Final Report

SIXTYHWW SENG3011 Report

Nikil Singh (z5209322)
Joshua Murray (z5207668)
Tim Thacker (z5115699)
Daniel Ferraro (z5204902)

Contents

Requirements and Use Cases

API Requirements

Summary of requirements.

- ReqA1: Collect data about diseases and syndromes from a data source
    - ReqA1.1: Traverse a data source for cases of diseases / syndromes
    - ReqA1.2: Extract key pieces of information
    - ReqA1.3: Sort the data into a database
- ReqA2: Provide details on reports from data sources
    - ReqA2.1: Return reports based on search parameters such as disease / syndrome, data, location
    - ReqA2.2: Return reports based off search text
    - ReqA2.3: Paginate results
- ReqA3: Provide a user service
    - ReqA3.1: Database for users and their
    - ReqA3.2: Database to store the locations, diseases and syndromes users are following
    - ReqA3.3: An endpoint to register and login
    - ReqA3.4: An endpoint for users to follow / unfollow locations, diseases and syndromes
    - ReqA3.5: An endpoint to return a list of reports related to the locations, diseases and syndromes they are following

Analytics Platform (Frontend) Requirements

Summary of requirements.

- ReqB1: Displaying a variety of articles from data sources.
    - ReqB1.1: Feed displays data from our API.
    - ReqB1.2: Feed displays data from other groups API.
    - ReqB1.3: Article headline is linked to a page with more detail.
    - ReqB1.4: Detailed article page contains link to the source.
    - ReqB1.5: Articles posted in feed contain basic data such as headline, date published and data source.
- ReqB2: Prediction on disease impacts in different countries.
    - ReqB2.1: Can parse our data to extract required information.
    - ReqB2.2: Performs a prediction based on country and disease.

  - ReqB2.3: Performs predictions on cases.
  - ReqB2.4: Performs predictions on deaths.
  - ReqB2.5: Use other group APIs for extracting data.
  - ReqB2.6: Has an API front to fetch for predictions.
- ReqB3: A global map displaying outbreak data.
  - ReqB3.1: Displays markers for articles from feed.
  - ReqB3.2: Displays markers for search results.
  - ReqB3.3: Displays marker for a specific article when viewing a detailed version.
  - ReqB3.4: Clicking on marker shows information on location and disease.
  - ReqB3.5: Map is displayed on site.
- ReqB4: Personalised feeds for users.
  - ReqB4.1: Displays articles based on locations the user follows.
  - ReqB4.2: Displays articles based on diseases the user follows.
  - ReqB4.3: Users can follow/unfollow a disease.
  - ReqB4.4: Users can follow/unfollow a location.
- ReqB5: User Management.
  - ReqB5.1: User can login to account.
  - ReqB5.2: User can logout of account.
  - ReqB5.3: User can sign up for an account.
- ReqB6: Search Locations and Diseases
  - ReqB6.1: Search option displayed at the top of the page.
  - ReqB6.2: Can search by location only.
  - ReqB6.3: Can search by disease only.
  - ReqB6.4: Can search by both disease and location.
  - ReqB6.5: Search results act as a feed and update predictions, and map.
- ReqB7: Following
  - ReqB7.1: Users can follow/unfollow diseases.
  - ReqB7.2: Users can follow/unfollow locations.

Use Cases

Epic Story: ***As a*** *user,* ***I want*** *to quickly access articles and predictions,* ***so that*** *I can stay updated on recent news.*

Use Case 1	**View General Predictions**
Story	**As a** user, **I want** predictions displayed, **so that** I can see the predicted outcomes of current epidemics.
Requirements Covered	A1, A2, B2
Actors Involved	User
Basic Flow	User goes to the homepage. The feed on the homepage calls the prediction API, which calculates the prediction for a featured disease (whatever is most current) and a select few countries.
Alternate Flow	If the selected country does not have a prediction due to a lack of data, it will display there is insufficient data for that country with the specific disease.

Use Case 2	**View the Latest Articles**
Story	**As a** user, **I want** the most recent news articles displayed, **so that** I can keep up to date with the latest outbreaks around the world.
Requirements Covered	A1, A2, B1
Actors Involved	User
Basic Flow	User goes to the homepage. The feed on the homepage calls our API and other groups API to retrieve articles to populate the feed.

Use Case 3	**View Article in more Detail**

Story	**As a** user, **I want** to be able to click on an article to view a more detailed report, **so that** I can read into more detail and view the original source.
Requirements Covered	A1, B1.3, B1.4
Actors Involved	User
Basic Flow	User goes to the homepage. User then clicks on article title in feed to go to page with more detailed information on article, where there is also a link to the original source.

