STATEMENT

Objective

Develop an application to detect plagiarism in student submissions of programming assignments by using various methods. The system should detect syntactic similarities and behavior-preserving transformations like splitting code into methods, renaming variables, etc. The application should detect similarities in programs written in **Java**.

Plan

Week	Plan
Week 1 **Development Environment**	1. Setup dev systems and establish process. 2. Setup Jenkins. 3. Setup basic web REST (Jersey) endpoint. 4. Setup DB and storage system (MySQL). 5. Database design and analysis.
Week 2 **Backend Logic**	1. Implement Backend logic engine with basic functionality: a. Changes in syntax b. AST 2. Setup basic user authentication system.
Week 3 **First increment Delivery** ***Deliverables:*** *Working product with AST and basic diff for 2 files.* *Working Web UI*	1. UI development (AngularJS/JS/HTML/CSS) 2. Integration
Week 4 **Logic Enhancement**	1. Functionality Enhancement 2. Winnowing or n-grams 3. ML Techniques
Week 5 **Fine tuning**	1. Finalize the logic and functionality 2. Develop endpoints for new data
Week 6 **Second (Final) Increment Delivery**	1. UI enhancement with new features 2. Final Integration
Week 7 **Final Presentations**	

References: https://theory.stanford.edu/~aiken/publications/papers/sigmod03.pdf

USE CASES

Case 1: Ability to view list of assignment submissions of all students	
Use Case	The Grader should be able to view list of all assignment submissions
Primary Actor	Grader
Goal in Context	To view different submissions and analyses them for plagiarism
Preconditions	There must be one or more submissions for a given assignment from students
Trigger	NA
Scenario	1. The grader logs in to the system. 2. System provides a list of assignments.
Exceptions	
Priority	High Priority
When available	First Increment
Channel to actor	A web interface where the grader can view the list of submissions
Secondary Actor	The students
Channels to Secondary Actors	The student submits his/her assignment or files using the student portal
Open Issues	

Case 2: Ability to select java files to analyze for plagiarism	
Use Case	The Grader should be able to select one java file to analyze for plagiarism
Primary Actor	Grader
Goal in Context	To detect whether the student has done plagiarism in this file or not
Preconditions	The project should be preloaded into the system and it has files to be selected for
Trigger	Grader chooses the student to see all his submissions
Scenario	1. The grader logs in to the system. 2. System provides a list of assignments. 3. He chooses an assignment or project to grade from a list. 4. System provides a list of all students who have submitted the assignment. 5. Grader chooses the student from the list. 6. Grader chooses the file to see the report.
Exceptions	Files might not load into the system.
Priority	Moderate priority, this is planned to be implemented in later increments
When available	Second Increment
Channel to actor	A web interface where the grader can view the file
Secondary Actor	The student whose assignment is being graded
Channels to Secondary Actors	The student submits his/her assignment or files using the student portal
Open Issues	Need to figure secondary alternatives in case the file does not load into the system

Case 3: View the overall plagiarism percentage of a student	
Use Case	Grader views an overall percentage for plagiarism in the student report page
Primary Actor	Grader
Goal in Context	To detect whether the student has conducted plagiarism
Preconditions	1. Grader has to be logged in to the system. 2. Grader chooses an assignment and a student from the list. 3. Student has submitted his/her assignment.
Trigger	The grader clicks on an assignment and a student from the list
Scenario	1. The grader logs in to the system. 2. System provides a list of assignments. 3. He chooses an assignment or project to grade from a list. 4. System provides a list of all students who have submitted the assignment. 5. Grader chooses the student from the list. 6. System provides a detailed student report with an overall percent and individual probability for different checks. 7. The detailed student report page will present the grader with an average probability score from all models.
Exceptions	If the student does not follow standard conventions while uploading the file, unknown exceptions may arise
Priority	High priority, this is one of the first checks that needs to be implemented
When available	First Increment
Channel to actor	A web interface that provides a detailed student report
Secondary Actor	A student who the grader is grading.
Channels to Secondary Actors	The student submits his/her assignment or files using the student portal
Open Issues	Detecting similarity gets more difficult as students come up with new methods while plagiarism.

