Project outline: Student identity verification for GitHub Classroom

# Problem outline

A number of members of UCL staff are using GitHub classroom for projects. This provides a convenient way for tutors to distribute starter code in a repo but more crucially ensures that tutors and PGTAs have access to the student repositories for the purposes of supporting students and marking code.

GitHub Classroom supports group and individual projects, however there is currently no way to:

1. Reconcile a GitHub account to a specific UCL student. Students can create any username on GitHub and are not required to use their UCL email address. This leads to two potential issues:
   1. Staff do not know which students’ code they are reviewing
   2. There is a risk of plagiarism/contract cheating as it is harder to verify that the code was written by the actual student rather than someone else
2. Reconcile groups on Moodle with groups on GitHub Classroom. Students typically join a group on Moodle and this is used for group coursework submissions. There is no way to synchronise this with GitHub Classroom (there is a Moodle LTI to GitHub Classroom however it doesn’t support this feature and the LTI is not currently fully enabled at UCL). Instead one student from each group has to create the group in GitHub Classroom and then the other students select the group to join it. There is potential for students to select and join the wrong group.

The aim of this project is to develop an open source solution that allows staff to reconcile GitHub accounts to students registered on a given course.

The process that is used for COMP0019 is understood to be as follows:

* Student creates GitHub account (and uses any email address or arbitrary account name)
* Student creates an assignment in GitHub classroom.
* A server generates random 256-bit token is generated for all students on the course. The list of email addresses is somehow uploaded by the course tutor based on csv format file that is extracted from Moodle or Portico. The tokens are emailed to students using the email address from the course.
* Student responds with their token, email address and GitHub account (you would need to work how how to do this, I think in COMP0019 they upload a file to their GitHub classroom repo with their email address and token and there is a process triggered by webhook that runs on their server app that handles the authentication process)
* The response with the token/email pair is checked for valid token and then maintains a record of the GitHub name to student UCL email. This record can be subsequently used to provide tutors with identities of students and is also used in the auto grading process (a separate feature).

You are not obliged to adopt this workflow. If you can identify a different process that achieves the same result please do so.

The second aspect is to provide a way to also authenticate whether someone should be in a given GitHub Classroom group assignment or not by reconciling their GitHub ID with a list of groups extracted from Moodle. You will need to investigate whether this is possible.

In the first instance the target audience is course tutors and PGTAs at UCL, though in principle it should be usable by similar staff in other institutions.

Ideally the developed solution should be implemented on UCL infrastructure with a test solution established for COMP0035. This would be contingent on TSG being able to give access to a suitable server, this might not be possible in the timescales, you would need to contact them to find out.

# Suggested research/preparation

* Read documentation to understand [how GitHub Classroom works](https://docs.github.com/en/free-pro-team@latest/education/manage-coursework-with-github-classroom/teach-with-github-classroom).
* Read documentation to understand how [GitHub webhooks work](https://docs.github.com/en/free-pro-team@latest/developers/webhooks-and-events/about-webhooks).
* [Contact TSG](https://tsg.cs.ucl.ac.uk/contact-us/) to find out if it would be possible to host a web server that would be used by UCL students and would have a webhook to GitHub. I believe they do this for COMP0019 but they may not wish to, or may not be able to, do this for a student project. It would be better to find out sooner rather than later if this is an option or not.
* Research [Python/Flask](https://flask.palletsprojects.com/en/1.1.x/), if you haven’t used it before then you might want to look at [Miguel Grinberg’s mega tutorial](https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world). Consider the packages that might be useful. This is not an exhaustive list but possibly Flask-SQLAlchemy (if you use a database), Flask-Mail (for email), there also seem to be numerous libraries to support Flask with GitHub webhooks.
* Consider the workflow and identify what could be done given the technologies involved.
* Set up your dev environment and check whether you can get access to the relevant components e.g. can you access my GitHub Classroom? Do you have a GitHub account? Do you need to register in any way with GitHub to develop using webhooks? Install Python/Flask.
* If you wish to carry out a more academic (rather than purely technical) research into the context of the project then consider reviewing [papers that outline the problem of contract cheating or plagiarism in higher education programming assignments](https://scholar.google.co.uk/scholar?q=contract+cheating+in+higher+education+programming+assignments&hl=en&as_sdt=0&as_vis=1&oi=scholart), and consider any potential solutions that are suggested.

# Requirements

## Functional

1. MUST: Allow staff to provide a csv or txt file with student email addresses.
2. MUST: Provide a way to authenticate between student GitHub and UCL email address
3. SHOULD: Provide a way to authenticate that a student in a group in GitHub classroom matches a student in a group extracted from Moodle.
4. COULD: Provide the ability for coding assignments issued in GitHub classroom to be auto graded using unit tests.
5. COULD: Provide a way for coding assignments in GitHub Classroom to be checked for potential [code similarity using MOSS](https://gist.github.com/danielmai/9162349).

## Non-functional

1. SHOULD: Be supported by automated unit tests.
2. COULD: Configure GitHub Actions to run the unit tests when the code is changed.
3. SHOULD: Deploy a version of the solution for COMP0035 which should be hosted within UCL infrastructure. Contact TSG though note they may not be able to support this in the timescales.
4. MUST: Be developed in Python, preferably using the Flask framework. I would like to maintain and extend the solution and am most familiar with this technology.
5. MUST: Be open source with a suitable open source license.
6. MUST: Be documented so that any member of staff at UCL could implement the solution for their own course. Ideally the code will be in GitHub with an appropriate README to document the implementation.