

Phase A

Team 5

Members: Dujia Guo, Sahil Ghanghas, Sankar Gireesan Nair, Yash Shah

1)Plan and target programming language

We plan to target programs written in Java language for our plagiarism detection tool. Java is one of the most popular programming languages used by students in school coding projects. Thus, chances of plagiarism are very high in Java based projects. Hence, we decided to target Java projects. The tool is expected to find similarities between two projects and each of these project folders may have multiple files and sub-directories. Different kinds of simple and complicated transformations would have to be caught by the tool. We plan to create a web application and thus technologies such as HTML, CSS, JavaScript, jQuery etc. may find heavy use in our project. The core logic of plagiarism detection will be written entirely in Java. Backend of the application can be developed using Oracle SQL. The project will be implemented in phases. For first phase, we have thoroughly understood the requirements and brainstormed to come up with a set of use cases that will be implemented in later stage. The identified use cases can be seen elaborately in the next section of this document. We have also created mockups of some screens(UI) of our project and these will serve as a blue print when building the front end. We plan to understand the current competitive products in the market and study their various features so that we can come up with an optimized algorithm for plagiarism detection. We may use external libraries for the purpose of parsing and generating regular expressions/Abstract Syntax Trees but we will ensure that the core logic that will reside in the heart of our project is fully developed by us. We may plan to apply additional machine learning techniques that can help us provide a better solution. For phase B, we would be focusing on the design of our application and thus UML diagrams will play an important role here and lay a stepping stone for actual coding to commence. We will be building the core software architecture and thus the interfaces in this phase. We will implement the system, perform testing and create miscellaneous documentation in phase C. Code reviews and peer reviews in between the implementation phase will serve as a good evaluation of our progress. We plan to build our system collaboratively such that all members have an equal stake in the code generation process along with making other software engineering work products. We will use NEU GitHub for source control and issue tracking and will make extensive use of other suggested tools for managing and delivering a successful project.

2)Use Cases

1)

Use Case:	Login/Signup
Primary Actor:	Faculty of school/college
Goal in Context:	Faculty can have a private account in the web application. This can also enable to view reports of tests performed by him earlier.
Preconditions:	It is necessary for the faculty to register.
Trigger:	A faculty decides to create an account to use a web based plagiarism detector.
Scenario:	Steps: <ol style="list-style-type: none">1. Faculty register in to the system2. Details of school/university is to be provided.3. Faculty need to verify his identity.4. Faculty can login in to the web app after his/her account is successfully registered.
Exceptions:	<ol style="list-style-type: none">1. A unique user name is required. System should prompt user to enter any other username if not unique.2. Register with an education email ID. There should not be any previous accounts linked to that email ID.3. Password should have lowercase, uppercase, numeric and special characters
Priority:	Medium priority, to be implemented after first pass as this use case is an independent functionality and a extra security layer.
When available:	Phase C of project(implementation).
Channel to actor:	A desktop/laptop with a modern browser and internet connection.
Secondary Actor:	System Administrator
Channels to Secondary Actors:	A desktop/laptop with a modern browser and internet connection.
Open Issues:	<ol style="list-style-type: none">1. How will we verify that the education email id is of an existent school/college?

2)

Use Case:	Upload Projects for Plagiarism detection
Primary Actor:	Faculty of school/college
Goal in Context:	To detect plagiarism in two Java based projects whose source folders are present in the faculty's local machine.
Preconditions:	It is necessary for the faculty to login to the system successfully in order to upload the project folders and start with the test.
Trigger:	A faculty decides to test plagiarism in two java based projects whose source folders are available in the faculty's local machine.
Scenario:	<p>Steps:</p> <ol style="list-style-type: none">1. Faculty logs in to the system2. System displays the main testing home page to the user3. User uploads first project folder from the local machine by clicking on the browse button4. User then uploads second project folder from the local machine by clicking on the browse button.5. Once both the projects(Java based) are successfully uploaded, user can start the detection test by clicking on the "Start Test" button.
Exceptions:	<ol style="list-style-type: none">1. Login credentials are incorrect or not recognized. In this case, user is asked to enter the correct credentials or Signup, without which he/she cannot proceed further.2. User doesn't upload one or more project folders or uploads blank folder and clicks "Start Test" button. In this case the user is asked to upload a non-empty folder.3. User uploads non-Java based project folders. In this case, system responds with a "Unknown source" message and ask users to upload a Java based project.

