­Prevent, Mitigate, and Recover (PMR) Insight

Collective Knowledge System (PICK)

Test plan

Version <2.2>

05/09/2020

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

|  |  |
| --- | --- |
| Initial Release: | 1.0 |
| Current Release: |  |
| Indicator of Last Page in Document: | \* |
| Date of Last Review: | 4/28/2020 |
| Date of Next Review: | 4/18/2020 |
| Target Date for Next Update: | 4/22/2020 |

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. Gates

Dr. Salamah

Dr. Roach

Steven Roach

Jake Lasley

Customer:

Dr. Oscar Perez

Vincent Fonseca

Herandy Denisse Vazquez

Baltazar Santaella

Florencia Larsen

Erick De Nava

Software Team Members:

Elizabeth Barragan

Jose A. Leon Cordero

Yamel E. Hernandez

Manuel Delgado

Change Summary

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1.0 | 4/16/2020 | Manuel | Creation of Document & Document Control |
| 1.1 | 4/16/2020 | Elizabeth | Added a few database tests cases (Sec 3) and Test Suite DB3 (Sec 4) |
| 1.2 | 4/16/2020 | Yamel | Added section 1.5, added Ingestion test cases to section 3, & Ingestion Test Suit in Section 4 |
| 1.3 | 4/16/2020 | Antoine | Added tables 3 and 4 in section 3. |
| 1.4 | 4/16/2020 | Manuel | Added Vector Subsystem to section 3 as well as intro |
| 1.5 | 4/22/2020 | Elizabeth | Added Sec.1.1 ,1.2 |
| 1.6 | 4/27/2020 | Elizabeth | Updated Sec. 3: DB Test Suite,  Added DB Test 1, 2, Added DB Test 1,2 to the Appendix, Added Sec 1.6 |
| 1.7 | 4/27/2020 | Yamel | Added Sec 4 intro & deleted Sec 7 Other Sections |
| 1.8 | 4/28/2020 | Manuel | Added section 1.3 and test section Graph and Vector 4.9 to 4.19 |
| 1.9 | 4/28/2020 | Yamel | Added to the appendix for Ing1 & Ing 2 |
| 2.0 | 4/28/2020 | Antoine | Added 1.5, deleted table 3 in section 3, edited new table 3 in section 3, renamed section 4, added 4.12, and edited section 5 |
| 2.1 | 4/28/2020 | Elizabeth | Added DB Test 4-7, Added DB Test 4-5 in Appendix |
| 2.2 | 05/09/2020 | Elizabeth | Updated the DB Test suite, Updated DB tests, updated Appendix |

Note: The template presented in this document was taken from:

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

Note: The template presented in this document was taken from: Donaldson, S., and S. Siegel, *Successful Software Development*. Upper Saddle River, NJ: Prentice Hall, 2001, pp. 321-323 and modified by Humberto Mendoza and Steve Roach.

Supplementary information is from:

Pfleeger, S. *Software Engineering, Theory and Practice*. Upper Saddle River, NJ: Prentice Hall, 1998, p. 365.

Table of Contents

[Document Control ii](#_Toc39953724)

[Approval ii](#_Toc39953725)

[Document Change Control ii](#_Toc39953726)

[Distribution List ii](#_Toc39953727)

[Change Summary ii](#_Toc39953728)

[1. Introduction 1](#_Toc39953729)

[1.1. Purpose 1](#_Toc39953730)

[1.2. Scope 1](#_Toc39953731)

[1.3. System Overview 1](#_Toc39953732)

[1.4. Suspension and Exit Criteria 1](#_Toc39953733)

[1.5. Document Overview 1](#_Toc39953734)

[1.6. References 2](#_Toc39953735)

[2. Test Items and Features 3](#_Toc39953736)

[2.1. Components to be Tested 3](#_Toc39953737)

[2.2. Components not Implemented 3](#_Toc39953738)

[3. Testing Approach 4](#_Toc39953739)

[4. Test Breakdown 6](#_Toc39953740)

[4.1. Test DB 1 6](#_Toc39953741)

[4.2. Test DB 2 7](#_Toc39953742)

[4.3. Test DB 3 8](#_Toc39953743)

[4.4. Test Ing 1 9](#_Toc39953744)

[4.5. Test Ing 2 10](#_Toc39953745)

[4.6. Test Ing 3 2](#_Toc39953746)

[4.7. Test Ing 4 3](#_Toc39953747)

[4.8. Test EC 1 3](#_Toc39953748)

[4.9. Test GC1 5](#_Toc39953749)

[4.10. Test GC2 2](#_Toc39953750)

[4.11. Test GC3 2](#_Toc39953751)

[4.12. Test GC4 2](#_Toc39953752)

[4.13. Test GC5 2](#_Toc39953753)

[4.14. Test GC6 2](#_Toc39953754)

[4.15. Test GC7 2](#_Toc39953755)

[4.16. Test GC8 2](#_Toc39953756)

[4.17. Test VS1 2](#_Toc39953757)

[5. Test Schedule 3](#_Toc39953758)

[6. Appendix 5](#_Toc39953759)

# Introduction

## Purpose

The purpose of this document is to describe the kinds of tests that are ran in the system and provide guidance through specification/documentation for a tester to run tests accordingly for all parts of the system. In this test plan, system testing will be conducted along with some unit tests. Moreover, this document provides structure for organizing, scheduling, and managing of the tests ran in system. By providing analysis of the tests, this document allows the analysis of test coverage, identifies repetition of test, and improves test reproducibility.

