Software Requirements Specification

for

Log Analysis Framework

Version 1.1 approved

Prepared by Alex King, Rami Dyab, Adisesh Viruru, Matthew Zhang

Team 21

CSCI 5801

02/28/2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Team 21	2/14	Initial Version	1.0
Team 21	2/28	Requirements Specification Review: CSCI 5801 - Requirements Review Report	1.1

1. Introduction

1.1 Purpose

This document provides a detailed description of the Log Analysis Framework along with its requirements, features, constraints, and general operations. The purpose of the Logfile analysis framework (Loafr) is to improve the process of reviewing and analyzing log data generated by the client's testing platform. The Loafr is not software designed for *testing* products, rather it's software designed for analyzing the *results* of those product tests.

1.2 Document Conventions

Section headers are bolded with size 18 font. These section headers refer to an overall description of what the section will contain. Section headers are numbered 1-6. Subsection headings are bolded as well with size 14 font. The headers describe what we will be discussing in further detail in that subsection. Subsection headings are numbered corresponding to the section they are included in.

In section 4 of this document, we discuss functional requirements the system shall include. These requirements are grouped in order of use cases and are labeled by REQ-#. Priorities for high-level requirements are inherited by detailed requirements.

1.3 Intended Audience and Reading Suggestions

This document will mainly be intended for testing and analysis teams of the organization. From this, the article will contain thorough information regarding the functionality, features, as well as requirements for the given product. The suggested sequence of reading is in chronological order as the article provides an overview of the system followed by requirements for use and implementation.

1.4 Product Scope

The software system is meant to reduce the client's high testing costs by optimizing the 'review and analyzing' part of the process. The software will act as a tool for system engineers to analyze the test data more efficiently and with fewer errors. The software will also be able to adapt to different tests and types of data, independently.

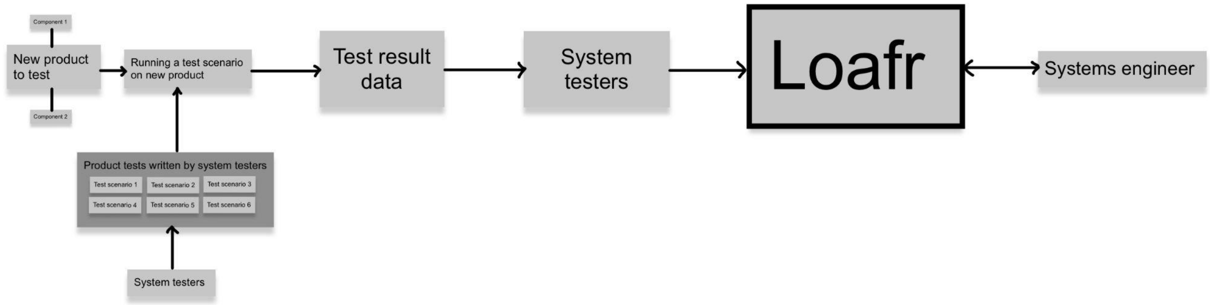
1.5 References

This document refers to and is related to the "CSCI 5801 002 S22 Project Loafr.pdf" that has been posted to the Canvas course site.

2. Overall Description

2.1 Product Perspective

Loafr is to replace the current data analysis system which has conflicting formats and requires too much time for system engineers to use effectively. The success of the Loafr software is contingent upon the accuracy of the data it receives from the product tests themselves. For this reason, it is a component at the end of a larger system as the diagram below shows:



2.2 Product Functions

Loafr shall have the following functions:

FUNC-1: Allow the user to perform advanced search, filter, and sort operations on the test data. For example, the user could perform those operations by a variety of variables in the data set (minimum values, maximum values, specified range, etc.)

FUNC-2: The system shall have a flexible way to add new data types and data sources so the system can adapt to the testing of different products.

FUNC-3: The system shall allow the user to analyze data within a reasonable amount of time (user will not need to wait more than 8-12 seconds on average for results).

FUNC-4: The system shall have 1 standardized data logging format

FUNC-5: The system shall have a responsive interactive mode where the user can view and interact with logged data.

FUNC-6: The system shall have a batch processing mode where data can be logged either by input files from the user, or automated scripts run by the testing system. Loafr shall still be functional while in batch processing mode.

