Spice Girls

Prevent, Mitigate, and Recover (PMR) Insight Collective Knowledge (PICK) Software Design Document

Version 1.0

March 4, 2020

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

Initial Release:	Version 1.0
Current Release:	Version 1.0
Indicator of Last Page in Document:	$
Date of Last Review:	March 4, 2020
Date of Next Review:	March 14, 2020
Target Date for Next Update:	March 20, 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
Elsa Tai Ramirez
Jake Lasley

Customer:

Mr. Vincent Fonseca
Mr. Baltazar Santaella
Ms. Herandy Vasquez
Ms. Florencia Larsen
Dr. Oscar Perez
Mr. Erick De Nava

Software Team Members:

Ana Zepada
Dima AbdelJaber
Ricardo Sanchez
Luis Ochoa
Scott Honaker

Change Summary

The following table details changes made between versions of this document

Version	Date	Modifier	Description

1.0	3/4/2020	Spice Girls	Creation of Document

Table of Contents

7. Database Description

7.1. Data Schema

1. Introduction

1.1. Purpose and Intended Audience

The purpose of creating the software design document is to aid in the development of the design and structure of the system that the team will build. It gives guidance on the design. The SDD document shows how the system can be separated into components to simplify the implementation. The intended audience are the guidance team, the software engineering teams, and the clients: Mr. Vincent Fonseca, Mr. Baltazar Santaella, Ms. Herandy Vasquez, Ms. Florencia Larsen, Dr. Oscar Perez, and Mr. Erick De Nava.

1.2. Scope of Product

PICK shall be a tool used by the white team analysts in order to efficiently sort through documents pertaining to adversarial assessments. These include computer log files and screenshots. These documents are then used to piece together an attack log to analyze the way in which the blue team responds to the red team's attack. Without the tool, analysts are currently having to open up all the files that they wish to reference in their attack graphs. In addition, this system shall simplify the way in which data is filled for nodes in the attack graph. The ultimate goal of the system is to reduce the amount of time doing each analysis to approximately two weeks. LSH recognizes the complexity and the time it takes to
analyze the applicable logs, observation notes, and other artifacts gathered from an adversarial assessment from the red, blue, and white teams and generate a report that presents the events that took place during the adversarial assessment. They want a system that would aid their analysts in correlating red team's activities to blue team's responses and represent the events that took place during an adversarial assessment graphically. UTEP and LSH are collaborating to develop Prevent, Mitigate, and Recover (PMR) Insight Collective Knowledge System (PICK) that will provide the ability to correlate red team's activities to blue team's responses and graphically represent the events that took place during an
adversarial assessment.

1.3. References

[1] Tai Ramirez, Elsa, *Prevent, Mitigate, and Recover (PMR) Insight Collective Knowledge System (PICK)* [SRS] El Paso, TX: UTEP, 2020

1.4. Definitions, Acronyms, and Abbreviations

1.4.1. Definitions

Data Cleansing	Data cleansing is the removal of unwanted characters from uncleansed TMUX log file; removal of blank rows from uncleansed excel log file; and removal of blank lines from uncleansed log file.
Data Validation	Data validation is the process of inspecting data in the cleansed log files based on predefined data validation rules.
Log Entry	Splunk takes the validated log files and convert them into normalized data. The normalized data are called log entries. Users of the system can filter and edit log entries.
Significant Log Entry	A log entry selected by the user and associated with a vector. The attributes are the same as for a log entry. The system stores significant log entries. Splunk stores log entries in the normalized data files.
Timestamp	Denotes time in hours:minutes, date in month:date:year, and section in am/pm.
Significant log entry	Denotes a log entry that is associated to at least one vector.

1.4.2. Acronyms

UTEP	The University of Texas at El Paso
LSH	The Lethality, Survivability, and HSI Directorate
SDD	Software Design Document
PICK	Prevent, Mitigate, and Recover (PMR) Insight Collective Knowledge

1.4.3. Abbreviations

e.g	For example
i.e	That is

1.5. Overview

The document is divided into six sections. The first section gives a description of the overall system and how all the components relate to each-other. The following four sections are detailed descriptions of subsections of the system. Each of the detailed descriptions of subsystems gives the subsystem name, its general description and classes. It also goes into describing the subsystem's responsibilities and contracts. The database section shows the relational diagram of the database as well as the schema for the database.

