SUP'COM

المدرسة العليا للمواصلات بتونس
ÉCOLE SUPÉRIEURE DES COMMUNICATIONS DE TUNIS

Training cycle of telecommunications engineers

Engineer Internship Report

Log files analysis tool
I-MONITOR

Submitted by:

Aymen HAMROUNI

Duration:

July 2nd, 2018 — August 31st, 2018

Host Enterprise:

FOD
freedom of dev
services

Internship Supervisor:

Mrs. Randa AMICH

مدينة تكنولوجيا المواصلات – طريق رواد كلم 3.5 – 2083 الغزالة أريانة – الهاتف : 000 857 71 216 + – الفاكس : 829 856 71 216 +
Cité Technologique des Communications - Route de Raoued Km 3,5 - 2083 El Ghazala Ariana
Tél : + 216 71 857 000 - Fax : + 216 71 856 829 - E-mail : contact@supcom.tn - Web : www.supcom.tn

Preface and acknowledgement

From July 2018 till August 2018, I participated in an internship at **FreedomOfDev Services (also known as FOD)**, an IT consulting company which specializes in the study and development of custom software as well as setting up and supporting nearshore and offshore software solutions for European customers. This internship project is a part of my 3- year ICT Engineering program which I conduct at **Sup'Com ; Higher School of Communication of Tunis** . During these two months, I worked on an assignment project to develop an IT log management solution called "**I-MONITOR**" .The main content was to set up a data analytics tool that provides actionable insights from several types of structured and unstructured data sources.

This topic suited my major in information and communication technologies, and also brought me to a very new and interesting area of using business intelligence technologies in Infrastructure Monitoring. Through the assignment, I did not only gain a lot of knowledge but more importantly, I had a great chance to sharpen my skills in a professional working environment. Apart from the business intelligence technologies that I have learnt, I have had also the opportunity to improve my communication skills with training and practice through giving presentations, discussing with the supervisors, experts in the field and other staffs within and outside the company. I am very grateful to Mrs. Randa AMICH, my supervisor at FOD for being always by my side and more importantly for her enthusiastic encouragements and precious instructions during my internship period. Randa gave me very valuable instructions in a brief time and put me in contact with many experts in the field. I would like to express my gratitude to Mrs. Yosr BEN AISSA; a senior professor of English at Sup'Com for helping me

fix this report’s errors and also all the workers at FOD for their warm welcome and generous hospitality.

Abstract

Log files are created by devices or systems in order to provide information about processes or actions that were performed. Detailed inspection of these logs can reveal system abnormal behaviors and even potential security breaches and it can show us system weaknesses.

This report focuses on how to develop a tool to analyze certain log data types (mainly Talend, Bonita and MySQL platforms) and detect anomalies and weird behaviour. Nowadays, performing a detailed log inspection is frequent in many multi-server platforms and can often be bothering and time-wasting that's why it is crucial to look for a solution that can quickly identify and even foresee system's potential problems. In this paper, we have done an offline analysis based on pre-recorded data. In reality, if the objective is to come up with solutions for detecting anomalies in real-time, additional requirements and constraints would be imposed on the algorithms to be used.

This approach uses:

- Talend software as a data extraction, transformation and warehousing.
- Elastic search's search engine technique to enable processing of large data sets in a faster parallel way
- Kibana plugin for data visualization.

In this report, the method for providing such a log analysis tool has been planned from scratch and explained in detail.

Contents

List of Figures

List of Tables

Overview

Anomaly detection plays an important role in the management of modern large-scale distributed systems. Logs, which record system runtime information, are widely used for anomaly detection. Traditionally, developers (or operators) often inspect the logs manually with keyword search and rule matching.

The increasing scale and complexity of modern systems, however, make the volume of logs explode, which hinders manual inspection. In order to reduce this kind of effort, we have developed a user-friendly tool, which we called "**I-MONITOR**", to inspect certain log files types and outline these anomalies in a visual dashboard making it less time consuming and easier for developers to detect exceptions and perform an in-depth analysis. This tool has been evaluated on a private production log dataset, with a total of 4,452,221 log messages and 2.352 anomaly instances.

I believe that our work, with the evaluation results, can provide guidelines for adoption of this tool and provide references for future development.

Host Organization

Freedom of Dev Services (aka FOD) was founded in 2005. It is an enterprise specialized in the study and development of bespoke software as well as the implementation and support of nearshore and offshore software solutions for European customers. It contributes effectively to the digital transformation of companies.

By maintaining flexibility and agility, FOD provides a development of high quality and customized software solutions.

Figure 1 : Logo Freedom of Dev Services

In order to ensure this quality of services for customers, Freedom of Dev Services has set up a team of highly qualified software engineers capable of taking charge of large and complex projects.

The teams listed below are organized according to their expertise in development:

- Quality test team.
- Dynamic teams according to the needs of the projects working

on different axes:

- Java/Java EE
- BonitaBPM
- Mobile
- .Net
- Talend
- Big Data

This figure below presents FreedomOfDev different axes:

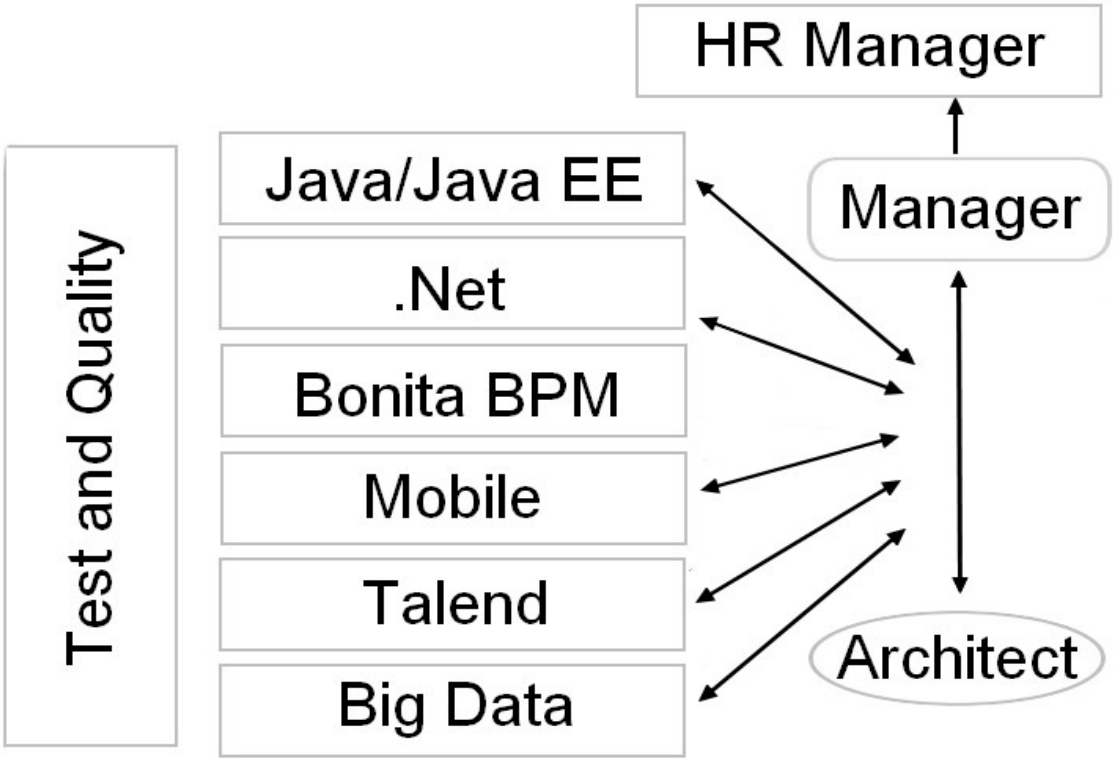


Figure 2 : Freedom of Dev Services axes

Chapter 1: Project Initiation

1.1 Introduction

In this chapter, I will be covering certain preliminary points about the project initiation.

Before diving into theoretical and technical matters which are covered in the chapters to come, I will firstly give an overview of:

- The context of my internship.
- The methodology of work followed during the realization phase.
- The objectives that I seek to achieve along with a motivation part in which I'm going to highlight current practice issues.

1.2 Context of Project

Current software application often produces (or can be configured to produce) some auxiliary text files known as log files. Such files are used during various stages of software development, mainly for debugging and profiling purposes.

Use of log files helps testing by making debugging easier. It allows to follow the logic of the program, at high level, without having to run it in debug mode. Nowadays, log files are commonly used also at customer's installations for the purpose of permanent software monitoring and/or fine-tuning. Log files became a standard part of large application and are essential in operating systems, computer networks and distributed systems, they are often the only way to identify and locate an error in software, because logs are not affected by any time-based issues known as probe Effect. This is the opposite of an analysis of a running program, when the analytical process can interfere with time-critical or resource-critical conditions within the analyzed program.

Some of the tasks of nowadays developers is to be able to analyze these informations, however, going through log files manually can be very hard and time-consuming because of their complexity and large size. In fact, the journalized events generated by the equipments are often very large and can have complex structure. Although the process of generating log files is quite simple and straightforward, log analysis could be a tremendous task that requires enormous computational resources, long time and sophisticated procedures. This often leads to a common situation, when log files are continuously generated and occupy valuable space on storage devices and nobody uses them and utilizes enclosed information, however, choosing to ignore them is a very bad decision one can make and this is because of several reasons. as a matter of fact, Online services using the Internet or IP technologies put in place by companies are constantly growing, and the need for security and traceability to minimize the vulnerability of a system against accidental or intentional threats has become paramount.
In this context, my project consists in developing an application that is both efficient and easy-to-use for filtering and analyzing logs.

1.3 Current Practice and State of Technology

While searching the Internet, I encountered a multitude of "log" file analysis tools. In what follows, I will introduce a few that I consider some of the most important tools in terms of efficient and among the most commonly used.
These tools are:

- **AWStats**: AWStats is an open source Web analytics reporting tool, suitable for analyzing data from Internet services such as web, streaming media, mail, and FTP servers. AWStats parses and analyzes server log files, producing HTML reports.

It can display the number of hits, unique hits, current visitors and other informations such as host's time, browser, OS, etc. AWStats is free software.

• **Webalizer**: The Webalizer is web log analysis software, which generates web pages of analysis. It is one of the most commonly used web server administration tools.

Statistics reported by Webalizer includes: number of hits and current visitors, country of origin of visitors, amount of data downloaded …

These measurements can be represented graphically, and according to different time scales such as per month, per day, or per hour.

• **Advanced Log Analyzer**: Advanced Log Analyzer is a powerful traffic analysis software. It generates a large number of reports such as sites referring to visitors, the number of downloads per day, the number of clicks and hosts per day, the most popular search engines, search keywords, etc. . It can be used as visitors counter and tracking tool to monitor the activity of a website. The main advantage of this analysis tool lies in its reports. It can recreate the visitor's path from the log files and produce reports including the user flow on the site.

• **Sawmill** : Sawmill is a universal log analysis/reporting tool for almost Any logging system including web, media, email, security, network and application logs.

