Team – 104
Plagiarism Detector
Plan Document

Language for detection: Python

Plan:

1. Objective:

A plagiarism detecting web-based system which takes the source codes of multiple students as input, detect the similarity of these codes and decide whether plagiarism exists, and provides detail report about the similarity for users to review and print.

2. Users and personas of the project:

Professors and TAs in universities who want to detect the code-plagiarism among students.

Professors and TAs are different type of users. Professors can configure the detecting details of the system (like threshold of plagiarism, the types of plagiarism to detect…), while TAs can only upload the files of two students, detecting the similarity between codes and reviewing the report.

3. Implementation details:

1) Process mode: use agile development process, iterate and release often, react to changes.
2) Types of plagiarism to detect: moving functions or methods to another location in the same file, breaking functions into separate files, renaming variables, extracting sequences of statements into methods, sharing same comments.
3) Algorithm we might use: AST, token
4) Back-end: Java
5) Front-end: Node.js, React, Bootstrap
6) Version control: Git
7) Version control host: GitHub

4. Schedule:

1) Requirement Analysis & prototyping: 1 week
2) Planning: 1 week
3) Modeling: 1week
4) Construction: 2 weeks
5) Deployment: 1 week
6) Take feed-back and start other iterations: the rest of the semester

5. Division of team work:

1) All team members should participate in planning and modeling process, fully understand the functionalities of the system.
2) At back-end side, each member focus on specific models and connect each other with interfaces.
3) Other tasks like UI design, testing… are assigned depends on the different skills of members.