

Mentor Check-In #1 & #2

Fri, Nov 12 and Dec 3 2021

Commit Document: `meeting-agenda` / `meeting-agenda-2` (by noon before the meeting)

Commit Document: `meeting-notes` / `meeting-notes-2` (by midnight after the meeting)

Each team will be assigned a teaching assistant as a mentor. You will have two 30 minute meetings, one after [Project Pitches](#), and one after the [MVP Presentations](#). We won't be able to fit all the teams during the lecture time, so your mentor will reach out to you for scheduling. All team members are expected to attend the mentoring meetings (unless there are extenuating circumstances that you have communicated with your mentor).

It is important that you come to the meeting prepared so that the meeting runs efficiently. Therefore, your team will be required to prepare an agenda before the meeting. The agenda should include a list of points you wish to discuss with your team mentor. The agenda should also include a progress report summarizing all progress up until the meeting. It should include:

- achieved & missed milestones from your project plan
- any difficulties that you encountered
- any changes made to your design or project plan

During the meeting, one team member should take brief meeting notes, which should include a summary of discussions, advice from your mentor, new decisions, and any changes made to your plan & milestones moving forward.

You might think that these requirements are a burden. They will, in fact, ensure that you get the most out of your mentor meeting and that problems are resolved before they become serious and risk derailing your project. Also, learning how to manage a project with a minimal amount of bureaucracy is a valuable skill.

Milestones

We are currently on track with the milestones we laid out in our project plan.

Achieved Milestones

- Report blockage by clicking on the map
- My reports page that displays all of the user's past reports
- Comments on blockages
- Update status for blockage reports
- View history of status updates
- Basic subscription system
- Simple upvote/downvote system for activity points/karma=
- Basic mapping system for the karma levels that make the users start at level 1 and upgrade to level 2 when they have 3 activity points.

Missed Milestones

None, all milestones have been achieved so far in the project plan

Difficulties

Mobile-First Design

Trying to design for both desktop and mobile users has been a difficulty since the very beginning. Despite the paradigm of mobile-first design being easier to extend to both, the design layouts and patterns that we thought would work best for desktop were often radically different than what would work for mobile, and the mobile-friendly decisions would be very clunky and inefficient on desktop. Because of this, we are now mostly using desktop-friendly design.

Status Report Backend with MongoDB

Figuring out the backend for update status reports was also a bit tricky since we had to add an active field to each of the posts to indicate whether it's an archived post or not. Archived posts are previous status updates that are not the most recent one displayed on the map. We originally thought that each post could have a list of update status objects under it to indicate the previous update statuses but that would mean creating a whole new schema so it was better just to have update status create a new post instead with a parentBlockage field pointing to the direct parent update status post above it and set that post's active field to false and the newest one's active field to true to indicate that it is the most recent status update to be displayed on the map for users to see.

Karma Backend with MongoDB

Backend (a common theme here) for upvotes and Karma was also fun because a user's activity score updates every time anybody votes on one of their blockages (possibly among other things, later). The two main approaches are basically recalculating every time it's updated, or every time it's needed. We went with recalcing on update, and used some helpful mongoose schema methods to deal with the fact that this needs to happen on at least five different routes.

Comment Backend with MongoDB

Setting up the backend for the comment feature using mongoose was tricky at first since we made each comment refer back to the post id and have the post refer to a list of comment id's as well for the ease to displaying comments as well as deleting the comment id from the post when the user presses delete.

Refreshing History

Refreshing the history after update status was difficult without the popup disappearing and the user having to re-click on the map marker each time.

Subscription Notifications

Sending notifications at the proper times will be tricky since we will likely need constantly-running code in the backend that checks if it is the proper time to send notifications for a subscription. Parsing subscription times will also be a little tricky, especially since the HTML input for times returns them as strings.

Design Changes

No Longer Mobile-First Design?

We are reconsidering our decision to make our design mobile-first. Though our app can be adapted for mobile use, it is a much more pleasant experience to use on a desktop since you are able to see many elements at once rather than having to switch between panels.

Rebranding Karma

Another minor change-- a decision to use the terminology "Activity Score" in the frontend over "Karma", although we still internally call it Karma

Project Plan Changes

None, we are currently on track with our project plan.

Discussion Points

- For subscription alerts, should we do push notifications or is the notification bar enough?
- Should we send the notifications 30 min before the subscribed time to give the user time to plan their alternative route to get to work in the morning for example? Should we send alerts of blockages that happened in between the subscribed times that are still persisting into the subscribed times or just new ones that happened within those times?
- Possible implementation of extra fields in Blockage (expected start and end times) to make Subscription more useful
- Should we still integrate google calendar for scheduling the subscription times or is the current model of selecting repeating times throughout the week enough?
- Should we incorporate a mobile version of our app still?