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

CRC Classes

Version 1.0

2/18/2020

Document Control

Approval

The Guidance Team and the customers 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: | 2/18/2020 |
| Date of Next Review: | 2/21/2020 |
| Target Date for Next Update: | 2/28/2020 |

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:

Steven Roach

Elsa Tai Ramirez

Customer: Dr. Oscar Perez

Vincent Fonseca

Herandy Denisse Vazquez

Baltazar Santaella

Florencia Larsen

Erick De Nava

Software Team Members:

Jose Antoine Leon Cordero

Elizabeth Barragan

Manuel Delgado

Yamel Hernandez

Abel Rodriguez

Change Summary

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1.0 | 2/18/20 | Team 5 | Finalize and merge all team member’s CRC classes |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Document Control ii](#_Toc32916521)

[Approval ii](#_Toc32916522)

[Document Change Control ii](#_Toc32916523)

[Distribution List ii](#_Toc32916524)

[Change Summary ii](#_Toc32916525)

[1. Classes & Responsibilities 1](#_Toc32916526)

[1.1. Graphical User Interface 1](#_Toc32916527)

[1.1.1. Change Configuration 1](#_Toc32916528)

[1.1.2. Directory Configuration 1](#_Toc32916529)

[1.1.3. Event Configuration 1](#_Toc32916530)

[1.1.4. Export Configuration 1](#_Toc32916531)

[1.1.5. Filter Configuration 1](#_Toc32916532)

[1.1.6. Graph Builder Configuration 2](#_Toc32916533)

[1.1.7. Icon Configuration 2](#_Toc32916534)

[1.1.8. Log Entry Table 2](#_Toc32916535)

[1.1.9. Log File Table 2](#_Toc32916536)

[1.1.10. Node Graph 3](#_Toc32916537)

[1.1.11. Node Table 3](#_Toc32916538)

[1.1.12. Relationship Table 3](#_Toc32916539)

[1.1.13. Team Configuration 3](#_Toc32916540)

[1.1.14. Vector Configuration 3](#_Toc32916541)

[1.1.15. Database Configuration 4](#_Toc32916542)

[1.2. Backend 4](#_Toc32916543)

[1.2.1. Audio Transcriber Module 4](#_Toc32916544)

[1.2.2. Database Module 4](#_Toc32916545)

[1.2.3. Ingestion Module 4](#_Toc32916546)

[1.2.4. Maltego Module 5](#_Toc32916547)

[1.2.5. Network Module 5](#_Toc32916548)

[1.2.6. OCR Module 5](#_Toc32916549)

[1.2.7. Project Module 5](#_Toc32916550)

[1.2.8. Splunk Module 5](#_Toc32916551)

[1.2.9. Version Control Module 6](#_Toc32916552)

[1.3. Data 6](#_Toc32916553)

[1.3.1. Enforcement Action Report 6](#_Toc32916554)

[1.3.2. Event Configuration 6](#_Toc32916555)

[1.3.3. Graph 6](#_Toc32916556)

[1.3.4. Icon 6](#_Toc32916557)

[1.3.5. Log File 7](#_Toc32916558)

[1.3.6. Node 7](#_Toc32916559)

[1.3.7. Relationship 7](#_Toc32916560)

[1.3.8. Significant Log Entry 7](#_Toc32916561)

[1.3.9. Vector 7](#_Toc32916562)

[2. Collaborations 8](#_Toc32916563)

[2.1. Graphical User Interface 8](#_Toc32916564)

[2.1.1. Change Configuration 8](#_Toc32916565)

[2.1.2. Directory Configuration 8](#_Toc32916566)

[2.1.3. Event Configuration 8](#_Toc32916567)

[2.1.4. Export Configuration 8](#_Toc32916568)

[2.1.5. Filter Configuration 8](#_Toc32916569)

[2.1.6. Graph Builder Configuration 8](#_Toc32916570)

[2.1.7. Icon Configuration 8](#_Toc32916571)