FUNC-7: The system shall work on typical engineering desktops and laptops (Windows OS).

2.3 User Classes and Characteristics

The most important user class consists of the testing and analysis team. This represents the most privileged level because this system is designed with ease of use as a main priority. Furthermore, this specific user class will be utilizing the service the most frequently. Therefore, the testing and analysis team will be given access to all features and functionality of the system.

2.4 Operating Environment

Loafr shall run in a typical engineering environment. This includes running the system on a Windows operating system using a desktop or a laptop. Loafr shall be available as a downloadable application that will be supported on the Windows OS (Windows 10 or Later). The system engineers who will be using this system are required to operate using the typical engineering environment described. This requirement has been implemented because Windows 10 is currently seen as a minimum industry standard. Therefore, support on older versions will not be provided. *(FUNC-7)*

2.5 Design and Implementation Constraints

One of the limitations that are present is the amount of data that can be processed at a given point in time. The system cannot have an infinite amount of memory. Therefore, a limit to the amount of input data will be put in place.. However, limiting the amount of data that can be processed will in turn allow the application to optimize the given input. Therefore, developers will need to find the perfect balance between the amount of data to be inputted and the processing speed and capabilities of the system.

Another limitation involves the operating system that this application will be deployed on. As of now, Windows and Linux are the primary operating systems for the software. However, later updates may see compatibility for MacOS to extend the usability and flexibility of the application.

2.6 User Documentation

The user documentation will consist of a variety of components to ensure that the product is received in a professional and efficient manner. This includes a detailed description of what the product is, manuals on how to install and use the software, troubleshooting basic issues, and support contact information for issues that cannot be resolved by the user. All of these components will be given to the user as a hard copy with digital versions being available as well. The main goal for documentation is to allow the user to be self-sufficient in setting up and utilizing the software.

2.7 Assumptions and Dependencies

One assumption that we have made involves the operating systems used by the consumer. The application will not be designed to be compatible with older versions of a given operating system. For example, we will assume that the user is using at least Windows 10. We believe that this requirement is reasonable as Windows 10 is seen as an industry standard, especially as Windows 11 has recently been released and is not yet widespread. Furthermore, supporting these older versions will lead to unnecessary cost increases as these versions are not even used commonly. Therefore, the Loafr system will only have support for Windows 10 and newer.

Another assumption for Loafr is that data that is yet to be logged by the system, the files shall be formatted. This format shall be consistent in order to match the standardized format that is expected by the system. If input files are not formatted correctly, the system will not be able to recognize data and place logs into the correct output format. For example, zip files will not be accepted. In this scenario, the user will need to unzip the file and extract all supported files for input. This feature will lower costs of production as will as improve output quality because the system is expecting a certain type of standard input and will not need additional attributes to account for a variety of files.

3. External Interface Requirements

3.1 User Interfaces

Loafr shall include an interactive user interface that will allow the user to view and analyze data that has been logged by the system. In the interactive mode, the user will be able to select between different data types and sources to get different views of the logged data. The user will also be able to interact with the data by selecting between various sorting options and will be able to filter through various data types. The user interface shall also allow the user to view data via different visualized forms (graphs, histograms, timelines, etc). As user-friendliness is a top priority of the Loafr system, no rigorous training will be required to use the system. Users should not require more than a few hours, or the amount of time to read through and understand the instruction manuals and documentation, to understand, implement, and utilize the system.

3.2 Hardware Interfaces

The system, Loafr, shall be able to run on a desktop or laptop that are equipped with the Windows OS. The user shall be able to install the software to their machine and launch the program that is using a compatible version of Windows. Using the mouse and keyboard, the user will then be able to interact with the software and access the data that has been logged by the system. The machine must also contain a CPU that is fast enough to handle batch logging large amounts of data at a time without unexpected interruptions. Again, the user will not need to wait more than 8-12 seconds on average for output to be displayed. For this reason, we will require that the user uses a machine with at least a processor equivalent to a 7th generation Intel i5 processor as well as a minimum of 12 gigabytes of ram. Furthermore, as all data will be logged by the system, we will also require at least one terabyte of storage, preferably SSD.

