Team: T10 TeamWork

Assignment: SDD draft.

Due Date: 03/03/2020

Professor: Dr. Roach

**System Design Document (SDD) Draft**

List of identified subsystems, components, and classes:

* Validation & Ingestion
  + Validation
    - Script handler
    - Log file
    - ValidationIngestionWindow
    - Enforcement Action Report
    - DirDialog
    - Splunk
  + Ingestion
    - ValidationIngestionWindow
    - LogEntry
    - SQL Light
* Project Initialization
  + Configuration
    - ProjectConfigWindow
    - DirDialog
    - CalendarDialog
    - Event Configuration
  + Startup
    - Main
* Navigation
  + Regex
    - Filter
    - Log Entry
    - NavigatorWindow
* Graph Creation
  + Vector Node Modification
    - Vector
    - Node
    - Relationship
    - VectorTableWindow
  + Graph Configuration
    - Graph
    - GraphWindow
    - Node
  + Graph Export
    - Graph

**Detailed Description of Component Validation**

**Component Description**

The validation component is in charge of validating all the log file data to ensure it is ready to continue to the ingestion process. It runs a cleansing script on the log files and checks if they meet the requirements specified by the user. Any encountered errors are shown to the user for the user to address. The validation component utilizes the Script handler, Log file, ValidationIngestionWindow, Enforcement Action Report, DirDialog, and Splunk.

**Class Description Script Handler**

The Script Handler class represents the cleansing script used on the log files during the validation process. It has the responsibility of knowing the cleansing script directory path to be able to run it.

**Contract Run Script**

Contract number: 1

Contract name: run script

Contract description: runs the cleansing script on the given log file.

Protocol 1:

Signature: runScript(filePath)

input filePath- log file directory path to run script on, type string.

output - none

Pre-Cond: the given filePath is the correct directory path and exists.

Post-Cond: the cleansing script is executed on given log file.

**Class Description Log File**

The Log File represents a file of any type, for example, csv, jpeg, pdf, etc., that have been ingested by the system. It has the responsibility of knowing its own file type, knows ingestion errors, and know all its own content.

**Contract Log Ingestion**

Contract number: 2

Contract name: Log Ingestion

Contract description: Given a log file, the class will return the given log type, the log content and if there are any ingestion errors.

Protocol 1:

Signature: logIngest(anyTypeLog)

input anyTypeLog - Log file of any type.

output fileType - log file type (i.e. csv, jpeg, pdf, etc…)

output logContent - data information from the log file.

output ingestionErrors - information of an error if it exists.

PreCond: Log file has to exist.

Post-Cond: Log file stays unchanged.

**Class Description ValidationIngestionWindow**

The ValidationIngestionWindow class is the visual representation of the handler of the validation and ingestion process of the log file data. The superclass will be ProjectConfigWindow, and will handle the transition to NavigationWindow.

**Contract Ingest Data Selection**

Contract number: 3

Name: Ingest Data Selection

Contract description:

Protocol 1:

Signature: ingestDataSelection(validLogFile)

input validLogFile- files that are validated.

output: none

Pre-Condn: the given LogFile has to be previously validated.

Post-Condition: LogFiles are now LogEntries.

**Class Description Enforcement Action Report**

The Enforcement Action Report class represents the report that will be generated with each invalidated log file. This class will include the error(s) associated with the log file that makes it invalidated, and thus convey to the analysts the reasons as to why a log file is invalidated.

**Contract Create Error Report**

Contract number: 4

Contract name: Create error report

Contract description: Creates an enforcement action report at the detection of an error in a log file.

Protocol: 1

Signature: createErrorReport(logFileInfo, errorType)

input logFileInfo: Log file information to display in the enforcement action report, so that the analyst can know which log file is invalidated and must be changed or approved as is.

input errorType: The error type(s) associated with the invalidated log file. The error type(s) enables the analyst to see which validation rules the associated log file broke.

output: Enforcement Action Report

Pre-Cond: Log file must not be null and error type must be a valid error for generating an enforcement action report.

Post-Cond: Log file must remain unchanged.

**Class Description DirDialog**

The DirDialog class will handle the directory choosing function whenever the software requires it. The superclass will be ProjectConfigWindow as that is where all the information will be returning to once a directory has been chosen.

**Contract Directory Chooser**

Contract number: 5

Contract name: Directory Chooser

Contract description: Display a file chooser window that will select the directory for log data.

Protocol 1:

Signature: dir\_dialog(self, dialogtitle)

self-identify method as an instance of the class DirDialog

dialogtitle- name for the dialog that will be created.

output - none

Pre-Cond: a dialogtitle has been given

Post-Cond: the instance saves the file directory path