2. Decomposition Description

2.1. System Collaboration Diagram

The PICK System will be divided as follows:

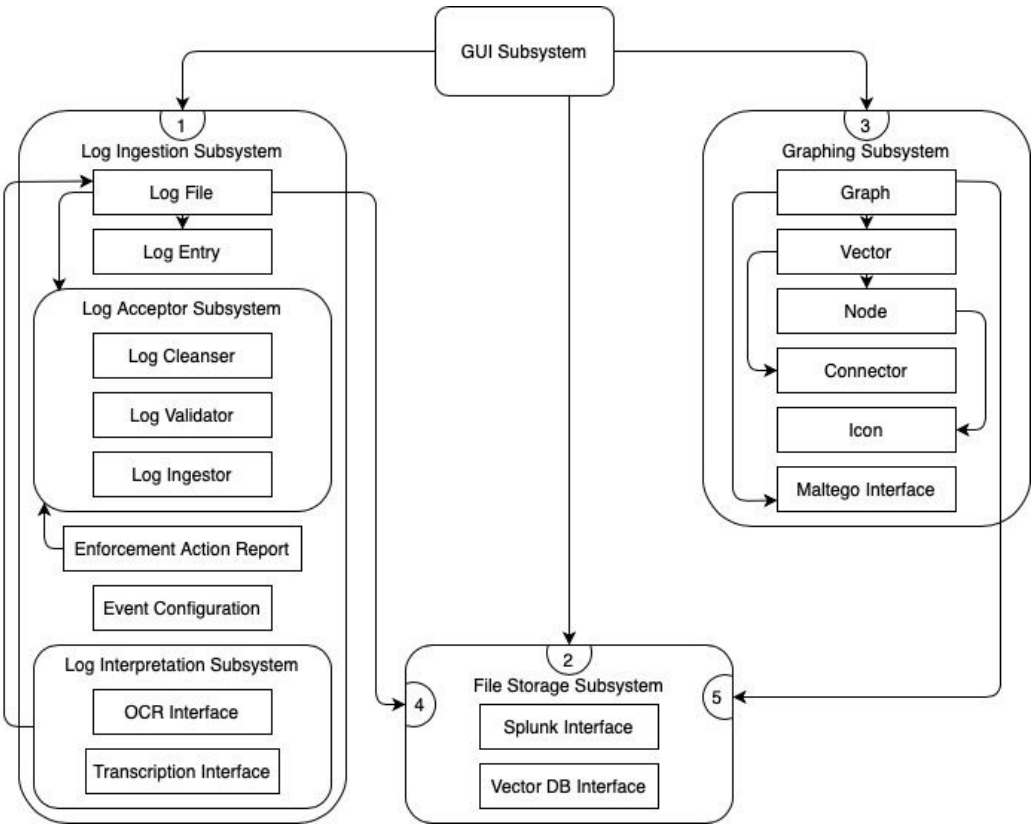


Figure 1: System Collaboration Diagram

2.2. Subsystem and Component Descriptions

The following section will describe the subsystems of the system and the classes they contain.

2.2.1. User Interface Subsystem

The GUI subsystem will handle any input and output of the system. It will allow the user to alter the system through prompts.

The class of the subsystem is:

- User Interface

The contracts of the subsystem include:

- Graph Interaction
- Vector Interaction

- Node Interaction
- Icon Interaction
- Event Creation
- Data Storage Interaction

2.2.2. Graphing Subsystem

The classes of the subsystem include:
- Graph
- Vector
- Node
- Icon
- Connector
- Maltego Interface

The contracts of the subsystem include:
- Graph Settings
- Know Vector Components
- Change Vector Components
- Know Node Details
- Change Node Details
- Know Icon Components
- Know Connector Components
- Change Connector Components
- Implement Maltego

2.2.3. File Storage Subsystem

The file storage subsystem has interfaces to the vector database and to Splunk.

The classes for the subsystem include:
- Splunk Interface
- Vector DB Interface

The contracts of the subsystem include:
- Implement Splunk
- DB Interaction

2.2.4. Log Ingestion Subsystem

The log ingestion subsystem allows the user to create an event. It will allow the user to designate directories, access log files, interpret the log files, and split the log files into log entries.

The classes for the subsystem include:
- Log File
- Log Entry
- Log Acceptor Subsystem
    - Log Cleanser
    - Log Validator

    - Log Ingestor
- Enforcement Action Report
- Event Configuration
- Interpretation Subsystem
    - OCR Interface
    - Transcription Interface

2.3. Dependencies

The Graphing Subsystem and Log Ingestion Subsystem will depend on the GUI subsystem. File Storage Subsystem will depend on the The Log Ingestion Subsystem. Log File and Enforcement Action Report both depend on the Log Acceptor Subsystem. Log Entry depends on Log File. Graph depends on Maltego Interface and Vector. Vector depends on Node and Connectors. Node depends on Icon.

