

# 6.170 Project 4

TEAM REFLECTION BY ANJI REN, DYLAN JOSS, LILY SEROPIAN, VICKY GONG

## What Went Well

Despite the disruption of our ability to work as a whole group in person due to frequent interviews, Thanksgiving, and in general, members of our team flying in and out, what went really well was how much we focused on making our design very good before we began our application. We really prioritized the quality of our design, which made decisions later on very straightforward. In-depth discussion and awareness of edge cases led to our very solidified understanding of what were the purposes and concepts, complexity and scope, of our target problem and problem solution.

As a few unforeseen problems did surface when we started coding, we were extremely realistic about how much time various proposed solutions to the problems would take, and reduced the complexity and scope without sacrificing functionality as needed. In the latter half our project, we rarely met as an entire team at all, but because we stayed engaged in the project and available online, working remotely was very effective. We also got along well as a group, which made teamwork also very pleasant, and allowed us to easily communicate and exchange labor with our teammates as needed.

## What Could Have Gone Better

- misunderstanding about MVP requirements; we implemented all of the functionality of our MVP but didn't have that functionality accessible through the UI in the manner the TAs expected
- testing earlier would've pointed out bugs and reduced the need to scramble to write tests at the last minute
- was a \$@&%-ton of work; perhaps a project of smaller scope would have been better
- working up to deadline on Tuesdays
- better communication about handling disagreements on technical implementation
- more strict adherence to the coding conventions agreed upon
- consistency of variable/reference names; e.g. the inconsistent usage of 'org', 'orgName', and 'organization' as names for the same concept led to a lot of unnecessary bugs
- when an error/bug was found, being more concrete/clear about the sequence of actions taken that led to the bug so as to help the person who wrote the code debug

### Error handling and tests

For one, the way errors and testing could have been handled better. We did not include error handling and testing in our MVP because of our lack of time, however, because of this, error handling was unclear. What we should have done is discussed and planned how errors and testing would be handled, and have each person write errors and testing for the parts they coded based on a prior error. These would all be based on the testing convention that we should have defined prior. Moreover, because tests should have been written earlier as it would have made it more clear what exact behaviors/errors we wanted when a bad request occurred.