SENG3011
FINAL REPORT

MapOutbreaks

Team - Coders

Amol Jain - z5097081

Bhawna kundu - z5130378

Clinton Huynh - z3417167

Sunreet Singh - z5130780

Table of Contents

1. Introduction

The report is written to discuss our final completed project in terms of requirements, Project Management and what improvements can be made.

The website we are using to host our prototypes is,

For API - http://35.189.54.60:5000
For Final Working prototype - http://35.189.54.60:12000

Test input files and other files can also be found in the GitHub repositories.

2. Requirements

As per the team's analysis of the project, we tried to implement each and every necessary requirement for the project as well as add some further features to make the prototype user friendly and more useful to solve real world problems.

At the time of writing the Initial Report at the beginning of the semester, we were only given the sole task of creating a prototype which displays all the outbreaks around the world.

As of this moment, we have successfully completed development of a working prototype, MapOutbreaks, which has the capability of scraping and displaying all the data provided on CDC website with a detailed report analysis for each outbreak around the United States. The prototype also has the capability of fetching outbreak alerts/news from Google news and displaying it in a visually appealing manner with pin markers all over the globe.

The current list of requirements is:

Requirements/Features	Achieved?
Develop a scraper to gather data from the main data source. - Scrapes real time data - Stores the data in a database - Detailed report analysis(Hospitalised/ Deaths/Reported Cases)	Yes Yes Yes Yes
Develop an API to provide disease reports on demand, following the input and output complying with the format specified. - Produces JSON output in the specified format - Speed Of Execution	Yes Yes Can be improved.
Published API documentation with parameter specifications, capability to execute and test the API online along with Swagger documentation. - Shows error wherever required - Produces a valid swagger document	Yes Yes Yes
Develop a web application that integrates disease reports from the API we developed and one more API. - Integrated with other API	Yes Yes
Displayed all the outbreaks in a user friendly and visually appealing manner. - Displays outbreaks from CDC as well as Google News API - Filters/searches a particular outbreak with keyterms/ location/time period - Distinguish outbreaks as large/small(red marks on the globe) - Provided url's to know more about a particular outbreak	Yes Yes Yes Yes Yes

3. Use Cases

3.1 Use Case 1

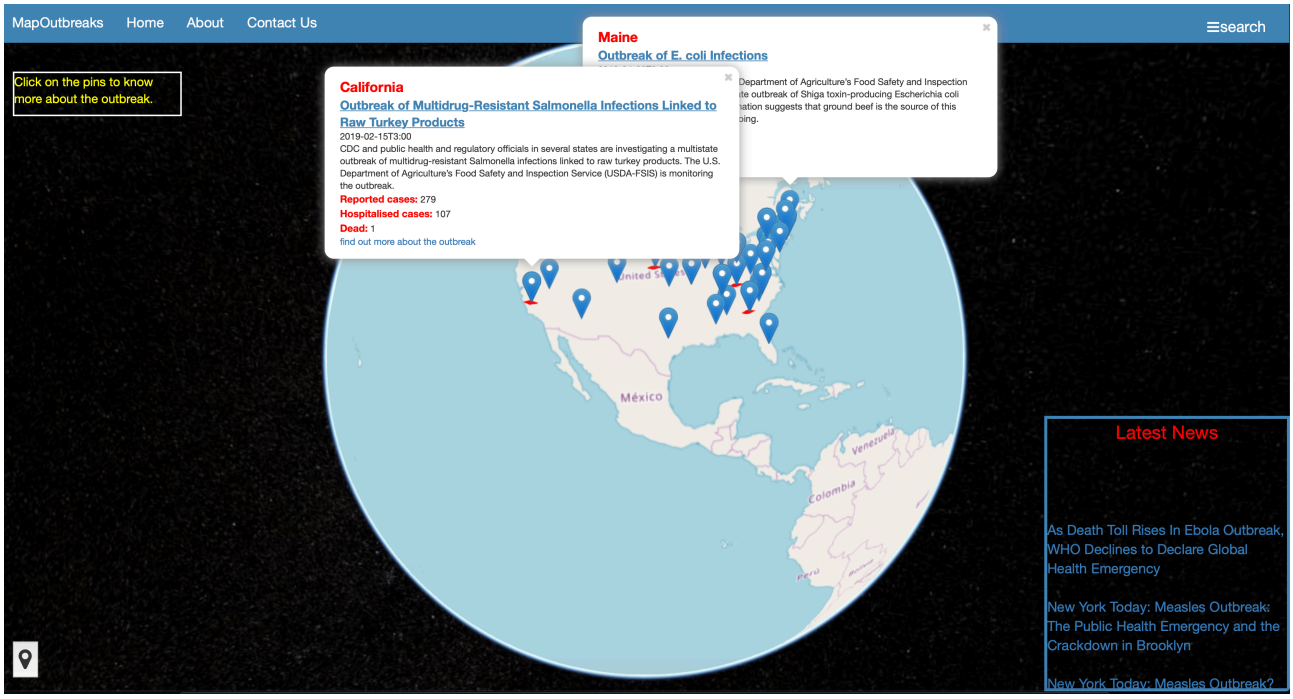
Users : Health Organisations and Medical Organisations

Description : User wants to know about the health outbreak at a particular location.

Walkthrough :

1. Goes to the website
2. Clicks at a pin marked on the globe to know more about the outbreak.
3. Red marks under the pin shows the large outbreaks so that medical emergencies can be rushed at that particular location or better arrangements of medical services (E.g. More vaccination awareness etc.)