[2.1.8. Log Entry Table 8](#_Toc32916572)

[2.1.9. Log File Table 8](#_Toc32916573)

[2.1.10. Node Graph 8](#_Toc32916574)

[2.1.11. Node Table 8](#_Toc32916575)

[2.1.12. Relationship Table 9](#_Toc32916576)

[2.1.13. Team Configuration 9](#_Toc32916577)

[2.1.14. Vector Configuration 9](#_Toc32916578)

[2.1.15. Database Configuration 9](#_Toc32916579)

[2.2. Backend 9](#_Toc32916580)

[2.2.1. Audio Transcriber Module 9](#_Toc32916581)

[2.2.2. Database Module 9](#_Toc32916582)

[2.2.3. Ingestion Module 9](#_Toc32916583)

[2.2.4. Maltego Module 9](#_Toc32916584)

[2.2.5. OCR Module 9](#_Toc32916585)

[2.2.6. Project Module 9](#_Toc32916586)

[2.2.7. Splunk Module 9](#_Toc32916587)

[2.2.8. Version Control Module 9](#_Toc32916588)

[2.3. Data 9](#_Toc32916589)

[2.3.1. Enforcement Action Report 9](#_Toc32916590)

[2.3.2. Event Configuration 9](#_Toc32916591)

[2.3.3. Graph 9](#_Toc32916592)

[2.3.4. Icon 9](#_Toc32916593)

[2.3.5. Log File 9](#_Toc32916594)

[2.3.6. Node 10](#_Toc32916595)

[2.3.7. Relationship 10](#_Toc32916596)

[2.3.8. Significant Log Entries 10](#_Toc32916597)

[2.3.9. Vector 10](#_Toc32916598)

# Classes & Responsibilities

This section will identify the classes the PICK system will contain and responsibilities of classes to achieve the intended behavior of the PICK system. A responsibility details a requirement for the class. Classes will fall into one of three categories: Graphical User Interface (GUI), Backend, or Data; These categories specify an overall goal of the classes.

## Graphical User Interface

### Change Configuration

The Change Configuration class is in charge of logging all of the changes done in the system.

The following is a list of responsibilities for the Change Configuration class:

1. Keeps track of any changes in the system displaying them in a list.
2. Provides a button to undo and or commit.

### Directory Configuration

The Directory Configuration is in charge of providing an interface to collect data relevant to directory structure from the user.

The following is the list of responsibilities for the Directory Configuration class:

1. Provides fields for “Root directory”, “Red Team’, “Blue Team”, “White Team” to be specified.
2. Provides button to start “data ingestion”.
3. Collaborates with the Project Module class by giving the class the information collected.

### Event Configuration

The Event Configuration is in charge of providing an interface to collect data relevant to an event.

The following is the list of responsibilities for the Event Configuration class:

1. Provides fields for “Event name”, “Event description”, “Event start time stamp”, “Event end time stamp”.
2. Provide a button to save event.
3. Collaborates with Project Module class by giving the information collected.

### Export Configuration

The Export Configuration class is in charge of allowing for a graph to be exported in a specified format.

The following is a list of responsibilities for the Export Configuration class:

1. Provides the options to export the graph in the formats of a PNG or JPEG.

### Filter Configuration

The Filter Configuration window is in charge of displaying entries and constraints the user sets on how they would like to see entries, either containing keyword(s), by teams, event type, and by the time entries contain.

The following is the list of responsibilities for the Filter Configuration class:

1. Filtering entries requested by keywords
2. Filtering request by the creator team
3. Filtering request by the event type
4. Filtering entries requested through start timestamps and end timestamps

### Graph Builder Configuration

The Graph Builder Configuration class is in charge of navigating through vectors and being able to create a graph through various operations.

