SNRU Journal of Science and Technology

Journal home page: snrujst.snru.ac.th

The development of web-based application for notification of network status and assigned task management

Pattadon Tanooserm, Satit Kravenkit*

Department of Computer and Information Science, Faculty of Applied Science and Engineering, KhonKaen University, 42000 Thailand

*Corresponding Author: satikr@kku.ac.th

Received: 27 April 2017; **Revised:** 21 August 2017; **Accepted:** 22 August 2018; **Available online:** 1 September 2018

Abstract

The TOT public company limited, the sales regional and service center at the 2nd KhonKaen, is responsible for monitoring and maintaining the network devices across Northeast Thailand. The service center agents of each province are reported the notified problems of network devices from the customer service center; the staff, subsequently, are assigned to cope with such failure of network devices. Recently, 2nd service center at KhonKaen has used the Zabbix, an open source program, to monitor the working status of network devices, thus saving time spent on problem-solving and waiting time spent on waiting for the problem notification from the customer service center. The use of the open source Zabbix program enables modulator to rapidly obtain the notification of defects of network devices; however, this operation, which use documents forms to record the repair device information and send it to the others operators (employee), suffers from complicated process and delay. To solve the problem, we developed a web application for notification of network status and assigned task management. The overview of the developed web application consists of two layers. In the backend, the Mongo DB is created as the database which records the status log from Zabbix. In the frontend, developed web application implemented by Node.js get the failure of network devices from Mongo DB and send it as a task to the modulator and operator to fix them. Moreover, the time spent on the tasks operation, including receiving, transferring and completing, is shown to discover the total time. In the evaluation, the results of user satisfaction testing the developed web application yield the good level (4.02) which assure the appropriate for user-friendly and GUI (Graphic User Interface) to the TOT staff. Moreover, the amount of times process of maintenance of devices network using the developed application is significantly reduced from 70 to 20 minutes, comparing to the traditional process. It has the high opportunity to eliminate the problem from the customer.

Keywords: Zabbix; monitoring network devices; Mongo DB; Node.js

1. Introduction

TOT [1] public company limited, regional sales and customer service at the seconds is located in KhonKaen. It is headquartered which communicate with all of the service centers and offices in the Northeast of Thailand. The company establishes the regional maintaining department responsible for monitoring and receiving the notification concerning problems of network devices at the TOT provincial center over the Northeast.

With respect to the repairing operation process of the network devices, the provincial TOT center is directly notified from customers via the telephone contact. Afterward, the staff deal with those failure

devices following the identified location where the devices are installed. Presently, due to the considerable number of networks over the 20 provinces, the staff are unlikely to effectively respond to the customers' request; the Zabbix [2] have been introduced to eliminate the problems by which the staff immediately monitor and acknowledge the impaired devices through the application not such traditional means as telephone call. In the previous studies, [3] demonstrate the Zabbix primarily serve the valuable functions, including the price, the system requirement, and the reaction to failure when compared with the monitoring software such as Caci, Nagios XI, and SCOM2012. Moreover, the previous works [4 – 6] studying the monitoring resource the in cloud computing infrastructure select the Zabbix program as the tools for their works. Also, the ZABBIX monitoring alert and reply with LINE Application to monitor the status of resource usage from the server and the installed software application is developed [7]. This developed system immediately informs the administrator via the Line program, when the server and software application is an error. Moreover, the [8] evaluate that the Zabbix is the IT-monitoring tool offering the great performance and deliver email or SMS notifications to the administrators. Therefore, Zabbix is the superb monitor programs and suitable for use in the TOT 2nd Khonkaen to monitor the network devices status (success or failure) and report the results to the modulator. The use of Zabbix makes it possible to report the network devices status. However, the modulators are required to immediately transfer the tasks to the operators in the form of paper-based notification, thereby leading to certain drawbacks. To illustrate, there may be some case that the first operators are unavailable to carry out the informed task, they may be obliged to accordingly transfer the tasks to the second operators. Thus, transferring task may bring the modulator and operator the difficulties of addressing the case, leading to time consumption.