Sawmill supports 850 different common log formats from the full range of popular devices: web servers, media servers, mail servers, firewalls, gateways, and so on. However, this tool is not free compared to other tools available.

• **The Elasticsearch Stack (ELK)**: Elastic Stack is a group of open source products from Elastic designed to help users take data from any type of source and in any format and search, analyze, and visualize that data in real time. The products in the group are Elasticsearch ; a RESTful distributed search engine built on top of Apache Lucene,

Logstash ; a data collection engine that unifies data from disparate sources and Kibana ; an open source data visualization and exploration tool. A fourth product, Beats, was subsequently added to the stack which is a "data shipper" that is installed on servers as agent used to send different types of operational data to Elasticsearch either directly or through Logstash

1.4 Motivation

Although the applications mentioned above (known as log file analyzers or log files visualization tools) are numerous, they don't often provide a detailed error reports nor dashboards for in-deep analysis, they mostly offer basic features, superficial reports with general information.
Such tools are undoubtedly useful, but their usage is limited only to a certain end with log files of certain structure and most importantly doesn't natively support various sources such as Talend, MySQL and Bonita.

1.5 Offered Solution

The log analysis must be transparent, it must not impact the application services.
The consumption of CPU, memory and I / O resources to collect logs must be as low as possible. The log analysis platform must not impose constraints on the various application services, it must adapt to the many log formats generated. Moreover, it must be reliable in order to avoid "false positives". Finally, it must be sized in accordance with the volume of all logs to ensure real-time analysis.
It is a "turnkey" solution based on the assembly of three products: Elasticsearch, Talend and Kibana.

This allows a great flexibility of configuration to develop a product that perfectly meets its own needs. The technologies employed are rather young but benefiting from a large and responsive community.
Our solution offers multiple features:

- The log-files are indexed as documents in Elasticsearch; a No-SQL distributed database.
- Users have only one statistics tool to handle.
- The analysis of a large amount of data is done in real time.
- The tool's performance is linear, it does not deteriorate as the number of log recorders increases.
- Information's collection does not affect user's experience. In fact, it is not necessary to install neither browser extensions nor additional framework packs besides Java Runtime Environment.
- Formatting and data enrichment are done in real time, at the same time as indexing.

1.6 Conclusion

In this chapter, we explained the context of the accomplished work. Also, we did a brief review on the current practice and State of technology along and mentioned some of the reasons that made us offer our own solution.
In the following chapter, I will be covering several parts such as the definition of the work methodology, the work environment, the overall architecture of the application and the functional analysis.

Chapter 2: Project Methodology

2.1 Introduction

This chapter is dedicated to presenting the work methodology and the development environment chosen during my internship.

To do so, I have divided this chapter into two parts. The first part includes an overview of the work methodologies and justifies my choice picking a specific method. The second part presents the development environment.

2.2 Work Methods

As part of this internship, I was interested in studying the most popular methods, to be able to choose the most suitable method for my project.

2.2.1 Development approaches

Although there is a multitude software development methodology, choosing the appropriate one for a given project is a pretty important decision.

I present below, a brief study on which mine was based.

2.2.2 Agile Methods

Agile methods are a particular approach essentially dedicated to the management of IT projects. They rely on iterative and adaptive development cycles (commonly known as sprints) based on the evolving needs of the customer. Moreover, they make it possible to involve all employees as well as the client in the development of the project.

These methods generally make it possible to better respond to the client's expectations in a short time (partly thanks to the latter's involvement) and therefore constitute a gain in productivity as well as a competitive advantage for both the customer and the supplier side.

The table below shows a comparison between certain methodologies:

	Description		

		Advantage	Inconvenient
XP	-Dedicated to projects of less than 10 people.	- Pair Programming. - Guided development	-Maintenance. -No analysis phases.
SCRUM	-Management practices and work organization.	-Time management. -Team management. -Project sharing across sprints	-No development practices.
DSDM	-It gives the direction of the project through the business objectives.	-Priority approach requirement. -Effective project management	-Cannot be the solution to all types of projects. -Team communication based on the documentation.

Table 1 : Description of Few Agile Methods

2.2.3 Chosen Method

After going through the different agile methods, we have chosen to pursuit the implementation of our project with the SCRUM methodology and this is because the latter represents several advantages over other development methodologies, it ensures the smooth running of the project within the team and guaranties tangible progress of the work.

Team spirit, communication, collaboration, and flexibility are all factors that ensures customer satisfaction.

In order to highlight these factors and guaranty a better progress, we followed the work strategy described above and divided the SCRUM team of this project into mainly 3 people:

- 1 trainee who forms the development team.
- 1 supervisor who plays the role of the Scrum Master

- The customer who plays the role of the Product Owner

In our case, the role of the Product Owner was entrusted to a quality assurance manager, her task was specifying the customer's need and keeping an eye of the project flow: Weekly meetings have been scheduled with her, to fix the scope of the next iteration "**SPRINT**" and to perform a demonstration on the accomplished work during the last iteration.

2.3 Work Environment

2.3.1 Hardware Environment

The development of this tool was performed on a laptop that has the following features:

Characteristic	Type
Processor	Intel Core i5-6200U
RAM	8 Go
Hard drive	1 To
OS	Windows 8.1 64x

Table 2 : Hardware Environment

2.3.2 Software Environment

In this part, I will present an overview of the technologies and the tools used for the realization of this project.

- **Talend:**

Talend is an open source data integration platform that helps you in effortlessly turning raw data into business insights. It provides various software and services for data integration, data management, enterprise application integration, data quality, cloud storage and Big Data.

Below, you will find the various products released by Talend since its launch:

Figure 3 : Talend Offerings

Among all these products, the most used products are Talend Open Studios, as they are available free of cost and anyone can download and use them. Therefore, and after further researches on what Open Studios' products can offer, we have found that the IT Solution; Talend for ESB meets our needs and offers wide range of features such as:

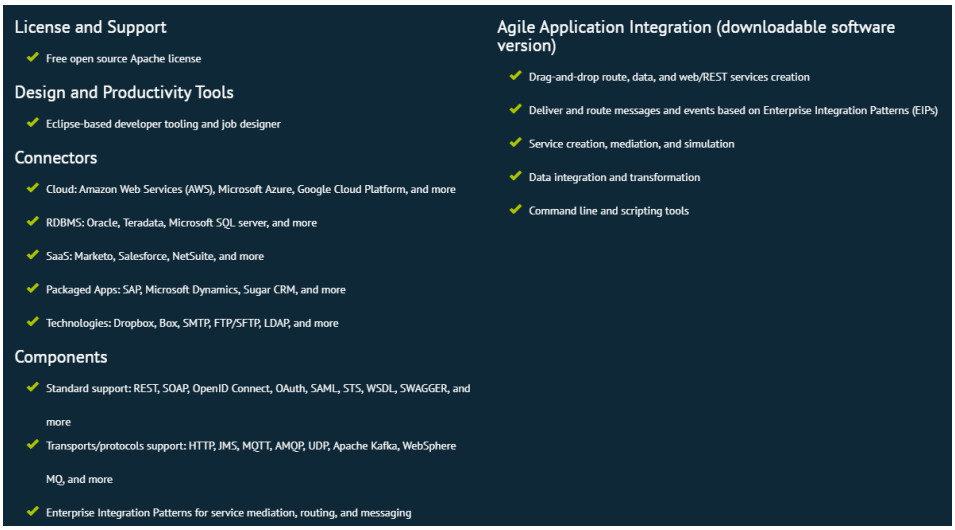


Figure 4 : Features of Talend Open Studio for ESB

- Elasticsearch:

For our project, we needed an open-source, RESTful, distributed search and analytics engine for data storage, Elasticsearch, after thorough research,

was the right tool for this job, In fact, it provides a distributed, multitenant-capable search engine with a HTTP web interface and schema-free JSON documents Elastic search supports real-time GET requests, which makes it suitable as a NoSQL data store.

Also, what made us choose elastic search is the fact that it was developed alongside an analytics and visualization platform called Kibana.

Figure 5 : Elasticsearch Logo

- **Kibana:**

Kibana is an open source data visualization plugin for Elasticsearch. It provides visualization capabilities on top of the content indexed on an Elasticsearch cluster. Users can create bar, line and scatter plots, or pie charts and maps on top of large volumes of data.

Figure 6 : Kibana Logo

- **Batch:**

Batch is a programming language. It is used to create script files executable on Windows operating system. Normally, normally these files have an extension of ".bat" or ".cmd". When being executed, they open a "Command Prompt" window, which normally has a typical black background, white text. The batch files (*.bat, *.cmd) are called script files which can contain commands interfering operating system.

Note: a language that is equivalent to **batch** but used for **Linux** operating system is **Shell**, with script files ending in ***.sh**.

2.4 Conclusion

In this chapter we reviewed various work methodologies and highlighted the most useful in our case then we exposed the development environment; both hardware and software.

Chapter 3: Project Analysis

3.1 Introduction

The Analysis Phase is the first phase in a Project Development Life Cycle. It is where you break down the deliverables in the high-level Project Charter into the more detailed business requirements.
The focus during this chapter is on what is needed and not how the needs are met. Determining how the needs are met would be determined during the Design Phase. During this phase we will identify the overall direction that the project will take through the creation of the project strategy documents and backlogs.

3.2 Project Requirements

In this section we will list the different functional and non-functional needs of our application

3.2.1 Functional Requirements

The functional requirements are the different features our application must offer to the users. In what follows, we will identify and list what we found our tool must satisfy.
The application needs to:

- Be able to process and understand Talend, Bonita and MySQL log files structure.
- Be able to ignore unwanted data in log files.
- Be able to complete the mentioned above tasks in bulk given a specific directory.
- Be able to differentiate between log files' type.
- Be able to upload the result data to database.
- Be able to visualize the output data in a dashboard.

3.2.2 Non-Functional requirements

Non-functional requirements represent the implicit requirements that the tool must meet. In what follows, we present the non-functional needs that must be fit:

- **Performance**: The application must be reliable and always functional; it must be crash-proof and covers all possible use cases and operations' scenarios. Also, it must have a reduced response time and optimized processing time.

- **Ergonomics**: The application must offer an aesthetic, exploitable and interactive interface making it easy for users to manipulate.

3.3 Functional analysis

3.3.1 Identifying Actors

In the following figure, we present the system's context diagram:

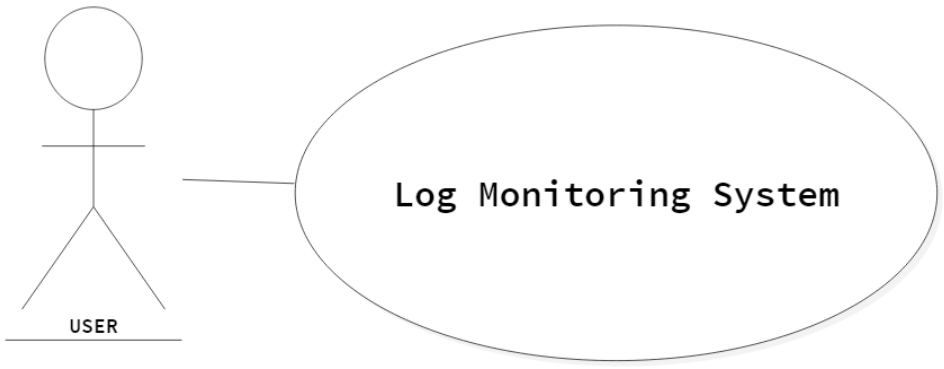


Figure 7 : System Context Diagram

From this diagram, we can initialize a primary actor that interacts with our system, this actor "**USER**":

- Provides necessary entries such as logs' types and directories.
- Customizes application's dashboard to suit his needs.