## Scope

This test plan encompasses the release 1.0 including the following versions:

* V0.1: Executable GUI
* V0.2: Adding Ingestion Subsystem
* V0.3: Adding Vector/Graph Subsystem
* V0.4: Adding Database Subsystem

## System Overview

Testing will be done at system level, ensuring system integration was done correctly. Testing each feature as part of the whole system. Operations the system will contain are the ones that will be tested and are limited to the components implemented. Ingestion of log files into the Pick system, user graphical interface of the ability of the system, creation of a vector and its graph and finally permanent storage in the mongo database of any information inputted into the system

## Suspension and Exit Criteria

Testing for the PICK system is crucial in the eyes of the clients. Because it is a system that aims to optimize the goals of our clients, we define the following strict suspension and exit criteria for testing of the PICK system.

Suspension Criteria

Upon failure of 5 critical test cases, the execution of testing will be suspended to address the failures. This definition is motivated by the nature of these tests – most critical tests have implications in various parts of the system, therefore if one critical test fails, the failure of another critical test is almost ensured and an indication of a major issue of the PICK system.

Exit Criteria

Testing of the PICK system is to be finalized based on a compound test pass rate. Of this compound test pass rate, all critical tests must pass, while 80% of the non-critical must pass. These criteria are motivated by the implemented features of the PICK system – critical features of the PICK system impact the efficiency of the Lethality, Survivability, and HSI Directorate (LSH) to conduct their analysis, and the inability of a critical feature to pass a test will have direct implications on efficiency and the performance of other system features. Though non-critical features of the PICK system may also impact efficiency of LSH in minor ways, potential failures do not impact the overall goal of the system or other features.

## Document Overview

This document outlines how we will implement and test our system. Section 2 will describe the test items and features that will be tested of those items, which will include a table or a class diagram for visualization. Section 3 will discuss how we will approach testing the items and features from section 2 in tabular format. The tables will list the type of test, it’s objective, and the level of criticality of the test. Section 4 will document the tests in a table per test. The table will have the test number, its status, title, approach, who performed the test, and when the test was concluded. Section 5 will list and describe the test schedule in a table, information on it will include a date, who will be performing the test and a short description. Section 6, the Appendix, will depict and illustrate outputs for analysis.

## References

[1] SDD\_team5.docx

[2] Team5SCM.docx

[3] 22c\_Test\_Plan.ppt

[4] SRS\_v7.docx

# Test Items and Features

In this section, components, classes, functions and features to be tested are defined. Additionally, we define those which are not to be tested.

## Components to be Tested

The following is a list of the components and their functions and features that will be tested:

* Real-World Objects
  + Data transformation
  + Instantiation
  + JSON representation
* Signal Controller
  + Signals handling of all models
* Database
  + Splunk
* Ingestion Subsystem
  + Splunk
  + Cleansing
    - Extra spaces
    - Non-Ascii Characters
  + Enforcement Action Report & Error Handling
* Graphical User Interface
  + Window frame adjustment
  + Button events
  + Input processing
  + Model representation
  + Error handling
  + Default security settings
* Vector Subsystem
  + Graph
  + Node Table

## Components not Implemented

The following is a list of features that will not be tested:

* Version Control
* Undo/Redo action
* Audio transcriber
* OCR
* Networking

# Testing Approach

The section demonstrates in a table format the types of test that will be performed on the system. The table provides a description for types of tests such as but not limited to exercising system functions, approximating operational use, or stressing the system to find out limitations. Each component or subsystem to be tested has a description of the type of test, test case identifier for each objective and its importance level.

Table 1: Database Test Suite

|  |  |  |
| --- | --- | --- |
| **TEST SUITE DATABASE** | | |
| **Description of Test Suite** | **This test suite has test cases for the functionality of the Database subsystem** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| DB 1 | Test to save, load, update, of an Event Configuration object in the DB | Critical |
| DB 2 | Test to save, load, update, of a Vector object in the DB | Critical |
| DB 3 | Test to delete a Vector in the DB | Critical |

Table 2: Ingestion Test Suite

|  |  |  |
| --- | --- | --- |
| **TEST SUITE INGESTION** | | |
| **Description of Test Suite** | **This test suite will have appropriate test cases for the functionality of the ingestion subsystem and process** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| Ing 1 | Test for cleansing for additional spaces, lines, & binary data | Critical |
| Ing 2 | Test parsing in Splunk | Critical |
| Ing 3 | Test for error messages in enforcement action report & error class | Critical |
| Ing 4 | Test ingestion process to make sure log files get flagged for the state that they are in | Critical |

Table 3: Event Configuration Test Suite

|  |  |  |
| --- | --- | --- |
| **TEST SUITE EVENT CONFIGURATION** | | |
| **Description of Test Suite** | **This test suite defines test cases for probing data transformation and representation of the event configuration model** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| EC 1 | Test event configuration instantiation with user input data | Critical |

Table 4: Graph Component

|  |  |  |
| --- | --- | --- |
| **TEST SUITE GRAPH COMPONENT** | | |
| **Description of Test Suite** | **This test suite defines test cases for probing the creation and functionality of the Graph Component** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| GC1 | Test creation of a Graph | Critical |
| GC2 | Test creation of a Node | Critical |
| GC3 | Test creation of a Relationship | Critical |
| GC4 | Test functionality of movement for a Node | Normal |
| GC5 | Test functionality of movement for a Relationship | Normal |
| GC6 | Test deletion of a Node | Critical |
| GC7 | Test editing of a Node | Critical |
| GC8 | Test editing of a Relationship | Critical |

Table 5: Vector Subsystem Test Suite

|  |  |  |
| --- | --- | --- |
| **TEST SUITE VECTOR SUBSYSTEM** | | |
| **Description of Test Suite** | **This test suite defines test cases for probing the creation and functionality of the Vector Subsystem** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| VS1 | Test editing correlation between a Node and a Significant Log Entry | Critical |

# Test Breakdown

This section documents the tests performed on the system. Each subsection in here will encapsulate the type of test performed, who performed it, how they approached it, the method used, and nature of the test. These tests can be found from the test suites in section 3.

## Test DB 1

**Objective:** The objective of this test is to verify that the Event Configuration object is saved into the database, loaded in the system, and updated in the database when changes are done through the GUI.

**Notes:** The Event Configuration content stored in the DB will then be reflected in the GUI.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: DB1 | | | | Current Status: Passed | | |
| Test title: Test to save, load, update, of an Event Configuration object in the DB | | | | | | |
| Testing approach: This test will be conducted through the Graphical User Interface and will be reflected in GUI and the command prompt/shell. | | | | | | |
| STEP  1 | OPERATOR ACTION  Obtain the MongoDB executable file path and paste it in the \_\_init\_\_.py file located in the Source directory in the following line of code:  db = subprocess. Popen([**'mongodb executable path here’**], shell=**True**)  (See Appendix 1.1)  Run the \_\_init\_\_.py script | PURPOSE  Initial Condition | | | EXEPCTED RESULTS  The MongoDB server is running alongside of starting a project in the background. | COMMENTS  The connecting details are displayed in the command prompt/shell.  (See Appendix 1.2) |
| 2 | Click on the Even Configuration icon in the toolbar menu to pop out the Event Configuration Window. Enter the details in the text fields and click on the “save” button.  (See Appendix 1.3)  Enter the following example:  **SQL Attack**  **This is an SQL attack**  **12:49 04/18/20 PM 11:49 04/18/20 PM**  **usr/local/**  **usr/local/red**  **usr/local/white**  **usr/local/blue** | Starting the Graphical User Interface | | | The Event Configuration object is inserted into the “EventConfiguration” collection in the DB. | An acknowledgement is printed in the command prompt/ shell.  (See Appendix 1.4) |
| 3 | Click the “Close” button in the Event Configuration window in the GUI and reopen it by clicking on the Event Configuration icon in the toolbar menu. | Reflect the Event Configuration details in the GUI. | | | The Event Configuration object details coming from the document in the database are reflected in the Event Configuration window in the GUI |  |
| 4 | Edit any of the Event Configuration details through the text fields in the Event Configuration window.  Click on the “Save” button to save it in the database. | Update the changes in the DB. | | | The Event Configuration object details changes are reflected in the document within the Event Configuration collection in the DB. | An acknowledgment is printed in the command prompt/shell.  (See Appendix 1.5) |
| Concluding Remarks:  N/A | | | | | | |
| Testing Team:  Elizabeth Barragan | | | Date Completed:  05/07/20 | | | |

## Test DB 2

**Objective:** The objective of this test is to verify that the Vector object is saved into the database, loaded in the system, and updated in the database when changes are done through the GUI.

**Notes:** The Vector content stored in the DB will then be reflected in the GUI.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: DB1 | | | | Current Status: Passed | | |
| Test title: Test to save, load, update, of a Vector object in the DB | | | | | | |
| Testing approach: This test will be conducted through the Graphical User Interface and will be reflected in GUI and the command prompt/shell. | | | | | | |
| 1 | Click on the VectorDB icon in the toolbar menu to open the Vector Database window. Click on the “Add Vector” button to add a vector to the list of vectors. Add the vector details and click on the “Save Changes” button.  (See Appendix 2.1)  Enter the following example:  **Ddos**  **This is a Ddos Attack** | Starting in the Graphical User Interface | | | The Vector object is inserted into the “Vector” collection in the DB. | Assumption: The project has been started, and the MongoDB server is running in the background.  An acknowledgement is printed in the command prompt/shell.  (See Appendix 2.2) |
| 2 | Close the Vector Window in the GUI and reopen it by clicking on the Vector icon in the toolbar menu. | Reflect the Vector details in the GUI. | | | The Vector object details coming from the document in the database are reflected in the Vector Database tab. |  |
| 3 | Edit any of the Vector details through the text fields in the Main window in the Vector tab.  Click “save changes” to save it in the database.  (See Appendix 2.3) | Update the changes in the DB. | | | The Vector object details changes are reflected in the document within the Vector collection in the DB. | An acknowledgment is printed in the command prompt/shell.  (See Appendix 2.4) |
| Concluding Remarks:  N/A | | | | | | |
| Testing Team:  Elizabeth Barragan | | | Date Completed:  05/7/20 | | | |