To eliminate the problem, a web application of notification and assigned tasks management of network device is developed. The objective of the developed web application is to inform the current status of the network devices which has the problem to the modulators and systemically send it to the operators. The system is designed to detect the working status of the network devices three times per two minutes. In case that a failure of the network devices is detected, the developed web application retrieves such information and informs it to the modulator. When the modulators receive the tasks, the assigned tasks are subsequently transferred to the operators. Finally, the operators undertake the transferred tasks and inform the results upon completing the task via the developed web application. The remaining sections of the paper is organized as follows. Section 2 deals with the material and methods, including the proposed overview of the developed web application, the investigation of analysis and design system, and testing the functions. Section 3 deals with the result and discussion, especially, the user satisfaction level and benefit of total time operation for using the developed application. Section 4, the research studies is concluded.

2. Materials and methods

In this section, the model of development application is proposed. Then, the primary functions and entity of the data model are analyzed and design. Lastly, those functions within the application are tested.

The proposed model

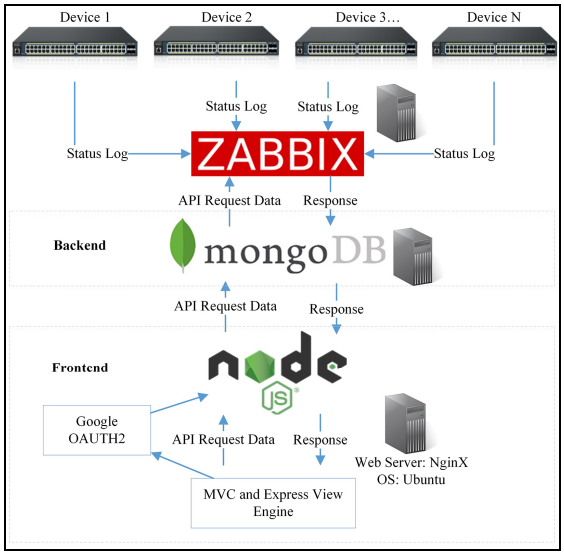


Fig. 1 The developed web application model

We divide the proposed the application model (see Fig. 1) into two layers including, backend and frontend layers. The main technologies and tools within the backend are the Zabbix and Mongo DB. The Zabbix is an enterprise open source monitoring software for network and applications which track the status of various network service, server and network hardware, while the Mongo DB [9] is the open source document database with the scalability and flexibility for the web application which is suitable for the Node.JS application. Considering the frontend, the four technologies are deployed. Firstly, the Node.js [10]; open source cross-platform JavaScript runtime which let developer use JavaScript for server-side script to produce dynamic web page content before the page in the response to the web browser. Secondly, Google OAuth 2.0 [11] is an authorization framework which allows the applications to get the limited access to user's resources on an HTTP service. The OAuth 2.0 protocol is used by Gmail for authenticating a Google account and authorizing access to user data. Thirdly, the Nginx [12], is a web server, which can also be used as a reverse proxy, load balancer, and HTTP cache. Moreover, Nginx is a high performance, stability, simple configuration, and low resource consumption. Lastly, Ubuntu [13] is a free and open source operating systems built on Debian's architecture infrastructure and consist of the Linux server, desktop, and operating system versions.

The basic role and communication among the backend and frontend layers are described as follows. In the first step, the status logs which detail report the status of network devices locating within and other regions are recorded to the Zabbix server. The second step, the Mongo DB which provide a flexibility and dynamic document database, send the request data (failure of network devices status) to Zabbix server and receive the response data. The Last step, the web application which is implemented by Node.js using the pattern of Model-View-Controller (MVC) send the request and receive the response information to the Mongo DB. This information is the operation related to the failure of network devices which extend the assignment of the tasks by the operators and modulator. For the web application, we use the Google OAuth 2.0 to implement authorization to access Google APIs; thus, the developed web application is able to use the OAuth 2.0 to obtain permission from users. The Nginx which install into the Ubuntu server is selected as the web server to serve the rapid processing with the

number of information. In the next section, the system analysis and design of the developed application are described.

System Analysis and Design

System Analysis

According to the system analysis, we mainly collect the information from the user requirements namely, the modulator and operator.

