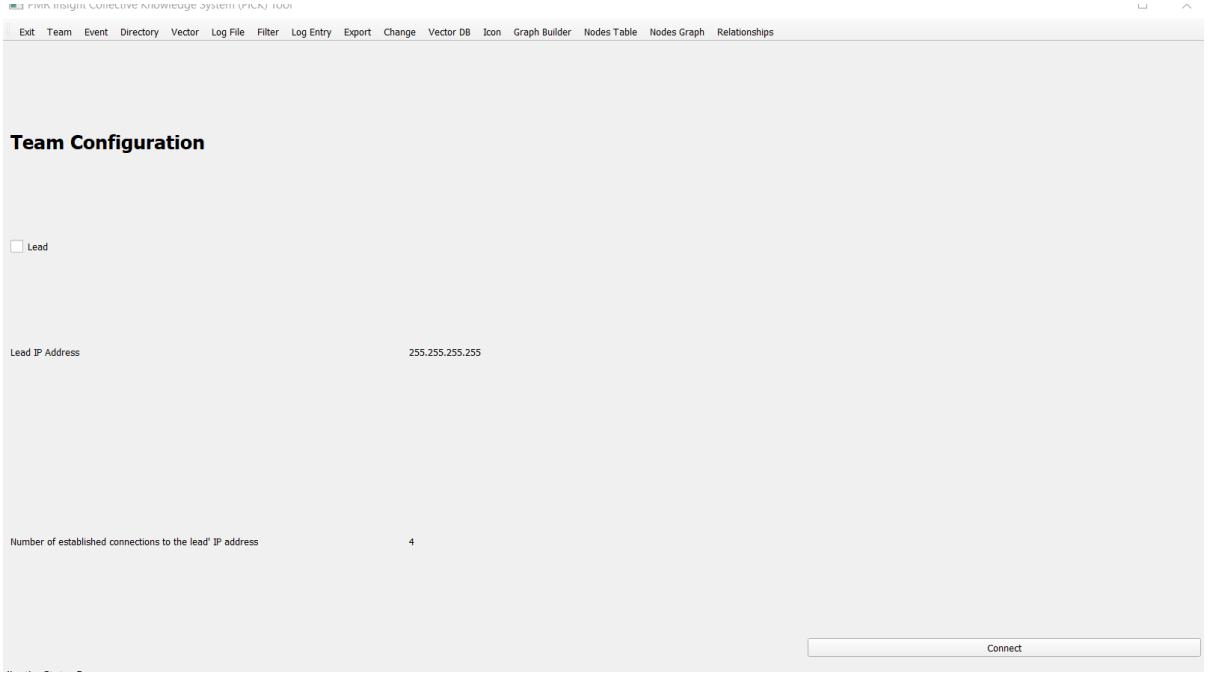
7.1. Disclosure

At this time, Team404 does not have any other sections to add.

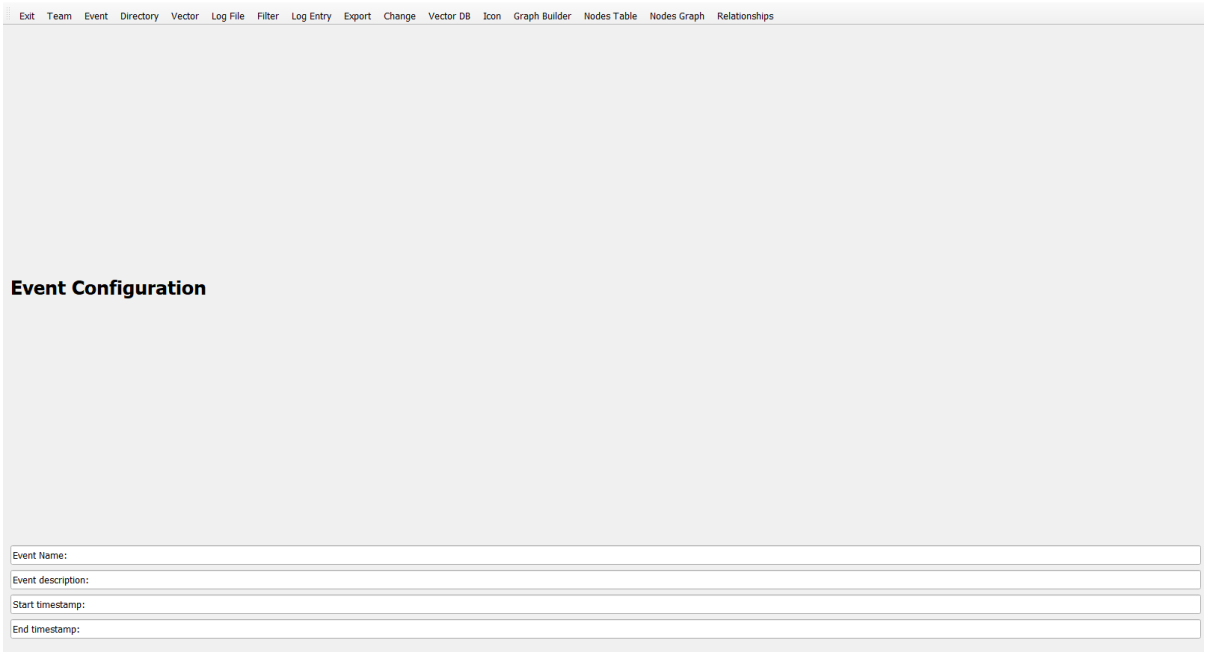
8. Appendix

Section 8 contains an appendix of figures (or images) depicting the GUI. These figures are referenced throughout the PICK Tool Test Plan

