Keikaku 企画

PMR Insight Collective Knowledge (PICK)
Test Plan
Version 1.1
05/07/2020

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

Initial Release:	0.1
Current Release:	1.1
Indicator of Last Page in Document:	[END]
Date of Last Review:	05/07/2020
Date of Next Review:	N/A
Target Date for Next Update:	N/A

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. Ann Gates, Dr. Steve Roach, Mr. Jake Lasley

Clients:
Mr. Vincent Fonseca, Mr. Baltazar Santaella, Ms. Herandy Vazquez, and Mr. Erick De Nava

Software Team Members:
Mr. Anthony Desarmier, Mr. Angel Villalpando, Mr. Mario Delgado, Mr. David Rayner, Mr. Valentin Becerra, Mr. Jorge Garcia

Change Summary

The following table details changes made between versions of this document

Version	Date	Modifier	Description
0.1	04/07/2020	Anthony DesArmier	Added Template
0.2	04/14/2020	Angel Villalpando	Completed sections 1.1-1.4
0.2	04/15/2020	David Rayner	Completed 1.5, added suite to section 3, and two test cases to section 4.
0.2	4/15/2020	Anthony DesArmier	Formatting, grammar
0.3	04/27/2020	Angel Villalpando	Completed Test case tables for TBM 4, 5, 6.
0.3	04/27/2020	Jorge Garcia	Completed Test case tables for TBM 1-3 and 7-9.
0.3	04/27/2020	David Rayner	Completed test case tables for ING 1-5 and DP 1-2. Completed section 6.
0.3	04/27/2020	Valentin Becerra	Completed test case tables for Graphing GPH 1-7
0.3	04/27/2020	Anthony DesArmier	Completed test case tables for Data persistence DP 3-5.
0.3	04/27/2020	Mario Delgado	Completed test case tables for Data persistence DP 6-7.

0.4	04/28/2020	David Rayner	Updated ING 3-6
1.0	04/28/2020	Anthony DesArmier	Updated all Test Suites and Test Cases. Formatting, grammar.
1.1	05/05/2020	Jorge Garcia	Updated some test suite and added screen shots.
1.1	05/07/2020	Angel Villalpando	Performed tests for TBM 1-6; GPH 1-4. Made some grammatical corrections.

TABLE OF CONTENTS

1. Introduction

Purpose

The purpose of this document is to outline the Test Plan for the PMR Insight Collective (PICK) system. This document will include the organizational responsibilities, the test approach, and the test schedule. This document will primarily discuss testing from the customer's point of view and should not be considered a general testing strategy, an integration test plan, or a unit test plan. By conducting the test cases proposed in this document, the customer should be able to demonstrate that the system performs that which it is intended to do.

Scope

The PMR Insight Collective Knowledge (PICK) is the software system for which this Test Plan is written for. PICK is a software system to help Prevent, Mitigate, and Recover Analysts analyze vast amounts of data collected during an Adversarial Assessment (AA) by allowing them to quickly search through, view, correlate, and build visual documents which help explain the AA itself to uninvolved personnel. The customers - in this case PMR Analysts - currently must sift through the vast amounts of generated data from the AA by hand which severely hinders their workflow and efficiency in developing a report with visual aids for which to explain the nature of the AA to other personnel.

PICK will allow the customers to insert all the data generated from an AA into its system and display an organized, searchable database of that information. The customers can then quickly and efficiently find and correlate relevant data events together and help craft timelines which describe the significant events and their relations to one another during the AA. PICK will then assist the customers in crafting a visual representation of these series of events as attack graphs in order to help visualize the timeline of the AA. This assistance of analyzing the data generated by the AA and constructing visual representations of significant events will substantially reduce the time and work hours needed by the customers to understand and construct a report on the results of the AA to deliver to other personnel.

System Overview

The PICK system utilizes several python libraries for the graphical user interface which must be tested to ensure that they perform their desired tasks. Additionally, the system heavily interacts with the Splunk Extract, Load, and Transform (ELT) system. The interaction with this system requires testing to ensure that the data sent to a from it follow the specifications outlined by the design. Finally, ensuring that the system correctly creates vectors, each with respective log entries, is important to the overall success of this system. These items are the focal points for the testing outlined in this document.

Suspension and Exit Criteria

If at any point a critical test fails, testing will be suspended. Critical tests are intended to assess the functionality of the major components within the system. If any of these major components are not functioning as intended, several subsequent tests dependent on this component will also fail or will not be testable. For this reason, testing shall be suspended source code redeveloped to restore functionality to such major components.

Once all critical tests have passed, testing shall be complete and the system shall satisfy the core requirements laid out in the initial specification of the system.

Document Overview

The test plan document consists of the following sections:

Introduction:
This section describes the overview of the testing plan. It includes the purpose of the document, the overall scope of the project, and the suspension, exit criteria regarding system tests to be run.

Test Items and Features:
This section describes the testing items (e.g. components, classes, functions or methods) and the features to be tested.

Testing Approach:
This section describes the testing approach we the development team are to establish. The type of tests to be run in order to test system functions. Each test is to contain a description and unique test identifier.

Test cases:
This section describes the tests that were run, including test input, test procedures and outcomes. Each test is divided by the following sections: test number, current status, title, approach, step, operator action, purpose, expected results, comments, remarks, conclusion, date completed, and team that performed the test.

User Interface Testing:
This section describes the interaction between the system and user components Including consistent terminology, shortcut keys, menu selections, and presentation, flexibility in navigation between windows and interface elements and potential error handling that will inform user of critical operations.

Test Schedule:
This section describes the completion dates of each test.

Other:
This section describes the other potential test documentation such as:

- 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:
References of expected output and explicit directions for analysis of output.

References

[1] V. Becerra, A. DesArmier, J. Garcia, D. Rayner, A. Villalpando and Mario Delgado, "Keikaku_SDD_V2," El Paso, 2020.

2. Test Items and Features

Feature: File Ingestion
Class: Validator
Class: SplunkManager

Feature: File Cleansing
Class Validator

Feature: File Validation
Class: Validator
Class: EnforcementActionReport

Feature: Log entry to vector assignment
Class: LogEntry
Class: IDDict
Class: Vector

Feature: Sort/Filter log entries and nodes
Class: Sort
Class: Filter

Feature: Export vector table
Class: ExportGraph
Class: Vector

Feature: Export vector graph
Class: ExportTable
Class: Vector

Feature: Graphing
Class: GraphEditor
Class: GraphEditorScene
Class: GraphEditorView
Class: GraphEditorWindow
Class: NodeItem
Class: RelationshipItem
Class: VectorItemGroup

Feature: Search and Filter
Class: Sort
Class: Filter

Feature: Data storage
Storage is to be done through a file system (pickle serialization operation)

Feature: Lead-Host data management
Class: Sync
Class: ProjectMerge

Feature: Commit management
Class: History

3. Testing Approach

Table 1.

TEST SUITE <Ingestion>		
Description of Test Suite	The following test suite is to evaluate the functionality of the start ingestion process the system is to perform.	
Test Case Identifier	**Objective**	**Criticality**
ING 1	Open Event configuration dialog in response to File->Event selection. Save Event configuration (name, description, start, and end times) in response to save button clicked.	Critical
ING 2	Open directory configuration in response to Directory button clicked. Start ingestion process once valid directories (root, red, white, and blue) specified.	Critical
ING 3	Create copies of root directory files. Initiate cleansing operation on root directory files.	Critical
ING 4	Initiate validating operation on cleansed root directory files. Generate enforcement action reports for invalid (non-ingested) files.	Critical
ING 5	Initiate ingestion operation on validated root directory files. Populate log entry table with ingested parsed entries. (if log entries have made it to Splunk).	Critical
ING 6	Initiate ingestion operation on invalid root directory files (force ingestion).	Normal

Table 2.

TEST SUITE <Table Modifications>		
Description of Test Suite	The following test suite is to evaluate the modifications (add, remove, and editing) on the following tables: Vector, Node, and Relationship.	
Test Case Identifier	**Objective**	**Criticality**
TBM 1	Add and Remove entries to Vector table.	Critical
TBM 2	Edit entries on Vector table.	Normal
TBM 3	Add and Remove entries to Node table.	Critical
TBM 4	Edit entries in Node table.	Normal
TBM 5	Add and Remove entries to Relationship table.	Critical
TBM 6	Edit entries on Relationship table.	Normal

Table 3.

