International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT228137

Study on the Development and Implementation of Ubiquitous Bots for the Discord Interface

Abhinand G, Roshni Balasubramanian
Department of Information Science and Technology, College of Engineering Guindy, Anna University, Chennai, Tamil Nadu, India

Article Info
Volume 8, Issue 1
Page Number : 212-221

Publication Issue :
January-February-2022

Article History
Accepted : 05 Feb 2022
Published :17 Feb 2022

ABSTRACT

Over the past few years, Discord's popularity has grown tremendously, expanding its target user demographic. Discord, in recent times, has broken the walls that they had previously encapsulated themselves within, and are growing to be much more than an application meant exclusively for gamers. Now, colleges, schools, offices, and groups with common interests have come together to establish communities where all parties are benefited. Discord provides a platform for developers to contribute to the community and various guilds with its publicly available Application Programming Interface (API) - called the Discord API. This paper aims to analyse bot applications, taking our own bot created for student communities, called 'Kanmani', written in Python, as an example. Development of a bot involves a multitude of steps right from creating an application on the developers' portal, writing code for the backend of the bot, ensuring maintenance of security features, and hosting it on a platform that allows the service to be always available to Discord servers. Depending on the requirements of a server and its members, an appropriate bot can be developed and added using Discord's API.

Keywords: Bot, Application Programming Interface (API), Platform as a Service (PaaS), Discord, Ping Monitoring

I. INTRODUCTION

Discord is an application, founded in 2015, that allows users to connect virtually, regardless of operating system and application type; the only requirement is to have an internet connection and use the Discord app from any platform- webapp, desktop app, mobile app etc.

The application provides features like text channel communication, audio calls, video calls and recently even started to support playing games on the interface. These features are available on both one-to-one personal conversations, and many-to-many group conversations via guilds. The terms 'servers' and 'guilds' can be used interchangeably and refer to a collection of channels (text and voice/video) for a group of people to communicate through.

Copyright: © the author(s), publisher and licensee Technoscience Academy. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License, which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited

Discord continues to improve its features, efficiency and business model and is only set to grow into a larger entity in the future. Discord provides features for integrate external bots (to the applications' guild/server) using libraries for each programming language [1].

Apart from the users' experience, Discord also provides a portal for developers to innovate using their API. Using this API, a variety of applications can be created according to the requirements of a community, and it can be added to an unlimited number of guilds with the requirement of undergoing a verification process before being available to more than one hundred guilds.

Generally, with a variety of people making use of an application, there comes the need for technological intervention to perform or automate common features and tasks, moderation, and for community engagement.

This is where bots and webhooks come into play. Instead of Discord applying the features an individual requires for their personal server throughout Discords' framework, a bot can be coded by a developer and included in each server catered to the individual's needs. Apart from this, several Discord bot marketplaces online can allow a user to add any bot that matches their requirements, securely to their server with a click of a button after granting the bot permissions and access to their guild.

Webhooks can be created to automate messages and API requests for certain functions. To take an example, a server created to learn about cloud computing can add webhooks to fetch recent tweets, articles or videos pertaining to the subject. This automation creates ease for a community to learn together and get information rather than relying on possibly incorrect information.

II. BOT APPLICATION CASE STUDY

The Discord bot we created and personified as 'Kanmani' is meant for students to ease into the process of studying. After the pandemic, as more students opt for online learning via Discord [2][3], it was evident that there was a requirement for a tool to help in productivity along with wellness parameters. This bot has been created with the help of the Discord API using discord.py library for Python [4], and other third-party APIs, and has additionally been honoured by Discord as a verified bot as of 9th February 2022.

Looking into the bot, the reasons aiding towards the success includes the following: understanding the student requirement and pain points followed by the bot's enhanced commands' ease of use (PEOU & PU) [5], following security parameters, creating tools for moderators, and finally, creating features that enhance community engagement and responses.

'Kanmani' has become popular amongst various student servers and study groups, tallying up to over 80 communities.

To support the claims made on Discord's popularity, Fig. 1 plots the number of registered users on Discord exceeded 300 million in 2019 as did the company's funding - which reached nearly $380 million in 2020 [6] [7]. Discord has over 150 million active users a month, further strengthening the need for technological arbitration for moderation, utilities, fetching relevant information, datetime tools etc.

