

Web Apps: Using Bots to Expand the Functionality of Discord

Contents

Project Description	2
Aims and Objectives	2
Ethics	3
Work Plan	3
Week 1-2 (Feb 3 rd – Feb 16 th)	3
Week 3-4 (Feb 17 th - March 1 st).....	3
Week 5 -6 (March 2 nd – March 15 th)	3
Week 7 – 8 (March 16 th – March 29 th)	4
Week 9-10 (March 30 th – April 12 th)	4
Week 11-12 (April 13 th – May 3 rd)	4
References	4

Project Description

Discord. (2020) is an open service that anyone can use for free, it aims to replace services like Skype and Google Hangouts while introducing many features that work great for organising, collaborating and sharing. Discord has an API (Discord Developer Portal. (2020)), meaning that we're able to create bots to interface with this API and in turn add functionality to Discord. Currently, Discord alone doesn't have great support when it comes to group monitoring, it doesn't record participation levels or control role assignment of tasks for group projects. I aim to fix this, by introducing Discord bots we're able to add group monitoring, allowing for the bot to produce weekly reports on individual's contribution to the group. Moreover, a system where someone can be designated as the group supervisor and their linked group will need to be implemented, ensuring that the correct permissions are granted. Secondly, I'd like to implement a bot where people can submit questions during a lecture, then the most upvoted questions will be sent to the Lecturer for them to answer. This could either be done live during the lecture or post-lecture.

I'm implementing this bot to monitor group participation to avoid the situations where you have some doing more than others during group projects, furthermore, the bot will be able to keep a log of everything, this being useful if the group supervisor wishes to look over and confirm each individual's participation. I will aim to implement the weekly reporting function as a graph, showing who has contributed the most and least, this can be seen by anyone in the group. It may be necessary that there is some input from the Group Supervisor in the cases where people try to trick the bot by spamming or making conversation outside of the scope of the project in the Discord server.

The secondary function, submitting/upvoting questions during lectures is implemented to solve the issue where students have questions but often don't ask them. Either because they don't have the confidence to ask or think the question isn't valid. Using a Discord bot, people could submit their questions, other students can then see these questions and upvote. People won't be completely anonymous; they will have a randomly generated name/number that is related to their actual student number. Attached to this secondary function the Discord bot could post when lecture notes have been added or when the lecture recording has been made live. It would also be another method to ensure that people are aware that coursework is now live.

Overall, using Discord for more education-based activities provides a platform for students to discuss and work together, while supervisors are still able to monitor and see contribution rates of students.

Aims and Objectives

I believe most of the aims and objectives were covered in the Project Description, but I'll summarise here:

- Implement Monitoring of Group work.
- Logging of group discussions via messages.
- Simple natural language processing.
- Upvoting system for questions in lectures.
- Registration system for someone to join the Discord server, ensuring they're a verified student.
- Automated permission system
- (If spare time) Implement a notification system when Coursework, Recordings and materials are made live.

Ethics

After some discussion with my Supervisor we've agreed that an ethics proposal will have to be submitted. It will involve storing people's messages and creating reports about each student within a group. The only identifiable information that will be used by this bot is the Student number. After a deeper understanding of the Discord API I'll have a greater grasp on if the ethics proposal will need to be updated.

In the case ethics is declined the project will still be able to go ahead, we'd be able to test the bot using other bot accounts. Moreover, through the development stage while ethics approval is still pending, I'll test the bot in an isolated private server with no users other than myself.

Work Plan

Below I've created a 12-week plan, at the end of each two-week period I'll summarise what has been successful and what problems have arisen. If I'm moving ahead of time, I'll look into implanting the extra functionality for notifying students about lecture recordings, coursework and so on. Week 1 and 2 are focused around generating a core understand of the Discord API and getting a baseline bot running and ready for adding functions. Week 3 and 4 is where I start to implement the system for monitoring group participation. Week 5 and 6 is for implementing more features like logs and some simple natural language processing. Week 7 and 8 is for implementing the Lecture side of the bot and ensuring the bot can maintain stability under heavy requests while ensuring that the bot doesn't take too long to process a request. Week 9 and 10 is the period for generating a draft final report and creating a simple step-by-step of how someone would deploy the Discord bot onto a server. Week 11-12 gives a buffer period where core functionality of the bot can still be changed if needed and code can be altered if desired after a meeting with supervisor. It will also give some extra time if I've fallen behind in previous weeks.

Week 1-2 (Feb 3rd – Feb 16th)

- Understand and explore Discords (Discord Developer Portal. (2020).) API at a deep level
- Understand Discord Rules/Guidelines regarding bots.
- Submit Ethics
- Create Design of how the bot will work for each of its functions.
- Get a bot running as a continuous service ready to take requests by using Python to interface with the API.

Week 3-4 (Feb 17th- March 1st)

- Create a system where users must register with a bot before they are given access to the server.
- A backend database system to store registered users, could be done via .CSV.
- Implement a monitoring system, recording student's participation.
- Produce participation reports weekly or through user commands.

Week 5-6 (March 2nd – March 15th)

- Supervisor Meeting.
- Implement a basic level of natural language processing, where a supervisor is flagged if possible inappropriate language is used.
- Add a system to store logs of group chats which can only be accessed by group supervisors.
- Implement a permission system that is automated via the bot to ensure that Group supervisors have permissions over students.

Author: Kieran Parnell 1613910
Supervisor: Frank C Langbein

Week 7 – 8 (March 16th – March 29th)

- Lecture question upvoting system.
- Test bot for maximum requests that can be made per min. (Fix code if too low)
- Refactor previous functions to ensure understandability of code.

Week 9-10 (March 30th – April 12th)

- Creation of draft final report.
- Check code and commenting.
- Create a step-by-step of how to install the bot.

Week 11-12 (April 13th – May 3rd)

- **Meeting to discuss draft final report and if anything should be changed before deadline.**
- Finishing touches on Final Report

**Work plan includes working over Easter break.*

References

Discord Developer Portal. (2020). Discord Developer Portal — API Docs for Bots and Developers. [online] Available at: <https://discordapp.com/developers/docs/intro> [Accessed 1 Feb. 2020].

Discord. (2020). Discord — Discord - Free Voice and Text Chat. [online] Available at: <https://discordapp.com/> [Accessed 1 Feb. 2020].