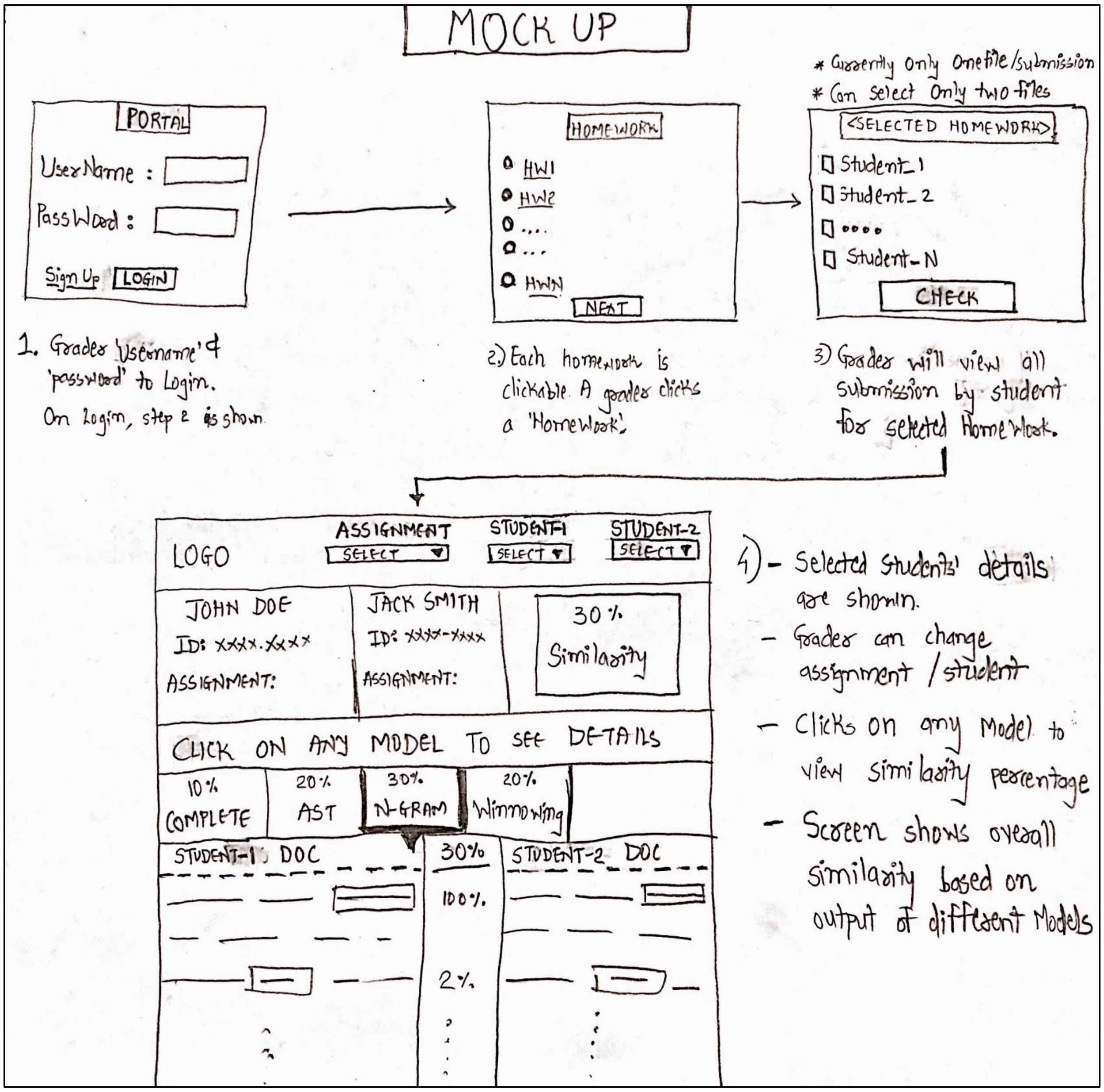
Case 4: View probability score from different plagiarism models	
Use Case	Grader sees the probability scores from different model for plagiarism in the student report page
Primary Actor	Grader
Goal in Context	Find behavior-preserved code to detect whether the student has done plagiarism by copying parts of the assignment
Preconditions	1. Grader has to be logged in to the system. 2. Grader chooses an assignment and a student from the list. 3. Student has submitted his/her assignment.
Trigger	The grader clicks on an assignment and a student from the list
Scenario	1. The grader logs in to the system. 2. System provides a list of assignments. 3. He chooses an assignment or project to grade from a list. 4. System provides a list of all students who have submitted the assignment. 5. Grader chooses the student from the list. 6. System provides a detailed student report with an overall percent and individual probability for different checks. 7. The detailed student report page will also present the probability score from multiple models like ASTs, winnowing, n-gram, etc.
Exceptions	If the student does not follow standard conventions while uploading the file, unknown exceptions may arise
Priority	Moderate priority, this is planned to be implemented in later increments
When available	Second Increment
Channel to actor	A web interface that provides a detailed student report that displays scores from different models.
Secondary Actor	A student who the grader is grading
Channels to Secondary Actors	The student submits his/her assignment or files using the student portal
Open Issues	ASTs should be converted into a form that can be easily compared. Choosing the right value of n to perform n-grams is challenging.

