

Reflection (MVP)

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

Team reflection

Our project was to set out to build MIT Hacktrack, a web app that allows members of the MIT community to share what they are hacking on and discover what their peers are working on. Initially the project didn't seem to be intrinsically complicated enough to be a final project. Much of what we envisioned was login and posting and commenting associated with users which wasn't too far from what we did on the Fritter app. However from discussing the design we discovered intricacies such as what a post is (a project or also a comment) and what can users follow (other users or just projects). We also decided to add richer functionality as opposed to just posting text to include posting rich content such as images and video clips for a better experience and tagging projects by topics and adding search and filtering functionality to discover projects. Through these discussions and additional features we have a better idea of what we are trying to build and the complexities that come along with it.

Building the MVP was a chance for all of us to get more practice building a non-trivial web application in a team setting. We've all taken 6.005 and some of us have taken group project classes like 6.813, 6.S198 or done internships/urops where we had to collaborate with others on designing an application. However, this project introduced new components of collaboration-based design that we had not encountered before such as the teamwork plan, agendas, progress reports and the design document which gave us all a chance to develop our professionalism and collaboration skills.

What went well?

- Agendas and meeting minutes made our mentor meetings more productive and organized which helped us get the most out of them. The agenda helped us keep track of the topics to be discussed and the meeting minutes let us keep a record of the decisions that were made and the feedback from the TAs, which we referred to and relied on after the meeting.
- Making a unanimous design decision is always the hardest part of collaborated design. We decided in our team contract that conflicts on whether a feature was to be included were to be resolved by majority consensus. This helped us to really speed up the design process since there almost always were features which still had proponents and opponents after thoroughly debating their usefulness.

- We wouldn't have been able to make these decisions as quickly as we did if communication in the team wasn't fluent. We noticed that communication was mostly fluent during in-person meetings. It is really easy for things to be lost in translation and difficult to resolve conflicts and confusion remotely. Luckily (and thanks to our facebook group chat and video calling), we were able to put together our MVP with no big issues despite not being able to meet as often as we would have liked. Moving forward however won't be as easy since our new features will be more complex and require more collaboration.

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

- Discussing the high level design before jumping into implementation was very useful as it helped us decide what to implement and assign different tasks to everyone. While we were able to discuss high level implementation decisions like which templating engine to use and what our schema should have, we didn't get a chance to discuss low-level implementation details like which Objects should implement which methods so we ended up with duplicate code which could have been easily avoided by discussing some of these lower-level details.
- Up until now the only deadlines we have been following as a team have been those defined by 6.170 Guidelines. As a team we haven't really set any internal deadlines for ourselves. Even though we have generally been able to develop simultaneously by dividing the work such that no one's tasks are blocking on anyone else, we did come across one instance where someone was blocked. This could have been mitigated if we had set deadlines. We plan to start getting into the habit of setting and following our own deadlines since there are few 6.170 deadlines left before the end of the project.

Summary of key lessons learned:

- Agendas and minutes greatly improve productivity of mentor meetings
- Going with majority vote helps speed up the design process
- Communication is key, and is most fluent during in-person meetings
- It's important to talk about low-level, not just high-level implementation details
- Setting team deadlines mitigates the effects of potential blocks by teammates