



Wild Card Bot

Team 25 Product Backlog

Konstantin Liehr, Arun Thirumavalavan, Nikolas Damalas, Kyle Arrowood, Michael Gu, and Matthew Wang

Problem Statement:

Although there are many Discord bots out there, most only serve a single purpose or function in a Discord server. This leads to some servers having a swarm of various bots for all the different functions they need, mucking up the user lists with bots and requiring users to know the specific syntax and setup for each bot. Wild Card Bot will solve this problem by being a modular bot with the ability to add new functions made by anyone quickly and easily. It can moderate your server, play music, or anything you want it to if you make the right function.

Background Information:

Audience:

Hundreds of millions of people use Discord for various communication purposes everyday. With the pandemic, Discord has become even more useful as some meetings that were once in person are now being held over Discord. Despite the huge amount of people who use Discord, there are no Discord bots with a complete set of server moderation features that allow users to add functionalities.

Similar Applications:

- Nugget bot: multipurpose moderation bot
- Rhythm/GroovyBot: Plays music
- MEE6: Sets up auto-roles and has custom commands
- YAGPDB.xyz: General purpose bot. Assigns roles, moderates, server stats,

Limitations:

The limitations of other similar Discord bots is their restricted functionality and lack of customization. Many Discord bots either play music, assign roles, or allow for custom commands. However, there are few bots that are capable of doing all the essential functionalities necessary for the automation and assistance of a Discord server.

Similarly, most Discord bots are not customizable. Moreover, there is no easy way for developers to add additional functionality or commands. The only Discord bot from the Similar Applications section above that features custom commands is MEE6. However, MEE6 restricts custom commands to only manipulate or create roles, and the user must download the custom commands

plugin to have access to this feature. This results in very minimal additional functionality the user can achieve and an external download that complicates customization.

Wild Card Bot tackles these problems by allowing for easy customization with the modular function system and including a complete set of functionalities the average Discord server requires.

Functional Requirements:

1. Modular Function System:

- a. As a Server Administrator, I want to add functions to the bot using Discord's file sharing feature.
- b. As a Server Administrator, I want to remove functions from the bot with a command.
- c. As a Server Administrator, I want to add and remove functions to the bot via a function folder.
- d. As a developer, I want to dynamically load functions and features to the bot from separate files.
- e. As a developer, I want functions to be able to use libraries not found in the main bot.
- f. As a developer, I want to know what format to code commands in.
- g. As a developer, I want to know what languages I can code commands in.
- h. As a Server Administrator, I want to know if the function I am adding uses the same command as a currently implemented command.
- i. As a Server Administrator, I want to specify and change which roles can use a command.
- j. As a Server Administrator, I want to know where I can download additional modules.

2. Administration Services:

- a. As a Server Administrator, I want the bot to automatically delete messages containing banned words
- b. As a Server Administrator, I want the bot to warn users if they try to use a banned word.
- c. As a Server Administrator, I want to be able to edit the list of banned words.
- d. As a Server Administrator, I want the bot to kick/ban users with a command.
- e. As a Server Administrator, I want to create text and voice channels with a command.
- f. As a Server Administrator, I want to assign and change user roles with a command.
- g. As a Server Administrator, I want the bot to have the ability to remove messages

with links in certain channels.