TEST SUITE <Graphing>		
Description of Test Suite	The following test suite is to evaluate the functionality of the graph editing process.	
Test Case Identifier	**Objective**	**Criticality**
GPH 1	Add a node to the graph.	Critical
GPH 2	Add a relationship to the graph.	Critical
GPH 3	Remove a node from the graph.	Normal
GPH 4	Remove a relationship from the graph.	Normal
GPH 5	Move nodes and relationships on the graph.	Normal
GPH 6	Toggle visibility of node elements on the graph.	Normal
GPH 7	Changes made on table views reflects on the graph.	Normal

Table 4.

TEST SUITE <Data Persistence>		
Description of Test Suite	The following test suite is to evaluate the data persistence of the system. This includes (Event, Vector, Log File, Log Entry, Directory and Node, Relationship) configuration's data and the Graph's data.	
Test Case Identifier	**Objective**	**Criticality**
DP 1	Event configuration save and load.	Normal
DP 2	Directory configuration save and load.	Normal
DP 3	Log File configuration save and load.	Normal
DP 4	Log Entry configuration save and load.	Normal
DP 5	Vector configuration save and load.	Normal
DP 6	Node configuration save and load.	Normal
DP 7	Relationship configuration save and load.	Normal

4. Tests

Ingestion

Objective: To establish proper functionality of the start ingestion process.
Notes: Access to different test files with various data.

Test No.: ING 1		Current Status: Passed		
Test title: Create event details.				
Testing approach: This test will be conducted on the event configuration dialog. Field inputs are selected and output messages are observed.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Start system. Click “File->Event”.	Display the Event Configuration dialog.	Event Configuration dialog opens.	
2	Click “Save Event” button.	Save event with missing fields.	Prompt stating “name or description” input fields empty.	Input fields “Name” and “Description” should be empty.
3	Dismiss prompt. Enter “Event A” in “Event Name” field. Click “Save Event” button.	Save event with one field (name) empty.	Prompt stating “name or description” input fields empty.	Input field “Description” should be empty.
4	Dismiss prompt. Clear “Event Name” field. Enter “Test description” in “Event Description” field. Click “Save Event” button.	Check with one field (description) empty.	Prompt stating “name or description” input fields empty.	Input field “Name” should be empty.
5	Dismiss prompt. Enter “Event A” in “Event Name” field. Click “save event”.	Check if time is in valid range.	Prompt stating “invalid end time”.	Both “Event Start Time” and “Event End Time” fields are “12:00 01/01/2000 AM”. End time should be after start time to be valid.
6	Set start time to “12:00 01/01/2001 AM”.	Check if time is in valid range.	Prompt stating “invalid end time”.	End time should be after start time to be valid.
7	Set end time to “12:00 01/01/2010 AM”.	Check if all fields are valid	Prompt stating “event saved”.	Event configuration has been created.
Concluding Remarks: Tests provided the correct response prompts.				
Testing Team: Keikaku		Date Completed: 04/15/2020		

Test No.: ING 2			Current Status: Passed	
Test title: Save team directory paths.				
Testing approach: This test will be conducted on the directory configuration dialog. Directory paths are selected and output messages are observed.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Perform ING 1.	Create Event Configuration.	Event Configuration has been created.	Initial condition.
2	Click on the “Directory” button on the Log File configuration tab.	Display the Directory Configuration dialog.	Directory Configuration window is displayed.	
3	Remove text from all fields. Click on the “Start Data Ingestion” button.	Check with all fields empty.	Prompt stating “field is empty”.	
4	Click the “browse” button next to the “Root Directory” field and select a folder named “Root”.	Check to see if browse file picker works. Populate “Root Directory” field.	“Root Directory” field is populated with name of the folder “Root”.	
5	Click the “browse” button next to the “Red Team Folder” field and select a folder named “RedTeam” found within the Root folder.	Populate “Red Team Folder” field.	“Red Team Folder” field is populated with name of the folder “RedTeam”.	
6	Click the “browse” button next to the “Blue Team Folder” field and select a folder named “BlueTeam” found within the Root folder.	Populate “Blue Team Folder” field.	“Blue Team Folder” field is populated with name of the folder “BlueTeam”.	
7	Click the “browse” button next to the “White Team Folder” field and select a folder named “WhiteTeam” not found within the Root folder.	Populate “White Team Folder” field. Construct an invalid directory structure.	“White Team Folder” field is populated with name of the folder “WhiteTeam”.	
8	Click the “Start Ingestion” button.	Check if directory structure is valid.	Prompt stating “Directory not found in Root”.	
9	Click the “browse” button next to the “White Team Folder” field and select a folder named “WhiteTeam” found within the Root folder.	Populate “White Team Folder” field. Construct a valid directory structure.	“White Team Folder” field is populated with name of the folder “WhiteTeam”.	
10	Click the “Start Ingestion” button.	Check if directory structure is valid.	Prompt stating “Directory structure valid”. File Ingestion process initiated on Root directory.	
Concluding Remarks: Tests provided the correct response prompts.				
Testing Team: Keikaku		Date Completed: 04/29/2020		

Test No.: ING 3			Current Status: Passed	
Test title: Initiate cleansing action on files				
Testing approach: This test will be conducted on the cleansing operation. One .log input file will be cleansed.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Perform ING 1 and ING 2.	Create Event Configuration. Create Directory Configuration.	Event Configuration has been created. Directory Configuration has been created.	Initial condition.
2	Add the "test_cleanse.log" from the TestData directory to the Red Team directory "PickData->Root->RedTeam".	Add a test file to cleanse.	"test_cleanse.log" is in the Red Team directory	The "test_cleanse.log" file has empty lines and non-ascii characters planted.
3	Click "Ingest" button on the Log File Configuration tab.	Start ingestion operation. Check to see if cleansing status is true.	"test_cleanse.log" appears in "PickData->Copies" directory. Log file details (File name, source, cleansing, validation, ingestion, and acknowledged flags) populated on Log File table and green check mark under "Cleansing Status" field.	"test_cleanse.log" is a copy. Results might take several seconds before appearing. A green check mark is considered cleansed. A red X is considered not cleansed.
4	Open "test_cleanse.log" file in "PickData->Copies" directory.	Check to see if file has been cleansed.	"test_cleanse.log" was copied and the copy was stripped of empty lines/rows, and invalid binary characters.	"test_cleanse.log" copy file was stripped of empty lines and non-ascii characters.
Concluding Remarks: Other file formats (.csv) need to be tested.				
Testing Team: Keikaku		Date Completed: 04/30/2020		

Test No.: ING 4		Current Status: Passed		
Test title: Initiate validation action on files				
Testing approach: This test will be conducted on the validation operation. Two .log input files are selected: one valid file and one invalid file.				
STEP	OPERATOR ACTION	PURPOSE	EXEPCTED RESULTS	COMMENTS
1	Perform ING 1 and ING 2.	Create Event Configuration. Create Directory Configuration.	Event Configuration has been created. Directory Configuration has been created.	Initial condition.
2	Add the "test_invalid.log" and "test_valid.log" files from the TestData directory to the Red Team directory. "PickData->Root->RedTeam".	Test with valid and invalid log files.	"test_invalid.log" and "test_valid.log" is in the Red Team directory.	The valid file has timestamps that are in range based on the event start and end times and contains no missing timestamps. The invalid file has missing timestamps or out of range timestamps.
3	Click "Validate" button on the Log File Configuration tab.	Start validation operation.	Log file details (File name, source, cleansing, validation, ingestion, and acknowledged flags) populated on Log File table and green check mark for "test_valid.log" and red X for "test_invalid.log" under "Validation Status" field.	A green check mark is considered valid. A Red X is considered invalid.
4	Click on the "test_invalid.log" entry on the Log File table.	Display the Enforcement Action Report.	The Enforcement Action Report is populated with errors found on each line.	Errors include timestamp missing or timestamp out of range.
Concluding Remarks: Other file formats (.csv) should to be tested.				
Testing Team: Keikaku		Date Completed: 04/30/2020		

Test No.: ING 5			Current Status: Passed	
Test title: Initiate ingestion action on files				
Testing approach: This test will be conducted on the ingestion operation. One valid .log input file is ingested.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Perform ING 4 (Steps 1 – 2)	Create Event Configuration. Create Directory Configuration. "test_invalid.log" and "test_valid.log" files in RedTeam directory.	Event Configuration has been created. Directory Configuration has been created. "test_invalid.log" and "test_valid.log" files in RedTeam directory.	Initial condition.
2	Click "Ingest" button on the Log file configuration tab.	Start ingestion operation.	Log file details (File name, source, cleansing, validation, ingestion, and acknowledged flags) populated on log file table and green check marks under "Cleansing Status", "Validation Status", and "Ingested Status" field for "test_valid.log".	A green check mark is considered passed. A red X is not considered failed.
3	Click "Log Entry Configuration" tab.	Display the Log Entry table.	Log entry details (Line, source, timestamp, event, vector) fields populated on Log Entry table.	At this point the log entries should be visible in Splunk.
Concluding Remarks: Other file formats (.csv) need to be tested.				
Testing Team: Keikaku		Date Completed: 04/30/2020		