3.3 Software Interfaces

Loafr shall be an executable application that will run on the Windows operating system. The operating system shall be a newer version of Windows (version 10 or later). Input consists of files being uploaded by the user, specifically pdf and txt. Meanwhile, output, such as log data as well as any updated data, will be saved to the device's hard drive as CSV files. This will again ensure the ability of the application to function without internet connection.

3.4 Communications Interfaces

As Loafr is an application that is to be installed locally onto a machine, it will not require an internet connection to operate. This seems reasonable as it allows users to access the application at any time. Furthermore output, and log data will be saved onto the local machine as files, in the form CSV. Since the data is saved locally, the user will have the option to easily send and receive data. Since these files may contain sensitive information, downloaded files from the application will be encrypted by Loafr itself. This ensures that although they can be transferred from machine to machine, files cannot be viewed freely unless uploaded to Loafr. In order to upload a downloaded Loafr file back into Loafr, the file must first be converted to pdf or txt. Refer to section 3.2 for hardware requirements in order for the Loafr system to take files as input, operate on them, and then download output to the machine's local drive.

4. System Features

4.1 Sorting

4.1.1 Description And Priority

Loafr shall allow the user to sort data that has been logged by the system. This includes sorting by different data types (categories) and sorting by different scales. This is a High priority feature. Category sorting will depend on the type of data (number, string) that is being stored. Examples of sorting include numerical and alphabetical order. Users will also have the option to implement custom sorting procedures. For example, sort category y given the element in y has attribute x.

4.1.2 Stimulus/Response Sequence

- When the user is viewing data in Loafr's interactive mode, the system shall have an option to allow the user to view the data in various sorted formats. For example, when sorting quantitative categories, the user will be able to represent the data from greatest to least and vice versa.
- When the user is using batch processing mode. They can specify sorting ahead of time for the data to be displayed.
- Once the user selects their desired sorted view, the system shall sort the data in 10 seconds on average (other operations may take longer, the overall average wait time will not exceed 8-12 seconds).
- Once the data has been sorted, the system shall immediately display the sorted data to the interactive UI.

4.1.2 Functional Requirements

REQ-1: The system shall include a rapid sorting algorithm, such as quicksort, that will sort data in nlogn time complexity. *(FUNC-1)*

REQ-2: The system shall allow the user to select between various sorting categories. These categories shall include: *(FUNC-1)*

- Sorting by new to old data.
- Sorting by largest to smallest data (numerical).
- Sorting by most to least relevant data (user may define).
- Sorting by alphabetical order (if applicable).

(FUNC-1)

REQ-3: Once the user has selected their desired sorted view, Loafr shall sort the data in the background in 10 seconds on average. Once the data has been sorted, the system shall immediately display the sorted data to the interactive user interface. *(FUNC-1)*

REQ-4: An error message shall be displayed if the system sorting algorithm has failed. Common error messages will include invalid sorting types (alphabetical sort on numerical data) and incorrect input type (user accidentally included a string value in a category on integers). *(FUNC-1)*

4.2 Filtering

4.2.1 Description And Priority

Loafr shall allow the user to filter through data that has been logged. This would allow the user to select the type of data they would like to see and not display other data that is irrelevant to the user. This is a High priority feature.

4.2.2 Stimulus/Response Sequence

- When the user is viewing data in Loafr's interactive mode, the system shall have an option to allow the user to filter through data.
- Once the user selects the filter they would like to apply to the data, the system shall edit the data view to only show the data relevant to the selected filter. This operation will take on average 9 seconds.
- Once the data has been filtered, the system shall immediately display the filtered data to the interactive UI.

4.2.2 Functional Requirements

REQ-5: The system shall include a rapid filtering algorithm that will filter data in log(n) time complexity using a linear search pattern. *(FUNC-1)*

REQ-6: The system shall allow the user to filter between various categories. These categories shall include:

- Filtering by different data types and attributes
- Filtering by date added

- Filtering by value

(FUNC-1)

REQ-7: Once the user has selected their desired filtered view, Loafr shall filter the data in the background. Once the data has been filtered, the system shall display the filtered data to the interactive user interface. *(FUNC-1)*

REQ-8: An error message shall be displayed if the system filtering algorithm has failed. Common errors include invalid filter type (filter out all string values > 5). If a filter is applied and now changes are made to the data (no values satisfy filter), a message will appear alerting the user. *(FUNC-1)*

4.3 Searching

4.3.1 Description And Priority

Loafr shall allow the user to search data that has been logged by the system. This consists of the user being able to type keywords that correspond to data in the system. This is a High priority feature. This feature will be done in linear time complexity and will require on average 9 seconds.

