INTELLIGENT MONITORING SYSTEM

BY

YAP JIA WEI

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology
(Perak Campus)

MAY 2018

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: ____

____

____

Academic Session: _____________

I ____

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_________________________ _________________________

(Author's signature) (Supervisor's signature)

Address:

____

____

____

____

Supervisor's name

Date: ____________________ **Date**: ____________________

INTELLIGENT MONITORING SYSTEM

BY

YAP JIA WEI

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology
(Perak Campus)

MAY 2018

DECLARATION OF ORIGINALITY

I declare that this report entitled "**INTELLIGENT MONITORING SYSTEM**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ___________________________

Name : YAP JIA WEI

Date : 20 August 2018

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and appreciation to my supervisor, Dr. Ooi Boon Yaik for this opportunity to provide invaluable guidance all the way until completion of this project. To my parents for their unconditional love, support and constant encouragement, for without them I would not be where I am today. Finally I would like to extend my sincere thanks to all who have directly or indirectly aided me in the completion of this project.

ABSTRACT

This project aims to visualize machine utilization layout in the lab. The machine utilization is determined by user interactivity with the machine. Currently, information about user interactivity with machine is missing from existing monitoring solutions. Existing monitoring solutions unable to visualize about machine utilization. In this project, user interactivity and machine power mode information will be collected from the machine. The collected information will be used to visualize machine utilization of all machines in the lab. The process involving collecting data, storing data, data uploading, data processing, data migration, data analysis and visualization. Data will be collected from all machines in the lab and stored locally. If network is available, data will be uploaded into centralized online database. After collection of data period, entire database is extracted out as a JSON file. The extracted file will be processed and imported into local held database server. The data in local database server will be analyzed and eventually visualized. The visualization of machine utilization shows which machine is mostly and least used.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	Application Programming Interface
BIOS	Basic Input/Output System
CPU	Central Processing Unit
FTP	File Transfer Protocol
GB	Gigabyte
GPU	Graphics Processing Unit
HTTP	Hypertext Transfer Protocol
ID	Identification
IDE	Integrated Development Environment
IMAP	Internet Message Access Protocol
IT	Information Technology
JSON	JavaScript Object Notation
MB	Megabytes
OS	Operating System
POP3	Post Office Protocol Version 3
SSH	Secure Shell
URL	Uniform Resource Locator

1.1 Motivation and Problem Statement

Level of machine utilized by students in the labs is remain unknown. No one knows which machines in the labs are the most favour, the least favour as well as average favour among students. There is a missing information about how often the machine is utilized by students. Students access machines in the labs based on their own preferences for instances, machine position in the lab and machine performances. The students then utilize these selected machines to process their stuffs. As a result, this leads to different machine utilization levels within the labs. Some machines have higher utilization level compared to other machines because of some of the machines in the labs are more favourable within the students whereas some might not. In this case, the least used machines are not well utilized to their expected value.

On the other hand, information about power mode of machine is also remain unknown. Information about power mode of machine could be used with machine utilization by students to show whether there is someone using the machine or not. Sometimes the machine is powered on however no one is utilizing it. The machine is wasting power and resources. In this case, the machines should be considered as low level of utilization since it is power on without any purpose and user interactivity.

In the long terms, the highly utilized machines in the labs are usually get worn faster than usual and more likely have to be replaced with new ones. However, the least utilized machines are almost remain as new condition. In fact, the machines could be constantly swapped to force all machines getting uniform utilization rate in the lab. Hence, machines user interactivity level and machine power status are captured to keep track of the utilization rate of all machines.

1.2 Project Scope

The project will be delivering 1 system and analysis and visualization of collected data. A monitoring system will be developed and implemented into all lab machines. The system has 4 main tasks:

1. Capturing user interactivity with machines.
2. Capturing machine power status.
3. Capturing machine resources performances.
4. Uploading data into online centralized database.

The collected data will be analysed and visualized in graphic charts. The following shows main graphic charts will be displayed:

1. Utilization of user interactivity over power on mode of all machines.

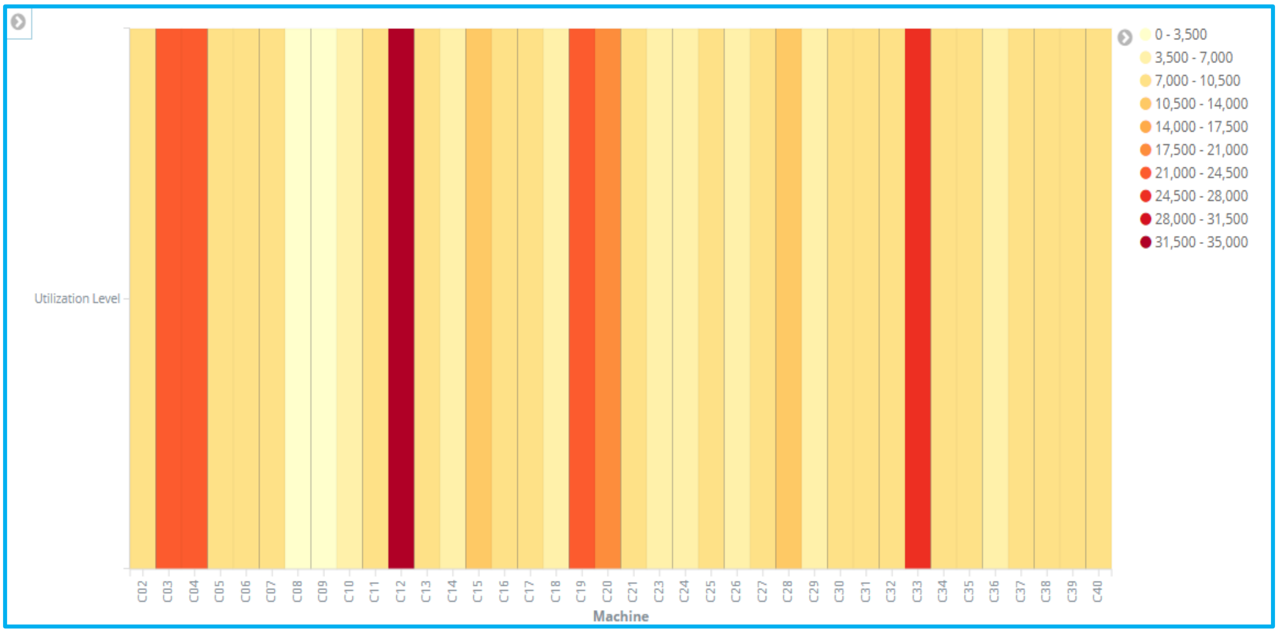


Figure 1.1 Example of Heat Map of utilization level of all machines

1.3 Project Objectives

1. To collect information of user interactivity with machines.
2. To collect information of power mode of machines.
3. To analyse the relationship between user interactivity with machines and power mode of machines.
4. To visualize the relationship between user interactivity with machines and power mode of machines.
5. To visualize machine utilization layout in the lab by using information of user interactivity with machine.

1.4 Project Contributions

From capturing user interactivity with machine and machine power status to analyse and visualize the level of machine utilization, highly and least utilized machines could be identified. The highly utilized machines are getting worn faster than usual and hence they more likely to be replaced with new ones however least utilized machines are not utilized to its actual value. Therefore, highly and least utilized machines can be swapped manually with each other to make sure the even utilization rate over all machines in the labs despite different students' preferences in accessing machines. This also cut down unnecessary hardware equipment expenditures in purchasing new equipment to replace fast getting worn equipment.

In addition, machine utilization layout in the labs could be visualized out relative to the time. Overall machines utilization layout is understood with respective of time. Therefore, one can get understand about the overall machine utilization of lab across the time. By using this information, the need of creating new labs is determined since if the overall machine utilization of lab is underutilized, there is unnecessary to create a new lab.

2.1 Review on Existing Machines Monitoring Software

2.1.1 Review on Zabbix

According to a Zabbix (2016) documentation, Zabbix responsible in the network and the health and integrity of servers monitoring. Nevertheless, Zabbix could be used to extend its functionalities to monitor desktops with provided and user defined parameters. Based on Zabbix (2016) documentation, Zabbix is a software which mainly provide monitoring and reporting solutions. It is monitoring many items however lacking of specific important aspects of parameters in case of proposed problems. It collects numbers of items which is machine resources performance data however without rate of user interactivity with machines to indicate how often the machines is been used. Zabbix monitoring solution is comprised of server, database storage, proxy and agent. The server is responsible for the agents report availability and integrity information and statistics, database storage is responsible for data storage of collected data and agent is responsible for collected and report the collected data to the server. In our case, only 2 ends are involved in the monitoring solution which is the agent to monitor and report the collected data and database storage to store the reported data in order to simplify the chain processes in between the solution. Zabbix able to visualize the stored data in graphs format by using the collected data. It visualize the data with simple and custom graphs. Both of the graphs only show the data relative to the time without the ability of computing several types of data into one new type. In proposed solution, both data of user interactivity and machine power status are computed and visualized to give information of level of machine utilization in terms of how often user is interacting with the machine and is the machine is in idle mode without user interacting for each machine.

2.1.2 Review on Nagios Core

According to a Nagios Core (2016) documentation, Nagios Core able to monitor IT infrastructure components, including system metrics, network protocols and services. Nagios Core is easy to configure to monitor private services and attributes of machine. For example of private services, memory usage, CPU load, disk usage and services state of machine. It emphasizes on its detailed monitoring in public services include HTTP, POP3, IMAP, FTP, and SSH protocols of machines. It basically monitors almost all attributes of machine however without collecting rate of user inputs from keyboard and mouse to indicate user utilising the machine. In visualization, Nagios Core does not has the ability to analyse and visualize the data into graphical manners, it relies on plugins to include the visualization feature into its own framework. It is designed to return and pass collected data to external applications for processing and visualizing the data. Thus, it unable to analyse and compute the collected data into more useful information and visualize the information in graphical format.

2.1.3 Review on Solarwinds

According to Solarwinds (no date), Solarwinds also able to monitor main 4 metrics of machine including CPU utilization, physical memory consumption, virtual memory and disk performance. In fact, these metrics are able to be monitored in many existing solutions since these performance metrics are basic attributes of machine. Solarwinds also could be configured to monitor many other metrics of machine yet information of user interactivity with machines and power status mode of machine are missing to be computed and visualized to show level of machine utilization.

2.1.4 Review on PRTG

According to PRTG (no date), types of performance metrics could be monitored are CPU utilization and operating temperature, memory usage, network bandwidth, free hard disk space and system information. This shows that PRTG provides machine performance metrics monitoring like other monitoring solutions do. PRTG is lacking of monitoring user utilization with machine and machine power mode to keep track of how often the machine is been used instead of only how machine is behaving. Most of the existing monitoring solutions provide and keep track the trends of machine resources performance data without providing user utilization level with machine.

2.1.5 Review on OpManager

According to OpManager (no date), basic machine performance metrics could be monitored. It mainly monitor metrics which is CPU, memory, hard disk and network.

It keep tracks of the behaviours of machine by using these information. On the other hand, user interactivity with machine and power mode of machine cannot be collected from the OpManager. Thus, machine utilization level by users is unknown in this solution.

.

2.2 Comparisons between Existing Machines Monitoring Software in the Market

Feature / Software	CPU utilization monitoring	Memory monitoring	Hard disk monitoring	Network bandwidth monitoring	Keyboard input rate monitoring	Mouse input rate monitoring	Machine power mode monitoring	Visualization of machine utilization by user
Zabbix	✓	✓	✓	✓	✗	✗	✗	✗
Nagios	✓	✓	✓	✓	✗	✗	✗	✗
Solarwinds	✓	✓	✓	✓	✗	✗	✗	✗
PRTG	✓	✓	✓	✓	✗	✗	✗	✗
OpManager	✓	✓	✓	✓	✗	✗	✗	✗

Table 2.1 Comparisons between Existing Computers Monitoring Software

From table 2.1, all 5 solutions could be concluded that not providing a monitoring utilization solution as the proposed solution.

3.1 Design Specifications

3.1.1 Methodologies and General Work Procedures

In this project, there are 5 main steps, software development, software implementation, data migration, analysis and visualization of data. In order to monitor the machines and user interactivity with machines in the lab environments, a Windows service application and Windows application will be developed under .NET framework by using C# language. Both applications will be implemented within every machine in lab with running as monitoring solution to capture defined metrics and user interactivity with machine. The applications will be distributed into each machine in all installed Windows OSs by using Windows batch file to automate the installation of Windows service and application.

Data migration takes place after data in machines is uploaded into Firebase. Whole Firebase will be exported as a JSON file and the file will be underwent data processing before import into Elasticsearch local database by using Javascript. Eventually, analysis and visualization of data is performed by using Kibana.

3.1.2 Tools to Use

Software

1. Visual Studio 2013 Ultimate
   - IDE platform to develop Windows C# service application
2. Windows batch script
   - Automation of installing Windows C# service application into machines in multiple OS
3. Firebase
   - Central data storage
   - Export whole database as JSON file
4. Javascript
   - Data processing before import into Elasticsearch database
   - Automation of importing the JSON file which exported from Firebase into Elasticsearch database

5. Elasticsearch
   - Data storage for analysis and visualization
6. Kibana
   - Analyse and visualize data from Elasticsearch

Hardware

1. Laptop
   - The device for application development

Model	Asus X550LD
Operating System	Windows 10 Home
Processor	Intel Core i5-4200U
Memory (RAM)	8GB
Graphic Card	NVIDIA GeForce 820M & Intel(R) HD Graphics Family
.Net Framework	Version 4.5

Table 3.1 Specifications of Laptop

2. Desktop Computers
   - Machines to be monitored in the lab

3.1.3 System Performance Definition

1. Environment Compatibility
   - The service shall be compatible to execute in .NET framework version of computers in the labs.
2. Data Accuracy
   - Collected data from defined monitoring parameters should be accurate with the time.
3. Data Collection Interval
   - The event of collecting the data is fired with the correct and precise time interval.
4. Data Upload
   - Data to be synchronised with the central data storage is intolerant of missing and incomplete.
5. Multi-tasks Ability

- Tasks to collect and store data, and upload data shall be underwent concurrently without interfering each other processes.

3.2 System Design/Overview

3.2.1 System Architecture

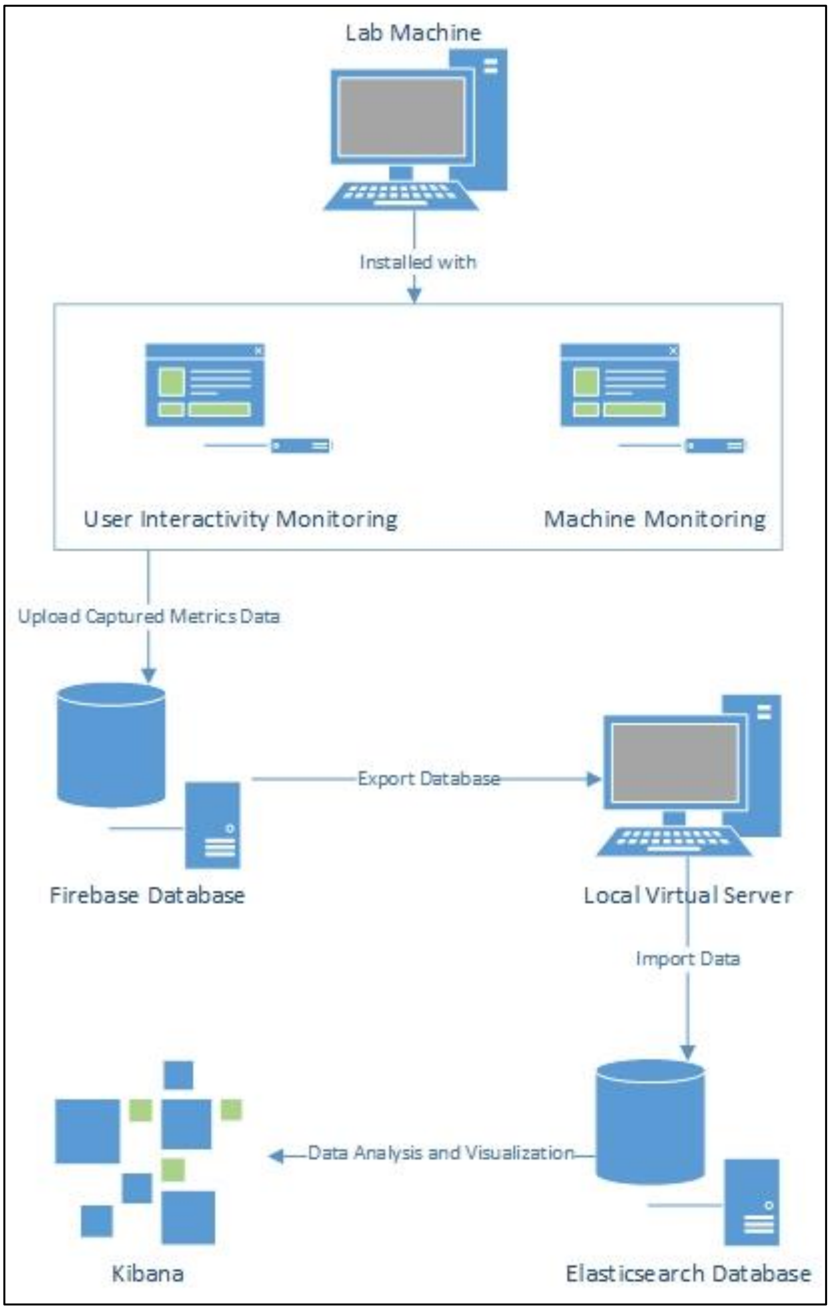


Figure 3.1 System Architecture

All lab machines are installed with 2 software which is user interactivity and machine monitoring. User interactivity monitoring is developed as Windows C# application without user interface and executing in background without interfering with user desktop windows. Machine monitoring is developed as Windows service application. Both of the software is designed to be long-running as long as the OS is not shut down, the software is stopped once the OS is shutting down. These 2 software will be installed into lab machine to perform capturing data steps.