Use Case 4	**Map of Outbreaks**
Story	**As a** user, **I want** to be able to view a map of where outbreaks are, **so that** I can view where in the world the outbreak is.
Requirements Covered	A1, A2, B3
Actors Involved	User
Basic Flow	User goes to the homepage or article page or search results. The site loads the map, which is populated with markers as to where outbreaks are from the feed.

Use Case 5	**Map of Outbreaks Details**
Story	**As a** user, **I want** to be able to identify which disease the marker on the map is referring to, **so that** I can learn which disease is affecting which country.
Requirements Covered	A1, B3
Actors Involved	User
Basic Flow	User goes to the homepage or article page or search results. The site then loads the map. The user then clicks on the marker which displays the disease and location.

*Epic Story: **As a** user, **I want** to search for particular locations/diseases, **so that** I can find articles of relevance.*

Use Case 6	**Search by Disease**
Story	**As a** user, **I want** to search by disease, **so that** I can view articles about that disease.
Requirements Covered	A2.2, B6.1, B6.3
Actors Involved	User
Basic Flow	User goes to the homepage. The user selects the all option for countries and the specific disease in the disease option in the search section. User clicks search. The site then loads a set of results matching the specified criteria.

Use Case 7	**Search by Location**
Story	**As a** user, **I want** to search by location, **so that** I can view articles about that location.
Requirements Covered	A2.2, B6.1, B6.2
Actors Involved	User
Basic Flow	User goes to the homepage. The user selects the all option for diseases and the specific location in the location option in the search section. User clicks search. The site then loads a set of results matching the specified criteria.

Use Case 8	**Search by Disease and Location**
Story	**As a** user, **I want** to search by location and disease, **so that** I can view articles about that location and the disease affecting it along with relevant predictions.
Requirements Covered	A2.2, B6.1, B6.4, B6.5
Actors Involved	User
Basic Flow	User goes to the homepage. The user selects the specific disease for diseases and the specific location in the location option in the search section. User clicks search. The site then loads a set of results matching the specified criteria

	along with a section showing the relevant predictions.

*Epic Story: **As a** user, **I want** to have a custom feed, **so that** I can quickly view articles relevant to me.*

Use Case 9	**Login for the feed**
Story	**As a** user, **I want** to be able to login, **so that** I can view a feed more relevant to me.
Requirements Covered	A3, B5.1
Actors Involved	User
Basic Flow	User goes to the homepage. The user clicks on login. The site loads a page with text boxes for the user to input data. The user inputs their email and password. They then click login. Upon logging in, the user is redirected to the homepage which displays their personal feed.
Alternative Flow	If the email and/or password is incorrect, the user is alerted of this. The user can then re-enter their details to login.

Use Case 10	**Signup up for the feed**
Story	**As a** user, **I want** to be able to sign up, **so that** I can get a personalised feed more relevant to me.
Requirements Covered	A3, B5.2
Actors Involved	User
Basic Flow	User goes to the homepage. The user clicks on login. The site loads a page with text boxes for the user to input data. The user inputs their email and password. The user clicks sign up. Upon signing up, the user is redirected to the homepage which displays an empty personalised feed.
Alternative Flow	If the email has already been used, the user is alerted of this. The user can then re-enter their details to sign up.

Use Case 11	**Logout**

Story	**As a** user, **I want** to be able to logout, **so that** no one else can access my account.
Requirements Covered	A3, B5.3
Actors Involved	User
Basic Flow	User clicks on logout. The site logs the user out.

Use Case 12	**Follow Disease**
Story	**As a** user, **I want** to be able to follow a specific disease, **so that** my personalised feed contains articles related to diseases I'm interested in.
Requirements Covered	A3.4, B7.1
Actors Involved	User
Basic Flow	Begins with the user already logged in. The user uses the search function to search for a specific disease. At the top of the search results the user then clicks on the button to follow the disease. The site updates the users personalised feed to contain articles related to that disease.

Use Case 13	**Follow Location**
Story	**As a** user, **I want** to be able to follow a specific location, **so that** my personalised feed contains articles related to location I'm interested in.
Requirements Covered	A3.4, B7.2
Actors Involved	User
Basic Flow	Begins with the user already logged in. The user uses the search function to search for a specific location. At the top of the search results the user then clicks on the button to follow the location. The site updates the users personalised feed to contain articles related to that location.

Design

Changes Made from Feedback

In the original concept there was no prediction mechanism proposed. Upon receiving feedback from our tutor that perhaps another novelty feature is required, we decided to add the prediction feature. Additionally, we received feedback to incorporate more data sources to fill in the gaps of information. Although we did not implement this exact method for filling in gaps of data due to the difficulty of parsing other groups data, we instead opted to use averages to fill in that data.

We originally had endpoints for adding and removing articles, however our tutor suggested we add a lock to prevent malicious users from breaking our data source. We achieved this through a private API key which must be provided for those endpoints.