Figure 1:



3.2 Use Case 2

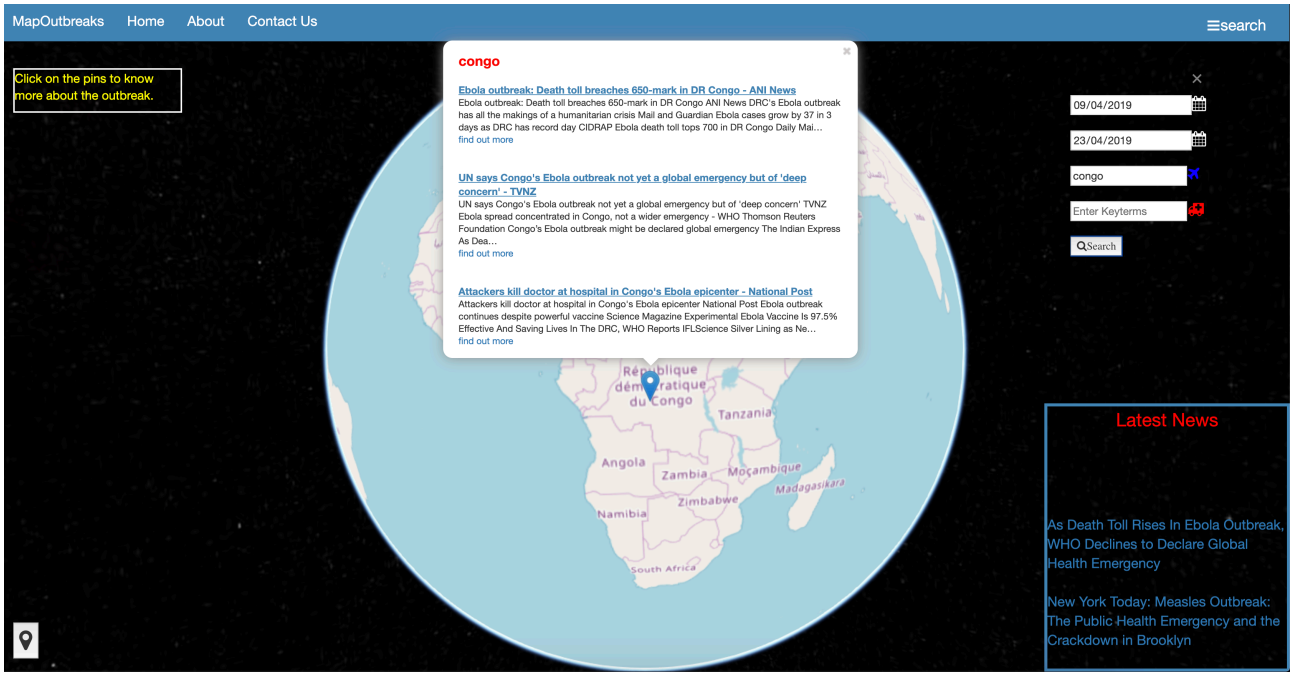
Users : Travellers, any other user and Medical Organisations

Description : User wants to travel to a location(international or local) and wants to know about the outbreaks so that they can prepare accordingly.

Walkthrough :

1. Goes to the website
2. Clicks on the search bar displayed on the top right of the website
3. Enter the period of interest and the location they are interested in and clicks search.
4. The globe rotates and a marker is placed at that location with all the details about the outbreak.
5. User can know more about the outbreak by clicking “find out more” which direct the user to the webpage displaying more information about the outbreak.

Figure 2 :



3.3 Use Case 3

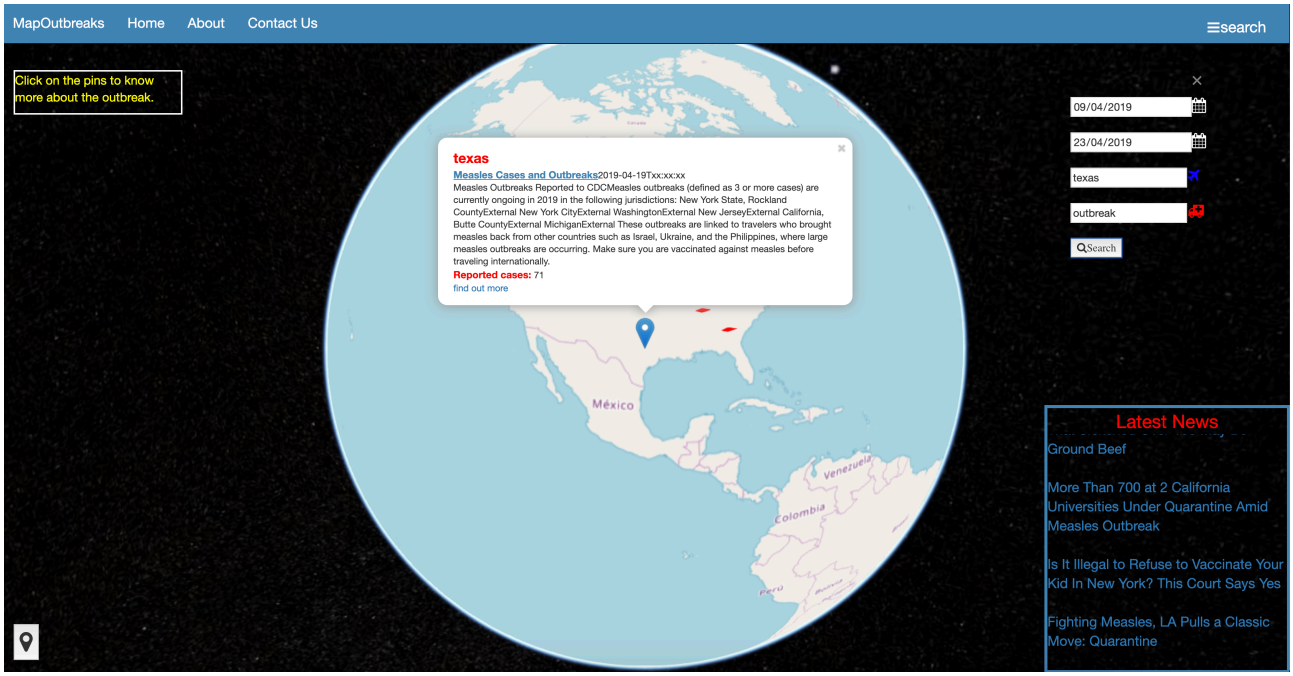
Users : Travellers, any other user and Medical Organisations

Description : User wants to travel to a location(international or local) and wants to know about the outbreaks so that they can prepare accordingly.