- h. As a Server Administrator, I want to change the permissions of a role or channel via a command.
3. Basic Bot Usage:
 - a. As a user, I want to know what commands are available to use with the bot.
 - b. As a user, I want to know what a command does and ways to invoke it.
 4. Music Function:
 - a. As a user, I want to play songs from youtube by providing a youtube link.
 - b. As a user, I want to queue multiple songs to be played.
 - c. As a user, I want to add and remove songs from the queue.
 - d. As a user, I want to view the queue.
 - e. As a user, I want to clear the queue.
 - f. As a user, I want to skip a song.
 - g. As a user, I want to have a private channel where I can listen to music without the interference of other users.
 5. Decision Maker:
 - a. As a user, I want to have the bot randomly pick between a set of options to help me make decisions.
 - b. As a user, I want to have the bot to randomly generate a number between a range of values.
 6. Poll Maker:
 - a. As a user, I want to have the bot generate a poll that members of the server can vote on.
 - b. As a user, I want the bot to announce the result where the poll was created with voting statistics.
 7. Google Search:
 - a. As a user, I want to get a google search result in the server with a command.
 - b. As a user, I want there to be an option to jump to the search in my browser from an extra command.
 - c. As a user, I want there to be an option for the bot to give me a short answer rather than extra unneeded information.
 8. Basic Financial Info:
 - a. As a user, I want to see basic stock information when I enter a ticker in chat
 - b. As a user, I want to see expanded stock information.
 - c. As a user, I want to get a link with more info about the stock.
 9. Voice Interaction: (If time allows)
 - a. As a user, I want to speak to the bot telling it to do different commands.
 - b. As a user, I want the bot to talk back to me. (Optional feature)
 10. Game statistics display (Riot and Steam (If time allows)):
 - a. As a user, I want to display my overall game statistics for a game of choice.

- b. As a user, I want to actively know what games my friends play.
 - c. As a user, I want to be able to see when me or my friends have sales on their wishlist.
 - d. As a user, I want to know what games me and my friends have in common.
 - e. As a user, I want the bot to share my game statistics with others on the server.
11. Riot API Integration (League of Legends)
- a. As a user, I want to display information for the game I'm currently in.
 - b. As a user, I want to get information on a specific champion.
 - c. As a user, I want to get information on a specific account.
 - d. As a user, I want to see my recent match history.
 - e. As a user, I want to see my champion mastery score.
 - f. As a user, I want to see information about an in-game item.
 - g. As a user, I want to see what pro players build on my champion.
 - h. As a user, I want to see what jungle pathing is the best.
 - i. As a user, I want to see the matchup rates of my laners to know which lanes to focus and help.
 - j. As a user, I want to see my teams individual win rates to know who can carry and which players to enable.
 - k. As a user I want to see tips for laning on how to manage minions in all forms: slow pushing, hard pushing, freeze and what to do in each situation.
 - l. As a user, I want to see if there are any ongoing in-game events or special modes.
12. Code Manual Documents (If time allows: Java, Python, C):
- a. As a user, I want to display code documentation.
 - b. As a user, I want to be able to easily access the code documentation.
13. Clock Integration:
- a. As a user, I want to be able to run a stopwatch on the server.
 - b. As a user, I want to be able to set a timer for a specific length of time on the server.
 - c. As a user, I want the timer to have an alarm when it is done.
 - d. As a user, I want to type a command to give a timezone in a certain city.

Non-Functional Requirements:

Architecture:

We plan to develop the modular function system and utilize it as a platform to develop the other crucial or bonus features. Once we have finished this foundational platform, we can easily divide our work into different functions and commands.

Most of our development will be using Python. We will also be using the Discord API to enable our bot to interact with server components such as text and voice channels. Bonus features will also utilize other APIs including the Riot Games API and Steamworks API, obtaining statistics

and other desired data.

Performance:

The bot should respond to the user within 1 second of a user requesting something from it. Since there will be a wide variety of different actions that the bot can do, this time may vary depending on the complexity of the task at hand, but in general nothing should take much longer than a couple of a seconds from when something is requested of the bot. The bot will be available for use 24 hours a day.

Usability:

For the server administrator, the bot should be easy to download, set-up, and customize to meet the needs of the server. The set-up and customization process should be easy enough such that someone who has no experience with Discord bots should be able to set-up the bot and customize it without any extra assistance. The general user should never have to interact with code.

For the user, the bot should be easy to interact with. Each user should be able to easily use the bot to perform commands given that it is allowed by the bot depending on their role in the server. For example an average user will not be able to do all of the same commands that a server administrator would.

Security:

Only server administrators should have access to the administration commands and be able to add and remove functions from the bot.

This means it is also the server administrator's choice in whether or not malicious code can be used by the bot as they have control over which files it runs. We are in no way responsible or accountable for what the server administrator chooses the bot to do as we have designed the bot as a platform to where users, with the server administrator's permission, can implement their own features.