**Use Case Scenario Name**: Modify Vector DB

**Description:** Workflow an analyst goes through to modify the global vector DB that includes log file ingestion, log entry association, and local vector DB modification.

**Actors:** Analyst, Splunk, OCR Tool, Transcription Tool

**Pre-condition:**

1. The event configuration must be initialized.
2. The root directory must be initialized.

**Post-condition:**

1. The analyst’s push table is updated to reflect the pushed vector DB.
2. The server saves the pushed vector DB to a system data store.
3. The lead receives the pushed vector DB and is able to approve, reject, or view the changes

**Flow of events:**

Step 1: The analyst starts the ingestion process.

Step 2: The system cleans all log files in the set root directory.

Step 3: The system validates all log files, creating enforcement action reports where appropriate.

Step 3: The system sends all validated textual log files to Splunk.

Step 4: Splunk ingests the textual log files as log entries.

Step 5: The system collects all the log entries parsed using Splunk and stores them in a system data store.

Step 6: The OCR tool ingests image log files as log entries.  
Step 7: The system collects all the log entries from the OCR tool and stores them in a system data store.

Step 8: Transcription tool ingests video/audio log files as log entries.  
Step 9: The system collects all the log entries from the transcription tool and stores them in a system data store.

Step 10: The system informs the analyst that all log files have been ingested.

Step 11: The analyst creates a vector.

Step 12: The system stores the vector in a local data store.

Step 13: The analyst requests log entries by providing search criteria to the system

Step 14: The system provides the analyst with log entries that satisfy the search criteria.

Step 15: The analyst associates some of the provided log entries to the created vector.

Step 16: The analyst requests to inspect the created vector.

Step 17: The system displays a graphical and tabular representation of the vector.

Step 18: The analyst modifies the nodes created from the associations.

Step 19: The system updates the graphical and tabular representation of the vector.

Step 20: The analyst defines relationships between nodes.

Step 21: The system updates the graphical and tabular representation of the vector.

Step 22: The analyst modifies the relationships between nodes.

Step 23: The system updates the graphical and tabular representation of the vector.

Step 24: The analyst pushes the local vector DB.

Step 25: End of use case.

**Alt:** Step 1: The analyst chooses to not ingest log files.

Step 1.1: Use case continues at step 11.

**Alt:** Step 11: The analyst chooses to use an existing vector.

Step 11.1: The analyst requests log entries by providing search criteria to the system

Step 11.2: The system provides the analyst with log entries that satisfy the search criteria.

Step 11.3: The analyst associates some of the provided log entries to a chosen vector.

Step 11.4: The analyst requests to inspect the chosen vector.

Step 11.5: Use case continues at step 17.

**Alt:** Step 18: The analyst chooses not to modify the nodes.

Step 18.1: Use case continues at step 20.

**Alt:** Step 20: The analyst chooses not to define relationships between nodes.

Step 20.1: Use case continues at step 24.

**Alt:** Step 22: The analyst chooses not to modify relationships between nodes.

Step 22.1: Use case continues at step 24.

**Alt:** Step 24: The analyst chooses not to push the local vector DB.

Step 24.1: End of use case.