Walkthrough :

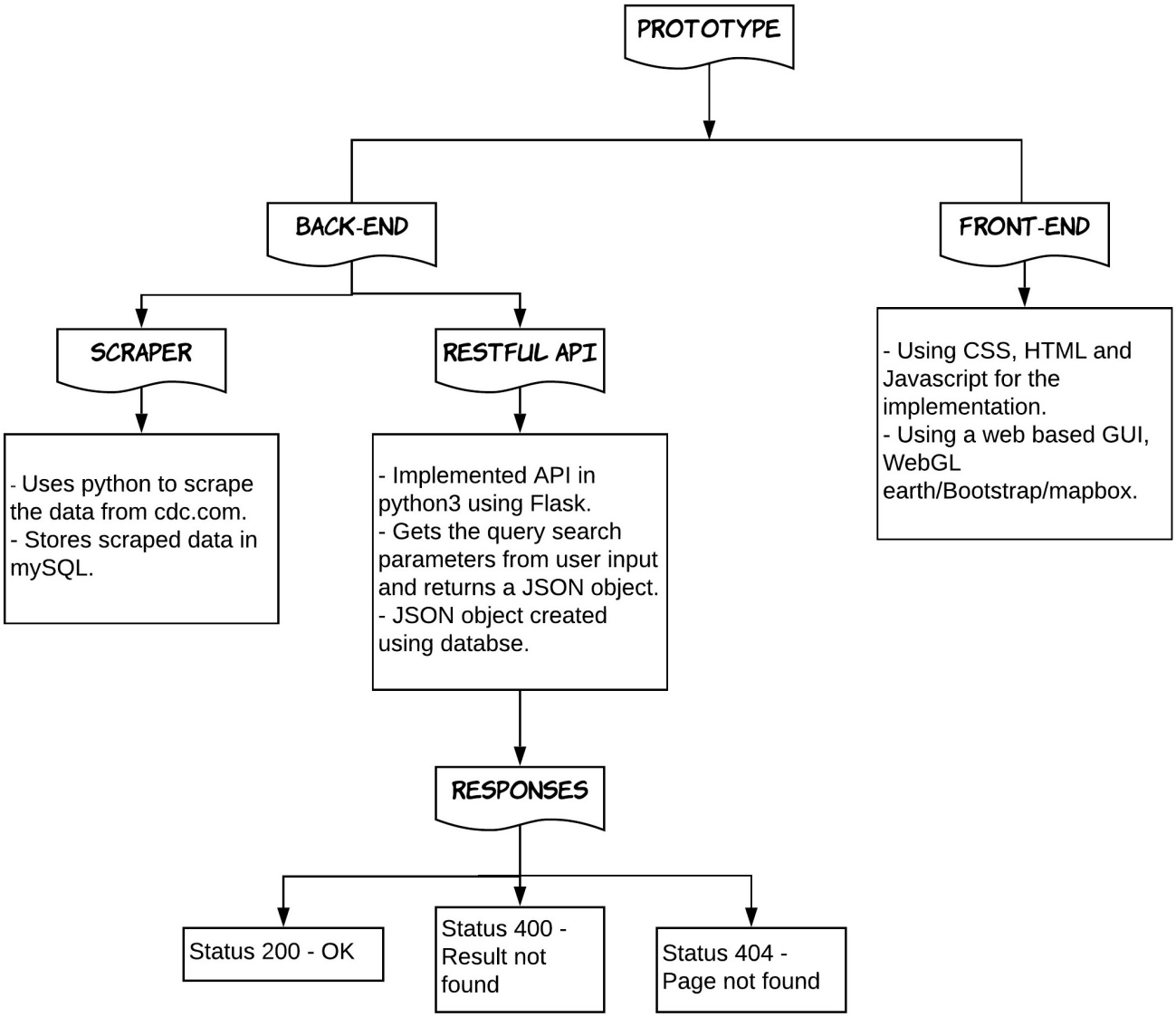
1. Goes to the website
2. Clicks on the search bar displayed on the top right of the website
3. Enter the period of interest , location and a particular disease/ outbreak they are interested in and clicks search.
3. All the markers disappear other than the one the user is interested in.
4. Additional details: The Latest News box shows the news articles which the user can click and is redirected to that particular webpage.The pin at the left bottom corner of the screen helps the user to relocate(moves the globe) to America(where most outbreaks are displayed).

Figure 3:



4. Design, implementation and Testing

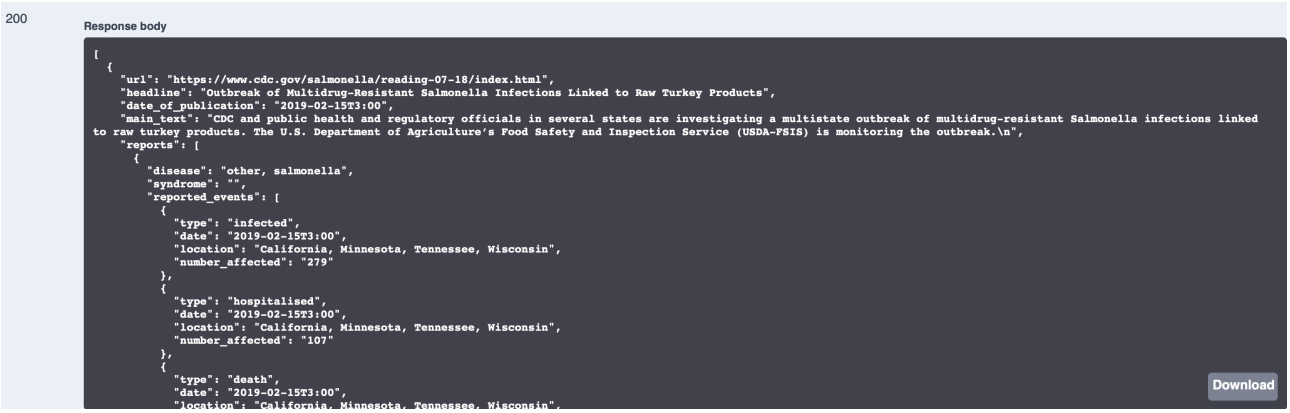
The project was divided into different subparts which are described below:



Each subpart describes the languages and environment used for implementation.For our Initial Report, we were instructed to clarify how parameters are passed to the API. As such, we will now explain our improvements since then in details. Example of request url with parameters(Period of interest, location):

http://35.189.54.60:5000/show?start_date=2019-02-15T03%3A00%3A00&end_date=2019-02-25T03%3A00%3A00&location=california&keyterm=fever

JSON response(STATUS 200: OK) :



Example of request url with parameters(Period of interest, location, keyterm(s)):

