Piazza Reward Bot – Team 10

Zijun Lu, Zhiyuan Ma, Kai Gao, Rachel Chen

Problem

Piazza is a popular learning management platform that allows teaching staff and students to interact in a forum-style format. According to Business Wire, Piazza has been used by more than 5 million students and over 100,000 professors from 2,000 schools in 90 countries worldwide, and instruction has expanded from STEM to all disciplines[1]. There is no doubt that Piazza is a useful tool for instructor-student interaction, especially in the software engineering domain. However, there are two problems that a reward bot would help solve.

Piazza provides a standard interface for students and instructors to share resources and engage in productive discussions. However, Piazza does not offer many incentives for students to answer and ask questions, nor regular activity summaries. We believe that if teaching staff have the option to offer course credit incentives based on students' Piazza activity, it will significantly increase the initiative of enrolled students to participate. Our reward bot can help achieve this through the integration of an open-source and self-hostable online chat service and by providing customizable summaries of class performance in a shared group channel (e.g. on Discord). This will be especially useful for courses that are just starting to use Piazza, such as many non-computing courses, since students will not be used to Piazza and participate less.

Furthermore, another problem is that Piazza is a paid platform. The unpaid version is available for classes of 25 students or less but hides many useful functionalities. For example, if the class participation statistic is disabled, instructors cannot easily figure out the most active students in the class. If the teaching staff want to provide extra credit to students who contribute the most to the course forum, it will be painful and time consuming to deduce who to reward. Therefore, the introduction of our reward bot will help instructors struggling with course budget in a small class setting by regularly retrieving Piazza data in a customizable manner.

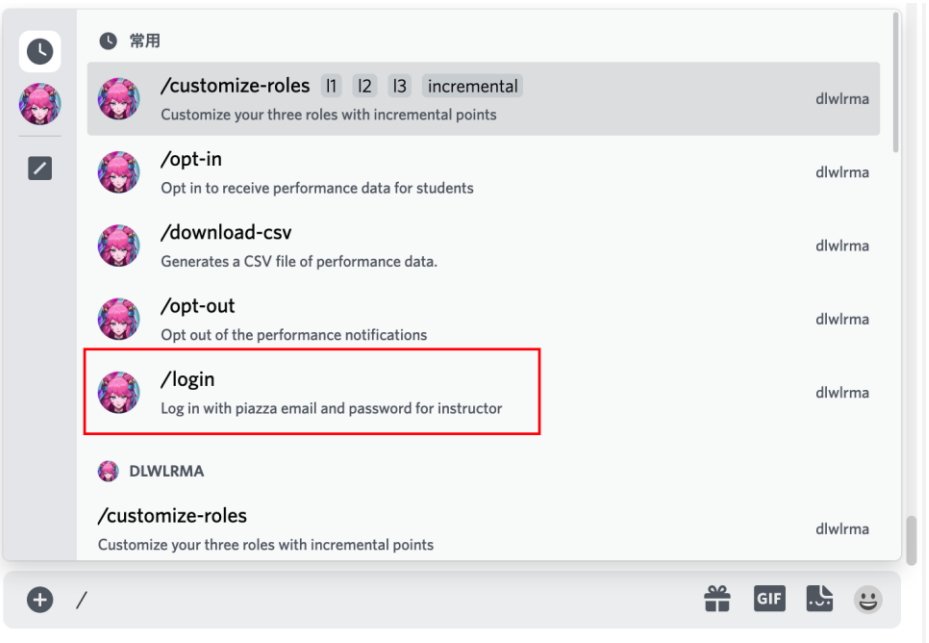
Primary Features and Screenshots

Our bot has a variety of features designed to help students track their Piazza forum participation and to motivate students to participate more on Piazza. The bot collects data from Piazza on the different types of post activity from students: questions (including notes), answers and follow-ups to questions, views on posts, and endorsements for questions and answers/follow-ups. Based on the quantity of each activity type, our bot assigns points to each student. Instructors can customize the weights on each activity type when calculating points. These points are used to compare student participation on Piazza and are announced through daily performance summary messages and private incentive messages from the bot. The points also determine the roles assigned to students on Piazza. More points mean higher level roles, providing another way for students to track their participation compared to their classmates. We explain the primary features with more detail below.

