Phase A Report

# Implementation Plan

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. Mockups of this interface can be found in the Wireframes section of this report.

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. A detailed description of these use cases can be found below in the Use Cases section of this report.

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.

# Use Cases

|  |  |
| --- | --- |
| **Use Case:** | Instructor Login |
| **Primary Actor:** | Instructor |
| **Goal in Context:** | The aim of the Instructor is to gain access to the plagiarism detector. |
| **Preconditions:** | 1. The Instructor should have an account. 2. The Instructor should have access to assignments. |
| **Trigger:** | Opening the web address in a browser |
| **Scenario:** | 1. The Instructor opens the web application. 2. When prompted for username and password, the Instructor provides a valid username and password. 3. Once login is successful, the Instructor is directed to the homepage.   The homepage will have a facility that enables the Instructor to upload the assignments. |
| **Exceptions:** | The username and password does not match.  The Instructor can try entering the login details again or cancel the operation. |
| **Priority:** | High, must be implemented. |
| **When available:** | Phase C |
| **Channel to actor:** | The Instructor interacts with the system through a web interface. |
| **Secondary Actor:** | Admin |
| **Channels to Secondary Actors:** | The Admin may interact with the system or other actors through a web interface. |
| **Open Issues:** | 1. Should we have a separate Admin user who will control the access to the detector? 2. Can we display any non-sensitive information related to the plagiarism detector/subject on the landing page such that login is not required? 3. Will there be any provisions to recreate password/username if the user has forgotten them? 4. Should any other fields be displayed along with the username and password fields? For example, a subject field. |

|  |  |
| --- | --- |
| **Use Case:** | Admin Creates Account for Instructors |
| **Primary Actor:** | Admin |
| **Goal in Context:** | The aim of the Admin is to grant the Teaching Assistants (TAs) access to the plagiarism detector. |
| **Preconditions:** | The Admin should be logged in. |
| **Trigger:** | The Admin clicks on Create Account  tab. |
| **Scenario:** | 1. The Admin clicks on Create Account tab. 2. The Admin is presented with a form to fill up.   The form requests additional information regarding the TA.   1. The Admin clicks on Create Account button. |
| **Exceptions:** | 1. The TA already has an account.   The Admin cancels the create account operation or might replace the TA’s current account details with the new account details.   1. Some fields are left blank.   The Admin is prompted to refill the blank fields. |
| **Priority:** | Medium, not urgent |
| **When available:** | Phase C |
| **Channel to actor:** | The Admin and the system communicate using web interface |
| **Secondary Actor:** | Teaching Assistant |
| **Channels to Secondary Actors:** | The TA interacts with the system through the web application. |
| **Open Issues:** | 1. What should be the fields in the Create Account form? |

|  |  |
| --- | --- |
| **Use Case:** | Examine Saved Plagiarism Detection Report |
| **Primary Actor:** | Instructor |
| **Goal in Context:** | The aim of the Instructor is to review reports previously generated by the  plagiarism detector. |
| **Preconditions:** | 1. The Instructor should have an account. 2. The instructor should be logged in. 3. The Instructor should have access to assignments. 4. The Instructor should have already generated a plagiarism report for at least one assignment |
| **Trigger:** | The Instructor clicks on the Review button on homepage. |
| **Scenario:** | 1. The Instructor clicks on the Review button. 2. The Instructor clicks on one of his courses. 3. The Instructor clicks on one the selected course’s assignments. 4. The Instructor clicks on one of the assignment’s previously generated reports. 5. The Instructor is presented with a list of instances of reported plagiarism. 6. The Instructor can click on an instance of plagiarism to review details. |
| **Exceptions:** | N/A |
| **Priority:** | Medium, not urgent |
| **When Available:** | Phase C |
| **Channel to Actor:** | The Instructor interacts with the system through a web interface. |
| **Secondary Actor:** | Teaching Assistant |
| **Channels to Secondary Actors:** | The TA interacts with the system through the web application. |
| **Open Issues:** | Should only courses and assignments with previously generated reports be navigable in the review section? If not, an option to generate a report for given course/assignment should be presented. |

# Wireframes