

Reword - a Reward Bot

MIGUEL LOPEZ, Virginia Tech, United States

MAKSIM KIRILLOV, Virginia Tech, United States

SHERMAN MILLER, Virginia Tech, United States

ARJUN VELLANKI, Virginia Tech, United States

LEILA MENG, Virginia Tech, United States

PUCHUAN SONG, Virginia Tech, United States

Abstract: The problem that we are trying to solve is the lack of participation from people in group projects. The bot that we are proposing to create would track each group member's participation in the work and reward them accordingly using many different ways, from positive reinforcement messages to providing visual points to the user when they participate, using which the team could reward each other accordingly.

ACM Reference Format:

Miguel Lopez, Maksim Kirillov, Sherman Miller, Arjun Vellanki, Leila Meng, and Puchuan Song. 2024. Reword - a Reward Bot. 1, 1 (May 2024), 3 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

Teachers want their students to ask questions, help each other, and reward students who participate in discussions. This may be a difficult task to manage in a large class or if they are overloaded with a lot of simultaneous tasks. Similarly, employers want their employees to support newer employees by answering questions and encouraging new hires to ask questions. The companies want to reward their teams for helping other workers, but that requires them to keep perfect track of who answers questions without any biases or uncertainties.

Keeping track of participation is an essential step for achieving this goal, but there is no motivation for why someone would participate. Humans are naturally greedy and would not do a task unless there is an inherent reward present, let it be practical or not. According to a study by Ritchell Nacional, when students are rewarded, they are more likely to participate in the activity that they are told to participate in. [2] In this specific study, the students were rewarded with virtual points that serve no point outside of the learning platform, yet it massively increased participation by students through the process that the article calls "gamification". [2] This principle applies to not only students but the human race as a whole, be it in the educational or professional environment, human beings want to be rewarded for their work and efforts, which is what our bot is trying to accomplish through a similar form of "gamification" as described in the article.

Authors' addresses: Miguel Lopez, Virginia Tech, United States, miguelpz@vt.edu; Maksim Kirillov, Virginia Tech, United States, maksim@vt.edu; Sherman Miller, Virginia Tech, United States, Millersd25@vt.edu; Arjun Vellanki, Virginia Tech, United States, arjunv905@vt.edu; Leila Meng, Virginia Tech, United States, leilame@vt.edu; Puchuan Song, Virginia Tech, United States, spuchuan19@vt.edu.

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 the author(s) 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.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

Manuscript submitted to ACM

Manuscript submitted to ACM

Computers are punctual and never make "human errors". The deterministic nature of a program could keep track of the frequency at which users respond in a messaging application without any errors or flaws a human awarder might face. A program such as our proposed bot could keep track of usernames, statistics on participation, and how helpful their response was through a voting system. Users could interact with the bot by reacting to messages to access statistics through an ever-updating leaderboard. A teacher or employer could set up this bot easily, which will result in their students or employees being more motivated to respond and ask questions. This will create an environment for active learning and team building which would not have been present otherwise.

2 RELATED WORK

One of the relevant software engineering tools is Git. Tools that use Git, such as GitHub or GitLab allow for the sharing of code and version control. These particular tools are similar to the bot that we are proposing since both will offer statistics of the usage of each team member, with the one difference being the Git-based tools focus on the coding aspect, while our bot will focus on the communication between teammates.

A research study by Verma et al. about the functionalities of a discord bot shows how having a bot added to a server adds new functionality and experiences to the users. [1] The bot addresses the problems people were having in their server. [1] The non-functional requirements that are listed are comparable to similar existing bots, but address different problems from the one that we are solving with our bot. [1]

3 SOFTWARE ENGINEERING PROCESS

A software engineering process our group plans to use is the prototyping method which is an iterative method. This method allows us to be flexible if requirements change or are not initially presented clearly enough. It also lets us have an actual prototype that we can use in the field to receive feedback from unbiased reviewers who will provide feedback on how it can be improved. Since our bot will be interacting with humans, it is imperative to test the usability and accessibility of the bot and make adjustments as necessary. Furthermore, having a viable prototype lets us experiment with the ideas of how a user would interact with the bot, and rather than leaving it as purely hypothetical speculation, we can actively test the features and the additions that can be added at a later time.

4 HIGH-LEVEL DESIGN DECISIONS

One of the main choices that we made for utilizing our bot was the usage of the Python language. This was the ideal language to use due to its lightweight and easy-to-use nature with a lot of documentation already prevalent for the creation of the discord bots. This language also specializes in speed and precision, which is a good option for scalability when the bot becomes popular enough to require high loads. The main technique that we have decided to use for developing the bot is through the use of prototyping, where after every change we create a prototype and test out the bot for potential changes and issues that needs to be addressed. This technique allowed us to swiftly accomplish our predetermined goals for our project, following the exact specifications we laid out at the beginning of the semester. This technique allowed us to swiftly identify key weaknesses in our code and easily fix them.

5 IMPLEMENTATION PROCESSES

The bot utilizes version control using git and GitHub to track changes and allowing rollbacks if necessary. The usage of these tools can be expanded to use issues to keep track of fixes or improvements regarding bugs, requested features,

patches, etc, as well as working with continuous integration tools such as GitHub Actions to pair with automatic deployment to a cloud service.

6 TESTING APPROACH

Black Box Testing is testing without focusing on internal code and only paying attention to functionality. By testing the input and output to see if the Bot provides the expected results, the code quality will be measured on how many tests fail. The input would be commands given to the Bot as well as reactions to messages, and the expected output would be compared with what the Bot returns to the user.

7 DEPLOYMENT AND MAINTENANCE

For deployment, as the tool is a simple Discord bot and not a critical tool that requires constant uptime and is a small codebase, we can utilize a Basic Deployment strategy. The tool can also be deployed in the cloud or other types of services that allow hosting from servers as that will allow for continuous uptime of the bot. For maintenance, once again as the tool has a very small codebase, we can use corrective software maintenance to fix any issues as they come up.

8 LIMITATIONS AND FUTURE WORK

Currently the tool is limited to usage within Discord servers, but the concept can be easily adapted to other platforms by using their respective APIs. The tool also currently lacks the scoring and rewarding functionalities, which are features that need to be completed as future work.

REFERENCES

- [1] Gauri Mathur Anirudh Verma, Shashikant Tyagi. 2021. A Comprehensive Review on Bot - Discord Bot. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* 7, 2 (April 2021), 1–5. <https://doi.org/10.32628/CSEIT2172100>
- [2] Ritchell B. Nacional. 2022. Gamifying Education: Enhancing Student Engagement and Motivation. *Puissant* 5 (March 2022), 716–729. https://www.researchgate.net/publication/372689745_Gamifying_Education_Enhancing_Student_Engagement_and_Motivation