## Test DB 3

**Objective:** The objective of this test is to delete a vector object from the DB

**Notes:** The vector deleted from the DB will be reflected in the GUI.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: DB 3 | | | | Current Status: Passed | | |
| Test title: Test to delete a Vector from the DB | | | | | | |
| Testing approach: This test will be conducted through the Graphical User Interface and will be reflected in GUI and the command prompt/ shell. | | | | | | |
| STEP  1 | OPERATOR ACTION  Click on the VectorDB icon in the toolbar menu to pop out the Vector Database window. Click on the “Delete Vector” button.  (See Appendix 3.1) | PURPOSE  Starting in the Graphical User Interface | | | EXEPCTED RESULTS    A pop-up window asks for a confirmation to delete the vector.  (See Appendix 3.2) | COMMENTS  Assumption:  The project has been started, and the MongoDB server is running in the background. |
|  | The user clicks on the “ok button to confirm deletion of the vector. | Delete Vector Confirmation | | | The vector is deleted from the list of vectors in the “Vector Database” window and the vector is deleted from the DB. | An acknowledgment is printed in the command prompt/ shell.  (See Appendix 3.3) |
| Concluding Remarks:  N/A | | | | | | |
| Testing Team:  Elizabeth Barragan | | | Date Completed:  05/7/20 | | | |

## Test Ing 1

**Objective:** The objective of this test is to confirm whether the system performs cleansing properly and if not to determine which functionality of it fails.

**Notes:** Ingestion subsystem is not completely implemented yet. Practice data must be created specifically to point out the functionality of cleansing. At least 3 files must be created for this test. Test duration should be within 5 minutes.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: Ing 1 | | | | Current Status: Passed | | |
| Test title: Test for cleansing for additional spaces, lines, & binary data. | | | | | | |
| Testing approach: This test will be performed by setting up practice data to ingest into the system | | | | | | |
| STEP | OPERATOR ACTION  Begin test by creating practice data and by saving it in the uncleansed directory  Have one file to have extra lines, another to have extra spaces, and the last one to have binary data | PURPOSE  Initial condition to properly test | | | EXEPCTED RESULTS  The system accepts the dummy files and begins process, so the user can make sure that conditions are met for cleansing | COMMENTS |
|  | Start the cleansing process by running the system | This is so you can make sure the system will accept the files and begin cleansing | | | The system begins the cleansing process without errors | If errors are found in this phase, the system doesn’t accept any files at all and is considered a failure. |
|  | After the system begins cleansing it lets the user know it is done cleansing. At this stage the user needs to check Cleansed directory made from the cleansing process. | This is to make sure that the system properly cleanses the files, and you know exactly which functionality lacks. | | | The system cleanses the files without any issues, and this can be seen in the cleansed folder. | An error occurs here, the file with the extra lines is cleansed properly along with the one with extra spacing, along with non-ascii words erased. |
| Concluding Remarks:  Test passes by removing white spaces and the non ascii characters. | | | | | | |
| Testing Team:  Yamel Hernandez | | | Date Completed:  04/26/20 | | | |

## Test Ing 2

**Objective:** The objective of this test is to confirm whether the system performs parsing through splunk properly and if not to determine which functionality of it fails.

**Notes:** Ingestion subsystem is not completely implemented yet. Practice data must be created specifically to point out the functionality of parsing through splunk.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: Ing 2 | | | | Current Status: Passed | | |
| Test title: Test Parsing in Splunk | | | | | | |
| Testing approach: This test will be performed by setting up practice data to ingest into the system | | | | | | |
| STEP | OPERATOR ACTION  Begin test by creating practice data and by saving it in the uncleansed directory | PURPOSE  Initial condition to properly test | | | EXEPCTED RESULTS  The system accepts the dummy files and begins process, so the user can make sure that conditions are met for cleansing to be sent to splunk. | COMMENTS |
|  | Start the process by starting the software. | This is so you can make sure the system will accept the files | | | The system begins the cleansing process without errors and files are then sent to splunk | If errors are found in this phase, the system doesn’t accept any files at all and is considered a failure. |
|  | After the system begins the system lets the user know, and prompts the user to input their credentials for splunk | This is to make sure that the system properly accepts the files into splunk | | | The system takes the files from the cleansed folder and uploads them to splunk. |  |
| Concluding Remarks:  Test passes by parsing properly the files. | | | | | | |
| Testing Team:  Yamel Hernandez | | | Date Completed:  4/20/20 | | | |

## Test Ing 3

**Objective:** The objective of this test is to confirm whether the system creates an enforcement action report properly and if not to determine which functionality of it fails.

**Notes:** Ingestion subsystem is not completely implemented yet. Practice data must be created specifically to point out the functionality of the report to point out errors.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: Ing 3 | | | | Current Status: Passed | | |
| Test title: Test for error messages in enforcement action report & error class | | | | | | |
| Testing approach: This test will be performed by setting up practice data to ingest into the system | | | | | | |
| STEP | OPERATOR ACTION  Begin test by creating practice data and by saving it in the uncleansed directory  Have some files with errors and know where the errors are located and what the errors are. | PURPOSE  Initial condition to properly test | | | EXEPCTED RESULTS  The system accepts the dummy files and begins process, so the user can make sure that conditions are met to create the enforcement action report. | COMMENTS |
|  | Start the process by running the system. System should first begin cleansing then it should send the log files to be checked for validation by the enforcement action report. | This is so you can make sure the system will accept the files and begin the initial steps | | | The system finds an error or errors. | If errors are not found in this phase, it means the enforcement action report isn’t properly working and it needs more work done to the class. |
|  | Once the system finds errors, a report is generated and is shown to the user. | This is to make sure that the enforcement action report is producing proper output for the files that have errors. | | | The system generates a report for which files have errors, where the errors are and what the errors are. |  |
| Concluding Remarks:  System produces a terminal output with the line number for the error and lists what the issue is. | | | | | | |
| Testing Team:  Yamel Hernandez | | | Date Completed:  4/30/20 | | | |

