Asia-pacific Journal of Convergent Research Interchange
Vol.8, No.10, October 31 (2022), pp.37-47
http://dx.doi.org/10.47116/apjcri.2022.10.04

Development and Performance Evaluation of Chatbot System for Office Automation

Yoonkyo Jung[1]

[1] *Assistant Professor, Department of Computer Science, Republic of Korea Air Force Academy, South Korea, ykjung.rokafa@gmail.com*

Abstract: Many employees perform repetitive office tasks inefficiently, hence, spend considerable time on them. Furthermore, existing tools for saving time on repetitive tasks were developed for large-scale services such as banks and messengers. Accordingly, for small groups of workers, repetitive tasks continue to be time-consuming. This study developed a user-friendly chatbot system based on KakaoTalk and evaluated its performance in automating repetitive tasks in a small workforce. First, functional and performance requirements were identified on the basis of the KakaoTalk chatbot policy to construct a practical chatbot system. Subsequently, a KakaoTalk channel was established and a chatbot server was set up to operate the chatbot system. The first chatbot was developed in a simple manner for minimizing difficulties in its development and for user convenience. In order to meet the system requirements, the chatbot performance was improved by writing the source code of programs used by backend chatbot servers in different programming languages. Furthermore, the use of several techniques such as database redesign and load balancing for improving the server performance was examined. For the determination of the most effective technique for enhancing the chatbot performance, the total request per second and average latency were measured. By evaluating and comparing the performance of the first chatbot and the newly developed chatbot, it was verified that the chatbot system met the functional and performance requirements. Thus, this study shows what requirements should be set to operate the chatbot system and which method is effective to improve the chatbot's performance.

Keywords: Chatbot, Database, Golang, Office Automation, Performance Analysis

1. Introduction

With the development of science and technology, many organizations are striving to automate repetitive tasks to reduce the workload on employees. In particular, chatbots are used in online banking for real-time interactions with customers [1], and online travel agencies also use chatbots as a consumer-facing tool[2]. However, chatbot systems are mostly used for large-scale services. Small groups of office workers typically perform administrative tasks inefficiently. For example, in the task of acquiring specific information about the members of a group, it is time-consuming and expensive to enter the information directly into an open Excel document. If the group uses the information for several purposes, employees need to manually organize the information. Such tasks are simple, but they are time-consuming.

In order to reduce the work burden on employees by eliminating unnecessary work, many companies minimize work by using a bulletin board that is accessible to all employees[3]. Online survey tools such as Google Forms simplify the task of conducting surveys[4]. However, when a bulletin board is used, personal information can be exposed, since all employees can access it. Furthermore, it is difficult to

Received: June 30, 2022; 1st Review Result: August 12, 2022; 2nd Review Result: September 16, 2022
Accepted: September 30, 2022

manage collected information using Google Forms because the information is stored on an external server. Other online survey tools lack functions that administrators can check and save the survey data whenever they want. To automate the repetitive tasks, this study developed a KakaoTalk chatbot that simplifies the task of collecting and managing information. The chatbot was improved by using several techniques, and its suitability for providing useful chatbot services with small groups of workers is demonstrated. The improved chatbot meets all the requirements for actual use in a work environment.

2. Related Work

A chatbot refers to a chatting robot, which is a software program designed to simulate human-like conversations online through text messages or speech[5]. It can respond to user messages in real time, and there are many types of chatbots that cater to different user requirements.

2.1 Chatbot Development

In previous studies, various chatbots have been developed and tested to provide customized services for users. Han[6] checked the effects of voice-based chatbots by Middle school students for studying English speaking. Dong-ah Park[7] showed that Artificial-intelligence-based chatbots have been used by various industries to provide customer services on behalf of humans in non-face-to-face situations. In particular, Choi et al.[8] used chatbots for providing psychological health services, especially to provide psychological counseling to university students, during the COVID-19 pandemic. Yeen et al.[9] used chatbots for helping college students choose their major. The chatbots answer to specific questions of students and identify appropriate majors from a database on the basis of the information provided by students. Lim[10] examined the use of chatbots for teaching the subject of artificial intelligence to students. These studies showed that new features of chatbots can provide convenience to users, but there were no experiments to evaluate the performance of developed chatbots.