The modulator is the staff who firstly receive the notification network devices status from Zabbix and subsequently transfer it to the operators within and other region. The modulator requirements are investigated as follows. The modulator needs to register to the developed web application before login and login using the third provider service account (Google). The developed web application need to send the notification network devices from the Zabbix to the developed web application and record the network devices status to the Mongo DB. Furthermore, the developed web application need to record the date and time of receiving and transferring the tasks and transfer the task from the 1st operator to the 2nd operator within the same region.

The operator is the staff who receives the task from the modulator or operator and solves the problem. The operator requirements are analyzed as follows. The operator needs to register to the developed web application before login and access (login) the developed web application by using a Gmail account. The developed web application need to sends the task to the operators and record date and time when the operators receive, transfer and finish the task. Moreover, the developed web application need to transfer the task from the 1st operator to the 2nd operators with the same province/region. Also, the operator needs to report the network device status of transferring, receiving and finishing the task of operators.

System Design

We implement the use case diagram (see Fig. 2) to design the main function and services which are mainly analysed by the requirement of the modulator and operator.

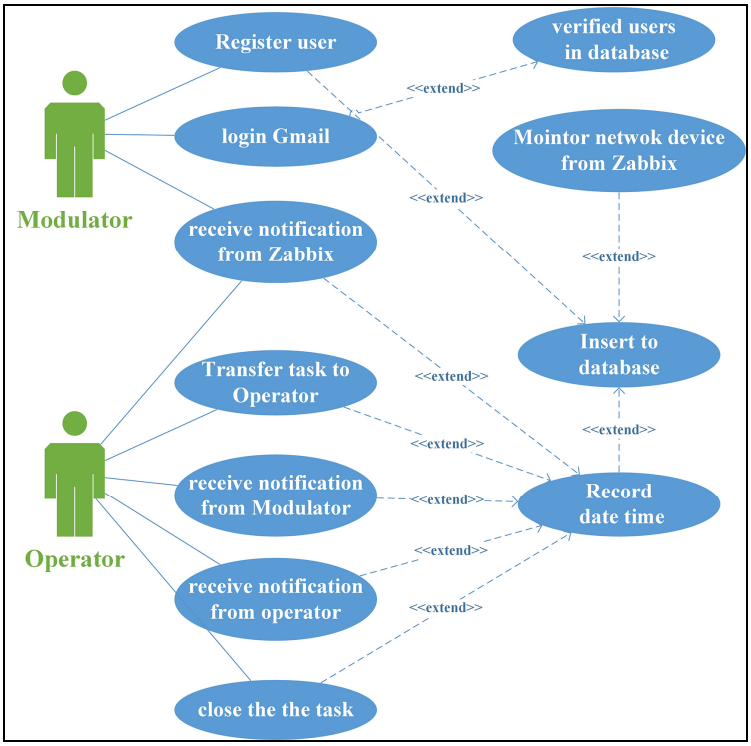


Fig. 2 The use case diagram of developed application

In the modulator section, the primary functions are demonstrated as follows. Considering the login process, the modulator uses the Gmail Account to log in. The developed web application receives the failure of notification network devices which are received by the Zabbix server and inform it to the modulator. The modulator gets the failure of network devices from the Zabbix and sends it as the assigned task to the operators. In all step of transferring tasks operation, the developed web application records and display date and time.

In the operator section, the main functions are illustrated as follows. The operator login using Gmail Account. The operator receives the notification network devices from the modulator. Also, the operator gets the tasks from the modulator and sent it to the 2nd operator. Moreover, the operator transfers the task to the operator in the same province/region and closes the tasks after the completed task operation. The date and time of the tasks operations are recorded and displayed by the developed web application.

Data Model

In the Mongo DB, the ZabbixDB is creased as the database consisting of four collections including Reportzabbixs, Regis, Users, and Sends. Also, we demonstrate the document and description within the collection as follows. In the Fig. 3, the Reportzabbixs collection record the notification information which is directly derived from the Zabbix. When the modulator login, the data within the Reportzabbixs collection is retrieved and displayed. The Regis collection record the user information which is received from the register from the developed web application. The Users collection demonstrated collect the data derived from the Google to authorize the Regis collection that the user login with Gmail account is already registered with the web application. In this collection, we assign the Document 1 overlap with the Document 2 to easily apply to other social media. The Sends collection records the tasks which transfer from the modulator to the operator.