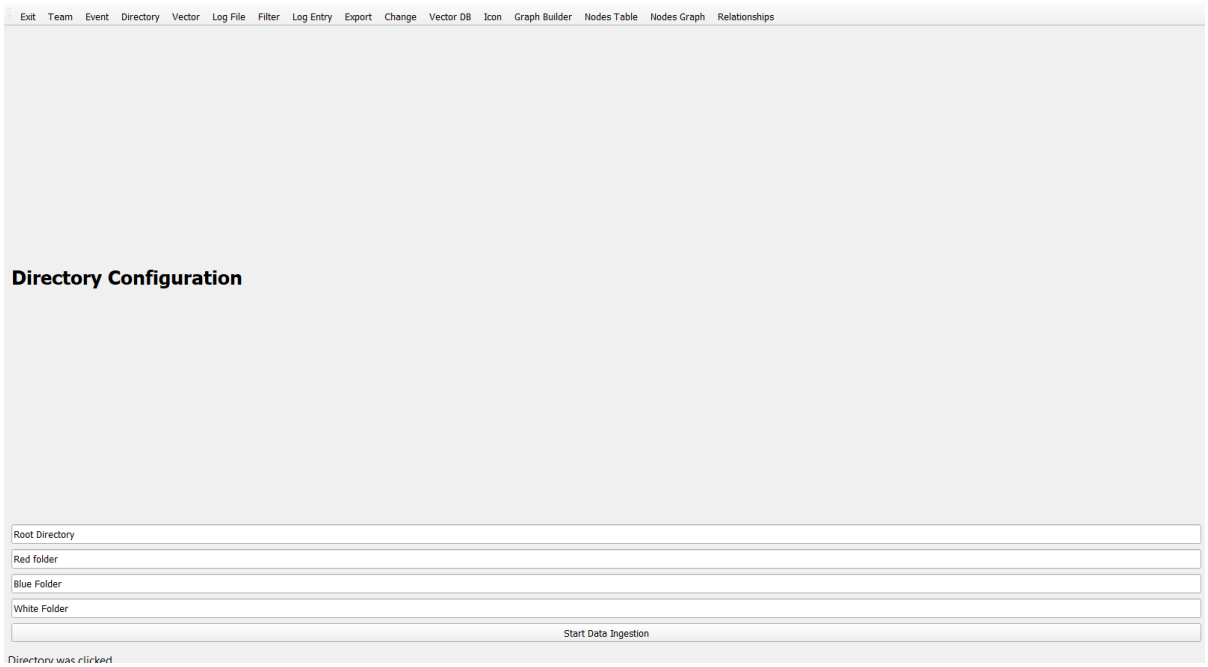
8.1. Figure 1



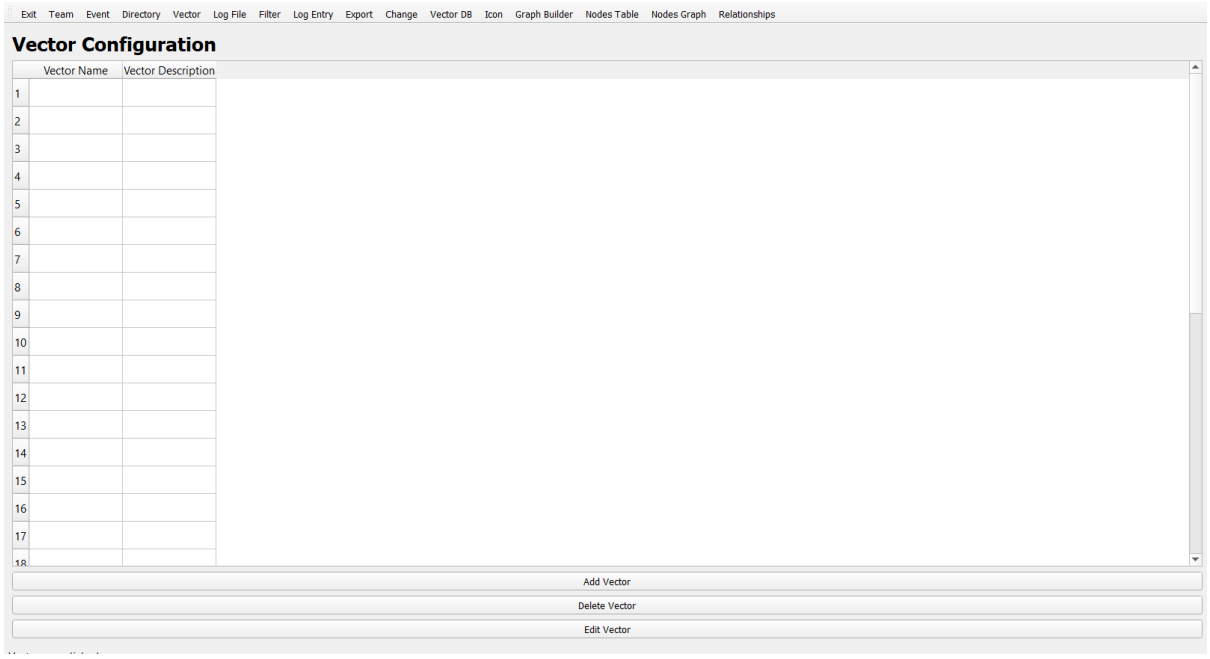
8.2. Figure 2



8.3. Figure 3



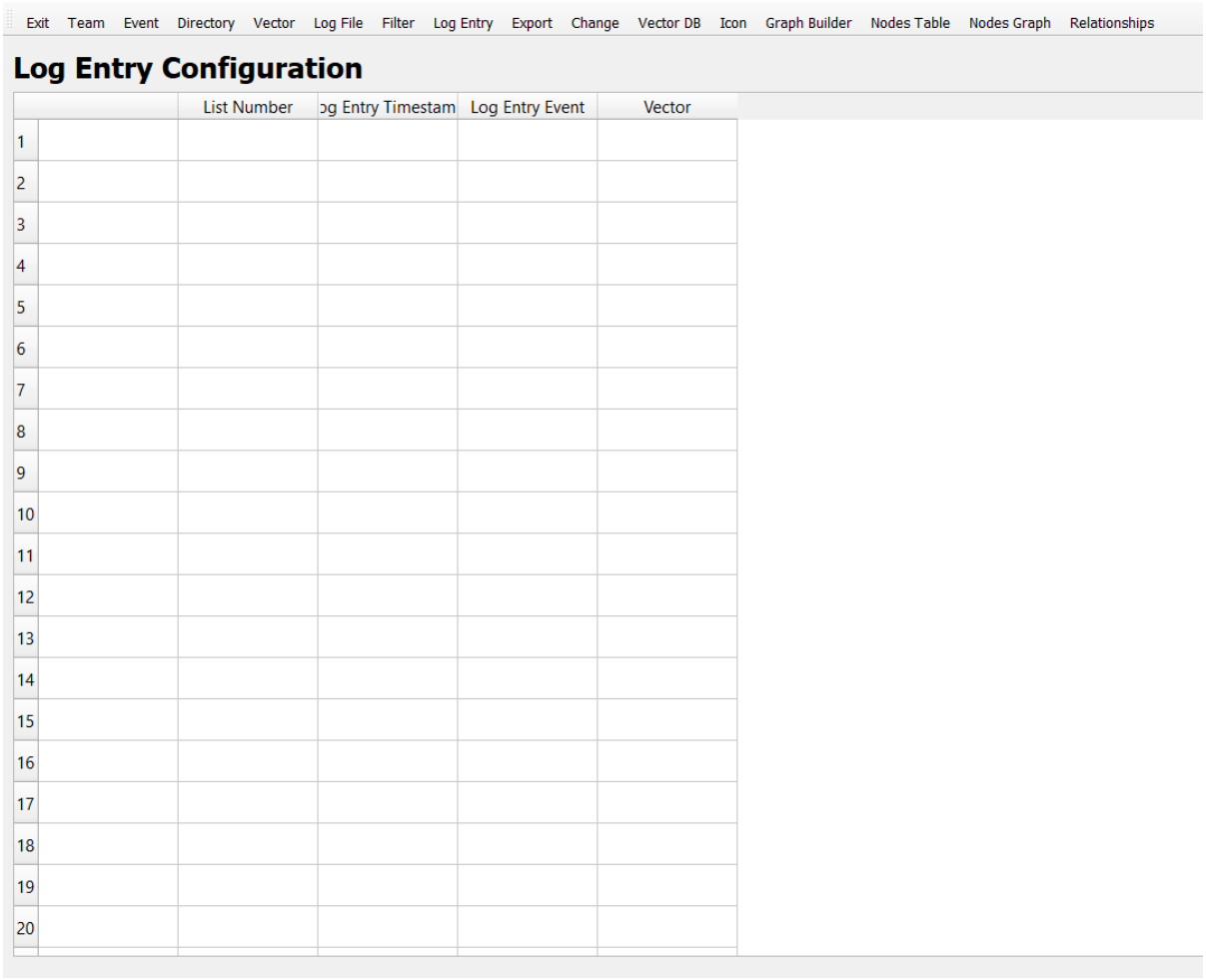
8.4. Figure 4



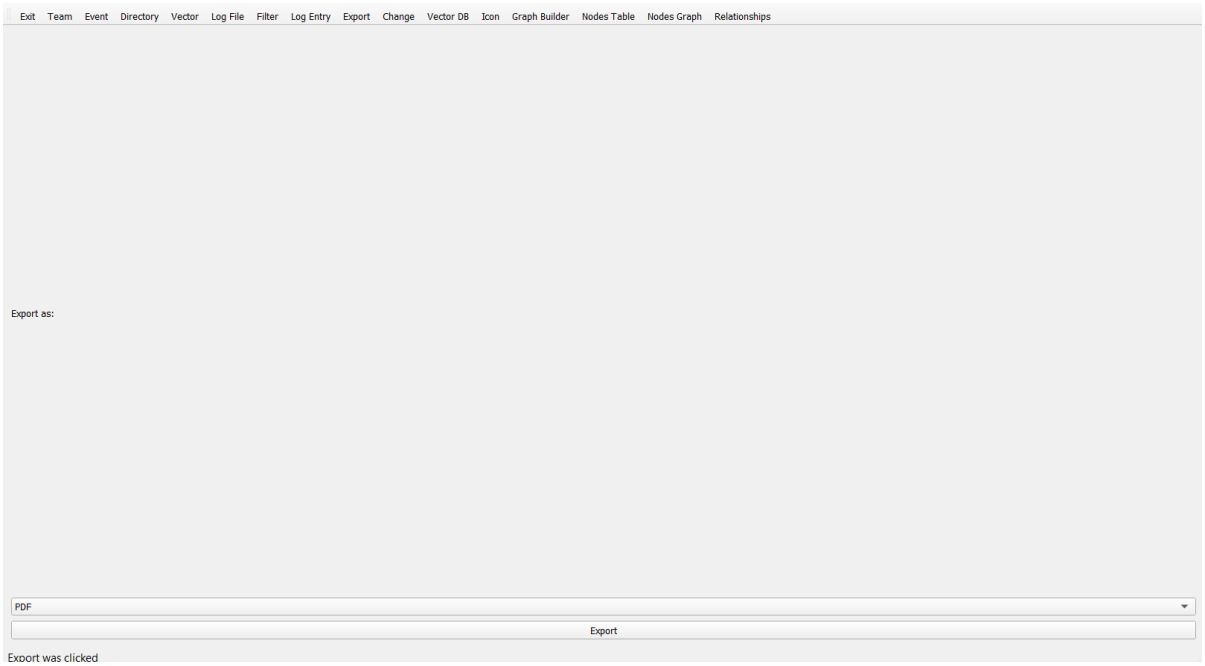
8.5. Figure 5

8.6. Figure 6