2.2 Performance Evaluation

Many previous works evaluated the function of the system with performance analysis. Bansal & Shricastava[11] and Rao[12] evaluated the performance of communication systems for mobile devices by comparing network throughput. Jader et al.[13] suggested the idea of controlling the load in the web servers to satisfy internet users and web servers from being overworked. Przegalinska et al.[14] designed a system for analyzing chatbot messages to measure chatbot performance. This study developed a chatbot and evaluated chatbot performance referring to these related works.

3. Research Methodology

This section presents the design of the chatbot system and explains how an experiment proceeded. A practical and user-friendly chatbot system based on KakaoTalk was developed, and experiments were conducted to evaluate the chatbot's performance.

3.1 System Design

3.1.1 Development Objectives

Optimizing a chatbot's server is important to improve the chatbot's performance. In this study, upgrading the server hardware was not considered for enhancing the chatbot system's performance. The objective was to design a practical chatbot system by modifying a message-processing program and a

network structure of the chatbot server.

Functional and performance requirements were set to construct a useful chatbot system for a real workplace. According to the KakaoTalk chatbot's policy, no response is provided to a request if the server does not respond within a specific time. Therefore, the server should have the capacity to handle a large number of queries when users access the chatbot concurrently. The server should also respond to requests on time.

[Table 1] Functional Requirement of the Chatbot System

Index	Functional Requirement
1	Chatbot has to identify user's name and authority.
2	Users have to be allowed to participate, check and cancel survey.
3	Administrators have to be allowed to make, check and delete survey.
4	Administrators have to be allowed to download csv file that contains results of survey.

[Table 1] shows the functional requirements that chatbot systems should meet to provide essential functionalities to users. A chatbot should check the user's name and authority to identify who sends the chatbot messages. It is also important to distinguish the authority to segregate the administrator and user functions. Only the administrator should have the authority to manage surveys, such as creating and deleting questionnaires, and after users complete their survey, the administrator should be able to check and download the survey results.

[Table 2] Performance Requirement of the Chatbot System

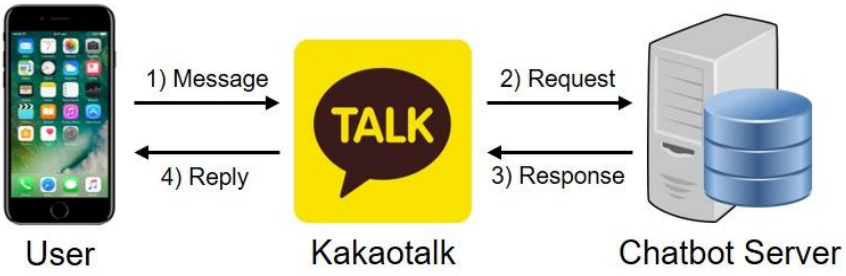
Index	Performance Requirement
1	Chatbot has to reply all request at least 100 per second.
2	Chatbot has to reply all request at least in 1 second under situation of 100 concurrent users.

The chatbot system should be capable of increasing the throughput and latency per second when multiple users use the service simultaneously. [Table 2] presents performance requirements for the use of the chatbot system in an actual workplace. If the chatbot server does not respond to a request on time, no reply message will be provided to the user. To avoid such a situation, the chatbot should be capable of responding to 100 requests/s. Even if there are 100 concurrent users, it should be capable of responding within one second.

