A project report

on

DisClassroom Bot

Submitted By

Samuel Mathew (20BCS116)

On

21st December, 2021

Under the guidance of

Dr. Uma Sheshadri

&

Mr. Shankar Biradar

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

DHARWAD

Abstract

In the wake of the COVID-19 pandemic, educational institutions were left with no choice but to shut down and conduct their classes online. This paved the way for video conferencing and online learning platforms, some of the prominent ones being Google Classroom, Zoom, Microsoft Teams, Google Meet and even lead to the creation of new ones. However, many students have found online classes to be cumbersome and hectic. Juggling applications to keep track of assignments, notices, announcements and communicating with teachers leads to confusion and a gap in communication.

This project aims to use the popular social media networking site, Discord, to create a web application, to implement the various features of an online learning platform, all in one location: Discord. Teachers will be able to post announcements, assignments, and quizzes, and students will be able to view and submit their classwork. Teachers can then review these submissions and grade the students accordingly. The application employs the Discord, Google Sheets and Google Drive API to accomplish its tasks, and uses MySQL for its database management system.

The aim is to provide an open source, public Discord bot, to aid educational institutions in setting up their online learning platform on Discord, and providing all the necessary tools one would need to do so. With the first edition of this bot, any educational institute can introduce this bot to their private Discord server, configure the bot for their server, and use its available commands. Configuring the bot involves firstly, identifying the two roles: Teachers and Students, which will provide access to members in the server having these roles, with the respective set of commands and secondly, identifying the different channels, as needed by the bot for its functioning. Every new member joining will have to identify themselves as a Student or a Teacher. In case of the former, they will have to input their personal details for the teachers' reference and in case of the latter, they will need approval by an admin or teacher already present in the server.

Contents

List of Figures and Tables

1. Introduction

With the abrupt shift of offline classes to online classes in light of the COVID-19 pandemic, educational institutions have been looking for the best options among the available alternatives for online learning platforms, to pick the one most viable for them. This opened up a door of opportunity for developers to create new, innovative apps and websites to help schools and colleges manage their online classes. While many have settled for solutions like Google Classroom, or Microsoft Teams, students have expressed their dissatisfaction with online classes. Students are expected to keep track of multiple apps or platforms to ensure they're up-to-date with their online classes and to stay in contact with their different subject teachers. This often leads to a gap in communication and overwhelms students. With the current climate of things with the pandemic, students and teachers are already under tremendous stress and pressure, and online classes have their own long list of downsides, ranging from technical issues, demotivation in studying, lack of proper teaching tools etc.

Thus, the need remains for a one-stop solution for the issue of having to use various apps to keep track of online classes, and some of the institutes have already opted to use a social media application, that has grown exponentially in popularity in recent times, Discord.

Discord is packed with many exciting features, with over 150 million monthly users and is based around forming communities, in the form of servers, and also provides text, voice, and video chat options. It even has a user-friendly interface, and most of all, it provides developers the opportunity to interact with their API by creating bot users, which can be fully customized and loaded with any desired functionality. Bots can range from being used for moderation purposes to making game bots. There are primarily two types of bots: private and public, which is decided by the bot's owner. Private bots can only be invited to servers by its owners and is generally made for specific servers, whereas public bots can be invited by admins of a server to their server, with the help of a link. Hence, this project aims to create a public bot useful to servers of educational institutes, whose invite can be made available publicly for people to use.