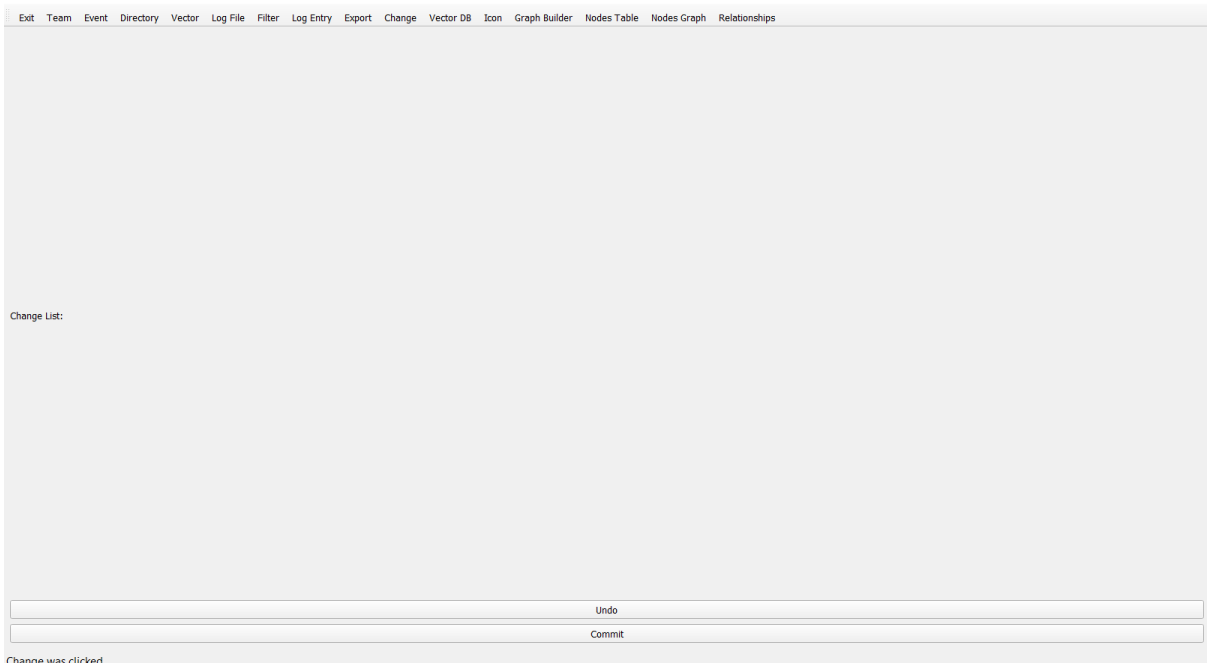
8.7. Figure 7



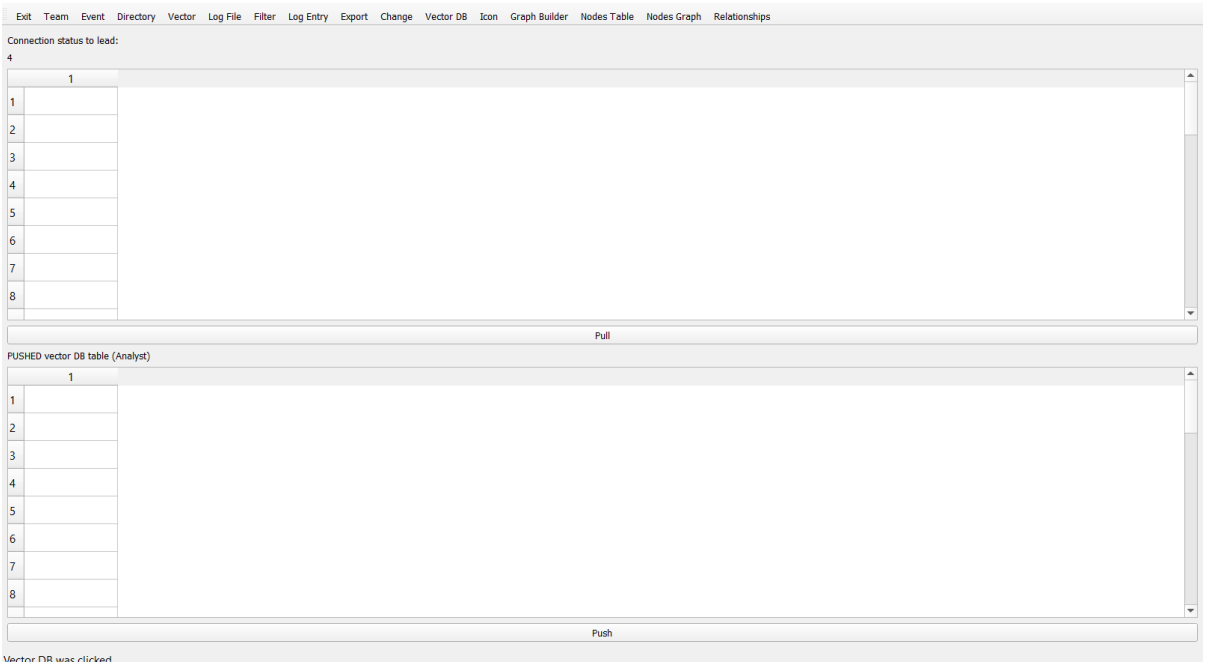
8.8. Figure 8



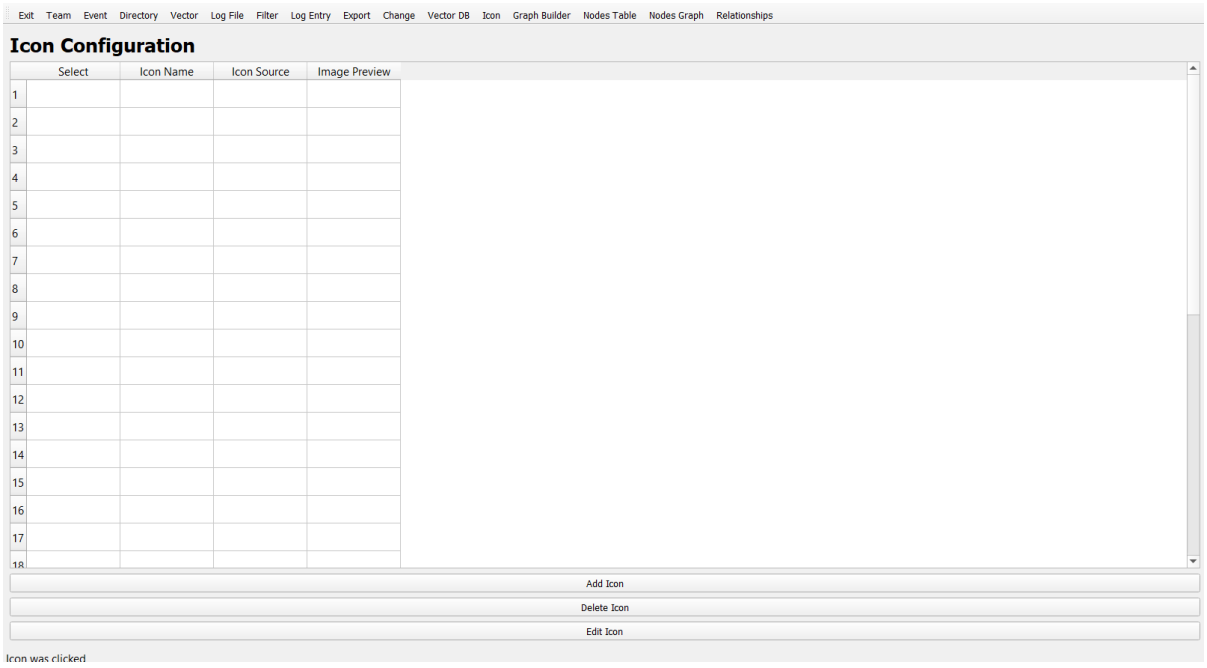
8.9. Figure 9



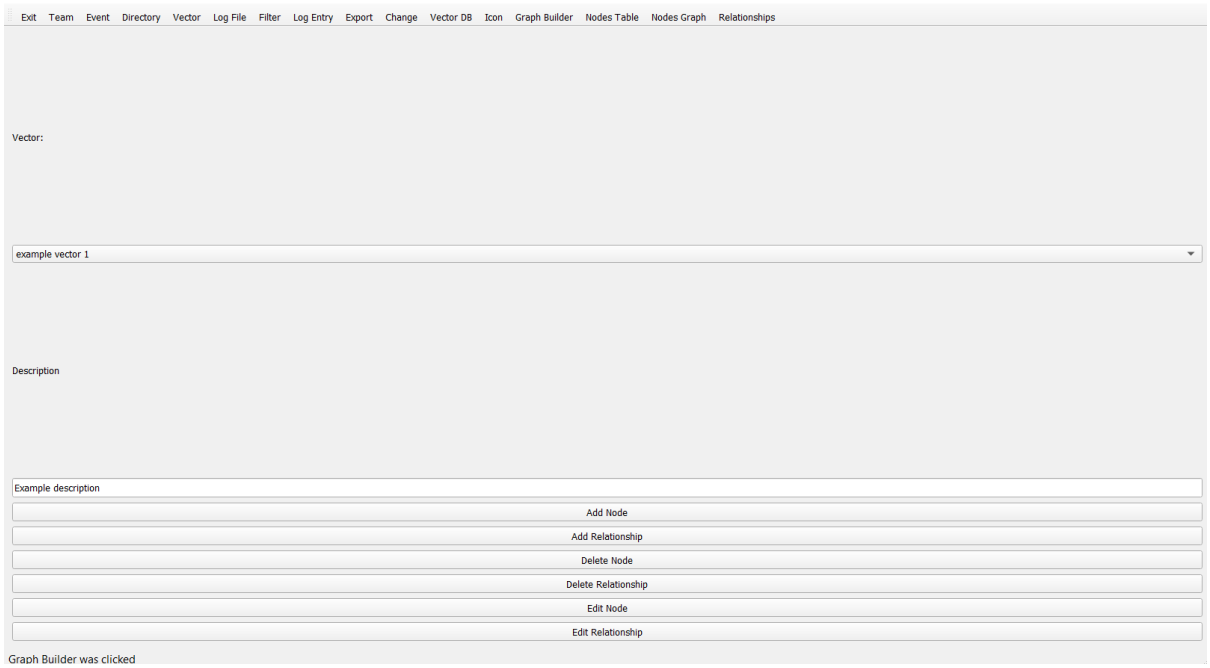
8.10. Figure 10



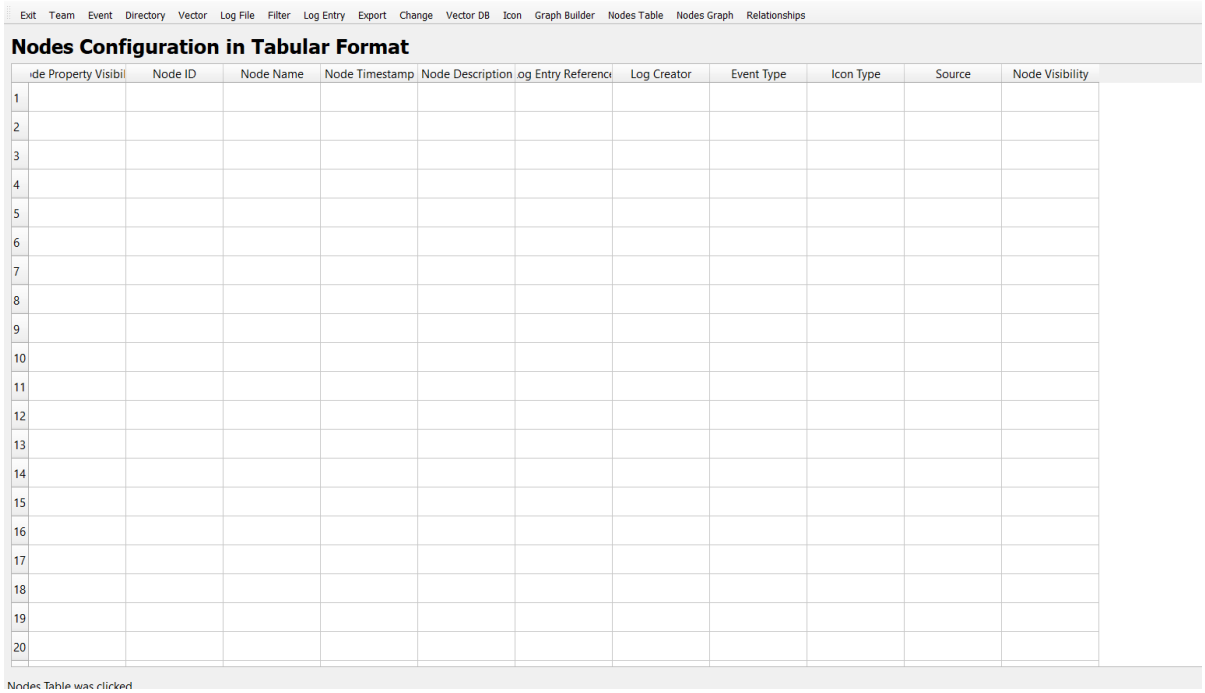