These dependencies mean that the GUI should be the first thing to be developed and the File Storage Subsystem should be the last thing to be developed. Log File should be developed before Log Entry.

3. Detailed Description User Interaction Subsystem

3.1. Component Description

Component name: User Interaction Subsystem
Purpose: To allow the user to input information into the system and to view the state of the system.
Classes: User Interface

3.2. Class Description: User Interface

<table>
<tr><td colspan="2">Class: User Interface</td></tr>
<tr><td colspan="2">Superclass:</td></tr>
<tr><td colspan="2">Subclasses:</td></tr>
<tr><td colspan="2">Private Responsibilities:</td></tr>
<tr><td colspan="2">Contract 1: Graph Interaction</td></tr>
<tr><td>Responsibilities</td><td>Collaborations</td></tr>
<tr><td>1. Prompt to show/hide node
2. Prompt to change icon
3. Prompt to show/hide node name
4. Prompt to show/hide node id
5. Prompt to show/hide node description
6. Prompt to show/hide node timestamp
7. Prompt to show/hide node description
8. Prompt to show/hide log entry reference
9. Prompt to show/hide log creator
10. Prompt to show/hide event type
11. Prompt to show/hide icon type
12. Prompt to show/hide source
13. Prompt to change orientation
14. Prompt to change interval units
15. Prompt to change interval
16. Display attack graph
17. Display timeline
18. Display table
19. Export CSV of the graph
20. Export PDF of the graph</td><td>Graph (7) Vector (8) Node (10) Icon (12) Connector (14) Maltego Interface (16)</td></tr>
<tr><td colspan="2">Contract 2: Vector Interaction</td></tr>
<tr><td>Responsibilities</td><td>Collaborations</td></tr>
<tr><td>21. Prompt to add node from vector
22. Prompt to delete node from vector</td><td>Vector (9) Connector (15) Splunk Interface (17) Log Entry (20)</td></tr>
</table>

23. Prompt to filter through log entries 24. Prompt to search through log entries 25. Prompt to change vector name 26. Prompt to change vector description 27. Prompt to delete vector 28. Prompt to change connector name 29. Prompt to change connector parent node 30. Prompt to change connector child node 31. Prompt to add connector 32. Prompt to delete connector 33. Prompt to add vector name 34. Prompt to add vector description	
Contract 3: Node Interaction	
Responsibilities	**Collaborations**
35. Prompt user to create node from log file 36. Prompt user to create blank node 37. Prompt user to change node name 38. Prompt user to change node description 39. Prompt user to change node timestamp 40. Prompt user to change node source 41. Prompt user to create node 42. Prompt user to delete node	Node (11)
Contract 4: Icon Interaction	
Responsibilities	**Collaborations**
43. Prompt user to create icon 44. Prompt user to delete icone 45. Prompt user to change icon name 46. Prompt user to change icon path	Icon (13)
Contract 5: Event Creation	
Responsibilities	**Collaborations**
47. Prompt user to name event 48. Prompt user to add event description 49. Prompt user to select time range 50. Prompt user to select root directory 51. Prompt user to select blue team folder 52. Prompt user to select red team folder 53. Prompt user to select white team folder	Event Configuration (26, 27)
Contract 6: Data Storage Interaction	
Responsibilities	**Collaborations**
54. Push changes to vector database 55. Pull changes from vector database	Vector DB Interface (18)

4. Detailed Description Graphing Subsystem

4.1. Component Description

Component Name: Graphing Subsystem
Purpose: Knows about the graph and its components
Classes: Graph, Maltego Interface, Vector, Node, Connectors, Icon

4.2. Class Description: Graph

Class: Graph	
Superclass:	
Subclasses:	
Private Responsibilities:	
Contract 7: Graph Settings	
Responsibilities	**Collaborations**
1. Know node visibility 2. Know name visibility 3. Know id visibility 4. Know description visibility 5. Know node timestamp 6. Know orientation 7. Know interval units 8. Know interval 9. Know log entry visibility 10. Know log creator visibility 11. Know event type visibility 12. Know icon type visibility 13. Know source visibilityChange node visibility 14. Change name for nodes 15. Change id visibility 16. Change description visibility 17. Change node timestamp 18. Change orientation 19. Change interval units 20. Change interval 21. Change log entry visibility 22. Change log creator visibility 23. Change event type visibility 24. Change icon type visibility 25. Change source visibility	