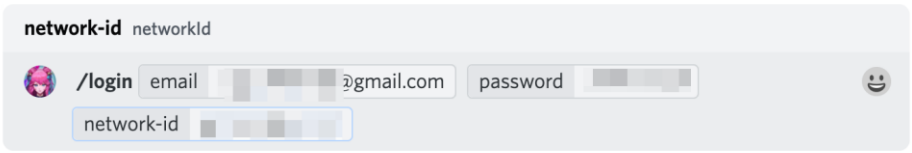
USE CASE 1 - Daily Performance Summary:
Our bot sends a message once a day in a public Discord channel announcing the 3 students who have earned the most points for the past day. This will recognize the achievements of those students and motivate others to be more active on Piazza. Points are automatically calculated by the total number of Piazza activities, but instructors can customize weights to determine how important each activity type is.

To utilize the bot features, the instructor will need to associate a specific Piazza course ID from the Discord channel by login command:

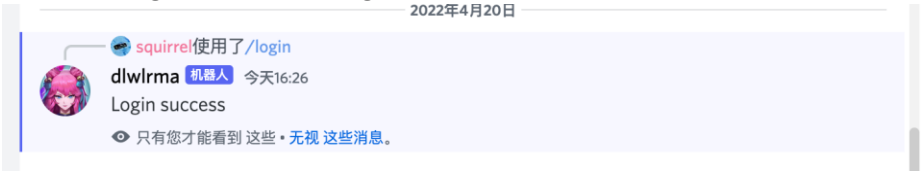
Login command:



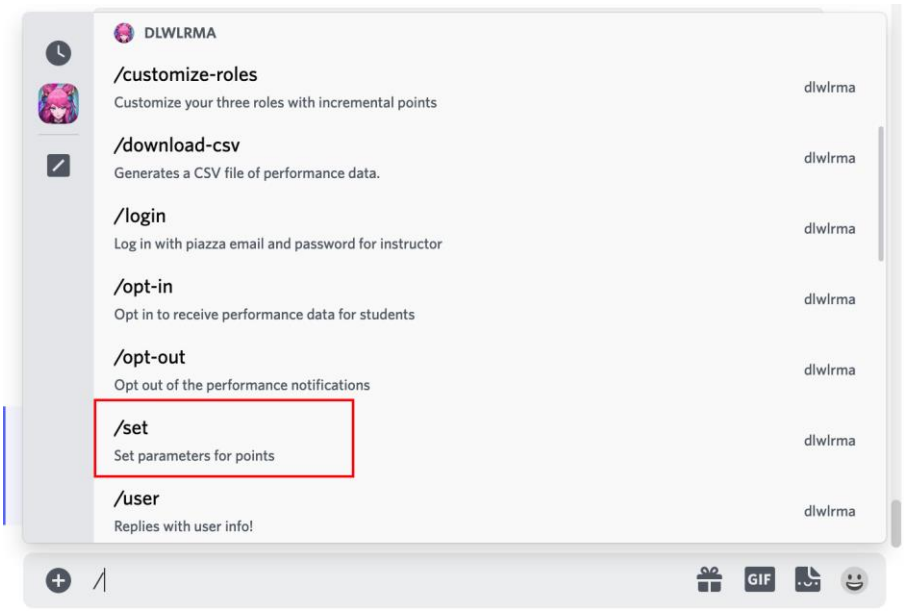
Instructor inputs login:



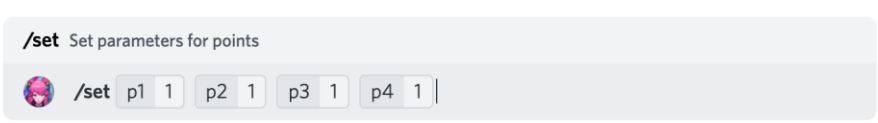
Expected result: Private login success message in Discord channel



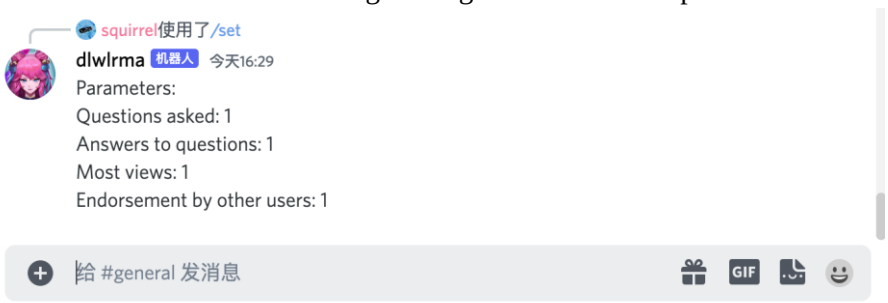
To set the weights for each Piazza activity type, the instructor can use the '/set' command:



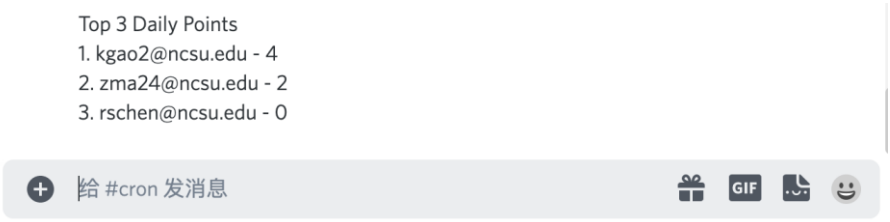
They will enter the respective weights for the “Questions”, “Answers/Follow-ups”, “Views”, and “Endorsements” categories.



Expected result: The bot will send the following message in the channel upon success:



Sample performance summary message in public Discord channel:

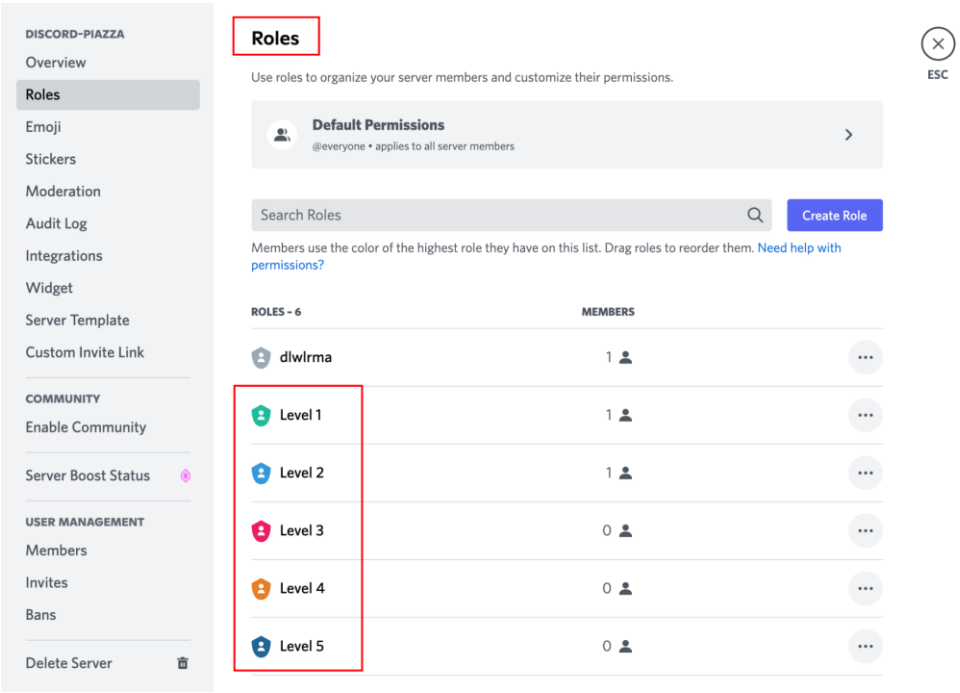


The instructor will need to keep in mind that the database, which stores the Piazza data and point parameters, is only updated once a day, so they may not see points updated correctly until the next day if they want to change the point parameters, which are each 1 by default.

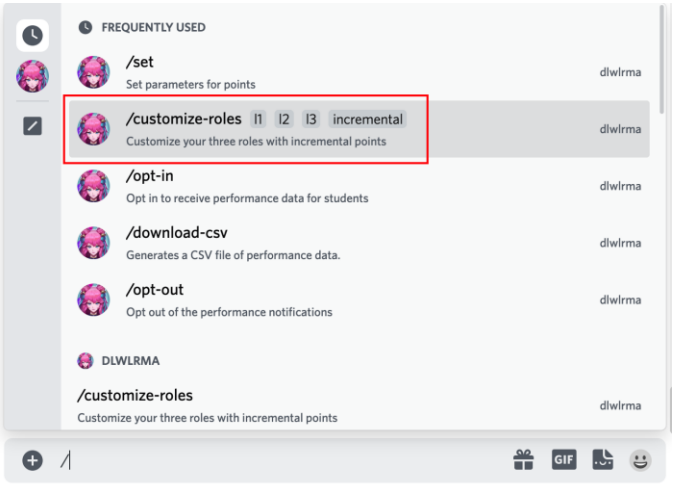
USE CASE 2 - Discord Roles and Levels:
Each student will have a level assigned to them in Discord indicating how active they are. More points equates to a higher level. Students can only see their level in Discord when they opt into private incentive messages (use case 3) from the bot. The levels will allow students to see how their Piazza activity compares to their classmates'. Instructors can use levels as another way to determine how to reward credit based on class participation. This will incentivize students to be more active on Piazza to raise their level.

