

# RAB - Remote Assistant Bot

Sumukh Bapat

sumukhbapat@vt.edu

Department of Computer Science  
Virginia Polytechnic Institute and State University  
Blacksburg, Virginia, USA

Shawal Khalid

shawal@vt.edu

Department of Computer Science  
Virginia Polytechnic Institute and State University  
Blacksburg, Virginia, USA

Pratik Nalage

pratikn@vt.edu

Department of Computer Science  
Virginia Polytechnic Institute and State University  
Blacksburg, Virginia, USA

Shunyu Yao

shunyu@vt.edu

Department of Computer Science  
Virginia Polytechnic Institute and State University  
Blacksburg, Virginia, USA

## ABSTRACT

With the absent of the accessibility and time-sensitiveness to/of in-person communication under COVID, cooperation among software engineers becomes unproductive and inefficient. Team members can get stuck on questions that can be easily answered in a in-person communication scenario while they have to wait for hours in a remote working setting, just because the other team members they want to communicate with are not available; and answering through texting can be just as time-consuming. We propose a simple and direct solution to this problem: Remote Assistant Bot (RAB), an automated platform dedicated to fast and customized software engineering collaboration. Its question-answer-oriented workflow enables easy management of remote assistant among team members, all for creating a time-saving and smooth collaboration experience as the whole world is moving online.

### ACM Reference Format:

Sumukh Bapat, Pratik Nalage, Shawal Khalid, and Shunyu Yao. 2022. RAB - Remote Assistant Bot. In *Proceedings of Virginia Tech (CS5704)*. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 INTRODUCTION

With the COVID-19 pandemic entering the stage of the Omicron variance, its impact on our daily life has been evolving quickly around us, for the least of it, the software engineering community. The conventional in-person working environment has become nearly impossible to desire and we are seeing a huge trend for all companies to shift to a remote working mode. One of the biggest impact is that the communication becomes really slow and inefficient.

The first priorities of collaboration tools is not only just communication, but also message indexing. On such tools, developers communicate not only the triviality of work life, but more often, they also ask technical questions, and expect useful answers such

as code snippets, configurations, parameters, links, papers, etc. The existing online collaboration tools, such as Slack, attempt to simulate just-in-time conversations in a social media fashion and slice conversations into chronological discrete messages, the important information mentioned above would be lying around in the chronological order, giving developers a hard time to categorize and find after a while. Therefore, such a workflow does not necessarily fit the requirement of providing a remote work environment: (1) The just-in-time setting of messaging makes conversations in-continuous, users have to wait for others to finishing typing, during which one can easily lose patience and switch to other tasks, then the other side of the conversation will have to wait again. This conventional messaging workflow might work under the scenario of social media, but it induces significant latency for a remote working environment; (2) Having chronological waterfall as the major way of displaying messages is not a suitable adaptation for a remote working setting, either. Timestamps might be a useful piece of information, but it is not the number 1 property of messages containing important technical information, but rather the categories of these messages. In other words, when is not important, but what.

Based on the highlights observed above, we propose a remote working collaboration tool: Remote Assistant Bot (RAB). To tackle with the problems mentioned above, RAB will utilize a question-answer-oriented workflow.

## 2 RELATED WORK

Slack is a popular remote working collaboration platform that takes over the software engineering community in recent years. It utilizes a traditional waterfall chronological display of messages, with some extra functionalities that accommodate SE users, such as posting code. [4]

Remote Access Plus is a secure and broad enterprise remote support solution that helps system administrators and IT help desk technicians troubleshoot Windows, Mac, and Linux computers from a central location [2]. The users can access endpoints from anywhere, anytime, connect to remote computers with unattended remote access, seamlessly switch between multiple monitors connected to client computers, perform administrative operations using Quick Launch, diagnose computers without hindering productivity, access Command Prompt using system account and have other features.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

CS5704, February, Blacksburg, VA, USA

© 2022 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

GoToMyPC is a remote access desktop software that enables the user to work from home [3]. It allows access to files, applications, and network without any hassle. It helps in increasing your business productivity. This application has high security. You can get remote access from Android, Kindle, iPad, and iPhone. It is easy to install. It also helps you with transferring files between computers with ease. This tool supports multi-monitors. Users can copy-paste files between connected PCs.

