Automation Tool for Case Assignment

Siddharth Mitta[1], Anitha Sandeep[2],

[1] UG Student, Department of CSE, R.V. College of Engineering, Karnataka, India
[2] Assistant Professor, Department of CSE, R.V. College of Engineering, Karnataka, India

ABSTRACT

In a production environment with huge computation to be performed it's easier to perform task by offloading tasks to machine instead of wasting man hours. Thus, an automated system has provided the organization the ability to perform the same tasks more efficiently with reducing the human errors. Hence with automation system in production it provides a better decision-making result. Annual survey of industries report 2018 states that more than 75% of industries are moving towards automation systems to perform tasks. The Worldwide Support pick up cases on a daily basis where each case is a specific problem pertaining to a certain customer. To make the process of the allotment of cases to the Support team much easier, the Support team uses Auto Case Assignment Bot. Auto case assignment Bot is a management tool that is used to suggest case to the Support Engineer's based on their skills and based on shift timings and case priority. This bot manages to solve the problem of case assignment which reduces the disparity in the number of cases being assigned to each employee.

Keywords: Salesforce, Service level agreement ,8X8

1. INTRODUCTION

The Auto Case Assignment Bot, is a management tool that can be used by the Worldwide Support Team of any company to suggest cases to the System Reliability Engineers based on a number of metrics such as their skills, availability on a particular day and the priority of the case that has been raised by the customer. Before the implementation of auto case assignment bot, cases were presented to the System Reliability Engineers manually which would lead to a lot of disparity in the number of cases picked by each person and thus, this was one of the major problems that this tool resolved once it was brought into production. It also saved a lot of manpower and made the managers work much easier as it provided a centralized platform to keep track of the number of cases each System Reliability Engineer handles in a day. competitive market environment business organization are aiming to acquire a greater number of customers, customer satisfaction is an important factor for the success of business [1]. It is observed that satisfied customers tend to become loyal customers helping in company growth. Therefore, its seen that investing on new customer is difficult then retaining the loyal customer. Also dissatisfied customers tend to spread out negative words about the company in the public which can create a bad impression of the product in the minds of other customers preventing them from buying from the company. Loyal customers share their good experience with others and also would recommend the product to others which leads to more customers approaching the company. Therefore, customer support plays an important role in an organization.

The report focuses on designing an automated system [2][3] and regulating this system. It is essential in the production environment to obtain accuracy with the system as well as remove human intervention as it leads to human error[4]. The system focuses on ensuring an even distribution of cases among the Support Engineer as well creating an alert message if required based on status of Support Engineer, service level agreement being missed and backlogs present. The project also focuses in designing a dashboard as easy user interface for the bot.

2. RESEARCH METHODOLOGY

Fig 2.1 shows the proposed model flow chat. The auto case assignment bot gets the cases information from salesforce wherein we obtain the case ID, case type, case owner and date the case was created. The support engineer status and skill are obtained 8X8 which provides the bot to separate support engineer who are available and assign cases to these engineers. The obtained result is sent using chat app to notify the engineer about the case assigned.
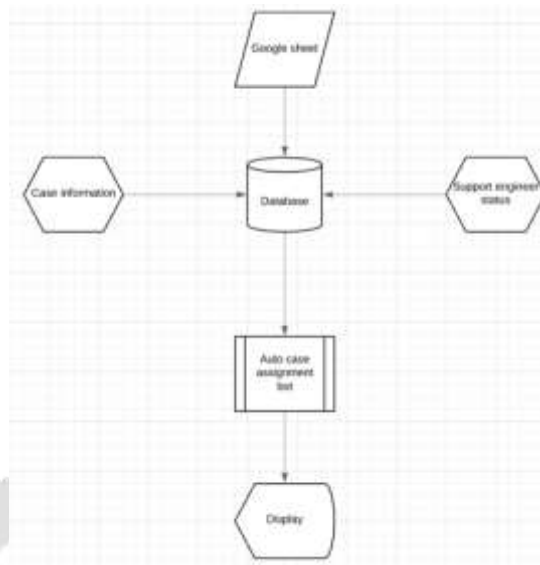
Figure 2.1: Methodology

A. SLA missed alert message

The auto assignment bot sends alert message to manager to ensure no SLA is being missed. Fig 2 shows the SLA flow chart where the case details are fetched from salesforce [5]. Based on case information and date of creation of case each case is provided an SLA. SLA varies depending on cases priority.

