## PROJECT OVERVIEW

Our goal is to design and implement an application that detects plagiarism on coding assignments written in Python. The application will be web based and use a database to store results and user information. The primary users will be Instructors and Administrator. It will require Instructor GitHub credentials and work with GitHub repositories.

## FUNCTIONALITY

The user will be able to go to the website where the application is hosted and register an account. During registration the user will provide his/her GitHub credentials that has access to all student assignment repositories. Once registered and logged in the user will have to set up a course by providing a name and the GitHub repo URL to that course. Once the course is created the user will be able view the course and add assignments to it. Assignments will be added by giving a generic name and also the exact name of the of the assignment directory as it is show on GitHub. Once an assignment is created, the user will be able to hit "Generate Results" button after all the students have completed and pushed their assignments onto GitHub (all generated results will be automatically saved onto a database for future access). A completion bar will appear until generation is done and once results have been generated the user will be able to click on s"View Results" where they will be able to see the results of that specific assignment. The results page will display a list of student couples color coded in red, orange, or yellow depending on the likeliness for plagiarism between the couple. Clicking on a result will bring you to a page that displays a comparison between the assignments, have an option to export the comparison, and an option to view student names (up until this point all student names will be hidden). If the user is an administrator, they will only have access to a list of accounts and will be able to delete any of those accounts.

## TECHNOLOGIES

The algorithm will be used to detect plagiarism on Python code. We plan on using Java to handle the backend functionality of the application and JavaScript, HTML, and CSS for the front end. The application will use MongoDB for storage and it will be deployed on Heroku in order to be easily accessible to all users from any PC.

## TIMELINE & WORK MANAGEMENT

The project will be organized using agile methodologies and will be divided into three phases that consist of A - gathering requirements and creating a wire-frame, B - designing the application, and C - implementing algorithms. A test-driven approach will be used for all programming phases and additional integration testing will be created before each phase is complete. Work will be divided and managed using Jira and Confluence and communication using Slack. The team will meet up 2-3 times a week for scrum meetings. GitHub will be used for version control in individual branches. Once the code is reviewed by all team members it will be pushed to the master branch. For more difficult tasks, pair programming will be required.