3.1.2 Chatbot Development

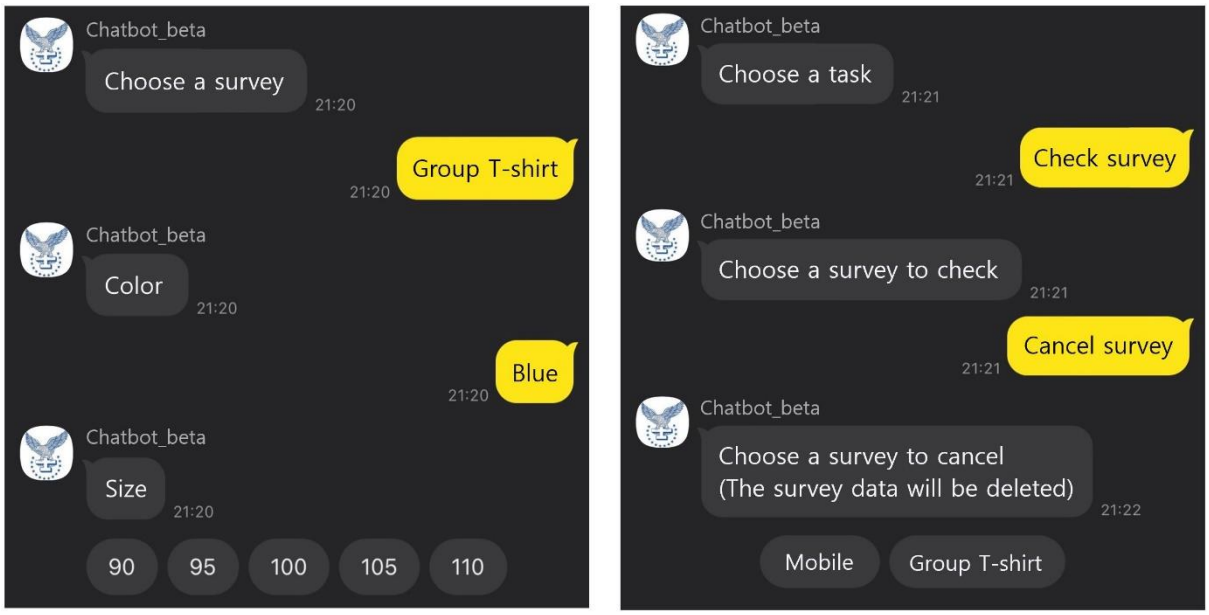
After the system requirements were set, a KakaoTalk-based chatbot service with high user accessibility was developed. For the construction of a KakaoTalk-based chatbot, a KakaoTalk channel was established to receive messages from users. A chatbot server was also set up to process the messages.

Before the chatbot was developed, a KakaoTalk channel service was created and Kakao i Open Builder was set up[15]. The KakaoTalk channel could provide an interactive service to users. Users could easily send a message after adding the KakaoTalk channel. The channel was connected to a chatbot server to customize the survey function and manage the user messages.



[Fig.1] A Process of Message Request and Response using Kakaotalk Chatbot

[Fig. 1] illustrates the process of receiving a user's request and responding via the KakaoTalk channel. When a user sends a text message to the chatbot channel through KakaoTalk, JSON data are transmitted to the chatbot server. The data contain information such as the user's ID, properties, and utterance. Upon receiving the data from the KakaoTalk channel, the chatbot server processes the request message for generating a response. The server generates an appropriate response message and delivers it to the user.

