# UTAS Discord Bot MyLO Integration

## Background

I’ve been continuously developing and improving a Discord bot for facilitating Online and Blended Learning. (In case you haven’t used it/heard of Discord, it is effectively Slack/Teams but much more user friendly, and software \***already**\* used by most students outside of their studies). We’ve been using the bot for some time now across 10+ different unit offerings, and I believe you’ve seen myself and Ian Lewis talk about this at Teaching Matters last year. The bot tracks student attendance and engagement on Discord, gives out achievements, informs students about upcoming (f2f and online) sessions. It’s feature set is intended to improve the quality of online and blended teaching by providing features for both Staff and Students. The bot’s functionality and security at this point is stable, and I am looking to expand its capabilities.

You can find a current user guide showing most of the bot’s features at the following URL: <http://131.217.172.176/guide>

Note: The link above is just an IP address, however we are in the process of deciding upon a suitable domain name for this.

You can find the code for the bot open-sourced here: <https://github.com/lfwells/discord-lecture-reader>

## Proposal Summary

In short, adding MyLO / Brightspace API integration into the Discord bot would help provide an even more engaging and user-friendly experience for students and staff using our Discord-enabled units (in much the same way that MyLO MATE does an \***excellent**\* job at improving the MyLO user experience and efficiency for staff). It would also be a reason for more staff to incorporate this tool in their own units. Currently, the list of students and their data is loosely and **manually** tied to the wealth of information on MyLO via students using their UTAS username on Discord. This is difficult to enforce and doesn’t allow the bot to securely give students any of the information on MyLO (such as grades/content/etc).

With a secured OAuth token linking student and staff Discord accounts to their authenticated MyLO user account, the Discord bot can safely provide students with real-time and up-to-date answers to their queries. Students would interact with the bot or with staff on the Discord server, and the answers to their queries (such as “Have I completed all the tutorial work?”, “When is the next assignment due?” etc) can be provided instantly and cleanly, without the need for either staff or students to navigate the (sometimes difficult) MyLO interface for this information. Note that responses to queries can be shown either publicly (i.e., to all students on the Discord server), or privately to students via a Direct Message or “hidden” reply.

## Security/Access Considerations

Note that ALL planned integration would have no visual impact on the actual MyLO site itself, everything in this document describes features that read \***in**\* to the stored back-end data of MyLO and sits entirely separate to MyLO.

Initial development of this integration would be limited to a select number of units of which I am Unit Coordinator, or close collaborator Ian Lewis is a collaborator. These initial units would be from:

* KIT109 Games Fundamentals (Sem 2 2022, Sem 1 2023)
* KIT208/724 Augmented and Virtual Reality (Sem 2 2022)
* KIT207 Games Design and Production (Sem 2 2022)
* KIT305/721 Mobile Application Development (Sem 1 2023)

The long-term goal of the Discord Bot and the MyLO integration is to be used in any unit where a Unit Coordinator has undergone the proper training to use the tool.

The features documented below vary in terms of what OAuth “scope” they would need, and given the sensitive nature of data on MyLO, I would take an iterative approach to development here, with cautious permissions/scope granted to the bot -- i.e. I would begin with read-only features like retrieving grades or content, and cautiously expand later to write-requests for automated grading etc. (this iterative approach would involve some back-and-forth with your team, as the scopes change over time). Initial development would also take place on the MyLO “dev-2” instance (i.e., a non-live instance) to avoid bugged code making irreversible changes to live units. After chatting with Connor about the security of the app, I am confident that there is extremely low risk in terms of the security of this system in terms of accidentally revealing sensitive student information.

The OAuth credentials for the application would be stored securely on the server, and not committed to source control.

## Infrastructure (already in place)

The current infrastructure for the bot is advanced and stateful – i.e., it has a persistent server and database running, along with an administrative “back end” web page for Unit Coordinator to manage configuration, class lists etc. It is on this back end that Unit Coordinators would link their MyLO accounts to a unit Discord server via OAuth. Students would link their MyLO accounts via Discord itself with a uniquely generated link. The database would store OAuth tokens and MyLO account IDs in the database. The server is hosted on the ARDC Nectar Research Cloud, and the database is stored using Google Firebase. The \***only**\* MyLO-related information that would be stored would be the MyLO account id, OAuth token, and the Organisational Unit ID for a unit would be stored for each linked Discord server. MyLO-related information can be stored separately on a database hosted on NeCTAR, if required.

## Support

All support for this system (i.e., requests for help/how to use/etc.) would be handled by myself (Lindsay Wells). Extensive documentation has already created and would be updated as the MylO integration is added. As such, it is not expected that this would have any significant amount of workload. In addition, other staff members who use the system can easily provide basic-level support.

In terms of maintenance of the code and server, again it is expected that this will be maintained purely by myself. However, in the event of me not being able to continue supporting the project, other staff members such as Ian Lewis, Lachlan Hopkins, Connor Deckers, or any other future collaborators could be easily trained to take over this role.

## Features

Below is a list of possible features I have thought of initially (some even suggested by students, who are already very excited by the possibility of expanding the Discord bot).

Features for an authenticated Student account (note this is opt-in, if they press the button to authenticate themselves) – many of these features have been requested by students themselves, really illustrating the value of this project:

* Retrieval of all grades / specific grades
* Retrieval of all attendance / specific attendance
* Query of Assessment tasks
  + Next due task
  + All remaining tasks
  + All submitted tasks
  + (Above would include direct links to drop boxes)
* Automated reminders of upcoming due dates
* Retrieval of unit content
  + By week/module/topic name
  + Either as a link or as Discord formatted post
* A way of seeing all MyLO-integrated Discord servers that the student has access to
  + To support alumni joining servers, etc
* (WRITE ACCESS, Possible but potentially problematic): Automated assignment submission via Discord upload

Features for a Unit Coordinator / the Bot might include:

* Authentication of students
  + Students cannot impersonate another student by simply using their username
  + Non-enrolled students can be auto-kicked or auto-identified for restricted access (e.g., as alumni)
* Retrieval of grades / specific grades
  + For a specific student (e.g., to reply to a student querying this that doesn’t know how to operate the bot)
  + As a summary on the administrative back end, correlating grades with other stored information such as post count, attendance, achievement count etc
* Retrieval of attendance / specific attendance
  + For a specific student
* (WRITE ACCESS) automatic population of grades/attendance based upon Discord-recorded attendance
* Tagging/direct messaging of students without submissions
* Identification of Discord members without a linked MyLO account (indicator of disengagement)
* Automated cross-posting of Announcements to Discord announcements channel
  + (WRITE ACCESS) automatic posting of anything from Discord announcements channel to MyLO announcements channel
* Automated posting of timely content (e.g., post the lecture slides in lecture chat at start of online session, including automatic extraction of links feature already implemented)
* Automated mirroring of MyLO Content
* Automated notifications of MyLO Content changes (e.g., “tutorial 8 base code updated to fix bug”)
* Automated Discord “role” assignment based upon MyLO Group allocation (i.e., can tag all “Hobart” students or can tag all “Monday 2pm Tutorial” students – currently this is manually managed by staff and students)
  + Can restrict access to online tutorial rooms based upon MyLO group allocation.
* Generated links to student profiles / progression reports

## Project Feasibility / Support

Some points to illustrate my capacity and capability to complete this project:

* I have already begun considerable research into the MyLO/Brightspace API and have a solid understanding of OAuth practices, plus have the support of Connor who has practical experience of working with the MyLO API itself
  + This research has included exploring the MyLO Mate open-sourced code, as well as “inspecting” as it runs in the browser
* The Discord bot code is mature, and I have more than a decade of experience programming applications of this scale
* I have received a Teaching Merit Award directly related to my work on the Discord bot
* This project would be part of a greater body of work I am currently undertaking with ICT and the Digital Futures team to produce an ICT MyLO template, and best practices for using MyLO
* I have over a decade of experience in programming in web technologies, and am a full-time Level B academic in the school of ICT teaching these concepts
* I have the backing of
  + Professor Anna Shillabeer (ICT Head of School)
  + Assoc. Professor Kristy de Salas (Line Manager)

## Next Steps (Pragmatic)

The things I need to progress are to effectively begin the steps documented on this page: <https://docs.valence.desire2learn.com/basic/oauth2.html>

* Registration of an OAuth application credentials
* Setting the initial OAuth scope of the application to get a proof of concept working
  + A simple example would be to extract the list of students for an Organizational Unit that the authenticated Unit Coordinator has access to
    - I believe the scope to enable for this would be **enrollment:orgunit:read**
  + Full list of scopes here: <https://docs.valence.desire2learn.com/http-scopestable.html>
* Configuring available OAuth redirect URLs:
  + For now, the redirect URL would likely be <http://131.217.172.176/myloAuth>
  + If Brightspace config doesn’t allow IP addresses or non-https, then I can easily get a domain name up and running for this
* I’m not 100% sure how it works behind-the-scenes, but initial proof of concept development could be restricted to the “dev2” MyLO environment, I assume a separate OAuth application registration needs to be done for each environment?

## References

This approach of tying a Discord bot to a LMS has been done in the past with a good level of success. One such paper documenting this is “Learning during COVID-19 Pandemic: Online

Education Community, based on Discord” by Monica Vladoiu and Zoran Constantinescu

<https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9324863&casa_token=dgDVwTQUenoAAAAA:eIBaPMFz0nLxsm6mfDX4PooEiMMGSfrl9Id80kOPJ3gJ1q414Dei22XueEvnhpTQXDnNSrBUXps>

Note that my approach would be more fully featured and usable than the one documented there.

I also refer you to recent presentations at Teaching Matters for more information about the Discord bot: <https://www.youtube.com/playlist?list=PLEQ0dw7cj0Ncu3npwGS7HTjEJqwrNcMpp>