Discord's popularity can also be attributed to the innovative approach of allowing developers to add their own bots onto the interface, creating a one-of-a-kind experience for all parties. Bots, webhooks, a variety of channels and permissions can all be factors to why the application has emerged successful rapidly within a short time frame.

This also gives scope for improvement for the company's products which could yield even more possibilities for the future of innovative additions to servers and communities.

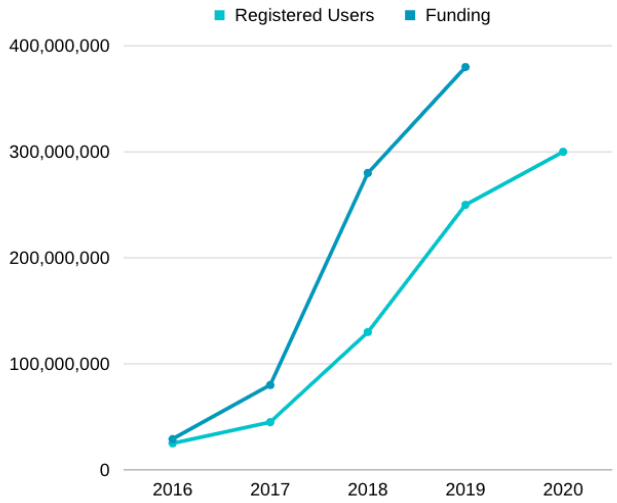

Figure 1. Popularity Surge in the past 5 years

The following sections provide an overview of what sets Discord's bots and webhooks apart from most other social media interfaces and how these integrations benefit the user in terms of functional and non-functional aspects.

III. LITERATURE REVIEW

IIgnacio Nunez Norambuena and Alexandre Bergel [8] researched on the application of Artificial Intelligence and Natural Language Processing on discord bots and developed a bot that provides a ranked list of Discord users that are experts in a particular set of topics using word occurrence table and word embeddings. Kruglyk, Vladyslav & Bukreyev, Dmitriy & Chornyi, Pavlo & Kupchak, Evgeniy & Sender, Andrey [9] researched on the possibilities of using Discord as an effective online environment during emergencies.

Duvvuri, Venkata & Guan, Qihan & Daddala, Swetha & Harris, Mitchel & Kaushik, Sudhakar [10] developed an AI chatbot which could predict depression symptoms from conversations that happened from a Discord chat with a 73% accuracy in predicting seven key symptoms for depression. Cerezo, Jhonny & Kubelka, Juraj & Robbes, Romain & Bergel, Alexandre [11] built an expert recommender chatbot that sends responses to the user using Sentence Classification Techniques and Term Frequencies.

Tkachenko, Oleksandr & Shevchenko, Andrii [12] researched and analysed the general problems that could impact a ubiquitous Discord bot, which in particular would search for music on YouTube and view the weather.

Wright, Devin & Severance, Timothy & Knutson, Charles & Krein, Jonathan & Buchanan, Tyler [13] studied and developed an autonomous Discord bot that could help improve online course experience, particularly due to the impact of the COVID-19 pandemic which affected all parts of the world mercilessly.

Amit Patil, K Marimuthu, Nagaraja Rao A and R Niranchana [14], in their 'Comparative Study of Cloud Platforms to develop a chatbot', talk about the new era of technology and the need for interactive conversing between chatbots and humans. They talk about various platforms to deploy chatbots and compare the efficiency and suitability amongst them.

Muhmmad Ashfaq, Jiang Yun, Shubin Yu & Sandra Maria Correia Loureiroc [5] discuss the factors of user satisfaction when dealing with AI powered service agents, an example being chatbots. They discuss perceived ease of use (PEOU) and perceived usefulness (PU).
These are important factors to take into consideration when conducting

requirement analysis and creating the final product, in this case, a Discord bot

IV. REQUIREMENT ANALYSIS

The first step prior to creating a bot is to identify the Functional and Non-Functional Requirements relating to the bot. The primary function of a bot would be to automate certain tasks which are usually performed by users and end up taking a lot of time and energy. Discord bots can be categorised into many sects like Utility/Administration bots, Entertainment bots, etc.