8.11. Figure 11



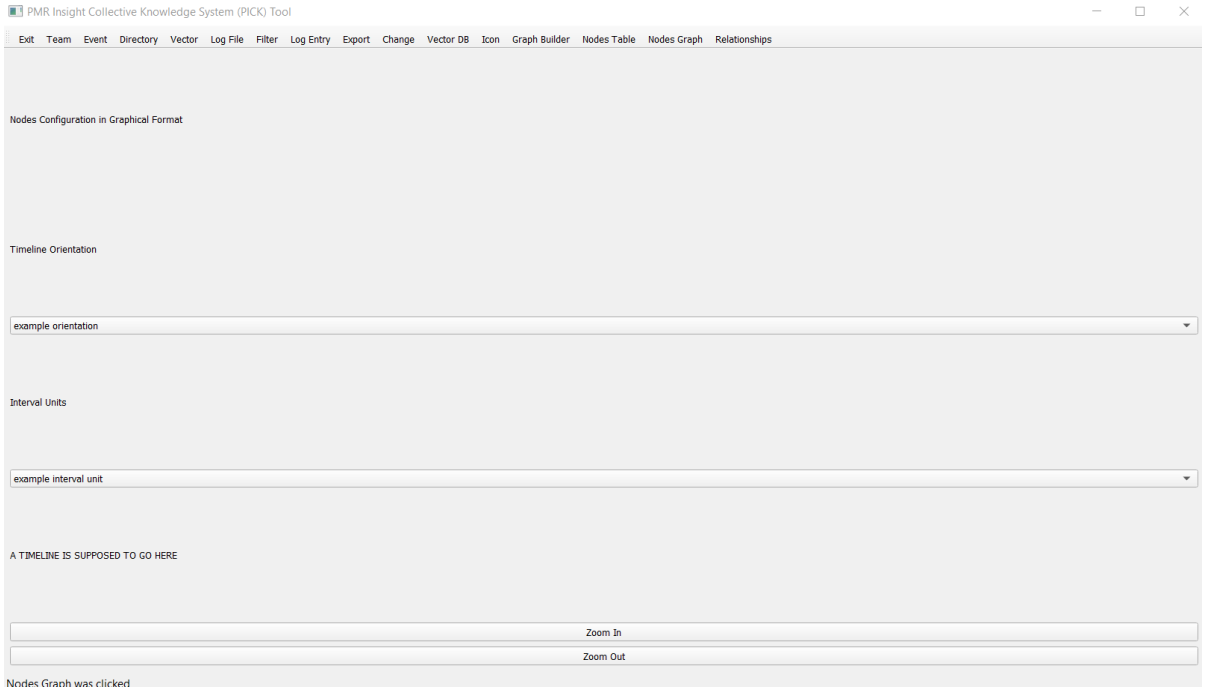
8.12. Figure 12



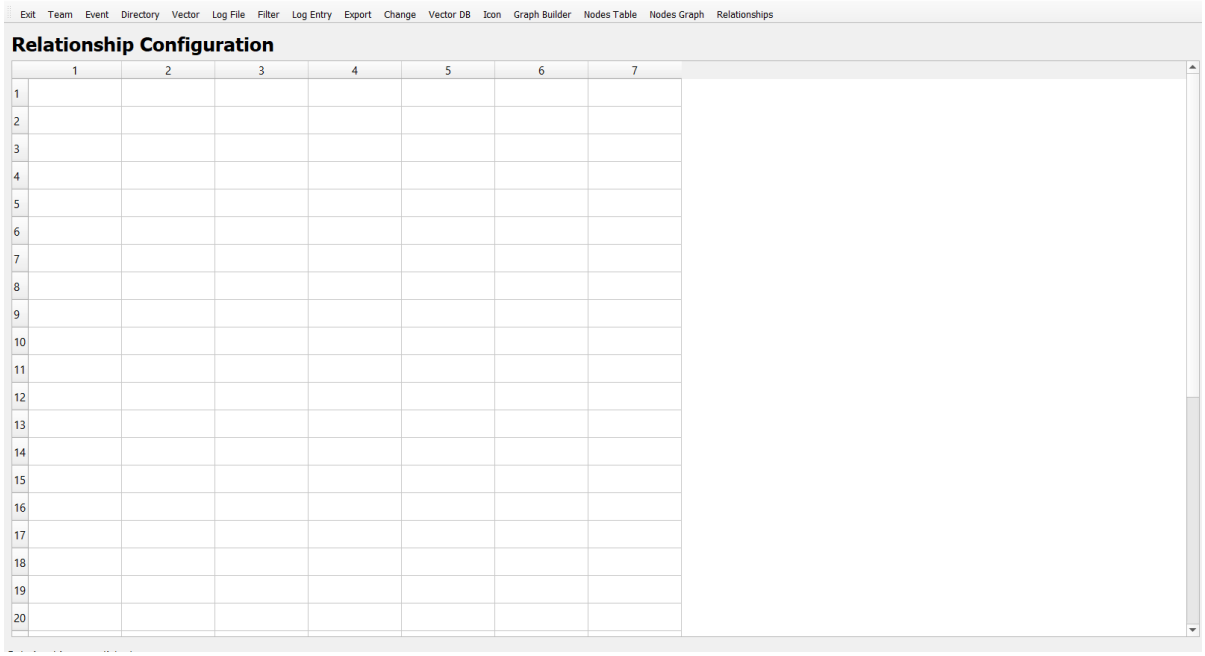
8.13. Figure13



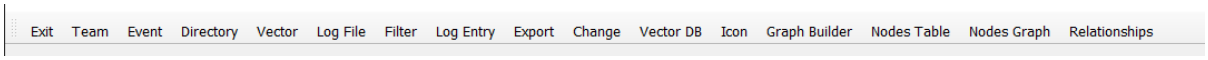
8.14. Figure 14



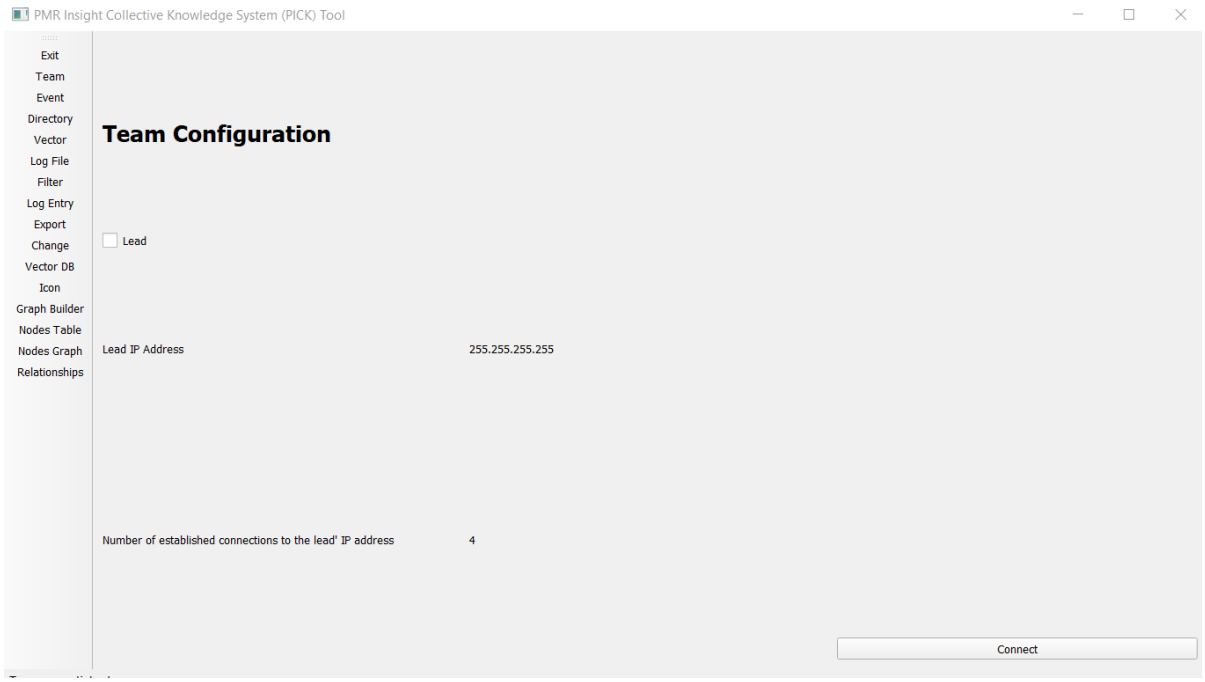
8.15. Figure 15



