CS5500 Managing Software Development

Northeastern University

Spring 2018 - Prof. Annunziato

# Overview

The project should be full-on in development mode. You should be full-on into Jira (http://cs5500-jira.ccs.neu.edu) to manage your sprints and backlogs. All system artifacts should be on your team's repository on CCIS github (<http://github.ccs.neu.edu>) and you should be using maven to control building and testing. If you haven't deployed Jenkins, you will be using it as your integration tool and enforcing policy about good code via SonarQube and Sonarlint locally.

# Reminder on the Grading (Repeat)

Phase C will be graded as the accumulation of grades over each sprint. This is the second sprint. Your grade in a sprint is an assessment of the team's overall progress (40%) and your individual contribution (60%). Remember, you will be graded on functionality achieved, quality, style, and documentation. The night before your sprint review, you must complete a peer review, through which you will evaluate the work of your team-mates. The peer review is for the teaching team and private to the teaching team. Ultimately, your grade will be based on the evidence we nd in your Github repo, your Jira project, what we can observe of your system, the feedback from your peers, and the discussion in the sprint review. Completing the peer review is required. Failure to complete the review will mean a 0 for your individual score. Meeting the expectations denes B work. If your team does this workthe functionality works, demonstrates quality, and ensures maintainability, and the team is working wellyou earn a B. Hit on more than these points, your grade goes up. Miss on these points and your grade goes down. You cannot lose on stretch work; but if you miss on it somehow (doesn't quite work, didn't do a good job of the work), it won't be as valuable to your grade. We've added some ideas for stretch goals. You can add your own. A stretch must advance the project either by adding useful functionality or imp

# Sprint 2 Expectations

### Base Expectations (90%)

Plagiarism not working from UI

Test cases

Comments.

1. Faculty use a faculty UI, Students use a student UI, Admin use an admin UI. Not completed.
2. At least 4 users exist in the a database. They can be hard coded for now. Two users in student role, one user in faculty role and one user in admin role. Depending who logs in, the correct UI is displayed. Next sprint Admin can change user roles. Done
3. Faculty and student UI display at least one semester, with one course, with one assignment. Faculty can CRUD semester, courses and assignments next sprint.
4. Faculty user can kickoff a comparison for the single assignment. ( Pending )
5. Plagiarism threshold can be hardcoded for now, e.g., 75%, but faculty should be able to set it next sprint. (Pending )
6. Faculty user can view a report for a single submission from a single pair of students. Next sprint any faculty should be able to login and view multiple submissions from multiple students. (Pending ) Linked aws.
7. Submission displays score next to its label and is highlighted if above a threshold. (Pending)
8. The report should include a similarity score, students, and files in question. Files can be viewed side by side with similar lines highlighted . (Yes but now on AWS)
9. A pair of students can submit multiple files, or a zip file, or a directory, or a github link to a directory for the single assignment. Next sprint you will generalize being able to select from any course/assignmet. In this sprint, the identities of the students can be hard coded in the database.(Done )
10. Performs the comparison against multiple files used in two submissions for the single assingment (Done)
11. Admin sees list of users. Next sprint admin can change the role of users, CRUD users. (Done local)

### Stretch (10%)

1. Performs the comparison against multiple submissions for a single assignment, for a single course. (Done)
2. Performs comparisons for any new submission from same two students, e.g., incremental comparisons (Done)
3. System sends a message (email, text) when plagiarism has been detected (Not done)

### **Base Expectations (85pts.)**

1. (10pts.) Faculty use a faculty UI, Students use a student UI, Admin use an admin UI. 7points Admin UI was left but backend was available.
2. (10pts.) At least 4 users exist in the a database. They can be hard coded for now. Two users in student role, one user in faculty role and one user in admin role. Depending who logs in, the correct UI is displayed. Next sprint Admin can change user roles. 10pts ->10points
3. (10pts.) Faculty and student UI display at least one semester, with one course, with one assignment. Courses and assignments can be hardcoded in this sprint. =-> 10pts
4. (5pts.) Faculty user can kickoff a comparison for the single assignment. -> 3 pts ( local )
5. (5pts.) Plagiarism threshold can be hardcoded for now, e.g., 75%, but faculty should be able to set it next sprint. -> 5pts
6. (5pts.) Faculty user can view a report for a single submission from a single pair of students. -> 3 pts (local)
7. (5pts.) Submission displays score next to its label and is highlighted if above a threshold. -> 3(local)
8. (10pts.) The report should include a similarity score, students, and files in question. 7pts(show in localhost). 7pts
9. (10pts.) A pair of students can submit multiple files, or a zip file, or a directory, or a github link to a directory for the single assignment. 7pts (Hardcoded but works)
10. (10pts.) Performs the comparison against multiple files used in two submissions for the single assignment. 7 pts (comparison is done at backend shown on localhost )
11. (5pts.) Admin sees list of users. Next sprint admin can change the role of users, CRUD users. ->0

### **Stretch (15pts)**

1. (7pts) Performs the comparison against multiple submissions for a single assignment, for a single course. 7 pts (done at backend)
2. (3pts.) Performs comparisons for any new submission from same two students, e.g., incremental comparisons. 3pts
3. (3pts.) System sends a message (email, text) when plagiarism has been detected.
4. (2pts.) Files can be viewed side by side with similar lines highlighted. 2pts JPlag

Sprint 3

1. 7pts Submissions are compared incrementally to avoid long waits
2. 5pts Available on local : Generalize being able to select from any course/assignment. In this sprint, the identities of the students ***can not*** be hard coded in the database.