With the spread of the COVID-19 epidemic, vaccination against COVID-19 has become an important means to control the epidemic [5]. The social bots used software engineering methods to develop and are now used in various social media platforms that affected the public perceptions of vaccine effectiveness and shared public health knowledge, complicating the climate of opinion on social media. The engagement of social bots in social media developed new communication strategies for software engineers.

COVID Robot is a robot build that can spot two or more people standing within 6 feet of each other could play an important role in maintaining social distancing requirements during the current or a future pandemic [1]. This robots locates, approaches, and notify groups of two to six people who are standing too near each other to move apart.

### 3 DESIGN

There are three major components that form our system: we need a database to store all relevant information, a server to process the user queries and a messenger that acts as a front-end for the user.

- *Messenger* - We're using Facebook messenger as our front-end platform. A user will send a query as a message to and the bot will respond via replies via messenger. Since most of the user's already have messenger, we can leverage it and use it as our front-end.
- *Application Server* - A server that handles the requests, processes the data from the front-end and interacts with database accordingly. It is further composed of: (1) Tokenizer - In order to better understand the user's query, we will break it down into words and tokenize it. If the tokens matched, we can return the corresponding answer data as a response. If the token is not found in the database, the user's query will be redirected to the available software engineers via a webhook; (2) Linguistic - This component will try to understand the user's query by using a Natural Language Processing unit. It will try to learn the semantics of the English language by preprocessing and removing unnecessary tokens; (3) Webhook - a Javascript implemented webhook used to redirect users to either another senior teammate or web-crawled answers.
- *Database* - Database that stores metadata regarding questions and answers.

In general, users start their queries by typing in their questions in the Messenger front-end, they can give their questions more descriptions. After the questions being sent to the server's side, the tokenizer will extract a list of keyword tokens. These keywords will be used to find the most similar questions that are already answered in the database, and the answers will be returned to users. But if existing answers are not similar enough, users will eventually be

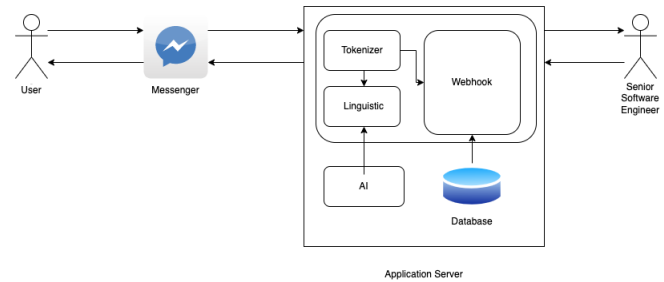


Figure 1: System Architecture

directed to other software engineer in the same team through a webhook, or to a crawler to parse specific information from the web. The overall interactions among components can be demonstrated by 1, and the user workflow can be shown by 2.

Our project will be integrated with the messenger app. Hence, we are using the Messenger app functionalities for the primary interaction between users and our system. We will require you to use Javascript to create a web hook, to scrape specific information to generate the user response. We need to use the dialog flow to read the user messages, tokenize them and then process the content. The bot can only understand the English language and can only respond in English. Hence, another constraint we have is that the user needs to use only the English language to communicate with the Bot.

We can additionally use the Behavioral Design patterns for our system. As our system is reactive in nature. It responds to user queries depending on what the user is asking. Our system can use the chain of responsibility, where the bot answers the questions if it can, else it passes the questions to the next available person. Hence, we can link the actions taken by the systems in a chain. We can also use the command and the interpreter design pattern. As our system is a chatbot, we are dealing with languages and hence we need to process different words, especially in the English language.

We can also use some of the structural patterns like the Facade or the Bridge pattern. As we have a bot which performs multiple actions, we can declare the bot instance using a single class which can represent our entire subsystem. Hence, structural patterns like Facade can be used in such scenarios. Also the structural patterns like Bridge can help us to integrate the messenger app with our system, where the user can interact with the messenger and our system will respond through the messenger.