TABLE I. REQUIREMENT ANALYSIS

Functional Requirements for a Utility Bot	Functional Requirements for Kanmani	Non-Functional Requirements
The bot must provide a basic help command which lists out all the possible commands which could be used by the people in the server	The bot must allow batching study sessions by following the efficient productivity technique – Pomodoro Method [11]	The bot must have portability such that it can be used in any kind of discord server. At the same time the bot must be user friendly to ensure that it can be used easily by someone who just added it to their server
The bot must handle administration tasks like allocating permissions via roles to new members of a server	The bot must set reminders whenever the corresponding command has been given	The security and safety of the bot must not be compromised. Utmost care must be taken to ensure that only administrators and people with special roles in the server are allowed to access these commands. Personal information relating to the clients in the server/guild like user IDs, guild IDs shouldn't be compromised, and such data shouldn't be manipulated at any cost. Only necessary information should be sent and received from the bot's backend.
The bot must have functionality to let admins remove/ban users from the server via commands and/or automation	The bot must provide health reminders every 2 hours when the corresponding command has been given	
The bot must have functionality to manage messages in channels, i.e., send messages and delete other messages for spam control	The bot must provide a functionality where it can let users anonymously post messages in the channel, which could be used as a form of venting to get rid of emotional distress	
The bot must be relevant to the server; for example, there is no requirement for a stock trading stats bot in a study server meant for middle schoolers.	The bot must provide memes and study tips to the user when the corresponding command has been given	

V. USAGE OF APPLICATION PROGRAMMING INTERFACES

The Discord API is primarily based around 2 layers, a REST API that works around HTTP requests, and a WebSocket connection which is used for real-time events [1]. A Web Socket is a new feature of HTML5, which enables web pages to use the Web Socket protocol for full-duplex communication with a remote host. It introduces the Web Socket interface and defines a full-duplex communication channel that operates through a single socket over the Web [17]. In order to authenticate a bot on Discord, the API either allows the developer to use a bot token or use an OAuth2 bearer token using the OAuth2 API. A simple Bot Token Authorisation Header looks like

Authorization: Bot MTk4NjIyNDgzNDcxOTI1MjQ4.Cl2FMQ.ZnCjm1XVW7vRze4b7Cq4se7kKWs

Discord's API also has several mechanisms that prevent proper usage. Its 'Rates Limit' feature is an access control mechanism that ensures the resources are not abused or overused by excessive spamming by any server. The HTTP methods often share the same rate limit regardless of method type. For example, GET, POST and DELETE can share the same rate limit if the route for API access is the same. Discord, as of 2022, has set a request ceiling of 50 requests per second to the API. The example code snippet has been displayed below for users exceeding the rate limit [1]

```
<HTTP/1.1 429 TOO MANY REQUESTS
<Content-Type: application/json
<Retry-After: 65
<X-RateLimit-Limit: 10
<X-RateLimit-Remaining: 0
<X-RateLimit-Reset: 1470173023.123
<X-RateLimit-Reset-After: 64.57
<X-RateLimit-Bucket: abcd1234
<X-RateLimit-Scope: user
{
  "message": "You are being rate limited",
  "retry_after": 64.57,
  "global: false
}
```

Whether Discord.py is used or Discord.js, either allow the developers to incorporate third-party APIs to provide functionalities to the bot.

In case of Kanmani, the Reddit API has been made use of to fetch memes, study tips and motivation tips to the users who have triggered the corresponding commands. Table II illustrates the GET requests which were made to different subreddits to retrieve different information

TABLE II
REDDIT API REQUESTS CREATED BY KANMANI

HTTP Request	Sub-Reddit	Description
GET /api/v1/me/learnprogramming	r/learnprogramming	The bot gives programming tips to the user
GET /api/v1/me/TodayILearned	r/TodayILearned	The bot gives random facts to the user
GET /api/v1/me/MotivationalPics	r/MotivationalPics	The bot gives motivational pictures/quotes /stories to the user
GET /api/v1/me/['Memes', 'dankmemes', 'funny']	r/Memes, r/dankmemes, r/funny	The bot selects a meme randomly from the sub-reddits specified and sends it to the user in the respective channel

The response consists of a breakdown of the subreddits in JSON format.