The following are the responsibilities for Graph Builder Configuration class:

1. Selecting a vector
2. Adding a description to a vector
3. Adding / Deleting a node
4. Adding/ Deleting a relationship
5. Editing a node
6. Editing a relationship

### Icon Configuration

The Icon Configuration purpose is to allow the user to modify the specified icons by changing their appearance from a default icon to uploading an image for them. This configuration window is also in charge of displaying a preview of the uploaded image, so the user can see the change of icon before submitting it.

The following is the list of responsibilities for the Icon Configuration class:

1. Add an icon
2. Delete and icon
3. Preview an icon

### Log Entry Table

The Log Entry Table GUI window purpose is to display all the log entries that are saved in the system and be able to manipulate the way the log entries are listed from ascending or descending order based on the attribute’s column header the user clicks on.

The following is the list of responsibilities for the Log Entry Table class:

1. Display all log entries currently saved in the system’s log entry database in a log entry table

2. Format all the log entries’ data into the log entry table by sectioning off its attributes in the appropriate columns of “List number”, “Log entry timestamp”, and “Log entry event”.

3. Manipulate the ordering of all the log entries in the log entry table from either ascending to descending order based on the attribute selected by the user

4. Allow the user to check or de-check the checkbox next to each log entry

5. List all the vectors saved in the system’s database within drop-down boxes in the “Vector” column of the log entry table

6. Request all log entries’ data from system’s database in order to create and display the log entry table

7. Request a reference to all vectors from the system’s database in order to list all the vectors within drop-down boxes in the “Vector” column of the log entry table

### Log File Table

The Log File Table GUI window purpose is to display all the log files’ that have been ingested and all log files that have tried to be ingested. The window will also be able to manipulate the way the log files are listed from ascending or descending order based on the attribute’s column header the user clicks on. Lastly, the window will be able to display each log file’s enforcement action report when the user clicks

The following is the list of responsibilities for the Log File Table class:

1. Display all log files currently saved in the system’s log file database in a log file table
2. Format all the log files’ data into the log file table by sectioning off its attributes in the appropriate columns of “File name”, “Source”, “Cleansing status”, “Validation status”, and “Ingestion status”.
3. Manipulate the ordering of all the log files in the log file table from either ascending to descending order based on the attribute selected by the user
4. Display the chosen log file’s enforcement action report within an enforcement action report table when the user presses the chosen log file’s “View enforcement action report” button.
5. Format the enforcement action report data into the enforcement action report table by sectioning off its attributes in the appropriate columns of “File name”, “Line Number”, and “Error Message”.
6. Send verification action confirmation and the associated log file to the ingestion module to manually validate log file when the user presses the “Validate” button within the enforcement action report table section
7. Request all log files’ data from system’s database to create and display a log file table

### Node Graph

The Node Graph class allows the nodes to be displayed graphically being able to manipulate the nodes in the graph.

The following is a list of responsibilities for the Node Graph class:

1. Display nodes in a timeline.
2. Specify intervals in which the nodes will be displayed.
3. Have the ability of zoom in and zoom out the nodes in the graph.

### Node Table

The Node Table class allows the nodes to be displayed within a table along with its node properties.

The following is a list of responsibilities for the Node Table class:

1. Display nodes in a table.
2. The node’s properties that will be displayed are; Visibility, ID, Name, Timestamp, Description, Log Entry Reference, Log Creator, Event Type, Icon Type, and Source.

### Relationship Table

The Relationship Table GUI window’s purpose is to get information from the user to create correlations between node entries. This window creates relationships by asking the user for the parent and child of the relationship.

The following is the list of responsibilities for the Relationship Table class:

1. Request Label Name
2. Requesting Parent
3. Requesting Child
4. Displays Relationship ID

### Team Configuration

The Team Configuration GUI window will prompt the user to input an I.P. address to connect to, or in the case that an analyst will be a host, it will prompt them to enter their I.P. address so that others can connect. Moreover, the team configuration window will display the number of analysts working on a project.