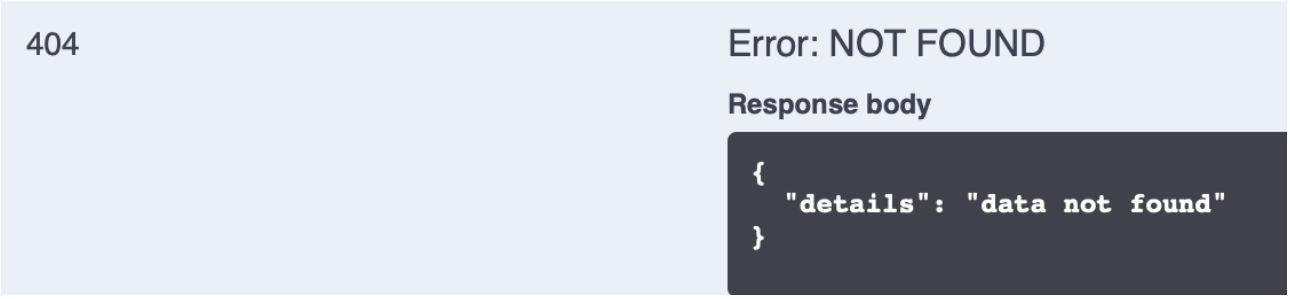
http://35.189.54.60:5000/show?start_date=2019-02-25T03%3A00%3A00&end_date=2019-02-15T03%3A00%3A00&location=california&keyterm=fever

JSON response (STATUS 400: BAD REQUEST):

Example of request url with parameters(Period of interest, location) :

http://35.189.54.60:5000/show?start_date=2019-02-15T03%3A00%3A00&end_date=2019-02-25T03%3A00%3A00&location=New%20York

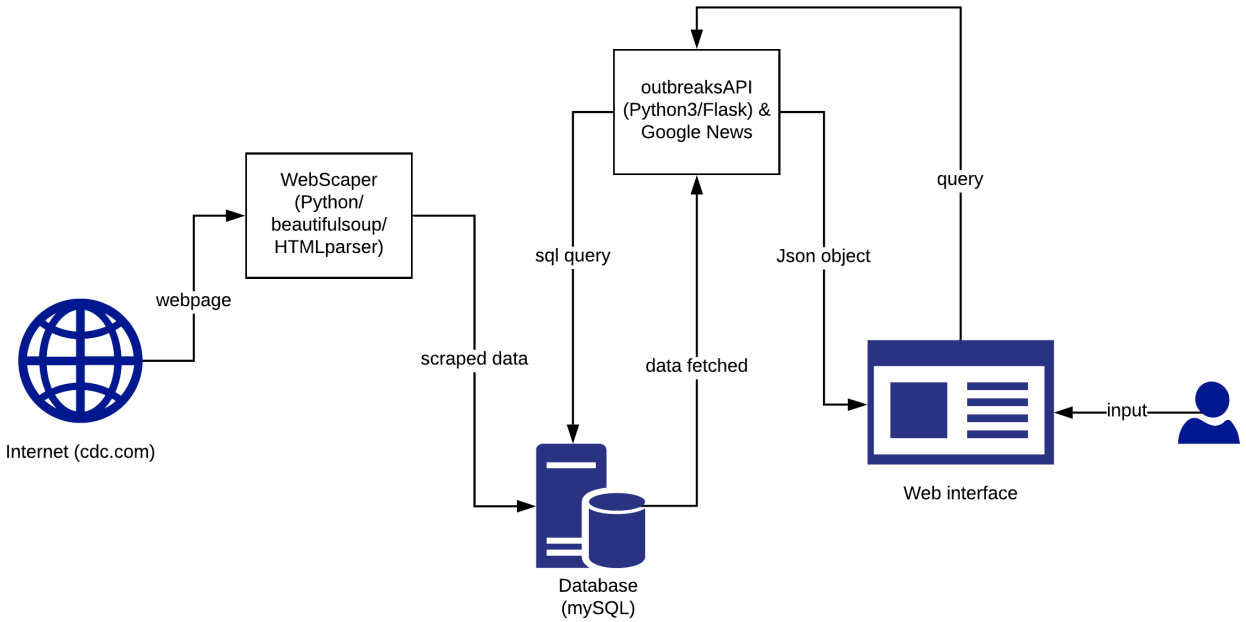
JSON response (STATUS 404: NOT FOUND):



4.1 Architectural Design

We had an architectural diagram which we designed and presented in the initial report and building up on that after working on the feedback provided during the meetings/demo we were successfully able to mend our mistakes and produced a better version of our prototype.

The model digram below shows the interaction of client side and the server side giving more overview on how the prototype is actually working.



Model 1

The architecture has three main parts:

1. The Web interface which the user sees and interacts with.

2. The second component is the interaction between the API and the mySQL database.

3. The third component consists of the interaction between the API and the Web interface.

The user inputs the data they want to get information(Outbreaks in a particular location with a period of interest) about which is then passed as

the query string parameters (start date, end date, key terms and locations) in the API. The API then filters the data from the database and produces the JSON object which is returned to the web interface and displayed in a visually appealing manner.

Below are some of the more snapshots from our website:

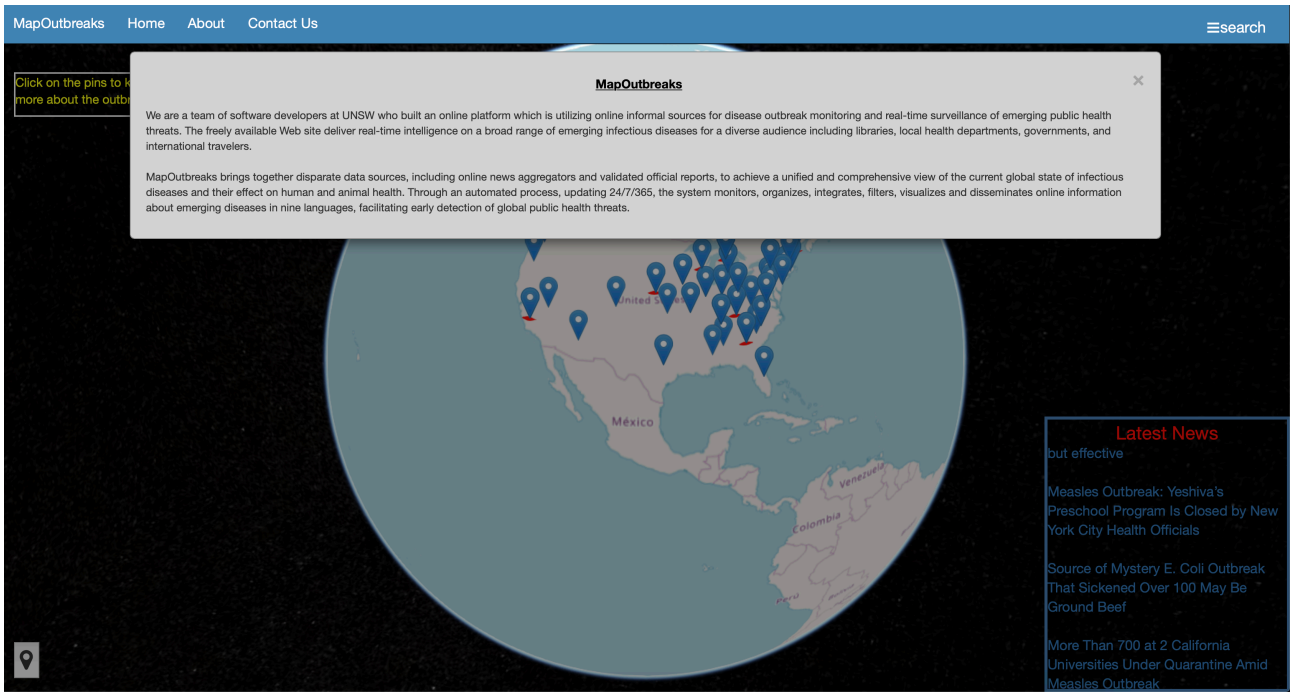


Figure 1 : Home page with the about popup.

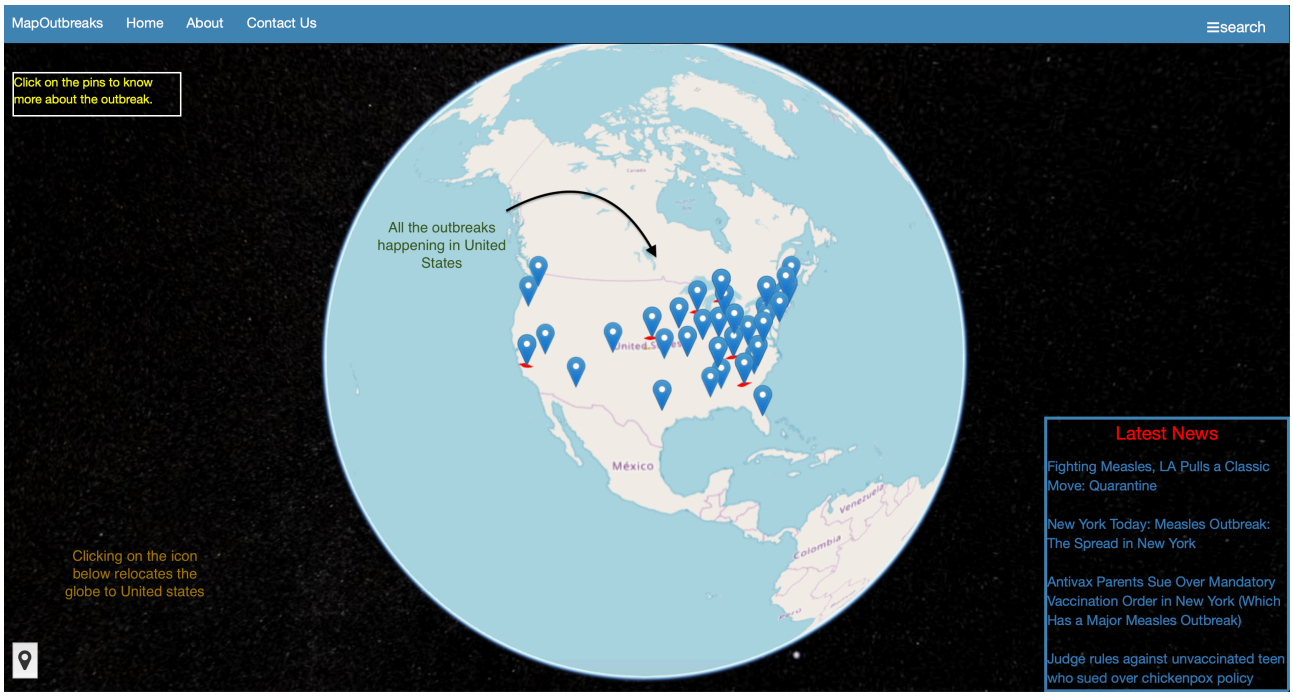
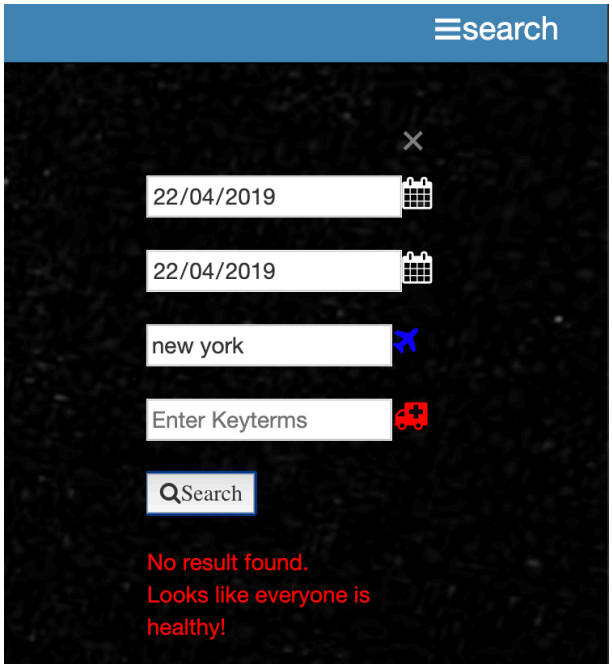
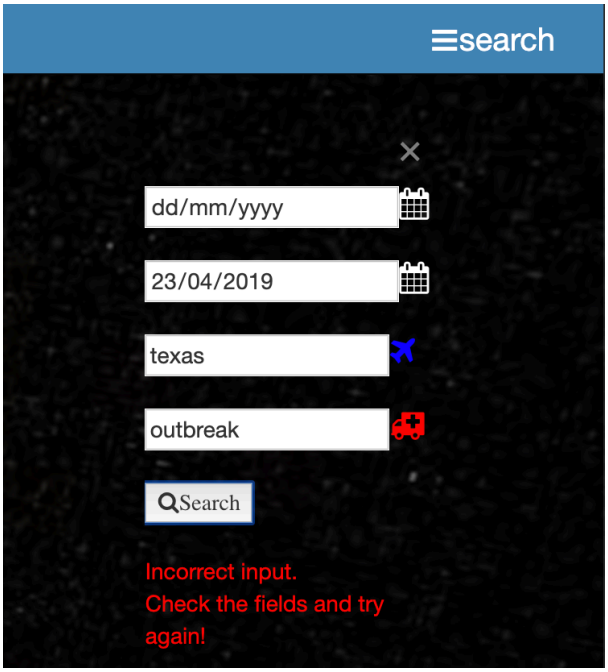


Figure 2 : All outbreaks happening around United States and latest news from google news.