Reportzabbixs collection

Document	Description
_id	Identification
Name	Devices name
Region	Region of devices location
Province	Province of devices location
first_date	Date and time of the notification task which is sent from the Zabbix
diff_time1	Date and time of the notification task which are received by the modulator

Users collection

Document1	Document2	Description
_id		Identification
Google		Google account
	Email	User email
	name	User name
	token	Token Identification
	id	User Identification

Regis collection

Document	Description
_id	Identification
Username	User name
Email	User email
privilege	User privilege (modulator, operator)
province	Office of staff in the province
Region	Office of staff in the region
Tel	Telephone number

Sends collection

Document	Description
_id	Identification
name	Device Name
region	Region of devices location
province	Province of devices location
first_date	Date and time which receive the notification from the Zabbix
get_time	Date and time which receive the task
Send_time	Date and time which transfer the task
get_timeO	Date and time which is gotten by the operator
Send_timeO	Date and time which is sent by operator
Close_time	Date and time which is finished by operator
Status_time	Time status
Time_total	Total time until get to finish the task
Diff_time	The different time of the tasks which is gotten by operator

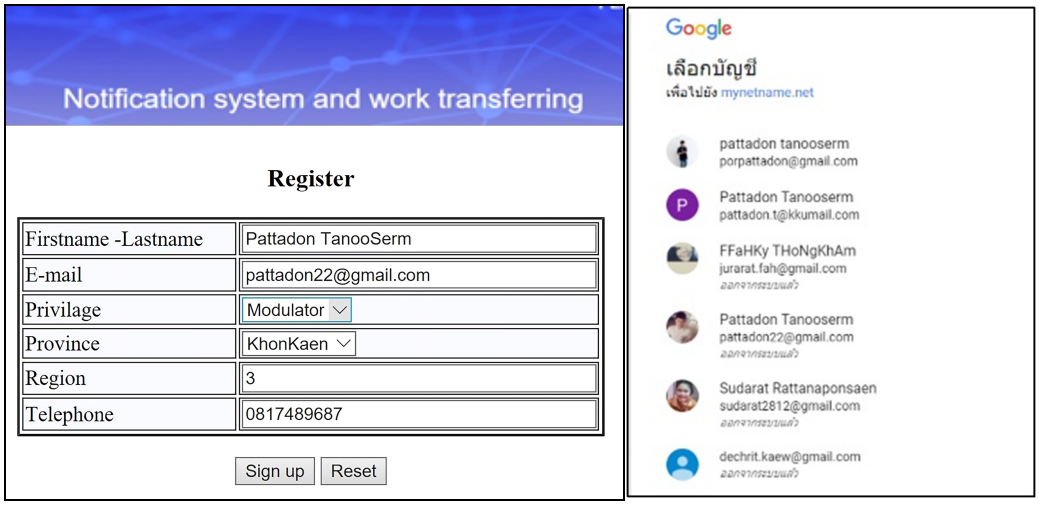
Fig. 3 the four collections within the ZabbixDB

Experiment and Test

The developed web application is tested by the modulator, operator, supervisors, and staff within the 2nd TOT department to evaluate the capability and correctly of the functions. We take the assigned data to the system to following the condition and requirement of users. The process of system testing is divided into two sections; register and login testing and transferring task testing demonstrated as follows.

Register and Login testing

The user needs to register with their profile including first name, last name, email, the privilege mode (modulators, operators), the province, region, and telephone (see Fig 4 (a). After completed register, users can use the Gmail account to log in and access to the system as shown in Fig 4 (b).



(a) (b)

Fig. 4 The register and login application forms

Transferring Task testing

When the modulator logins, the system displays the user profile and the list of failure of network devices including Identification (ID), name, location (the region and province), date, time, the problem of devices, and status as shown in the Fig. 5. The users can edit their profile by selecting the Account Management button. In case that there are seven lists of network devices, we provide detailed explaination as follows. In the first (0001) and last (0079) ID refer to the tasks having been received by a modulator (Pattadon), where 4358.08 minutes is spent and 6500.99 minutes of total time, respectively. Both tasks are sent and completed to and by Mr.Deachrit Keawpradab. The next one, the tasks id of 0002 and 0003 is received by a modulator, waiting for sending to the operators, while both tasks ID of 0004 and 0005 are not received. Lastly, the ID 0078 is received and sent to Mr. Dechrit and Ms.Sudarat, who are the first and seconds operators, respectively. To fulfill the objective of developing the system, we need to find the total time of the task. Thus, once GET and SEND buttons are selected, the different time (minutes) between previous and current operations are immediately shown. To easily monitor, the three status are assigned including, the Zabbix alerts failure to modulator, modulator sends the tasks to operators, and operators complete the tasks.

