ihatequarantine -- Kevin Li (PM), Derek Leung, Justin Shaw, Albert Wan
SoftDev2 pd9
P04 -- Data Visualization
2020-04-23

Epidemics Comparison: Day by Day

Task division

- Kevin Li
    - Project manager
    - Update README, make sure devlog is up to date
    - Update design doc to reflect changes to project midway through
    - Some coding for the D3 part (specifically hovering over a line for more detailed info)
- Derek Leung
    - D3 work
        - Drawing the base line graph (scale does not change)
        - Allow the user to choose how many days to show on the X-axis (minimum 100 days, maximum 800 days)
    - Will help with JavaScript if necessary
- Justin Shaw
    - D3 work
        - Toggling specific lines on and off on the graph
        - Scale Y-axis accordingly as diseases are switched off by the user
    - General JavaScript work
- Albert Wan
    - Will mainly work with Flask and Python to read the CSV file
    - CSS/Bootstrap styling

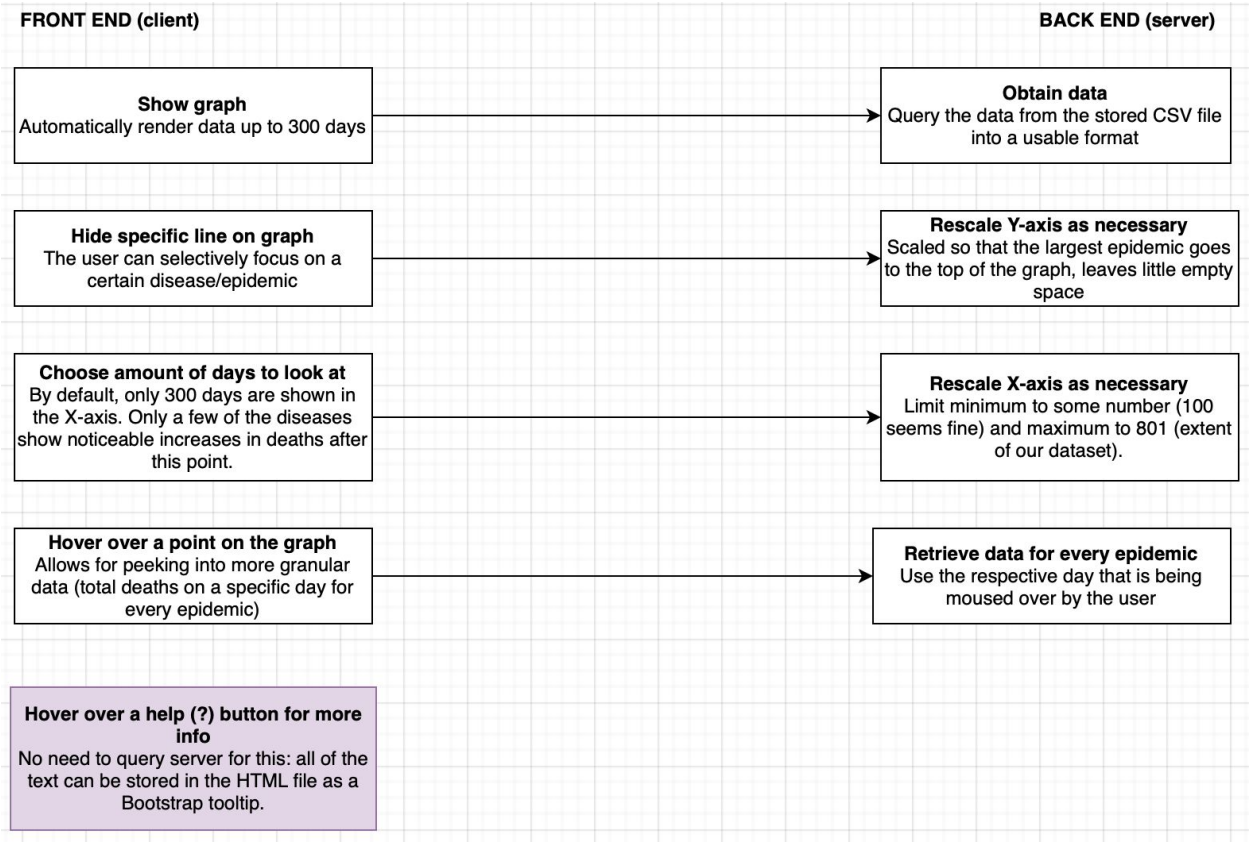
Project Description

This website is an interactive line graph that allows the visitor to compare various epidemics/pandemics in recent history (e.g. the 2010 Hispaniola Cholera outbreak, COVID-19, etc.) in terms of total deaths, day by day. These features are included for interactivity with the website:

- Hiding specific lines to focus more closely on the remaining outbreaks
- Choose the amount of days to look at
    - By default, the data only shows up to 300 days into an outbreak.