These 2 monitoring solutions will upload captured data into Firebase online real-time database whenever the network is available and machine OS is on.

Once achieving the collecting data period and Firebase is comprised of data, data migration is carried out. All data inside Firebase will be exported out manually as JSON type file. After that, the exported JSON file is imported into locally held Elasticsearch database server with the help of Javascript to perform data processing before importing and import.

Finally, data in Elasticsearch is used to analyse and visualization to achieve the objectives.

3.2.2 Machine Monitoring Service

3.2.2.1 Machine Monitoring Start-up Flowchart

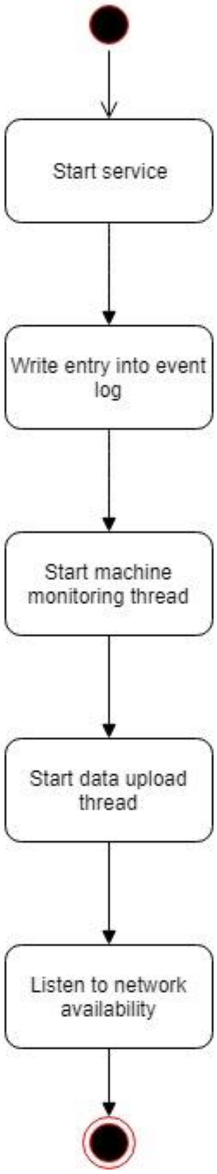


Figure 3.2 Machine Monitoring Service Start-up Flowchart

The above diagram is showing about flowchart of machine monitoring service when started by the OS system. The start-up type of service is configured as automatic so that the machine monitoring process is begin whenever the OS is ready to capture metrics as soon as possible. Entry in log represents the start-up time of service.

There are 2 threads will be executed in parallel manner. One is responsible for machine monitoring and another thread is for data uploading task.

Network availability is crucial in data uploading task, therefore service is registered to listening any network is up and connected. The service will respond whenever the network is available to perform data upload task.

3.2.2.2 Machine Monitoring Thread Flowchart

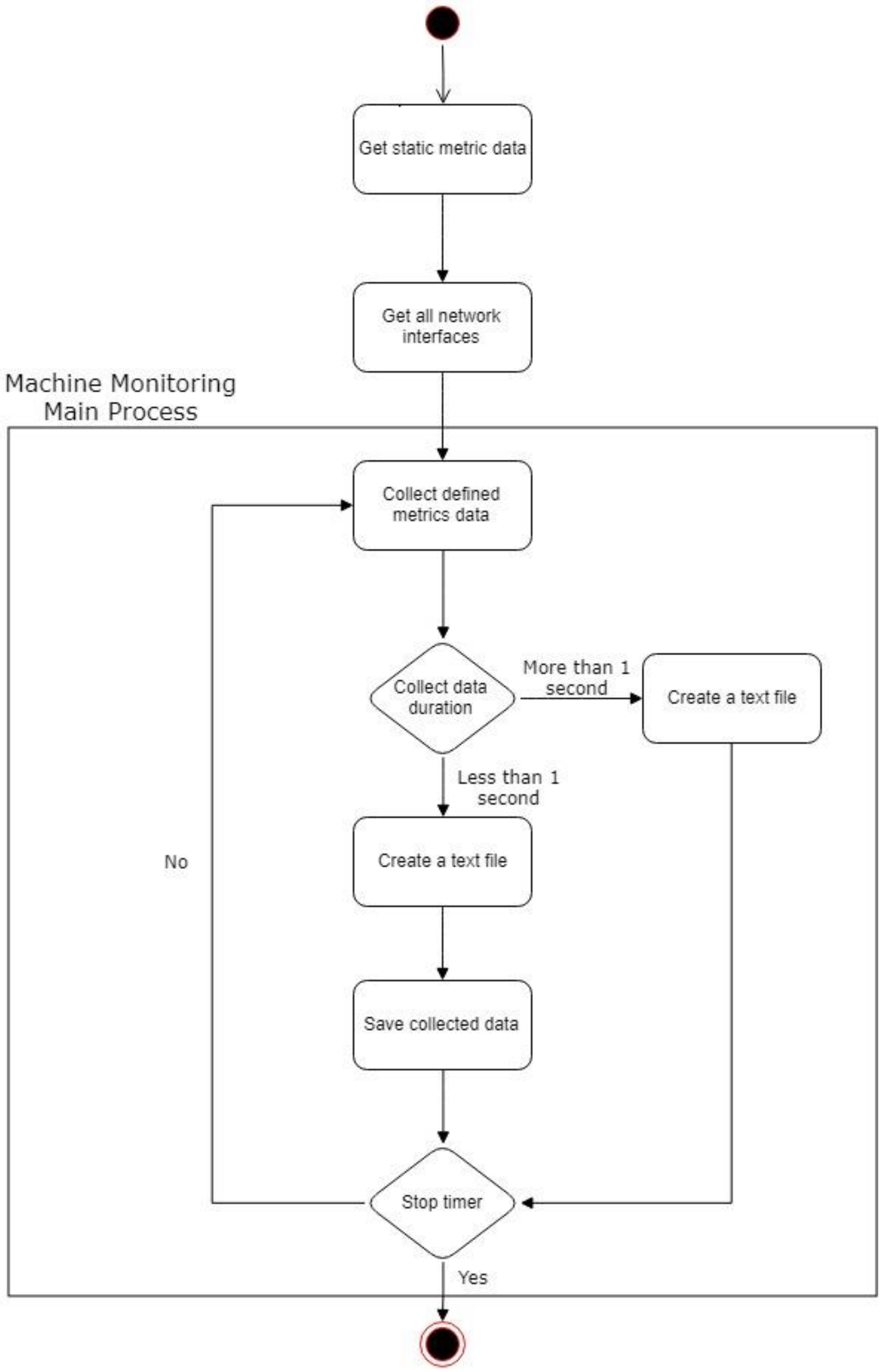


Figure 3.3 Machine Monitoring Thread

Above is showing that flowchart of machine monitoring thread which is initiated after service start-up. The thread is important as part of the monitoring process. Since monitoring is done through iteration of capturing metrics, machine monitoring main process is fired with timer set with 1 second time interval. The process will be repeated every 1 second to execute monitoring steps. The duration of collecting data is limited within 1 second. Since time interval of each monitoring process is 1 second, the collect data duration must be within 1 second in order to ensure all metrics data is accurate relative to the time without data interfere with next monitoring process. If the duration is more than 1 second, no data will be stored and an empty text file will be created with the fired time as name of file.

3.2.3 User Interactivity Monitoring Application

3.2.3.1 User Interactivity Monitoring Start-up Flowchart

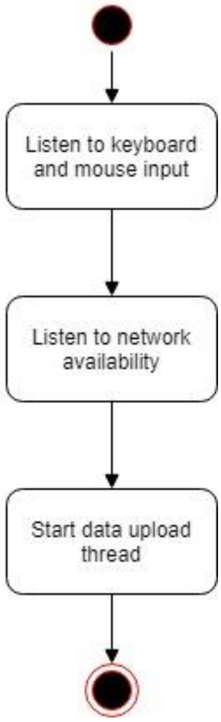


Figure 3.4 User Interactivity Monitoring Start-up Flowchart

As a result of successfully started machine monitoring service, the user interactivity monitoring application will be started by the machine monitoring service as well. The application is started with administrator permission to listen to any keyboard and mouse input from the user. The administrator permission is required to overcome any other program is executing with administrator permission. In case of other program is executing with administrator permission, the application is unable to listen to input inside that program.

3.2.3.2 User Interactivity Monitoring Main Process Flowchart

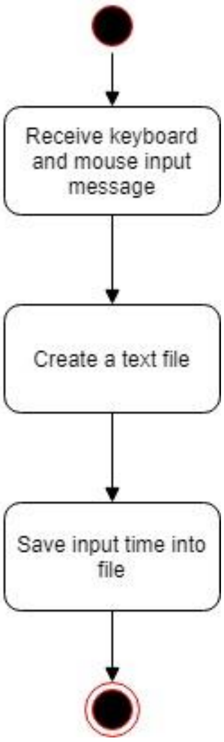


Figure 3.5 User Interactivity Monitoring Main Process Flowchart

In order to receive keyboard and mouse input message, application is registered with keyboard and mouse raw input to listen any input message from the user. Raw input will send out an input message to notify there is an input occurs and the input message time is recorded to indicate there is user interacting with the machine at that moment.

3.2.4 Data Uploading Flowchart

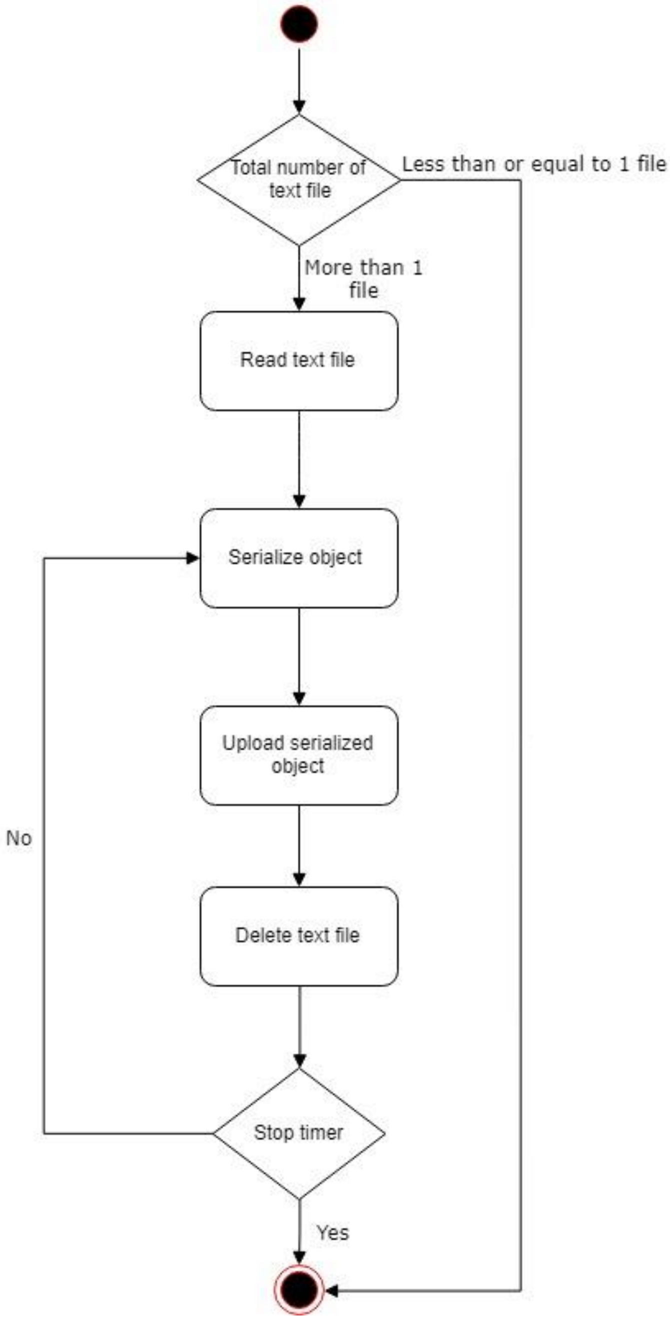


Figure 3.6 Data Upload Flowchart

Steps to perform data upload task is identical for both machine and user interactivity monitoring. The data upload thread is fired through a timer with defined time interval to ensure the uploading process is keep going whenever the network is available without storing huge amount of text file locally. Once the file is uploaded successfully and getting unique ID which generated by Firebase, the file will be deleted to save up the storage space.

3.2.5 Firebase Database Structure

2 Firebase real-time databases are created due to the constraints of storage of one Firebase real-time database. In order to create 2 databases, 2 projects are added.

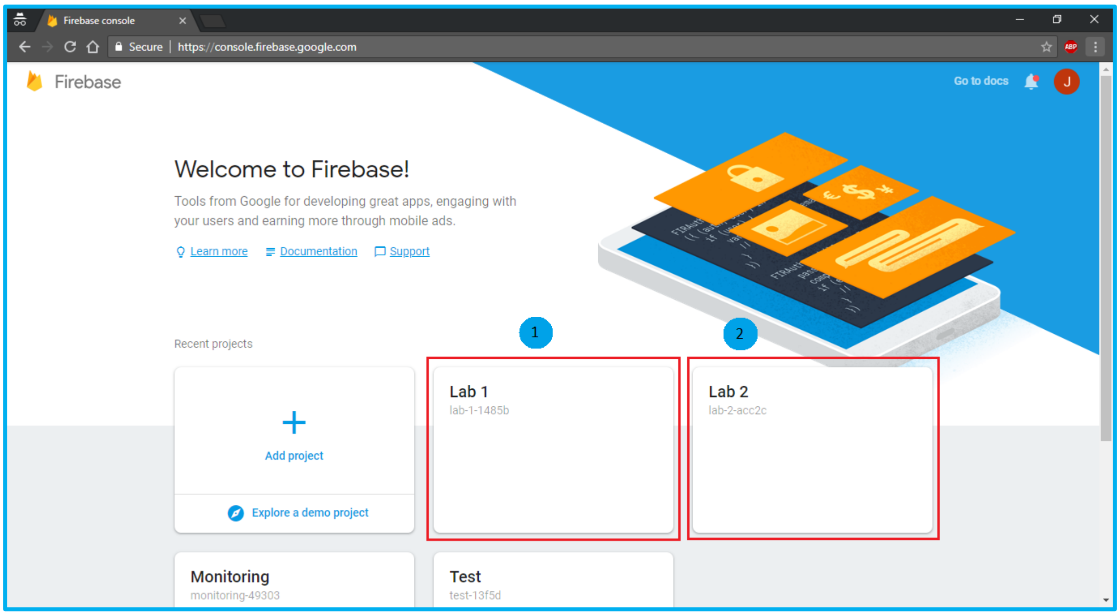


Figure 3.7 Create Firebase Project

Inside project, one real-time database can be created under database tab.

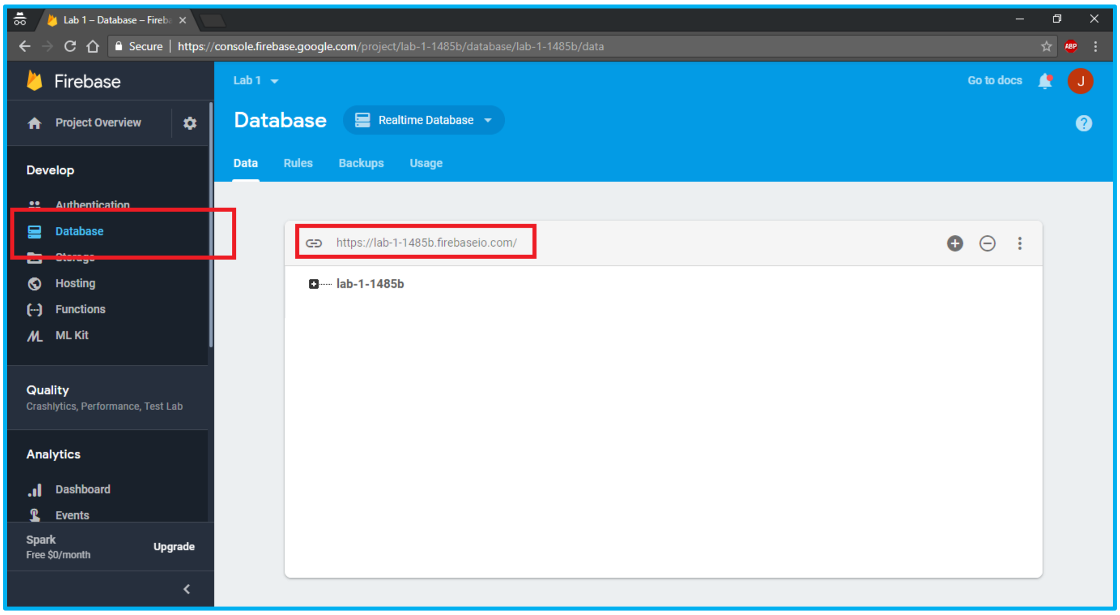


Figure 3.8 Create Real-time Database

A real-time database URL is generated after created database. The URL is used to access the database to insert data. The URL is permanent and unchangeable.

Each operating system uses different URL to access the database to prevent exceeding total storage allowed. Each URL offers 1 GB storage size.