We tried to handle all the errors, some errors that you can get while searching about a particular outbreaks are:





4.2 API's used

The team decided on to integrate with Google News API for more outbreaks happening around the world. All the news were filtered with health and Outbreaks filters.

4.3 Testing

The scraper was coded to analyse the web pages to get the brief but useful data which was used in our user friendly front-end interface. After spending good amount of time, trying several regex expression, we used a few which works efficiently and fast. The Scraper program was tested over and over to get it working properly.The initial API prototype had some major issues. These issues made requesting data from API troublesome and was returning incorrect output. After having feedback from our mentor, we fixed the API to be consistent with specifications. The JSON object returned by the API then was checked for returned HTTP status, data and correctness of data. Further, We tested the front-end in real time with console on chrome. This part took the most time due to heavy testing to get everything right.

5. Summary

Benefits and Achievements

The final design and implementation of our prototype benefits from several advantages. We have attempted to emphasise usability for all of our pieces of software.

As mentioned in our previous reports, we chose to develop our API using Python because one of our aims was to make the software as cross-platform and portable as possible. We have achieved this, and the prototype can be accessed via the link provided. However, we made the decision to develop the GUI in HTML, CSS and Javascript.We tried to make the prototype working on every platform and also on devices with different screen sizes.

Our final prototype places a great emphasis on making the experience for the user as easy and simple as possible. We have ensured that there are not too many buttons or popup windows on the interface to avoid unnecessary clutter so that it becomes intuitive to the user as to exactly what action each button performs.

The prototype displays the content of each outbreak in a popup bar rather than a whole new page which makes it easy to understand, clean and simple. In the date field we used the calendar input so the user does not need to manually input them the date each time and can instead use year/month/date from the calendar only.

In the end, we were satisfied with the product that we came up with.There can be some modifications/features that can be made to the current prototype which has been mentioned latter in the report.

6. Team Organisation

6.1 Responsibilities

Our team consisted of four members, and work was organised so that it would capitalise on each of the members' strengths.

1. Amol

    - Implemented a basic working API.

    - Assisted with deliverable reports.

2. Bhawna

    - Worked on the Scrapper.

    - Developed the GUI component of the platform in HTML, CSS and Javascript.

    - Worked on the Swagger Documentation and assisted with the development of various features.

    - Took lead in communications and management and envisioned the goals to be achieved.

    - Worked on creation and maintenance of deliverable reports.

3. Clinton

    - Implemented a basic working API.

    - Assisted with deliverable reports.

4. Sunreet

    - Setup the GitHub repositories.

    - Worked on the Scrapper and developed a fully working API.

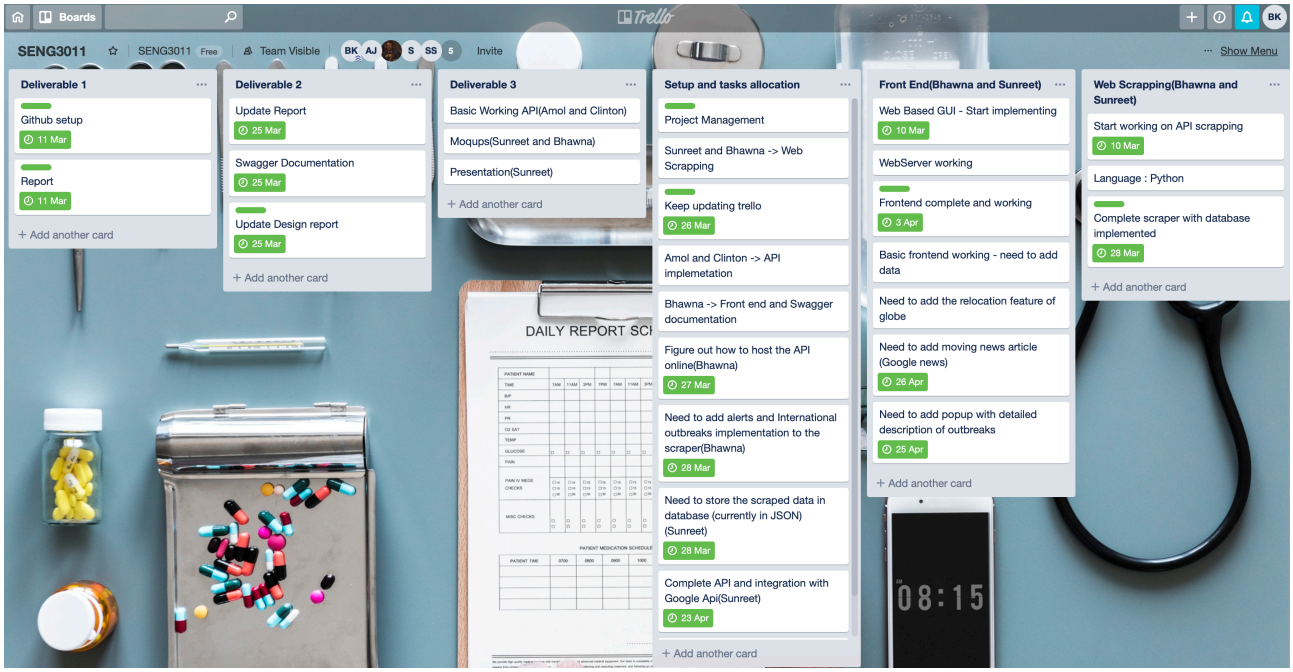
    - Connected the front-end and the backend and performed code reviews on the other codebases in the project.