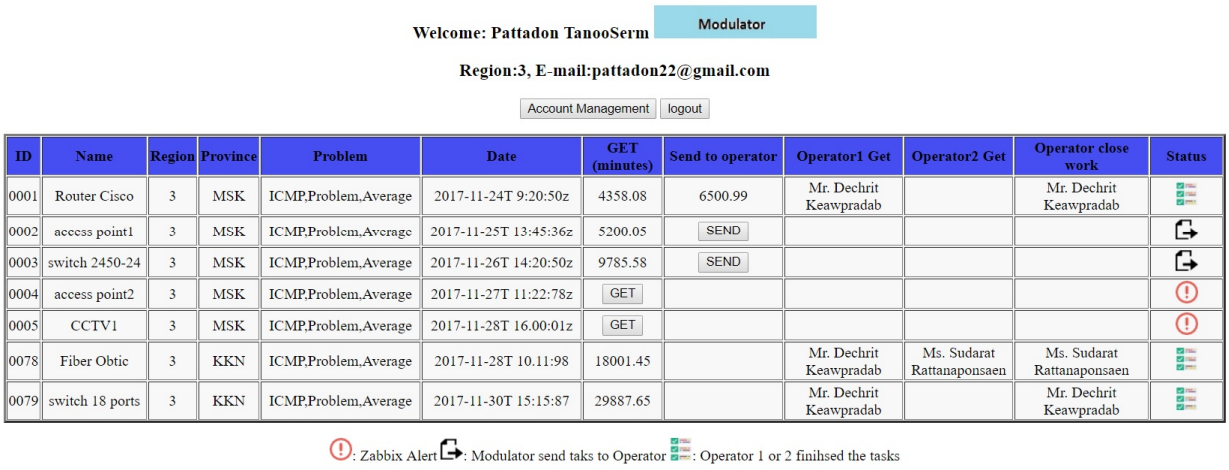


Fig. 5 The modulator application form

After the modulator (Mr. Pattadon) transfers the tasks to the operator (Mr. Dechrit), the operators' form is displayed (see Fig. 6). The first record is the ID 001 task, which is not completed by the operator; thus the GET button and the sign status appear. In the second record, the GET button and CLOSE button are chosen with the total time of 3358.78 and 14789.35 minutes, which means that the operator Mr. Dechirt individually finishes the received and completed tasks; thus, the sign status is shown. In the last record, the operator receives tasks, where 369.56 minutes are spent, and transfers it to the second operators, where 415.58 minutes are spent; thus, the yellow status appears.

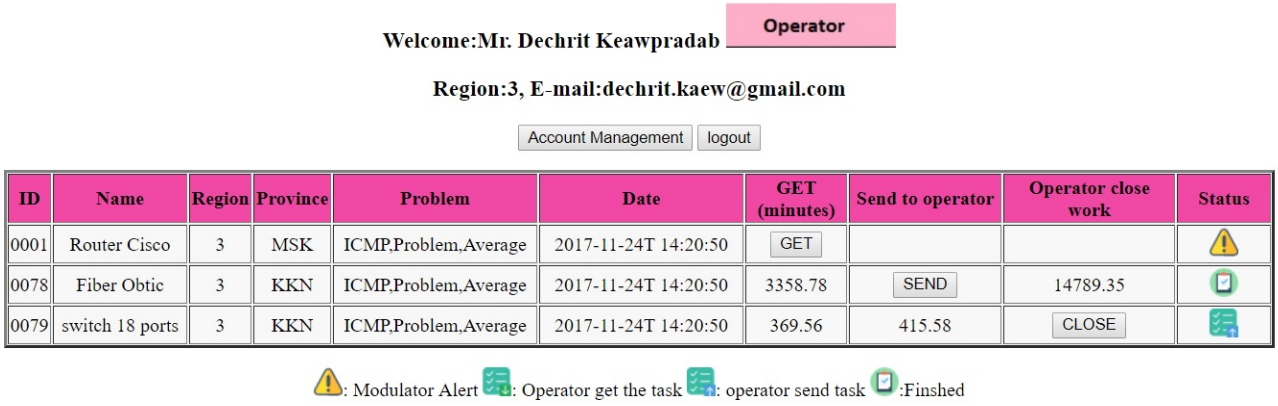


Fig. 6 The operator application form

In the Fig. 7, the second operator received the task ID 0079 from the first operator. Because Ms.Sudarat fails to get this task, the alert red sign, and the GET button appear. The second operator is the last person who has to complete the task but fails to send this task to other operators. After the GET button is selected, the CLOSE button is shown.

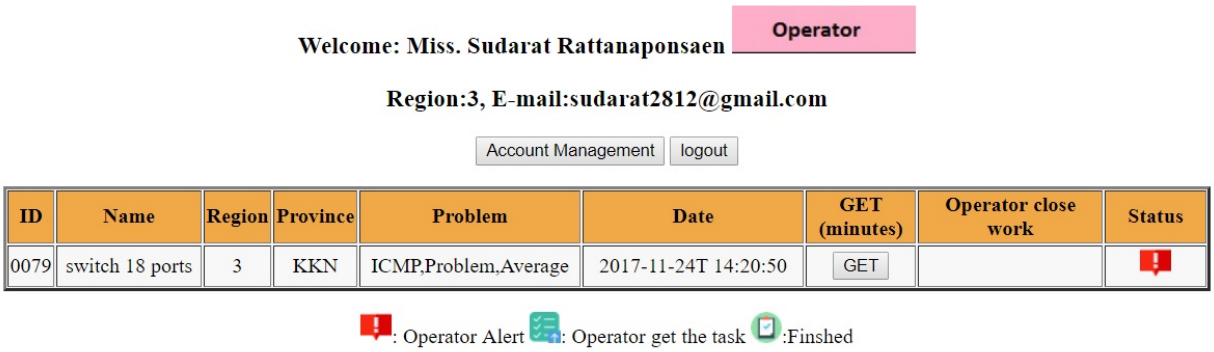


Fig. 7 The 2nd operator application form

3. Results and Discussion

In this section, we aim to measure the performance of the developed web application by considering 2 criteria including (1) the user satisfaction for testing the main and sub functions work, and (2) the total time process of maintenance of devices network using the developed application. The first criteria is selected to prove that the developed web application provides the suitable user-friendly and GUI (Graphic User Interface) to the TOT staff. The second criteria is selected to effectively evaluate the reduction of the time maintenances operation by modulators and operators.

In the first of criteria, the Likert scale is divided into five (1 to 5) levels, namely very poor, poor, moderate, good, and very good. In the evaluator, the supervisors and the staffs take the active part. The data were collected, analyzed and described as the descriptive statistics: max (Mx), min (Mn), mean (Avg) and standard deviation (S.D.) (See Table 1).

The results show that the developed web application is rated the good level (4.02 Score). The highest score is the number 6 (4.40 score); thus, we argue that the total time and operators of the finished tasks can attain the satisfaction. Meanwhile, the lowest score is the number 4, accounting for 3.60. In this case, the disadvantage of this topic is that the users cannot later edit the operator 1 and 2; result to missing the operator.

Table 1 The evaluation of the satisfaction point

Point of view	Mx	Mn	Avg	S.D.
1. Testing the registration and login of the developed web application.	5	3	4.04	0.32
2. Testing the account management button and form which enable the user to edit their profile.	5	3	4.15	0.40
3. Testing the GET button to receive the tasks and checking the displayed time stamp.	5	2	4.02	0.69
4. Testing the SEND button and transferring form which identifies the required operator and checking time stamps.	5	2	3.60	0.78
5. Checking the status in each of operational tasks.	5	2	3.89	0.99
6. Testing the Close button which completes their tasks and checking the displayed date and time.	5	2	4.40	0.44
Average	5	2.30	4.02	0.60

In the second of evaluation, we compare the total time process of maintenance of devices network between the previous process and the current process (see Table 2). The result show when the developed application is replaced on the traditional process, the total time decrease from 70 to 20 minutes, accounted for 71.42%. Moreover, the developed application enables the number of step process is

reduced from 6 to 5 steps. According to the comparison, we imply that the developed application led the user to spend the short time for maintenance of devices networks operation.