4.3. Class Description: Vector

Class: Vector	
Superclass:	
Subclasses:	
Private Responsibilities:	
Contract 8: Know Vector Components	
Responsibilities	**Collaborations**
1. Know vector name 2. Know vector time range 3. Know vector description 4. Know nodes belonging to vector 5. Know connectors belonging to vector	
Contract 9: Change Vector Components	
Responsibilities	**Collaborations**
6. Change vector name 7. Change vector time range 8. Change vector description 9. Add nodes 10. Delete nodes 11. Add connectors 12. Delete connectors 13. Change connectors	

4.4. Class Description: Nodes

Class: Nodes	
Superclass:	
Subclasses:	
Private Responsibilities: Knows the next node number in sequence, knows the node name, knows the node id, knows the node timestamp, knows the node's related file path (if any)	
Contract 10: Know Node Details	
Responsibilities	**Collaborations**
1. Provides node name nodeName(int nodeID) returns String nodeName	

pre: nodeID must have been assigned to a node **post:** String nodeName corresponding to nodeID is returned 2. Provides node description nodeDescription(int nodeID) returns String nodeDescription **pre:** nodeID must have been assigned to a node **post:** String nodeDescription corresponding to nodeID is returned 3. Provides node timestamp nodeTimestamp(int nodeID) returns Timestamp nodeTime **pre:** nodeID must have been assigned to a node **post:** Timestamp nodeTime corresponding to nodeID is returned 4. Provides related log file logFilePath(int nodeID) returns String filePath **pre:** nodeID must have been assigned to a node **post:** String filePath corresponding to nodeID is returned, null is returned if none exists	
Contract 11: Change Node Details	
Responsibilities	**Collaborations**
5. Change node visibility changeVisibility(boolean switch) **pre:** none **post:** the node becomes/stays visible if switch is true, the node becomes/stays invisible if switch is false 6. Change icons for nodes changeIcon(String name) **pre:** name must be one of the names of icons already stored **post:** the icon for the node changes to match the icon with the given name 7. Change name for nodes changeName(String name) **pre:** none **post:** the name for the node changes to match the name provided 8. Change id visibility changeIDVisibility(boolean switch) **pre:** none **post:** the node id becomes/stays visible if	

switch is true, the node id becomes/stays invisible if switch is false 9. Change description visibility changeDescriptionVisibility(boolean switch) **pre:** none **post:** the node description becomes/stays visible if switch is true, the node description becomes/stays invisible if switch is false 10. Change node timestamp changeTimestamp(Timestamp time) **pre:** time must be a valid Timestamp **post:** the time for the node changes to match the time provided 11. Create node Node(String name) **pre:** none **post:** log file created with node id being the next number in the sequence, timestamp being 00:00 00:00:0000, description left blank, and name as provided Node(String name, LogFile file) **pre:** log file must be valid **post:** log file created with node id being the next number in the sequence, timestamp of log file, description of log file and name as provided 12. Delete node deleteNode() **pre:** none **post:** only given node deleted	

4.5. Class Description: Icon

Class: Icon	
Superclass:	
Subclasses:	
Private Responsibilities:	
Contract 12: Know Icon Components	
Responsibilities	**Collaborations**
1. Know icon name 2. Know icon path	
Contract 13: Change Icon Components	

Responsibilities	Collaborations
3. Create icon 4. Delete icon 5. Change icon name 6. Change icon path	

4.6. Class Description: Connector

Class: Connector	
Superclass:	
Subclasses:	
Private Responsibilities:	
Contract 14: Know Connector Components	
Responsibilities	**Collaborations**
1. Know connection name 2. Know parent node 3. Know child node	
Contract 15: Change Connector Components	
Responsibilities	**Collaborations**
4. Create connection 5. Change connection name 6. Change parent node 7. Change child node 8. Delete connection	

4.7. Class Description: Maltego Interface

Class: Maltego Interface	
Superclass:	
Subclasses:	
Private Responsibilities:	
Contract 16: Implement Maltego	
Responsibilities	**Collaborations**
1. Implement Maltego	

5. Detailed Description File Storage Subsystem

5.1. Component Description

Component name: File Storage Subsystem
Purpose: Persistently stores changes made to vectors, nodes, connectors, icons and graphs.
Classes: Splunk Interface, Vector DB Interface