4.3.2 Stimulus/Response Sequence

- When the user is viewing data in Loafr's interactive mode, the system shall have an option to allow the user to search for data by entering keywords, numbers, or dates that correspond to data the user wants to see.
- Once the user selects the search they would like to see, the system shall search for data that corresponds to and is related to the user search.
- Once the data has been searched and applicable data has been found, the system shall display the data to the interactive UI.
- If no data is found that matches the search, a notification will be displayed.

4.3.2 Functional Requirements

REQ-9: The system shall include a rapid search algorithm that will accurately locate data that is relevant to the user search (linear time, average wait time of 9 seconds). *(FUNC-1)*

REQ-10: The system shall allow the user to use a search bar to enter keywords and numbers to search for relevant data. *(FUNC-1)*

REQ-11: Once the user has selected their desired search view, Loafr shall search the data in the background. Loafr shall compare keywords to data to find relevant results while displaying a loading message to the user that the system is searching. Once relevant data has been located, it will display a list of the relevant data to the UI. *(FUNC-1)*

REQ-12: If no relevant data is found, the system shall display a message describing that no results were found. *(FUNC-1)*

REQ-13: An error message shall be displayed if the system search algorithm has failed. *(FUNC-1)*

4.4 Batch Processing

4.4.1 Description And Priority

Loafr shall have a running automatic script mode (batch processing) that will handle the data in much larger terms without the user constantly monitoring or putting inputs in. This would allow the user to find general trends over large amounts of data. This is a High priority feature.

4.4.2 Stimulus/Response Sequences

The user would be able to put the software in batch processing mode, select what data should be examined and what data should be reported back in the log. After the batch processing is completed, the data should be accessible using the interactive mode with the specified data reported.

4.4.3 Functional Requirements

REQ-14: Loafr shall include a batch processing mode in which the system will handle a large quantity of data (1 GB) at a time and format the data to be accessible by a user using the interactive mode. *(FUNC-6)*

REQ-15: Loafr shall periodically save data that has been logged (every 100 mb) while in batch processing mode. *(FUNC-6)*

REQ-16: If an error occurs, Loafr shall display a message to the UI and shall make a save on everything that is in the buffer stage of being saved (was getting ready to be saved, but not done yet) up to the point of error. *(FUNC-6)*

4.5 Interactive Mode

4.5.1 Description and Priority

Loafr shall have an Interactive mode that would allow the user to interact and view the data that has been logged by the system. In this mode, the user can view data trends, sort/filter through data, and compare data against other sets of data. This is a High priority feature.

4.5.2 Stimulus/Response Sequences

When the user launches the application, the user will have an option to view data in interactive mode. When the user selects interactive mode, the application will display a catalog of data files for the user to view. The system will respond to user commands and actions appropriately.

4.5.3 Functional Requirements

REQ-17: Loafr shall include an interactive mode that will present a UI which will allow the user to view, interact with, and analyze the data that has been logged by the system. *(FUNC-5)*

REQ-18: The interactive mode shall be a responsive system that will react immediately to user commands. However, the wait time on output from the system will depend largely on the function being used. For example, the user may need to wait 10 seconds when sorting data and 9 seconds when filtering data. *(FUNC-5)*

REQ-19: If an error occurs while in interactive mode, the system shall display a message. This message will be relevant to the operation being called, such as sorting, as well as the specific error, such as invalid sorting types. Loafr shall then safely exit the application without corrupting data that is stored by the system. *(FUNC-5)*

4.6 Standardize Logged Data

4.6.1 Description and Priority

Loafr shall have one format to standardize data. When data is being logged, the system must be able to take the input data and place them in a set location to allow data to be consistent and organized. The output data file type will be CSV. This is a High priority feature.

