Phase A Report

**Aim**: Team One, over the course of three phases, will develop an interactive interface for instructors, teaching assistants, and administrators to oversee the detection of plagiarism. In this summary, 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.

**Constraints**: While this project could be expanded in the future, the scope of this iteration is constrained as described below.

* This project is intended to be used to detect plagiarism in programs written in Java.
* Only single file programs will be considered for plagiarism at this time.
* Comments will not be taken into consideration when detecting plagiarism.

**Logistics**: Our 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.

**Implementation Plan**: This project has three intended types of users:

* Instructors

Instructors can create, modify, delete courses and assignments related to the course. Instructors can also assign and remove TAs from his courses.

* Teaching assistants (TAs)

TAs can upload the assignment submissions to the courses that they are associated. After uploading submissions, they can run plagiarism detection on the submissions. They can review the results of their courses’ plagiarism detection. These actions can be performed by the Instructors as well.

* Administrators (admin).

Admin users handle the creation and deletion of users of this application.

Our approach for detecting plagiarism between any two files borrows heavily from the Rojas Method1 and consists of the three phases described below.

* Preprocessing: Removal of unnecessary components of a program, such as comments.
* Recording: Ordering functions based on character count.
* Running a comparison against the restructured submissions: Results 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.

JavaParser2 toolset is used for processing and modification of submissions.

**Schedule**: Our 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 our 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