The following is the list of responsibilities for the Team Configuration class:

1. Record user input regarding I.P. address to connect to and user role (analyst or lead)
2. Display current count of analysts on the PICK system

### Vector Configuration

Vector Configuration is class that will provide fields to have information inputted.

The following is the list of responsibilities for the Vector Configuration class:

1. Provides buttons to “Add vector”, “Delete vector”, and “Edit vector”.
2. Provides fields in a table to input and display “Vector name”, “Vector description”.
3. Provides ability to sort information inside of the table to rearrange it.
4. Collaborates with the Database Module class to save information of each individual vector.

### Database Configuration

The Database Configuration window is composed of setting two different interfaces for the lead and analyst. The analyst’s window will allow them to pull and download a vector from the database and push it back to the database when they are done working on it. The lead’s window will get the analyst’s pushed work and resolve merging conflicts between vectors and save them to the database.

The following is the list of responsibilities for the Database Configuration class:

1. Resolving conflicts when merging vectors.
2. Pull requests from the user.
3. Push requests from the user.

## Backend

### Audio Transcriber Module

The Audio Transcriber Module’s main purpose is to transcribe both audio files and the audio from video files into text for the system to verify if the information can be validated and ingested into log entries.

The following is the list of responsibilities for the Audio Transcriber class:

1. Transcribe audio file sent by ingestion module into text file
2. Transcribe audio from video file sent by ingestion module into text file
3. Send text file of transcribed audio file to ingestion module to verify the text data
4. Send text file of transcribed audio from video file to ingestion module to verify the text data

### Database Module

The Database Module is in charge of storing, interpreting, and managing all project data. This module will consist of a relational database which will store tables of data, where each data type will have its own table. Moreover, the database of this module will enable the use of filtering, searching, and editing of data.

The following is the list of responsibilities for the Project Module class:

1. Transform changes or user input into database queries.
2. Transform database query results into data objects.
3. Collaborates with the Filter Configuration class by receiving queries based on filter inputs. [1.1.5.1 – 1.1.5.4]
4. Collaborates with the Icon Configuration class by adding or removing icons from the database. [1.1.7.1 – 1.1.7.2]
5. Collaborates with the Log Entry Table class by sending log entries for it to display. [1.1.8.6]
6. Collaborates with the Log File Table class by sending all log files for it to display. [1.1.9.6]
7. Collaborates with the Node Table class by sending all node data related to the current vector for it to display. [1.1.11.2]
8. Collaborates with the Relationship Table class by sending all relationship data related to the current vector for it to display. [1.1.12.1 – 1.1.12.4]
9. Collaborates with the Vector Configuration class by sending all vector information for it to display. [1.1.14.4]
10. Collaborates with the Ingestion Module class by storing all ingested files and their metadata. [1.2.3]
11. Collaborates with the Maltego Module class by storing all graph information for it to interpret. [1.2.4]
12. Collaborates with the Project Module by sending it all project information for it to disperse amongst other project classes. [1.2.7]
13. Collaborates with all Data classes to store and interpret object data. [1.3.1 – 1.3.9]

### Ingestion Module

The ingestion module is in charge of leveraging other backend modules to interpret, format, and ingest log files. Moreover, the ingestion module is in charge of scanning for changes to log files in project directories and updating log files/entries if any changes are detected.

The following is the list of responsibilities for the Project Module class:

1. Scanning for files in project directories.
2. Scanning for changes to files.
3. Appending ingestion results as metadata to ingested files.
4. Converting validated files into log entries.

### Maltego Module

The Maltego Module’s main purpose is to be able to send and receive data from the Maltego software in order to create a graph from a vector’s nodes’ and relationships’ data.

The following is the list of responsibilities for the Maltego Module class:

1. Send a vector’s nodes’ and relationship’s data into Maltego

2. Receive the vector’s graph that is outputted by Maltego

3. Create a graph from a vector that is sent from DB manager

4. Send the vector’s graph to DB manager

5. Create an updated graph from a vector that is sent from version control

6. Send the updated graph of the vector to version control

### Network Module

The Network module is in charge of sending and receiving project data and changes between clients. To do this, this module establishes the data interchange formats, decodes/encodes project data, and establishes a safe transmission protocol of the data over a closed network.