3.3.2 Global Use Case Diagram

The following figure shows the Global use case diagram of our project. This diagram can give us a global view of the application.

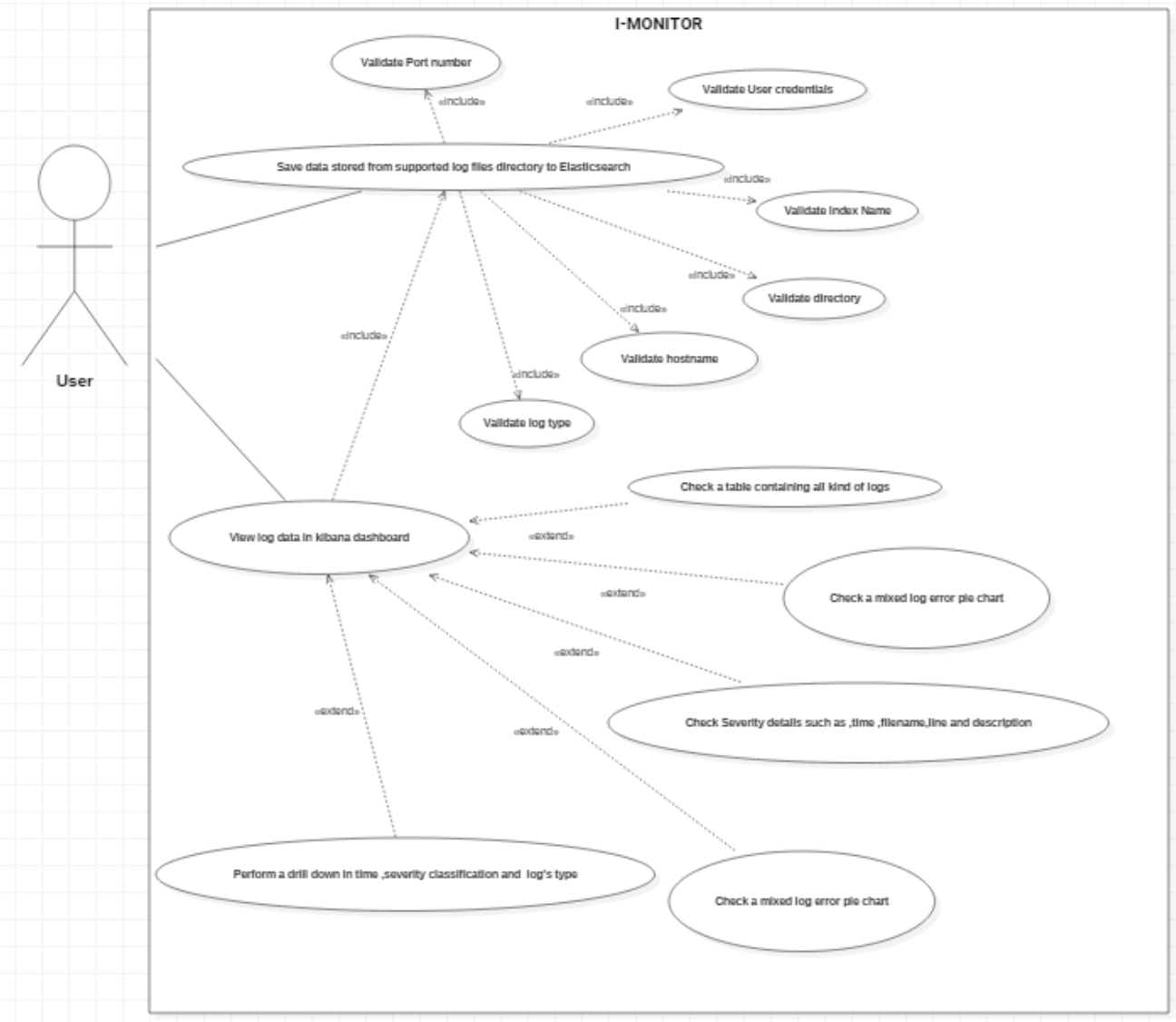


Figure 8 : System Use Case Diagram

3.4 Sprints

In this section, we will define the design and the realization of the product's phases. Each phase is implemented in a Sprint fixed with the product owner in a weekly meeting "Sprint planning".

During this Internship, we have divided the work into 4 Sprints.

3.4.1 Sprint 1: Visualize Talend log-file's data in Kibana

This sprint lasted a week of development where we set up the part of visualizing Talend log file data in Kibana.

3.4.1.1 Sprint 1 Backlog

User stories	Description	Story points	Tasks	Number of hours
1 – As a user, I want to have a structured file from Talend log file.	From a log file, you have to know how to separate the information each according to its type and classify it in a csv file in column format where each column corresponds to a specific type of information known beforehand.	3	1- Read log file line by line	4
			2 – Find a suitable separator (field separator) to extract each type of information	8
			3- Store processed data in a csv file	4
2- As a user, I want to be able to transfer the data stored in delimited files to Elasticsearch	From a delimited file (csv), it is necessary to know how to associate each line to a structured document in JSON format, that after, will be transferred to the Elasticsearch DB	3	1 – Find a generic schema for documents	4
			2- Associate each type of information with the document attribute	6
			3- Generate a JSON file containing the list of documents created and start writing these documents in Elasticsearch database	6

3- As a user, I want to be able to view the data stored in Elasticsearch	From the documents stored in ES, you have to know how to visualize a table containing each type of information of each attribute and its value	1	1- Create a table representing the stored data	3

Table 3 : Sprint 1 Backlog

3.4.1.2 Sprint 1 Use Case Diagram.

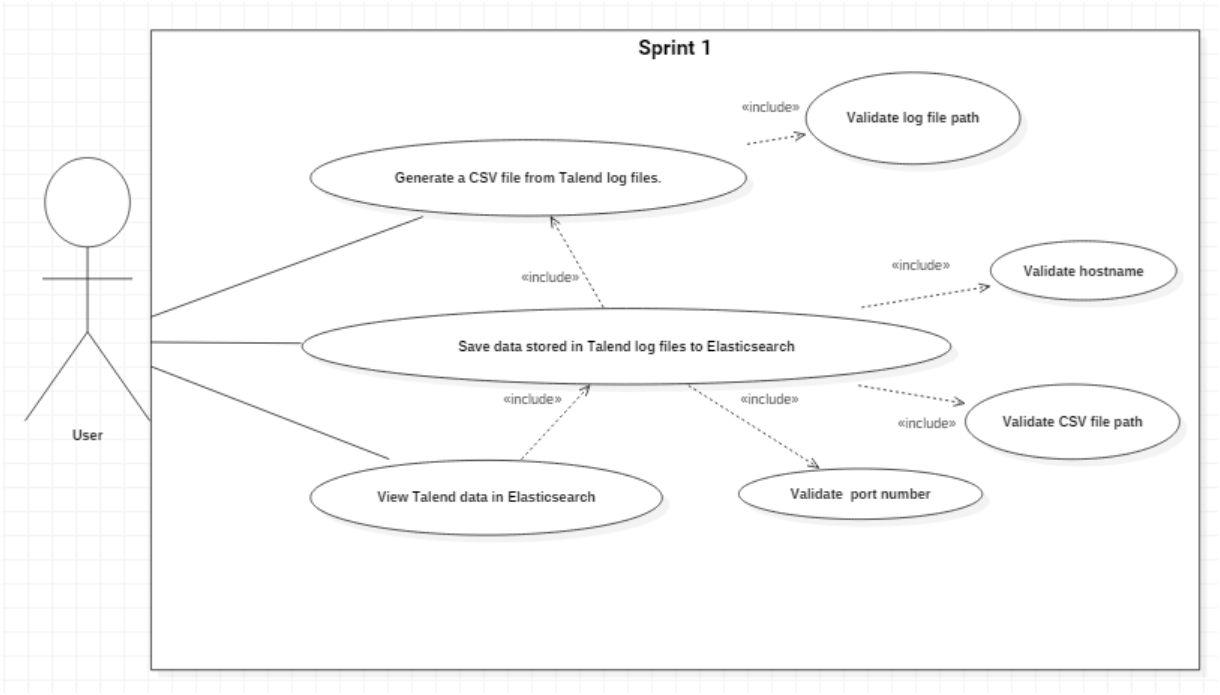


Figure 9 : Sprint 1 Use Case Diagram

- Choice 1: Generate a CSV file from Talend log files

Actor: User.

Description: User has the right to create a structured file from Talend log file.

Pre-condition: A Talend log file must be available.

The nominal scenario
1. User fills in required entries. (Log file path, CSV file Path)
2. User starts the process.

3. A CSV file is generated in the output path specified.

Table 4 : Sprint 1-Choice 1- Nominal Scenario

The alternative scenario
1. One of the entries is invalid.
2. User re-fills all the entries.
3. User starts the process.
4. A CSV file is generated in the output path specified

Table 5 : Sprint 1- Choice 1- Alternative Scenario

- Choice 2: Transfer data stored in Talend log file to Elastic search

Actor: User.

Description: User has the right to transfer data from a CSV file to Elasticsearch.

Pre-condition: A CSV file must be available, and the device must be connected to a network if Host is remote.

The nominal scenario
1. User fills in required entries. (CSV file Path, Elasticsearch hostname and port number)
2. User starts the uploading process.
3. All CSV data are uploaded to Elasticsearch.

Table 6 : Sprint 1-Choice 2- Nominal Scenario

The alternative scenario
1. One of the entries is invalid.
2. User re-fills all the entries.
3. User starts the process.
4. All CSV data are uploaded to Elasticsearch.

Table 7 : Sprint 1-Choice 2- Alternative Scenario

- Choice 3: View Talend data in Kibana

Actor: User.

Description: User has the right to visualize Talend data in Kibana.

Pre-condition: Talend data must be available in Elasticsearch.