The flow chart shows the case details such as owner of case and priority to calculate the SLA. Engineer id are fetched for each engineer and checks if SLA is being missed by the engineer [6]. In case the SLA is being missed an alert message is sent to engineer and manager.
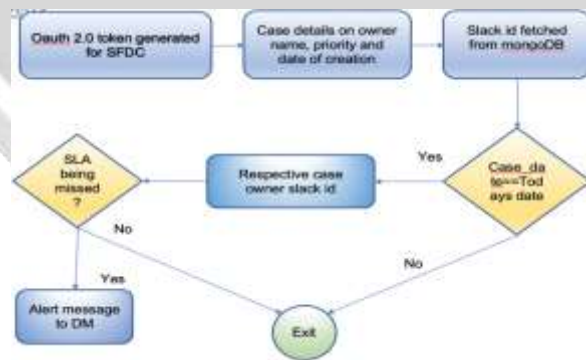
Figure 2.2: SLA missed flow chart

B. Backlog calculations for assigning cases

It's important to check the case backlog before assigning any case to engineer. Thus, backlogs are checked for each engineer and based on the current backlog case is assigned. An engineer with least backlog and whose skills match with case are considered for assigning cases [7].

Fig 2.3 shows the flow for backlog calculations where in case at different scenario are checked before incrementing the backlog counter. Only cases with customer response is not given or engineer is working on the case is considered as backlog [8].
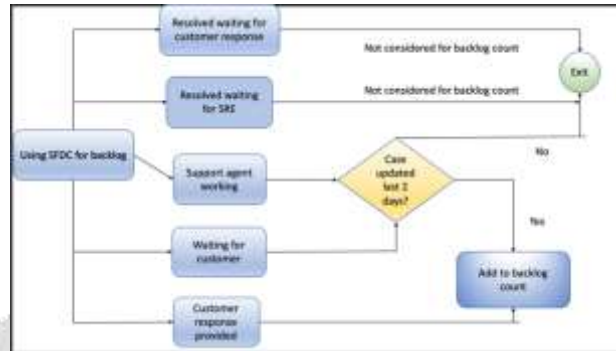

Figure 2.3: Backlog calculations

C. Alert message based on status

It's important for auto case assignment bot to know the status of engineer before assigning cases. But it's important for the bot to send an alert message if an engineer is on break or work offline for longer than 30 minutes.

Fig 2.4 shows flow chart for alerting the engineer and manager if an engineer is on break or work offline for more than 30 minutes. This ensure the engineer to change their status so that the bot can assign cases to them [9][10].
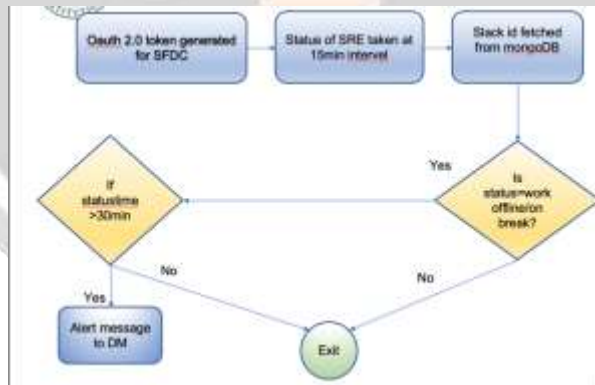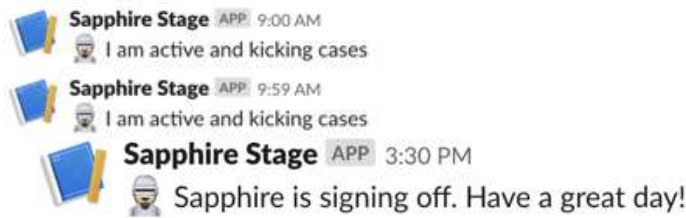

**Figure 2.4:** Flow chart for alert message based on status

3. RESULTS AND DISCUSSION

The objective for designing an automated case assignment bot as management tool was completed. To know the status of the bot and to check if bot is assigning cases to engineer, an user interface was made [11]. Fig 3.1 shows the user interface to check the status of case assignment bot.

Figure 3.1: Bot status

The objective to check the alert message were sent to engineer on various scenario was also checked and the results were obtained using user interface as shown in Fig 3.2.
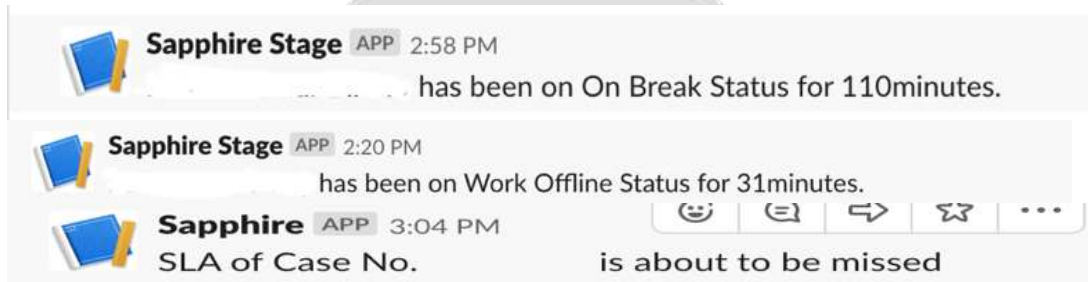
Figure 3.2: Alert message results

