Initial Design

Project Summary

Here we'll discuss what our ultimate project really is. What stakeholders we aiming to please. What goal we hope to accomplish, etc. As of now we have yet to finalise our project choice, but we are thinking of the following ideas:

1) **Plague Inc with real-time data**: create some sort of web game similar to the popular mobile game Plague Inc except with real-time data. Players would be able to replay disease scenarios and their effects.
2) **Correlations vs Causation of viral diseases with Twitter**: Create a web app that displays a world-map and a timeline of diseases and have a way to view the tweets and 'hashtags' that were trending during that time. This will either reveal profound insights of the spread of diseases and certain outbreak and its correlation with social media events, or provide humorous and seemingly random correlation vs causation of a disease outbreak with certain tweets.

Software Architecture

The software architecture that the team decided for this project is outlined as below and the reasoning in why we chose the following software systems.

<insert amazing tech stack dataflow diagram here of entire system architecture>

Web Scraper Design:

Here we will discuss the tech stack used, why we used them over other alternatives and how the web scraper will work. This is our rough idea of the tech stack so far:

- Python3, using the 'scrapy' library
    - Quick to prototype
    - Easy to use
    - Team has strong familiarity with the language

API Design:

Here we will discuss the tech stack used, why we used them over other alternatives and how the API will work. This is our rough idea of the tech stack so far:

- Flask? Express.js? to host the endpoints.
- Google Firebase Backend for database storage of scraped data

API Documentation Website Design:

Here we will discuss the tech stack used, why we used them over other alternatives and how the API documentation will work. This is our rough idea of the tech stack so far:

- SwaggerHub is not free for more than 1 person so probably not.
- StopLight.io seems to be the way to go.

Website Design

Here we will discuss the tech stack used, why we used them over other alternatives and how the website will work. This is our rough idea of the tech stack so far:

- React for frontend.
- NodeJS backend to interact with the API we created.

Deployment

Here we will discuss how we will deploy all of the systems previously explained. Here are our rough ideas so far.

- AWS Lambda for website + API Documentation Website + API:
    - Has no cap, meaning we could lose money
- Google Cloud for website + API Documentation Website + API:
    - Has a cap. $300 free trial.
- Google Firebase for our database

DOCUMENT DRAFT END

"1. Describe how you intend to develop the API module and provide the ability to run it in Web service mode

2. Discuss your current thinking about how parameters can be passed to your module and how results are collected. Show an example of a possible interaction. (e.g.- sample HTTP calls with URL and parameters)

3. Present and justify implementation language, development and deployment environment (e.g. Linux, Windows) and specific libraries that you plan to use."