## Test Ing 4

**Objective:** The objective of this test is to confirm whether the system flags log files properly when ingesting.

**Notes:** Ingestion subsystem is not completely implemented yet. Test duration should be within 5 minutes.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: Ing 4 | | | | Current Status: Passed | | |
| Test title: Test ingestion process to make sure log files get flagged for the state that they are in. | | | | | | |
| Testing approach: This test will be performed by setting up practice data to ingest into the system | | | | | | |
| STEP | OPERATOR ACTION  Begin test by creating practice data and by saving it in the uncleansed directory | PURPOSE  Initial condition to properly test | | | EXEPCTED RESULTS  The system accepts the dummy files and begins process, so the user can make sure that conditions are met. | COMMENTS |
|  | Start the process by running the system & having it print out the logfile status at every stage of ingestion. | This is so you can make sure the system will accept the files and to make sure all flagging components work. | | | The system runs and rejects one file (uncleansed), creates an enforcement action report, other files get flagged cleansed, and so on for the validation & ingestion status. |  |
|  | After the system begins the process, we can check the flagging of the files in the log file class | This is to make sure that the system flags entries properly. | | | The system flags the files properly. | If no files are flagged then that means the log file class is not properly flagging the files. |
| Concluding Remarks: | | | | | | |
| Testing Team:  Yamel Hernandez | | | Date Completed:  4/30/20 | | | |

## Test EC 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: EC 1 | | | | Current Status: Pending | | |
| Test title: Test event configuration instantiation with user input data | | | | | | |
| Testing approach: This test will be performed by initiating a new project, which requires the user to input event configuration information directly. | | | | | | |
| STEP | OPERATOR ACTION  Initialize the PICK system. | PURPOSE  We must initialize the system to be prompted to begin a new project | | | EXEPCTED RESULTS  System initializes, main window pops up alongside with a pop-up window prompting the user to continue or create a new project | COMMENTS |
|  | Select option to “create new project” | This sets the environment for the user to input initial project information | | | Pop-up window progresses to next stage in project initialization |  |
|  | Provide event configuration data as follows:   * Event description: “Test” * Event start timestamp: “10:10 10/10/10 AM” * Event end timestamp: 11:11 11/11/11 PM” * Root directory: “C:/Events/10-10-10” * Red directory: “C:/Events/10-10-10/Red” * White directory: “C:/Events/10-10-10/White” * Blue directory: “C:/Events/10-10-10/Blue” | This initializes event configuration attributes to test parameters | | | Event configuration object is created and stored in database, along with generated attributes “Lead” set to TRUE, “Lead’s IP Address” set to the current machine’s IP address, and “Connections Established” set to 0. |  |
|  | Select option to begin ingestion | This option sets up the project and begins ingestion | | | The user has access to the main window |  |
|  | On the menu bar, select “File”, then select “Project” | This step opens the project information menu, which displays all the information for the current project. | | | This window defaults to the event configuration information; therefore, the tester should see the previously input test parameters in step 3, along with generated parameters mentioned in the expected results of step 3. |  |
| Concluding Remarks: This test signifies a successful recording, storing, and displaying of event configuration information that is input by a user. | | | | | | |
| Testing Team:  Jose Antoine Leon Cordero | | | Date Completed:  TBD | | | |

## Test GC1

**Objective:** Verify creation of Graph with GC1

**Notes:** It is a component test to verify correct implantation of a graph.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: GC1 | | | | Current Status: Pending | | |
| Test title: Testing for creation of a Graph | | | | | | |
| Testing approach: Testing creation of Graph with one correct input parameters and two graphs that won’t generate because of incorrect parameters past into construct.  Environment: PyCharm Professional 2019.3.2  Runtime Version: 11.0.5+10-b520.30 amd64  VM: OpenJDK 64-bit Server VM by JetBrains s.r.o  With pyQt5 == 5.14.1  Test cannot run without PyQt5 installed at least version 5.14.1, test is critical, Graph is one of the most important aspects. | | | | | | |
| STEP  1  2  3 | OPERATOR ACTION  Run “VectorTest.py” found inside “Test” directory inside of “Graph” directory. (Refer to Appendix)  Click on the “+” sign tab above the tabular view of significant log entries.  Enter “Graph1” on the name section click save. | PURPOSE  Testing creation of graph for each vector.  Generate different graphs.  Create a different vector in order to create a graph. | | | EXEPCTED RESULTS  An instance of the PICK system will open on the main window.  A pop-window asking for name and optional description will show.  A new vector will be added to the vector tabs and a new graph will be shown inside the vector. | COMMENTS |
| Concluding Remarks: Has not been tested yet because it has not been completed. | | | | | | |
| Testing Team:  Manuel | | | Date Completed: TBA | | | |