Test No.: ING 6			Current Status: Passed	
Test title: Initiate force ingestion action on files				
Testing approach: This test will be conducted on the force ingestion operation; two input files are selected one .csv file and one .log file.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Perform ING 4 (Steps 1 – 2)	Create Event Configuration. Create Directory Configuration. "test_invalid.log" and "test_valid.log" files in RedTeam directory.	Event Configuration has been created. Directory Configuration has been created. "test_invalid.log" and "test_valid.log" files in RedTeam directory.	Initial condition.
3	Click on the "test_invalid.log" entry on the Log File table. Click "Acknowledge" button on the Log file configuration tab.	Start forced ingestion operation.	Log file details (File name, source, cleansing, validation, ingestion, and acknowledged flags) populated on log file table and green check marks under "Cleansing Status" and "Ingested Status" field, and a red X for "Validation Status" for "test_invalid.log".	A green check mark is considered passed. A red X is not considered failed.
4	Click "Log Entry Configuration" tab.	Display the Log Entry table.	Log entry details (Line, source, timestamp, event, vector) fields populated on log file table.	At this point the log entries should be visible in Splunk.
Concluding Remarks: Other file formats (.csv) need to be tested.				
Testing Team: Keikaku		Date Completed: 04/30/2020		

Table Modifications

Objective: To confirm vector, node, and relationship entries are being added, deleted or modified inside their respective tables.
Notes: The vectors node, and relationship entries will have specific id's and will be generated once added.

Test No.: TBM 1			Current Status: Passed	
Test title: Add and delete entries to Vector table.				
Testing approach: This test will be conducted in the vector table using an add vector button and a delete vector button.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Click "Vector" button. Click "Add Vector" button	Creates a new vector.	New vector with unique id is added to table and vector drop down menu.	
2	Click "Add Vector" button again.	Creates another new vector.	New vector with unique id is added to table and vector drop down menu.	
3	Click the second vector row.	Select a vector from the table.	The vector is darkened to notify user has selected it.	
4	Click "Delete Vector" button.	Remove a vector from the table.	The vector is deleted from the table.	
5	Click on the remaining vector row.	Select a vector from the table.	The vector is darkened to notify user has selected it.	
6	Click "Delete Vector" button.	Remove a vector from the table.	The vector is deleted from the table.	Table is left empty.
7	Click "Delete Vector" button.	Remove a vector from the table.	Nothing happens because table is empty.	
Concluding Remarks: All of expected results met. Added and removed vectors successfully.				
Testing Team: Keikaku		Date Completed: 05/02/20 A.V.		

Test No.: TBM 2			Current Status: Passed	
Test title: Edit entries on Vector table.				
Testing approach: This test will be conducted on the vector table, field inputs are selected and then input text data is updated.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Click “Vector” button. Click “Add Vector” button	Create a new vector.	New vector with unique id is added to table and vector drop down menu.	
2	Click the only vector row.	Select a vector from the table.	The vector is darkened to notify user has selected it.	
2	Double click on the vector name. Erase any text and type “Vector A”.	Set a name for the vector.	The vector cell allows user to input information.	
3	Press “Enter” key.	Saves the name of the vector.	The vector cell exits editing mode. The vector cell displays “Vector A”.	
4	Double click on vector description. Erase any text and type “Vector Description”.	Set a description for the vector.	The vector cell allows user to input information.	
5	Press “Enter” key.	Saves the description of the vector.	The vector cell exits editing mode. The vector cell displays “Vector Description”.	
Concluding Remarks: All of the expected results obtained. Successfully edited the vector attributes.				
Testing Team: Keikaku		Date Completed: 05/02/20 A.V.		

Test No.: TBM 3		Current Status: Passed		
Test title: Node table entry addition and removal.				
Testing approach: This intends to test the successful addition and removal of entries to the Node table, with each addition creating a unique Node ID, and each removal disposing of it.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Click “Vector” button. Click “Add Vector” button.	Create a new vector.	New vector with unique id is added to table and vector drop down menu.	A vector is needed to perform this test.
1	Move to “Node Configuration tab” Click on “Add Node” button.	A new Node entry is added to the Node table.	A new Node entry is displayed on the table, along with the generated unique Node ID.	
2	Click on the “Add Node” button 3 times.	3 new Node entries are added to the Node table.	3 new entries will be appended to the existing entry in the Node table, along with their unique Node IDs.	
3	Click on the last Node entry on the Node Table.	Highlight the last Node entry on the Node table along with all its properties.	The entry should appear visibly highlighted.	
4	Click on the “Delete Node” button.	Remove the selected Node entry from the Node table.	The Node entry is removed from the Node table.	
5	Holding shift on the keyboard, click all the remaining Node entries on the Node table.	Highlight all of the Node entries on the Node table along with all its properties.	The selected entries should appear visibly highlighted.	
6	Click the “Delete Node” button.	Remove the selected Node entries from the Node table.	The Node entries are removed from the Node table.	The table should be empty.
Concluding Remarks: All of the expected results were obtained. Node entries and IDs were generated successfully.				
Testing Team: Keikaku		Date Completed: 05/02/20 A.V.		

Test No.: TBM 4		Current Status: Passed		
Test title: Editing of entry in Node table.				
Testing approach: This is intended to test the editing of the various cells of a given entry in a Node table, which include Node Name, Timestamp, Description, Log Entry Reference, Log Creator, Event Type, Icon Type, Source, and Node Visibility.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Click “Vector” button. Click “Add Vector” button.	Create a new vector.	New vector with unique id is added to table and vector drop down menu.	A vector is needed to perform this test.
2	Move to “Node Configuration” tab. Click “Add Node” button.	A new Node entry is added to the Node table.	A new Node entry is displayed on the table, along with the generated unique Node ID.	
3	Double-click on the cell in the column labeled “Node Name.”	This is intended to test the ability to select the “Node Name” as an editable field.	The Node entry’s cell under the column “Node Name” displays a cursor and is ready to take user input.	
4	Type “Node A” for the Node entry.	This is intended to demonstrate the ability to edit the field “Node Name.”	The cell under the column “Node Name” displays “Node A”.	
5	Click outside of “Node Name” cell or press Tab.	This is intended to finalize the changes to the “Node Name” cell provided by the user.	The cell under the column labeled “Node Name” displays “Node A”.	
6	Double-click on the cell in the column labeled “Node Timestamp.”	This is intended to test the ability to select the “Node Timestamp” as an editable field.	The Node entry’s cell under the column “Node Timestamp” displays a cursor and is ready to take user input.	
7	Type “12:00 PM” for the Node entry.	This is intended to demonstrate the ability to edit the field “Node Timestamp.”	The cell under the column “Node Timestamp” displays “12:00 PM”.	
8	Click outside of “Node Timestamp” cell or press Tab.	This is intended to finalize the changes to the “Node Timestamp” cell provided by the user.	The cell under the column labeled “Node Timestamp” displays “12:00 PM”.	
9	Double-click on the cell in the column labeled “Node Description.”	This is intended to test the ability to select the “Node Description” as an editable field.	The Node entry’s cell under the column “Node Description” displays a cursor and is ready to take user input.	
10	Type “Description” for the Node entry.	This is intended to demonstrate the ability to edit the field “Node Description.”	The cell under the column “Node Description” displays “Description”.	
11	Click outside of “Node Description” cell or press Tab.	This is intended to finalize the changes to the “Node Description” cell provided by the user.	The cell under the column labeled “Node Description” displays “Description”.	

12	Repeat steps 9-11 for “Log Entry Reference, Log Creator, Event Type, Icon Type, and Source”, typing “This is a test” for each input.	This is intended to test the edit ability of the cells under the columns labeled “Log Entry Reference, Log Creator, Event Type, Icon Type and Source” on the Node table.	The cells under the columns labeled “Log Entry Reference, Log Creator, Event Type, Icon Type, and Source” should display “This is a test”.	
13	Click on the “Node Visibility” check box.	This is intended to test the toggling on/off of the Node’s visibility in the graph from the Node Table.	The check box next to the selected entry in the Node table should appear checked.	
Concluding Remarks: All of the expected results obtained. Successfully edited all editable fields on the Node entry.				
Testing Team: Keikaku		Date Completed: 05/02/20 A.V.		