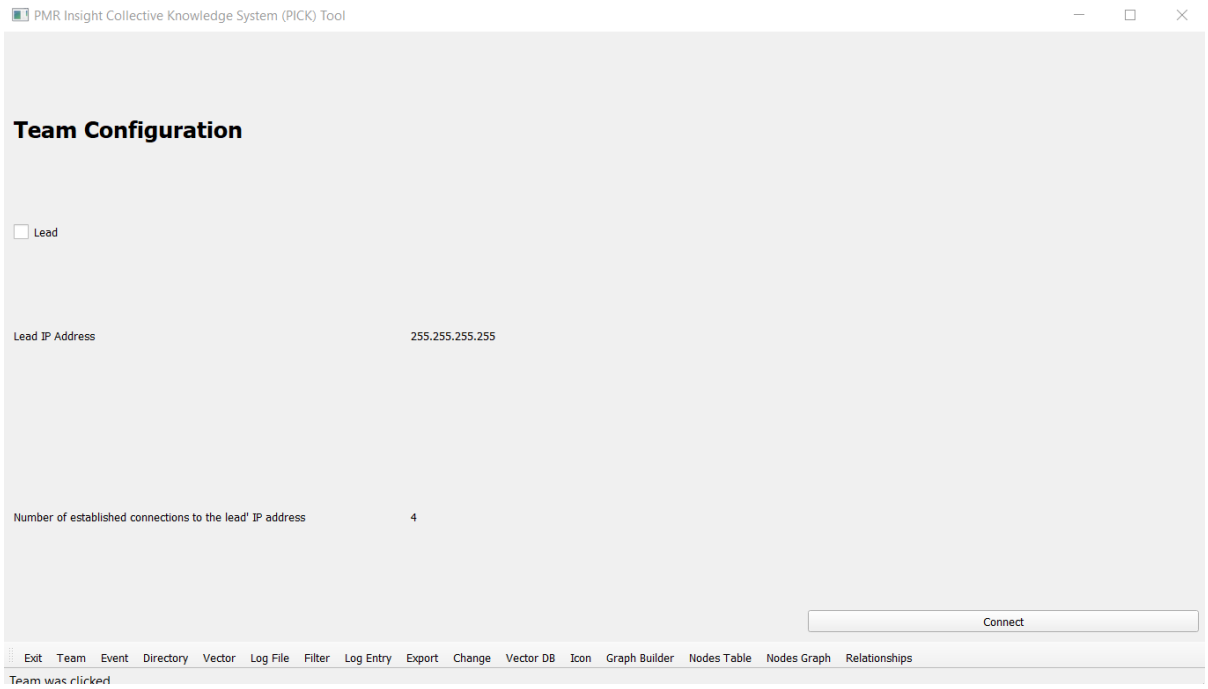
8.16. Figure 16



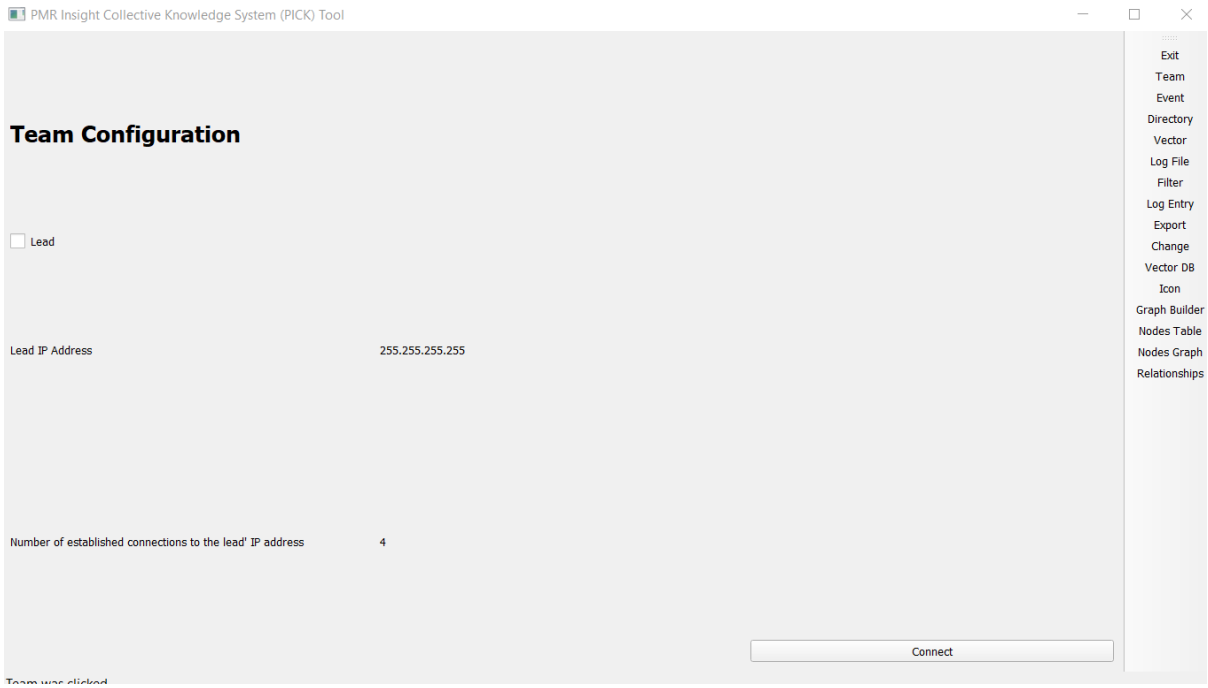
8.17. Figure 17



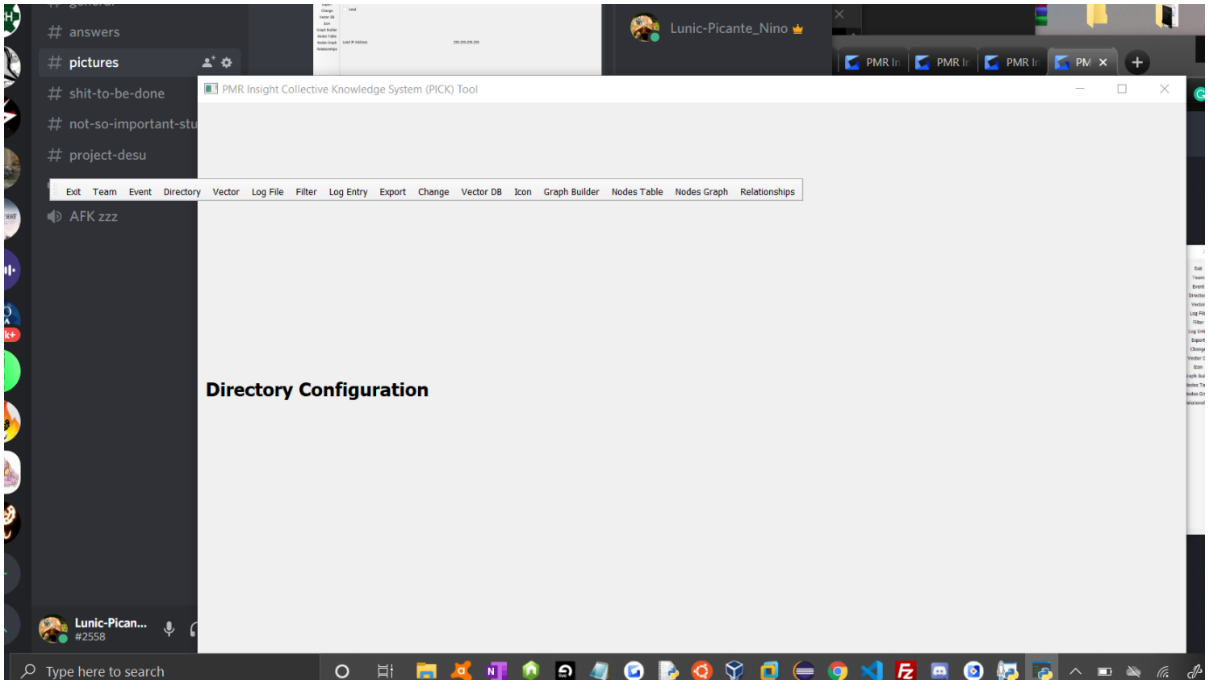
8.18. Figure 18



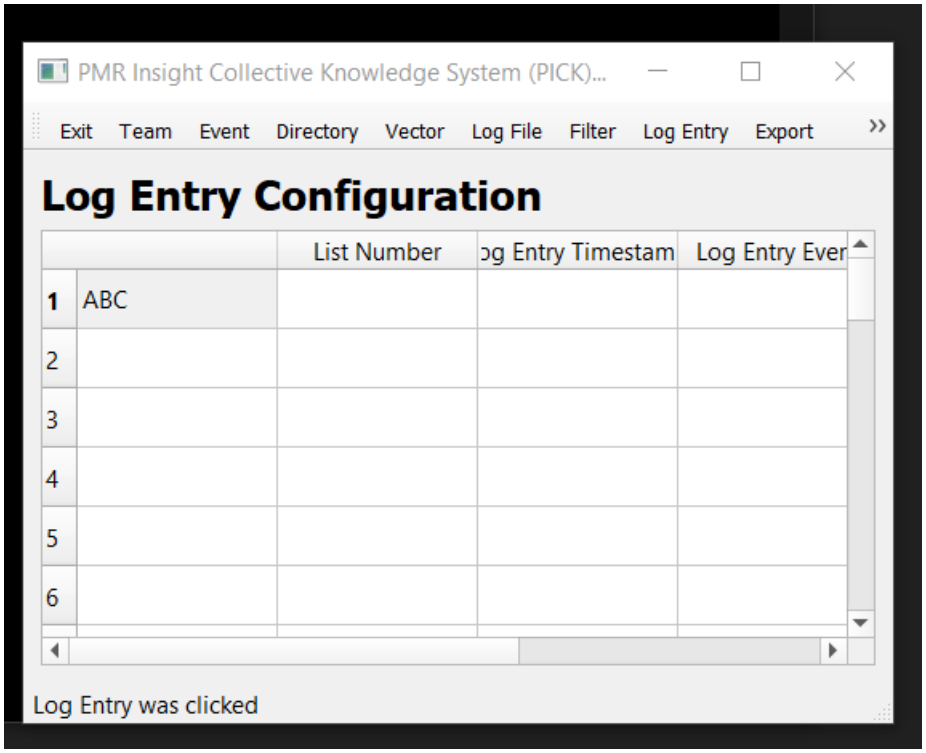
8.19. Figure 19



8.20. Figure 20



