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

Test Plan

Version <.1>

4/15/20

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

|  |  |
| --- | --- |
| Initial Release: | 0.1 |
| Current Release: | 0.1 |
| Indicator of Last Page in Document: | & |
| Date of Last Review: |  |
| Date of Next Review: |  |
| Target Date for Next Update: | 4/16/2020 |

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:

Jake Lasley

Steven Roach

Customer:

Dr. Oscar Perez

Vincent Fonseca

Herandy Denisse Vazquez

Baltazar Santaella

Florencia Larsen

Erick De Nava

Software Team Members:

Jay James

Cristian Molina

Sergio Nogami

Adrian Sosa

Jesus Gomez

Change Summary

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 0.1 | 4/15 | Everyone | First draft of the Test Plan |
|  |  |  |  |
|  |  |  |  |

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

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

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

Supplementary information is from:

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

Table of Contents

[Document Control ii](#_Toc22915465)

[Approval ii](#_Toc22915466)

[Document Change Control ii](#_Toc22915467)

[Distribution List ii](#_Toc22915468)

[Change Summary ii](#_Toc22915469)

[1. Introduction 1](#_Toc22915470)

[1.1. Purpose 1](#_Toc22915471)

[1.2. Scope 1](#_Toc22915472)

[1.3. System Overview 1](#_Toc22915473)

[1.4. Suspension and Exit Criteria 1](#_Toc22915474)

[1.5. Document Overview 1](#_Toc22915475)

[1.6. References 1](#_Toc22915476)

[2. Test Items and Features 2](#_Toc22915477)

[3. Testing Approach 3](#_Toc22915478)

[4. Test XX 4](#_Toc22915479)

[4.1. Test <<test id>> 4](#_Toc22915480)

[5. User Interface Testing 5](#_Toc22915481)

[6. Test Schedule 6](#_Toc22915482)

[7. Other Sections 7](#_Toc22915483)

[8. Appendix 8](#_Toc22915484)

# Introduction

The overview of the project and test plan is explained in this section and goes into further detail in Sections 2-5. The purpose, scope, system overview, test approach overview, document overview, and references will be found in this section.

## Purpose

The purpose of the Test Plan document is to provide detailed information of the testing approach and schedule conducted for the PICK-PMR Insight Collective Knowledge (PICK). This document represents the ultimate review of the Software Requirements Specification Document (SRS), the Software Design Document (SDD) and the code used to implement the system. The purpose is to verify that the functionality of the PICK system works according to the requirements specified by the client.

The intended audience for the Test Plan is the following:

**Guidance Team Members:**

Jake Lasley

Steven Roach

**Customer:**

Dr. Oscar Perez

Vincent Fonseca

Herandy Denisse Vazquez

Florencia Larsen

Erick De Nava

**Software Team Members:**

Jay James

Adrian Sosa

Cristian Molina

Sergio Nogami

Jesus Gomez

<< Identify the project and stipulate the test plan purpose by indicating what the document contains (e.g., organizational responsibilities, test approach, test schedule. There are generally four different types of test plans: project test plan that describes the overall strategy for testing; the system test plan that describes the system from the customer’s point of view; integration test plan that describes integration of units and subsystems; unit test plan that describes modules or classes. This section needs to identify which of these this document is.>>

## Scope

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

|  |  |  |
| --- | --- | --- |
| **Version** | **Release Date** | **Description** |
| 1.0.0 | 2/6/20 | First GUI version |
| 1.1.0 | 2/20/20 | Added Filter Configuration |
| 1.2.0 | 3/5/20 | Integrated Splunk |
| 1.3.0 | 4/2/20 | Added transcription, cleansing and validation |
| 1.4.0 | 4/16/20 | Added Graph |

## System Overview

The Lethality Survivability & Human Systems Integration Directorate (LSH) works with the Department of Defense (DOD) to perform operational testing on technology systems to provide secure, resilient capabilities in the expected operational environment. The LSH or White team will analyze the flow of events that occurred during adversarial attacks between the Red and Blue teams to validate what happened during the attack. The Red team performs the attacks and the Blue team tries to mitigate them. The scope of PICK PMR Insights Collective tool will cover the need of the LSH to draw relationships between the events that occurred between the Red and Blue teams. The PMR tool will not draw the correlation of events itself but rather ease the process of drawing the correlations for the analyst.