Database URL	Operating System
https://lab-1-1485b.firebaseio.com/.json	Windows 7 x64 and Windows 10 x64
https://lab-2-acc2c.firebaseio.com/.json	Windows 7 (Oracle) and Windows 7 (SAS)

Table 3.2 Database URL of Operating System

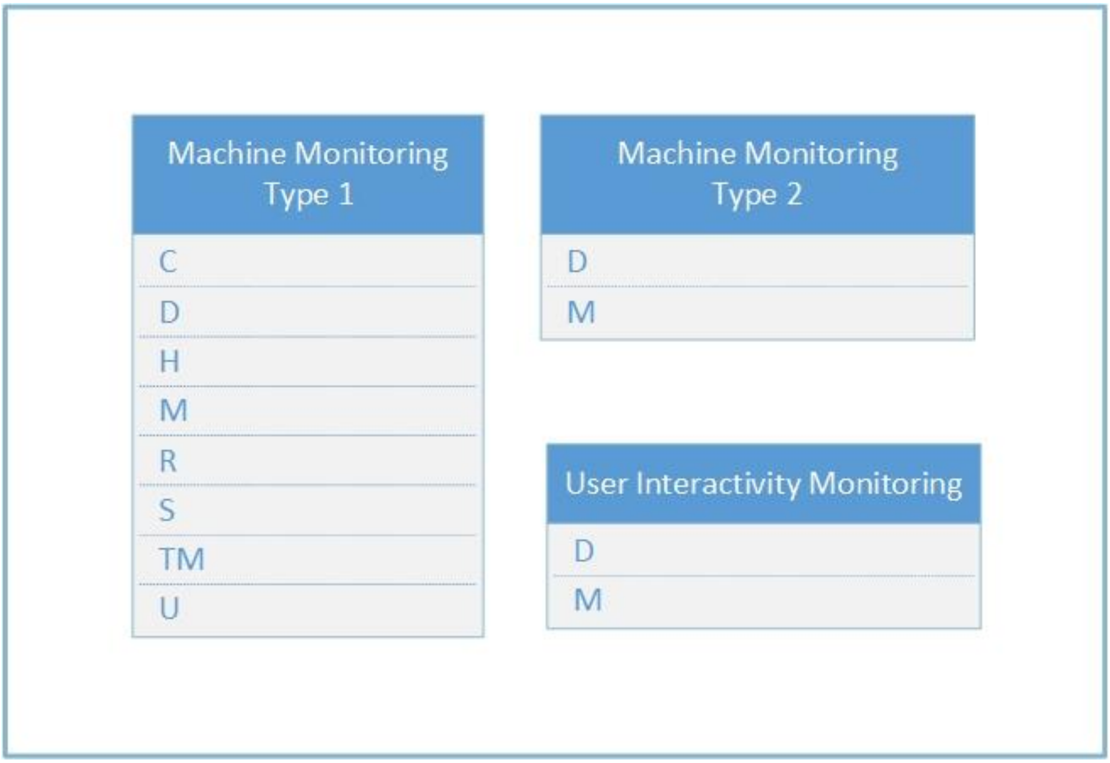


Figure 3.9 Object Types Stored In Firebase

Data stored in Firebase is in JSON object-type format. The database structure is a flat structure without nested object in ensuring every object is an independent data. Each object is assigned with a unique key. The database is created without any primary and foreign key.

Machine monitoring has 2 types of objects to be stored in Firebase. Machine Monitoring 1 object is used when collect data duration less than 1 second whereas Machine Monitoring 2 object is used when collect data duration is more than 1 second.

Key D is used to indicate date time when collecting data and, keyboard and mouse input happens. Key M is used to differentiate each machine inside the lab.

Every time of inserting object into Firebase, a unique key will be generated automatically by Firebase to identify each object.

Key Name	Description
C	CPU utilization %
D	Date time in milliseconds format
H	Hard disk utilization %
M	Machine name
R	Received network (bytes)
S	Sent network (bytes)
TM	Total physical memory (bytes)
U	Physical memory in use (bytes)

Table 3.3 Keys in Database Objects

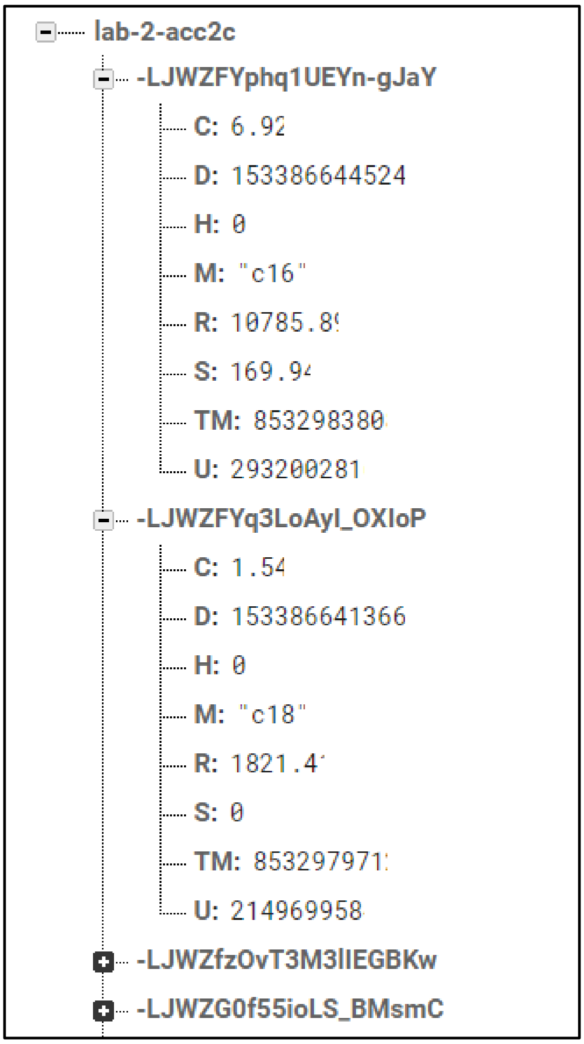


Figure 3.10 Structure of Firebase

3.2.6 Elasticsearch Database Structure

Elasticsearch configurations are performed by using console tab in Kibana.

Figure 3.11 Command to Create Index

The above command is to create a new index in Elasticsearch database.

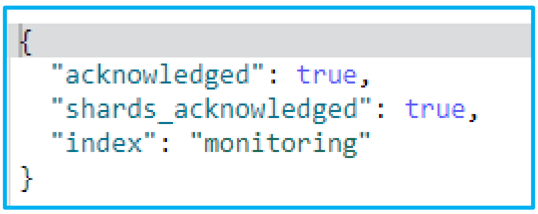


Figure 3.12 Success Result Returned From Creating Index

```
PUT monitoring/_mapping/all
{
  "properties": {
    "C" : {
      "type": "float"},
    "D" : {
      "type": "date"},
    "H" : {
      "type": "float"},
    "M" : {
      "type": "text",
      "fields": {
        "keyword": {
            "type": "keyword",
            "ignore_above": 256}
                }
          },
    "R" : {
      "type": "float"},
    "S" : {
      "type": "float"},
    "TM" : {
      "type": "long"},
    "U" : {
      "type": "long"},
    "I" : {
      "type" : "integer"}
  }
```

Figure 3.13 Command to Create Fields Mapping

```
{
  "acknowledged": true
}
```

Figure 3.14 Success Result Returned From Putting Mapping

The fields mapping allows to define fields and field type of each document in created index. Command in Figure 3.14 puts mapping in monitoring index under all type.

All fields in Elasticsearch is same as keys in Firebase except extra field named I in Elasticsearch. Field I is used to indicate user interactivity with the machine, value of 1 indicates there is a keyboard or mouse input at that time.

```
{
  "_index": "monitoring",
  "_type": "all",
  "_id": "24",
  "_score": 1,
  "_source": {
    "C": 4.23,
    "D": 1533542258190,
    "H": 1.13,
    "M": "C11",
    "R": 61.12,
    "S": 0,
    "TM": 8532983808,
    "U": 1712402432
  }
},
{
  "_index": "monitoring",
  "_type": "all",
  "_id": "25",
  "_score": 1,
  "_source": {
    "D": 1533542255042,
    "M": "C11",
    "I": 1
  }
},
{
  "_index": "monitoring",
  "_type": "all",
```


Figure 3.15 Structure of Elasticsearch

4.1 Implement Machine and User Interactivity Monitoring

```
@echo off
echo Installing service.
@SET INSTALLUTILDIR=
@for /F "tokens=1,2*" %%i in ('reg query "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft^
                            \NET Framework Setup\NDP\v4\Full" /v "InstallPath"') DO (
    if "%%i"=="InstallPath" (
        SET "INSTALLUTILDIR=%%k"
    )
)

%INSTALLUTILDIR%\installutil.exe "%~dp0Monitoring Solution.exe"
pause
net start "Quantum"
pause
```

Figure 4.1 Implement Monitoring Solutions Batch Script

Batch script is ran as administrator to automate installation of both machine and user interactivity monitoring. The script will find out the file path of .NET framework version under version 4. The file path will be used to look for installutil.exe which is a Windows service application installer to install machine monitoring as service. Installation paused after service is installed successfully, and proceeding to start service once any key is pressed from keyboard.

There are 40 machines are implemented with the monitoring solutions. Each machine has 4 different OSs. One same batch script is used to install in all machines and all Oss. Installation is took place in each OS of each machine manually.

Operating System	Number of Machine
Windows 7 x64	40
Windows 7 (Oracle)	40
Windows 7 (SAS)	40
Windows 10 x64	40

Table 4.1 Number of Installed Machine

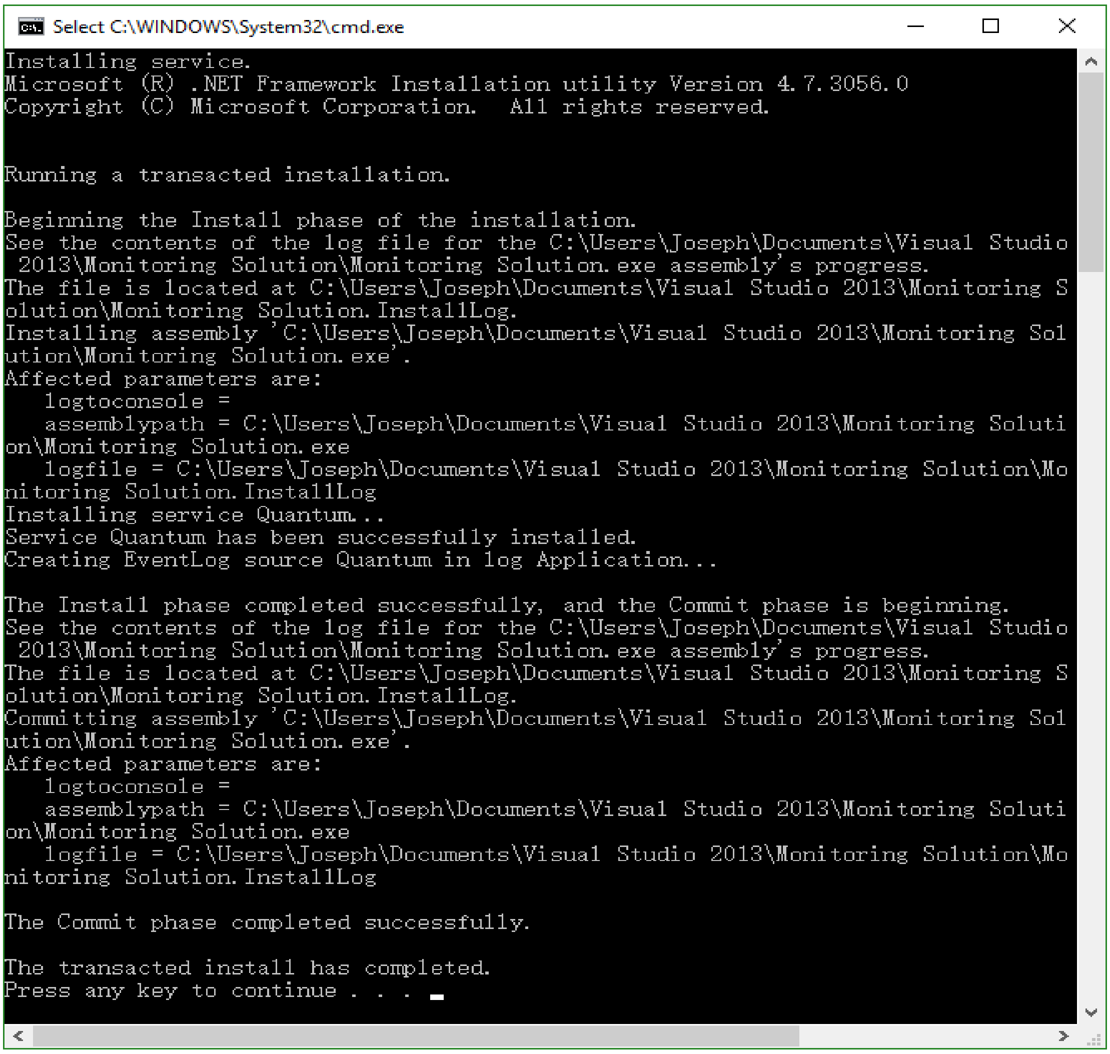


Figure 4.2 Output after Implementing Monitoring Solutions

The message about status of progress will be displayed along the installation. In the end, installation will be inform had been installed successfully. Next, pressing any key to start up the service to perform monitoring immediately.

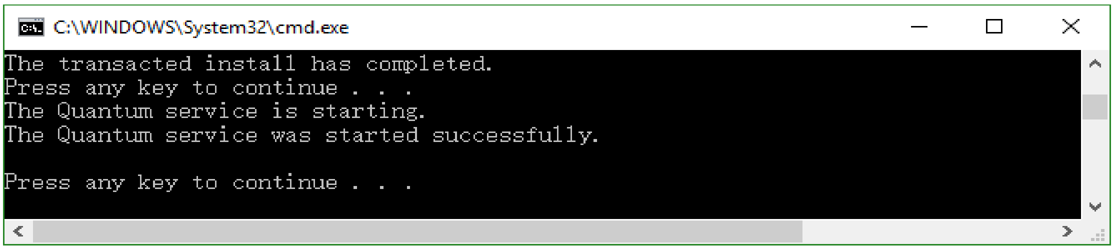


Figure 4.3 Output after Starting Machine Monitoring Service

The service will be started successfully when there is no error in the start-up phase of the service.

5.1 Data Export

Data from Firebase is expected to be exported out and import into Elasticsearch database. Whole Firebase is exported out as a huge JSON format file. Firebase real-time database is first transferred into Backup database due to Firebase is unable to directly export database as JSON file. This is because export function requires read request and read request is limited to not more than 256 MB per request. In this case, data stored in Firebase is larger than 256 MB. 2 real-time Firebase database are exported and 2 JSON files are downloaded.

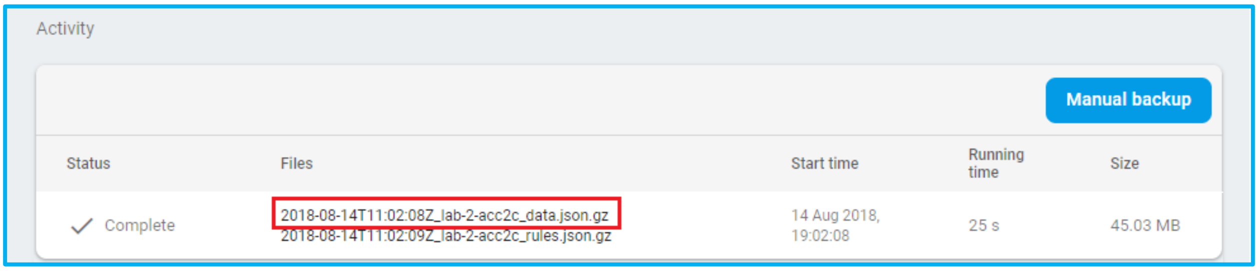


Figure 5.1 Export Firebase

JSON file which contains whole Firebase can be downloaded with much more smaller size since it is compressed with gzip compression. Above showing 2 files, data and rules of database. In this case, only data is needed to download. The downloaded JSON file is stored in local host.

```
{
  "-LIe9-9bM92DXdSzDjyX" : {
    "D" : 1532934961547,
    "M" : "c11"
  },
  "-LIe9-NY1fMqfkCAx5Bu" : {
    "D" : 1532934962004,
    "M" : "c11"
  },
  "-LIe9-TpbjJ5kJXZ8NSL" : {
    "D" : 1532934969388,
    "M" : "c11"
  },
  "-LIe9-UnPfKmAGZoBIbo" : {
    "C" : 12.31,
    "D" : 1532936728835,
    "H" : 37.35,
    "M" : "c11",
    "R" : 56.82,
    "S" : 138.78,
    "TM" : 8532983808,
    "U" : 1480577024
  },
  "-LIe9-_7uP1Wok1HlcLk" : {
    "D" : 1532934970011,
    "M" : "c11"
  },
  "-LIe9-a7gNT1GdTLiJo4" : {
    "C" : 10.38,
    "D" : 1532936729849,
    "H" : 23.72,
    "M" : "c11",
    "R" : 184.5,
    "S" : 108.87,
    "TM" : 8532983808,
    "U" : 1486663680
  },
  "-LIe9-fO54jB01E149An" : {
    "D" : 1532934971003
```


Figure 5.2 Exported JSON file

In exporting Firebase, the unique key associated with each object is exported as well.

5.2 Data Processing and Import

Data processing and import is done with the Javascript and run under node.js engine environment. A Javascript is written to automate both operations since the JSON file is consists of millions of independent objects. Data processing is performed in each object before import it into Elasticsearch index.

```
const pipeline = fs.createReadStream('./lab-2-acc2c_data_2.json')
  .pipe(JSONStream.parse('*'));
```

Figure 5.3 Read JSON file

The exported JSON file will be opened and read. The JSONStream.parse function splits and convert entire JSON file into each smaller independent objects. Since the JSON file is consist of multiple objects, one object is designed to be processed per time.

```
pipeline.on('data', function(data){
     if (!data.hasOwnProperty("C")) {
         data["I"] = 1;
     }
     total++;
     data["M"] = data["M"].toUpperCase();

     send(data, ++i);

     pipeline.pause();
     setTimeout(() => {
     pipeline.resume();
   }, 50);
});
```

Figure 5.4 Data Processing

In order to differentiate machine monitoring with user interactivity monitoring, new field named ‘I’ is created in case of user interactivity database object. The field is assigned with value of integer 1 to show there is a user input. In addition, machine name is converted into upper case to ensure consistency across all machine names in the lab.

Operation of reading data from underlying stream is performed every 0.05 seconds. Therefore, each object is read and processed every 0.05 seconds. This is to slow down rate of import, the error of too many requests is getting from local Elasticsearch server if there is no explicit restriction by using pause and resume implementation. The pause method is to stop ‘data’ events happening and results in data remains in internal buffer without reading whereas resume method is to resume ‘data’ events happening and read data from underlying stream.