VI. ESTABLISHING CONNECTION: BOT AND API

Bots can be created and deployed by writing code in our preferred language, since a plethora of options have been provided. The process of creating the bots and deployment follow a sequential order starting from the choice of interface to write code till choosing the cloud platform on which we host it or using alternative hosting options as discussed in the following sections.

This business decision is only an important factor if the bot is being built for a public server where users require the bot's availability round the clock. In case of adding the bot for personal use on one's own guild, the bot's code can be written locally on the developers' machine.

.

After setting up the coding environment and installing the necessary dependencies required for our particular bot application, the next step is to register the bot with its name on the Developers' Portal.

Upon successful registration, a secret Bot Token will be revealed and is used to authorise and identify the bot. Each bot requires a `Client` object to be created in order to establish a connection to the Discord WebSocket and API [4]. After this, any function can be written aligned with the syntax and terminology provided in the documentation to enable bot-server interaction.

VII. WORKING OF BOT APPLICATION

The bots listen for a specified command prefix, in our example of Kanmani "!!". Once this sequence of symbol(s) is inputted on a server on which the bot application is present and is granted permission to read the messages, the bot is triggered and identifies which response to give from the functions written. If the command is not found (or improperly typed), the console is triggered with an error message:

Ignoring exception in command None:
discord.ext.commands.errors.CommandNotFound:
Command "{text entered }" is not found

If the command is identified, the required output is pushed to the guild. At times, there is an external third-party API call made to fetch some information.

For example, if the user would like coding tips and have imputed a command for the same- the API call is made and fetches the suitable information. This information is then packaged and formatted according to the code and sent to the server.

As discussed earlier, bot applications can be hosted on the cloud. In order to keep the bot up and running nonstop, a ping monitor tool [18] can be used.

This can be implemented by establishing a webservice with the code interface and the ping monitor sets the HTTP(S) monitor type for the connection periodically to keep the interface alive. This serves a dual purpose of pinging the cloud console to inform the system that the connection should not terminate, and to notify the developer if there is an error and the console has ceased the running of the code.

This method can be used if the bot is being created for personal use and is not required to always be listening for commands. This bot can then be integrated on personal guilds, where external users are not expecting the bot to be available/online for fetching responses for their requests.

Fig. 2 displays the overall workflow from user to end system for the programmer designing the bot application for server(s).

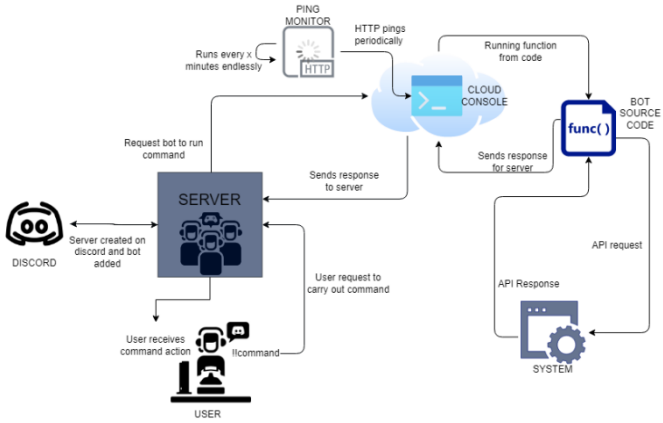

Figure 2. Workflow of the Bot Application (User to Backend)

VIII. CLOUD-BASED HOSTING

In the past decade, cloud-based services (Platform as a Service or PaaS) have increased in number and popularity. They provide platforms where chatbots can be created, developed, and deployed. There are some platforms that supplement these features with additional features such as built-in artificial intelligence [14].

Platforms such as Heroku (Salesforce subsidiary), Google Cloud Platform, Microsoft Azure and Amazon Web Services, provide the aforementioned services [19]. According to a bots' features and its requirements, a platform can be chosen.