Priority:	High priority, to be implemented in first pass as other functionalities will require a successful functioning of this use case.
When available:	Phase C of project(implementation).
Channel to actor:	A desktop/laptop with a modern browser and internet connection.
Secondary Actor:	System Administrator
Channels to Secondary Actors:	A desktop/laptop with a modern browser and internet connection.
Open Issues:	<ol style="list-style-type: none"> 1. What would be the maximum size of project folder that we can upload in the system

3)

Use Case:	Perform Plagiarism detection test
Primary Actor:	Faculty of school/college
Goal in Context:	To detect plagiarism in two Java based projects whose source folders are present in the faculty's local machine.
Preconditions:	It is necessary to upload the project folders and start with the test.
Trigger:	A faculty decides to start the plagiarism after uploading the projects.
Scenario:	Steps: <ol style="list-style-type: none"> 1. User clicks of "Start Test" button after both the project folders have been successfully uploaded 2. User select pause to pause the test 3. User selects start after pause 4. User selects stop
Exceptions:	<ol style="list-style-type: none"> 1. Pause is selected, process stops until resumed again 2. Stop is selected, process stops and cannot be resumed, progress report till that point is saved
Priority:	High priority, to be implemented in first pass as other functionalities will require a successful functioning of this use case.
When available:	Phase C of project(implementation).
Channel to actor:	A desktop/laptop with a modern browser and internet connection.
Secondary Actor:	System Administrator

Channels to Secondary Actors:	A desktop/laptop with a modern browser and internet connection.
Open Issues:	1. Is there a way to save the process checkpoints to resume them later.

4)

Use Case:	View test reports
Primary Actor:	Faculty of school/college
Goal in Context:	To view detailed plagiarism reports
Preconditions:	Plagiarism process has been performed on both the projects
Trigger:	User finishes the plagiarism process after completion or stops it in between and decides to view the details of the test
Scenario:	Steps: 1. User observes tool panel 2. User clicks on "View Report" button for a particular test
Exceptions:	1. User won't see the "View report" option if no tests have been ever performed by him
Priority:	High priority, to be implemented in first pass as other functionalities will require a successful functioning of this use case.
When available:	Phase C of project(implementation).
Channel to actor:	A desktop/laptop with a modern browser and internet connection.
Secondary Actor:	System Administrator
Channels to Secondary Actors:	A desktop/laptop with a modern browser and internet connection.

5)

Use Case:	Download test report
Primary Actor:	Faculty of school/college
Goal in Context:	Saving test report after
Preconditions:	It is necessary to upload the project folders and start with the test. The test should be completed for report generation.

Trigger:	Faculty decides to save the report in his local machine
Scenario:	Steps: <ol style="list-style-type: none"> 1. User clicks on “Save report” button for a particular test 2. User selects name for the report file 3. User selects format type for test report 4. User selects directory to save the report 5. User selects save
Exceptions:	<ol style="list-style-type: none"> 1. User selects cancel, warning given for losing progress and download process will be terminated
Priority:	High priority, to be implemented in first pass as other functionalities will require a successful functioning of this use case.
When available:	Phase C of project(implementation).
Channel to actor:	A desktop/laptop with a modern browser and internet connection.
Secondary Actor:	System Administrator
Channels to Secondary Actors:	A desktop/laptop with a modern browser and internet connection.
Open Issues:	<ol style="list-style-type: none"> 1. What would be the different formats in which the user can save the report

3)Mock-ups of System’s user interface

These are very rough versions of some of the pages of our system and the functionalities we have in mind as of now. We will refine our designs and come up with a pleasant User interface during subsequent project stages.

1)Login/Register



The image shows a web browser window with the title "Moqzilla". The address bar contains the URL "http://moqups.com". The main content area displays a login form with the heading "Login". Below the heading are two input fields: "Username" and "Password". A "Login" button is positioned below the password field. At the bottom right of the form, there is a link that reads "New User? Register here".

Moqzilla

← → ↻ http://moqups.com

Login

Username

Password

Login

[New User? Register here](#)


Moozilla

← → ↻ <http://moqups.com>

Register

 Username

 Email

 Password

2)Main page

Moozilla

← → ↻ <http://moqups.com>

Welcome to Plagiarism Detector Homepage

Upload Java Project folder 1:

▼

Upload Java Project folder 2:

▼





3)Results page

Mozilla

← → ↻

http://moqups.com

Results

▼ File Name	▼ Files matched	▼ View Report/ Save PDF
Project 1/Parser.java	Project 2/MyParser.java	 
Project 1/Login.java	Project 2/Old-login.java Project 2/New-login.java	 

Exit

Logout