The following functions will be tested:

* Event Configuration
* Team Configuration

## Suspension and Exit Criteria

This section provides a detailed, unambiguous description of when the team plans to stop testing, either due to the finding of several critical bugs where further testing would be undesirable or by the code passing all required tests.

<< “suspension criteria” describes when we suspend testing, to be resumed later. For example, if 40% of the test cases fail, or if any of the critical test cases fail. If there are no suspension criteria, indicate that all tests cases will be executed. “Exit criteria” indicates when testing stops. This could be based on run rate (number of test cases run divided by number of test cases specified) or pass rate (number of test cases passed divided by number of test cases run, or test cases passed divided by number of test cases specified). Nominally, we expect to run all of the specified tests. We want the pass rate to be high. We might specify that all critical tests must pass, and 90% of the non-critical must pass. In general, we want this to be high. >>

**Suspension Criteria:**

More than one critical test fails

More than 50% of non-critical tests fail

**Exit Criteria:**

All tests are run

All critical tests pass

90% of non-critical tests pass

## Document Overview

<<Describe the remainder of the document.>>

* Section 1 introduces the test plan and the purpose of the system.
* Section 2 describes the test items such as components, classes and functions of the system and the features to be tested.
* Section 3 describes the approach in detail that will be used in each test case.
* Section 4 provides detailed description of the conduction of each test.
* Section 5 is used to provide details of the user interface test.
* Section 6 contains the test schedule which has the completion dates for each testing activity.
* Section 7 has all extra information that may be related to the test plan but not to the course.
* Section 8 Appendix.

## References

[1] ￼Elsa Tai (2018), “Prevent, Mitigate, and Recover (PMR) Insight Collective Knowledge System (PICK)

Software Requirements Specification ” UTEP, Sept 2020.

[2] O. Perez et al, Requirements Definition Document, Lethality, Survivability and HSI Directorate, 2019

# Test Items and Features

<< This section describes the test items (e.g., components, classes, functions or methods) and the features to be tested. It may also list features not to be tested. A class diagram is useful. A table of features is useful. >>

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Item/Feature** | **Item/Feature Description** |
| TCI:1 | **Start and End Timestamp Format Check** | Check for valid timestamps. |
| TCI:2 | **Event Name Format Check** | Check for valid name. |
| TCI:3 | **Event Description Format Check** | Check event description is not empty. |
| TCI:4 | **Event Configuration Object Saved** | Check all fields of event configuration are valid. |
| TCI:5 | **Lead IP Address Format Check** | Check lead IP address exists and is a valid address. |
| TCI:6 | **Server Port Format Check** | Check port is valid and can be connected to for the specified IP address. |
| TCI:7 | **Number of Established Connections Input Check** | Checks number of established connections. |
| TCI:8 | **Lead Status Check** | Checks current lead status. |
| TCI:9 | **Login to Splunk Check** | Checks current connection to Splunk. |

# Testing Approach

This section describes the testing approach of the system and shall provide an explanation of the kinds of testing the team plans to use to ensure the PICK-PMR Insight Collective Knowledge system works as the users expect it to behave.

<<Describe the approach to be used to the test the system. This description includes specifying the types of tests to be performed, e.g., tests designed to exercise system functions one by one; tests designed to exercise sequences of functions that approximate operational use of the system; tests designed to stress the system to its design and requirements limits. The description lists the specific tests to be performed, but does not give the test steps. For each of these tests, give it a name and specify its objective. Label the criticality of the test cases. >>

1. Unit Testing
2. Integration Testing
3. System Testing

**3.1 Unit Testing**

For Unit testing, White-Box and Black-Box testing will be used in order to look at both the input with all the paths and the input with the expected output. White-Box will facilitate looking at the actual functionality of the code to make sure that the process is working as expected, (e.g. making sure there aren’t lines of code that are not being read). White box testing also facilitates the findings of bugs that are inside the code. Furthermore, Black-Box testing facilitates making sure that the resulting output of a method is what is expected for the associated input. This essentially ensures that even if a part of the code changes, the output should remain the same.

**3.2 Integration Testing**

For Integration testing, there will be a focus on the subsystem’s contracts and protocols. Since integration testing is about combining subsystems one by one and testing their interfaces, there will be a focus on testing the system’s collaborations through its contracts and protocols.