A dashboard as shown in Fig 3.3, was created for better user interface for manger to check the number of cases assigned by auto case assignment bot to engineer. The dashboard also shows the individual engineer cases accepted and rejected. Thus, the manger can track each engineer work as well as the case history.
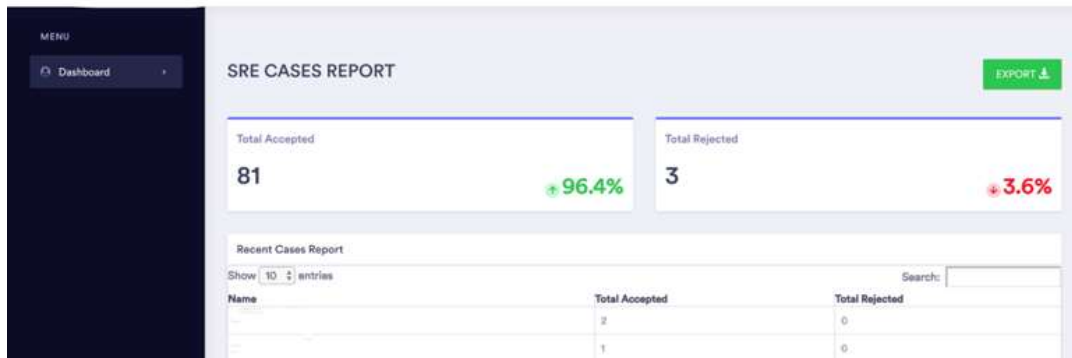
Figure 3.3: Dashboard

4. FUTURE WORK

With the rapid expansion of the company, there has been an increase in the number of employees and the cases to be tackled and hence a need for a more efficient and automated management tool for scheduling and tracking work progress of the employees. With an easy to use interface and simplified data flow, the proposed solution will reduce the hassle and scope for errors.

The currently designed system automates the process of assigning cases to the employees, thus, reducing the human dependency. The notification system - Slack responds in a timely manner. The case suggestion model enables the

employee to reject a case if he or she wants to do so. The bot has more functionalities such as a real-time dashboard which contains information regarding the cases accepted or rejected. The currently implemented system helps over a hundred employees per day.

To overcome the limitations of the project, the following will be taken up in the future:

- The employee seniority will be taken into consideration when assigning the cases and higher priority cases will be given to senior employees.

- A kill switch will be introduced to ensure graceful shutdown of the bot.

- Feature health monitoring has to be introduced to ensure that all the modules of the bot are working properly.

5. REFERENCES

[1] J Markovski, H. Hu, "A synthesis-centric model-based systems engineering framework for reliable supervision of systems with general distributions", *Paper presented at: IEEE International Conference on Automation Science and Engineering*, 2014.

[2] Q. H. Wu, D. P. Buse, J. Q. Feng, "E-Automation, an Architecture for Distributed Industrial Automation Systems", IEEE International Journal of Automation and Computing, 2004, pp.17-25.

[3] Q. H. Wu, D. P. Buse, "An architecture for e-Automation", IEEE computer &control engineering, 2003, pp.38-43.

[4] Nian.De. Jiang, Yu.Cheng. Wei, "Substation Automation System Architecture Study", Grid technology, vol. 27, no.10, 2003, pp. 48-51

 [5] Cornelia Győrödi, Robert Győrödi , George Pecherle , Andrada Olah," A comparative study: MongoDB vs. MySQL", 2015 13th International Conference on Engineering of Modern Electric Systems (EMES), , vol. 10, no. 3, pp. 1609-1510, 2015

[6] Tadepalli Sarada Kiranmayee, "A Survey on the Role of Cloud Computing in Social Networking Sites", *International Journal of Computer Science and Information Technologies*, vol. 6, no. 2, pp. 1509-1512, 2015.

[7] A. Singh, S. Sharma, S. R. Kumar, S. A. Yadav, "Overview of PaaS and SaaS and its application in cloud computing", *2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*, pp. 172-176, 2016.

[8] A. Jula, E. Sundararajan, Z. Othman, "Cloud computing service composition: A systematic literature review", *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3809-3824, 2014.

[9] Li Liang, Ligu Zhu, Wenqian Shang , Dongyu Feng , Zida Xiao, "Express supervision system based on NodeJS and MongoDB", 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), , vol. 11, no. 3, pp. 1520-1612, 2017

[10] Takeshi Ogasawara, "Workload characterization of server-side JavaScript", 2014 IEEE International Symposium on Workload Characterization (IISWC), , vol. 5, no. 1, pp. 1009-1012, 2015

[11] Tadepalli Sarada Kiranmayee, "A Survey on the Role of Cloud Computing in Social Networking Sites", International Journal of Computer Science and Information Technologies, vol. 6, no. 2, pp. 1509-1512, 2015