After data processing, the processed object is passed into send(obj, _id) function to import into Elasticsearch database.

```
var client = new elasticsearch.Client({
  host: 'localhost:9200',
  log: 'warning',
  requestTimeout : Infinity
});
```

Figure 5.5 Establish Elasticsearch Connection

A connection to the local Elasticsearch database is established by using elasticsearch.js API.

```
function send(obj, _id){
    console.log(obj);
    console.log("ID: " + _id);
    client.create({
    index: "monitoring",
    type: "all",
    id: _id,
    body: obj
    }, function(error, response) {
    if (error) {
        if(error["status"] == 429){
            fail++;
            setTimeout(() => {
                send(obj, _id);
            }, 1200000);
        }
    }
    else{
        insert++;
        if(fail > 0)
            fail--;
    }
    console.log("Total imported records: " + insert);
    console.log("Total times to retry import: " + fail);
    });
}
```

Figure 5.6 Import Object into Elasticsearch Index

Object is designed to be imported into monitoring index under type, all. create API of Elasticsearch.js is an asynchronous method which return result of request without blocking the importing process. If error is returned from the request, the associated object will be reimport after 20 minutes. The waiting time for reimport is set to be long to protect initial import interval is not interrupted. For example, too many requests error message will be returned if reimport is performed immediately.

6.1 Data Analysis

Data in Elasticsearch index consists of 3 types of JSON objects. For the purpose of visualization which realizes the objectives, objects are analysed to provide more precise information. Duration of collecting data is 1 week, from August 6th 2018 Monday to August 10th 2018 Friday.

6.1.1 Create Scripted Fields

Before performing visualization on data, fields in mapping are computed to be transformed into new fields to aid in visualization since there are some fields consists of raw data without being handled. This results in more useful information to introduce visualization.

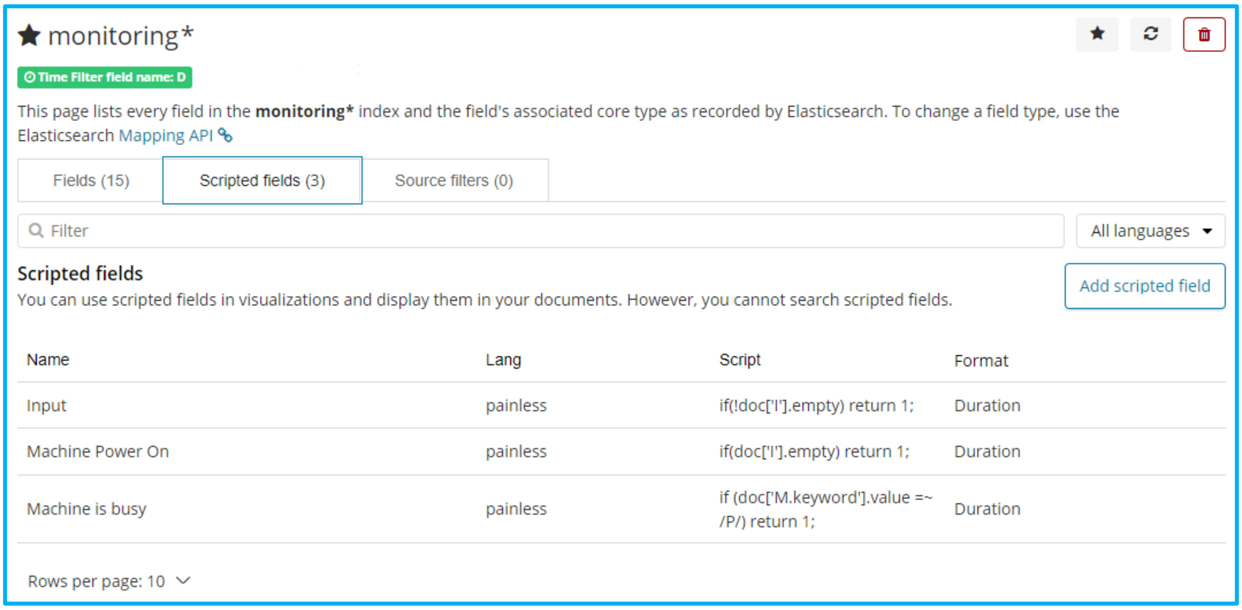


Figure 6.1 Scripted Fields Created in Kibana

Scripted field is a field using painless language script to perform manipulation on data of document and return defined type value. There are 3 scripted fields created in Kibana on created index. The purpose of all of the scripted field is to return 1 second which represented in minute unit. In visualization, minute unit is used instead of second but every document in index is representing 1 second of work. Therefore, there is a need to create a new field to represent duration which is in minute unit. Scripted field performs on each document per time and it will compute the new field individually results in new fields generated in all documents inside index.

6.1.1.1 Scripted Field for Machine Power on Duration

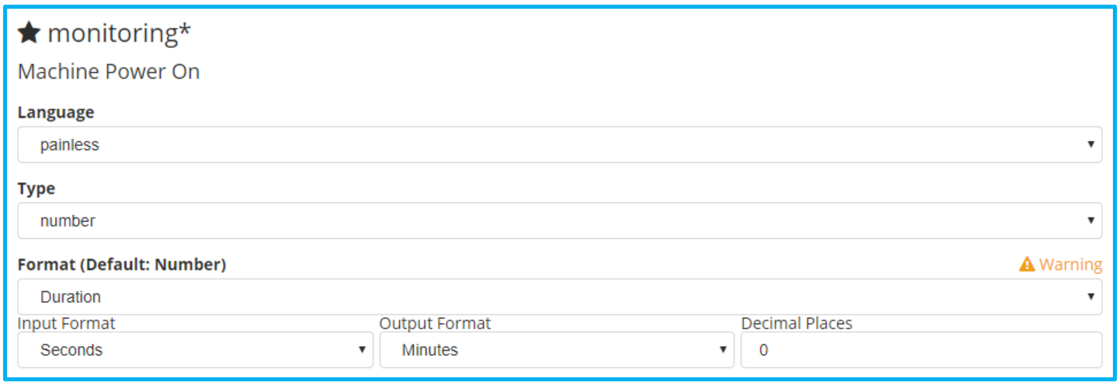


Figure 6.2 Scripted Field of Machine Power on Duration

This field is designed to return 1 second in minute unit in case of the machine power on. This indicates the duration of machine is on in minute unit. The return type of the field is number in duration format. The duration format will convert second into minute with 0 decimal place.

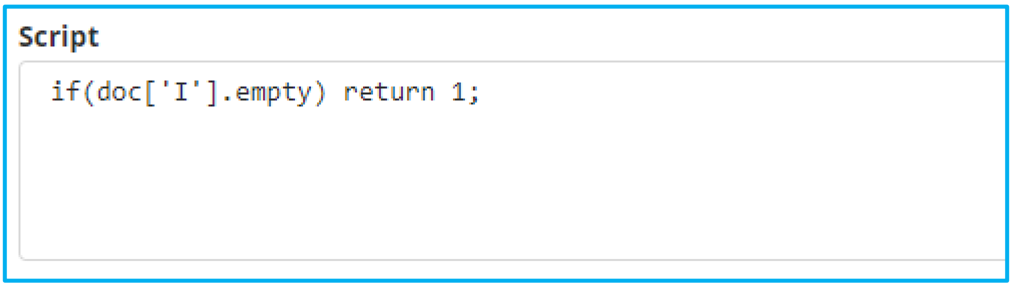


Figure 6.3 Script for Machine Power on Duration

In machine monitoring data object, there is no field ‘I’ exists. This script returns integer 1 provided that field ‘I’ is empty. Integer 1 will be passed as input in second format and converted into output in minute format with round off. This indicates machine is power on in 1/60 minute in case of 1 machine monitoring document.

#	C	1.54
	D	August 9th 2018, 11:05:25.396
#	H	1.37
#	In using without user input	0
#	Input	-
t	M	CO2
#	Machine Power On	0
#	R	1,982.98
#	S	1,017.08
#	TM	8,532,979,712
#	U	1,992,814,592
t	_id	1008958
t	_index	monitoring
#	_score	1
t	_type	all

Figure 6.4 Index Document after Creating Machine Power on Scripted Field

A new field is inserted into document which is 'Machine Power On'. The value of field is showing 0. In fact, 1/60 is a rational number and it includes decimal places. The output format of the field is set with 0 decimal place and above diagram is showing 1 document, therefore 0 is shown rather than decimal places value. In visualization, round off operation is performed after computing total sum of the field and no total sum 0 value will be returned.

6.1.1.2 Scripted Field for User Interactivity Duration

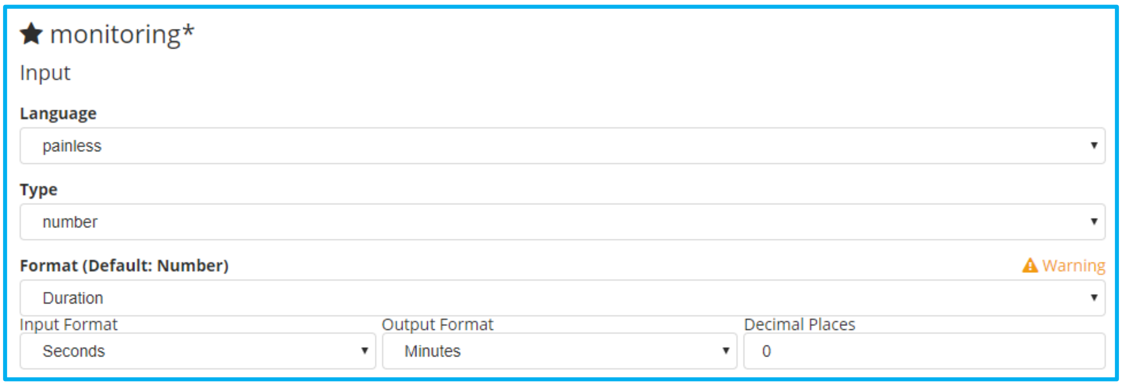


Figure 6.5 Scripted Field of User Interactivity Duration

This field serves same purpose as scripted field for machine power on duration. The difference is this field is designed in case of there is keyboard and mouse input from the user when user interacting with the machine. The field indicates there is someone using keyboard or mouse with 1/60 minute duration.

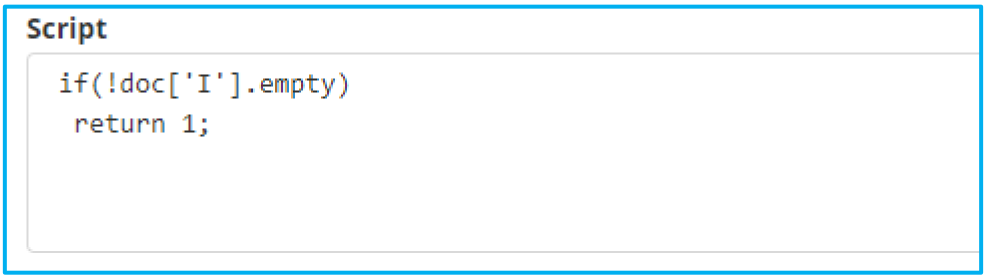


Figure 6.6 Script for User Interactivity Duration

In user interactivity data object, there is a field ‘I’ exists with value of integer 1. This script returns integer 1 provided that field ‘I’ is not empty. Integer 1 will be passed as input in second format and converted into output in minute format with round off. This indicates user is interacting with machine with 1/60 minute.

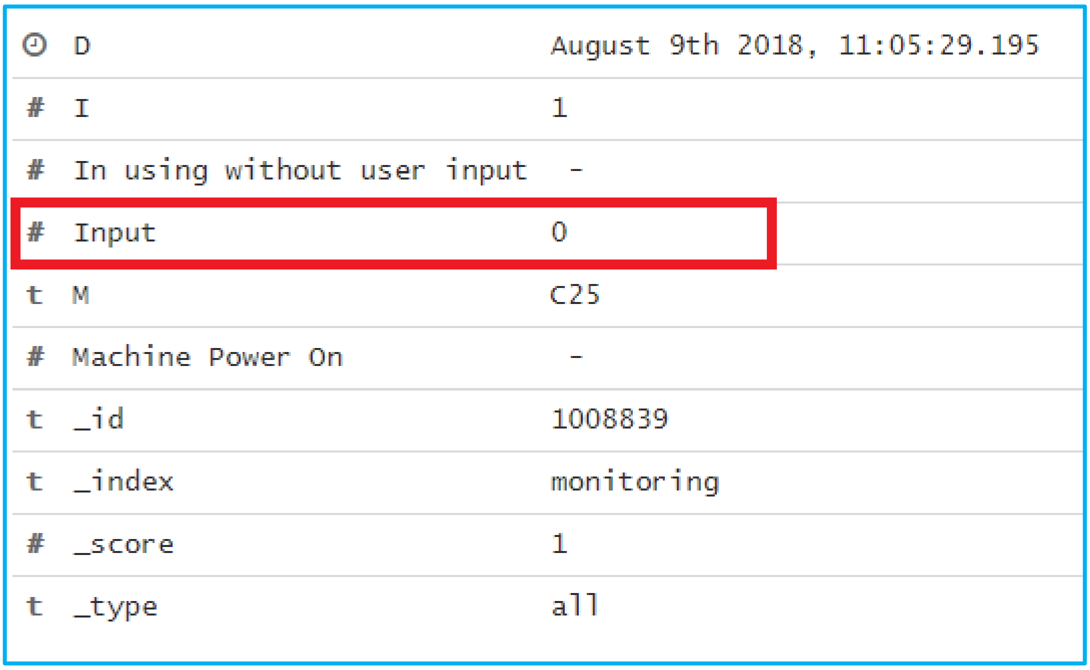


Figure 6.7 Index Document after Creating Input Scripted Field

A new field is inserted into document which is ‘Input’. The reason of getting value 0 is identical as scripted field for machine power on duration.

6.1.1.3 Scripted Field for Machine over Busy

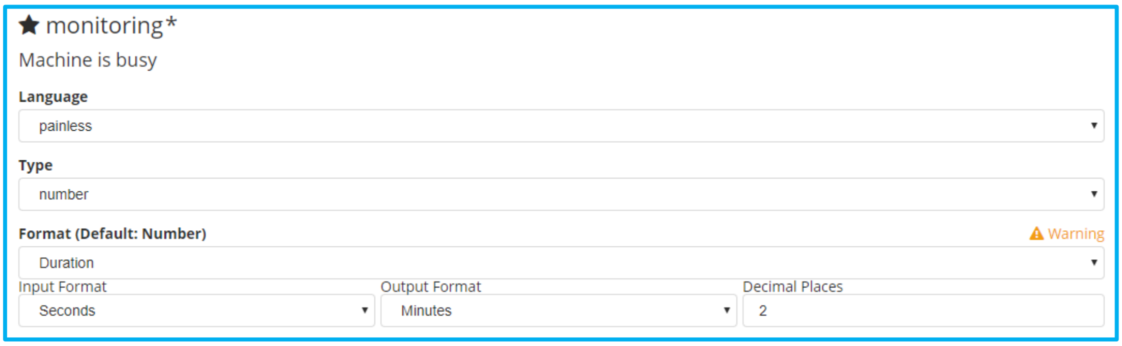


Figure 6.8 Scripted Field of Machine over Busy

This field returns 1 second in minute unit to claim that machine is in over busy mode. The return type of the field is number in duration format. The duration format will convert second into minute. Return value of 0 minute will be obtained if total sum of seconds is less than 30 seconds. Therefore, 2 decimal places are used in final output.

Script

```
if (doc['M.keyword'].value =~ /P/)
 return 1;
```

Figure 6.9 Script for Machine over Busy

In case of collecting metrics data of machine duration is more than 1 second, object with field of machine name which is appended with '/P' will be stored. The script checks whether machine name contains string 'P'. The field will return 1/60 minute if the condition is satisfied otherwise return nothing. The returned value indicates machine is over busy with 1/60 minute.

⏲	D	August 8th 2018, 11:05:16.330
#	I	1
#	In using without user input	-
#	Input	0
t	M	C07/P
#	Machine Power On	-
#	Machine is busy	0.02
t	_id	652937
t	_index	monitoring2
#	_score	1
t	_type	all

Figure 6.10 Index Document after Creating Machine is busy Scripted Field

A new field is inserted into document which is 'Machine is busy'. The value of the field is 0.02 with 2 decimal places.

6.2 Visualization

Visualization is based on 1 week period collected data from August 6th 2018 Monday to August 10th 2018 Friday.