## Test GC2

**Objective:** Verify feature creation of nodes from log entries works properly.

**Notes:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: GC2 | | | | Current Status: Pending | | |
| Test title: Test creation of a node | | | | | | |
| Testing approach: Testing Graph component of a vector | | | | | | |
| STEP  1  2 | OPERATOR ACTION  Run “VectorTest.py” found inside “Test” directory inside of “Graph” directory. (Refer to Appendix)  Right click over log entry with Log Entry Name of “Log Entry 4” and select add to current vector. | PURPOSE  Testing creation of a node with predefined log entry data.  Test creation of a node from a log entry. | | | EXEPCTED RESULTS  An instance of the PICK system will open on the main window with five predefined log entries in the log entry section and one predefined vector “Vector1”.  A node with name of “Log Entry 4” will be created inside the graph of “Vector 1”. | COMMENTS |
| Concluding Remarks: Has not been tested yet because it has not been completed. | | | | | | |
| Testing Team:  Manuel Delgado | | | Date Completed: TBA | | | |

## Test GC3

**Objective:** Verify creation of a relationship between nodes in a graph is working properly.

**Notes:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: GC3 | | | | Current Status: Pending | | |
| Test title: Test creation of a relationship | | | | | | |
| Testing approach: | | | | | | |
| STEP  1  2 | OPERATOR ACTION  Run “VectorTest.py” found inside “Test” directory inside of “Graph” directory. (Refer to Appendix)  From “Vector1” graph click and drag from node with name “Node2” to node with name “Node3” and release. | PURPOSE  Test relationship creation  Create new relationship graphically. | | | EXEPCTED RESULTS  An instance of the PICK system will open on the main window with three predefined significant log entries in the tabular view in the predefined vector “Vector1”.  Once the mouse is released a new relationship will have been created between node “Node2” to “Node3”. | COMMENTS |
| Concluding Remarks: Has not been tested yet because it has not been completed. | | | | | | |
| Testing Team:  Manuel Delgado | | | Date Completed: TBA | | | |

## Test GC4

**Objective:** Verify the graph contains correct functionality for the nodes displayed.

**Notes:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: GC4 | | | | Current Status: Pending | | |
| Test title: Test functionality of movement for a node | | | | | | |
| Testing approach: | | | | | | |
| STEP  1  2 | OPERATOR ACTION  Run “VectorTest.py” found inside “Test” directory inside of “Graph” directory. (Refer to Appendix)  Inside vector “Vector1” graph click and hold on node “Node3” and drag around the graph. | PURPOSE  Test movement of nodes on a graph.  Test node movement. | | | EXEPCTED RESULTS  An instance of the PICK system will open on the main window with three predefined nodes in the graph of predefined vector “Vector1”.  “Node3” will move around the graph without any problems. | COMMENTS |
| Concluding Remarks: Has not been tested yet because it has not been completed. | | | | | | |
| Testing Team:  Manuel Delgado | | | Date Completed: TBA | | | |

## Test GC5

**Objective:** Verify that moving a node in the graph with a relationship already created moves correctly.

**Notes:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: GC5 | | | | Current Status: Pending | | |
| Test title: Test functionality of movement for a relationship | | | | | | |
| Testing approach: | | | | | | |
| STEP  1  2 | OPERATOR ACTION  Run “VectorTest.py” found inside “Test” directory inside of “Graph” directory. (Refer to Appedndix)  Click and hold node “Node2” inside vector “Vector1” graph and move around. | PURPOSE  Test the relationships predefined between nodes are working without any mistakes.  Test movement of a node with a relationship already attached to it works properly. | | | EXEPCTED RESULTS  An instance of the PICK system will open on the main window with three predefined nodes in the graph of predefined vector “Vector1”.  Relationships attached to node “Node2” will move without any problems. | COMMENTS |
| Concluding Remarks: Has not been tested yet because it has not been completed. | | | | | | |
| Testing Team:  Manuel Delgado | | | Date Completed: TBA | | | |

## Test GC6

**Objective:** Verify deletion of a node deletes the node from the table and all relevant information connected to that node and does not do anything else.

**Notes:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: GC6 | | | | Current Status: Pending | | |
| Test title: Test deletion of a node | | | | | | |
| Testing approach: | | | | | | |
| STEP  1  2 | OPERATOR ACTION  Run “VectorTest.py” found inside “Test” directory inside of “Graph” directory. (Refer to Appedndix)  Right click over node “Node1” in graph of vector “Vector1” and select delete. | PURPOSE  Test all the correct information is deleted when a node is deleted.  Delete node | | | EXEPCTED RESULTS  An instance of the PICK system will open on the main window with three predefined nodes in the graph of predefined vector “Vector1”.  Node “Node1” is deleted and relationship to node “Node2” is also deleted. | COMMENTS |
| Concluding Remarks: Has not been tested yet because it has not been completed. | | | | | | |
| Testing Team:  Manuel Delgado | | | Date Completed: TBA | | | |

## Test GC7

**Objective:** Verify information editing of a nodes content is handled, saved and displayed correctly.

**Notes:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: GC7 | | | | Current Status: Pending | | |
| Test title: Test editing of a node | | | | | | |
| Testing approach: | | | | | | |
| STEP  1  2  3 | OPERATOR ACTION  Run “VectorTest.py” found inside “Test” directory inside of “Graph” directory. (Refer to Appedndix)  Right click over node “Node1” and select edit.  Click on Log Reference editable information and delete. Type in “I changed this” and click save. | PURPOSE  Test editing of a node already existent in a graph.  Edit node information.  Test editing of anode works correctly | | | EXEPCTED RESULTS  An instance of the PICK system will open on the main window with three predefined nodes in the graph of predefined vector “Vector1”.  A pop-up window with node “Node1” information will be shown to and editable.  Pop-window will disappear and node “Node1” on graph will now contain “I changed this” in the Log Entry Reference section | COMMENTS |
| Concluding Remarks: Has not been tested yet because it has not been completed. | | | | | | |
| Testing Team:  Manuel Delgado | | | Date Completed: TBA | | | |