Instructors can customize role/level names and the number of points to reach a new level using the '/customize-roles' command.

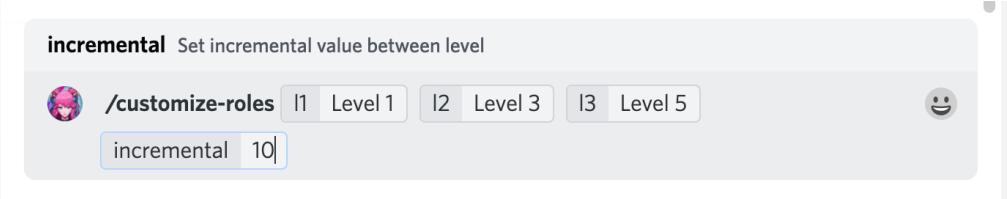
Roles/levels shown in the Discord server: (Notice: The role name "Level 1" is from the setting of this server. The title associated with "Level 1" can be customized.)



Example for '/customize-roles':

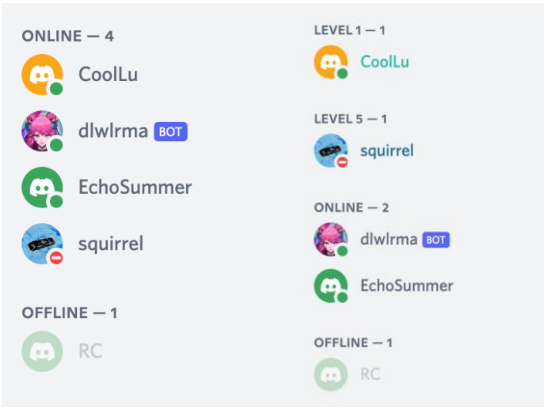


They will enter the level names and incremental points between levels. For example, the following command means Level 1 requires 10 points, Level 2 requires 20 points, and Level 3 requires 30 points.



Expected result: Message in Discord channel upon success:

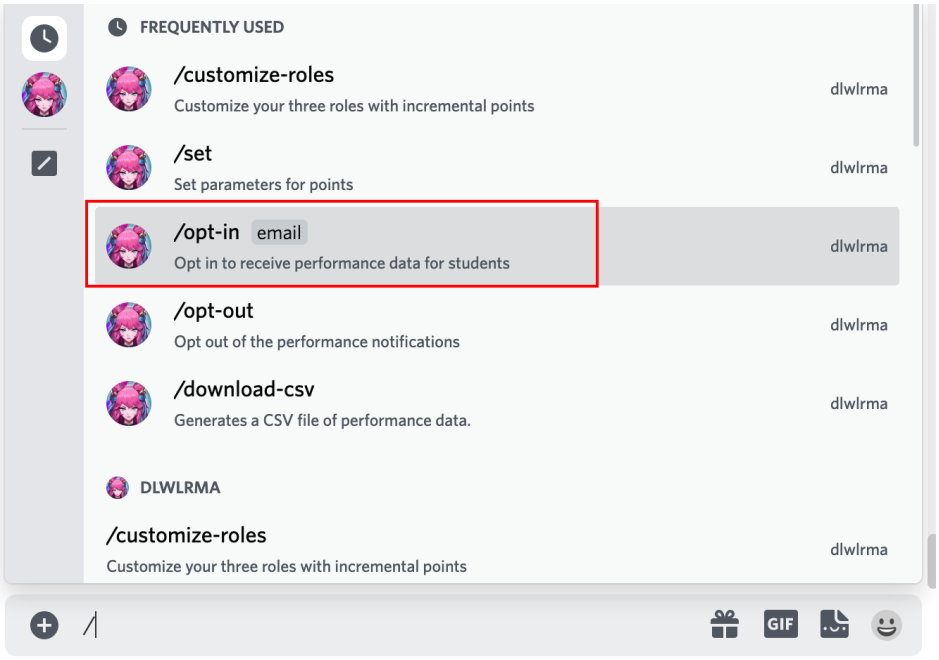
Before (left) and after (right) of '/customize-roles' command: Level assignment has been updated if users opt in to the incentive service and have reached point requirement for the levels.