2. Review of Literature

Delay in communication is one weakness of online learning that is reported by many researchers (Howland & Moore, 2002; Petride, 2002; Hara & Kling, 1999; Vonderwell, 2003). According to the study by Howland & Moore (2002), the communication between students and between students and instructor was a critical issue. The absence of face-to-face interaction between student and instructor contributed to negative perceptions of many students. Students felt unconfident in guidance when the feedback from instructor was delayed. In addition, in Howland & Moore's study (2002), they found that many students reported that it was difficult to get clarification on assignments, etc. due to lack of communication between student and instructor. The general impression of communication between students was also negative. The students often reported that the message board posting was ineffective and they were disappointed in the level and quality of communication (Howland & Moore, 2002). Though mediums of online communication have significantly improved since then, the other problems still remain. Another more recent study (Kari, Marianne, Eli, Asgeir & Christine 2021) that studied the impact of online classes amid the COVID-19 pandemic, concluded that many teachers and students found difficulty in adjusting to the various online mediums and had very little time to get used to the new technology, and found it labor-inducing to learn the various tools required of them to use.

3. Report on DisClassroom Bot

3.1. Platform – Discord

Discord was created on 13th May, 2015. Since then, it has grown into one of the leading social media platforms, with a demographic mainly consisting of young adults and teenagers. It is an application available on all devices and platforms, and its various features make it the perfect online study destination.

Discord consists of communities in the form of "servers". Users can create an account on the app/website, and join a pre-existing server or create their own server. Each server consists of channels, for different topics of discussions. Discord has options to text, voice and video chat, individually and even in groups. Roles in a server represent the different set of permissions grouped under a name, that can be assigned to user(s) in the server. Permissions for each channel can be modified role-wise or individual user-wise. Permissions decide the extent of a user's abilities in a server – which channels they can see or cannot, which channels they can write in, whether they can upload files or not etc. Discord even allows users to react to messages, using emojis, that can also be used in the text chat.

Each user, message, channel, server and other models on Discord have its own unique ID, which is usually a large integer, ranging from 17-19 digit numbers. This is important when it comes to programming, as these IDs help recognize and identify a user or a message uniquely.

Discord's developer portal allows any user to create a web application, that can be used for various purposes. The most commonly used application is "bots", which represents a bot user, that can be run using a script and interacts with Discord's API to automate various tasks and can be equipped with commands and listeners. Each bot user has a unique token that grants them access to the API. This token can be revoked or banned temporarily or permanently if the bot is abused, breaks any of the Terms of Service or exceeds any rate-limits repeatedly.

Bots can range in functionality from being simple bots – consisting of few commands and performing some automated tasks, to complex bots – games bots, music bots, bots that interact with other APIs to further enhance their functionality. Hundreds of public bots can be found online that perform a range of different operations, and server admins can decide to invite those bots to their server, as per their requirement.

3.2. API Wrapper - Discord.py

An API wrapper provides a way to access an API through a particular programming language or interface, which can help streamline the process of making API calls. There are several open-source Discord API wrappers available in different languages as per a developer's preference and proficiency.

Discord.py is the most popular API wrapper for Discord, written in Python and the second most popular API wrapper for Discord overall. It is open-source and maintained by a team of dedicated developers. This wrapper allows you to establish a connection with the Discord API, via the bot token, and incorporate commands and event listeners into your code, with the help of decorators. It also allows you to create internal tasks, that can be run in the background, in a loop if required. The documentation for discord.py is available online, as well as its code on GitHub.

3.3. Virtual Private Server – Google Cloud

Discord bots are required to be online at all times so that they can function effectively and be available for use at any time. Hence, we require a Virtual Private Server or a VPS, to keep the bot operational at all times, with an internet connection running at high speeds. For this we use a small VPS provided by Google Cloud, which runs Ubuntu OS, with 4 GB RAM, 20 GB storage on SSD, and has a download speed of 1689 Mbit/s and upload speed of 802.61 Mbit/s.

3.4. Database - MySQL

A MySQL relational database is locally hosted on the VPS and used for storing various necessary information persistently. It consists of three tables primarily – servers, students, and assignments.

"Servers" stores the information of each server that the bot joins and its respective configured channels and roles. "Students" keeps a track of all the students who join a server where the bot is present and stores their personal information provided by them. "Assignments" stores all the assignments given by teachers and all necessary details pertaining to it. Additionally, a table is created for each assignment, to keep track of all the submissions by the students.

