



## Sprint 5 Team Report

**Team: 108**

**Repo URL:** <https://github.ccs.neu.edu/cs5500/team-108-SP19>

**Reviewer:** Michael Weintraub

**Score:** 90

### Comments

The overall presentation was solid. The video didn't do your work any justice. The rain was extra noise. The recording sounded like you were in a deep tunnel. You would have been better off using a another, less exposed room.

The video made some unsubstantiated claims. For example, there was a claim that no SQL injection was possible. That's fine. However, true or not, the statement was not justified with technology or design.

The interweaving of demo into the slides was a good decision. It was very effective.

Seeing so much discussion dedicated to password development was disappointing. In class, the client was very clear that password protection was not important to the early development goals. The client had mentioned he preferred using out-sourced validations (e.g. Facebook or LinkedIn) to spending time developing this in-house. While the team presented the work it did, it is disappointing to see a focus on something client expressly said wasn't valuable. Now this all said, the work seemed very solid.

Client and users are used interchangeably. In this presentation, I'm the client and end users are something else. Perhaps the team is using system client and user interchangeably?

The pseudo-master idea is a nice idea, but it seems redundant with master given the rules put into the build process. The presentation missed an opportunity to explain why it chose to introduce this extra step, which appears to be so peer review could be instituted and additional rules like no code smells could be applied. For this class, changing the process to improve quality would have been an awesome point to make and earned the presentation higher marks. For the future, you should be encouraged to add these rules to the dev pipeline. For example, if you wanted to enforce no smells, you should have asked to change your Jenkins pipeline to have this rule.

Like the refactoring example!

Good process and teamwork section. Very good discussion of shortcomings/issues - appreciate the honesty.

Fair job. Managed to do a good job. The last part wasn't easy to follow.

The system demo was very good.