4.6.2 Stimulus/Response Sequences

When the user inputs data to be logged, Loafr will assume the input file is formatted correctly and consistent with other input files. The system will read data and format data to the standardized format. If the input file type does not match what Loafr expects (pdf, txt), an error message will appear.

4.6.3 Functional Requirements

REQ-20: When Loafr receives a file to input data, the system shall treat the input file as if it is properly formatted. The system shall read the data in the expected format and log data to the appropriate location. *(FUNC-4)*

REQ-21: When the user wants to view logged data, the system shall retrieve the data and display it as a standardized view. This data shall be retrieved from the database where all log files are saved and stored. When retrieving data, the system shall read data from the files and store them in memory for the system to access rapidly. *(FUNC-4)*

REQ-22: If an error occurs when logging data, the system shall save the data that has been logged and will display a message to the user. An error will also appear if an unsupported file type is passed as input. *(FUNC-4)*

4.7 Saving Logged Data

4.7.1 Description and Priority

Loafr shall have a system to save data that has been logged. Logged data shall be saved within CSV files and placed in appropriate folder locations that are specified by the user. This is a Medium priority feature.

4.7.2 Stimulus/Response Sequences

When the user inputs data to be logged, Loafr will save logged data on disk to allow the user to view data at different points in time. When the user wants to view data, the system will go to the appropriate location on disk and retrieve the data to be displayed. The csv file will need to be converted to either a pdf or txt file before uploading to Loafr.

4.7.3 Functional Requirements

REQ-23: When Loafr receives a file to input data, the system shall save data within CSV files in a folder location that is specified by the user prior to logging any data. *(FUNC-2)*

REQ-24: When the user wants to view logged data, the system shall retrieve the data from the appropriate output file from disk and display it to the user. *(FUNC-2)*

REQ-25: The system shall not overwrite any files that are logged and saved as CSV files. *(FUNC-2)*

REQ-26: If the user wishes to delete data, Loafr shall have the option to delete a file. The user shall have to confirm the removal of the file through the UI. Once the user confirms the removal, Loafr shall remove the appropriate CSV files from the system folder. *(FUNC-2)*

REQ-27: If an error occurs while retrieving a CSV file, Loafr shall display a message. *(FUNC-2)*

4.8 Time Analysis

4.8.1 Time-related analysis features

All data analysis capabilities shall have the option of using *time* as a criteria to further filter the data to the user's specifications. The user can specify a window of time to analyze and the system will exclude the data accordingly. This is ideal for analyzing events. This is a High priority feature. *(FUNC-1)*

4.8.2 Timeline default layout (Window of time)

The default layout for the user when displaying data shall be structured on a timeline to make the analysis more intuitive. This layout is also great for trimming the time frame of a search because the user can visually see the number of columns shrink as the time frame of their search shrinks. (See the glossary for a more detailed explanation of ‘window of time’). This is a High priority feature. *(FUNC-1)*

4.8.3 Excluded data columns

When the user only wants to view data points corresponding to a specified window of time, all the columns that are outside this time frame become discolored so the user gets a better sense of what data is being included and excluded in the results they're analyzing. This will help reduce human-error and produce more accurate interpretations of the test data. This is a Medium priority feature. *(FUNC-1)*

5. Other Nonfunctional Requirements

5.1 Performance Requirements

When Loafr runs with inputted files, the application will generate output in a reasonable time (on average 8-12 seconds). There will exist a correlation between the size of the file and the wait time for the output. However, there will also be a file size limit (up to 50 gigabytes) to ensure that the application will not be overloaded with data. As the user experience depends on the reliability as well as the speed of Loafr, the response time will be limited.

5.2 Safety Requirements

When generating output, Loafer may automatically install any new or updated data to the device's hard drive. By constantly creating logs of data, the user will never need to worry about losing information to system failures. Furthermore, in the event of a failure, the user can simply upload the most previous iteration of the log files and continue from there. When downloading files to the hard drive, Loafr will automatically encrypt the files to ensure that only they can only be viewed by an intended user. As customers of Loafr may be handling highly sensitive information, it is therefore critical that the information does not become compromised.