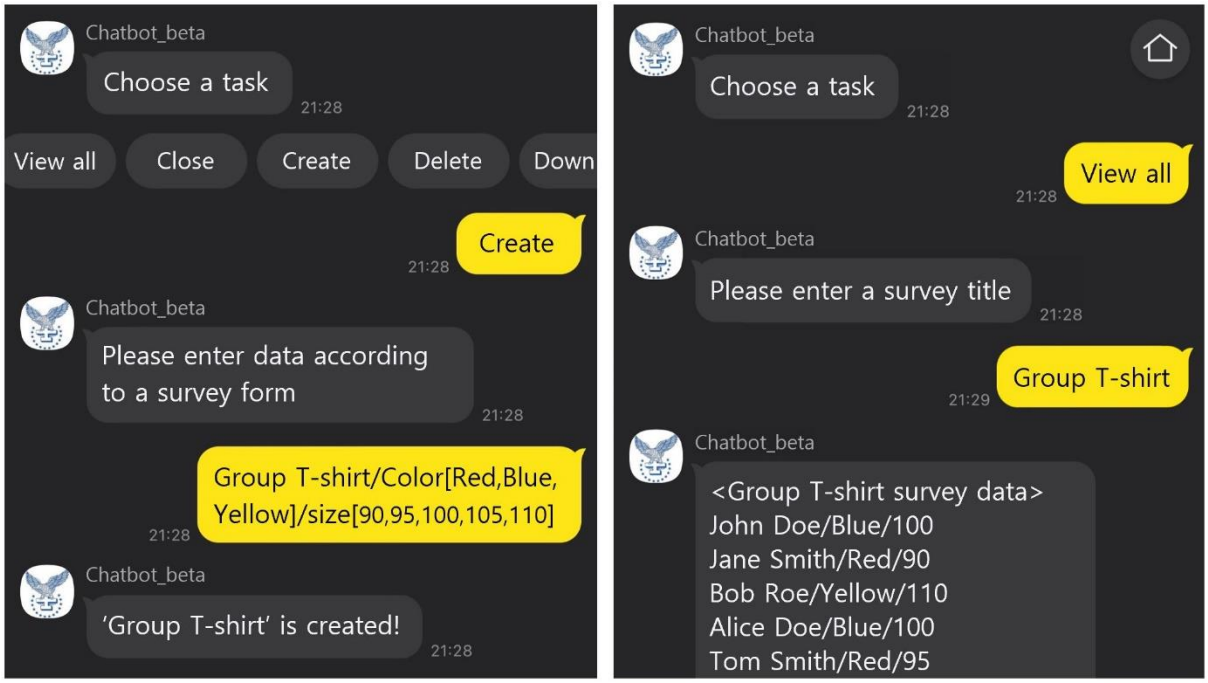


[Fig. 2] An Example of User Function

User and administrator functions are implemented separately. [Fig. 2] shows a typical example of a user function. Users can participate in a survey created in the chatbot channel. Furthermore, they can check their input data and withdraw from the survey.

As shown in [Fig. 3], administrators can create and delete surveys. They can also access all the information obtained from the survey and download the information whenever they want.

A chatbot server was set up to process messages from the KakaoTalk channel and to save user information. The chatbot server was constructed using Amazon Web Service (AWS) Free Tier, which helps all users construct for free. The operating system of the chatbot server is ubuntu 22.04.



[Fig. 3] An Example of Administrator Function

```
{
    "userRequest":{
        "timezone": "Asia/seoul",
        "params": {},
        "block": {
            "id": "<Block id>",
            "name": "<Block Name>"
        },
        "utterance": "<User Conversation>",
        "lang": "kr",
        "user": {
            "id": "<User botuserKey>",
            "type": "botUserKey",
            "properties": {
                "plusfriendUserkey": "<Kakaotalk Channel User id>"
            }
        }
    }
}
```

[Fig. 4] User Request Message in JSON Format

[Fig. 4] shows a JSON request message. The [userRequest][user][id] segment contains a botUserKey that is uniquely assigned to each user, and [userRequest][utterance] includes user speech. Therefore, the sender of a message can be identified and the contents of the message can be saved on the backend server. The server stores [userRequest][user][id] and [userRequest][utterance] segment pair to provide the administrator with KakaoTalk messages.

The first chatbot was designed to minimize difficulties in its development and by considering user convenience. The program for processing information received by the server was written in Python, and the server stored data in Excel sheets, which the administrator could easily check. The chatbot had performance issues when users accessed the chatbot system simultaneously. If the number of user requests increased, the response was delayed and the chatbot server was paralyzed. Therefore, the program code was migrated from Python to Golang, the database was redesigned from Excel to SQLite3, and load balancing was set to improve the system performance without upgrading the server hardware.