The nominal scenario
1. User visualizes Talend log file data in Kibana

Table 8 : Sprint 1-Choice 3- Nominal Scenario

3.4.1.3 Sprint 1 High Level Design

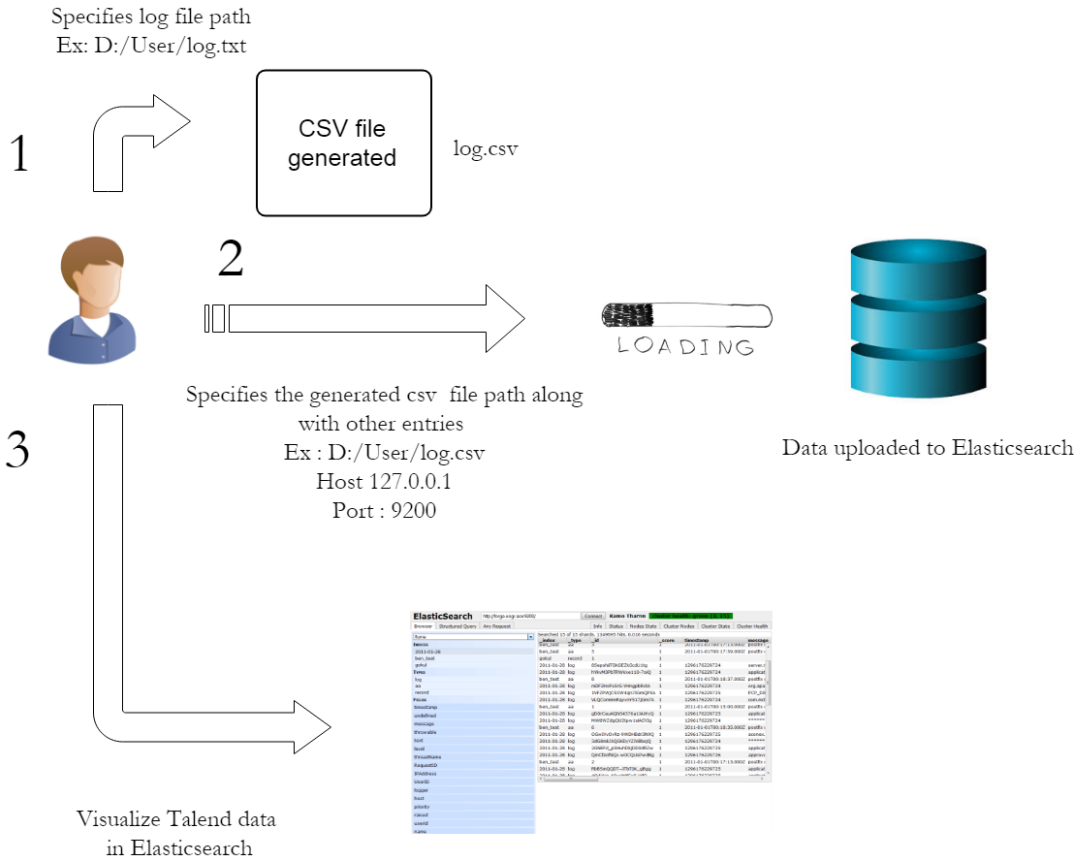


Figure 10 : Sprint 1 High Level Design

The diagram above illustrates the high-level design for the first sprint.

3.4.2 Sprint 2: Visualize both Bonita and Talend log files in Kibana

This sprint lasted one week of development where we set up the part of processing Bonita log files and the one-off visualizing them in Kibana, we also set up a user-friendly interface that allows the user to specify the log file type.

3.4.2.1 Sprint 2 Backlog

User stories	Description	Story points	Tasks	Number of hours
1-As a user, I want to be able to view the data stored in a Bonita log file	From a Bonita log file, you have to know how to separate the information each according to its type and classify them in a csv file in column format then transfer them to Elasticsearch for a possible visualization by Kibana	1	1- Find a suitable separator (field separator) to extract each type of information	8
			2-Store the processed data in a csv file	1
			3 - Find a generic schema for documents	1/2
			4 -Transfer the data to Elasticsearch	1/4
			5 -Visualize a representative table of the stored data	1/4

2-As a user, I want to have an interface easy to handle by which I must be able to choose the technology Bonita or Talend as well as other parameters (Hostname, Port, Elasticsearch index name)	Design an executable that can process both Bonita and Talend log files according to the choice of the user, this choice is made using a user-friendly interface. The user must also insert some parameters like; the Hostname, the location of the log files, the location of the csv file to generate, the hostname, the port, Elasticsearch index name.	3	1-Realize a Job Master which contains both the processing * of the Bonita log file and the * processing of the Talend log file (two Job sons) Treatment *: Log -> Kibana	3
			2-Execute one of the child jobs according to a choice described by a variable	8
			3- Pass the parameters entered by the user to each child job	2
			4-Realize an executable file that takes certain parameters such as : Hostname, Technology log, port … and transfer data to Elasticsearch	8

Table 9 : Sprint 2 Backlog

3.4.2.2 Sprint 2 Use Case Diagram:

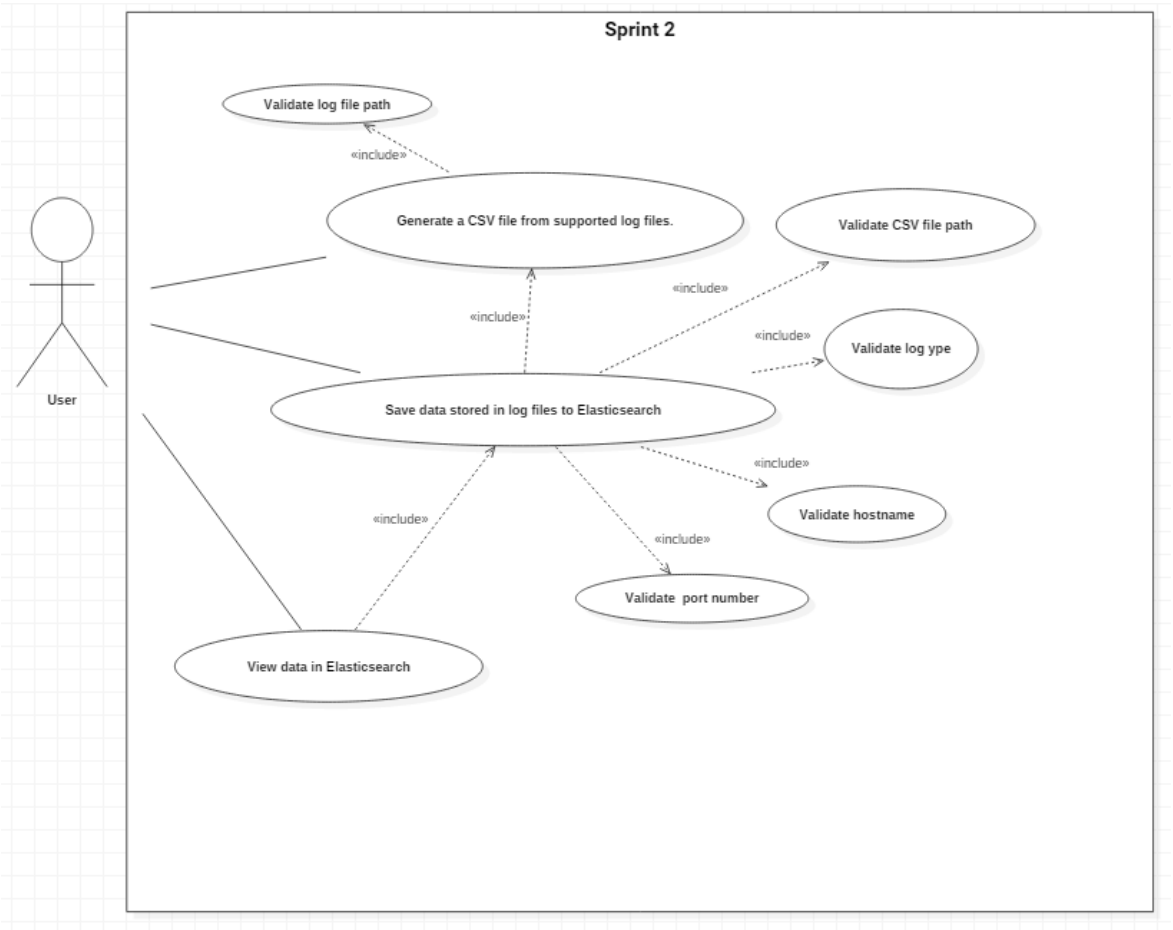


Figure 11 : Sprint 2 Use Case Diagram

- Choice 1: Generate a CSV file from Talend or Bonita log files

Actor: User.

Description: User has the right to create a structured file from Bonita log file

Pre-condition: A Bonita log file must be available.

The nominal scenario
1. User fills in required entries. (Log file path, CSV file Path)
2. User starts the process.
3. A CSV file is generated in the output path specified.

Table 10 : Sprint 2-Choice 1- Nominal Scenario

The alternative scenario

1. One of the entries is invalid. 2. User re-fills all the entries. 3. User starts the process. 4. A CSV file is generated in the output path specified

Table 11 : Sprint 2-Choice 1- Alternative Scenario

- Choice 2: Transfer data stored in Bonita log file to Elastic search

Actor: User.

Description: User has the right to transfer data from a CSV file to Elasticsearch.

Pre-condition: A CSV file must be available, and the device must be connected to a network if Host is remote.

The nominal scenario
1. User fills in required entries. (CSV file Path, Elasticsearch hostname and port number) 2. User starts the uploading process. 3. All CSV data are uploaded to Elasticsearch.

Table 12 : Sprint 2-Choice 2- Nominal Scenario

The alternative scenario
1. One of the entries is invalid. 2. User re-fills all the entries. 3. User starts the process. 4. All CSV data are uploaded to Elasticsearch.

Table 13 : Sprint 2-Choice 2- Alternative Scenario

- Choice 3: View Bonita data in Kibana

Actor: User.

Description: User has the right to visualize Bonita data in Kibana.

Pre-condition: Talend data must be available in Elasticsearch.

The nominal scenario
1. User visualizes Bonita log file data in Kibana

Table 14 : Sprint 2-Choice 3- Nominal Scenario

3.4.2.3 Sprint 2 High Level Design

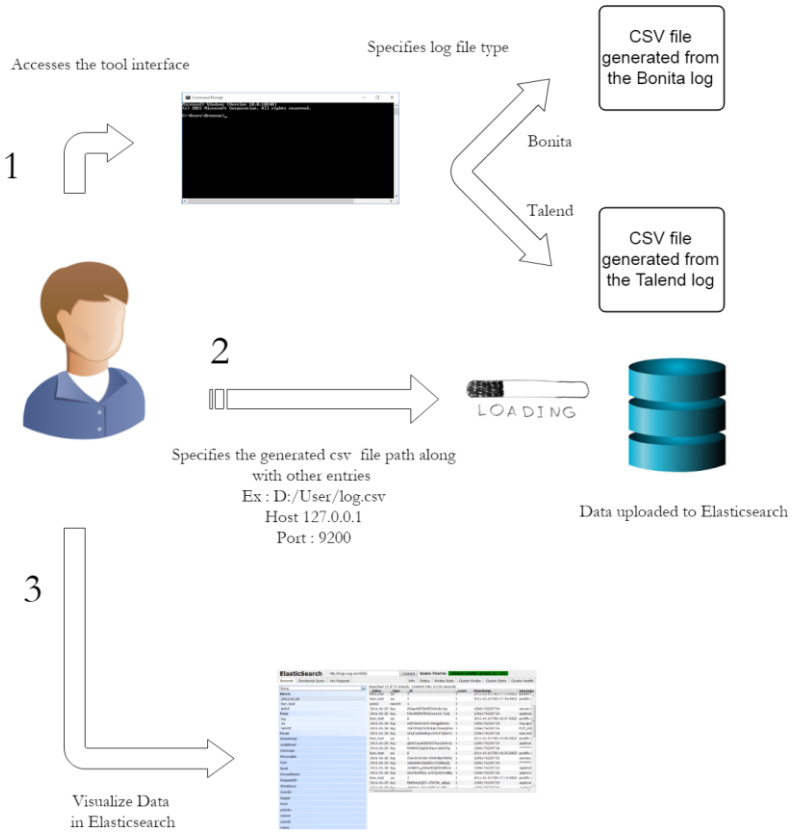


Figure 12 : Sprint 1 High Level Design

The figure above describes the high-level design for the second sprint.

