**Team 12 Feathersoft CRC**

**Classes:**

SRS section 3.2.2 describes the following classes of related real-world objects in the system. Real world objects identify physical or conceptual components of the system:

* Event Configuration
* Log File
* Enforcement Action Report
* Vector
* Significant Log Entry
* Node
* Relationship
* Icon
* Graph
* Vector Manager
* Log Entries Manager
* User Manager
* User
* Transcription Handler
* Validation (Splink) Handler

|  |  |
| --- | --- |
| **Concrete Class: Vector Manager** | |
| **Responsibilities (Does):**   * Compare Vector DB differences between Lead and Non-Lead * Download Vector and Graph from Lead Vector DB to user’s Vector DB * Upload Vector and Graph from user’s Vector DB to Lead approval status page * Load Vector from Local Storage * Download copy (Saving) of Vector to Local Storage * Load history of changes made in the Vector DB | **Collaborators:**   * The Vector Manager class will collaborate with Event Configuration to gain access to its attributes * The Vector Manager class receives log entries attributes from Significant Log Entry class * The Vector Manager class receives Graph attributes from Graph class * The Vector Manager class receives Vector attributes from Vector class |

|  |  |
| --- | --- |
| **Concrete Class: Event Configuration** | |
| **Responsibilities (Knows):**   * Knows event name. * Knows event description. * Knows event start timestamp. * Knows event end timestamp. * Knows root directory. * Knows red team folder. * Knows white team folder. * Knows blue team folder. * Knows where the master vector DB is stored. * Knows lead’s IP address. * Knows the number of established connections to the host machine.   **Responsibilities (Does):**   * Check directory structure. | **Collaborators:**   * The Event Configuration is a client of the user manager class who tells it how many users are connected. |

|  |  |
| --- | --- |
| **Log File** | |
| **Responsibilities (Knows):**   * Knows the log file name. * Knows the cleansing status of a log file. * Knows the validation status of a log file. * Knows the ingestion status of a log file. * Knows the acknowledgement status of a log file. * Knows audio transcribed text in one- minute intervals. * Knows transcribed text from audio extracted from video in one-minute intervals. * Knows extracted text from images. * Knows extracted text from pdf files. * Knows start date, end date, start timestamp, and end timestamp. | **Collaborators**:   * A Log File class depends upon the directory attributes provided by the Event Configuration class |

|  |  |
| --- | --- |
| **Enforcement Action Report** | |
| **Responsibilities (Knows):**   * Knows where an error occurs in a log file. * Knows why a specific line in the log file failed the validation test. | **Collaborators**:   * The Enforcement Action Report receives the status attributes and transcriptions from the Log File class |

|  |  |
| --- | --- |
| **Vector** | |
| **Responsibilities (Knows):**   * Knows its name * Knows its description | **Collaborators**: |

|  |  |
| --- | --- |
| **Significant Log Entry** | |
| **Responsibilities (Knows):**   * Knows the log entry number. * Knows the timestamp. * Knows the log entry content. * Knows the name of the log file from which it originates. * Knows the path of the log file from which it originates. * Knows the host / IP address. * Knows the source type.   **Responsibilities (Does):**   * Associates a log entry to at least one vector. | **Collaborators:**   * The Significant Log Entry class uses in association the name attributes form the Vector class |

|  |  |
| --- | --- |
| **Node** | |
| **Responsibilities (Knows)**:   * Knows its ID * Knows its name * Knows its description * Knows its log entry reference * Knows its log creator * Knows its event type * Knows its icon type * Knows its source * Knows its visibility | **Collaborators**:   * The Node class selects log entry attributes from log entry class to categorize as a node * The Node class selects the type attribute from icon class to select a visual for the node |

|  |  |
| --- | --- |
| **Relationship** | |
| **Responsibilities (Knows)**:   * Know its relationship ID * Know its parent ID (source node of the relationship) * Knows its child ID. * Knows its label. | **Collaborators**:   * A relationship is a client of the node class because it makes references to the parent (source) and child (destination) nodes. |

|  |  |
| --- | --- |
| **Icon** | |
| **Responsibilities (Knows):**   * Knows the path to an image. * Knows the icon name.   **Responsibilities (Does):**   * Provides icon to the node class. | **Collaborators:** |

|  |  |
| --- | --- |
| **Graph** | |
| **Responsibilities (Does):**   * Adds nodes * Edit nodes * Delete nodes * Adds relationships * Edit relationships * Remove relationships * Change icons * Add icons * Move nodes * Delete icons * Handle errors when trying to delete relationships/nodes * Handles errors when trying to edit nodes/relationships * Filter nodes and relationships * Export an image of the graph * Undo changes * Save changes * Keep track of a change list   **Responsibilities (Knows)**   * Know Export format * Know Orientation * Know Interval units * Know Interval * Know Position of nodes * Know Position of relationships | **Collaborators**:   * Graph is a client of Node because it provides its attributes for display and modification purposes * Graph is a client of Relationship because the positions of the relationship will be used to place them on the graph. |

|  |  |
| --- | --- |
| **Log Entries Manager** | |
| **Responsibilities (Does):**   * Check for changes between original log file entries and log entries * Save updated log entries to the log files * Saves updated log entries to local storage. | **Collaborators:**   * The Log Entries Manager class selects the log files from Log File Class to compare any changes in their deltas |

|  |  |
| --- | --- |
| **User Manager** | |
| **Responsibilities (Does):**   * Responsible for managing the number of connected users. * Listens for incoming user connections to the server. * Updates event configuration if there is a change in the number of connected users. (E.g. A user closes the application). * Differentiates the IP addresses between Lead and Non - Lead User   **Responsibilities (Knows):**   * The current I.P. addresses connected to the system. * The number of users connected to the system. | **Collaborators:**   * The User Manager selects the IP address of users from the User class. |

|  |  |
| --- | --- |
| **User** | |
| **Responsibilities (Knows):**   * The IP address of the User * Knows if the User is a Lead for the | **Collaborators:** |

|  |  |
| --- | --- |
| **Validation (Splunk) Handler** | |
| **Responsibilities (Does):**   * Check that the log files contain a timestamp per line * Check that log file contain timestamps that are bounded by the start data, end date, start time, and end time specified in the event configuration. * If the log file is of type CVS and the originator of the log file is from the white team, check that the log file contains timestamps that are within Lower limit of the range and Upper limit of the range * Certify log files as validated log file. * Change status of log file to pass or fail. * Create log files with same source of ingested log files * Cleanse Log Files from unwanted characters if log file is of type TMUX * Cleanse Log Files from blank rows if the log file is of type CVS * Create cleanse certification for Log Files * Create log files with same source of ingested log files * Cleanse Log Files from unwanted characters if log file is of type TMUX * Cleanse Log Files from blank rows if the log file is of type CVS * Create cleanse certification for Log Files | **Collaborators:**   * The Validation Handler class selects attributes of log files from the Log File class to check timestamps syntax and the presence of a timestamp. Update Log Files where changes were made in the cleansing process * The Validation Handler class selects timestamp requirements from the Event Configuration class to select logs the correspond to that timeframe |

|  |  |
| --- | --- |
| **Transcription Handler** | |
| **Responsibilities (Does):**   * Transcribe audio Log Files into text * Translate Image Log Files into text * Transcribe Video Log Files into text | **Collaborators:**   * The Ingestion class the audio, image, and video files from the Log File class to convert into text Log Files. |