Field	Type	Null	Key	Default	Extra
assignmentID	int	NO	PRI	NULL	auto_increment
serverID	bigint unsigned	YES		NULL	
subject	varchar(100)	YES		NULL	
title	varchar(100)	YES		NULL	
teacherID	bigint unsigned	YES		NULL	
assignmentLink	varchar(255)	YES		NULL	
hasDeadline	varchar(100)	YES		0	
deadline	varchar(100)	YES		NULL	
deadlineReminder	varchar(100)	YES		NULL	
isReminder	varchar(100)	YES		0	
deadlineOver	varchar(100)	YES		0	
totalMarks	int	YES		NULL	
marksheet	varchar(100)	YES		NULL	
marksReleased	varchar(100)	YES		0	
postTime	varchar(100)	YES		NULL	

Table 1: Assignments table's schema in the database

Field	Type	Null	Key	Default	Extra
ID	int	NO	PRI	NULL	auto_increment
serverID	bigint unsigned	YES		NULL	
entryCategoryID	bigint unsigned	YES		NULL	
welcomeChannelID	bigint unsigned	YES		NULL	
streamCategoryID	bigint unsigned	YES		NULL	
announcementsID	bigint unsigned	YES		NULL	
teachersLoungeCategoryID	bigint unsigned	YES		NULL	
staffRoomChannelID	bigint unsigned	YES		NULL	
studentRoleID	bigint unsigned	YES		NULL	
teacherRoleID	bigint unsigned	YES		NULL	
configured	varchar(100)	YES		0	
joinTime	varchar(100)	YES		NULL	

Table 2: Servers table's schema in the database

Field	Type	Null	Key	Default	Extra
studentName	varchar(100)	YES		NULL	
studentTag	varchar(100)	YES		NULL	
studentID	bigint unsigned	NO		NULL	
serverID	bigint unsigned	YES		NULL	
rollNo	varchar(100)	YES		NULL	
email	varchar(255)	YES		NULL	

Table 3: Students table's schema in the database

3.5. Introducing bot to a server

The bot can be introduced to any server by an admin, with the help of its invite link. Once invited to a server, the bot will prompt the admin to configure the bot for the server. Using the designated command, the admin is expected to configure the channels – entry channel, announcements channel and teacher's channel, and the two roles – Teacher and Student, for the server. Once this is done, any new user joining the server will be asked to pick whether they're a student or a teacher in the entry channel. If they're a teacher, the bot would request confirmation by another teacher to grant them the role and if they're a student, they will be asked for their details in DMs and then given the role. A help command is available to both teachers and students, for help with using their respective commands. All commands have to be prefixed with 'c!'.

3.6. Teacher interface

A teacher will have 5 available commands to them, which can only be used in the private channel selected upon configuration. These commands are

1. info – This command is used to retrieve the details about a student, inputted when they joined the server. This includes their name, roll number and email.
2. ping – This command is simply invoked to test if the bot is currently online, in case of unresponsiveness. If online, it will return the bot's response time.
3. post – This command is used to post an announcement/assignment/quiz in the announcements channel of the server, which is read-only for Students.
4. review – This command is used to review the submissions by students for an assignment and to mark them accordingly. To accomplish the latter task, the bot also incorporates the Google Sheets API. A Google Sheet is created each time when an assignment is assigned to all students, with a list of all the students in the server, and a column for marks. Teachers can use the "review" command to review the submissions by the students and then assign their respective marks in the sheet attached to the assignment.
5. release – Once a teacher has assigned the marks for all the students in the Google Sheet retrieved using "review" and saved changes, they can release the marks using this command. All students will receive their marks for the assignment via DMs. If any changes need to be made, they can be done so in the Sheet and released again. Only affected students will receive their updated marks the second time.