3.4.3 Sprint 3: Remove CSV Generation Phase and add MySQL log type.

This sprint lasted one week of development during which we set up the part of processing MySQL log files and the one-off visualizing them in Kibana, also, we removed the phase of generating a CSV file from the log file. Hence, Kibana is the only possible way to visualize data after processing.

3.4.3.1 Sprint 3 Backlog

User stories	Description	Story points	Tasks	Number of hours
1 - As a user, I want to eliminate the intermediate phase delimited file generation.	From a Bonita or Talend log file, you have to know how to transfer the data directly to Elasticsearch without going through the phase of generation a csv file.	1	1- Eliminate the intermediate step of csv file.	8
2-As a user, I want to be able to view the data stored in a MySQL log file	From a MySQL log file, you have to know how to separate the information each according to its type and transfer them to Elasticsearch for a possible visualization by Kibana	2	1- Find a suitable separator (field separator) to extract each type of information	7
			2 - Find a generic schema for documents	1/2
			3 -Transfer the data to Elasticsearch	1
2-As a user, I want to have a tool that allows the processing of Talend and Bonita files wholesale, this treatment is done on a directory containing the log files homogeneous, while ensuring the control on data	Design an executable that can process both log files Bonita, Talend or MySQL according to the user's choice. The user must insert certain parameters like; the location of the folder containing the log files (homogeneous), the hostname, the username, the password, Elasticsearch index	3	1-Modification of the job so that the processing is done on a set of files in a specific directory.	8

entry.	name.		2-Realize a Job Master which contains both the processing * of log file Bonita, Talend and MySQL Treatment *: Log -> Elasticsearch	2
			3- Check the data entered by the user and provide an explanatory display in case of error	10

Table 15 : Sprint 3 Backlog

3.4.3.2 Sprint 3 User Case diagram

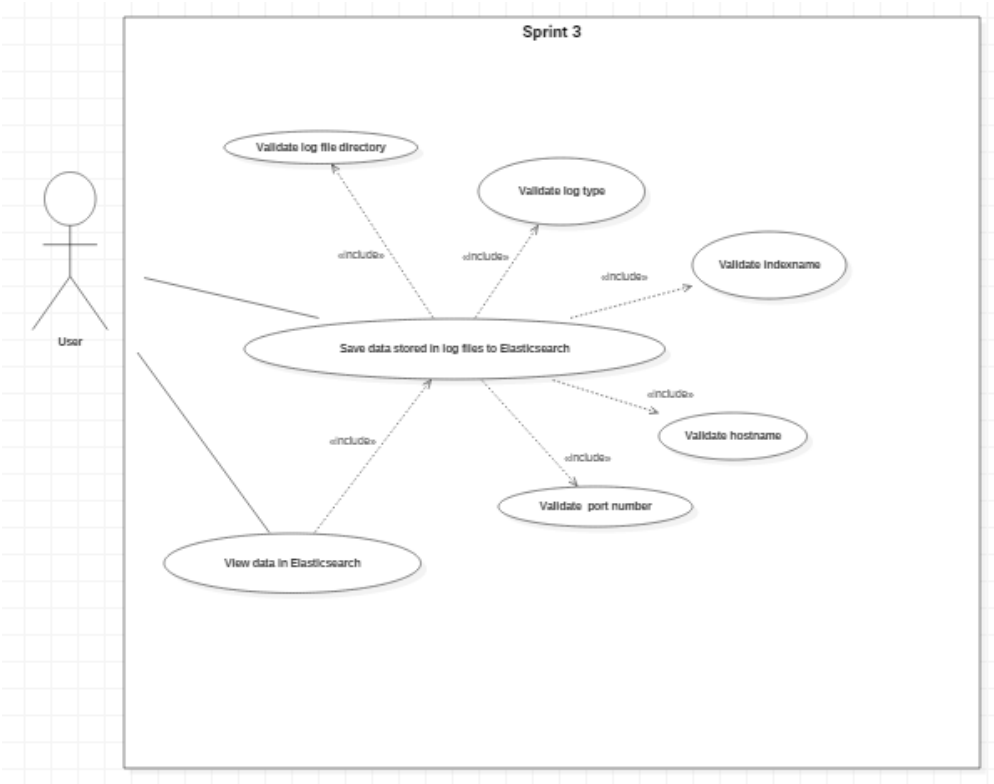


Figure 13 : Sprint 3 Use Case Diagram

- Choice 1: Transfer data stored in Bonita/Talend/MySQL log file to Elastic search

Actor: User.

Description: User has the right to transfer data from Bonita/Talend/MySQL file to Elasticsearch based on his choice.

Pre-condition: A Bonita/Talend/MySQL log file must be available along with a device connected to a network if the user has a remote Elasticsearch server.

The nominal scenario
1. User fills in required entries. (Log file path, Hostname, Port number)
2. User starts the process.
3. Log data is transferred to elastic search.

Table 16 : Sprint 3-Choice 1- Nominal Scenario

The alternative scenario
1. One of the entries is invalid.
2. User re-fills all the entries.
3. User starts the process.
4. Log data is transferred to elastic search.

Table 17 : Sprint 3-Choice 1- Alternative Scenario

- Choice 2: Visualize Talend/MySQL/ Bonita data in Kibana

Actor: User.

Description: User has the right to visualize Talend/MySQL/Bonita data in Kibana.

Pre-condition: Talend/MySQL/Bonita data must be available in Elasticsearch.

The nominal scenario
1. User visualizes Talend/MySQL/Bonita log file data in Kibana

Table 18 : Sprint 3-Choice 2- Nominal Scenario

3.4.3.3 Sprint 3 High Level Design

4 The figure below describes the high-level design for the sprint 3.

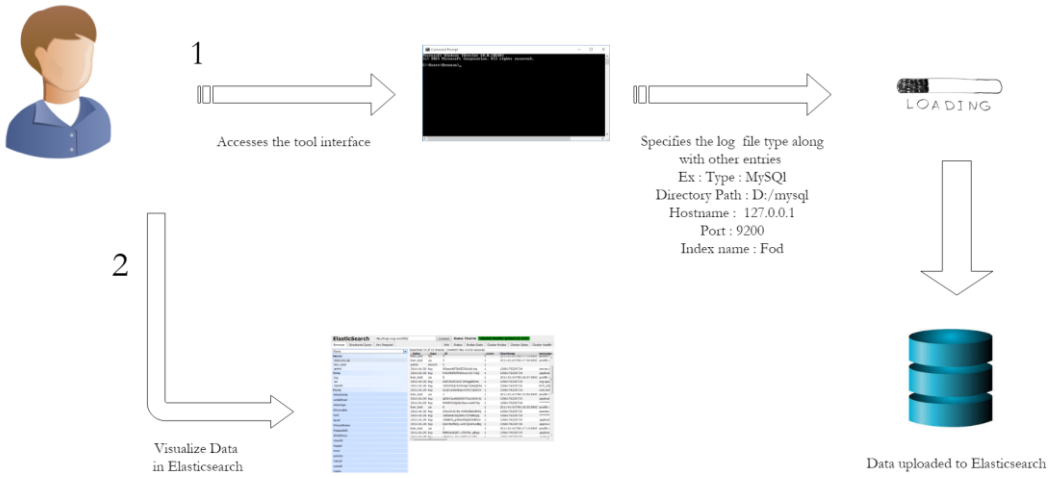


Figure 14 : Sprint 1 High Level Design

4.1.1 Sprint 4: Visualize Log file data in Kibana

This sprint is the last sprint, it lasted one week of development during which we set up the part of creating a Kibana dashboard that provides the user the necessary visualization to monitor and perform analytic and statistic researches.

4.1.1.1 Sprint 4 Backlog

User stories	Description	Story points	Tasks	Number of hours
1 - As a user, I want to have an interface, a kind of dashboard that allows to clear the information that seems useful from the different log files (MUST	From the documents of different log files stored in Elasticsearch, you need to set up a GUI using Kibana; this interface presents pie charts, statistical tables, and analytic charts.	2	1- Make a little documentation on Kibana; how to create visualizations? How to pull information using QUERY DSL and display it?	10

HAVE: a chart pie chart that allows to group the errors of the files logs)			2-For each type of log (Index pattern), the realization of graphs in curve, pie charts …	6
			3- Find a way to group and exploit log file information independently of the chosen index pattern in Kibana. .	4
			4-Realization of a dashboard which gathers the visualization of the information from log files	4

Table 19 : Sprint 4 Backlog

4.1.1.2 Sprint 4 User Case Diagram

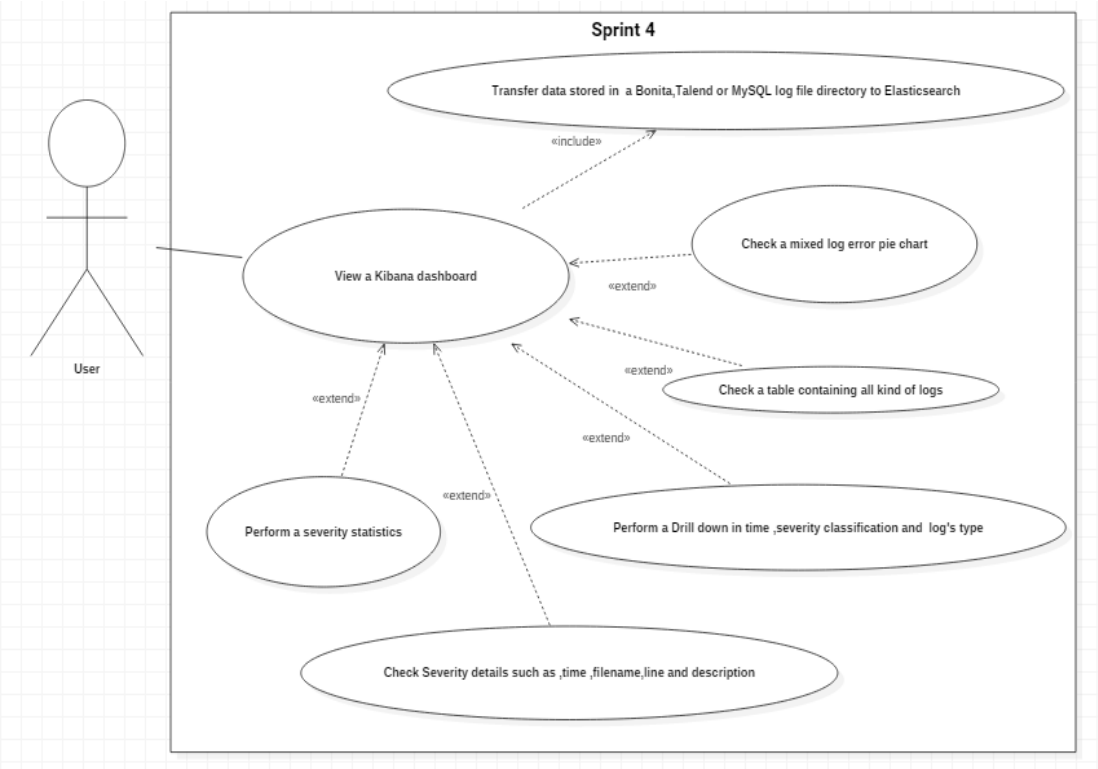


Figure 15 : Sprint 4 Use Case Diagram

- Choice 1: Transfer data stored in Bonita/Talend/MySQL log file to Elastic search

Actor: User.