3.2 Experiment

The first developed chatbot did not meet the system requirements. On the other hand, the newly developed chatbot was able to satisfy the performance requirements. The performance of the improved chatbot was compared with that of the first chatbot to check if it was suitable for use in an actual workplace. For the measurement of the performance of the chatbot, a given JSON format POST was repeatedly sent as the request message and the performance was evaluated from the total request per second and the response time. An open-source program, Locust, was used as the performance testing tool. Locust sent a POST request to the chatbot server and extracted the total request per second and the response time from HTTP requests. After the measurement of the chatbot performance, the results were compared to identify which chatbot shows a better performance. Also, the new chatbot was operated for more than 800 cadets at the Korea Air Force Academy to check if the chatbot is suitable for use in a real work environment.

4. Result and Discussion

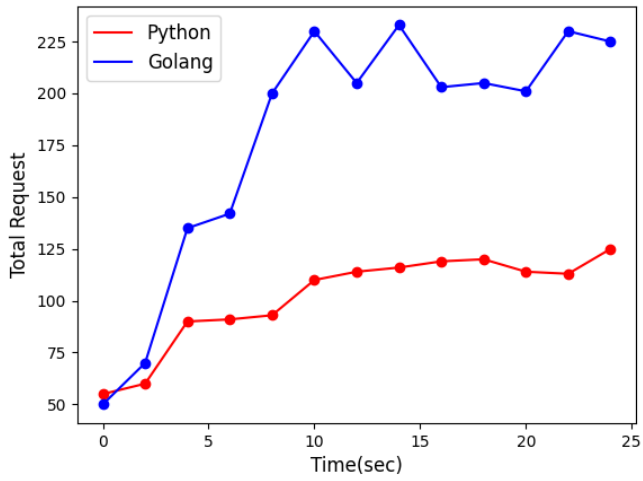
4.1 Result

In this section, the researchers describe the results of experiments on how much the chatbot performance is enhanced by modifying a message-processing program and a network structure of the chatbot server. After listing the results, the findings of the study are discussed to make sure that the main objectives are achieved.

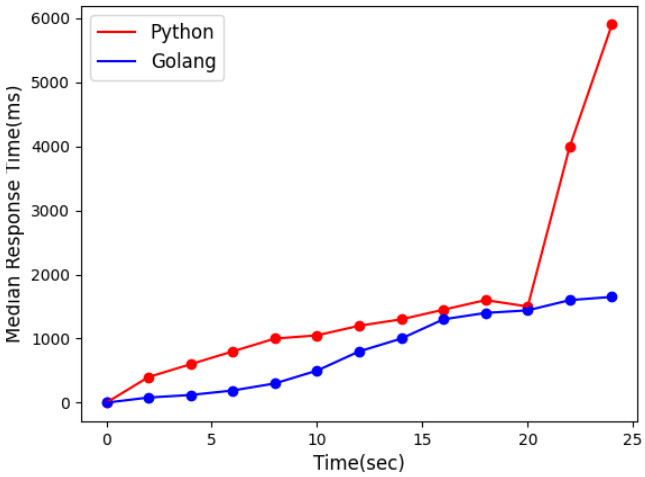
4.1.1 Performance Evaluation

In order to meet the system requirements, the authors designed the chatbot to store and process messages differently. As mentioned, code migration, database redesign, and load balancing were conducted to improve the performance of the KakaoTalk server. The performance of the server was determined after each of these techniques was applied, to ascertain which of these techniques worked best.

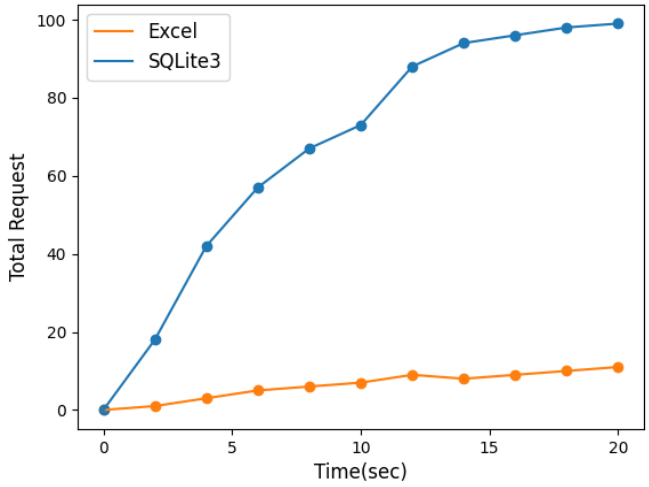
First, the source code language of the backend server was migrated from Python to Golang. The message-processing program of the initial chatbot server was written in Python 3 and Flask to simplify the development of the chatbot. To improve the chatbot's performance, the authors developed the chatbot's server using Golang and the net/http module, which shows better HTTP server performance. [Fig. 5] shows a comparison of the total number of requests per second before and after the code migration. Clearly, the chatbot server showed a higher capability to respond to requests when Golang was used. From [Fig. 6], it is apparent that the Golang program responded faster than the python program.



