

Instant Messenger Moderator Bot

Team: The Voltron Brigade Tristen F., Cooper T., and Hudson D.

SECTION 1: INTRODUCTION

We are building an automatic moderator bot for Discord, a VOIP/ instant messenger application. As daily users of Discord, our group wants to create something that we believe could be improved upon. We have handled discord bots from a users perspective, and we know many shortcomings that exist with current bots. Some bots have limited functionality, while others have not been well maintained and are incompatible with the current version of Discord. Our bot will be unique in its variety of features, as well as the quality of their implementation.

SECTION 2: CUSTOMER VALUE

Many users use Discord as a VOIP program for communicating in games, however users still use the platform for social interactions and “professional” environment servers used by various internet and real life communities such as The University of Tennessee, a person's friend group, work offices etc. An underlying problem is understaffed servers, when admins / staff go offline but users are still using the server, if malicious people/bots get invited to the discord they could spam the server with images/ voice or phishing links. Another common problem is giving out too much administration roles to users. This can be bad if the new administrator/staff turns out to use the elevated privileges for malicious intent such as deleting channels/banning/kicking users unfairly. The primary customer for our project is anybody who is running a discord server who is need of a moderator but does not want to moderate themselves or give the administration privileges to someone. Secondary customers include the members of the discord. The customer wants a bot who can automatically kick/ mute/ ban malicious people who join, and assign roles based on the amount of time since the user joined the server. The bot will also have Quality of life (QOL) implementations such as a translation, wiki lookup, music player, and a few text based games all within the interface. The users' experiences will be more streamlined when using our bot within their server.

SECTION 3: PROPOSED SOLUTION

The solution will deliver a seamless, lightweight moderator with a multiple of quality of life functions that can be used by the owner or regular members of the discord. The customer (owner / admin of the discord) doesn't need be present to administrate. The regular users of the server (non administrators) can have some interactive experiences with the bot and it's quality of life functions such as translate, wiki lookup, listening to music together etc. If the proposed solution works well, the bot will be seamless and will assist the customer in moderating their channel(s) and entertaining its members. The concept of a Discord bot is not new at all, however we aim to consolidate the concepts of a few bots that exist into one user friendly bot. Plus, this will be functional on the newest Discord version. Many bots have ceased development as their creators moved on to other projects.

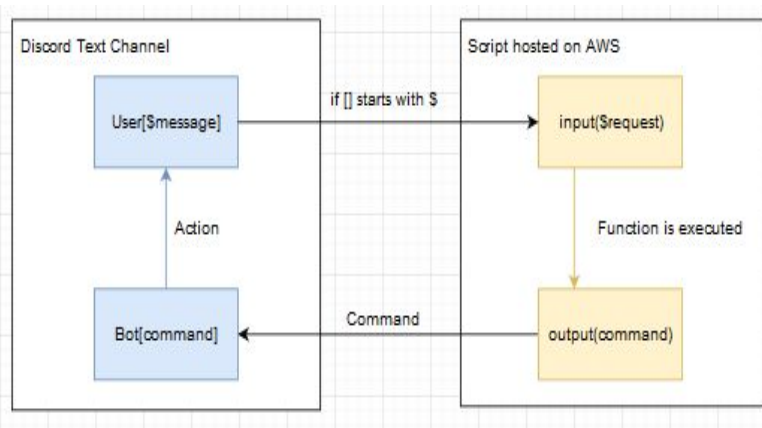
We have spoken with a single owner of a discord server and they seem to like the idea of a bot that can be used in their absence to remove malicious people and thinks their users would benefit from its features. The solution has been met when their server is free of spammers, people attempting to phish with links, and users who bully / are not welcome to the discord can be removed automatically, all without the need of the customer needing to be online for it to happen.

SECTION 4: TECHNOLOGY

This bot will first and foremost be able to moderate a server on Discord, with the ability to mute abusive chat, ban or kick abusive players, and even have adjustable settings as to when to perform these moderator actions. The main components of the software will be able to staff a server as effectively as a human moderator, 24 hours a day.

The script will be written entirely in Python, using discord.py (The discord API, very well documented) and will be hosted on an AWS EC2 instance. This is because the script is the brains of the bot, the bot simply is the “user” element inside the discord server that reads and sends text messages/commands. The

script itself being hosted elsewhere takes in the read input, parses it to ensure that it is a command and not just a random message being sent. We will use the \$ sign to delimit commands for example: *\$voteban <username>* will trigger a vote to ban a certain user by all online users at the time. The script will understand the first character \$ means a command, and send the next string into the script as input and output the proper command to the bot to take effect. During development, the system will be tested in a temporary server on the users localhost to ensure that each function is properly working.



SECTION 5: TEAM:

No one on the team has made an automated bot before, however the team has a member with some Python experience. The tools for developing the bot are new to us (such as libraries, Python, Pycharm IDE etc) however we have a good idea of architecture and implementation for the project alongside experience with Amazon Web Services. There are 2 main roles for the features we plan to implement. Implementing the moderator functions being the largest feature will require all the members actively working on it. While a lot of the smaller quality of life functions will require 1-2 members to work on them at a time. Team management roles will be the shared responsibility of the entire group. Making sure to assign planned features and milestones to themselves. Due to the modular nature of our project, roles can be rotated as features are implemented.

SECTION 6: PROJECT MANAGEMENT:

We believe this project is feasible, the basic features of moderation are implementable in our given timeframe. However the extra features will be created if we have the time afterwards to make. Meetings can be done weekly to discuss our progress for that week. There are no constraints along the lines of regulations or legality, as Discord fully supports bots and supplies great documentation as long as you register the account as a bot and not a human user. There are no social or ethical concerns entailed in this project, in fact this hopes to alleviate any social stress caused by users who do not want to give out administration to their friends they do not trust. As for any data we require, discord.py is well documented and Python has a lot of resources online. Amazon Web Services has a lot of tools to read and to assist us. If the secondary QOL features are not met, the bot can still fully function as a moderator in the server. However, given the timeline below it is not likely.

Week I and II

- After the first week or two we plan to have a blank bot that is capable of reading in commands and repeating them back to us to get I/O between the script on AWS and Discord working.

Week III and IV

- The third week starts finishing up a few of the basic moderating features such as kicking spammers, banning repeat offenders, adding/ promoting users to various ranks within the server

Week V and VI

- The fifth week we want to start having a fully fleshed out moderator that is capable of moderating a server by itself and move to making the quality of life functions on the bot itself.

Week VII and VIII

- Starting at week 7 we want to start finishing the quality of life features, and perhaps move in to text-based user interactive games such as a MUD or hangman.

Week IX and X

- Spend the last weeks of class polishing and attempting to break our functions. We will fix and limit as much user input exploitation that may occur.