Description: User has the right to consult pie charts, perform a severity check, perform a drill down …

Pre-condition: a Bonita/Talend/MySQL data must be available in Elasticsearch.

4.1.1.3 Sprint 4 High Level Design

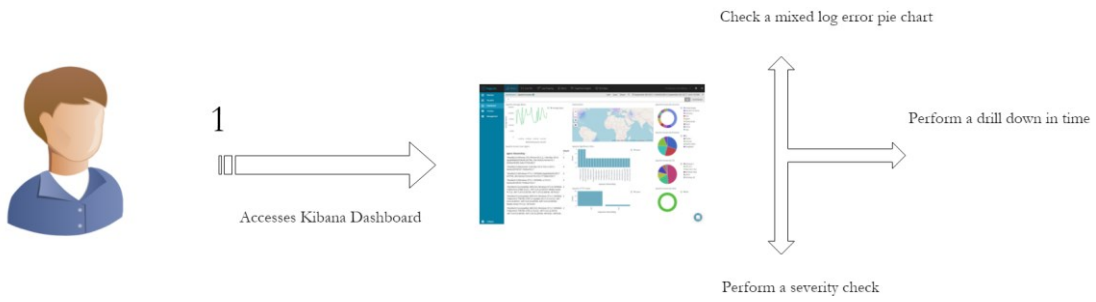


Figure 16 : Sprint 4 High Level Design

The figure above describes the high-level design for the last sprint.

4.2 Conclusion

In this chapter, we have studied the use case diagram and the physical architecture of our application to get a more detailed look at the project.

Chapter 4: Project Realization

4.1 Introduction

In this final chapter, we will talk about the phase in which the development of the application took place.

During the previous chapter, we have only broached theoretical matters by mentioning the project backlogs, however, now we will discuss the technical aspect by stating and showing the tool functionalities.

4.2 Achieved Work Description

During the development of this project, we have achieved a lot of features, as it is mentioned above in the backlogs of each sprint.

In this section, we will introduce some pictures taken from the tool. We will explain each one's operation and where it figures in the actual application.

4.2.1 Sprint 1:

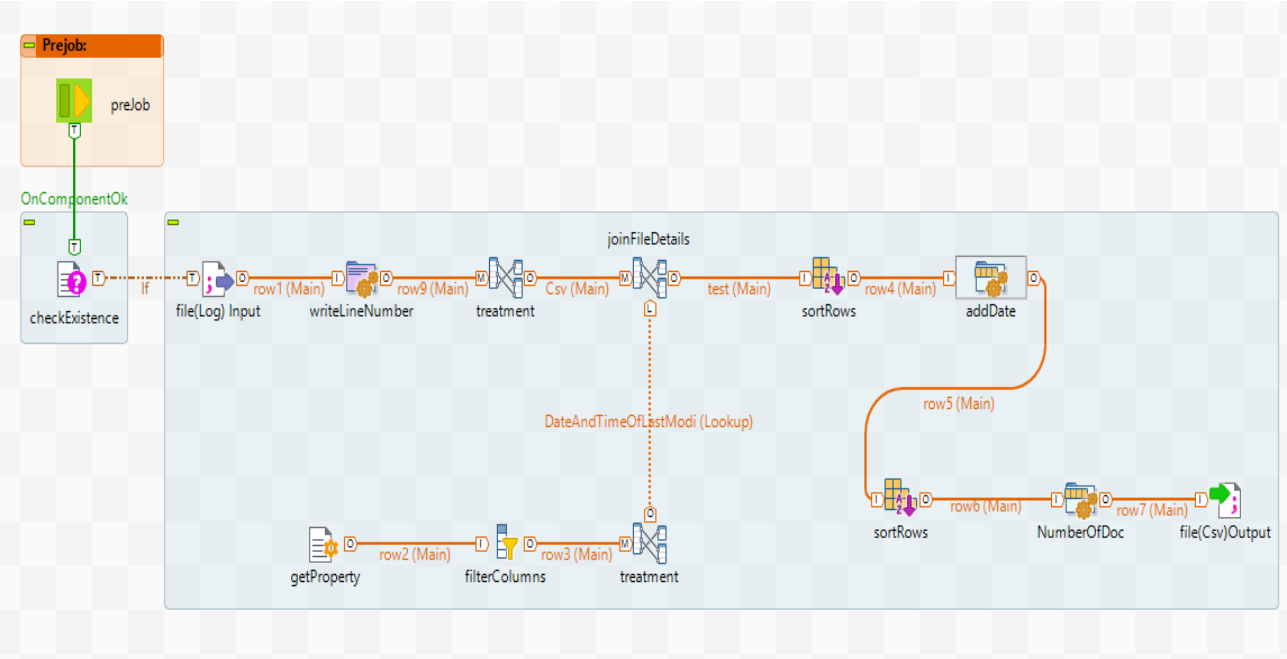


Figure 17 : Talend Job for CSV Generation

The figure above shows different Talend components used in order to accomplish CSV generation from Talend log files.

We must draw your attention that the process is ordered and it starts from the left to right, In fact, Talend component (tfileinputdelimited) which is we named here "file (log)Input" read the desired log files and passes the data row by row for processing to the next components until it reaches the Talend component (tfileoutputdelimited) which writes a CSV file with the output data in a user-provided path.

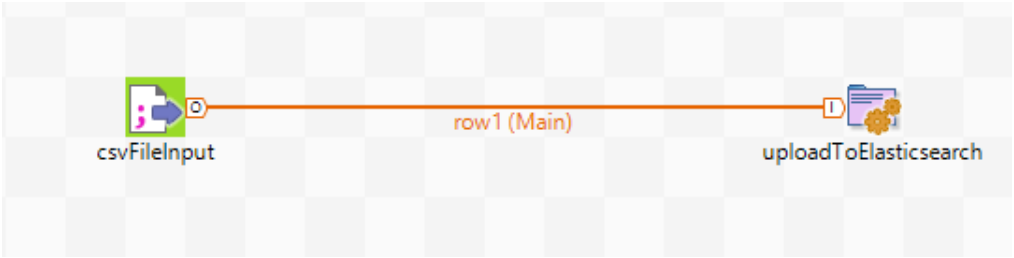
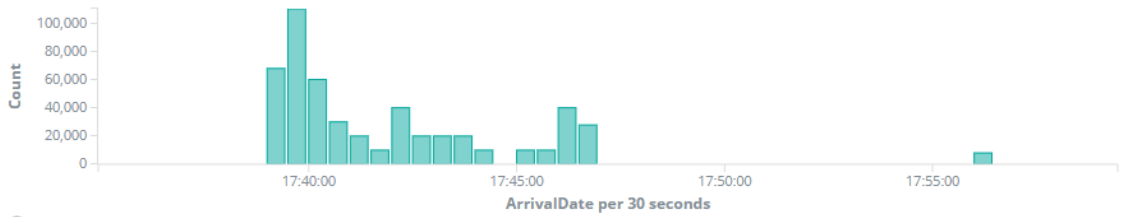


Figure 18 : Talend Job For Elasticsearch Uploading

The figure 18 shows the process of uploading data to Elasticsearch from a CSV file.

The left component reads the processed data and passes it to "uploadToElasticsearch" component which in its turn finishes the job by transferring it to Elasticsearch



Time	Client	_index	Filename	Line	Severity	Thread
September 17th 2018, 17:39:09.308	fod	talendfod	tesb4.log	1,724	INFO	pool-30-thread-1
September 17th 2018, 17:39:08.898	fod	talendfod	tesb4.log	12	WARN	pool-1-thread-1
September 17th 2018, 17:39:08.902	fod	talendfod	tesb4.log	18	INFO	pool-2-thread-1
September 17th 2018, 17:39:08.906	fod	talendfod	tesb4.log	19	INFO	pool-2-thread-1
September 17th 2018, 17:39:08.912	fod	talendfod	tesb4.log	24	INFO	pool-2-thread-1
September 17th 2018, 17:39:08.915	fod	talendfod	tesb4.log	30	INFO	pool-2-thread-1
September 17th 2018, 17:39:08.924	fod	talendfod	tesb4.log	37	INFO	pool-2-thread-1
September 17th 2018, 17:39:08.928	fod	talendfod	tesb4.log	43	INFO	pool-2-thread-1
September 17th 2018, 17:39:08.929	fod	talendfod	tesb4.log	44	INFO	pool-2-thread-1

Figure 19 : Talend Log Data in Kibana

The figure above illustrates the obtained data in Kibana after Elasticsearch uploading.

4.2.2 Sprint 2:

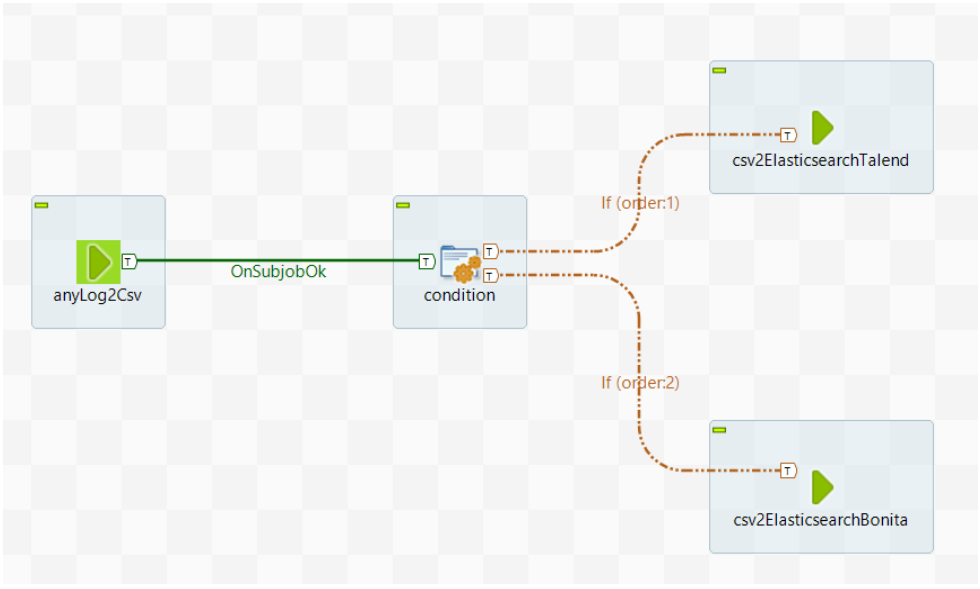


Figure 20 : Talend Master Job for Sprint 2

This figure above shows the Talend Master Job for sprint 2. The data flow starts from the left going through "anylog2cs" component which generate CSV file from the input log file, after that, the flow passes to the condition

component which determines where the resulted data goes, as you can see in this case, we only have two ways.