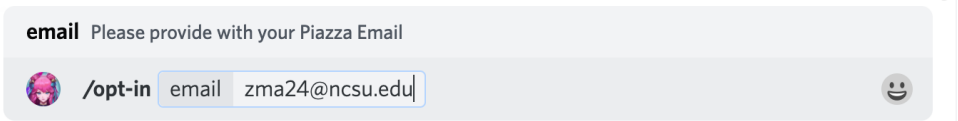
USE CASE 3 - Personal Incentive Message:
Students have the option to receive personalized private messages from the bot in Discord. Since the performance summary messages only include the top 3 students, most students will not know how many points they earn every day. By subscribing to the bot messages, the bot will update a student on the number of points they earn every day. Students can use this information to see how they compare to the top point-earning students every day, which will help motivate them to be more active on Piazza in order to catch up to the top students. Subscribing to the bot messages will serve as a precondition to seeing your level role, which will also help students compare their Piazza activity to their classmates'.

Students can subscribe and unsubscribe respectively to the bot messages with the '/opt-in' and '/opt-out' commands.

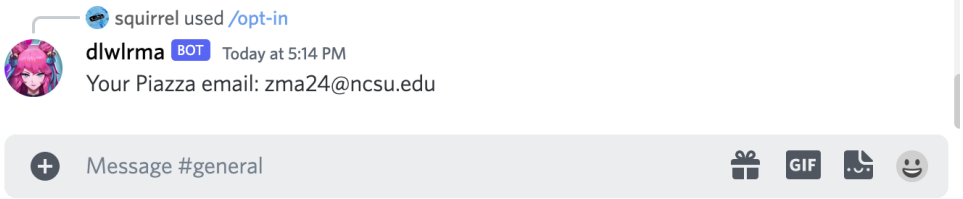
Subscribe from bot message with /opt-in command:



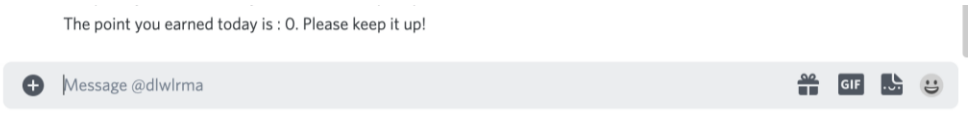
Student should enter the email used to register for Piazza to subscribe to the incentive message.



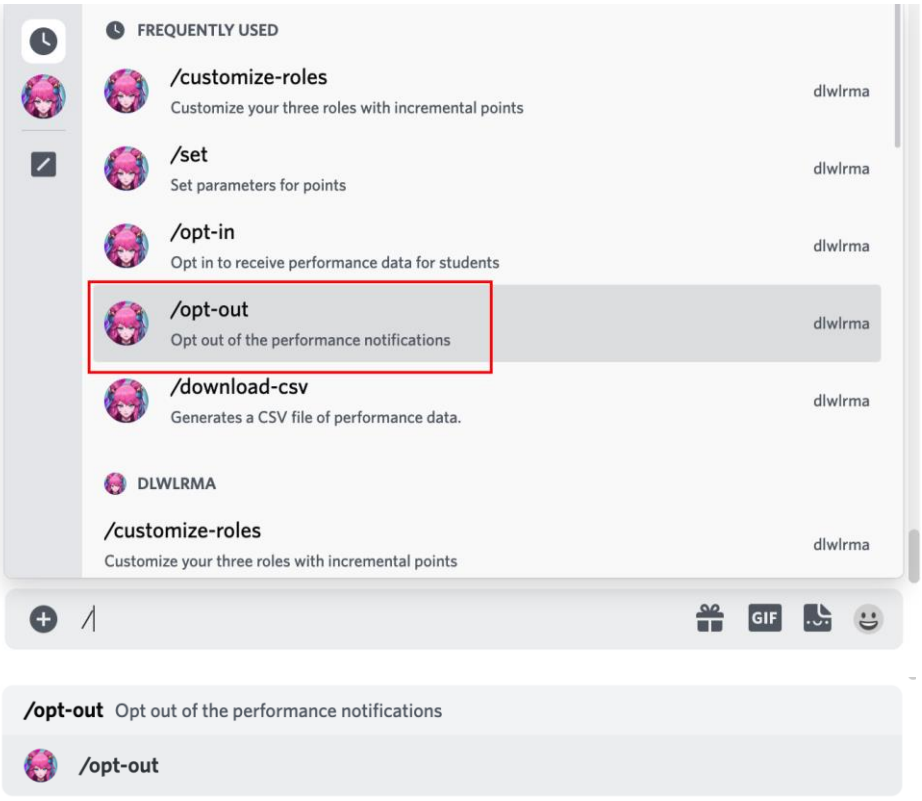
Expected result: Bot replies with confirmation message containing the email the student enters.



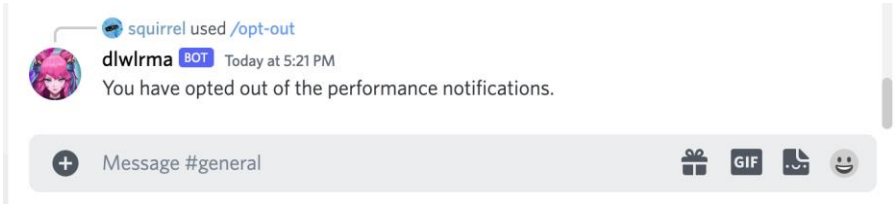
Example private bot message:



Unsubscribe from bot message with /opt-out command:



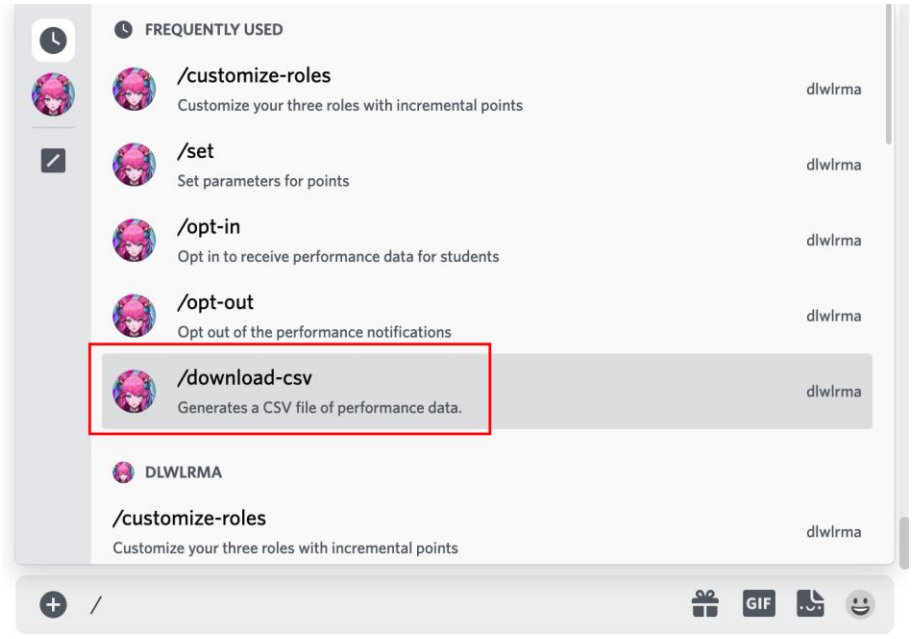
Expected result: Message upon success:

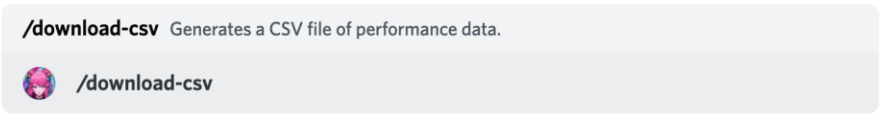


USE CASE 4 - Download Piazza Data:

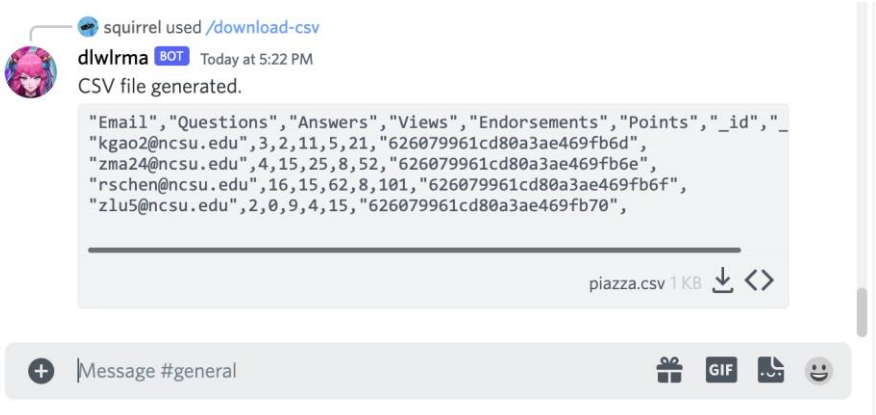
The instructor also can download a CSV file containing data for each student on their Piazza activity and total points as academic records. The points are according to the point parameters the instructor has set. The instructor will need to keep in mind that the CSV will only reflect the data currently stored in the database, which is only updated once a day. The CSV file can help the professor track overall participation activity in their class and decide how to reward points at various points of the semester, such as during midterms or at the end of the semester.

Instructors can generate a CSV file to download using the '/download-csv' command.



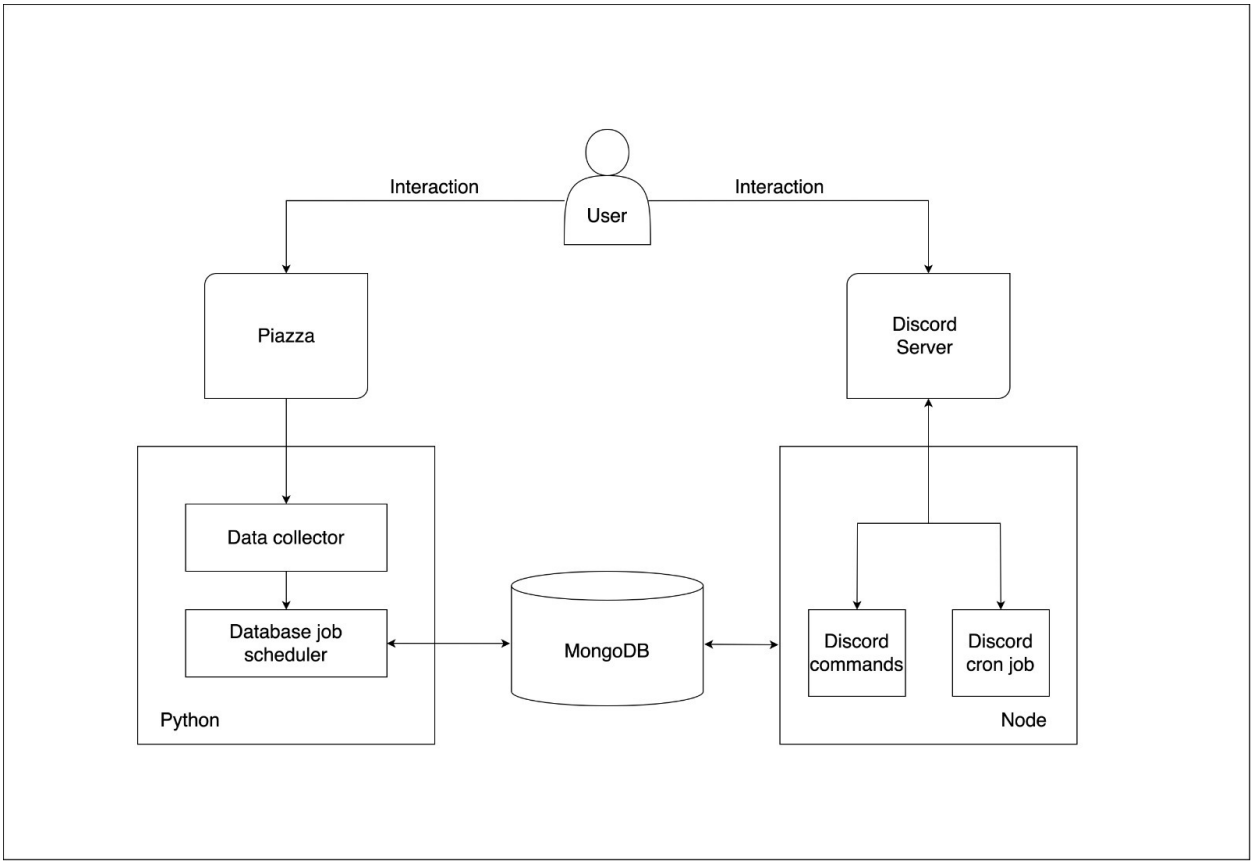


Expected result: CSV generated and ready for download.



Bot Architecture

We also present a diagram of our bot's architecture. The bot has two components; one interacts with Piazza to collect data and update the database, and the other interacts with Discord to process user commands and fetch information from the database based on the commands. The Piazza component is implemented in Python, while the Discord component is implemented in Node.js. The Piazza component uses an unofficial Piazza API (https://github.com/hfaran/piazza-api) to connect to Piazza from the data collector. The database job scheduler calls the data collector to get Piazza data, then uses the PyMongo library to interact with MongoDB and update the database accordingly. On the other hand, discord.js utilizes an event-triggered based scheme, in which our BOT waits for users to enter slash commands to trigger functions, such as database CRUD and sending messages to users. As our defined use cases involve messages sent in different time frequencies, the Discord cron job planner is implemented to handle such issues.



Reflections

This project was a good learning experience. The bot implementation was challenging, since we had to explore new APIs and libraries such as the unofficial Piazza API to scrape data, but it was more straightforward than the bot design and creating mocking data for user interactions to use in unit testing. It was interesting to explore the different design patterns from the lecture and learn about the scrum-ban process, which most of us were not familiar with. The scrum-ban practices were a bit tedious to implement, but they helped us plan and start our work earlier, especially with the daily stand-up meetings.

Below are some specific reflections regarding the implementation of our bot and how we could have improved the process:

1. When we were figuring out how to collect data from Piazza, we tried some data scraping libraries. However, we found that it is hard to scrape data from a dynamic website like Piazza. To solve this, we used an unofficial API implemented in Python to solve this problem. From this we learned the importance of utilizing existing APIs rather than trying to implement some functionality ourselves. This is especially true with a language like Python, which has many convenient APIs available.
2. We learned to plan out our output formats for functions more thoroughly before starting implementation. When we did the Piazza data collection work, we finished the first implementation very quickly, but when we were implementing scheduling jobs to update the database, we found that the format of our data collection output was not suitable. We had to refactor our code many times to get a suitable output format to provide to the next function. If we thought further ahead before starting our work and researched correct output formats for different libraries we were working with, then we would have saved a lot of time.
3. We also learned how pair programming techniques play an important role throughout our project development phase. The pair programmers not only bring different prior experiences/knowledge to better optimize the system or solve the problem, but also teach something new to the partner each time. We utilized this as we split into two smaller groups during the sprints, the Piazza group and Discord group. In case one person is not available to a query or question regarding program interaction in a work day, the available group member can step in and provide help to the other team as they were pair programming. We believe that continuing promoting pair programming in our collaboration can improve code quality and handle teams' time conflict.

Limitations and Future Work

Throughout bot development, we found several limitations regarding data retrieval and bot features. One limitation is that our bot cannot perform real-time updates to data. Data pulls from Piazza utilize HTTP requests and have to be spaced apart for some amount of time, since too many requests at once will cause data pulls to fail. When there are many posts in a Piazza forum, a data pull will take at least a few minutes to complete. Therefore, especially for Piazza forums with a lot of activity, it is not possible to provide instantaneous updates to the database (and thus the data sent in private messages, performance summaries, and CSV files). For forums with lower activity levels, real-time updates are not necessary. Since the database is only updated once a day in our bot, if the instructor wants to reset point calculation parameters, they also will not see instant updates to data if performance summaries are only sent once a day. Furthermore, the CSV file of Piazza data they can download will not reflect the most updated data,

but only the updates since midnight of the current day. We used an unofficial API to interact with Piazza, but if Piazza releases an official API in the future with a better, more reliable way to pull Piazza data, this problem could be mitigated.

Additionally, when working with the Piazza data, we filtered out certain types of Piazza activity. For instance, we did not count updates to posts as activity since it was trickier to decide how to reward points. We also removed anonymous and private posts, since it would be impossible to decide who to reward without knowing their identity, and it did not make sense to award points on posts other students could not access. Our test Piazza forum did not have the option to be anonymous only to classmates, so this would be another aspect of the data to explore in the future.

Another limitation is that the instructor will need to be familiar with MongoDB in order to set up a database for the class. They will have to manually input the Piazza course ID to MongoDB and input the database link and other credentials to a configuration file. We could provide written instructions for them, but since each class will have a unique database, Piazza forum, and Discord server to create, this part would be harder to automate.

Furthermore, there are a number of features we removed from our use cases or did not explore for simplicity. For example, we did not try integrating multiple courses into the bot, since this presents challenges with the database, Discord server(s), and bot implementation. It would also be hard to coordinate the bot with different teaching staff teams and groups of students. Some features we removed for simplicity from our use cases include a weekly option for receiving performance summary messages and personal incentive messages as well as customizing level roles with emojis. These features could all be added in future development for the bot to make it more versatile.

References

1. *Leading colleges and universities across the U.S. select the Piazza Q&A platform for Enhanced Virtual Learning experiences*. Business Wire. (2021, January 26). Retrieved February 17, 2022, from https://www.businesswire.com/news/home/20210126005439/en/Leading-Colleges-and-Universities-Across-the-U.S.-Select-the-Piazza-QA-Platform-for-Enhanced-Virtual-Learning-Experiences