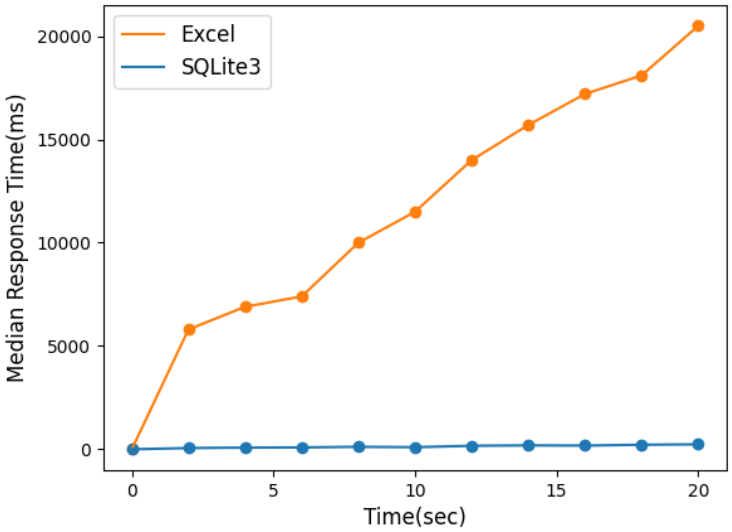
[Fig. 5] Comparison of Total Request per Second between Python and Golang