    - Assisted with creation and maintenance of deliverable reports

6.2 Organisation/Tools used

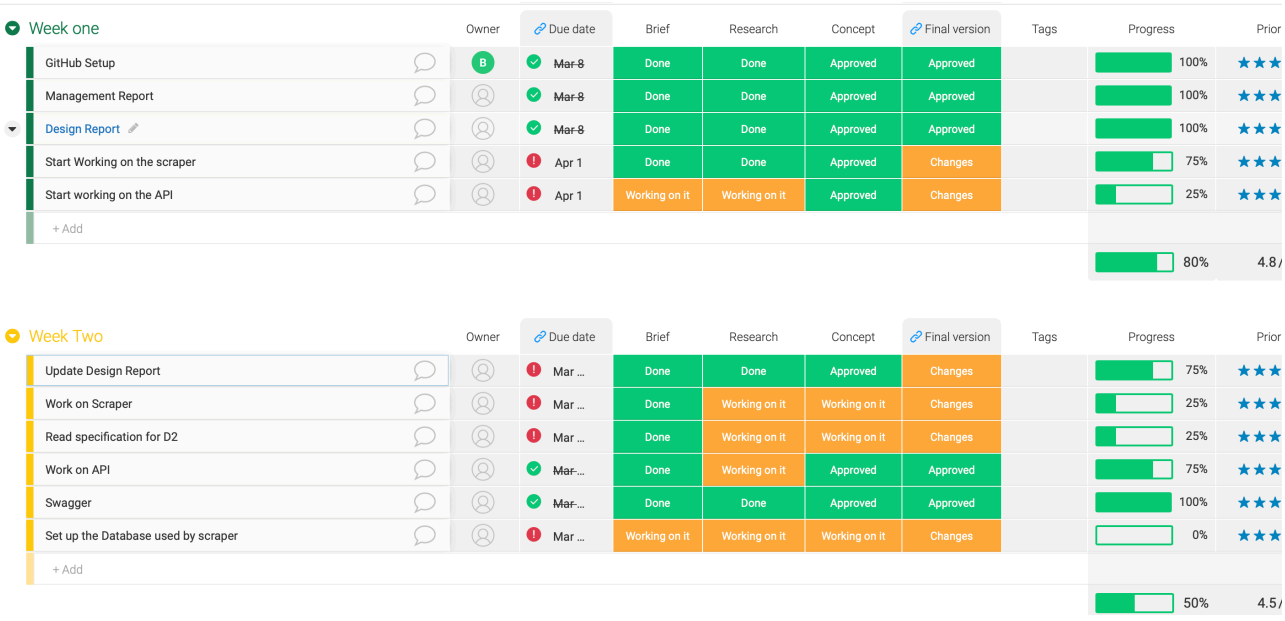
The team decided to use Trello and monday.com to allot the tasks and manage the deadlines.

The current Trello board looks like as shown below:



monday.com is a great platform where we can actually allot task to members and check on the progress.

The task desk for one and two looks as shown below and the rest was the similar to this:

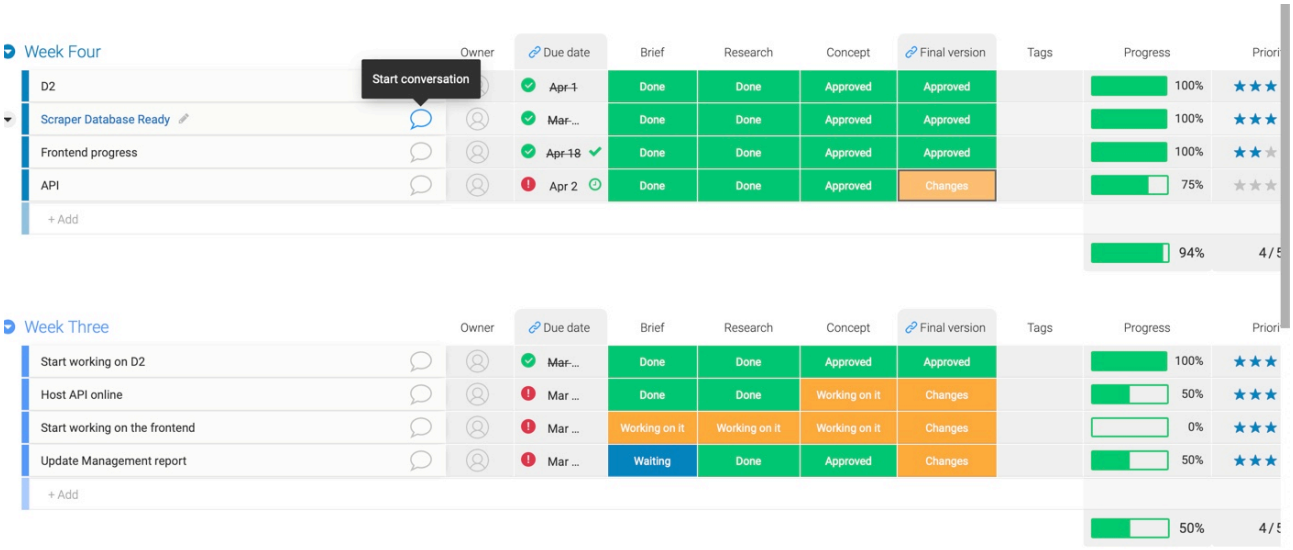


Week one

	Owner	Due date	Brief	Research	Concept	Final version	Tags	Progress	Prior
GitHub Setup	B	~~Mar 8~~	Done	Done	Approved	Approved		100%	★★★
Management Report		~~Mar 8~~	Done	Done	Approved	Approved		100%	★★★
Design Report		~~Mar 8~~	Done	Done	Approved	Approved		100%	★★★
Start Working on the scraper		Apr 1	Done	Done	Approved	Changes		75%	★★★
Start working on the API		Apr 1	Working on it	Working on it	Approved	Changes		25%	★★★
+ Add									
								80%	4.8

Week Two

	Owner	Due date	Brief	Research	Concept	Final version	Tags	Progress	Prior
Update Design Report		Mar ...	Done	Done	Approved	Changes		75%	★★★
Work on Scraper		Mar ...	Done	Working on it	Working on it	Changes		25%	★★★
Read specification for D2		Mar ...	Done	Working on it	Working on it	Changes		25%	★★★
Work on API		~~Mar ...~~	Done	Working on it	Approved	Approved		75%	★★★
Swagger		~~Mar ...~~	Done	Done	Approved	Approved		100%	★★★
Set up the Database used by scraper		Mar ...	Working on it	Working on it	Working on it	Changes		0%	★★★
+ Add									
								50%	4.5