## Test GC8

**Objective:** Verify editing of an already existing relationship is done correctly.

**Notes:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: GC8 | | | | Current Status: Pending | | |
| Test title: Test editing of a relationship | | | | | | |
| Testing approach: | | | | | | |
| STEP  1  2  3 | OPERATOR ACTION  Run “VectorTest.py” found inside “Test” directory inside of “Graph” directory. (Refer to Appendix)  Right click over relationship between node “Node1” and node “Node2” and select edit.  Click on relationship and delete label “Relationship1” and type in “Changed Relationship” | PURPOSE  Test editing of a predefined relationship between nodes.  Edit predefined relationship between nodes.  Verify edit was updated graphically. | | | EXEPCTED RESULTS  An instance of the PICK system will open on the main window with three predefined nodes in the graph of predefined vector “Vector1”.  A pop-window will be shown with the node “Node1” information.  Pop-window will be closed and edited information will be displayed on the graph window. | COMMENTS |
| Concluding Remarks: Has not been tested yet because it has not been completed. | | | | | | |
| Testing Team:  Manuel Delgado | | | Date Completed: TBA | | | |

## Test VS1

**Objective:** Test both table and node are being updated simultaneously when information is being edited in either form.

**Notes:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: VS1 | | | | Current Status: Pending | | |
| Test title: Test editing correlation between a node and a significant log entry | | | | | | |
| Testing approach: | | | | | | |
| STEP  1  2  3 | OPERATOR ACTION  Run “VectorTest.py” found inside “Test” directory inside of “Graph” directory. (Refer to Appedndix)  Double click on node “Node1” event type.  Delete “EventType1” and type “I changed this” | PURPOSE  Verify updating information on a table updated information on that node in the graph. | | | EXEPCTED RESULTS  An instance of the PICK system will open on the main window with three predefined significant log entries in the tabular view in the predefined vector “Vector1”.  Information will be highlighted blue.  Node “Node1” on graph will reflect change on event type | COMMENTS |
| Concluding Remarks: Has not been tested yet because it has not been completed. | | | | | | |
| Testing Team:  Manuel Delgado | | | Date Completed: TBA | | | |

# Test Schedule

This section demonstrates the schedule of the tests that were ran, the individual that ran the tests, and the tests that will be ran in the future.

|  |  |  |
| --- | --- | --- |
| **Task and date** | **People** | **Description** |
| DB 1  05/07/20 | Elizabeth | Test to save, load, update, of an Event Configuration object in the DB |
| DB 2  05/07/20 | Elizabeth | Test to save, load, update, of a Vector object in the DB |
| DB 3  05/07/20 | Elizabeth | Test to delete a Vector from the DB. |
| Ing 2  04/20/20 | Yamel | Test Parsing in Splunk |
| Ing 1  04/26/20 | Yamel | Test for cleansing for additional spaces, lines, & binary data |
| Ing 3  04/30/20 | Yamel | Test for error messages in enforcement action report & error class |
| Ing 4  04/30/20 | Yamel | Test ingestion process to make sure log files get flagged for the state that they are in. |
| EC 1  04/30/20 | Antoine | Test event configuration instantiation with user input data |
| GC1  4/29/20 | Manuel | Testing creation of a graph for a vector. |
| GC2  4/29/20 | Manuel | Testing creation of a node. |
| GC3  4/30/20 | Manuel | Test creation of a relationship. |
| GC4  4/29/20 | Manuel | Test functionality of movement for a node. |
| GC5  4/30/20 | Manuel | Test functionality of movement for a relationship. |
| GC6  4/30/20 | Manuel | Test deletion of a node. |
| GC7  4/30/20 | Manuel | Test editing of a node. |
| GC8  4/30/20 | Manuel | Test editing of a relationship. |
| VS1  5/1/20 | Manuel | Test editing correlation between a node and a significant log entry. |

# Appendix

**[Test DB 1]**

****

****

**Figure 1.1.** Windows & Mac OS MongoDB Executable Path

**A screenshot of a newspaper

Description automatically generated**

**Figure 1.2.** MongoDB server running in the background

**A picture containing clock

Description automatically generated**

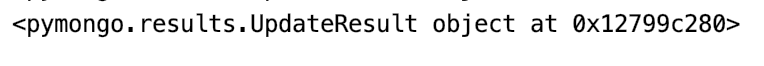
**A screenshot of a cell phone

Description automatically generated**

**Figure 1.3.** Event Configuration Toolbar icon & Event Configuration window

****

**Figure 1.4.** Acknowledgment of inserting an object in the DB

****

**Figure 1.5.** Acknowledgment of updating an object in the DB.

**[Test DB 2]**

**A picture containing clock, drawing

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated**

**Figure 2.1.** Vector DB Toolbar icon & Vector Database Window

****

**Figure 2.2.** Acknowledgment of inserting vector in DB

****

**Figure 2.3.** Acknowledgment updating vector in DB

**[Test DB 3]**

**A screenshot of a cell phone

Description automatically generated**

**Figure 3.1.** Vector window

**A screenshot of a cell phone

Description automatically generated**

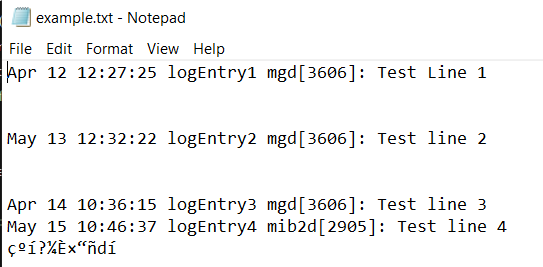
**Figure 3.2.** Pop-Up to confirm vector deletion