[Fig. 6] Comparison of Median Response Time between Python and Golang

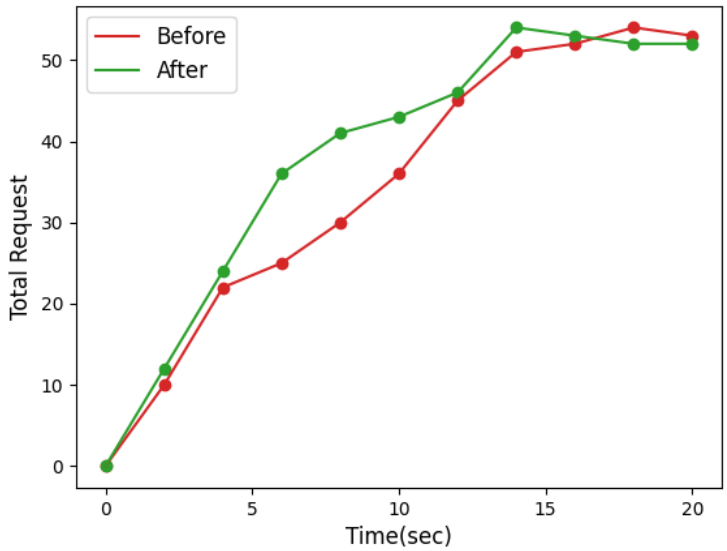


[Fig. 7] Comparison of Total Request per Second between Excel and SQLite3



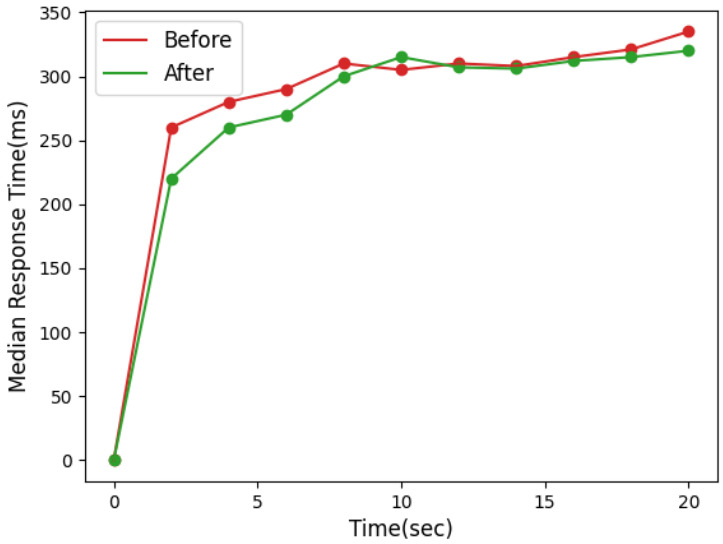
[Fig. 8] Comparison of Median Response Time between Excel and SQLite3

[Fig. 7, 8] show the performance improvement after the database was redesigned. Previously, data were stored in an Excel file so that administrators could easily check the survey data. However, the latency was high. Therefore, the researchers changed the database model to SQLite3. The results showed that the total number of requests per second that SQLite3 could handle was more than 40 times greater and that it was 100 times faster in responding to requests.



[Fig. 9] Comparison of Total Request per Second before and after Load Balancing

Load balancing was set to distribute the network traffic. Several back-end programs were run on limited server resources. As shown in [Fig 9, 10], however, there was no significant impact on performance improvement. It is because we applied the load balancing on a single server.



[Fig. 10] Comparison of Median Response Time before and after Load Balancing

4.1.2 Requirements Check

The improved chatbot was operated for actual users and operators to examine whether it met the functional and performance requirements.

[Table 3] The Result of Functional Requirements Check

Category	Requirement	Result
Function	Chatbot has to identify user's name and authority.	✓
	Users have to be allowed to participate, check and cancel survey.	✓
	Administrators have to be allowed to make, check and delete survey.	✓
	Administrators have to be allowed to download csv file that contains results of survey.	✓

[Table 3] presents the result of the functional requirement check. Evidently, the chatbot met all the functional requirements.

[Table 4] The Result of Performance Requirements Check

Category	Requirement	Result
Performance	Chatbot has to reply all request at least 100 per second.	✓
	Chatbot has to reply all request at least in 1 second under situation of 100 concurrent users.	✓

The performance requirements were checked to verify whether the chatbot satisfied the KakaoTalk chatbot policy. [Table 4] presents the result of the performance requirement check. The chatbot could

respond to concurrent users and was suitable for use in real workplaces.

4.2 Discussion

The experiment was conducted on a single server in an environment in which anyone could easily develop chatbots. Three techniques were used to enhance the performance of the initially developed chatbot, and the best technique among them was determined by measuring the server's performance. It was found that code migration and database redesign were effective in improving the chatbot performance, while load balancing on a single server had no effect. The chatbot was also operated in an actual work environment to check whether it met the system requirements. The chatbot satisfied all the functional and performance requirements and did not cause problems during its operation by a small group. In other words, it was determined which method was more effective to improve the chatbot's performance, and there were actually no issues when it was operated by a small group.

5. Conclusions

The primary objectives of this study were to develop a chatbot system suitable for use by small groups of users and to evaluate the chatbot's performance for verifying its suitability for use in an actual work environment. The experimental results clearly showed that the chatbot service met all the functional and performance requirements, which is necessary for preventing problems during the use of the chatbot system. Both users and administrators can use the chatbot in real workplaces.

This study has two limitations. First, it did not consider the functions users want. Second, it is difficult for workers to develop a chatbot service in a short time. For the development of a practical method for chatbot construction, further research is required to find a way to determine user preferences and simplify the chatbot construction task.