The following is the list of responsibilities for the Project Module class:

1. Establishing secure, closed connections with PICK clients/server.
2. Encode project data into JSON for transfer.
3. Decode JSON data received from clients/server over the network.

### OCR Module

The OCR Module’s main purpose is to be able to convert any image file into text for the ingestion module to be able to verify the data on the image.

The following is the list of responsibilities for the OCR Module class:

1. Convert image file sent from ingestion module into text file
2. Send text file of transformed image file to ingestion module to verify the text data

### Project Module

The Project Module is in charge of interpreting project settings and connecting the PICK system. It will act as the core of the system – the central point where input data (or output) is formatted and routed to the Graphical User Interface or other Backend components.

The following is the list of responsibilities for the Project Module class:

1. Load data from database for the GUI to interpret.
2. Process user input into changes.
3. Process information from backend into changes.
4. Collaborates with Directory Configuration to receive the directories for the project. [1.1.2.3]
5. Collaborates with all GUI classes by storing any user input as a change to be applied [1.1.1 – 1.1.15]
6. Interacts with Version Control Module by sending it a list of changes staged for commit [1.2.9.4]
7. Collaborates with Event Configuration class to receive the project event details. [1.1.3.3]

### Splunk Module

The Splunk Module’s main purpose is to be able to transform validated log files into log entries so the system can store into the log entry database.

The following is the list of responsibilities for the Splunk Module class:

1. Transform all the validated log files from the ingestion module into log entries

2. Send all the newly created log entries to the ingestion module to store in long entry database

### Version Control Module

The Version Control Module is in charge of packaging project changes into a commit, which is sent to the lead analyst for review. As a result, this module is also in charge of providing a sub-system to interpret and handle the merging of incoming commit requests. Upon a successful merge, this module will also be responsible for applying the merged changes.

The following is the list of responsibilities for the Version Control Module class:

1. Interpret a commit as a list of changes
2. Send accepted changes to database manager for application
3. Provide a mechanism to resolve conflicting changes
4. Collaborates with Project module by receiving changes staged for commit. [1.2.7.6]

## Data

### Enforcement Action Report

The Enforcement Action Report is in charge of knowing errors of every log file ingested.

1. The following is the list of responsibilities for the Enforcement Action Report class:
2. It will keep track of what line in the log file an error was detected and what type of error it is.
3. Collaborate with Database Module to provide line number and error message.

### Event Configuration

The Event Configuration class will store information pertaining to an event given by a user.

The following is the list of responsibilities for the Event Configuration class:

1. It will save the following attributes pertaining to a single event: event name, event description, event start timestamp, event end timestamp, root directory, red, blue, white team folders, lead, lead’s IP address, and connections stablished.
2. Collaborate with the Database Module class to save all the information regarding the Event Configuration class.
3. Collaborate with Database Module to declare a lead.
4. Collaborate with Database Module to obtain the number of established connections.

### Graph

The Graph class will be in charge of creating visual representations of a timeline created by the users.

The following is the list of responsibilities for the Graph class:

1. Exporting as a jpeg and/or png
2. Graph orientation
3. Position of Nodes
4. Position of relationships
5. Interval

### Icon

The Icon data class will store various icons specified from the user and will keep track of the file path it comes from.

The following is the list of responsibilities for the Icon class:

1. Storing icon name
2. Storing the file path.

### Log File

The Log File class assigns attributes that make files ingested trackable.

1. The following is the list of responsibilities for the Log File class:
2. It will assign these attributes “Cleansing Status”, “Validation Status”, “Ingestion Status”, “Acknowledgement Status”.
3. Collaborates with the class Database Module for verifying individual log files.

### Node

The Node class creates significant log entries into nodes, allowing the nodes to be represented in a graph.

