Team One, at the request of their client (Professor Frank Tip), will create plagiarism detection software. Over the course of three phases, the team will develop an interface for instructors, teacher assistants, and administrators to oversee the detection of plagiarism. In this summary of the team’s plan, the reader will find the expected input for the project, an explanation of the project’s interface, expected users of this project, and the underlying plagiarism detection methods that will be employed.

While this project could be expanded in the future, the scope of this iteration is constrained as described below. Firstly, this project is intended to be used to detect plagiarism in programs written in Java. Secondly only single file programs will be considered for plagiarism at this time. Thirdly, comments will not be taken into consideration when detecting plagiarism.

Team One’s plagiarism detection software will be written with a combination of Java, HTML, and JavaScript. The portion of the project that will detect plagiarism and manage underlying data storage and retrieval will be written in Java. The project’s data will be stored in a SQLite database. Users of this project will interact with this project by means of a web interface that will require a secure logon.

This project has three intended types of users: instructors, teaching assistants (TAs), and administrators (admin). Instructors can create, modify, delete courses. Instructors can also assign and remove TAs from his courses. Instructors can create, modify, and delete assignments from his courses. Both instructors and TAs can upload assignment submissions to assignments to courses that they are associated. Instructors and TAs can also run plagiarism detection against the submissions for assignments of courses to which they are associated. Instructors and TAs can also review the results of their courses’ plagiarism detection. Admin users handle the creation and deletion of users of this application.

The underlying functionality of this project relies on the process used to detect plagiarism between any two assignments submissions. Team One’s approach for detecting plagiarism between any two files consists of three phases: preprocessing, reordering, and then running a comparison against the restructured submissions. A mapping of original line numbers to post processed line numbers will be maintained in order to facilitate post-processing review by users. The processing and modification of submissions will be aided by use of the JavaParser2 toolset. Team One’s plagiarism detection algorithm borrows heavily from the Rojas Method1 cited below. The preprocessing involves the following: 1) removing all comments 2) standardizing indentation and spacing 3) changing string literals to a standard value 4) variable names are standardized 5) print statements are changed to a standard token 5) initialization and declaration statements are removed. The second step of the algorithm involves ordering functions based on character count. The final step of the algorithm involves comparing a submission against all other submissions for a given course assignment. This step will result in a score based on percent match between two given submissions. Submission pairs with a score above a given threshold will be flagged as potential instances of plagiarism.

Team One’s project is scheduled to be fully implemented by the end of the Northeastern’s Fall 2017 semester. The project will be implemented in three phases: Phase A: design, Phase B: documentation, and Phase C: implementation. All deliverables will be stored in Team One’s GitHub repository3.