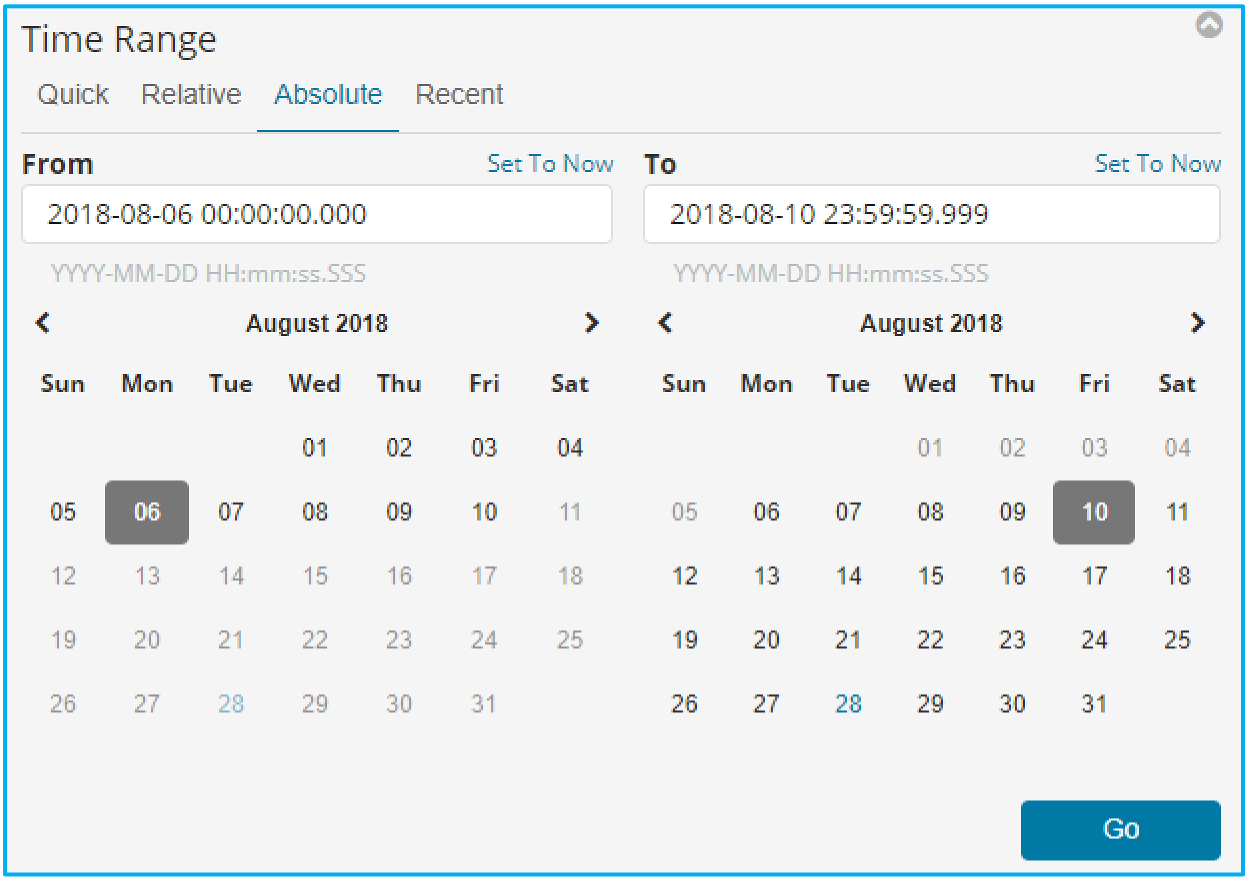


Figure 6.11 Time Range of Data in Visualization

6.2.1 User Interactivity Heat Map

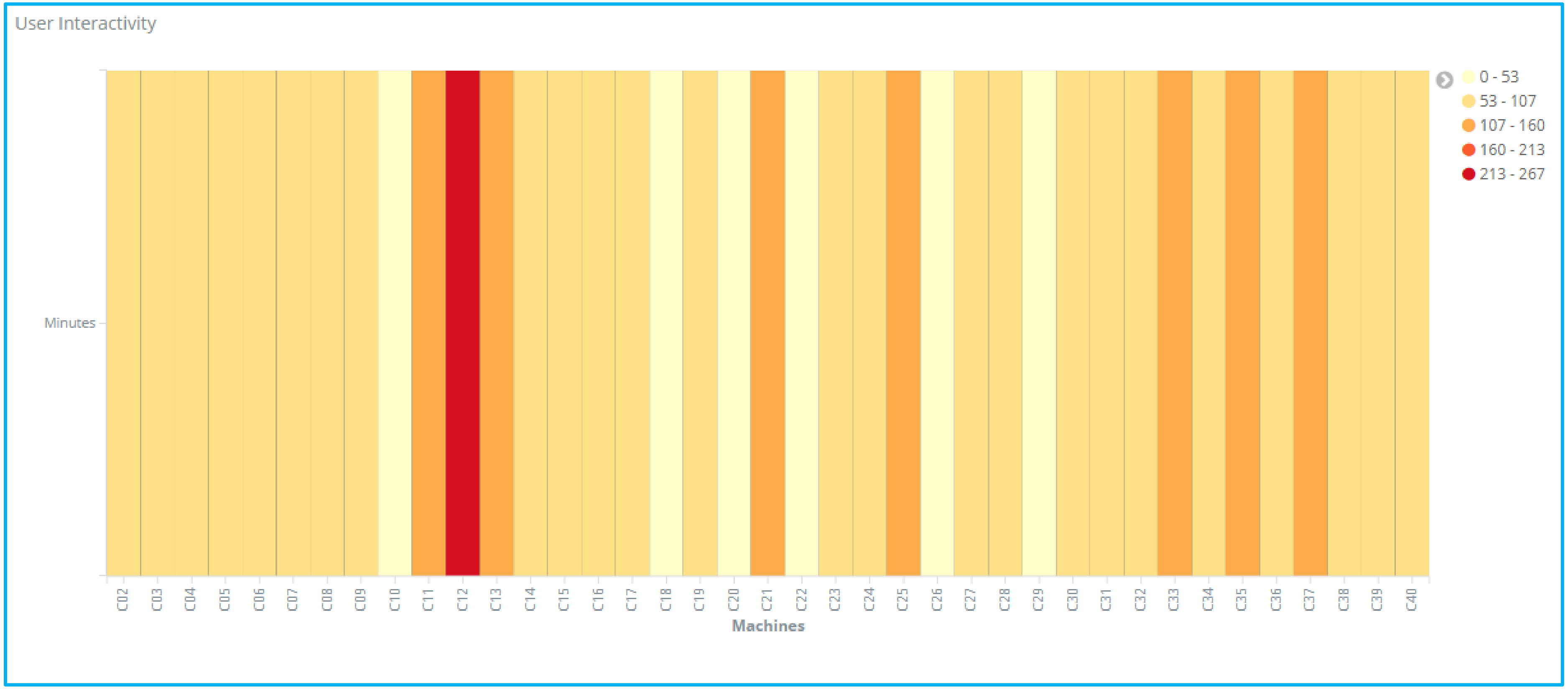


Figure 6.12 User Interactivity Heat Map

Table data of heat map is available in Appendix A

Figure 6.12 is showing the heat map of rate of user interactivity with machine in the lab. X-axis is about machines in the lab and Y-axis is duration of user interactivity in minutes. The heat map is computing user interactivity of 40 machines in lab. Each column of the machine representing minutes of user is interacting with machine by using mouse or keyboard. Number of category is set to 5 to produce more clear heat map visualization, otherwise it is hard to identify most, average and least used machine when there are too many colours are shown if a lot of categories are used. Range of each category is calculated by Kibana without manually configured.

From the heat map, there is a missing machine which is C01. This happens because C01 machine has no any user interactivity in the time range. Kibana relies on date time of object data of each machine to compute user interactivity. Machine with no user interactivity will not record anything therefore there is no date time data of machine C01 exists. This results in no machine C01 will be displayed even though it is 0 minute user interactivity in theoretical.

The heat map is providing knowledge information about all machines in the lab. This visualizes the machines utilization layout over the lab. Based on the heat map, machine C12 is known as the most utilized machine in the lab with the longest duration of user interactivity. On the contrary, machine C01 is known as the least utilized machine in the lab with none of user interactivity.

In addition, there are 6 machines with lowest user interactivity relative to machines shown in the heat map. Machine C10, C18, C20, C22, C26 and C29 have the lowest user interactivity relative to machines in the heat map. Furthermore, 7 machines, C11, C13, C21, C25, C33, C35 and C37, are considered as well utilized compared to other machines by having third highest rate of user interactivity.

Overall, most of the machines have second lowest user interactivity. This group can be used as indication of average rate to compare with other machines. Applying adjustment to machines which has higher or lower than the range of average rate results in all machines would have same average rate of user interactivity. This leads to ideal machine utilization layout in the lab.

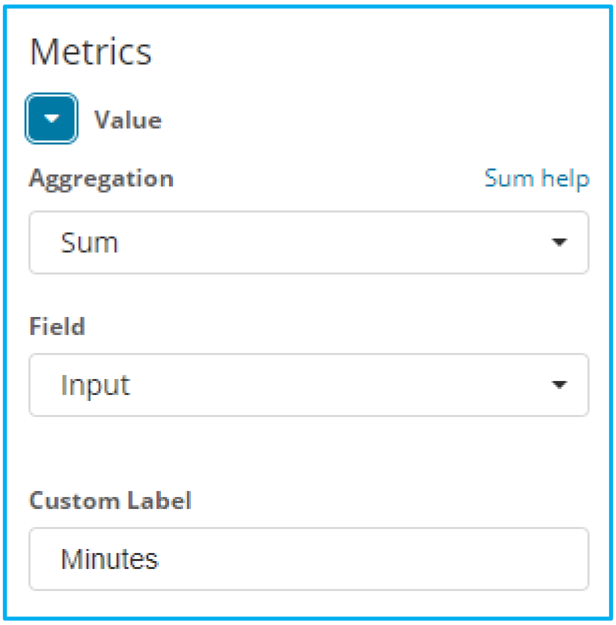


Figure 6.13 Y-axis of User Interactivity

Y-axis computes total sum of field ‘Input’ of each machine. Field ‘Input’ is the scripted field which created before visualization to compute duration of user interactivity.

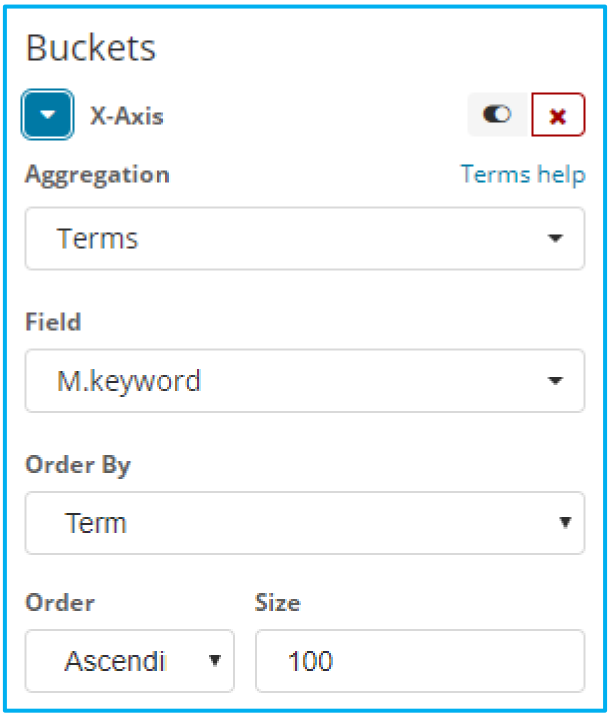


Figure 6.14 X-axis of User Interactivity

X-axis aggregates elements in term of machine name. Hence, it aggregates into 40 machines.

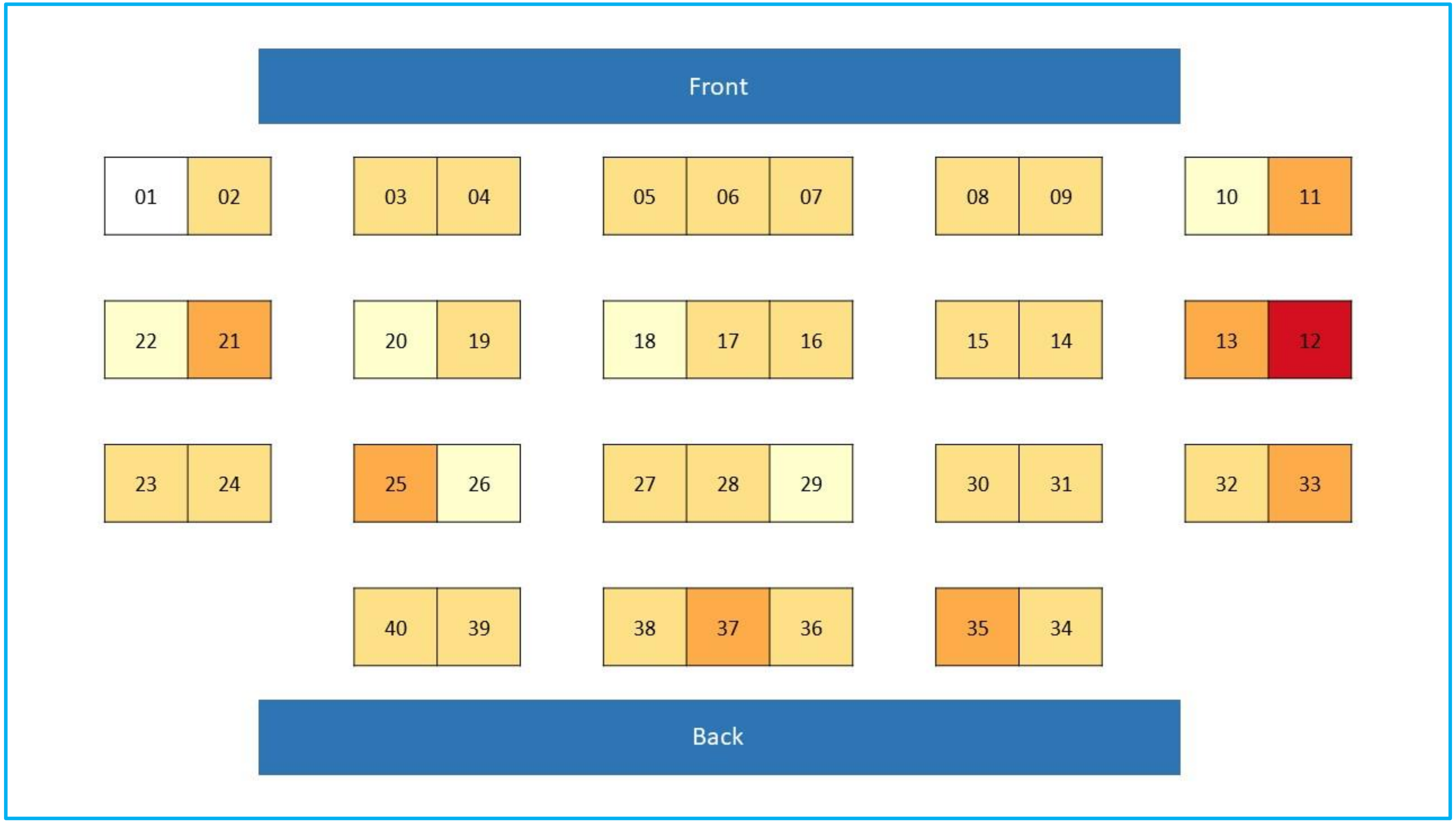


Figure 6.15 User Interactivity In Laboratory Physical Layout

6.2.2 Machine Power on Heat Map

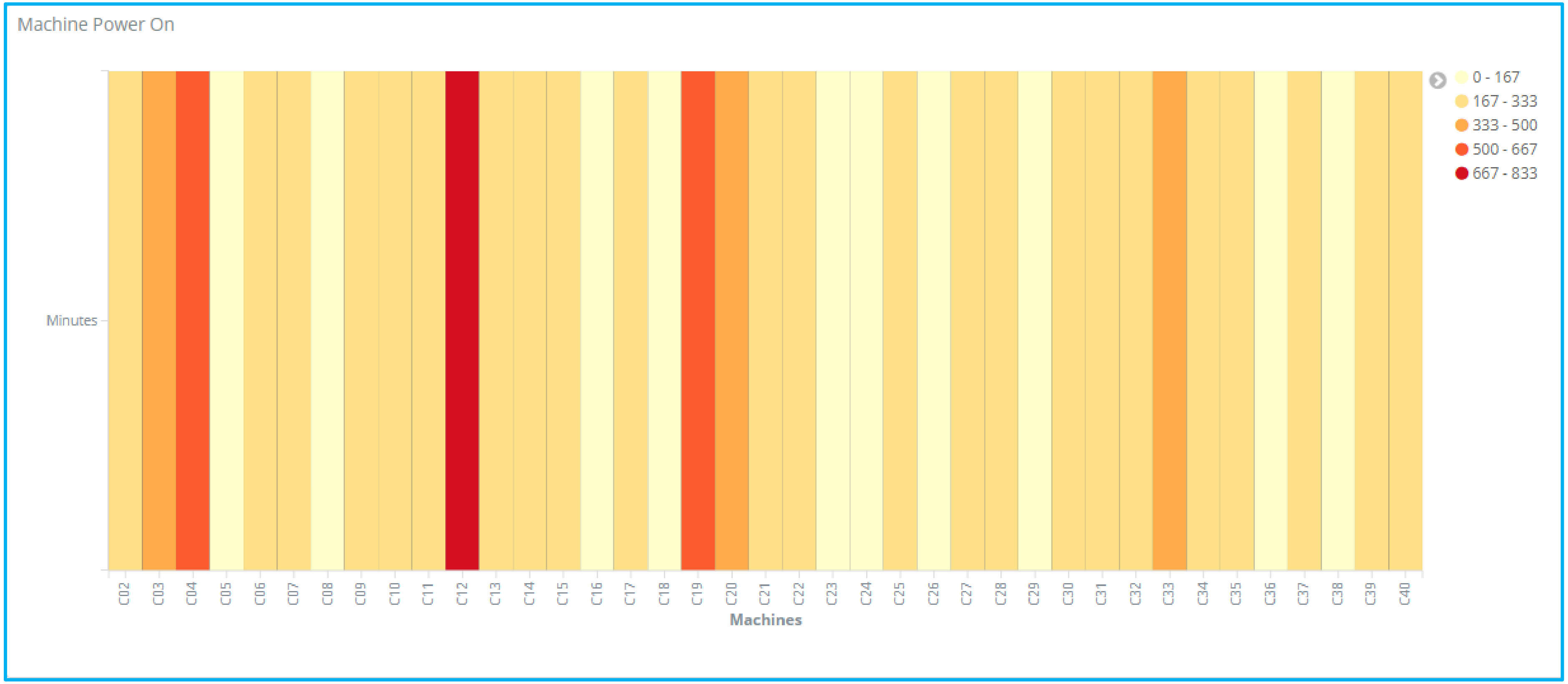


Figure 6.16 Machine Power on Heat Map

Table data of heat map is available in Appendix A

Figure 6.16 shows total minutes of machine is in power on mode during past 1 week period. This heat map computes total minutes by using 40 machines. X-axis is machine in the lab whereas Y-axis is total minutes of machine is in power on mode. Each column of machine shows the total minutes of power on mode of that machine. Number of category is set to 5 as in user interactivity heat map.