The Node’s class responsibilities are the following:

1. The attributes being assigned to each node consist of an ID, Name, Timestamp, Node Description, Log Entry Reference, Log Creator, Event Type, Icon Type, Source and Node Visibility.
2. This class is able to perform the operations of adding, deleting, editing, filtering and hiding nodes.

### Relationship

The Relationship class stores and creates correlations between nodes that allow the nodes to be connected in the system’s graph.

The following is the list of responsibilities for the Relationship class:

1. Creating a Relationship ID
2. Label for relationship
3. Nodes involved in the relationship, the Parent ID and Child ID.

### Significant Log Entry

The Significant Log Entry class stores all data from a log entry created from validated log files that has been associated to at least one vector by the user.

The Significant Log Entry’s class responsibilities are the following:

1. Store/know the following attributes of the log entry that is associated to a vector : “Log Entry Number”, “Log Entry Timestamp”, “Log Entry Content”, “Host”, “Source”, and “Source Type”.
2. Collaborate with DB Manager to edit allowed Significant Log Entry’s attributes
3. Collaborate with DB Manager to associate or disassociate Significant Log Entry to/from vector
4. Collaborate with DB Manager to send any or all Significant Log Entry’s data that is requested by DB manager

### Vector

1. The Vector class will save information relevant to a vector.
2. The following is the list of responsibilities for the Vector class:
3. It will store attributes of “Vector name” and “Vector description” for every vector.
4. Collaborate with Database Module to save vector name for every vector.
5. Collaborate with Database Module to save vector description for every vector.

# Collaborations

This section will describe collaborations among our system’s classes. A collaboration details a responsibility in which a class explicitly communicates with or leverages other classes. The format used for the collaboration of each class by listing below what other class is needed for the collaboration on the left and the numbers on the right is the responsibility number of the class in bold that requires a collaboration with the other class defined on the left.

## Graphical User Interface

### Change Configuration

Version Control Module :1-3

Database Module: 1-2

### Directory Configuration

Project Module: 1 & 2

### Event Configuration

Project Module: 1 & 2

### Export Configuration

Graph: 2-4

### Filter Configuration

DB Manager: 1 - 4

### Graph Builder Configuration

Node: 1-2

Graph 2-4

Relationship: 1-3

Vector: 1-3

Version Control Module: 1-3

Database Module: 1-2

Project Module: 1-3

### Icon Configuration

DB Manager: Responsibility 1 – 3

Graph: Responsibility 1 – 3

Node: Responsibility 1 - 3

### Log Entry Table

DB Manager: Responsibility 6 & 7

### Log File Table

DB Manager: Responsibility 6 - 8

### Node Graph

Database Module: Responsibility 1-3

### Node Table

Database Module: Responsibility 1-3

### Relationship Table

Version Control: Responsibility 1 - 3

### Team Configuration

### Vector Configuration

Project Module: Responsibility 1 & 2

### Database Configuration

Version Control: Responsibility 1 - 3

## Backend

### Audio Transcriber Module

Ingestion Module: Audio Transcriber Module Responsibility 3,4

### Database Module

### Ingestion Module

### Maltego Module

DB Manager: Maltego Module Responsibility 3,4

Ingestion Module : Maltego Module Responsibility 5,6

### OCR Module

Ingestion Module: OCR Module Responsibility 2

### Project Module

### Splunk Module

Ingestion Module: Splunk Module Responsibility 2

### Version Control Module

## Data

### Enforcement Action Report

Database Module:1 & 2

### Event Configuration

Database Module: 1 & 2

### Graph

DB Manager: 1 – 5

Node: 3

### Icon

DB Manager: 1 & 2

### Log File

Database Module: 3

### Node

Database Module: 1-3

### Relationship

DB Manager: 1 - 3

### Significant Log Entry

DB Manager: Significant Log Entry Responsibility 2,3,4

### Vector

Database Module: 1 & 2