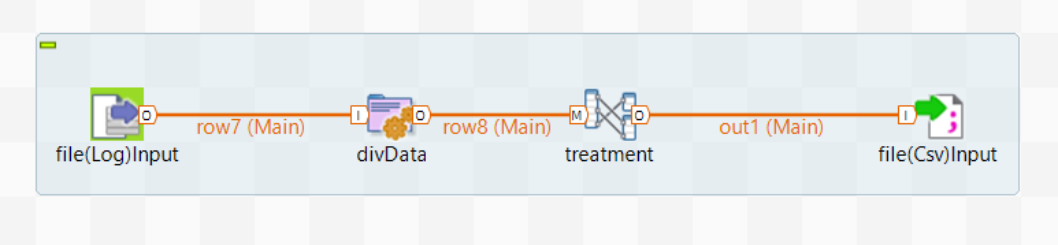


Figure 21 : Talend Job for CSV Generation /Bonita

This figure (21) is the equivalent of Figure 17 but this time, it's for Bonita logs

Figure 22 : Talend Job For Elasticsearch Uploading/Bonita

This figure (22) is the equivalent of Figure 18 but this time, it's for Bonita logs

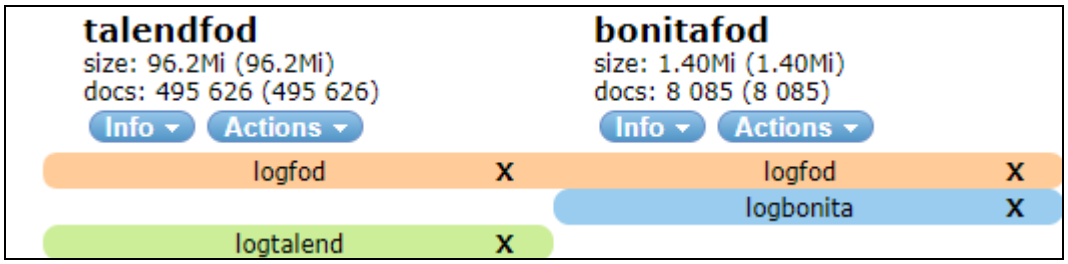


Figure 23 : Elasticsearch Indexes

This figure shows the indexes created after the uploading is done.

In this case, we have put each log file type in a specific index which is not required.

4.2.3 Sprint 3:

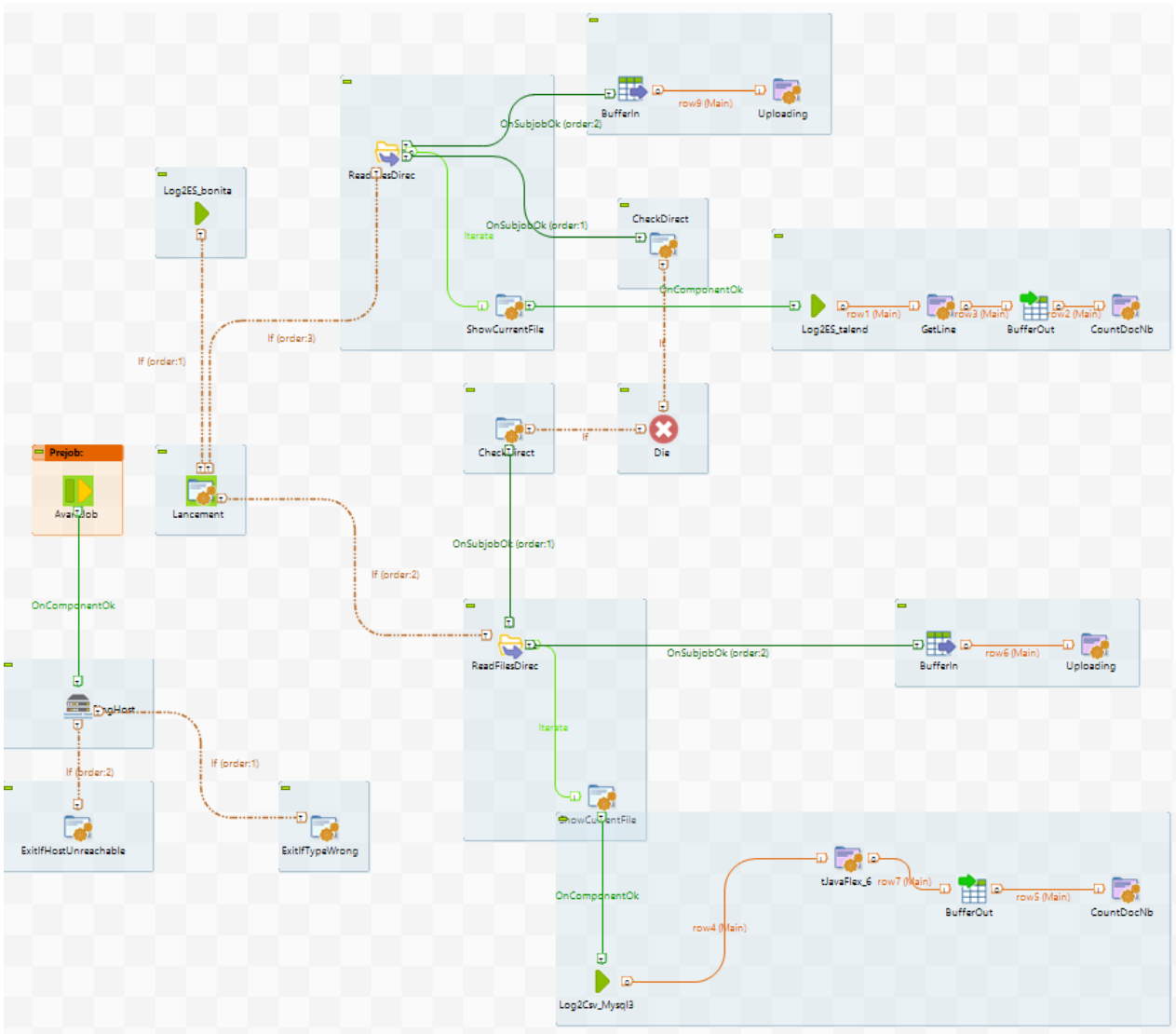


Figure 24 : Talend Master Job for I-MONITOR

The figure above illustrates the whole completed Talend Job for treating Talend/Bonita/MySQL files and uploading them to Elasticsearch.

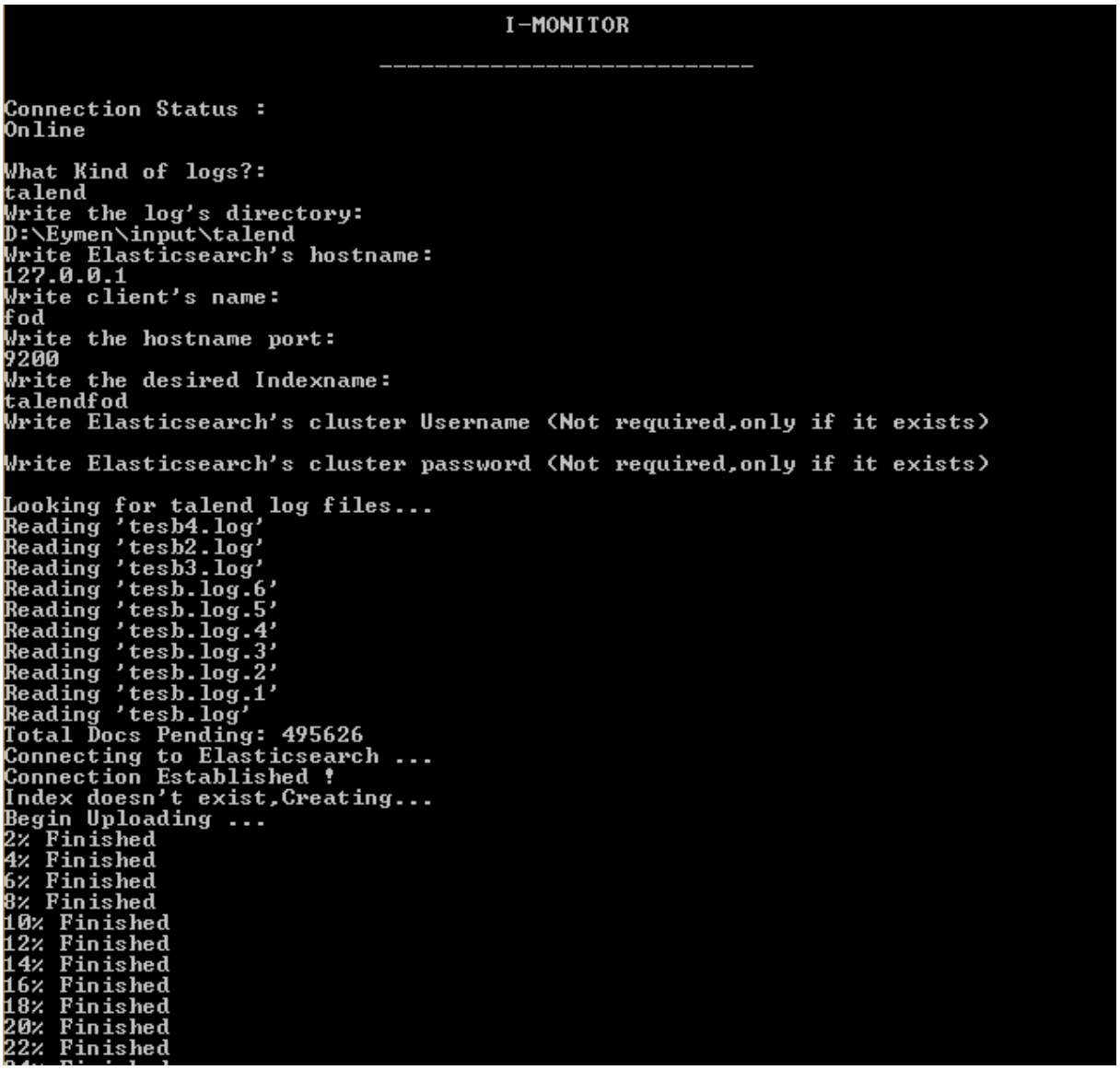


Figure 25 : I-Monitor Test Case with Talend Log Files

This figure demonstrates a test case of our application in which we have a provided the required entries and kept track of the uploading process.