**3.3 System Testing**

System testing is the follow-up of integration testing. Once all the components have been unit tested and integration tested, there will be a focus on testing the entire system through system testing. This is where we tested the main components of the system. In the team’s case, the requirements for the demo were followed to ensure that they functioned the way that the guidance team and our clients expect.

Table 1: Event Configuration Test Plan

|  |  |  |
| --- | --- | --- |
| * **TEST SUITE: Event Configuration** | | |
| **Description of Test Suite** | This test suite contains the test cases for the components of Event Configuration section of the PICK System. | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| TCI:1 | **Start and End Timestamp Format Check** | **Normal** |
| TCI:2 | **Event Name Format Check** | **Normal** |
| TCI:3 | **Event Description Format Check** | **Normal** |
| TCI:4 | **Event Configuration Object Saved** | **High** |

Table 2: Team Configuration Test Plan

|  |  |  |
| --- | --- | --- |
| * **TEST SUITE: Team Configuration** | | |
| **Description of Test Suite** | This test suite contains the test cases for the components of Team Configuration Page of the PICK System. | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| TCI:5 | **Lead IP Address Format Check** | **Normal** |
| TCI:6 | **Server Port Format Check** | **Normal** |
| TCI:7 | **Number of Established Connections Check** | **Normal** |
| TCI:8 | **Lead Status Check** | **High** |
| TCI:9 | **Login to Splunk Check** | **High** |

# Test Event Configuration

This section contains a set of tests for the Event Configuration component

## Test 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: TCI: 1 | | | | Current Status: Pending | | |
| Test title: Start and End Timestamp format Check | | | | | | |
| Testing approach: Convert the timestamps to datetime objects and check if the value of the start timestamp is less than the value of the end timestamp. Ensure the format of is the following: mm/dd/yyyy hh:mm tt | | | | | | |
| STEP  1 | OPERATOR ACTION  The operator enters a start timestamp and end time stamp where the start timestamp is less than the end timestamp. | PURPOSE  To ensure that timestamp falls within a logical range. (i.e. an event cannot start after its end date) | | | EXEPCTED RESULTS  The system will display an output  message validating the timestamp range. | COMMENTS |
| Concluding Remarks: Pending | | | | | | |
| Testing Team: Pending | | | Date Completed: Pending | | | |

## Test 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: TCI: 2 | | | | Current Status: Pending | | |
| Test title: Event Name Format Check | | | | | | |
| Testing approach: Ensure the input is in format of a string excluding special characters. | | | | | | |
| STEP  1 | OPERATOR ACTION  The operator enters a string in the Event Name slot. | PURPOSE  To ensure that Event Name is a string excluding non-alphabetical characters. | | | EXEPCTED RESULTS  The system will display an | COMMENTS |
| Concluding Remarks: Pending | | | | | | |
| Testing Team: Pending | | | Date Completed: Pending | | | |

## Test 3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: TCI: 3 | | | | Current Status: Pending | | |
| Test title: Event Description Format Check | | | | | | |
| Testing approach: Ensure the input is in format of a string excluding special characters. | | | | | | |
| STEP  1 | OPERATOR ACTION  The operator enters a string with special characters in the Event Description slot. | PURPOSE  To ensure that Event Description is a string excluding non-alphabetical characters. | | | EXEPCTED RESULTS  The system will display error message. | COMMENTS |
| Concluding Remarks: Pending | | | | | | |
| Testing Team: Pending | | | Date Completed: Pending | | | |

## Test 4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: TCI: 4 | | | | Current Status: Pending | | |
| Test title: Event Configuration Object Saved | | | | | | |
| Testing approach: Convert the timestamps, event name, event description into an event configuration object and check if a local instance of the system is saved into local storage. | | | | | | |
| STEP  1.  2. | OPERATOR ACTION  The operator enters name, description, start timestamp, and end time stamp.  The operator clicks Save Button. | PURPOSE  To ensure that a local instance of the system is copied into local storage. | | | EXEPCTED RESULTS  The system will not display a message. | COMMENTS |
| Concluding Remarks: Pending | | | | | | |
| Testing Team: Pending | | | Date Completed: Pending | | | |