In fact, machine C01 is also missing in the machine power on heat map. There is no machine C01 is shown in the heat map. The cause of this is equivalent to the cause in user interactivity heat map. There is no object data of machine C01 exists since machine C01 is not powered on during the time range. Object data contains date time field which is used by Kibana to filter relative to the time. Hence, machine C01 is not included into the heat map.

In conclusion of Figure 6.16, machine C12 is the most often turned on machine whereas machine C01 is the least often turned on machine. In overall, most of the machines fall into second lowest rate of turned on. This indicates this range is average of powered on in the lab.

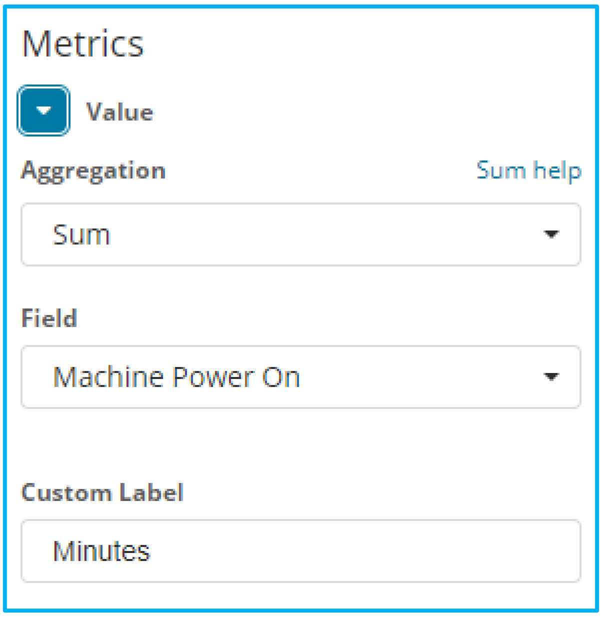


Figure 6.17 Y-axis of Machine Power on

Y-axis computes total sum of field ‘Machine Power On’ of each machine. The field is the scripted field which created in data analysis to compute duration of machine turned on.

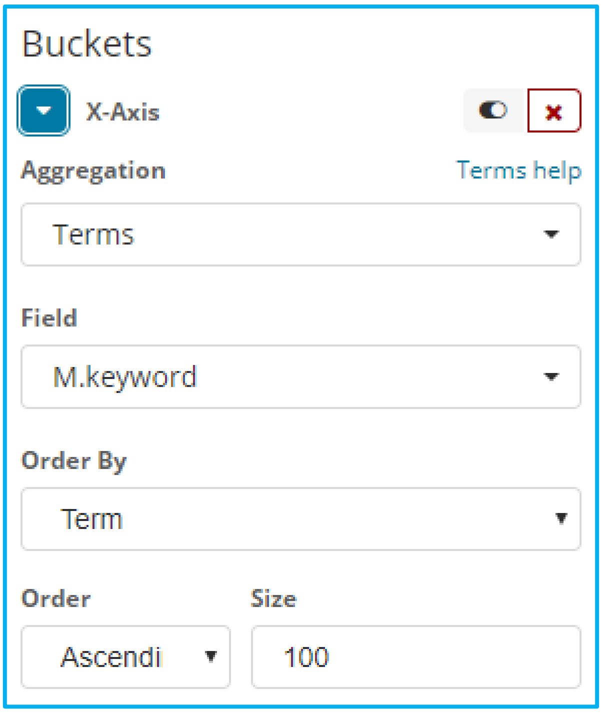


Figure 6.18 X-axis of Machine Power on

X-axis aggregates elements in term of machine name. Hence, it aggregates into 40 machines.

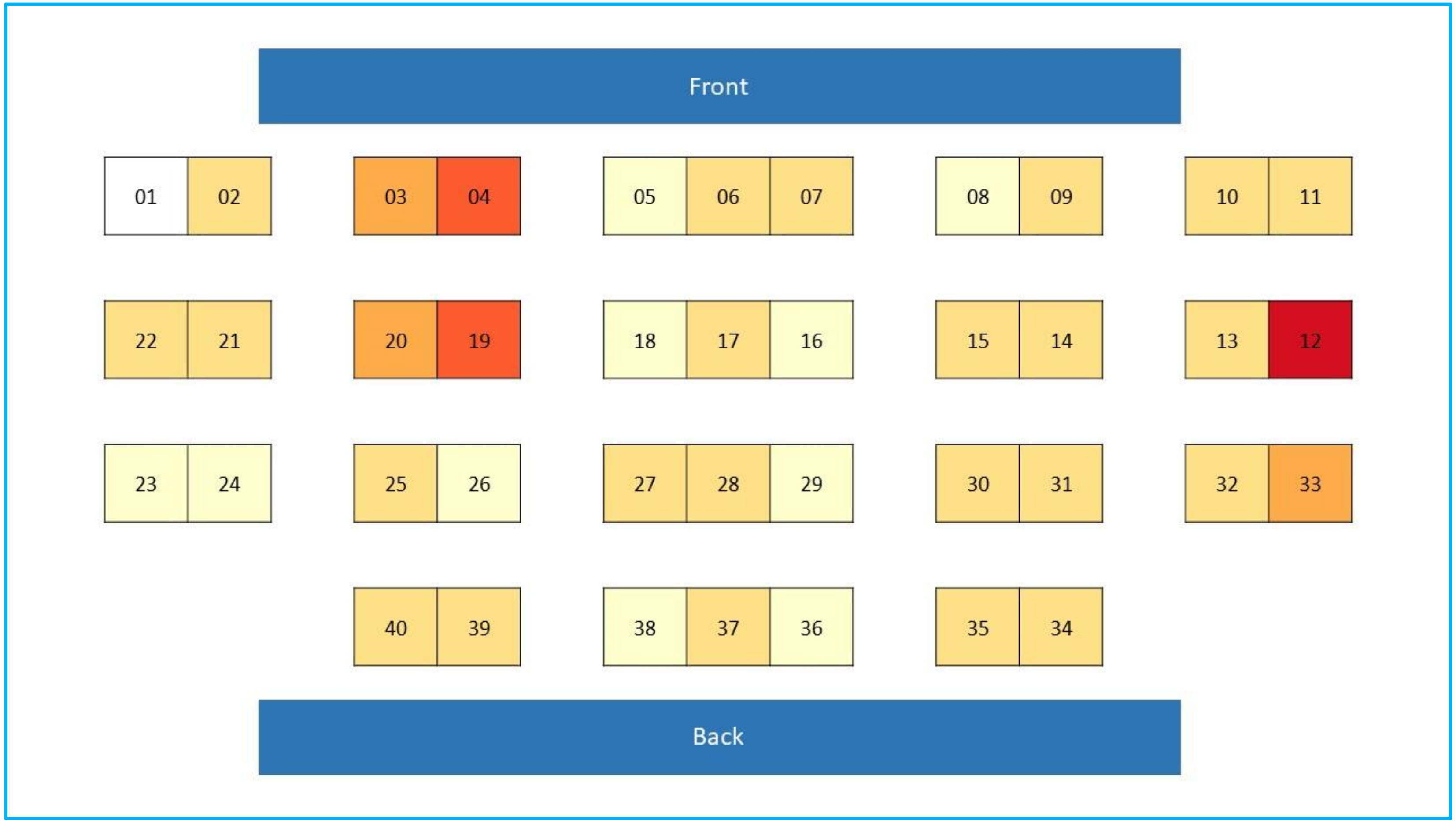


Figure 6.19 Machine Power on In Laboratory Physical Layout

6.2.3 Machine Power on and User Interactivity Histogram

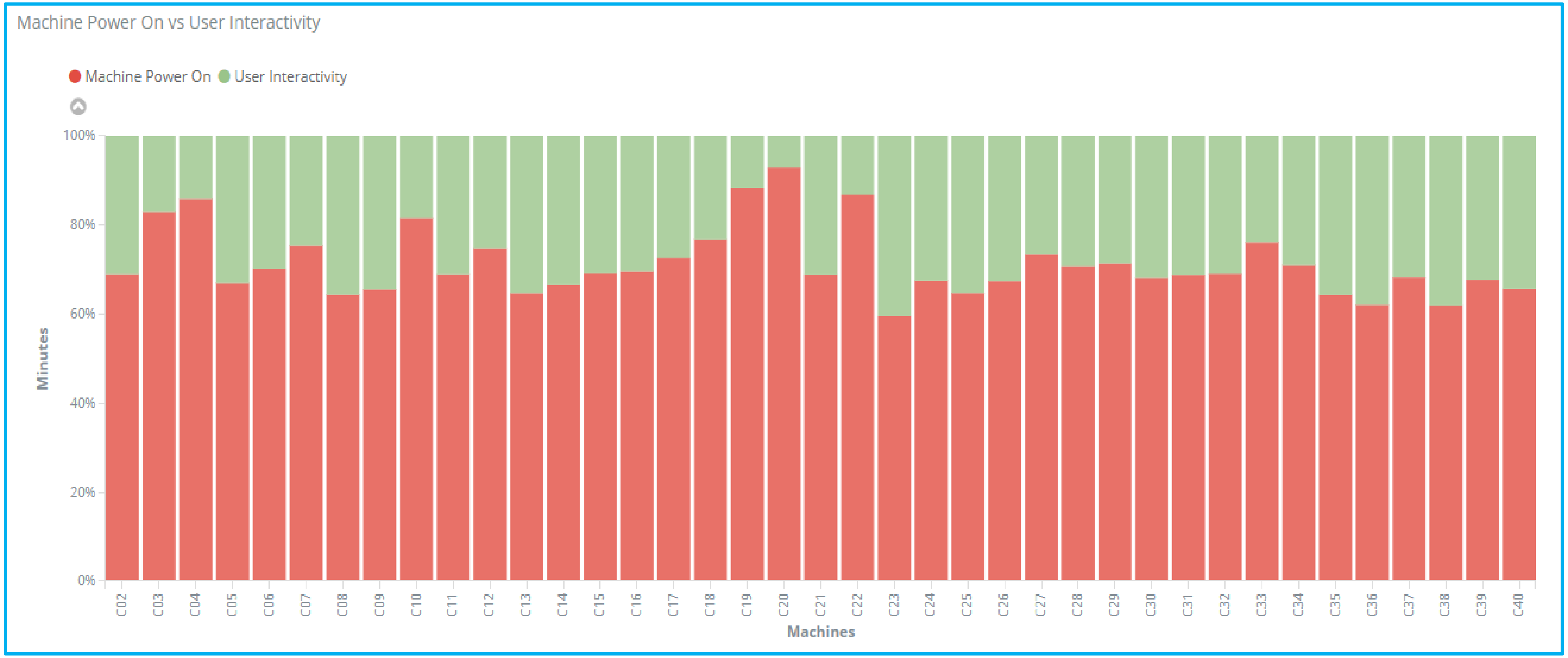


Figure 6.20 Machine Power on and User Interactivity Histogram

Table data of histogram is available in Appendix A

Figure 6.20 shows relationship between user interactivity and machine powered on. The histogram emphasizes whether a machine is utilized to full by using user interactivity when power is on instead of showing which machine is mostly used. A machine is considered as fully utilized in term of the machine power when both duration of user interactivity and machine powered on is equal. A machine could be turned on however there is no one using it, this is wasting power and resources.

Y-axis of the histogram is calculating percentage of minutes for user interactivity and machine powered on of each machine.

$$Percentage\ of\ x = \frac{Total\ duration\ of\ x}{y}$$

where *x* = user interactivity / machine powered on and *y* = total duration of user interactivity and machine powered on

The percentage is computed for each machine respectively. In fact, total duration of user interactivity and machine powered on is different in each machine. X-axis is showing all machines in the lab.

In Figure 6.20, machine C20 has the most unbalanced distribution of minute percentage between user interactivity and machine power. Machine powered on duration of machine C20 is much longer than user interactivity duration. This shows machine C20 is turned on however least using it through keyboard or mouse. This can be concluded as machine C20 is least utilized when the machine power is on. Meanwhile, machine C23 has most balanced distribution between user interactivity and machine power compared to other machines. The machine C23 is achieving most utilized in term of machine power. On the other hand, machine C01 is missing from the histogram due to reason which is mentioned earlier. In this case, machine C01 is not considered as least utilized machine since it is not turned on in term of machine power.

