Plan of Action

Target Language
We've selected Python as the target language to be tested on. At present, we've identified the library ANTLRv4 to be a powerful parser with support for different grammars. The motivation behind using this library is because we can later extend the same plagiarism system for any other language once we have the necessary AST data structures defined.

Frameworks
We will be using REACTJS for frontend development to implement upon the wireframes attached to this submission. The declarative nature of ReactJS suited our Agile development plan and we figured it'd be the best way to get our prototype to completion.

Java will be used as the backend system to implement the plagiarism detection mechanisms. We will have a REST based API hosted by Java. No backend database system is being used as per the use cases. We will however use the underlying file storage system to store files and generate the reports.

Ops/Testing
The application will be in the form a WAR file. It will be tested in production by deploying the WAR file to AWS. The staging environment will be hosted in each developers own personal machine.

Plagiarism Detector mechanisms

The following plagiarism detection mechanisms are the bare minimum being targeted:
1. Check for existing files in our own repository. 2. Detect change of variables. 3. Detect direct copy and paste.
4. Detect whitespace manipulated code

Schedule

We have 8 weeks until the start of the code reviews. Our plan is to complete development by 6 weeks and then give upto 2 weeks for finishing up with the backlog and the remaining of the functional and integration test cases. Development will include the writing of unit test cases by default.
The macroplan for the following 6 weeks of development is as follows. These tasks will further be subdivided into microtasks and be posted on Github issues.

Week 1:
Frontend: REACT templates for Use Case 1 and 2
Backend: Java setup API framework/parse Python code using ANTLRv4
Ops/Testing: Setup Elastic Beanstalk environment/Deploy WAR file.

Week 2:
Frontend: REACT templates for Use Case 3 and 4
Backend: Java implement upload assignment/store assignment
Ops/Testing: Test new Upload API

Week 3:
Frontend: REACT templates for Use Case 5 and 6
Backend: Java implement plagiarism detection method 1 with unit tests.
Ops/Testing: Test new UploadAndDetect API. Check if files are being created in the system. Create a cron job to delete the files in the system.

Week 4:
Frontend: REACT templates for Use case 7 and 8
Backend: Java implement download/flag the report
Ops/Testing: Test Download/flagging the report and check if the file is of the right format.

Week 5(Spring break: Reduced Work hours):
Frontend: REACT hardening for mobile and PC
Backend: Java implement plagiarism detection method 2 with unit tests.
Ops/Testing: Test method 2

Week 6:
Frontend: REACT hardening for mobile and PC
Backend: Java implement Plagiarism detection method 3 and 4 with unit tests.
Ops/Testing: Test method 3 and 4

Week 7:
Frontend: REACT tests with JEST/Backlog
Backend: Backlog
Ops/Testing: Functional and Integration testing/Backlog

Week 8:
Frontend: REACT tests with JEST/Backlog
Backend: Backlog
Ops/Testing: Functional and Integration testing/Backlog

Week 9-10:
Code Reviews