5.3 Security Requirements

As stated previously, Loafr customers may need to handle private information. Therefore, when downloading files off of Loafr, the contents of those files will be encrypted so that they are not exposed on the hard drive. Furthermore, an authentication process will be implemented in order for users to access the application. This may exclude multiple levels of security protocols to ensure that the user has access. For example, users may first need to log in with their credentials and then verify their login with a keycode.

5.4 Software Quality Attributes

Adaptability will ensure that the results of the system on the same data input will always remain the same. Different machines running the same operations on the same files will not produce different outcomes. Furthermore, the application will be able to be run extensively on end (support nonstop use for 12 hours) with a guarantee that data will not be lost or compromised.

All aspects of the system will be testable to ensure that customers are receiving a product that corresponds to their specific needs. All listed requirements will be tested extensively, especially with edge cases, to ensure that the system functions as intended.

5.5 Business Rules

Individuals who will operate this system are the data engineers. They will be able to access all log files and interact with the system fully. Due to this being the only users, we can assume that they have a familiarity as well as previous exposure to data analytics.

6. Other Requirements

Refer to Appendix C.

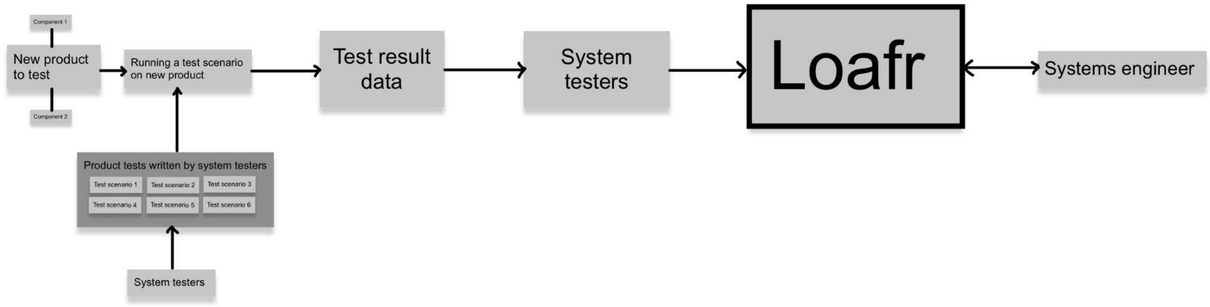
Appendix A: Glossary

Loafr: Log Analysis Framework

Window of time: A subset of the full test scenarios run time. For example, if test-A was a total of 10 seconds long, then *from 4 seconds to 7 seconds* would be a 'window of time' for test-A. From 1 second to 2 seconds is also a valid window of time. From 5 seconds to 12 seconds is not a valid window of time. Once the user specifies the window of time they want, then all the future data queries will only consider the data within that window of time when making calculations.

Appendix B: Analysis Models

System diagram: (Section 2.1)



P2 Diagrams: CSCI 5801 Loafr Models

Appendix C: To Be Determined List

Other requirements are expected to be added in subsequent versions.

Use-Cases:

Primary user: System Engineer
All the use-cases will be the system engineer prompting the Loafr user interface for a data query. The user interface will then perform the necessary operations on the test data and then return the result to the system engineer via the same user interface.

Use Case 1: Search Query
Laofr will allow the user to search for a specified variable in the input data. This could include an element with a certain key value or multiple numeric elements within a given range. A characteristic of searching is that the user can perform searches using any characteristic of the data. These searches can be as general or specific as the user requires.

Use Case 2: Filter Query
Similar to search queries, filter queries will allow the user to apply this operation to any attribute of the input data. Again, this could be general and include ranges of data, or specific so that only data with a specific characteristic remains. Most importantly, the user will have the ability and freedom to filter any relevant information pertaining to the input data.

Use Case 3: Sort Query
Sort queries will be available to attributes that are sortable. For example, if a category of data consists of true or false variables, then data will not be able to be sorted in terms of that category. Users will have the ability to sort numbers, such as in increasing order, or strings, such as alphabetical order. Furthermore, all sorting algorithms will be done in nlogn time to ensure that the operation will be completed as efficiently as possible.

It's important to note that all three queries can be used in conjunction with each other. For example, the user can filter out any elements with a specific attribute value, search from the remaining elements with a different attribute value, and then sort those remaining elements by yet another attribute. The primary goal of Loafr is for the user to have freedom in analyzing and representing input data.