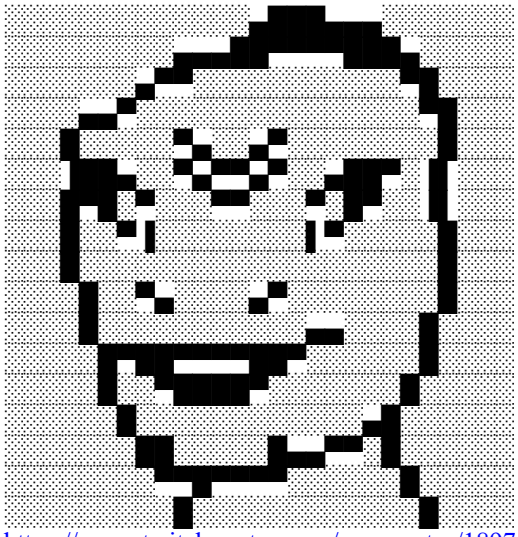
8.21. Figure 21



8.22. Text 1

Now, this is a story all about how
My life got flipped-turned upside down
And I'd like to take a minute
Just sit right there
I'll tell you how I became the prince of a town called Bel Air
In west Philadelphia born and raised
On the playground was where I spent most of my days
Chillin' out maxin' relaxin' all cool
And all shootin some b-ball outside of the school
When a couple of guys who were up to no good
Started making trouble in my neighborhood
I got in one little fight and my mom got scared
She said 'You're movin' with your auntie and uncle in Bel Air'
https://www.azlyrics.com/lyrics/djjazzyjeffthefreshprince/freshprinceofbelairthemesong.html

8.23. Text 2


https://www.twitchquotes.com/copypastas/1897

8.24. Text 3

ABC 123!

$