Taking an example of the Google Cloud Platform on top of which a Discord bot can be created, the platform requires a Virtual Machine set up as part of the Google Compute Engine [20]. Choosing the required instance on the platform and then running it ensures lower overhead along with a continually running instance that is always listening to the key command prefix from the Server on the user interface. This step is preceded with the connection to Google Cloud API with its key in our code where the functions of the bot are written, meaning that the bot has been successfully hosted and is ready to use by the members of servers on which the bot is present.

Another example is hosting the Discord bot on the EC2 instance of AWS [21]. This requires creation of an Ubuntu virtual machine, similar to the step from Google Cloud Platform. The key/pair is then generated after launch and can be downloaded. The instance has to be launched and then connected to the virtual machine created with the instance. The bot can then be run on the cloud platform and monitored via other services that AWS provides along with this.

The above two examples have been provided to understand how cloud-based services can negate the necessity of ping monitors and cloud consoles. Using sophisticated solutions and cloud platforms can increase security and lower overhead. Although alternate solutions are available to use and are free of cost, they require frequent rerunning and cannot accept more than a few requests at the same time, which often leads to stopping the program which then requires manual reboot of the console. They also require ping monitors from external tools to be integrated to ensure the console does not stop running the code.

Hence, cloud platforms with existing solutions are more desirable to run chatbots as they provide a variety of features that can be integrated as opposed to running the bot application locally on a developer's machine.

IX. VERIFICATION AND USAGE

Bot applications further extend security by only being added to servers and communities by the administrator of the server(s). This ensures that malicious intent bots are not added, which reduces the chances of spamming and user data/messages to be accessed by an external machine. Additionally, before adding an application, the permissions [15] for each feature must be granted- voice channel access, manage messages permission, view messages, etc.

The developer's portal and the developers' community servers are resources for correct coding practises and additional support for creating tools within the platform. The developer's portal categorises support parameters, making it extremely easy and efficient for access for developers, showing that the user experience for even developers is enhanced and not only the end users.

Once reaching seventy-five servers on the Discord platform, the bot applications are eligible for verification. The study bot we created has been verified by Discord, labelling it as one of the bots verified by Discord themselves.

This shows that the bot follows the functional and non-functional requirements, especially the efficiency of code and security features apart from its innovative features. As a bot application increases in popularity because of its unique features and abilities, it is advised to split portions of the functionalities into different logical processes by means of 'sharding' [1].

X. FUTURE ENHANCEMENTS

Upon working on the study bot Kanmani, it has become evident that end users would like an all-inclusive kit for studying. The features we have included so far are fetching information, tips, quotes from third party APIs, timers, reminders, countdown feature, pomodoro technique [16], venting feature etc. Learning from the feedback provided by the users, it is understood that music features would be a suitable addition, to play melodies that will ease server members into work mode.

Apart from this, migrating to slash commands for enhanced usability on the Discord interface is another additional improvement that can be made soon. The bot could also apply the usage of Natural Language Processing and Machine Learning techniques in order to make communication with users even better.

XI. CONCLUSION

Interfaces that allow adding bot applications and tools for automation encourage innovative implementations. Not only does this encourage developers to experiment with the various APIs and concepts, it also exposes developers to a variety of concepts, from computer networks, cloud computing to web services and security features which further strengthens the concepts of project based learning.
The Discord interface has encouraged thousands of developers to create innovative bot applications that can be added to millions of servers, automating simple tasks like scheduling events to setting reminders.

Keeping the end users in mind, applications must provide several components such as Discord's Bots and Webhooks, that allow us to add on certain features in a safe and secure manner that are required for that individual rather than installing component heavy applications with features that are unnecessary for most users.

XII. REFERENCES

[1]. "API docs for bots and developers," Discord Developer Portal. [Online]. Available: https://discord.com/developers/docs. [Accessed: 16-Feb-2022].

[2]. M. L. Arifianto and I. F. Izzudin, "Students' acceptance of discord as an alternative online learning media," International Journal of Emerging Technologies in Learning (iJET), vol. 16, no. 20, p. 179, 2021.

[3]. Discord to Support Synchronous Communication in Distance Learning, Advances in Social Science, Education and Humanities Research, Volume 560 Proceedings

of the 2nd Annual Conference on Blended Learning, Educational Technology and Innovation (ACBLETI 2020)

[4]. "API reference," API Reference. [Online]. Available: https://discordpy.readthedocs.io/en/stable/api.html. [Accessed: 16-Feb-2022].