3.7. Student interface

A student will have 3 available commands, which can only be invoked in DMs with the bot. The commands are –

1. view – This command allows students to view all of their assignments or all of their pending assignments, from all mutual servers they share with the bot, and see details about them.
2. submit – This command is used to submit an assignment. The student can even choose to turn in their assignment with no attachments.
3. resubmit – This command is used to resubmit a previously submitted assignment.

   Students also receive DM reminders of pending assignments, a day or an hour before the deadline of an assignment.

   "c!help" can be used to view all available commands and their purpose and "c!help <command_name>" can be used to get additional help with a specific command.

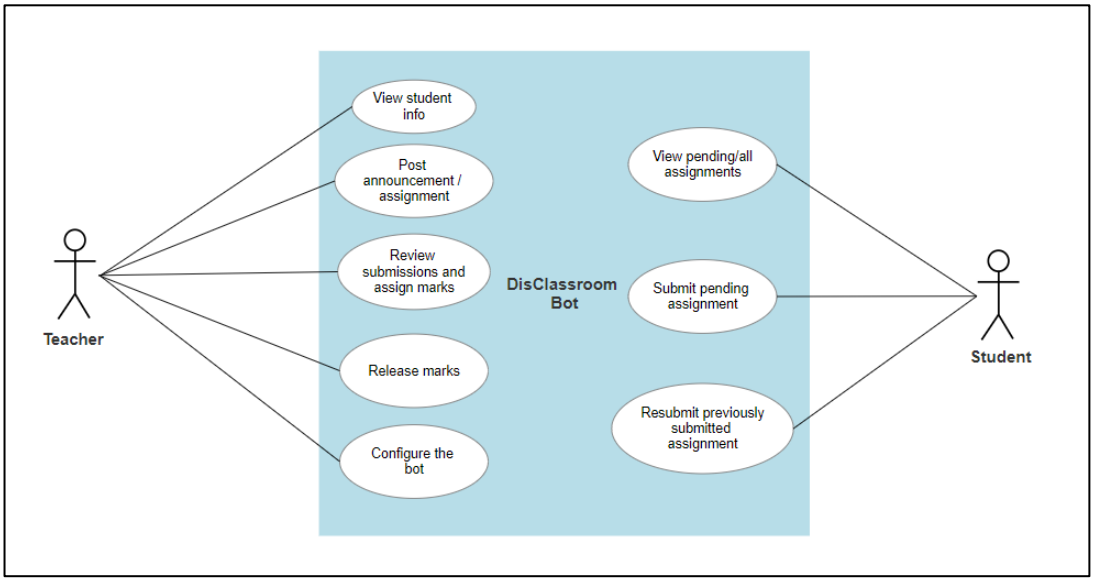

Figure 1: UML Use Case Diagram for DisClassroom bot

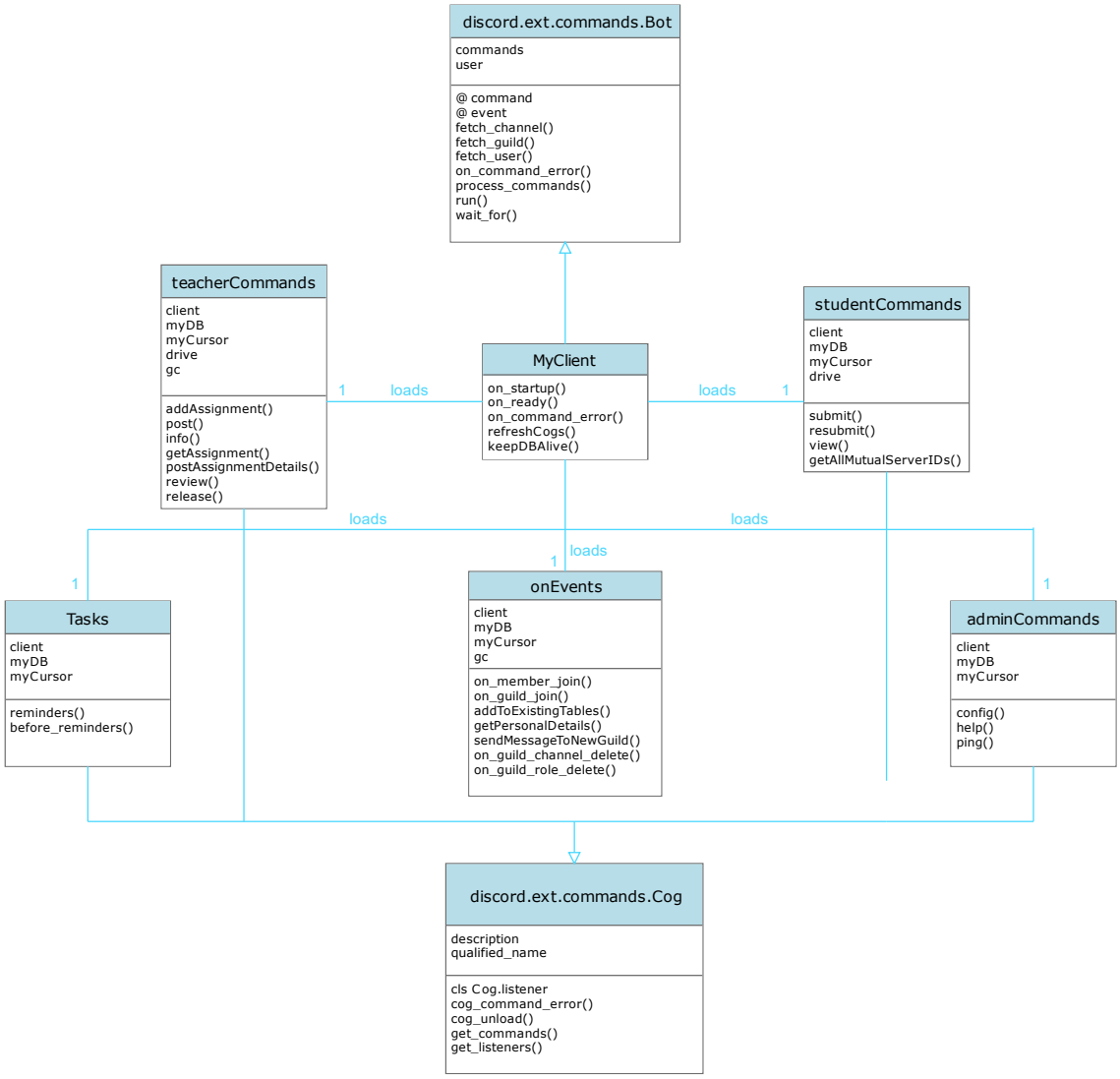

Figure 2: UML Class Diagram for DisClassroom bot

3.8. Cogs

Cogs are a part of discord.py that allow a developer to organize their script into collections of commands, listeners and some state into one class. This allows the code to be more organized and also the chance to group certain commands and listeners together. DisClassroom uses cogs to separate the admin, teacher and student commands, as well as to separate the event listeners and background tasks. The other advantage with using cogs is, they can be unloaded and re-loaded into the bot whenever required. This can be beneficial if there is a requirement to refresh some instance variable used within the cog.

4. Results

The bot is finally loaded and executed on the VPS, and is online and available for use. The source code for the bot is available on GitHub. With the help of this bot, educational institutions will be able to fully manage their online classes on just Discord. Classes can be taken on video calls that Discord allows on their platform by the means of voice channels in servers, while announcements, assignments and quizzes can be posted and tracked using the DisClassroom bot. Once set-up, teachers have to worry about nothing and the bot will deal with new students joining itself and assign them their role. The bot will also send DM reminders to students concerning their assignment deadlines.