Key Benefits

The key benefits of this design includes its simple design. This allows for easy navigation and lets users view data more easily. Furthermore, that data is linked to the map as markers allowing users to gain a greater grasp at the spread of the disease. Those markers also show the disease and exact location. The data is also linked to the prediction as that gets its data from the reports.

The predictions provided are accurate for the data/model provided. The level of accuracy increases with access to a larger amount of data to analyse. It also allows users to generate a prediction based on a specified country and disease.

Although the only aim of this site is to provide analytic information on diseases in various locations, it is done well. It serves a way of gathering data on diseases and viewing its scope on a global level, along with providing the necessary reports for outbreaks.

Another major benefit is that users can create accounts. These accounts allow the users to specifically follow certain diseases and locations. This in turn keeps users up to date with current outbreaks related to the diseases and locations followed through the personalised feed.

Organisation

How The Project Went

Major Achievements in Project

A working prediction, that employs the use of the regression library that uses a least-squares fitting algorithm to provide results to all users using the application. This took weeks to implement and involved figuring out how to parse data of varying layouts. Furthermore, no one in our team had any experience in implementing something this complex and we were overall impressed with the end result.

Even though other teams would only need to access the search endpoint of our API. We made the decision to extend the feature set, and add more endpoints to make a proper RESTful API. We added endpoints to add, edit and delete articles, and get articles more efficiently then just searching, which allowed for the frontend to have easier interactions with the API and furthermore allowed for easier debugging and testing during development.

A working news feed consisting of reports from all cities, countries, diseases and syndromes which the user has chosen to follow. Following this the implementation of maps in the user interface so users can visually see the locations of diseases and syndromes on a map. These two features were combined so any report from the feed or search results can be viewed on the map as markers based on location. This together created a better overall looking UI design where multiple features could be interconnected and related.

Issues/problems encountered

An issue encountered was the quality of the data and amount of it from our allocated source. This caused issues in sections such as prediction and providing overall information to users. Furthermore, the reports did not cover the true extent of the outbreak when compared to other data sources such as ProMed and CDC.

The other teams API's data and overall format were different to ours. As a result, we had to write extra code to get their data to conform to our format. This also caused issues with trying to read their data for prediction as the data was too difficult to parse or just did not contain the data needed. Another issue encountered was that those API did not contain longitude and latitude, so we had to use the geonames API to obtain that information.

Furthermore, with extracting data for prediction it was difficult to obtain exactly what was needed just from our data source. Also there would be gaps in information for certain countries due to the lack of information from the allocated data source. To overcome these, any information where data could not be properly extracted would be ignored and areas with slight gaps in information would be filled in with averages based on before and after events for a particular date.

Skills we wish we had before workshop

One major skill needed prior to this workshop would be skills in a frontend framework for all members of a project. For example, React which is also an industry standard would prove to be beneficial for all members. This is greatly needed as it heavily relies on the use of websites to display data and interact with. This would allow for production of the frontend at a more early stage, which allows the extra time to be dedicated to producing a better product or researching extra technologies to enrich the project.

A guide to using data visualisation would be needed as this project is based on collecting and displaying data in a variety of formats.

Some basic level of knowledge in the use of AI as it was heavily suggested to be incorporated for processing data and performing predictions. If AI was known, it would also serve as a good learning experience in applying that knowledge to try to implement within an area that is still undergoing PHD level research.

Things that could have been done differently

When developing the frontend, a number of things could have been done differently, such as:

- Using a component library such as bootstrap to reduce development time.
    - We did not use a component library in the development of this project, in favor of building all components ourselves in react. While this was beneficial in terms of deepening our knowledge with the react framework, it did cost us development time.
- Beginning the development of the frontend earlier in the development cycle.
    - We did not begin developing the frontend until the API was complete. This turned out to be a poor decision as there were many portions of the frontend that did not require the API to be complete and thus could have been developed in parallel to the API, and not doing so cost us wasting time that could have been devoted to further development on the frontend.
- Deciding on the final technology stack sooner.
    - Some portions of the technology stack such as which map API we were going to use and which teams API's were going to be incorporated were not decided until fairly late in development. Deciding these details earlier would have led to a clearer development path and saved us some time during the development.

- Incorporating more teams API’s into the prediction API
    - The main issue faced with our novel feature, the prediction API, was that we didn't have enough data to ensure completely accurate predictions. This could have been resolved by incorporating more data sources by using more teams API’s to increase the pool of articles and thus data that could be used to make predictions. Given more time, we would have implemented an interface for maintainers to add more API’s to our system efficiently. Abstracting the data sources from the main system would allow for a quicker expansion of the data sources and reduce the downtime for updating the system.