Northeastern University
CS5500 – Managing Software Development
Fall 2018, Professors Jump, Katz, and Weintraub

Sprint 2 Expectations

1 Overview

The last sprint was about getting started. The emphasis was on establishing the process and tooling to support it, and getting to know the legacy code. Now it's time to start putting some new features in place.

Depending on the team, product management's ask for individual-focused or group-focused messages are at different places for having accomplished the feature. This sprint, the request is to develop capabilities so users can communicate with specific users or groups.

2 Expectations

2.1 Functionality

1. Getting the legacy code base running if that's still an issue.
2. Adding a notion of user and groups of users to the system.
   - The entities should be persistent. It's the team's call what to use for persistence.
   - Provide basic user/group management: Create, Read, Update, and Delete capabilities.
   - Stretch (medium): Adding login with user name and password. The storage must encrypt passwords.
3. Directing messages to individuals and groups. Messages that are replies to messages sent to a group go to all other members of the group.
4. Stretch (medium to large): Add message persistence to messages sent to individuals and groups.
5. Stretch (large): add MIME types to messages. MIME [**M**ulti-purpose **I**nternet **M**ail **E**xtensions] are a standard way of identifying the type of a file. It's used in messaging systems so senders can transfer multiple types of files without needing special mechanisms, and clients can first identify and then deal with these files as appropriate.

2.2 Environment

1. The system is deployed to a cloud environment such as Amazon AWS or Mass OpenCloud. You are encouraged to find a free service. You must provide administrator access to the project executives.[1]
2. Stretch (small): team is using smart commits in `git`.
3. Stretch (small): Jenkins should inform the team of failure either in Slack.
4. Stretch (small): Github should inform the team of PRs via slack.

[1] This means your TA.

3 Reminders

3.1 Guidance

You own the system and the path. So achieving the goals will involve extending or creating additional classes and methods as needed. You will need to make many design decisions as well as decisions about what external libraries. You may be allowed to use external libraries for specific tasks, but you should to expect that the vast bulk of the system will be written by the team. This code should be written in Java. Please contact the instructors if you have any questions.

You can add your own stretch goals. A stretch must advance the project either by adding useful functionality or improving the process. If the stretch work isn't helping the project or the team, it will not be counted.

Please ensure that your code is fully self-contained (e.g., all libraries on which your system depends must be included in the distribution), documented properly, and accompanied by tests and installation instructions so that the TA's can install your system and run your tests without problems. You may expect that your TA will run your system on a personal machine or use some cloud service such as AWS. Stay in touch. While usually this means daily stand-ups in person, that's not feasible in a class like this. Instead, you should conduct daily *slack-ups* when you aren't meeting in person.[2] Even if you don't plan to do anything on the project that day, you should say so on slack. You should agree as a team that everyone will post on your project channel three things (you can add to this list if your team agrees more is needed):

1. What you did the prior day.
2. What you're planning to do that day.
3. If you are stuck on something or blocked by someone else on the team.

Don't forget to do `Team-mates`.

Attendance at the sprint reviews is required. Only excused absences will be given opportunity to make up the review with the TA, which is a one-on-one meeting at the first possible chance. Missing a sprint review means your sprint score may be 0. See the Late Work policy on the website if you have doubts about what will be excused (`https://course.ccs.neu.edu/cs5500/pol.html`).

Here are the technical requirements:

1. You must use Jira (`http://jira.ccs.neu.edu`) to manage your backlogs and store system artifacts on your team's repository on CCIS github (`http://github.ccs.neu.edu`).
2. You must use `Jenkins` for CI [Continuous Integration] unless you receive permission to use another tool. You **must not** alter the `Jenkinsfile` without Alex Grob's consent.
3. You must develop code using `Java` using at least JavaSE 1.8.
4. You must use `junit` for unit testing unless you receive permission to use another tool.
5. You must use `Apache Maven` to manage the builds for your project.

[2] Ideally this should be a daily thing. Every other day is acceptable, but every day is better. Less often will lead to issues.

3.2 Grading

The project will be graded as the accumulation of grades over each sprint and the final presentation.

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 teammates. The peer review is for the teaching team and private to the teaching team. Ultimately, your grade will be based on the evidence we find 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 minimal expectations defines the minimal passing grade. If your team does this work—the functionality works, demonstrates quality, and ensures maintainability, and the team is working well—you 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.