[5]. M. Ashfaq, J. Yun, S. Yu, and S. M. Loureiro, "I, chatbot: Modeling the determinants of users' satisfaction and continuance intention of AI-Powered Service Agents," Telematics and Informatics, vol. 54, p. 101473, 2020.

[6]. C. Coberly, "Discord has surpassed 250 million registered users," TechSpot, 13-May-2019. [Online]. Available: https://www.techspot.com/news/80064-discord-has-surpassed-250-million-registered-users.html. [Accessed: 16-Feb-2022].

[7]. "Discord - funding, financials, valuation & investors," Crunchbase. [Online]. Available: https://www.crunchbase.com/organization/discord/company_financials. [Accessed: 16-Feb-2022].

[8]. N. Norambuena and A. Bergel, "Building a bot for automatic expert retrieval on discord," Proceedings of the 5th International Workshop on Machine Learning Techniques for Software Quality Evolution, 2021.

[9]. V. Kruglyk, D. Bukreiev, P. Chornyi, E. Kupchak, and A. Sender, "Discord platform as an online learning environment for emergencies," Ukrainian Journal of Educational Studies and Information Technology, vol. 8, no. 2, pp. 13–28, 2020.

[10]. Duvvuri, Venkata & Guan, Qihan & Daddala, Swetha & Harris, Mitchel & Kaushik, Sudhakar. (2021). Predicting Depression Symptoms from Discord Chat Messaging Using AI Medical Chatbots.

[11]. J. Cerezo, J. Kubelka, R. Robbes, and A. Bergel, "Building an expert recommender chatbot," 2019 IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE), 2019.

[12]. O. Tkachenko and A. Shevchenko, "Some aspects of developing a universal server discord-bot," Digital Platform: Information Technologies in Sociocultural Sphere, vol. 4, no. 2, pp. 173–186, 2021.

[13]. D. Wright, T. Severance, C. Knutson, J. Krein, and T. Buchanan, "An autonomous discord bot to improve online course experience and engagement: Lessons learned amid the covid-19 pandemic," Proceedings of the Annual Hawaii International Conference on System Sciences, 2022.

[14]. A. Patil, M. K, N. R. A, and N. R, "Comparative study of cloud platforms to develop a chatbot," International Journal of Engineering & Technology, vol. 6, no. 3, p. 57, 2017.

[15]. A. Verma, S. Tyagi, and G. Mathur, "A comprehensive review on bot - discord bot," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, pp. 532–536, 2021.

[16]. Using the Pomodoro Technique® to help undergraduate students better manage technology-based multitasking during independent study: A design-based research investigation Salman Ahmed Usman, BSc (Hons), MSc, PGCert (LTHE) April 2020

[17]. Wikimedia Foundation. (2022, February 3). Websocket. Wikipedia. Retrieved February 16, 2022, from https://en.wikipedia.org/wiki/WebSocket

[18]. J. Dhillipan, N. Vijayalakshmi, and S. Suriya, "Network Monitoring System using PING methodology and Gui," Intelligent Systems Reference Library, pp. 13–22, 2019.

[19]. R. Z. Khan and M. F. ALi, "A Study of Cloud Computing," International Research Journal of Computer Science (IRJCS) , vol. 2, no. 5, May 2015.

[20]. "Build and run a discord bot on top of google cloud | google cloud blog," Google. [Online]. Available: https://cloud.google.com/blog/topics/developers-practitioners/build-and-run-discord-bot-top-google-cloud. [Accessed: 16-Feb-2022].

[21]. R. Pasupathy, "Building & Hosting a discord bot on AWS," Medium, 10-Sep-2021. [Online]. Available: https://towardsaws.com/building-hosting-a-discord-bot-on-aws-e157bd7faf78. [Accessed: 16-Feb-2022].

Cite this article as :

Abhinand G, Roshni Balasubramanian, "Study on the Development and Implementation of Ubiquitous Bots for the Discord Interface", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 8 Issue 1, pp. 212-221, January-February 2022. Available at doi : https://doi.org/10.32628/CSEIT228137
Journal URL : https://ijsrcseit.com/CSEIT228137