5.2. Class Description: Splunk Interface

Class: Splunk Interface	
Superclass:	
Subclasses:	
Private Responsibilities:	
Contract 17: Implement Splunk	
Responsibilities	**Collaborations**
1. Pull log files from Splunk 2. Export items to Splunk	

5.3. Class Description: Vector DB Interface

Class: Vector DB Interface	
Superclass:	
Subclasses:	
Private Responsibilities:	
Contract 18: DB Interaction	
Responsibilities	**Collaborations**
1. Pull updates to vectors and components from DB 2. Push updates to vectors and components from DB	

6. Detailed Description Log Ingestion Subsystem

6.1. Component Description

Component name: Log Ingestion Subsystem
Purpose: Deals with the initial input of files into the system
Classes: Log File, Log Entry, Log Cleanser, Log Validator, Log Ingestor, Enforcement Action Report, Evet Configuration, OCR Interface, Transcription Interface

6.2. Class Description: Log File

Class: Log File	
Superclass:	
Subclasses:	
Private Responsibilities:	
Contract 19: Know File Attributes	
Responsibilities	**Collaborations**
1. Know log file path 2. Know log file contents 3. Know cleansing status 4. Know validation status 5. Know ingestion status	Splunk Interface (17) Log Cleanser (22) Log Validator (23) Log Ingestor (24) OCR Interface (28) Transcription Interface (29)

6.3. Class Description: Log Entry

Class: Log Entry	
Superclass:	
Subclasses:	
Private Responsibilities:	
Contract 20: Know Entry Attributes	
Responsibilities	**Collaborations**
1. Know log file path 2. Know timestamp 3. Know log entry content 4. Know source	
Contract 21: Create Entry	

Responsibilities	Collaborations
5. Divide log file	Log File (19)

6.4. Class Description: Log Cleanser

Class: Log Cleanser	
Superclass:	
Subclasses:	
Private Responsibilities:	
Contract 22: Cleanse Logs	
Responsibilities	**Collaborations**
1. Remove empty rows and columns 2. Change cleansed status	

6.5. Class Description: Log Validator

Class: Log Validator	
Superclass:	
Subclasses:	
Private Responsibilities:	
Contract 23: Validate Logs	
Responsibilities	**Collaborations**
1. Check if log is in a given time range 2. Change validated status 3. Identify failed logs	

6.6. Class Description: Log Ingestor

Class: Log Ingestor
Superclass:
Subclasses:

Private Responsibilities:	
Contract 24: Ingest Logs	
Responsibilities	**Collaborations**
1. Take files from splunk and into the system 2. Change ingested status	

6.7. Class Description: Enforcement Action Report

Class: Enforcement Action Report	
Superclass:	
Subclasses:	
Private Responsibilities:	
Contract 25: Know Failed Logs	
Responsibilities	**Collaborations**
1. Know logs that do pass the validation 2. Know logs that do not pass the validation	Log Validator (23)

6.8. Class Description: Event Configuration

Class: Event Configuration	
Superclass:	
Subclasses:	
Private Responsibilities:	
Contract 26: Know Event Attributes	
Responsibilities	**Collaborations**
1. Know event name 2. Know event description 3. Know event time range	
Contract 27: Change Event Description	
Responsibilities	**Collaborations**
4. Change event name 5. Change event description	

6. Change event time range	

6.9. Class Description: OCR Interface

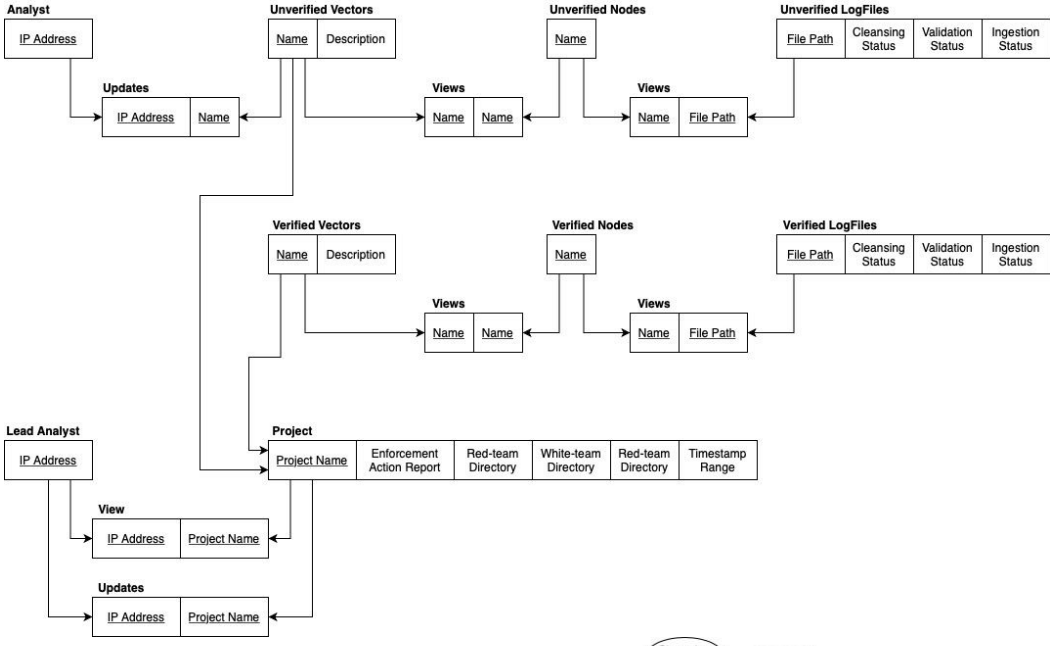
Class: OCR Interface	
Superclass:	
Subclasses:	
Private Responsibilities:	
Contract 28: Convert Visual Logs to Text Logs	
Responsibilities	**Collaborations**
9. Convert visual logs to text logs	

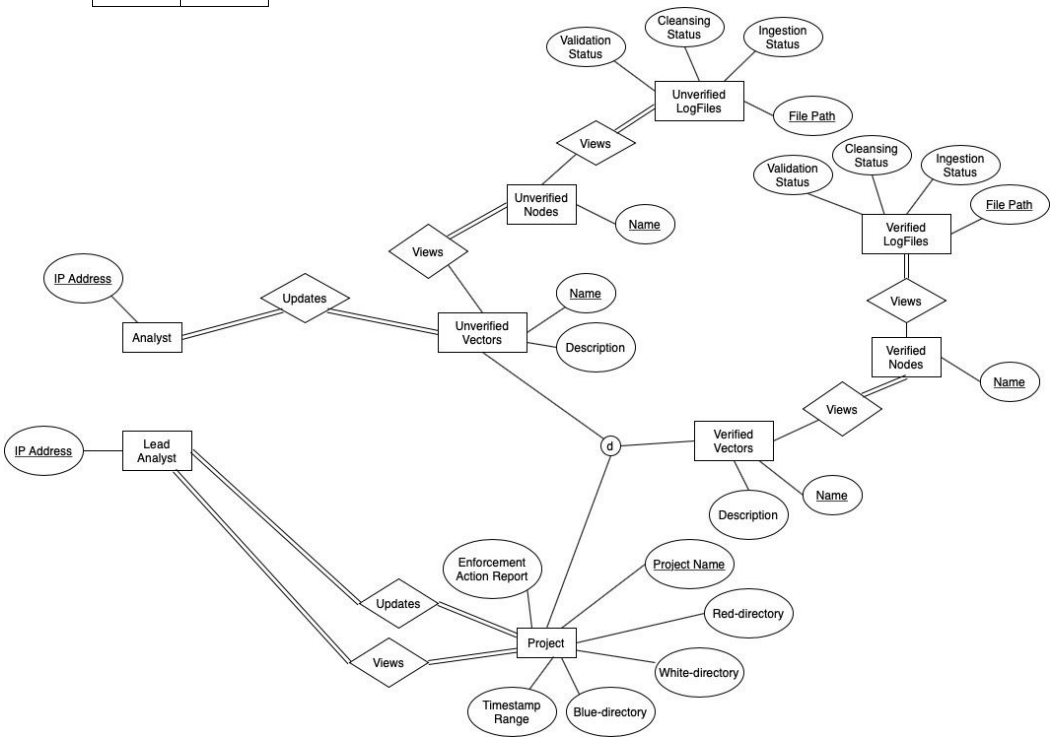
6.10. Class Description: Transcription Interface

Class: Transcription Interface	
Superclass:	
Subclasses:	
Private Responsibilities:	
Contract 29: Convert Audio Logs to Text Logs	
Responsibilities	**Collaborations**
10. Convert audio logs to text logs	

7. Database

7.1. Database Schema





$