### REFERENCES

- [1] 2021. COVID Robot Helps Keep People 6 Feet Apart During Pandemic. <https://www.webmd.com/lung/news/20211213/covid-robot-social-distancing>. Accessed: February 18, 2022.
- [2] 2022. GoToMyPC. <https://www.manageengine.com/remote-desktop-management/>. Accessed: February 18, 2022.
- [3] 2022. GoToMyPC. <https://get.gotomypc.com/>. Accessed: February 18, 2022.
- [4] 2022. Slack. <https://Slack.com/>. Accessed: February 18, 2022.
- [5] Adam S. Miner, Liliana Laranjo, and A. Baki Kocaballi. 2020. Chatbots in the fight against the COVID-19 pandemic. *NPJ Digital Medicine* 3, 65 (05 2020).

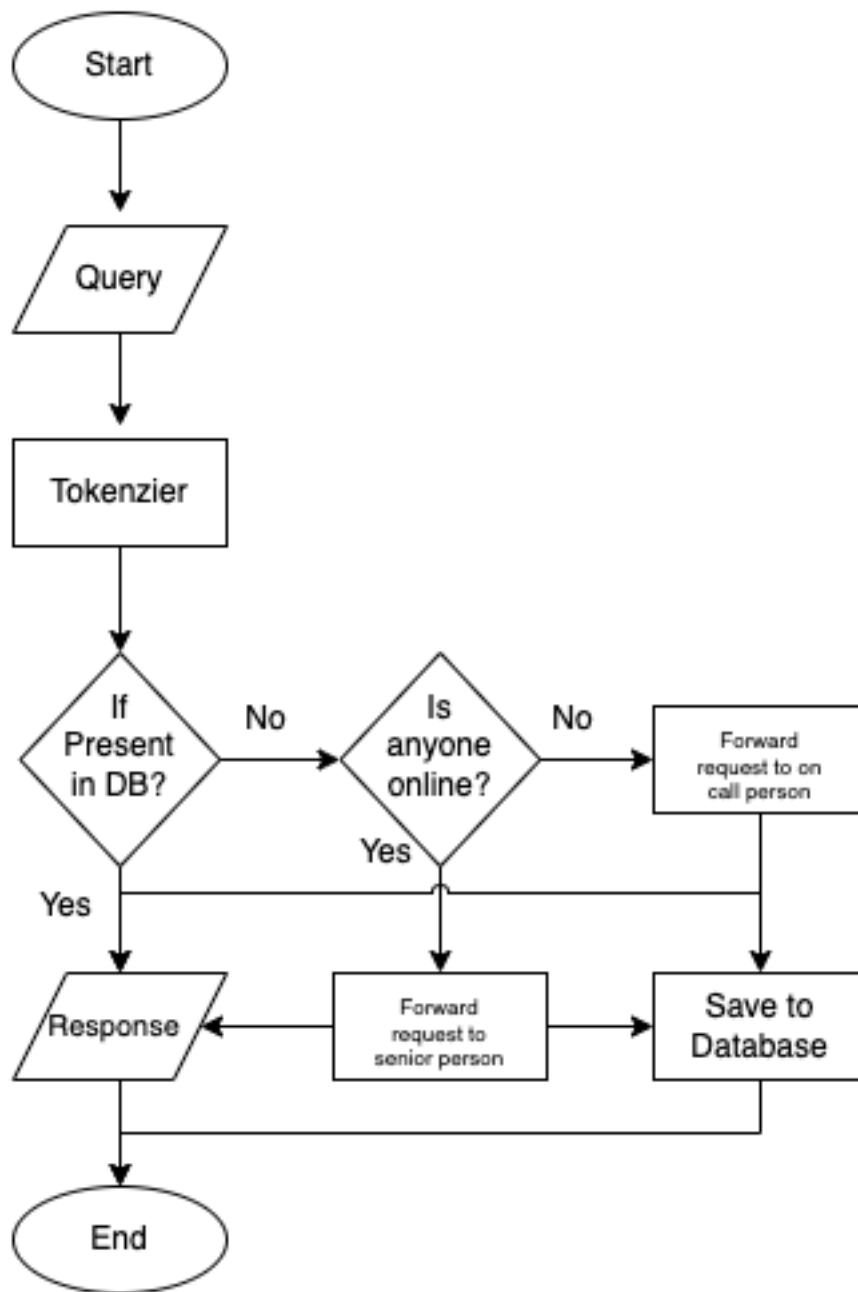


Figure 2: Flow Map