Project Plan

Functionality at each Milestone

POC:

For the proof of concept, we will want to have implemented basic forms of the home page, hazards page, routing page, sign-in page and profile page.

Home Page: For the home page, users should be able to view a list of all posted routes and hazards. The routes will not display the maps in their respective containers at this stage in the project but simply have the start address, end address and user that posted it. Hazards will contain some information about the posted report.

Hazards Page: On the hazards page, we will have a feed of hazards which contain information about the report. Users will also be able to report a hazard by clicking an add button in the bottom right which yields a pop-up window and allows users to input relevant hazard information.

Routing Page: There will be a routing page where users can create a route by specifying two addresses. There is a complete route button to publish it. At this stage in the project, there will not be a map displaying the created route.

Sign in/Sign out: Users have the ability to create users, sign in and sign out. When a user signs up they must supply their address.

Profile Page: Displays the profile of the current user, right now only displaying their username.

MVP:

For the minimum viable product, we will have the functionality of the proof-of-concept with maps displayed and the ability to report, confirm and settle hazards. There will be a search bar where users can search for other users. Users will also be able to follow and unfollow other users in this section, which affects which routes appear in their home page feed.

Home Page: For the home page, users will be able to view routes created by their friends. Also in the feed will be reports that users are able to confirm or deny. Users will have the ability to search for usernames and follow or unfollow them. Users can also query in the search bar for addresses to find hazards near an address.

Hazards Page: There will be a reports tab filtered chronologically for unconfirmed reports near a users' location. For each displayed report, users should be able to confirm or deny it. In a separate tab of confirmed reports (hazards), users can click them to settle them (meaning they are no longer a threat or issue).

Routing Page: At this stage, users will be able to specify two addresses and create a route. At this point, a map should be displayed showing the rough draft graphically. Red icons should appear where there are confirmed hazards. There will be a button to complete the route.

Sign in/Sign out: This page will be the same as POC except with improved styling.

Profile Page: The profile page should now display, username, followers/following.

Finished Product:

For the finished product, we will largely focus on UI and styling while the back-end of the project is largely completed. This could include re-arranging layout, or potentially adding any last-minute features.

Home Page: Improved styling from MVP. Users will be able to comment on, like and save routes posted by their friends.

Hazards Page: Improved styling from MVP

Routing Page: Improved styling from MVP

Sign in/Sign out: Improved styling from MVP

Profile Page: Improved styling from MVP. Users will be able to view the routes they have saved from their friends. They will also be able to view any of their previously taken routes along with the comments and likes of those routes.

Implementation Plan

Server:

Task 1: Have api calls for route making fully functional by the proof of concept date. **GARRETT**
Task 2: Implement functionality for storing data from users the current user is following as well as the data from the current user - This should be done by the mvp. **ANDREI**
Task 3: Implement functionality for storing all data for hazards and reports. This should be done by the mvp **REGGIE**

Sign in Page:

Task 1: Have a landing page with just a sign in and a create user option - This should have full functionality complete by the proof of concept **ANDREI**
Task 2: Implement UI for the create user page complete with form functionality and a search for neighborhood by address (back end and api calls for this task is separate) This should have full functionality - implemented by the proof of concept **NOAH**

Home Page

Task 1: Include all of the public routes of the users a user follows. - This should be done by the proof of concept date **NOAH**
Task 2: Include all of the hazards that have been requested and not approved yet for users to approve or deny. - This should be done by the mvp **REGGIE**
Task 3: Implement a search bar where you are able to lookup users and see their public routes and their stats. Also allow users to follow by search. -This should be done by the mvp **GARRETT**

Hazards Page

Task 1: Include all of the confirmed hazards and allow users to settle hazards if they are no longer relevant. - This should be done by the proof of concept date **ANDREI**
Task 2: Implement functionality for reporting, confirming, and settling hazards. -This should be done by the proof of concept date **REGGIE**

Routing Page

Task 1: Have two search bars from which users can search for two addresses and then create a route. - This should be done by the proof of concept **NOAH**
Task 2: Have UI show red icons for hazards and implement buttons for users to complete the route. - This should be done for the mvp. **GARRETT**

Profile Page

Task 1: Implement UI to show the username of current user and their following/followers info -This should be done by the proof of concept **ANDREI**
Task 2: Implement a feed of your trips and a feed of your saved trips. -This should be done by mvp **REGGIE**
Task 3: Implement a statistics page which will include mileage total trips and several other fun tidbits. -This should be implemented by the final product **NOAH**

What Happens if Something Goes Wrong?

As with any project, there are a number of things that can go wrong. There are a few key areas of the project that may create difficulties down the road which we will discuss here. The first is the display of maps. None of us have worked with the map API, so we anticipate there being some issues in creating routes and displaying maps. For example, if we route between addresses, and we want the route to avoid a certain hazard, is there a way to make the route automatically calculate the best way around using our chosen API? If not, our plan will be to have the route without the diversion, but with the hazard still displayed, for users to come to their own decision about how to deal with it. The second area of concern could be with our workflow. If one member of the team is lagging behind and struggling to complete tasks, we should have a plan to help them out and prevent any tasks from bottlenecking others. This should be covered by updates as soon as tasks are completed so we have an accurate idea of progress. If someone is lagging, we will address it by meeting and deciding how best to re-assign tasks and help them out while they are slammed with other classwork. Another issue we could run into is collectively being behind on tasks. If we feel that we are going to miss a milestone, we will meet, and potentially involve TAs to identify the problem, cut back on styling/UI work and focus on the bare bones functionality. The main key will be communication so that everyone is on the same page.