Test No.: TBM 5			Current Status: Passed	
Test title: Add and remove entries to the relationship table.				
Testing approach: This test will be conducted on the relationship configuration dialogue, using add relationship button and delete relationship button.				
STEP	OPERATOR ACTION	PURPOSE	EXEPCTED RESULTS	COMMENTS
1	Click “Vector” button. Click “Add Vector” button.	Create a new vector.	New vector with unique id is added to table and vector drop down menu.	A vector is needed to perform this test.
2	Click “Relationships” button on the Node Configuration tab. Click “Add Relationship” button.	Display the Relationship table. Add a new relationship to the Relationship table.	Relationship configuration is displayed. Relationship with a unique id is added to the table.	Parent and child columns remain empty.
3	Click “Add Relationship” button again.	Adds another relationship to the Relationship table.	A new relationship with a unique id is added to table.	
4	Click the second relationship entry on the Relationship table.	Select a relationship.	The relationship is darkened to notify the user the relationship is selected.	
5	Click “Delete Relationship” button.	Removes a relationship from table.	The relationship is removed from the table.	
6	Click the remaining relationship entry on the Relationship table.	Select a relationship.	The relationship is darkened to notify the user the relationship is selected.	
7	Click “Delete Relationship” button.	Removes a relationship from table.	The relationship is deleted from the table.	Table is left empty.
8	Click “Delete Relationship” button.	This is intended to remove relationship from table.	Nothing happens because table is empty.	
Concluding Remarks: All of the expected results were obtained. Successfully added relationship entries with unique IDs.				
Testing Team: Keikaku		Date Completed: 05/02/20 A.V.		

Test No.: TBM 6			Current Status: Passed	
Test title: Edit entries on the relationship tables.				
Testing approach: This test will be conducted on the relationship configuration dialogue. Field inputs are selected and the input text data is saved.				
STEP	OPERATOR ACTION	PURPOSE	EXEPCTED RESULTS	COMMENTS
1	Click “Vector” button. Click “Add Vector” button.	Create a new vector.	New vector with unique id is added to table and vector drop down menu.	A vector is needed to perform this test.
2	Click “Relationships” button on the Node Configuration tab. Click “Add Relationship” button.	Display the Relationship table. Add a new relationship to the Relationship table.	Relationship configuration is displayed. Relationship with a unique id is added to the table.	Parent and child columns remain empty.
3	Double click on the “Parent” cell.	This is intended to enable editing.	The relationship parent cell allows user to input information	
4	Type “ID A”. Press “Enter” key.	This is intended to finish editing.	The relationship “Parent” cell displays “ID A”.	
5	Double click “Child” cell.	This is intended to enable editing.	The relationship child cell allows user to input information	
6	Type “ID B”. Press “Enter” key.	This in intended to finish editing.	The relationship “Child” cell displays “ID B”.	This will create a relationship between current the node parent and node child.
7	Double click “Label” cell.	This is intended to enable editing.	The relationship label cell allows user to input information	
8	Type “Test Label”. Press “Enter” key.	This is intended to finish editing.	The relationship “Label” cell displays “Test Label”.	
Concluding Remarks: All of the expected results were obtained. Successfully edited editable fields in the relationship table.				
Testing Team: Keikaku		Date Completed: 05/02/20 A.V.		

Graphing

Objective: To establish proper functionality of the graphing process.
Notes: N/A

Test No.: GPH 1			Current Status: Passed	
Test title: Add items to graph editor view				
Testing approach: This test will be conducted on the GraphEditor control to test its ability to add items to the graph editor view.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Click “Vector” button. Click “Add Vector” button.	Create a new vector.	New vector with unique id is added to table and vector drop down menu.	A vector is needed to perform this test.
2	Click “Add Node” button in the Node Configuration Window.	Check if new node appears in the graph editor view.	The graph editor view displays new node under the current selected vector in the drop-down menu.	
3	Click “Add Node” button again for the same node in the Node Configuration Window.	Make sure that duplicate nodes do not exist.	The graph editor view remains unchanged. A message box pops up indicating that this node already exists.	
Concluding Remarks: All of the expected results were obtained. Node item was successfully added.				
Testing Team: Keikaku		Date Completed: 05/03/20 A.V.		

Test No.: GPH 2			Current Status: Passed	
Test title: Add items to graph editor view				
Testing approach: This test will be conducted on the GraphEditor control to test its ability to add items to the graph editor view.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Perform GPH 1 Step 1 and Step 2 twice with two unique nodes.	Create a vector and two unique nodes.	The graph editor view displays two new nodes under the current selected vector in the drop-down menu	Two nodes are needed to perform this test.
2	Selected both of the added nodes by clicking on the list numbers consecutively or by pressing Shift+ mouse click over selected nodes.	This selects the parent and child nodes that compose a relationship.	Both Node item entries have been highlighted to show they are currently selected.	At least two nodes must be selected for Step 3 to be performed.
3	Click “Relationship” button in the Node Configuration Window, click “Add Relationship” button.	A relationship should be created between the two, previously selected Node items.	The graph editor view displays new relationship line under the current selected vector in the drop-down menu.	
Concluding Remarks: All of the expected results were obtained. A new relationship item added successfully to the graph scene.				
Testing Team: Keikaku		Date Completed: 05/03/20 A.V		

Test No.: GPH 3		Current Status: Passed		
Test title: Remove Node items from graph editor view				
Testing approach: This test will be conducted on the GraphEditor control to test its ability to remove items from the graph editor view				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Perform GPH 1 (Steps 1 - 2)	Create a vector and unique node.	The graph editor view displays a new node under the current selected vector in the drop-down menu	A node is needed to perform this test.
2	Select Node and click “Remove Node” button in the Node Configuration Window.	Check if the specified node is removed in the graph editor view.	The graph editor view no longer displays the specified node under the current selected vector in the drop-down menu.	
Concluding Remarks: All of the expected results were obtained. A Node item was successfully removed from the graph scene.				
Testing Team: Keikaku		Date Completed: 05/03/20 A.V.		

Test No.: GPH 4		Current Status: Failed		
Test title: Remove Relationship items from graph editor view				
Testing approach: This test will be conducted on the GraphEditor control to test its ability to remove Relationship items from the graph editor view				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Perform GPH 2	Create a vector, two unique nodes, and a relationship between them.	The graph editor view displays new relationship line with its label under the current selected vector in the drop-down menu	A relationship is needed to perform this test.
2	Click “Relationship” button in the Node Configuration Window, select relationship, and click “Remove Relationship” button.	Check if new relationship line is removed in the graph editor view.	The graph editor view no longer displays the specified relationship line under the current selected vector in the drop-down menu.	
Concluding Remarks: Expected results were not obtained. Relationship item continues to be part of editor scene after clicking remove relationship.				
Testing Team: Keikaku		Date Completed: 05/03/20 A.V.		

Test No.: GPH 5		Current Status: Passed		
Test title: Dynamic movement of relationship lines and nodes				
Testing approach: This test will check the ability to update the position of relationship lines in relation to their parent and child nodes				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Perform GPH 2	Create a vector, two unique nodes, and a relationship between them.	The graph editor view displays new relationship line with its label under the current selected vector in the drop-down menu.	A node with a relationship is needed to perform this test.
1	Click and drag any node that is a parent or child of a relationship line in the graph editor view.	Check if the lines for the relationship lines move dynamically with their parent and child nodes.	The line coordinates for the relationship line change to match to new center point of the moved node.	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed: 05/05/2020		

Test No.: GPH 6		Current Status: Failed		
Test title: Toggle Vector visibility				
Testing approach: This test will check the ability to toggle visibility of the elements within a VectorItemGroup.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Perform GPH 2 twice using different vectors.	Create two vectors each with two unique nodes, and a relationship between them.	The graph editor view displays new relationship line with its label under the current selected vector in the drop-down menu	Two vectors with their node contents and relationships are needed to perform this test.
2	Select a Vector from the Vector Selection drop down menu.	Check to make sure that only that vector is visible in the graph editor view.	Only the nodes and their respective relationship lines are visible in the graph editor view.	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed: 05/05/2020		