# Test Team Configuration

## Test 5

**Objective:** Login Check

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: TCI: 5 | | | | Current Status: Pending | | |
| Test title: Lead IP Address Format Check | | | | | | |
| Testing approach: Check if the IP Address matches IPV4 format | | | | | | |
| STEP  1. | OPERATOR ACTION  Analyst enters the lead IP Address | PURPOSE  Analyst must specify what machine it is connecting to. | | | EXEPCTED RESULTS  No format error is given. | COMMENTS |
| Concluding Remarks: Pending | | | | | | |
| Testing Team: Pending | | | Date Completed: Pending | | | |

## Test 6

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test No.:** TCI: 6 | | | | **Current Status:** Pending | | |
| **Test title:** Server Port Format Check | | | | | | |
| **Testing approach:** Port should be an integer value between 1024 and 49151. Ports 1024-49151 are the User Ports and are the ones to use for your own protocols. | | | | | | |
| STEP  1. | OPERATOR ACTION  Analyst enters the port to connect. | PURPOSE  Analyst must specify what port they are connecting to. | | | EXEPCTED RESULTS  No format error is given. | COMMENTS |
| Concluding Remarks: Pending | | | | | | |
| Testing Team: Pending | | | Date Completed: Pending | | | |

## Test 7

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: TCI: 7 | | | | Current Status: Pending | | |
| Test title: Number of Established Connections Check | | | | | | |
| Testing approach: Established connections must be accurate and must enforce a limit of 20. | | | | | | |
| STEP  1.  2. | OPERATOR ACTION  Analyst enters the lead IP Address.  Analyst clicks Connect. | PURPOSE  We must see that the number of established connections accurately reflects the number of analysts connected. | | | EXEPCTED RESULTS  Established connections will increase or decrease accordingly when analysts connect or disconnect. If 20 analysts are connected, an error message stating the limit of connections is reached will be displayed. | COMMENTS |
| Concluding Remarks: Pending | | | | | | |
| Testing Team: Pending | | | Date Completed: Pending | | | |

## Test 8

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: TCI: 8 | | | | Current Status: Pending | | |
| Test title: Lead Status Check | | | | | | |
| Testing approach: Lead IP indicator must display for lead IP address entered. | | | | | | |
| STEP  1.  2. | OPERATOR ACTION  Analyst enters the lead IP Address.  Analyst with the matching lead IP connects. | PURPOSE  A lead IP for the event must be configured.  The analyst with the lead IP must be connected for the check box option | | | EXEPCTED RESULTS  The analyst with the IP address that matches the configured lead IP for the event will see an indicator that shows they are the lead IP. | COMMENTS |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

## Test 9

**Objective:** Login Check

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: TCI:9 | | | | Current Status: Pending | | |
| Test title: Login to Splunk Check | | | | | | |
| Testing approach: This test will need a valid Splunk account, valid IP address | | | | | | |
| STEP  1.  2.  3. | OPERATOR ACTION  Analyst enters the lead IP address, server port, name of Splunk index, Splunk username, Splunk password.    Check box if analyst the lead analyst    Click connect. | PURPOSE  Splunk credentials are verified, connection to Splunk is verified, host machine is identified, user is identified as host or client. | | | EXEPCTED RESULTS  Analyst connects to Splunk, status of analyst is created, and confirmation of connection appears on screen. | COMMENTS  If any of the inputs are incorrect the test should identify the incorrect input to the user. |
| Concluding Remarks: Pending | | | | | | |
| Testing Team: Pending | | | Date Completed: Pending | | | |

# User Interface Testing

<<This section focuses on the interaction between the user and the system. For testing the user interface, consider the following traits:

* Consistent terminology, shortcut keys, menu selections, and presentation
* Correct language, spelling, and grammar.
* Flexibility in navigation between windows and interface elements.
* Error handling that will inform user of critical operations.
* Follows standards and guidelines such as placement of scroll bars, windows, and menu items.

This section could be integrated into Section 4.

>>

# Test Schedule

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

|  |  |  |
| --- | --- | --- |
| **Task and date** | **People** | **Description** |
| TCI:1 |  |  |
| TCI:2 |  |  |
| TCI:3 |  |  |
| TCI:4 |  |  |
| TCI:5 |  |  |
| TCI:6 |  |  |
| TCI:7 |  |  |
| TCI:8 |  |  |
| TCI:9 |  |  |

# Other Sections

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

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

>>

# Appendix

<< possibly more readable to put the expected output here and refer to it in the previous sections. Might also provide explicit directions for analysis of output, if it’s easier to read as an appendix or if analysis is post execution. >>