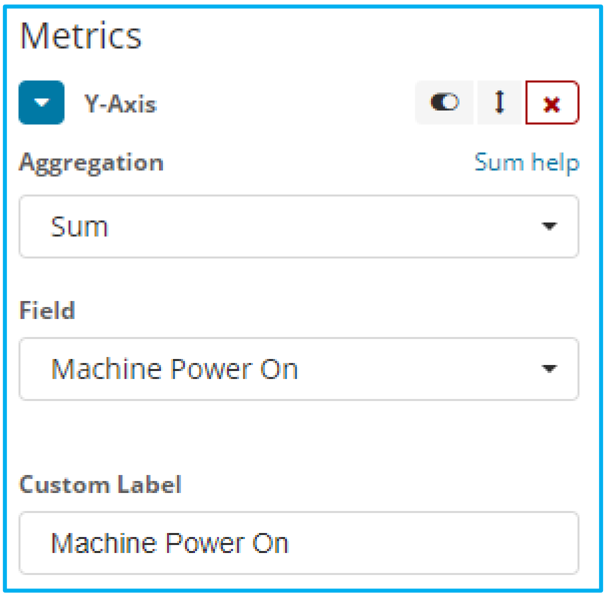


Figure 6.21 Y-axis of Machine Power on Histogram

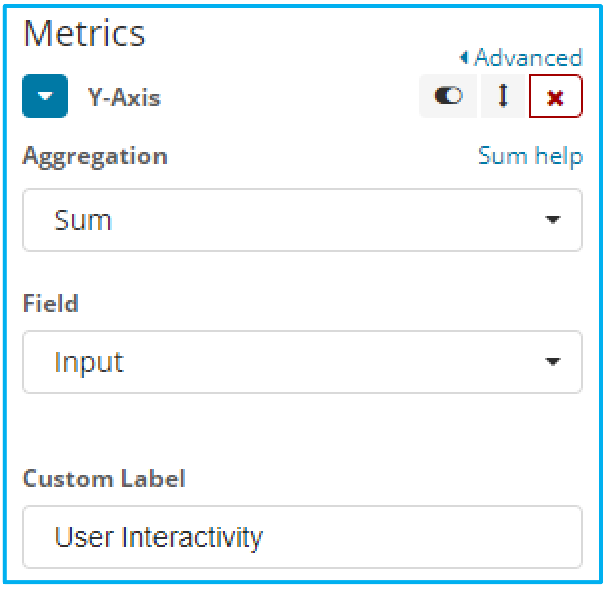


Figure 6.22 Y-axis of User Interactivity Histogram

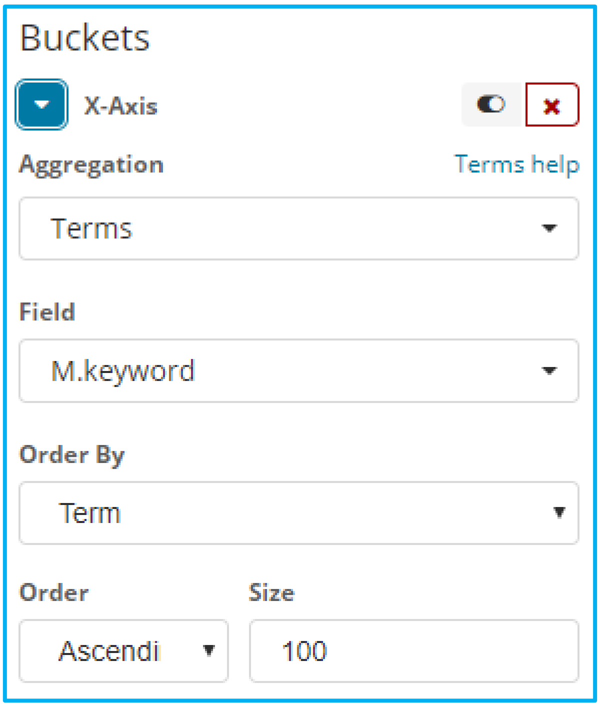


Figure 6.23 X-axis of Machine

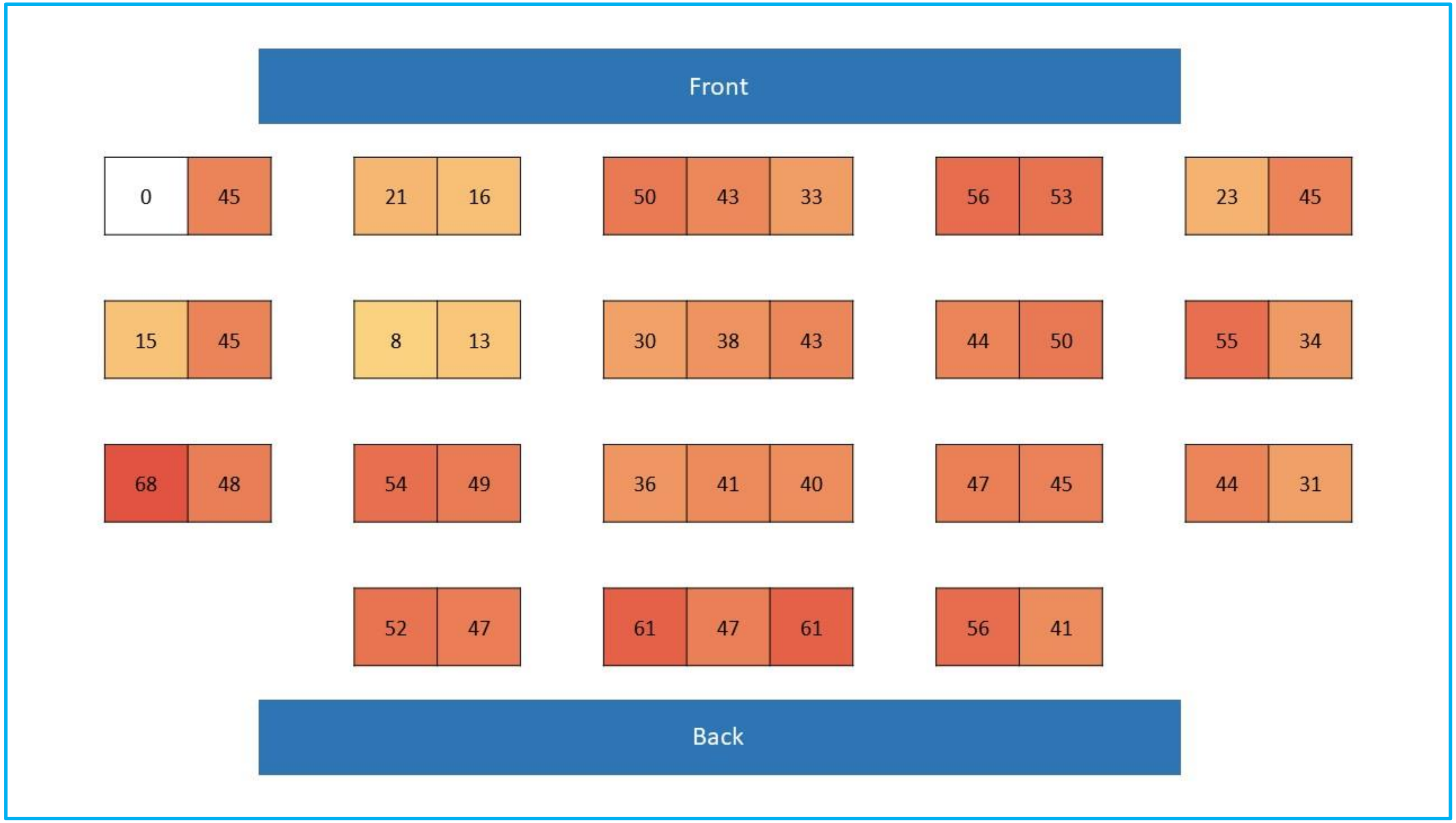


Figure 6.24 Machine Power on and User Interactivity In Laboratory Physical Layout

6.2.4 Machine over Busy Heat Map

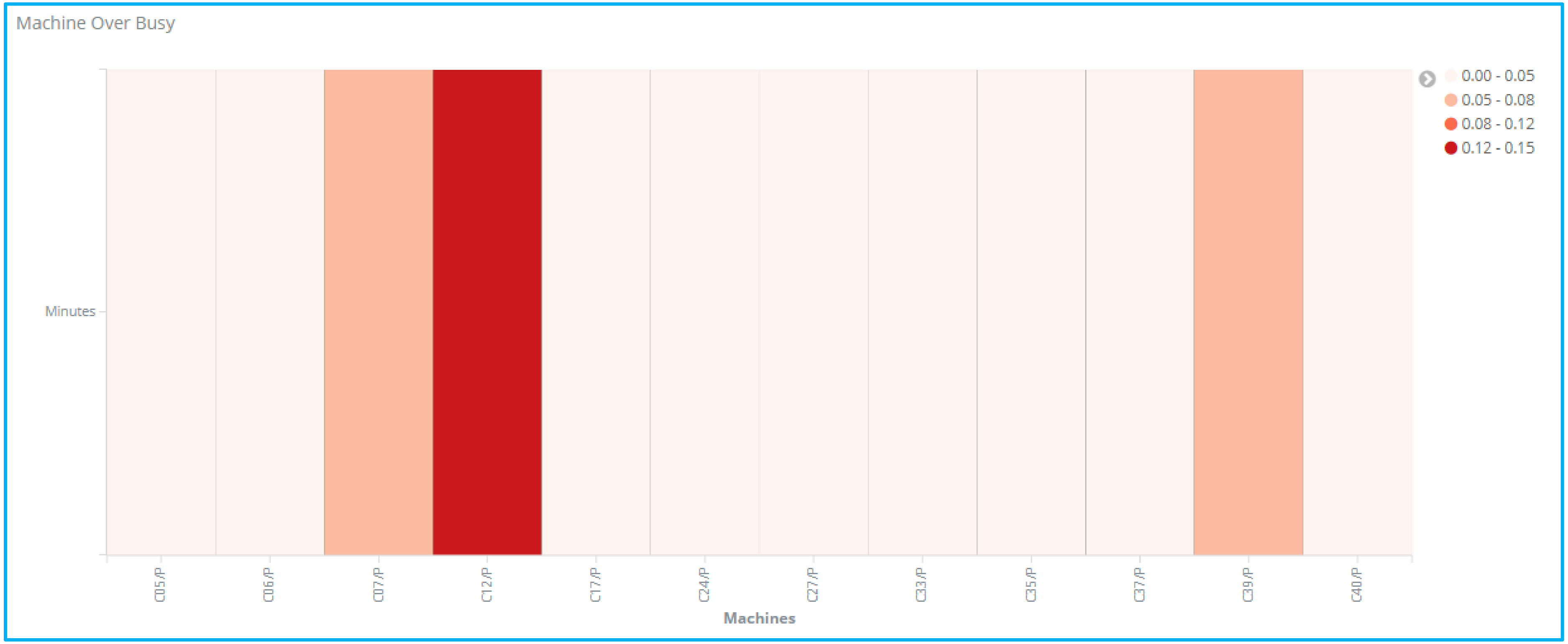


Figure 6.25 Machine over Busy Heat Map

Table data of heat map is available in Appendix A

Figure 6.25 heat map shows about amount of time machine is over busy in minutes. Machine monitoring involves collecting metrics every 1 second, duration of collecting task is limited to not more than 1 second. In case of more than 1 second, machine name is appended with '/P' and no metric data will be stored.

Machine over busy heat map represents how long a machine is in busy mode and unable to handle monitoring task. If a machine is reporting a lot of time in over busy mode, there is some reasons behind it. For instances, there is an aging or problematic machine exists in the lab. In both of the cases, performance of aging and problematic machine drops down and unable to perform as usual. The cause of problematic machine could be having malware, virus or Trojan which results in low performance. This piece of information can be used as first indicator to analyse performance of machine in the lab. In determining cause of huge amount of time machine is in over busy mode requires further detailed metrics and analysis.

In Figure 6.25, machine C12 has longest amount of time, 0.15 minute, in over busy mode. Since the amount is insignificant, machine C12 is considered as performing properly. Consequently, all other machines shown in heat map considered as performing properly as well. Machine never run into over busy mode will not be displayed in the heat map.

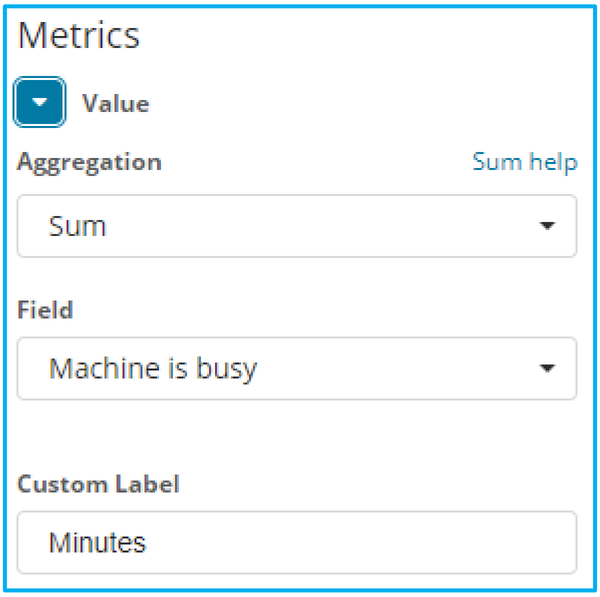


Figure 6.26 Y-axis of Machine over Busy

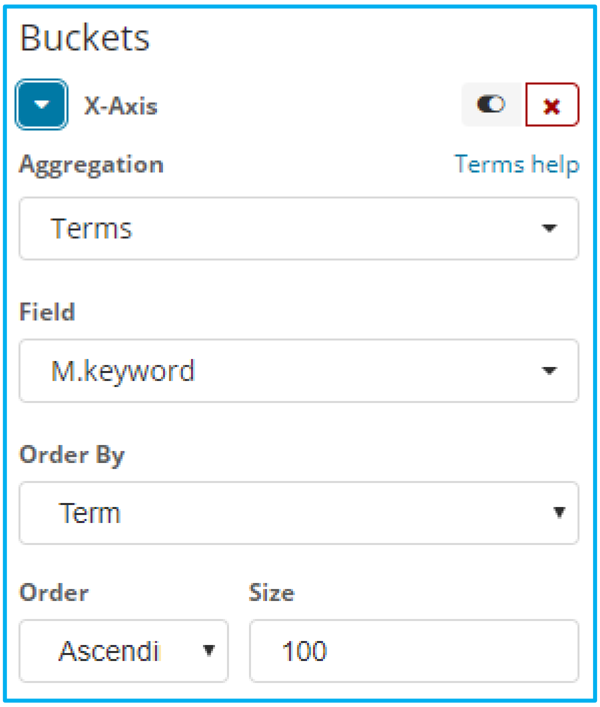


Figure 6.27 X-axis of Machine over Busy

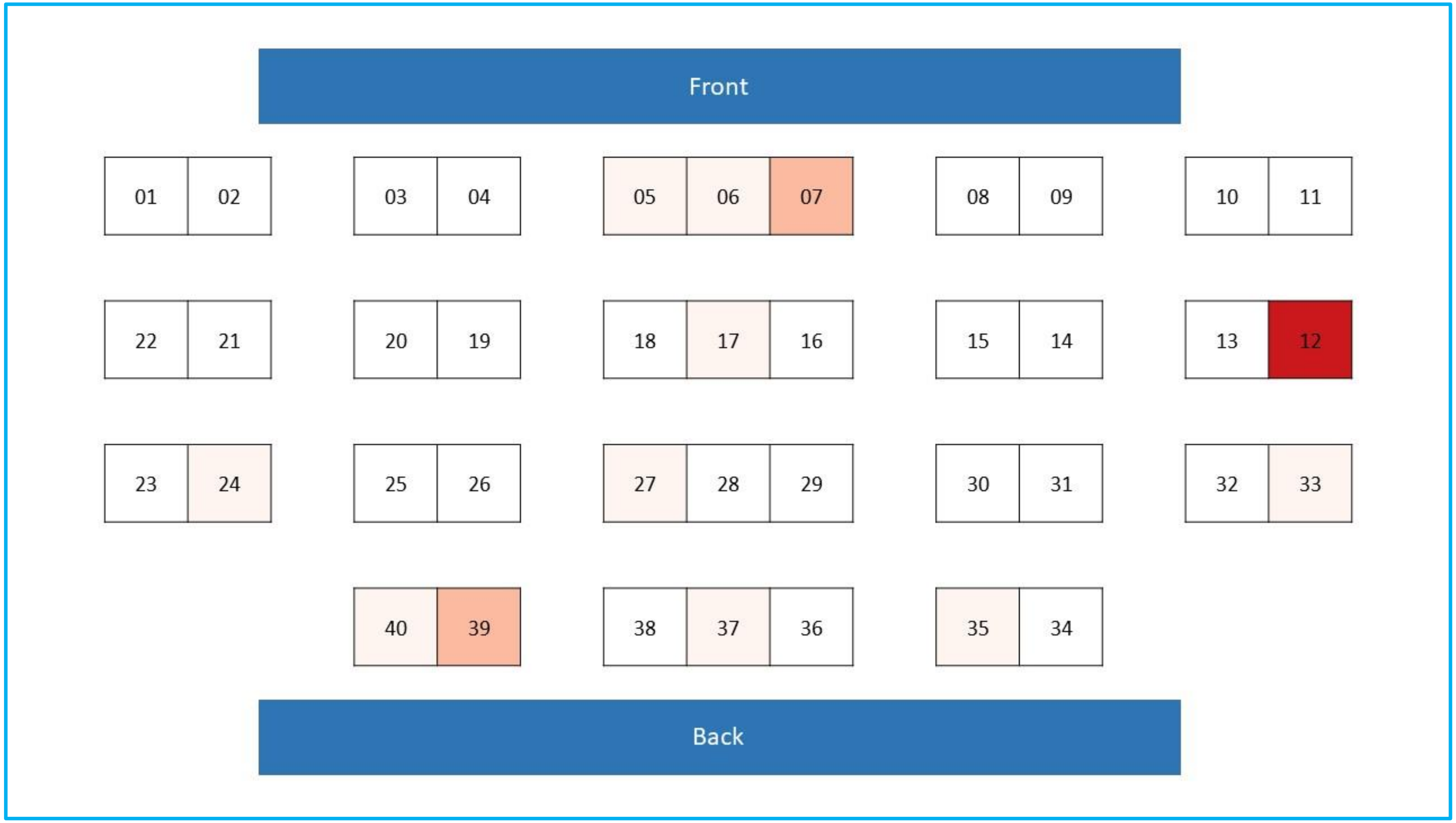


Figure 6.28 Machine over Busy In Laboratory Physical Layout

6.2.5 Summary of Visualization

	Lowest	Highest
1. User interactivity	C01	C12
2. Machine powered on	C01	C12
3. Utilization in term of power	C20	C23
4. Over busy	*	C12

Table 6.1 Summary of Visualization

* All machines except C12 and other machines shown in the heat map

There is a relationship between user interactivity and utilization in term of power. User interactivity reports how often user is interacting with machine by using mouse or keyboard input whereas utilization in term of power reports how often user is interacting when the machine is on. High user interactivity does not imply high utilization in term of power. Based on Table 6.1, machine C12 has the highest user interactivity and machine C23 has the highest utilization in term of power. This shows high user interactivity does not assure the machine is highly utilized when it is turned on. Behind high user interactivity, there could be wasting a lot of power and machine is powered on a lot of time. High user interactivity is great to tell which machine is mostly used in the lab while high utilization in term of power tells which machine is utilized most without wasting unnecessary power and resources.

7.1 Project Review, Discussions and Conclusions

All stated objectives have been met. The objectives are as follows:

1. To collect information of user interactivity with machines.
2. To collect information of power mode of machines.
3. To analyse the relationship between user interactivity with machines and power mode of machines.
4. To visualize the relationship between user interactivity with machines and power mode of machines.
5. To visualize machine utilization layout in the lab by using information of user interactivity with machine.

In developing machine monitoring service, start-up of service has to be immediately after OS is started to capture the metrics data as soon as possible without losing the information about machine power status. However, there is losing some metrics data even with the Windows service implementation. In start-up process of machine monitoring service, network availability listener is registered to receive notification of network status. The duration of registering the listener takes a lot of time which causes service failed to start and reports error in start-up process.

There are many network interfaces in machine therefore the names of all network interfaces are stored before starting collecting. In lab environment, some network interfaces are keep changing which causes error when collecting the network data and eventually whole service going down.

In developing user interactivity monitoring, keyboard and mouse input is monitored to record which time the keyboard and mouse is getting input. Implementing low level global keyboard and mouse hook causes lagging in keyboard input and mouse pointer. This affects user using machine and causes performance issues.

Uploading data into Firebase requires reading file which stored in local machine. Operation of opening and reading file causes writing data into file corrupts and the data is discarded. This happens because writing data is still going on and close file is not performed yet. Eventually, this leads to missing information.

In exporting entire Firebase, the exporting process is keep failing due to the size of Firebase exceeding 256 MB. After getting exported JSON file from Firebase, the import data into Elasticsearch database takes huge amount of time since the objects data is more than millions.

In visualization, feature of computation between aggregate fields to produce a new field is not allowed in Kibana. Value of 0 is not displayed when the machine does not have the total sum. This is because machine does not have the corresponding date time object data stored.

7.2 Novelties and Contributions

Existing resource monitoring solutions do not provide user interactivity and machine power status information as well as visualize the machine utilization in terms of user interactivity and machine power status. In this project, few main questions can be answered.

1. Which machine is mostly used by user?
2. Which machine is least used by user?
3. Which machine is average used by user?
4. Which machine is turned on longest?
5. Which machine is turned on shortest?
6. Which machine is turned on in average?
7. Which machine is well utilized without wasting power?
8. Which machine is wasting power?
9. Which machine is suspected as performance dropping?

In conclusion, machine utilization layout in the lab can be visualized to get understanding about machine utilization circumstance in the lab by using user interactivity information.

7.3 Future Work

To extend machine utilization, there are more metrics needed to be took in account instead of only user interactivity. Sometimes user is using machine without keyboard and mouse input. For instances, watching video, reading document, installing software, updating OS, video chatting and many more. Therefore, other metrics like CPU utilization, hard disk utilization, physical memory, network, GPU utilization, running processes and others are monitored as well to provide more complete machine utilization monitoring. Furthermore, aging and problematic machine could be identified by using metrics mentioned above.

Data processing and migration is an additional step to perform in order to import data from Firebase into Elasticsearch. In further developments, object data is directly insert into Elasticsearch database without going through Firebase to eliminate data processing and migration operation. This also eliminate time took to import data when the JSON file is huge.

Currently machine power mode is obtained through OS, implementing monitoring machine power in low level BIOS to provide more precise machine power status. For instances, machine could be idle in OS bootloader menu when no selection of OS is entered.