Test No.: GPH 7		Current Status: Failed		
Test title: Synchronization of table views and graph editor view.				
Testing approach: This test will check that the changes made on table views reflects on graph editor view.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Perform GPH 2	Create a vector, two unique nodes, and a relationship between them.	The graph editor view displays new relationship line with its label under the current selected vector in the drop-down menu.	A node with a relationship is needed to perform this test.
2	Change node name on Node Configuration Table.	Check to make sure that the Node name in the Node Configuration is the same as its respective node in the graph editor view.	Displayed Node name on Node Configuration Table is the same displayed in the graph editor view.	
3	Change node description on Node Configuration Table.	Check to make sure that the Node description in the Node Configuration is the same as its respective node in the graph editor view.	Displayed Node name on Node Configuration Table is the same displayed in the graph editor view.	
4	Change node description on Node Configuration Table.	Check to make sure that the Node description in the Node Configuration is the same as its respective node in the graph editor view.	Displayed Node description on Node Configuration Table is the same displayed in the graph editor view.	
5	Change node log creator on Node Configuration Table.	Check to make sure that the Node log creator in the Node Configuration is the same as its respective node in the graph editor view.	Displayed Node log creator on Node Configuration Table is the same displayed in the graph editor view.	
6	Change node event type on Node Configuration Table.	Check to make sure that the Node event type in the Node Configuration is the same as its respective node in the graph editor view.	Displayed Node event type on Node Configuration Table is the same displayed in the graph editor view.	
7	Change node icon type on Node Configuration Table.	Check to make sure that the Node icon type in the Node Configuration is the same as its respective node in the graph editor view.	Displayed Node icon type on Node Configuration Table is the same displayed in the graph editor view.	
8	Change relationship label on Relationship Configuration Table.	Check to make sure that the Relationship label in the Node Configuration is the same as its respective relationship line in the graph editor view.	Displayed Relationship label on Node Configuration Table is the same displayed in the relationship line on the graph editor view.	
9	Change relationship parent on Relationship Configuration Table.	Check to make sure that the Relationship parent in the Node Configuration is the same as its respective relationship line in the graph editor view.	The coordinates to the relationship line change to the new parent node in the graph editor view.	

10	Change relationship child on Relationship Configuration Table.	Check to make sure that the Relationship parent in the Node Configuration is the same as its respective relationship line in the graph editor view.	The coordinates to the relationship line change to the new child node in the graph editor view.	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed: 05/05/2020		

Data Persistence

Objective: To ensure data is persistent throughout the application's lifetime.
Notes: The storage is on a file system basis, where data is being serialized and stored to a file then retrieved once needed.

Test No.: DP1		Current Status: Passed		
Test title: Test the event configuration's data is being saved.				
Testing approach: The event configuration window is to be populated with data then once saved application is to be closed, then reopened.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Perform ING 1.	Create Event Configuration.	Event Configuration has been created.	Initial condition.
2	Close all windows.	Exit application.	Windows are closed.	
3	Start system. Click "File->Event".	Display the Event Configuration dialog.	Event Configuration dialog opens and displays entered information.	
Concluding Remarks: None.				
Testing Team: Keikaku		Date Completed: 4/28/2020		

Test No.: DP2		Current Status: Passed		
Test title: Test the directory configuration's data is being saved.				
Testing approach: The directory configuration window is to be populated with data then once saved application is to be closed, then reopened.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Perform ING 1 and ING 2.	Create Event Configuration. Create Directory Configuration.	Event Configuration has been created. Directory Configuration has been created.	Initial condition.
2	Close all windows.	Exit application.	Windows are closed.	
3	Click on the "Directory" button on the Log File configuration tab.	Display the Directory Configuration dialog.	Directory Configuration window is displayed and displays entered information.	
Concluding Remarks: None.				
Testing Team: Keikaku		Date Completed: 4/28/2020		

Test No.: DP3		Current Status: Passed		

Test title: Test the Log File tabular data is being saved.				
Testing approach: The log file configuration table is to be populated with data then once saved application is to be closed, then reopened.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Perform ING 3.	Create Event Configuration. Create Directory Configuration. Generate Log File tabular data.	Log file details (File name, source, cleansing, validation, ingestion, and acknowledged flags) populated on Log File table and green check mark under “Cleansing Status” field.	Initial condition.
2	Close all windows.	Exit application.	Windows are closed.	
3	Launch application again. Click “Log File Configuration” tab.	Display Log File table.	Log File table has been repopulated with saved data.	
Concluding Remarks: None.				
Testing Team: Keikaku		Date Completed: 05/05/2020		

Test No.: DP4		Current Status: Passed		
Test title: Test the Log entry tabular data is being saved.				
Testing approach: The log entry configuration table is to be populated with data then once saved application is to be closed, then reopened.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Perform ING 5.	Create Event Configuration. Create Directory Configuration. Generate Log File tabular data. Generate Log Entry tabular data.	Log entry details (Line, source, timestamp, event, vector) fields populated on Log Entry table.	Initial condition.
2	Click the “Commit” button on the main window. Click the “commit” button in the commit prompt.	This saves the desired data.	The current system configuration has been saved into the system cache folder as .pk files.	Failure to do this step will not persist data.
3	Close all windows.	Exit application.	Windows are closed.	
4	Launch application again. Click “Log Entry Configuration” tab.	Display Log Entry table.	Log Entry table has been repopulated with saved data.	
Concluding Remarks: None.				
Testing Team: Keikaku		Date Completed: 4/29/2020		

Test No.: DP5		Current Status: Passed		
Test title: Test the vector tabular data is being saved.				
Testing approach: The vector configuration table is to be populated with data then once saved application is to be closed, then reopened.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Click the “Vector” button.	Display Vector Configuration table.	Vector Configuration table displayed.	
3	Add 20 vectors to Vector table. Click “Add Vector” button 20 times.	Add vectors to Vector table.	Vector table has 20 entries.	
4	Modify random vectors with names and descriptions. (refer to TBM 2)	Populate table with random data.	Vector fields updated.	
5	Click the “Commit” button on the main window. Click the “commit” button in the commit prompt.	This saves the desired data.	The current system configuration has been saved into the system cache folder as .pk files.	Failure to do this step will not persist data.
6	Close all windows.	Exit application.	Windows are closed.	
7	Launch application again. Click the “Vector” button.	Display Vector table.	Vector table is repopulated with saved data.	
Concluding Remarks: None.				
Testing Team: Keikaku		Date Completed: 04/29/20		

Test No.: DP6		Current Status: Passed		
Test title: Test the Node tabular data is being saved.				
Testing approach: The Node Configuration table is to be populated with data then once saved application is to be closed, then reopened.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Click “Vector” button. Click “Add Vector” button.	Create a new vector.	New vector with unique id is added to table and vector drop down menu.	A vector is needed to perform this test.
2	Click the “Node Configuration” tab.	Display Node table.	Node table displayed.	
3	Add 20 nodes to Node table. Click “add node” button 20 times.	Populate table with nodes.	Node has 20 entries.	
4	Modify random nodes with names, descriptions, timestamps etc. (refer to TBM 4)	Populate table with random data.	Node fields updated.	
5	Click the “Commit” button on the main window. Click the “commit” button in the commit prompt.	This saves the desired data.	The current system configuration has been saved into the system cache folder as .pk files.	Failure to do this step will not persist data.

6	Close all windows.	Exit application.	Windows are closed.	
7	Launch application again. Click the “Node Configuration” tab.	Display Node table.	Node table is repopulated with saved data.	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed: 05/05/2020		

Test No.: DP7		Current Status: Passed		
Test title: Test the Relationship tabular data is being saved.				
Testing approach: The Relationship Configuration table is to be populated with data then once saved application is to be closed, then reopened.				
STEP	OPERATOR ACTION	PURPOSE	EXPECTED RESULTS	COMMENTS
1	Click the “Node configuration” tab.	Display the Node Configuration panel.	Node table is displayed.	
2	Click the “Relationship” button.	View relationship table.	Relationship table is displayed.	
3	Add 20 relationships to relationship table. Click “Add Relationship” button 20 times.	Populate table with relationships.	Relationship has 20 entries.	
4	Modify random relationships with parents, children, and labels. (refer to TBM 6)	Populate table with random data.	Relationship fields updated.	
5	Click the “Commit” button on the main window. Click the “commit” button in the commit prompt.	This saves the desired data.	The current system configuration has been saved into the system cache folder as .pk files.	Failure to do this step will not persist data.
6	Close all windows.	Exit application.	Windows are closed.	
7	Launch application again. Click the “Node configuration” tab. Click the “Relationship” button.	View relationship table.	Relationship table is repopulated with saved data.	
Concluding Remarks:				
Testing Team: Keikaku		Date Completed: 05/05/2020		

5. Test Schedule

Task and date	People	Description
04/30/19	David Rayner	Log ingestion test suite (test cases ING 1-5)
04/30/19	Valentin Becerra	Graph test suite (GPH 1-4)
05/01/19	Valentin Becerra	Graph test suite (GPH 5-7)
05/01/19	Jorge Garcia	Table Modifications (TBM 1-4)
05/01/19	Angel Villapando	Table Modifications (TBM 4-8)
05/02/19	Anthony DesArmier	Data Persistence (DP 3-4)
05/02/19	Mario Delgado	Data Persistence (DP 5-7)

6. Other Sections

Tests are to be portioned off to each member of the development team based on the four test suites in section 3. The suites capture the main components of the system. Each suite will have a lead team member, this is to ensure that the tests are being performed and yield appropriate results.

No training is required; however, the development team is to have an understanding of the system components and the type of testing that needs to be performed.

Python 3.4+, preferably 3.8 is required. This application supports any environment that has Python 3 support.

The following is a list of the current required python package installations:

- PyQt5==5.14.2
- PyQt5-sip==12.7.2
- python-dateutil==2.8.1
- python-dotenv==0.13.0
- splunk-sdk==1.6.12
- virtualenv==20.0.20
- virtualenv-clone==0.5.4

Splunk Enterprise is required in order to run the Splunk server. This requires a Splunk Enterprise download on the respective OS. The host's username, password port of the Splunk server, and index name to store the entries is required.

Storage is to be managed through serialization and saved on a file system. Therefore, currently there is no need for installation of a database.

No cost is to be associated with any of the software, since each of the libraries are open source.

7. Appendix

7.1. Ingestion

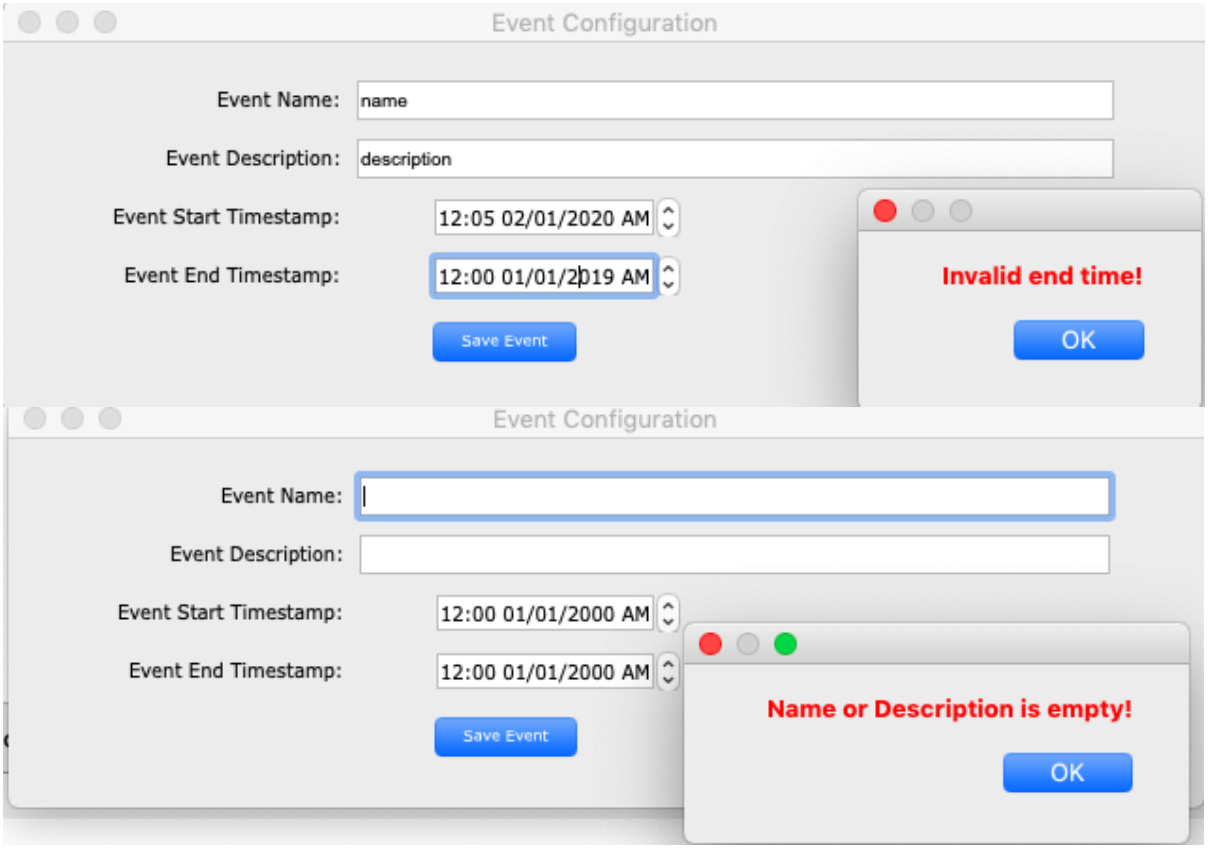

Figure ING 1

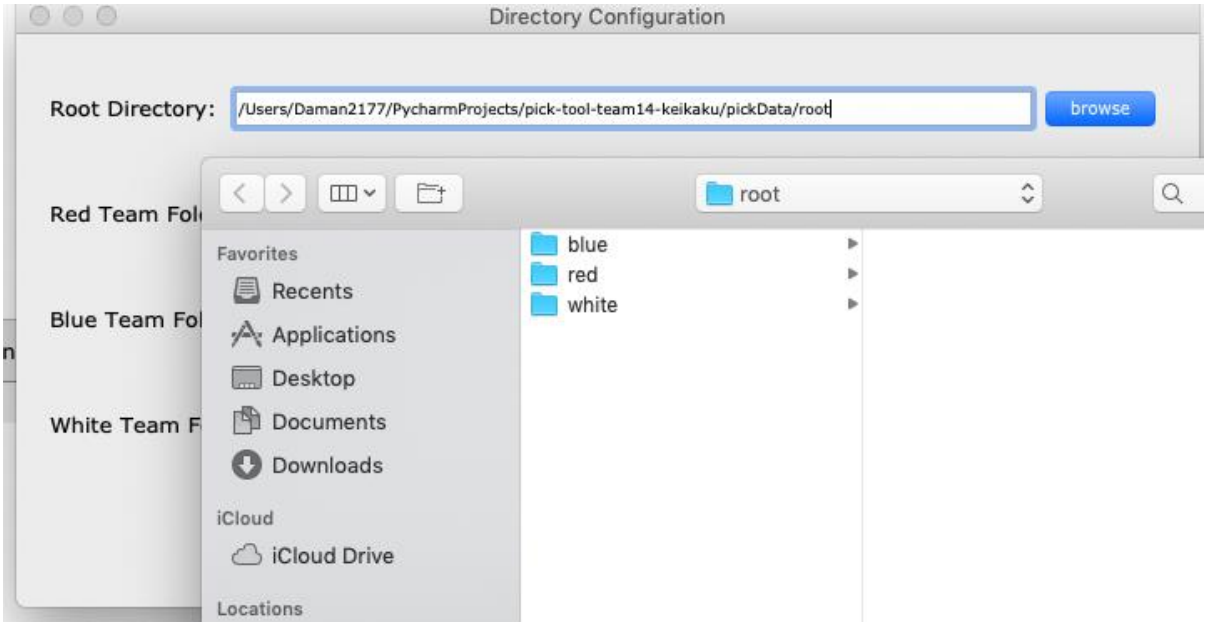

Figure ING 2.

