**Requirements Analysis Document**

Data Processing Calculator

CSCI 4712 Senior Capstone

Spring 2022

Augusta University

Augusta, GA

Date: 4/11/2022

Version 1

Team Members

Sebastian Gonzalez

AJ Chiong

Roshni Patel

Daisha Norman

Henry Holt

**Table of Contents**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | INTRODUCTION……………………………………………............. | | | | | 3 |
|  | 1.1 | | | Scope of System………………………………………………............. | 3 |  |
| 2 | REQUIREMENTS OF SYSTEM…………………………………...... | | | | | 4 |
|  | 2.1 | | | Functional Requirements……………………………………………... | 4 |  |
|  | 2.2 | | | Non-Functional Requirements………….…………………………….. | 5 |  |
|  | 2.3 | | | Use Cases……………………………………………………………... | 6 |  |
|  | 2.4 | | | Use Case Descriptions………………………………………………... | 7 |  |
| 3 | USER INTERFACE MOCKUPS……………………………….......... | | | | | 11 |
|  | 3.1 | | Main Menu…………………………………………………. | | 11 |  |
|  | 3.2 | | UploadData Page……………………………………………………… | | 11 |  |
|  | 3.2 | | QueryData Page………………………………………………………. | | 12 |  |
| 4 | SOFTWARE ARCHITECTURE………………………………........... | | | | | 13 |
|  | 4.1 | | Subsystem Decomposition…………………………………………….. | | 13 |  |
|  | 4.2 | | Hardware/Software Mapping………………………………………….. | | 14 |  |
| 5 | APPENDIX……………………………………………………………. | | | | | 15 |
|  | 5.1 | Appendix A……………………………………………………………. | | | 15 |  |

# Introduction

## Scope of System

Our data processing calculator measures the performance of the PARADISE ELT data framework compared to another ETL framework in this case we are implementing the AWS Glue framework. The calculator only supports one actor which is the client/user that will be uploading and entering the parameters for query.

The system tracks the length of time required to upload the selected files to the Paradise framework and then the AWS Glue framework. Once the files are uploaded, the client will be able to view the amount of time it took to upload the file(s), the amount of data that was uploaded and the historical data for the files uploaded. The system will also track the length of time that it will take to query the files that were uploaded for both Paradise and AWS Glue by using the parameters that the client puts in. The result will be shown on bar graphs for each of the two frameworks, the time it takes to query, and the historical data for the queries.

The system includes functionality uploading and querying JSON files. The system will automatically pull metadata that is stored in a database and the raw data will be stored in S3 buckets. This system is web-based and is accessible over the internet.

# requirements of system

## Functional Requirements

* Startup - This functionality performs the initial configuration of the system including its database and user interface.
* UploadData – This functionality first uploads the selected files to the data source and second creates a database storing necessary metadata of the files such as name, file size, type, date of storage needed for the ELT/ETL process as well as tracking and displaying the time need for each step. It will also present the time it takes to upload to each framework.
* QueryData – This functionality allows a user to perform a query on the data that has been uploaded and present the results in the form of graphs. This also includes the time it took for the system to query the data in the files.

## NON-Functional Requirements

* Platform
  + System is web-based, and a web browser will be used
  + System would be implemented using Microsoft .Net, PARADISE and AWS Glue framework
* Usability
  + System should implement a graphical user interface
  + Graphical user interface should be simplified, visible, and readable by all users
  + Pages should be navigable
  + User Interface should be user friendly

## Use Cases

Diagram

Description automatically generated

Figure 2.1: Use Case Diagram

## use case descriptions

|  |  |
| --- | --- |
| *Use case name* | StartUp |
| *Participating*  *actors* | N/A |
| *Flow of events* | 1. **The system will open to the main menu page.** |
| *Entry condition* |  |
| *Exit condition* | * Main menu is displayed. |
| *Security*  *requirements* |  |

Figure 2.2: Startup

|  |  |
| --- | --- |
| *Use case name* | UploadData |
| *Participating*  *actors* | Initiated by User |
| *Flow of events* | 1. The user is then given a button to select file. The user will be able to select one or more files that they will be able to upload to Paradise and AWS Glue. The user will then click the upload bottom for Paradise. 2. **The system will start a timer when the selected files start uploading to Paradise. The system** **creates a database storing the name, file size, file type, date of storage and label when all selected files are uploaded. When selected files complete uploading, the timer will stop and the upload time for Paradise will be displayed with the amount of data uploaded and the historical data.** 3. The user will then click the upload button for AWS Glue. 4. **The system will start a timer to upload the selected files to AWS Glue. The system creates a database storing the name, file size, file type, date of storage and label when all selected files are uploaded. When the selected files complete uploading, the timer will stop and the upload time for AWS Glue will be displayed along with the amount of data uploaded and the historical data for those files.** |
| *Entry condition* | * The user is at the Upload page. |
| *Exit condition* | * The Upload time and historical time is displayed. |
| *Security*  *requirements* | * The system will need to validate that the files that are uploaded are the correct file type. |

Figure 2.3: UploadData Use Case

|  |  |
| --- | --- |
| *Use case name* | QueryData |
| *Participating*  *actors* | Initiated by User |
| *Flow of events* | 1. The user clicks the Query Data button in the UploadData page. 2. **The system redirects the user to the QueryData page, which has the menu to select the range of years the user wants to query (all elicited from the ingested data) including a “Analyze Query” button.** 3. The user then makes the selection to query the data: he selects the years between 2005 and 2015. The user then clicks the “Analyze Query” button. 4. **The system starts the timer and initiates the query for PARADISE and AWS Glue. Once the query is done for PARADISE and AWS Glue, the query results are represented as a frequency of the number of occurrences of the for the top five vulnerabilities between 2005 and 2015 through a bar graph for each year. The user is also able to see the time it takes to query and the historical query time.** |
| *Entry condition* | * The user is at the upload page and clicking the query now button |
| *Exit condition* | * The user is shown the query results in the form of charts. |
| *Usability*  *requirements* | * The time for query and historical time will be shown for each framework. |

Figure 2.4: QueryData Use Case: Successful Query

|  |  |
| --- | --- |
| *Use case name* | QueryData |
| *Participating*  *actors* | Initiated by User |
| *Flow of events* | 1. The user clicks the Query Data button in the UploadData page. 2. **The system redirects the user to the QueryData page, which has the menu to select the the range of years the user wants to query (all elicited from the ingested data) including a “Analyze Query” button.** 3. The user then makes the selections to query the data from the menu: He selects the years between 2007 and 2009. The user then clicks the “Analyze Query” button. 4. **The system notifies the user that there is non-existent/insufficient data for the query inputted by the user via a banner.** |
| *Entry condition* | * The user is in the main menu |
| *Exit condition* | * The user is shown the notification. |
| *Usability*  *requirements* | * The user should be able to close the banner so he could make another query. |

Figure 2.5: QueryData Use Case: Unsuccessful Query

# USER INTERFACE MOCKUPS

## Main Menu Page

Graphical user interface, text, application, email

Description automatically generated

Figure 3.1: Main Menu

## 3.2 Upload Data Page

Graphical user interface, application, Teams

Description automatically generated

Figure 3.2: UploadData

## QueryData Page

Graphical user interface, application

Description automatically generated

Figure 3.2: QueryData Initial Form

# Software Architecture

## Subsystem Decomposition



Figure 4.1: Subsystem Decomposition

## Hardware/Software Mapping

Figure 4.2 : Hardware/Software Mapping

# APPENDIX

## Appendix A : Source Code

<https://github.com/rpatel121998/DataCalculator1.git>