Table 2 The comparison of the maintenance of devices network infrastructure.

Previous Process	Time	Current Process	Time
1. The network devices are the failure.	0	1. The network devices are the failure.	0
2. The customer informs the failure of network devices to TOT services center.	>= 30 minutes	2. The modulator receives and assigns the failure of network devices as the task.	6 seconds
3. The TOT services center inform the failure of network devices to the modulator.	>= 10 minutes	3. The modulator sends the task to the operator via the developed application.	<= 10 minutes
4. The modulator receives and assigns the failure of network devices as the task.	process 1 to 3	4. The operator receives the task.	10 minutes
5. The modulator sends the task by manual operation to the operator.	<= 30	5. The operator checks and repairs the network devices.	Process 1 – 4
6. The operator checks and repairs the network devices.	process 1 to 5		
The total time is 70 minutes.		The total time is 20 minutes.	

4. Conclusion

The objective of this research is to develop the web application for notification of network status and assigned task management for TOT public company limited at the 2nd KhonKaen. In the concept of the developed web application, the Mongo DB is selected as the database to record the status logs of all the network devices which are retrieved from the Zabbix server. The Node.js is developed as the web application to query the status of network devices from the Mongo DB and alert it to the modulator. For the application functions, the modulator gets and assign the failure and problem of network devices as the task to operators. Furthermore, the 1st operator can subsequently assign the tasks to the 2nd operators. The time usage of the tasks operation, including receiving, transferring and finishing, are display to find the amount of time. In the evaluation, the results illustrate that the developed web application is rated the good level (4.02 Score) for the user satisfaction. Furthermore, the benefit of using the application provide the short time of the process of maintenance of devices network.

5. Suggestions

In the future work, we improve the developed web application which is able to connect with LINE application and other social media to directly alert the assigned task to modulator and operator.

6. Acknowledgement

The research was supported by The TOT public company limited, the sales regional and service center at the 2nd KhonKaen.

7. References

[1] TOT public company limited, http://www.tot.co.th, 2 August 2017.
[2] Zabbix solutions, https://www.zabbix.com, 10 September 2017.
[3] O. Marik, S. Zitta, Comparative analysis of monitoring system for data networks, Multimedia Computing and Systems (ICMCS) International Conference 2014, Marrakech, Morocco, 14 – 16 April 2014.

[4] S. Taherizadeh, A.C. Jones, I. Taylor, Z. Zhao, V. Stankovski, Monitoring self-adaptive applications within edge computing frameworks: A state-of-the-art review, J.Syst Softw. 136 (2018) 19 – 38.
[5] T. Janpan, V. Visoottiviseth, R. Takano, A virtual machine consolidation framework for CloudStack platforms, The Information Networking (ICOIN2014) International Conference 2014, Phuket, 28 – 33.
[6] K. Fatema, V.C. Emeakaroha, P.D. Healy, J.P. Morrison, T. Lynn, A survey of Cloud monitoring tools: Taxonomy, capabilities and objectives, Journal of Parallel and Distributed Computing. 74 (2014) 2918 – 2933.
[7] A. Pacharaphak, ZABBIX Monitoring Alert and Reply with LINE Application, Master of Science, Mahanakorn University of Technology, Bangkok, 2016
[8] J. Hernantes, G. Gallardo, N. Serrano, IT Infrastructure- Monitoring Tools, IEEE SOFTWARE. 32(4) (2015) 88 – 93.
[9] Mongo DB solutions, https://www.mongodb.com, 1 November 2017.
[10] Node.js Introduction, https://nodejs.org/en, 2 November 2017.
[11] OAuth 2.0 concepts, https://oauth.net/2, 5 December 2017.
[12] Nginx solutions, https://www.nginx.com, 11 January 2017.
[13] Ubuntu Operating System, https://www.ubuntu.com, 11 January 2017.