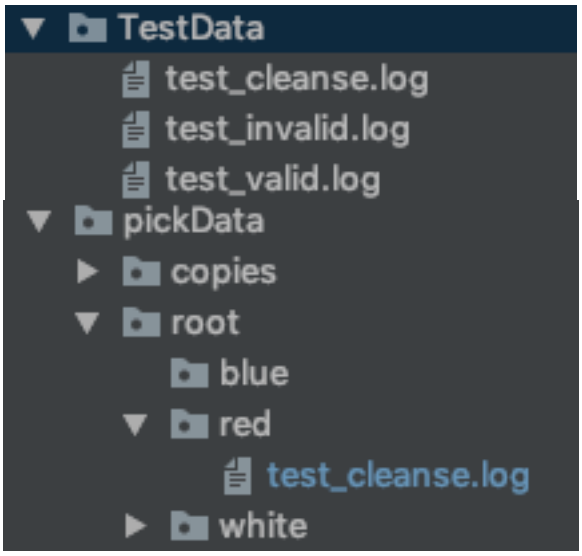

Figure ING 3

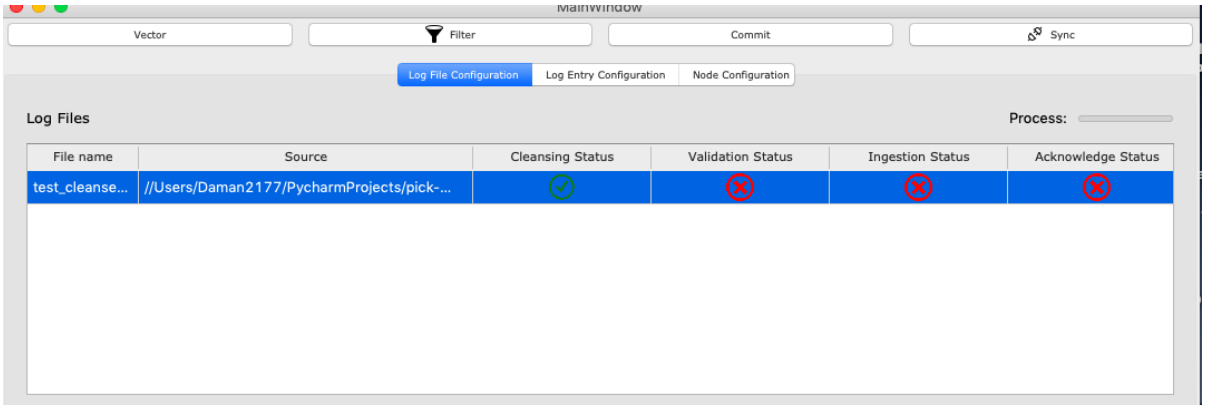

Figure ING 3.

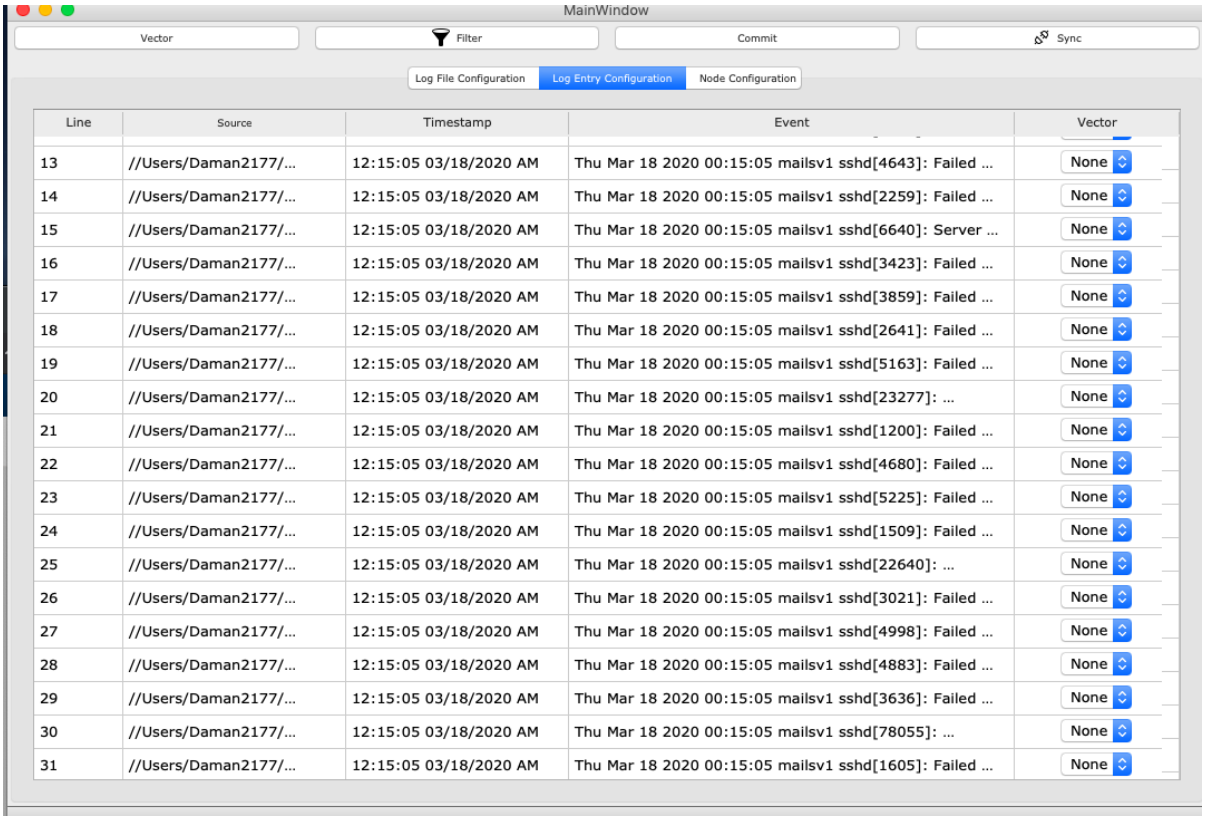

Figure ING 5

7.2. Table Modifications

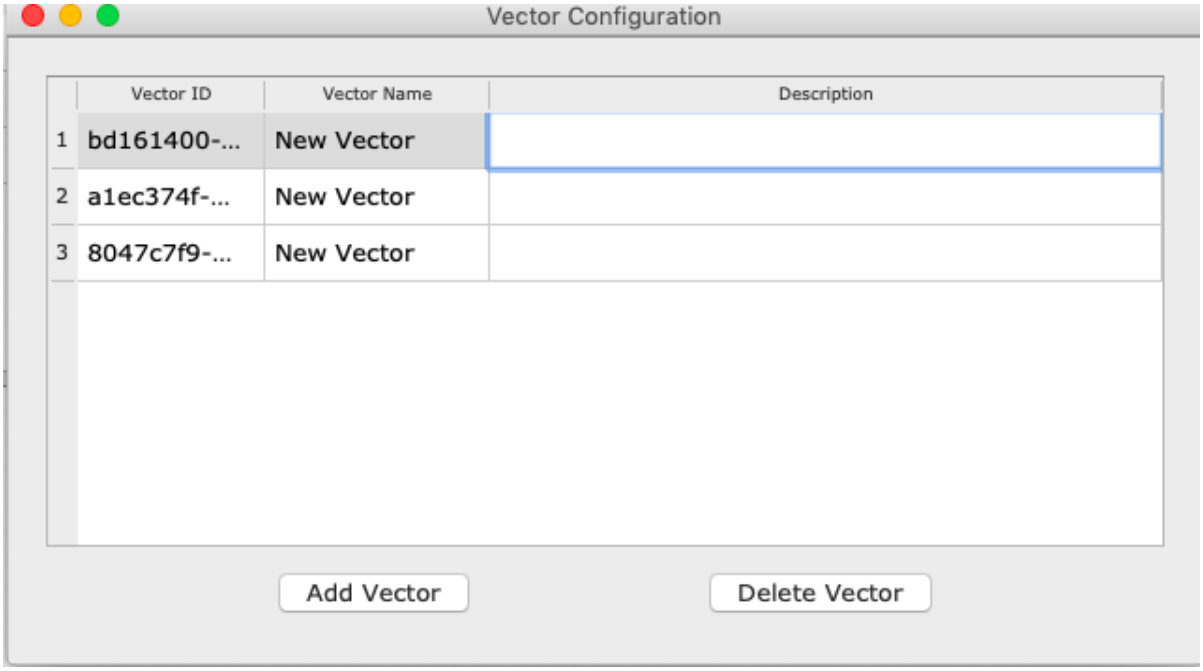

Figure TBM 3.

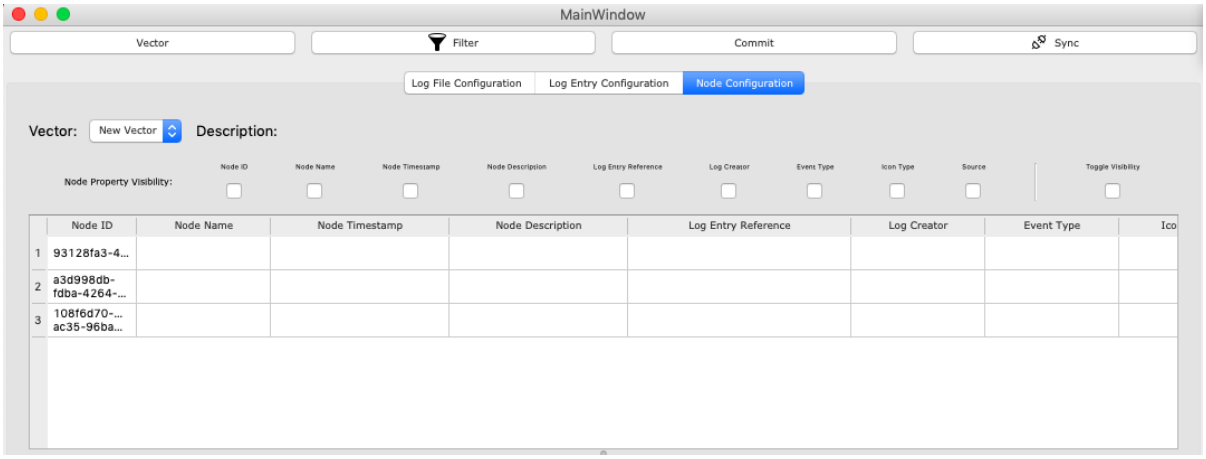

Figure TBM 4.

7.3. Graphing

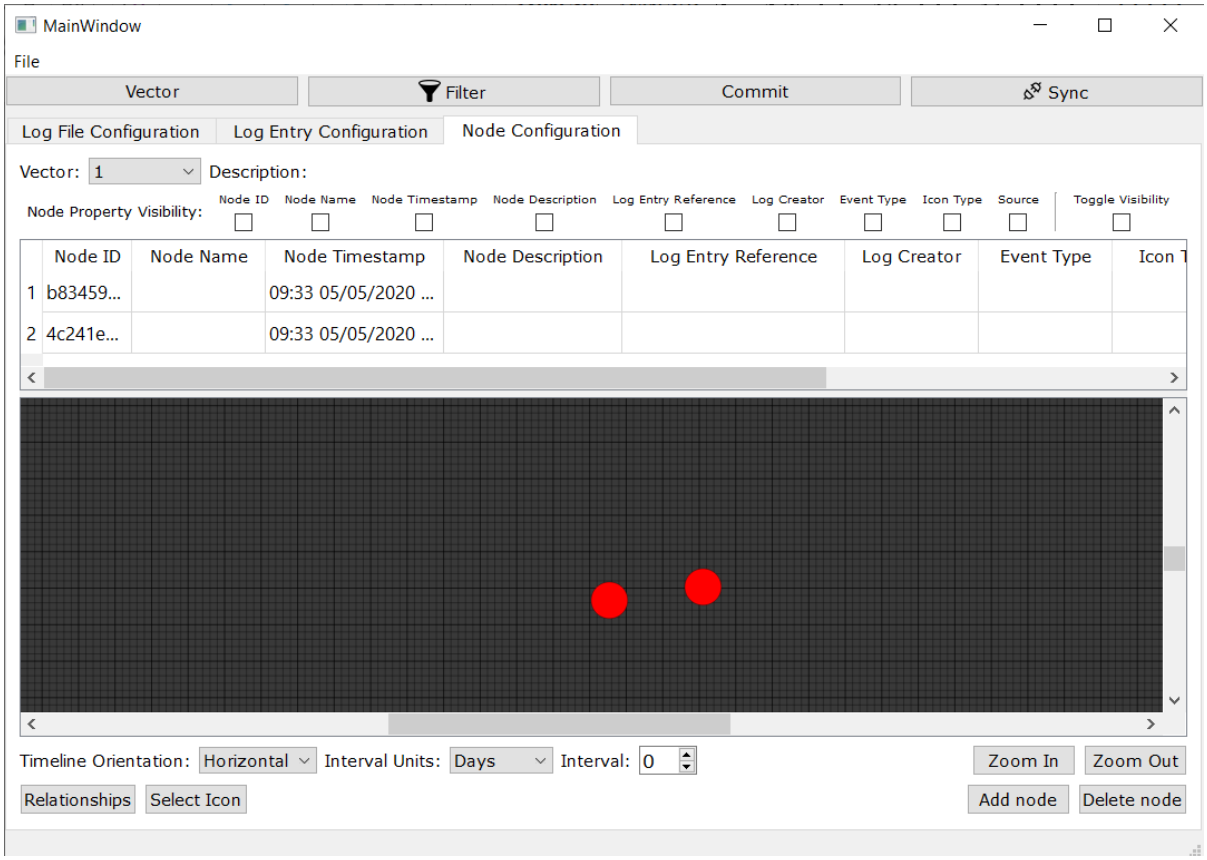

Figure GPH 2.

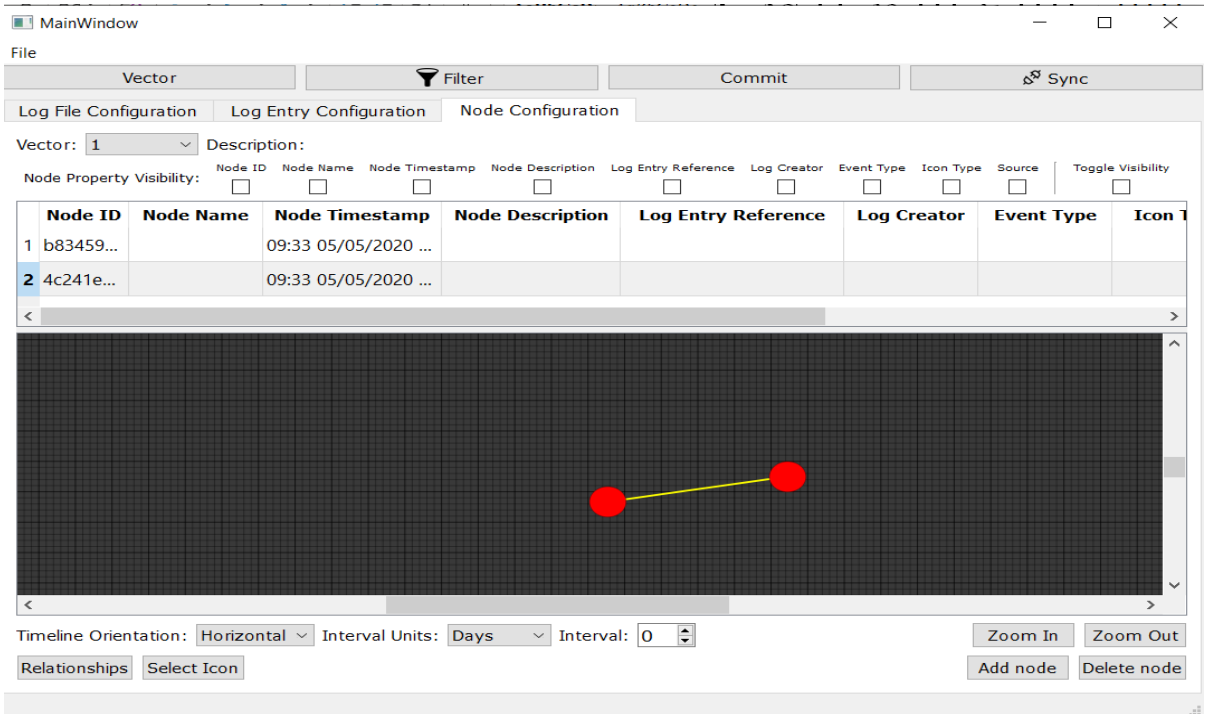

Figure GPH 5.

7.4. Data Persistence

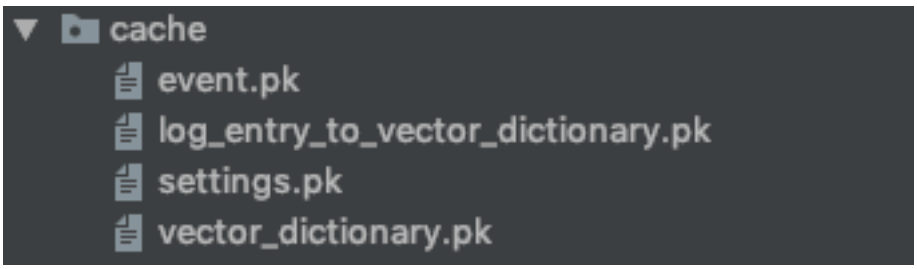

Figure Cached Data Persistence Files.

[END]