Case 5: View the similar documents side-by-side	
Use Case	Grader sees the similar documents side-by-side with the detected lines highlighted
Primary Actor	Grader
Goal in Context	To confirm whether a student has conducted plagiarism by looking at the similar lines
Preconditions	Grader must be logged in to the system. An assignment and student must be selected.
Trigger	Grader clicks on any of the similarity score in the student detail page
Scenario	1. The grader logs in to the system. 2. System provides a list of assignments. 3. He chooses an assignment or project to grade from a list. 4. System provides a list of all students who have submitted the assignment. 5. Grader chooses the student from the list. 6. System provides a detailed student report with an overall percent and individual probability for different checks. 7. The grader clicks on any of the scores provided in the page. 8. System displays the similar lines detected by that particular model side-by-side.
Exceptions	If any model does not provide a detailed explanation
Priority	Moderate priority, this is planned to be implemented in the first increment
When available	First Increment
Channel to actor	A popup window that displays the line comparisons side-by-side
Secondary Actor	NA
Channels to Secondary Actors	NA
Open Issues	Highlighting lines from models like AST will be challenging

Case 6: Save the generated report about a submission to local	
Use Case	Save the generated report about a submission
Primary Actor	Grader
Goal in Context	Allows the user to save a copy of the report generated for future use
Preconditions	Detailed analysis about an assignment submission(files/folders) has been generated by the system
Trigger	When the user clicks on "Save Report" button on the user interface
Scenario	1. The user views the report generated of a particular submission 2. If the user plans to save the report, he clicks on the 'Save Report' button on the user interface 3. On clicking the 'Save Report' button, the user is given options to select the format in which the report should be saved locally. 4. On selecting the format desired by the user, the file is downloaded and the user can view the report locally.
Exceptions	The user might have to retry downloading the file, if the download does not begin.
Priority	Least priority. The report need not be saved locally. For future retrieval, the report can always be generated by uploading the submissions to the system again.
When available	Third Increment
Channel to actor	The web interface provides a button 'Save Report'.
Secondary Actor	NA
Channels to Secondary Actors	NA
Open Issues	1. Time it will take to download the complete report 2. If the user does not select the format of the report

Case 7: Upload an assignment submission to the system	
Use Case	Upload an assignment submission to the system
Primary Actor	Students
Goal in Context	To allow the students to input submissions, which can be a file or multiple files or folders containing files, to the system in order to detect similarities between submissions.
Preconditions	The student should be able to login to the system.
Trigger	After the user clicks on the 'upload' button on the user interface.
Scenario	1. The student logs into the system. 2. Selects the files the student wants to upload and uploads them.
Exceptions	1. The student might upload a wrong file. 2.The student might have to re-upload the submission files, if the uploading fails the first time.
Priority	Highest Priority
When available	First increment
Channel to actor	The web interface of the system that provides the 'upload' button.
Secondary Actor	NA
Channels to Secondary Actors	NA
Open Issues	1. Time taken to upload the submissions. 2. If the submissions are zipped, then the files must be unzipped before running the algorithm over the project or assignment.

MOCK UP



* Currently only one file/submission
* Can select only two files

1. Grader 'Username' & 'password' to Login. On Login, step 2 is shown.

2) Each homework is clickable. A grader clicks a 'Homework'.

3) Grader will view all submission by student for selected Homework.

4) - Selected students' details are shown.
- Grader can change assignment / student
- Clicks on any Model to view similarity percentage
- Screen shows overall similarity based on output of different Models