Reflection (Final)

Team Hacktrack: George Ezenna, Kairat Ashim, Eric Manzi, Favyen Bastani

Team reflection

As a team, we feel that we achieved what we set out to do in this project. Although our finished product is not entirely what we envisioned, we are pleased with the results of our efforts. We've all improved our code quality and professionalism and we got an opportunity to hone our collaboration skills. We feel more prepared to work on large-scale apps in a team setting.

Overall, progress was a lot smoother after the MVP because we set team deadlines and discussed low-level details prior to implementation, among other things.

What went well?

- After completing the MVP, we decided that it would be highly beneficial for the team to set and follow our own deadlines that preceded those set by the class. We succeeded in setting these deadlines and followed through for the most part. We all agree that this was really helpful because all the implementation that followed was block-free. In no case was anyone's progress blocked by someone else's work, or lack thereof.
- For features that shared developers, we tried as much as possible to maximise communication between the developers so that lower-level design decisions could be made instead of only high-level decisions. These smoothed out the implementation process and minimized code duplication.
- We also saw some benefits of the peer reviews. For instance, after the MVP, one team member was critiqued in a peer review for being hard to reach and completing his tasks on the last minute. This time around, said team member did an exponentially better job communicating with the group and delivered his work well before the deadline.

What could have been improved? What would we do differently in the future?

- Using pull request system and reviewing code before merging it into the master branch may have helped to reduce errors when merging modifications. That said, it's not clear if the payoff would be worth it for such a small project considering the energy needed to review lots of pull requests.

- Use a better communication format than Facebook group chat for scheduling meetings and presentation times; there were a couple of instances where we made last-minute changes. In the future we would pick a scheduling app like Doodle to facilitate this.
- Sometimes it was difficult to figure out how to use some functionality implemented by other team members, like the e-mail templates or transitioning modals. Adding some documentation when adding a feature that could be used in other parts of the project would be helpful.
- Although we agreed on basic code styling, different parts of the codebase still had big differences, e.g. some places used IDs and Model.populate to fetch data while in other places we included a unique name and called a static method on the schema to fetch it. Extended the agreed-upon style to issues like this could make things more consistent.

Summary of key lessons learned:

- Unit testing is useful! We discovered subtle bugs like returning to the callback before performing an action from the tests.
- Javascript is an awful language! With dynamic, weak typing, more work needs to be spent on unit tests and many types of bugs are difficult to pinpoint.
- Communication is essential when working on a large group project. Interfaces need to be clearly defined in advance, and documentation is needed so everyone working on the project is aware of the main goals of various modules and the basic architecture. A good communication format goes a long way.
- Team deadlines work magic!
- Peer reviews encourage involvement and responsibility.