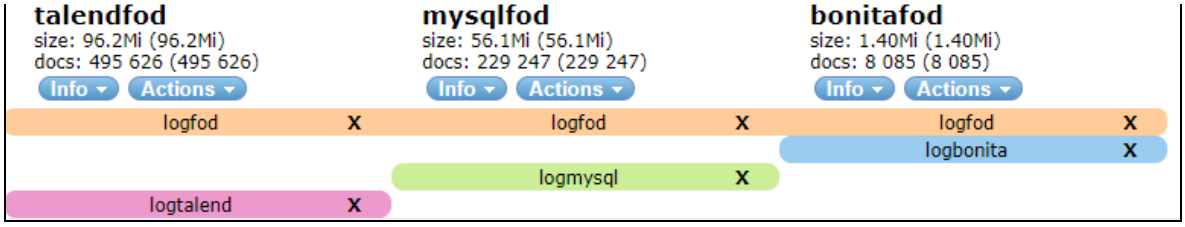


Figure 26 : Elasticsearch Indexes (Sprint 3)

The figure above shows 3 Elasticsearch indexes', each one was dedicated for a log type.

4.2.4 Sprint 4:

1–50 of 293

Time	Severity	Info1	Line	Filename	CreationDate	TimeStamp
September 17th 2018, 20:23:29.553	ERROR Mysql	Could not use /opt/log/mysql/log-slow-queries.log for logging (error 13). Turning logging off for the whole duration of the MySQL server process. To turn it on again: fix the cause, shutdown the MySQL server and restart it.	6,426,971	mysqld.log	August 15th 2018, 01:00:00.000	17:11:22
September 17th 2018, 20:03:15.818	ERROR Mysql	Could not use /opt/log/mysql/log-slow-queries.log for logging (error 13). Turning logging off for the whole duration of the MySQL server process. To turn it on again: fix the cause, shutdown the MySQL server and restart it.	6,426,971	mysqld.log	August 15th 2018, 01:00:00.000	17:11:22
September 17th 2018, 20:03:15.818	ERROR Mysql	Got error 147 when reading table './ODS/ValueSet'	6,426,941	mysqld.log	August 15th 2018, 01:00:00.000	15:40:40

Figure 27 : Kibana Data Table

This figure shows a Kibana data Table which illustrate several information about the stored data in Elasticsearch.

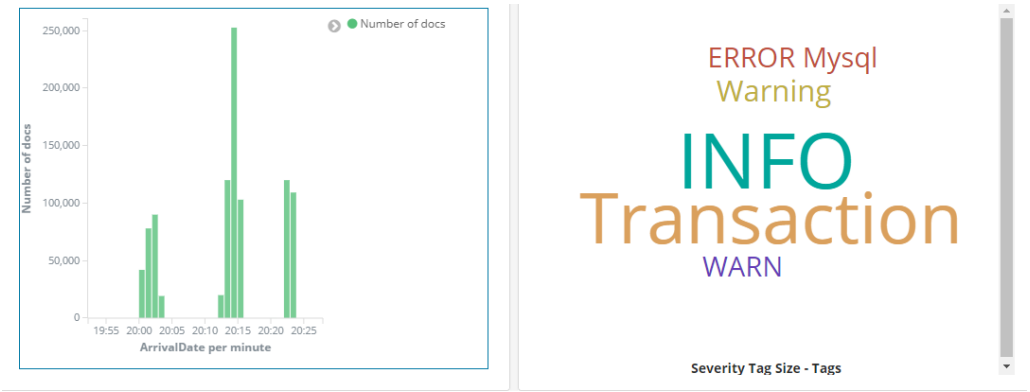


Figure 28: Kibana Tag Cloud and Vertical Bar

This figure illustrate two visualization:

The left one is a Tag Cloud which specifies the data's severity. The importance of each tag is shown with font size or color

The right one is a Vertical bar which shows the Number of documents arrived per minute

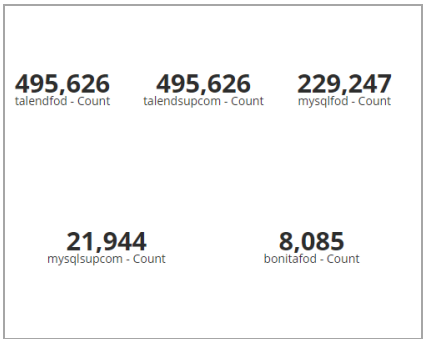


Figure 29 : Kibana Data Metric

This Kibana visualization highlight the number of documents stored in Elasticsearch ordered by index name.

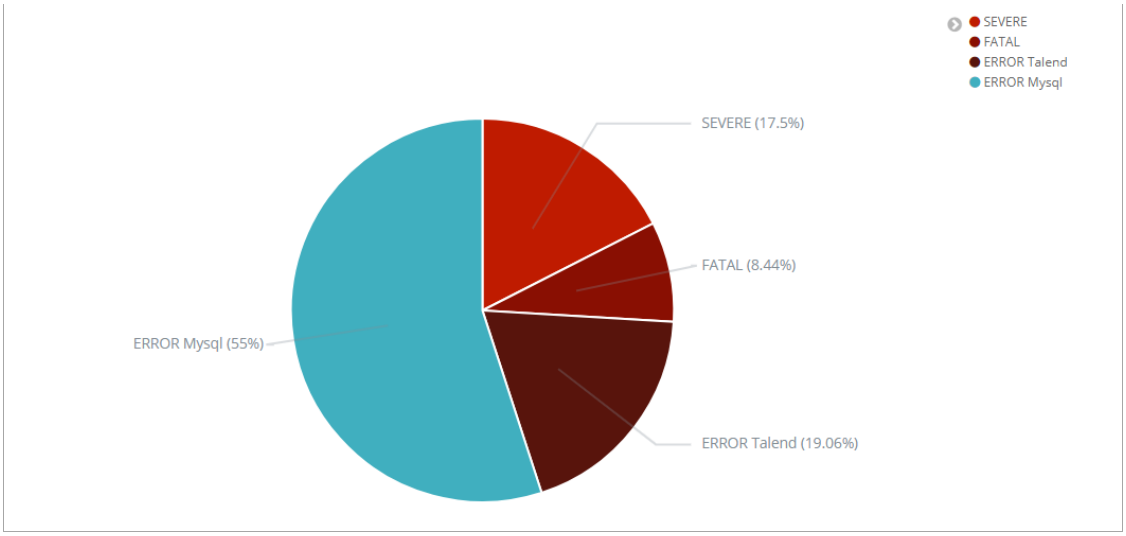


Figure 30 : Kibana Pie Chart for ERRORS

The figure above is a Pie Chart which display Error rate in Talend/MySQL/Bonita log files.

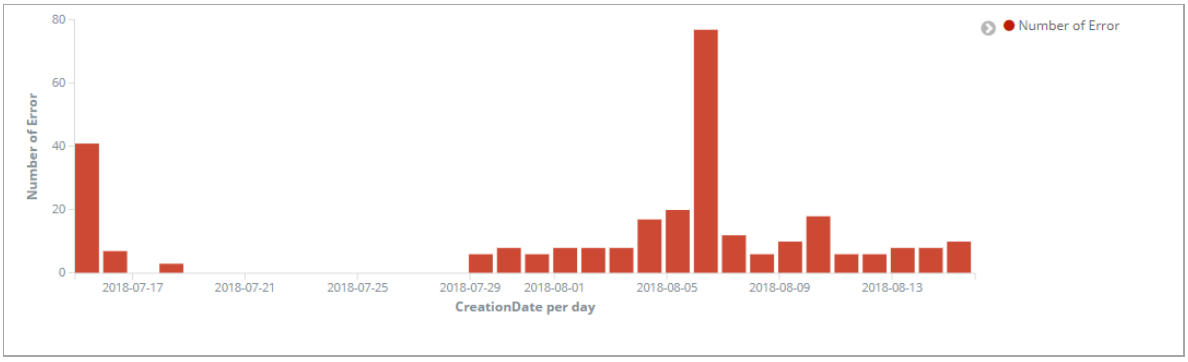


Figure 31 : Kibana Vertical Bar for ERRORS Number

This figure displays a Kibana vertical bar which highlights the number of errors in log files per day.

4.3 Conclusion

During this chapter, we have revealed some screenshots of the achieved work and discussed their operation.

General Conclusion

System's Monitoring become an indispensable activity in the world of IT. In fact, there is no doubt that our solution is recommended for any business that wants to keep things simple and save efforts. Providing metrics, charts, and other KPIs has turned out to be very useful and saves time.

In addition, the technologies evoked through this project are still young and can benefit from many innovations. The deployment of the log monitoring platform is modular according to the size that one wishes to analyze. The set of components that form the analysis chain is scalable horizontally. The implemented solution is composed only of open source products, so there is no cost of License to predict.

The internship allowed us to familiarize ourselves with professional life, to exploit fundamental notions in data analysis, and to deepen our theoretical knowledge acquired at Sup'Com.

During the internship, we spent a lot of time in the research, the documentation and the design of this project. This application was beneficial to us because, it allowed us to become familiar with several new platforms and technologies such as Talend, Elasticsearch and Kibana.

This tool remains open for extension and improvement specially to integrate the security aspect. The realization of this project does not mean complete improvement. For example, it is possible to extend this project by making the tool operate in real time or add other log types to widen the tool feature …. From an ergonomic point of view, this application could also be

improved by designing richer interfaces that have more advanced graphic components.

References

[1] VALDMAN.Jan : Log File Analysis PhD Report ,University of West Bohemia in Pilsen Department of Computer Science and Engineering, Univerzitni 8 , 30614 Pilsen ,Czech Republic
www.kiv.zcu.cz/site/documents/verejne/vyzkum/publikace/technicke-zpravy/2001/tr-2001-04.pdf (August 2018)

[2] www.elastic.co/guide/en/elasticsearch (July 2018)

[3] www.elastic.co/guide/en/kibana/ (July 2018)

[4] www.elastic.co/guide/en/elasticsearch/client/java-rest/ (July 2018)

[5] https://en.wikipedia.org/wiki/Log_file (August 2018)

[6] www.talendforge.org(July 2018)

[7] https://en.wikipedia.org/wiki/Batch_file (August 2018)

[8] T.G. PHAM (s1164163) INTERNSHIP REPORT USE OF IEC 61850 FOR ASSET MANAGEMENT IN MSc Telematics, EEMCS (July 2018)

[9] Google Web Directory: "A list of HTTP log analysis tools" http://directory.google.com/Top/Computers/Software/Internet/ /Site Management/Log Analysis (August 2018)

[10] J. H. Andrews: "Theory and practice of log _le analysis." Technical Report 524, Department of Computer Science, University of Western Ontario, May 1998. (August 2018)

[11] J. H. Andrews: "Testing using log _le analysis: tools, methods, and issues." Proc. 13 Th IEEE International Conference on Automated Software Engineering, Oct. 1998, pp. 157-166. (August 2018)