1. Elastic (n.d.) *Getting Started* [online]. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started.html (Accessed 12 July 2018)

2. Elastic (n.d.) *Painless Scripting Language* [online]. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/master/modules-scripting-painless.html#modules-scripting-painless (Accessed 12 July 2018)

3. Microsoft (2017) *Developing Windows Service Applications* [online]. Available from: https://docs.microsoft.com/en-us/dotnet/framework/windows-services/ (Accessed 03 August 2017)

4. Nagios Core (2016) *Nagios Core* [online]. Available from: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/toc.html (Accessed 03 August 2017)

5. OpManager (n.d.) *OpManager - Network Monitoring Software* [online]. Available from: https://www.manageengine.com/network-monitoring/help/ (Accessed 03 August 2017)

6. PRTG (n.d.) *PRTG All-In-One 24/7 Computer Monitoring Software* [online]. Available from: https://www.paessler.com/computer_monitoring (Accessed 03 August 2017)

7. Solarwinds (n.d.) *Windows Server Monitoring and Management with Server & Application Monitor* [online]. Available from: https://www.solarwinds.com/topics/windows-monitor (Accessed 03 August 2017)

8. Zabbix (2016) *Zabbix Documentation 3.4* [online]. Available from: https://www.zabbix.com/documentation/3.4/manual/introduction/about (Accessed 03 August 2017)

Table of Machine Power on and User Interactivity

Machines	Machine Power On (minutes)	User Interactivity (minutes)
C02	193	86
C03	448	92
C04	534	88
C05	155	77
C06	227	97
C07	312	102
C08	154	86
C09	171	90
C10	189	43
C11	286	129
C12	786	264
C13	218	119
C14	176	88
C15	239	106
C16	161	70
C17	233	88
C18	148	45
C19	513	67
C20	466	35
C21	238	107
C22	195	29
C23	97	66
C24	113	54
C25	225	122
C26	107	52
C27	260	94
C28	251	104
C29	120	48
C30	209	98
C31	211	96
C32	223	99
C33	451	142
C34	213	87
C35	206	115
C36	164	100
C37	290	135
C38	160	98
C39	196	93
C40	201	105

Table of Machine over Busy

Machines	Minutes
C05/P	0.02
C06/P	0.02
C07/P	0.05
C12/P	0.15
C17/P	0.02
C24/P	0.02
C27/P	0.03
C33/P	0.02
C35/P	0.03
C37/P	0.02
C39/P	0.07
C40/P	0.02

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 1, 2018	**Study week no.: 1**
Student Name & ID: YAP JIA WEI 1404123	
Supervisor: Dr Ooi Boon Yaik	
Project Title: Intelligent Monitoring System	

1. WORK DONE

- Machine monitoring parts done

2. WORK TO BE DONE

- User monitoring application development

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

- Slow progress

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 1, 2018	Study week no.: 2
Student Name & ID: YAP JIA WEI 1404123	
Supervisor: Dr Ooi Boon Yaik	
Project Title: Intelligent Monitoring System	

1. WORK DONE

- Machine monitoring parts done

2. WORK TO BE DONE

- User monitoring application development

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

- Slow progress

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 1, 2018	**Study week no.: 3**
Student Name & ID: YAP JIA WEI 1404123	
Supervisor: Dr Ooi Boon Yaik	
Project Title: Intelligent Monitoring System	

1. WORK DONE

- Machine monitoring parts done

2. WORK TO BE DONE

- User monitoring application development

3. PROBLEMS ENCOUNTERED

- User monitoring application causes lagging of the keyboard input and mouse pointer.

4. SELF EVALUATION OF THE PROGRESS

- Slow progress

_________________________ _________________________

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 1, 2018	**Study week no.: 4**
Student Name & ID: YAP JIA WEI 1404123	
Supervisor: Dr Ooi Boon Yaik	
Project Title: Intelligent Monitoring System	

1. WORK DONE

- Machine monitoring parts done

2. WORK TO BE DONE

- User monitoring application development

3. PROBLEMS ENCOUNTERED

- User monitoring application causes lagging of the keyboard input and mouse pointer.

4. SELF EVALUATION OF THE PROGRESS

- Slow progress

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 1, 2018	**Study week no.: 5**
Student Name & ID: YAP JIA WEI 1404123	
Supervisor: Dr Ooi Boon Yaik	
Project Title: Intelligent Monitoring System	

1. WORK DONE

- Machine monitoring parts done

2. WORK TO BE DONE

- User monitoring application development

3. PROBLEMS ENCOUNTERED

- User monitoring application causes lagging of the keyboard input and mouse pointer.

4. SELF EVALUATION OF THE PROGRESS

- Slow progress

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 1, 2018	Study week no.: 6
Student Name & ID: YAP JIA WEI 1404123	
Supervisor: Dr Ooi Boon Yaik	
Project Title: Intelligent Monitoring System	

1. WORK DONE
• Machine monitoring parts done
2. WORK TO BE DONE
• User monitoring application development
3. PROBLEMS ENCOUNTERED
• User monitoring application causes lagging of the keyboard input and mouse pointer.
4. SELF EVALUATION OF THE PROGRESS
• Slow progress

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 1, 2018	**Study week no.: 7**
Student Name & ID: YAP JIA WEI 1404123	
Supervisor: Dr Ooi Boon Yaik	
Project Title: Intelligent Monitoring System	

1. WORK DONE

- Machine monitoring parts done

2. WORK TO BE DONE

- User monitoring application development

3. PROBLEMS ENCOUNTERED

- User monitoring application causes lagging of the keyboard input and mouse pointer.

4. SELF EVALUATION OF THE PROGRESS

- Slow progress

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 1, 2018	**Study week no.: 8**
Student Name & ID: YAP JIA WEI 1404123	
Supervisor: Dr Ooi Boon Yaik	
Project Title: Intelligent Monitoring System	

1. WORK DONE

- Machine monitoring parts done
- User monitoring application

2. WORK TO BE DONE

- Implementation of system into all OSs of every machine in lab

3. PROBLEMS ENCOUNTERED

- Data uploading corrupts when writing and reading operations executed concurrently

4. SELF EVALUATION OF THE PROGRESS

- Slow progress

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 1, 2018	**Study week no.: 9**
Student Name & ID: YAP JIA WEI 1404123	
Supervisor: Dr Ooi Boon Yaik	
Project Title: Intelligent Monitoring System	

1. WORK DONE

- Machine monitoring parts done
- User monitoring application

2. WORK TO BE DONE

- Implementation of system into all OSs of every machine in lab

3. PROBLEMS ENCOUNTERED

- Application service corrupts in some machines

4. SELF EVALUATION OF THE PROGRESS

- Slow progress

______________________________ Supervisor's signature

______________________________ Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 1, 2018	**Study week no.: 10**
Student Name & ID: YAP JIA WEI 1404123	
Supervisor: Dr Ooi Boon Yaik	
Project Title: Intelligent Monitoring System	

1. WORK DONE

- Machine monitoring parts done
- User monitoring application
- Implementation of monitoring solution into all OSs of every machine

2. WORK TO BE DONE

- Collection of data
- Data analysis and visualization

3. PROBLEMS ENCOUNTERED

- Data migration between Firebase and Elasticsearch database due to different structure of database

4. SELF EVALUATION OF THE PROGRESS

- Slow progress

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 1, 2018	**Study week no.: 11**
Student Name & ID: YAP JIA WEI 1404123	
Supervisor: Dr Ooi Boon Yaik	
Project Title: Intelligent Monitoring System	

1. WORK DONE

- Machine monitoring parts done
- User monitoring application
- Implementation of monitoring solution into all OSs of every machine
- Collection of data

2. WORK TO BE DONE

- Import entire extracted Firebase database into Elasticsearch index

3. PROBLEMS ENCOUNTERED

- Errors when extracting entire Firebase as size of database is more than allowed export size 256 MB

4. SELF EVALUATION OF THE PROGRESS

- Slow progress

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 1, 2018	**Study week no.: 12**
Student Name & ID: YAP JIA WEI 1404123	
Supervisor: Dr Ooi Boon Yaik	
Project Title: Intelligent Monitoring System	

1. WORK DONE

- Machine monitoring parts done
- User monitoring application
- Implementation of monitoring solution into all OSs of every machine
- Collection of data
- Import data into Elasticsearch

2. WORK TO BE DONE

- Visualization of data

3. PROBLEMS ENCOUNTERED

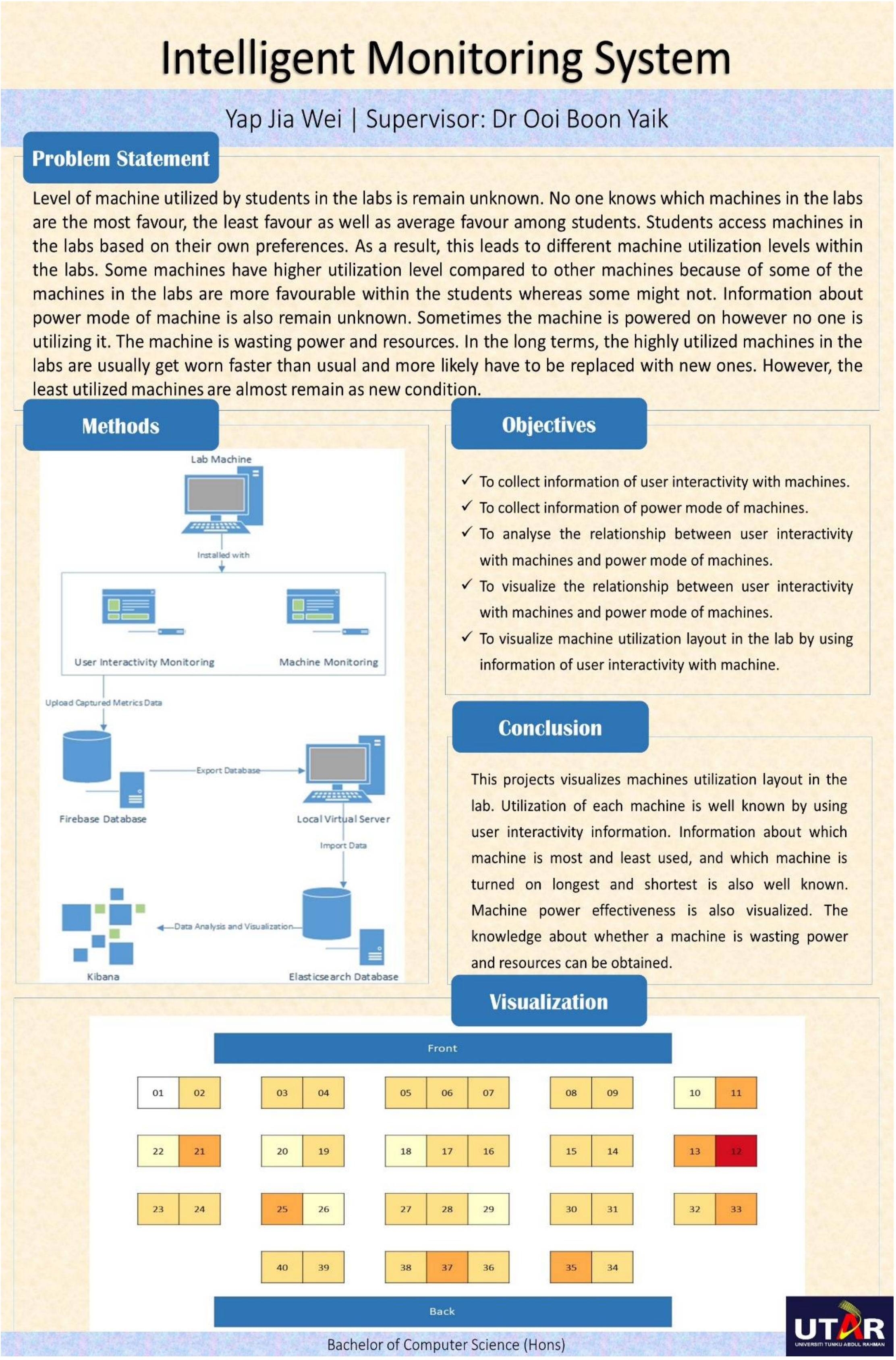
- Importing spends a lot of time due to millions of objects data

4. SELF EVALUATION OF THE PROGRESS

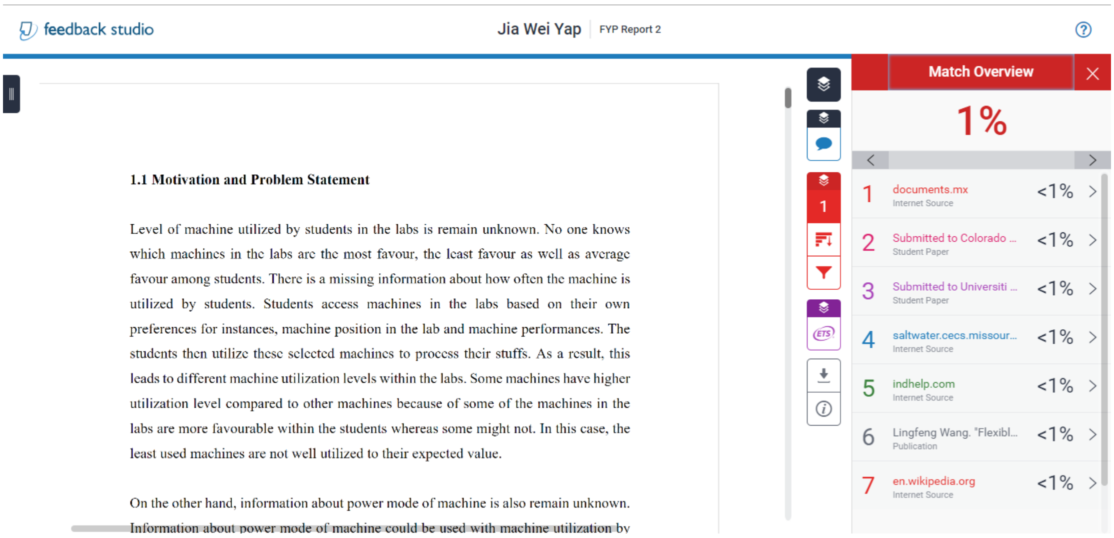
- Slow progress

Supervisor's signature

Student's signature


Intelligent Monitoring System
Yap Jia Wei | Supervisor: Dr Ooi Boon Yaik
Problem Statement
Level of machine utilized by students in the labs is remain unknown. No one knows which machines in the labs are the most favour, the least favour as well as average favour among students. Students access machines in the labs based on their own preferences. As a result, this leads to different machine utilization levels within the labs. Some machines have higher utilization level compared to other machines because of some of the machines in the labs are more favourable within the students whereas some might not. Information about power mode of machine is also remain unknown. Sometimes the machine is powered on however no one is utilizing it. The machine is wasting power and resources. In the long terms, the highly utilized machines in the labs are usually get worn faster than usual and more likely have to be replaced with new ones. However, the least utilized machines are almost remain as new condition.
Methods
Lab Machine
Installed with
User Interactivity Monitoring
Machine Monitoring
Upload Captured Metrics Data
Firebase Database
Export Database
Local Virtual Server
Import Data
Data Analysis and Visualization
Kibana
Elasticsearch Database
Objectives
✓ To collect information of user interactivity with machines.
✓ To collect information of power mode of machines.
✓ To analyse the relationship between user interactivity with machines and power mode of machines.
✓ To visualize the relationship between user interactivity with machines and power mode of machines.
✓ To visualize machine utilization layout in the lab by using information of user interactivity with machine.
Conclusion
This projects visualizes machines utilization layout in the lab. Utilization of each machine is well known by using user interactivity information. Information about which machine is most and least used, and which machine is turned on longest and shortest is also well known. Machine power effectiveness is also visualized. The knowledge about whether a machine is wasting power and resources can be obtained.
Visualization
Front
01 02 03 04 05 06 07 08 09 10 11
22 21 20 19 18 17 16 15 14 13 12
23 24 25 26 27 28 29 30 31 32 33
40 39 38 37 36 35 34
Back
Bachelor of Computer Science (Hons)
UTAR
UNIVERSITI TUNKU ABDUL RAHMAN

Plagiarism Check Summary



ORIGINALITY REPORT

1%	1%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	documents.mx Internet Source	<1%
2	Submitted to Colorado Technical University Online Student Paper	<1%
3	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1%
4	saltwater.cecs.missouri.edu Internet Source	<1%
5	indhelp.com Internet Source	<1%
6	Lingfeng Wang. "Flexible measurement point management in an industrial automatic supervision system", 2007 IEEE Autotestcon, 2007 Publication	<1%
7	en.wikipedia.org Internet Source	<1%

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	
ID Number(s)	
Programme / Course	
Title of Final Year Project	

Similarity	**Supervisor's Comments** **(Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index:________ % **Similarity by source** Internet Sources: _______________% Publications: __________ % Student Papers: _____________ %	
Number of individual sources listed of more than 3% similarity: _________	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

____
Signature of Supervisor

Name: __________________________

Date: ___________________________

____
Signature of Co-Supervisor

Name: __________________________

Date: ___________________________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (PERAK CAMPUS)

CHECKLIST FOR FYP2THESIS SUBMISSION

Student Id	1404123
Student Name	YAP JIA WEI
Supervisor Name	DR. OOI BOON YAIK

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Title Page
	Signed form of the Declaration of Originality
	Abstract
	Table of Contents
	List of Figures (if applicable)
	List of Tables (if applicable)
	List of Symbols (if applicable)
	List of Abbreviations (if applicable)
	Chapters / Content
	Bibliography (or References)
	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
	Poster
	Signed Turnitin Report (Plagiarism Check Result – Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report. ________________________ (Signature of Student) Date:	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. ________________________ (Signature of Supervisor) Date: