**Prevent, Mitigate, and Recover (PMR) Insight**

Collective Knowledge System (PICK)

Test plan

Version 1.0

04/16/2020

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

|  |  |
| --- | --- |
| Initial Release: | 1.0 |
| Current Release: | 1.0 |
| Indicator of Last Page in Document: | & |
| Date of Last Review: | 04/16/2020 |
| Date of Next Review: | 04/20/2020 |
| Target Date for Next Update: |  |

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

Elsa Tai Ramirez

Peter Hanson

Customer:

Dr. Oscar Perez

Vincent Fonseca

Herandy Denisse Vazquez

Baltazar Santaella

Florencia Larsen

Erick De Nava

Software Team Members:

Hector Dozal

Victor Vargas

Eduardo Lara

Irvin Bosquez

Gerardo Armenta

Change Summary

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1.0 | 4/14/2020 | Victor, Irvin, Eduardo | Progress in the section 1, 3 and 4 |
|  |  |  |  |
|  |  |  |  |

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

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

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

Supplementary information is from:

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

Table of Contents

[Document Control ii](#_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 PMR Insight Collective Knowledge (PICK) tool is the system to be tested for in this document. All test cases provided will be either confirmed to be working or to have failed the test. Below you will find the following test to be performed, along with the expected results are, and the team members who will be conducting the tests.

## Purpose

The main purpose of this document is to provide a proper guideline to test the software and deliver a functional project to the clients. Also, we are trying to identify the faults and errors that the project PICK that can contain.

This document will focus on the project PICK and it is a Project Test Plan, meaning it will focus on testing the project in general, specifying different parts of it in the different test suites. It contains different test suites focusing on different functionalities of the project, test cases and their descriptions, and a testing schedule.

## Scope

We will be testing a PMR Insight Collective Knowledge (PICK) tool to facilitate the process of writing reports by the White Team (LSH) about the ability of the blue team to defend against cyber-attacks by the red team. This software’s primary goal will consist of facilitating the job of the white team by employing different tools including: Sorting by chronological order based on the date of ingestion, automatic creation of a graphs to better represent when an attack has occurred, creation of vectors to organize all log files that pertain to a singular event, as well as other requirements described within this document. Our system shall focus firstly on improving the analysis of log files, our system shall not use the internet in any way to ensure the security of the system. The success of this project will be determined by the benefits given to the White team upon using this system.

## System Overview

The system that is being exercised is a PMR Insight Collective Knowledge (PICK) tool to facilitate the process of writing reports by the White Team (LSH) about the ability of the blue team to defend against cyber-attacks by the red team. This software’s primary goal will consist of facilitating the job of the white team by employing different tools including: Sorting by chronological order based on the date of ingestion, automatic creation of a graphs to better represent when an attack has occurred, creation of vectors to organize all log files that pertain to a singular event, as well as other requirements described within this document. The system’s functionalities being tested are mainly the ingestion and cleansing of log files; the graphing functionalities and other functions that pertain to the main goal of facilitating analysis.

## Suspension and Exit Criteria

For suspension criteria, we will suspend testing if 50% or more of the critical tests fail, and if 60% or more of the non-critical test fail. For exit criteria, we will set the requirement that all critical tests must pass and 80% or more of the non-critical tests must pass.

## Document Overview

The remainder of this test plan document will go into greater detail about the test plan. Section 2 will describe the test items and features, which includes (components, classes, function or methods), to be tested. Section 3 will describe the testing approach that will be used to test the system. It will describe the types of tests that will be performed for each system function one by one and label the criticality of each test case. Section 4 will document test input, specific test procedures, and outcomes for each specified test case. Section 5 will describe user interface testing by detailing the tests by how the user will interact with the system. Section 6 describes the test plan schedule, which will show the members who will be in charge of specific test completions by a certain date. Section 7 explains other test sections that are used to detail the test plan document in greater detail. Section 8 is the appendix and in there will be more detailed results from specific tests that were done throughout the test plan.

## References

[1] E. Tai-Ramirez & S. Roach, SRS\_v7. Internet: <https://github.com/CS4311-spring-2020/pick-tool-team06-team-404/blob/master/doc/SRSv7.pdf>, 2020 (Jan. 30, 2020).

# Test Items and Features

The following items are the ones that are going to be tested for in this document. Our team will be focusing on the main integral parts of the system to be tested, which includes log ingestion, conversion on logs into nodes, and ability to modify and create relationships between nodes in graphical and table formats. Each parts of the system to be tested has certain features associated to that part and will be tested as well.

|  |  |
| --- | --- |
| **TEST ITEMS & FEATURES** | |
| **Test Item** | **Features** |
| Log Ingestion | Cleansing |
| Log Ingestion | Validating |
| Log Ingestion | Ingesting |
| Nodes | Log Entry into Node |
| Graph Interaction | Modify Nodes |
| Graph Interaction | Relationship Between Nodes |

# Testing Approach

<<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. >>

Table 1: Test Plan

|  |  |  |
| --- | --- | --- |
| **TEST SUITE <Log File Ingestion>** | | |
| **Description of Test Suite** | **Used to test if log files from directories are ingested into the system without any errors** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| LFI I1 | **Cleanse log files** | **Critical** |
| LFI I2 | **Validate log files** | **Critical** |
| LFI I3 | **Ingest log files** | **Critical** |
|  |  |  |

Table 2: Test Plan

|  |  |  |
| --- | --- | --- |
| **TEST SUITE <Graph Interaction>** | | |
| **Description of Test Suite** | **Used to test some utilities about the graph whenever it is created.** | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| GI I1 | **Graph Node are created** | **Critical** |
| GI I2 | **Graph availability to move** | **Not Critical** |
| GI I3 | **Accept relationships between Nodes** | **Critical** |
| GI I4 | **Delete Nodes from graph** | **Not Critical** |

# Test XX

<<The purpose of this section is to:

* document test input, specific test procedures, and outcomes.
* establish test methods,
* explain the nature and extent of each test >>

<< for each test case, complete the following: >>

## Test LFI I1

**Objective:** The objective of this test is to verify that the system performs cleansing on the log files before ingesting.

**Notes:** Log files are supposed to be cleansed and validated before being ingested into the system, this test will make sure the log files are all cleansed and ready for validation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: LFI I1 | | | | Current Status: Passed | | |
| Test title: Cleanse log files to prepare them to be ingested | | | | | | |
| Testing approach: This test will be conducted using directories with different log files in a directory folder. Results will be viewed after cleansing is done. | | | | | | |
| STEP  <<N>> | OPERATOR ACTION  Describe the actions taken by the person executing the test procedure. Include the test suite, or the name of the test file (in this case, the contents of the file should be given in the appendix). | PURPOSE  Describe the reason for the step. | | | EXEPCTED RESULTS  Describe the expected response of the system being tested to the action specified under OPERATOR ACTION. This should be derived from the SRS and SDD. Clearly indicate how we determine whether the step passes. | COMMENTS |
| 1 | Begin test by having a directory containing folders with multiple files with different unwanted characters in every file. | Set up directories with files to be tested. | | | Files will be filled with unwanted and wanted characters. |  |
| 2 | Begin cleansing process. | Cleanse files from unwanted characters. | | | All files will not have any unwanted characters when finished. |  |
| Concluding Remarks: Cleansing script that was created is working well and removing all unwanted characters in a log file | | | | | | |
| Testing Team:  Hector Dozal | | | Date Completed:  04/14/2020 | | | |

# User Interface Testing

There are some interfaces in which the user has to interact with the application. We will try to go over some of them. Some of the things that will be tested are the following:

* Switch between tabs
* Dead point in which a node can be moved

# Test Schedule

Below you will find the following test schedule that will be used to show who will conduct certain tests and what date they are expected to test.

|  |  |  |
| --- | --- | --- |
| **Task and date** | **People** | **Description** |
| LFI I1  04/14/2020 | Hector Dozal | Created Cleansing script to remove unwanted characters from log files |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 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. >>