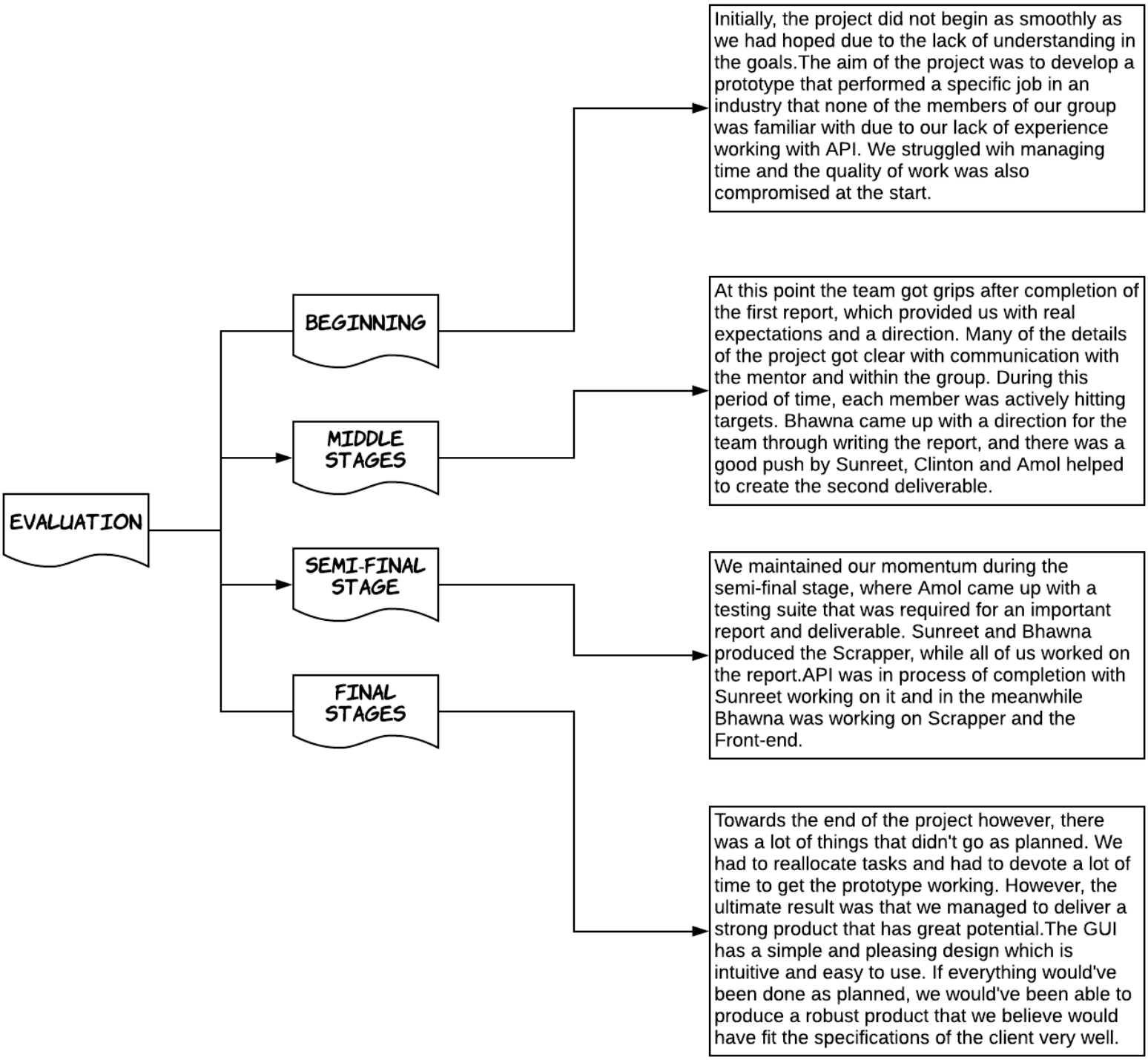
Apart from the above mentioned tools,

1. The team decided to have weekly stand-up meetings in which we worked on the feedback provided in the previous meets with the mentor.

2. The team also used Facebook messenger for all the communication and GitHub and LucidCharts to share the parts they have been working on.

6.3 Conclusion/Appraisal

As we've reached to the final stages of the project, looking back at our progress and challenges faced during the process of development, we tried to evaluate and put the whole experience as follows:



6.3.1 Issues Encountered

There were many instances where we had some issues with developing this prototype.

We encountered some problems to develop Swagger documentation and hosting swagger UI for our API. It was really tough to understand how the swagger documentation works and how its made. There was not many useful resources online to help us with it. Eventually, we used flask-restful api to make it work.

Next, the team was not used to Google Cloud platform. Again, lack of useful and easy to read resources online made it obscure.

For the front-end, we wanted to display the outbreaks in a visually appealing manner, we wanted to do something different from other teams, hence we used 3-D WebGL globe.

WebGL being an emerging software for 3-D visualisation was challenging to work with. We spent quite some time to display the globe and had to study the documentation throughly. Positioning and relocating the globe was hard to understand. In order to mark location on the globe, we had to hardcode countries and states which was time consuming.

6.3.2 Skill requirements

After completing the whole project, we think that it would've been better if we all had a background in how to build an API or using Google Cloud Platform.

We were not familiar with implementation of databases in real world applications, it would've been easier for us if we had some database management skills as well.

6.3.3 What we would do differently

For the prototype:

There were some extra features that the team thought of adding but due to the limited time frame, could not implement:

1. We planned on adding a list on the main webpage showing all the diseases so that the user can choose from them and search that particular disease which would have been easier.

2. The searching(internationally) was just limited to countries and we would like to add regions for more specific information and also we could make the searched things remembered so that the user does not have to enter stuff again and again.

3. The team also wanted to add future prediction based on the trends, how the outbreaks occurred in the past.

For team organisation and work arrangements:

Now that we are at the end of the project, we have a stronger understanding of how the API and other components like Google Cloud Platform, Swagger document works. This would allow us to hit the ground running, should we have to repeat the process again.

We will organise better leadership, structure and boundaries from the beginning. A casual atmosphere within the team helps its members to feel comfortable, but it can hinder performance at crucial stages.Work timelines will be enforced, created with the input of all members of the team, since not everything can be done in a rush last-minute. Information of expected deliverables should be perfect and clear, with no ambiguity and assumptions made. Also, while team members should not be pressured into tasks, they should not be given free opportunity to decline either.

Finally, each individual member should act professionally as well. Communications should be maintained at all times, and messages and instructions from the project manager should be acknowledged. Each member of the team would have to make a greater effort to be individually motivated and each member should be proactive in their participation in all stages of the project.

Overall, it was a pretty good experience and we got familiar with a good learning environment from the project ranging from working in a group, sticking to deadlines and working under pressure , time management, enhanced coding skills and better presentation skills.