**A picture containing drawing

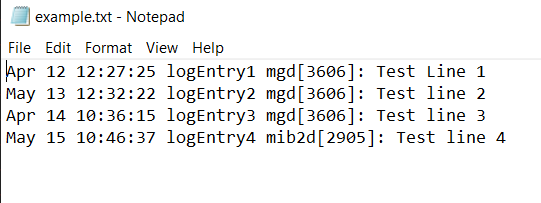
Description automatically generated**

**Figure 3.3.** Acknowledgment of deleting the vector in the DB.

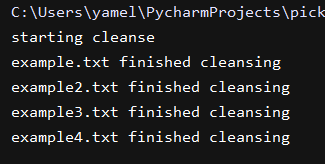
**[Test Ing 1]**

****

**Uncleansed Data with Non-Ascii Characters**

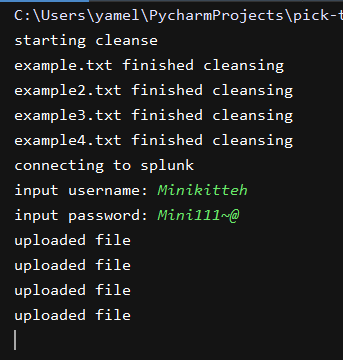
****

**Cleansed Data**

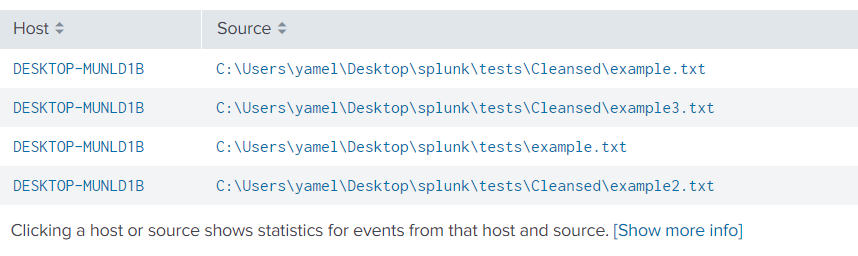
****

**Terminal Output**

**[Test Ing 2]**

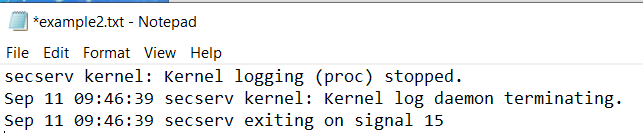
****

**Terminal Input & Output for Splunk**

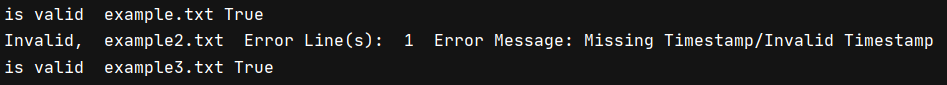
****

**Files on Splunk**

**[Test Ing 3]**

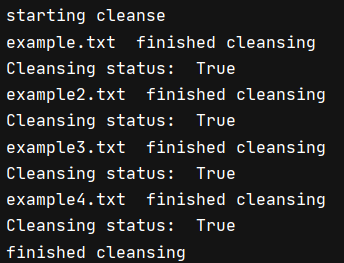
****

**File with Invalid Timestamp**

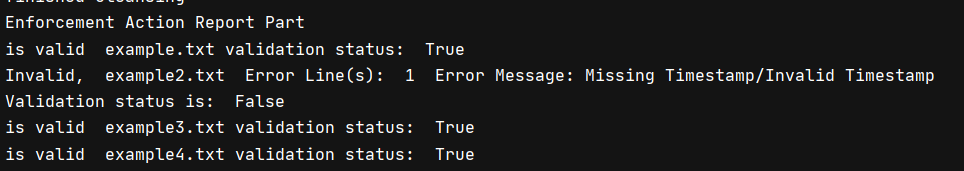
****

**Terminal Output**

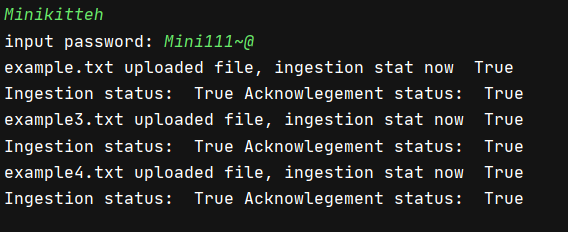
**[Test Ing 4]**

****

**Terminal Output Once Cleansing is Performed**

****

**Terminal Output Once Validation is Performed**

****

**Terminal Output Once Uploaded to Splunk**

\*