    - The user may set a minimum of 50 days, and a maximum of 800 days (the extent of the dataset used).
- Hovering over a spot on the graph for more granular data
    - Show the amount of deaths for every epidemic at once (with a vertical line running down)

Component map/available features to the user



Site map

A site map is not necessary for a project like this. We are planning to have only a single page. A separate help/FAQ page may be helpful, but even that can be implemented as a bunch of tooltips that get shown after hovering over a (?) button next to the appropriate section on the page.

Bootstrap or Foundation?

The majority of our group's members are more familiar with Bootstrap, and as such our frontend framework will be this.

Everything about data

- The dataset we are using is a CSV file that breaks down the total amount of deaths attributed to 13 separate pandemics (i.e. Cholera, COVID-19, Ebola), per day.
    - For example, day 1 of the COVID-19 outbreak had 0 deaths. It remained the case until day 11, the first death. Day 16, the second death.
    - The data for COVID-19 is up to date as of April 1st. It is still fairly early into this pandemic compared to the others, so the data will cut off abruptly.
    - Some of the diseases appear to repeat. This is because outbreaks of that disease have happened multiple times throughout history and in different regions.
        - To play nicely with D3, each day's data is a dictionary: day after beginning of epidemic and total deaths.
    - Original data: https://docs.google.com/spreadsheets/d/1zn_pqFBv9W9Hrfe-0LcfSYdywZHe4cOig4xQZ5mVaBQ/edit#gid=1624097889
    - Took only the necessary data and exported to a CSV file for easy manipulation/access.
    - Transposed columns to rows to play nice with Python CSV functions.
- We will not be using an sqlite3 database for this project, as we want to focus on sharpening our d3.js skills. An API will also be more trouble to deal with than necessary.
- We will obtain the data using Python functions in the utl folder.
- It will be passed through jinja2 variables (variable | tojson inside a <script> tag)
- The variable will then be able to be accessed in any other JS files that are linked.
- Data from backend to frontend diagram:

Python function returns:

{ Dictionary of the various diseases }					
'cholera-hispaniola-2010'	**'covid-19' (key)**	**'ebola-wafrica-2014'**	**'swine-2009'**	**'sars'**	9 more epidemics...

Value: dictionary

Days since pandemic began	Total deaths
0	0
1	0
...	...
65	3202

Passing from backend to frontend

1. Pass dictionary through home route as a jinja2 variable
2. Use <script> HTML tag to pass variable to other JS file(s) of the app using jinja2 bracketsand variableItojson
3. Access dictionary as you normally would, but in JS file(s)

-

Minimum viable product

- There will only be one main page
- A line graph automatically renders on the page.
    - One differently colored line is drawn per epidemic/pandemic.
    - The X-axis is the amount of days since the first day of each outbreak. For the MVP, we want to cap it. 300 days seems like a good default point because the numbers for most of the epidemics have stopped increasing in "drastic" amounts.
        - The dataset actually goes to 800 days.
        - For the MVP, we don't have to worry about resizing it since it's capped at 365 days.
        - Since we can't show 365 tick marks, we should split it into chunks (25 days, 50 days, 75 days, etc.) [edit: turns out D3 handles this automatically]
    - The Y-axis is the total amount of deaths caused by an outbreak.
    - Hovering over a point on the graph should show the exact number of deaths for each epidemic on that day
        - COVID-19's data abruptly ends, so there has to be an edge case where COVID-19's data stops being shown as soon as the data cuts off.
- Below the graph is a legend. Similar to Naviance's Scattergrams feature, the user should be able to click on a disease to hide the line for that disease. It is completely wiped (not just hidden, otherwise you'd be able to hover over an invisible line for data)
    - Do not let the user turn off the data for every single disease - just in case something screwy happens because of it.

Extras if we have time

- The user being able to pick how many days are to be shown on the X-axis
    - Limit between 50 days (arbitrary limit, I know, just gets rid of edge cases in case the user tries putting 1 in or something) and of course, 800 days.
    - Use an HTML input field for this.
- The Y-axis scales accordingly.
    - For example, turn off the data for Swine Flu/H1N1, by and away the largest dataset that dwarfs the others in total deaths.
    - If the scale doesn't change, there's a massive amount of empty space in the Y-axis. It should scale so that the next largest epidemic/pandemic reaches to the top of the graph.
        - Calculate the max amount of deaths (and some change, * 1.2) of a disease on a specific date