This is, however, the first version of the bot, and there are prospects for improvements and scope for future developments. Some of those include:

1) Configuring institute's own email domain and ensuring students provide only their official institute email address.
2) An email verification system to ensure the institute email belongs to the student.
3) Making enhancements to the bot, so that the various subjects can also be configured for each server and teachers assigned to each subject accordingly.
4) Updating the "c!post" command to make anonymous announcements and even taking Student role ping as an optional parameter for the same.
5) Feature for 2-way private comments on assignments.

5. Summary and Conclusions

I was able to use the Discord API, with the help of the API wrapper discord.py, to create a bot that could be useful to servers of educational institutions and provide an alternative to the current solutions that exist like Google Classroom or Microsoft Teams. Today, students are expected to use Gmail, Google Meets, Google Classroom and WhatsApp, all just to keep track of their online classes, assignments, notices and to stay in contact with their teachers and friends. With the help of my bot, Discord can serve as a one-stop solution for all these needs of an online class. A student can stay in touch with their friends and teachers, attend classes, view and submit their classwork, and read any announcement, all on the same application, which is available on all devices, for all operating systems. A teacher can also keep track of all their students, their submitted classwork and their marks, all in one place.

However, I have only taken the first step in making online classes less cumbersome for students. There are several other problems facing online classes that are yet to be addressed and targeted. With my bot, I also hope to make the process of online classes more fun and enjoyable for students. In the future, I hope to make the bot even more advanced, allowing students to store reminders on it, and to incorporate games and other tools that will make students more attentive and motivated to study online.

Appendix I

Terminologies associated with Discord

1. Server – Also referred to as “guilds”, servers are the spaces on Discord. They are made by specific communities and friend groups. Any user can start a new server for free and invite their friends to it.
2. Channel - Discord servers are organized into text and voice channels, which are usually dedicated to specific topics and can have role-wise or member-wise permissions.
    a. In text channels, users can post messages, upload files, and share images for others to see at any time.
    b. In voice channels, users can connect through a voice or video call in real time, and can share their screen with their friends.
3. User – It refers to a user on Discord i.e., a person’s account.
4. Member – It refers to a user, who is part of a particular server. They’re considered a “member” of the server.
5. DMs - Users can send private messages to other users as a direct message (DM), as well as start a voice or video call.
6. Roles - A role is a defined as a set of permissions with a name. Roles are made for a server and can be assigned to its members. Roles can be given a color (or left colorless) and they have a hierarchy. A member’s top-most role decides the color their name will appear in when they speak in that server. Roles also allow the members to appear separately in the member’s list.
7. Permissions – They are a way to limit and grant certain abilities to users. A set of base permissions can be configured at the guild (server) level for different roles. When these roles are attached to users, they grant or revoke specific privileges within the guild.
8. Bots - A Discord bot is an automated code/program that runs specific activities in your Discord channel, such as moderation and offering commands to members to easily accomplish certain tasks.

References

[1] **Hara, N., & Kling, R.. (1999)**. Students' frustration with a web-based distance education course. *First Monday Issue 4*(12).

[2] **Howland, J.L. & Moore, J.L. (2002)**. Student perceptions as distance learners in Internet-based courses. *Distance Education, 23*(2), p. 183-196.

[3] **Vonderwell, S. (2003)**. An examination of asynchronous communication experiences and perspectives of students in an online course: A case study. *Internet and Higher Education, 6*(1), p. 77-90.

[4] https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0250378

[5] https://discordpy.readthedocs.io/en/stable/index.html

[6] https://pythonhosted.org/PyDrive/

[7] https://docs.gspread.org/en/latest/api.html

[8] https://dev.mysql.com/doc/refman/8.0/en/

[9] https://discord.com/developers/docs/reference

Acknowledgements

I would like to sincerely thank Dr. Uma Sheshadri and Mr. Shankar Biradar for their continual guidance and support in helping the completion of this project and its report, as part of our Object Oriented Programming course for semester III of the B.Tech CSE program.

- Samuel Mathew [20BCS116]

21st Dec., 2021.