References

[1] Jon T. S. Quah, Y. W. Chua, Chatbot Assisted Marketing in Financial Service Industry, International Conference on Services Computing, Springer. Cham, pp.107-114, (2019)
DOI: https://doi.org/10.1007/978-3-030-23554-3_8

[2] Lin Li, Kyung Young Lee, Emmanuel Emokpae, Sung-Byung Yang, What Makes You Continuously Use Chatbot Services? Evidence from Chinese Online Travel Agencies, Electronic Markets, (2021), Vol.31, pp 575-599.
DOI: https://doi.org/10.1007/s12525-020-00454-z

[3] L. Hirschi,L. Schmid, D. Basin, Fixing the Achilles Heel of E-voting: The Bulletin Board, 2021 IEEE 34th Computer Security Foundations Symposium (CSF), pp.1-17, (2021)
DOI: http://dx.doi.org/10.1109/CSF51468.2021.00016

[4] https://www.google.com/forms/about/, Sept 16 (2022)

[5] Adamopoulou. E, Moussiades. L, An Overview of Chatbot Technology, Proceeding of IFIP International Conference on Artificial Intelligence Applications and Innovations, Springer. Cham, pp.373-383, (2020)
DOI: https://doi.org/10.1007/978-3-030-49186-4_31

[6] Da-Eun Han, The Effects of Voice-based AI Chatbots on Korean EFL Middle School Students' Speaking Competence and Affective Domains, Asia-pacific Journal of Convergent Research Interchange, (2020), Vol.6, No.7, pp.71-80.
DOI: http://dx.doi.org/10.47116/apjcri.2020.07.07

[7] Dong-ah Park, A Study on Conversational Public Administration Service of the Chatbot Based on Artificial Intelligence, Journal of Korea Multimedia Society, (2017), Vol.20, No.8, pp.1347-1356.
DOI: https://doi.org/10.9717/kmms.2017.20.8.1347

[8] Dajung Choi, Aeri Cho, Changdae Kim, Designing a University Counseling Center Guiding Chat-bot for Improving the Accessibility in COVID-19, The Korean Journal of Public Health, (2021), Vol.58, No.2, pp.13-24.
DOI: https://doi.org/10.17262/KJPH.2021.12.58.2.13

[9] Yeen HeuiYeen, Shin YongHa, Ko WooJun, Lee SeokJune, Koo MyoungWan, Coversational Major Recommendation Chatbot that Allows General Conversation and Major Recommendation, Proceeding of HCI Korea 2022, The HCI Society of Korea, pp.764-767, (2022)
Available from: https://conference.hcikorea.org/hcik2022

[10] Mika Lim, Development of AI Convergence Invention Education Program using Chatbot Tutor, Journal of the Korean Institute of industrial educators, (2022), Vol.47, No.2, pp.198-222.
DOI: https://doi.org/10.35140/kiiedu.2022.47.2.198

[11] Monika Bansal, Laxmi Shricastava, Performance Analysis of Wireless Mobile Adhoc Network with Different Types of Antennas, Asia-pacific Journal of Convergent Research Interchange, (2017), Vol.3, No.1, pp.29-39.
DOI: http://dx.doi.org/10.21742/apjcri.2017.03.03

[12] N. Thirupathi Rao, Performance Evaluation of Optimized Tandem Communication Network Model, Asia-pacific Journal of Convergent Research Interchange, (2019), Vol.5, No.3, pp.211-220.
DOI: http://dx.doi.org/10.21742/apjcri.2019.09.21

[13] O. H. Jader, S. R. M. Zeebaree, R. R. Zebari, A State of Art Survey for Web Server Performance Measurement and Load Balancing Mechanisms, International Journal of Scientific & Technology Research, (2019), Vol.8, No.12, pp.535-543.
Available from: https://www.ijstr.org/final-print/dec2019/A-State-Of-Art-Survey-For-Web-Server-Performance-Measurement-And-Load-Balancing-Mechanisms.pdf

[14] Aleksandra Przegalinska, Leon Ciechanowski, Anna Stroz, Peter Gloor, Grzegorz Mazurek, In Bot We Trust: A New Methodology of Chatbot Performance Measures, Business Horizons, (2019), Vol.62, No.6, pp.785-797.
DOI: https://doi.org/10.1016/j.bushor.2019.08.005

[15] https://i.kakao.com, Sept 16 (2022)