PBL 3 - SE

Group G (Geopath) - Final Report

September 1, 2024

Course Name:	PBL 3: Creative Design - Software Engineering

Name	**Student ID**
Li Muting	2600220461-3
Miao Ruolin	2600220464-8
Ito Daiki	2600220034-0
Biligsaikhan Khurtsbayar	2600220460-5
Khoo Zhenyu	2600220458-3

1 Introduction

Responsible Member: Biligsaikhan Khurtsbayar

Geopath is our team's approach to providing health-conscious people an opportunity to protect themselves and their loved ones from the consequences of being exposed to the environmental contamination known as air pollution. Armed with various methods of data visualization, Geopath aims to help people make informed and conscious decisions on when and where they can spend their time to avoid such harmful environments and live their lives little worry-free from sudden unknown illnesses and diseases that could occur due to air pollution.

2 Literature Review

In this section, out of the ten literature reviews done by the team members during the project duration, only four were chosen and summarized as they were the most significant and impactful literature reviews affecting the project.

2.1 The combination of map and air quality

Responsible Member: Miao Ruolin

Common air pollution combined with map app production difficulties are mainly divided into two points, one is the GPS problem, the other is the detection of air quality issues. In terms of GPS, Google, for example. First, rough maps are generated from satellites. Second, the company sent Google cars to survey all passable streets, counting the locations of traffic lights and so on, and marking these locations on a map in the form of dots. The positions of these points are defined using a spherical three- dimensional coordinate system. This three-dimensional coordinate system can be projected onto a map to form two-dimensional coordinates[1].
When it comes to detecting air quality, Pollutant emissions can be calculated by the formula E=L*C, where E represents the emission of pollutants, L represents the number of kilometers traveled by the vehicle, C represents the emission coefficient of pollutants. Although the above two problems have been reasonably solved, there are still limitations that are not enough money and time cannot follow the above steps to develop software. So, it is a good choice to use an API in order to reduce these limitations[2].

2.2 Air pollution's effect on pre-existing health conditions

Responsible Member: Biligsaikhan Khurtsbayar

Air pollution's effect on pre-existing health conditions: This literature review will discuss pre-existing health conditions and their effect on air pollution mortality rates. According to World Health Organization (WHO), around 92 percent of the world's population lives in

areas exposed to air pollution levels exceeding the WHO's recommended pollution exposure amount[3].
As high as these numbers are, according to multiple researchers in this field, researchers found a higher risk of mortality due to air pollution in groups of people with certain health complications. These health issues are cardiovascular, type-2 diabetes, and respiratory diseases. People with pre-existing cardiovascular disease are 1.8 times more likely to die from air pollution than healthy individuals[4]. People with type-2 diabetes were twice as much at risk of dying as those without diabetes[5]. People with pre-existing respiratory disease are 2.5 times more likely to die from air pollution than healthy individuals[4].

2.3 Cross-platform Development

Responsible Member: Khoo Zhenyu

Cross-platform development will be involved in the development process of mobile apps. Cross-platform development refers to "one set of code, running on multiple platforms". Cross-platform development defined by Charkaoui et al. in a research paper, cross-platform development is made possible by using the same code that can be deployed to multiple platforms and thus avoid needing to develop the same application multiple times, in different languages[6].
In addition to that, according to a paper done by Bishop and Horspool as compared to a natively developed application, which requires maintenance for each separate application made for each platform, cross-platform developed applications are much simpler and easier to maintain[7].

2.4 Air pollutants and AQI

Responsible Member: Li Muting

The United States Environmental Protection Agency (EPA) has established National Environmental Air Quality Standards (NAAQS) for six air pollutants[8]. These pollutants include sulfur dioxide (SO2), nitrogen dioxide (NO2), carbon monoxide (CO), ozone (O3), lead (Pb) and aerodynamic diameter less than or equal to 2.5 micrometer (PM2.5) and 10 micrometer particles (PM10). These gases and particles have significant impacts on climate change, ecosystems, human health, cultural heritage, and the economy[9]. Therefore, analyzing the concentration of polluting gases is important for protecting human health. The Air Quality Index (AQI) is an indicator of daily air quality. It is calculated based on the concentration of major pollutant gases and divided into six health problem levels[10].

3 Content Section

3.1 Requirements

Responsible Members: Li Muting, Biligsaikhan Khurtsbayar

To determine the application requirements for GeoPath, a survey consisting of 17 questions was sent to potential users of the application via Google Forms. The survey included open-ended, close-ended multiple choice, and Likert-type questions concerning the users' openness to sharing their information, knowledge on air pollution, and preferences in applications such as Geopath. Through analyzing the answers from 15 respondents and discoveries made during the prototype implementation process, our team determined the following requirements for developing the application:

- The application requires users to share their sensitive locational information to provide them with relevant air pollution data.
- The application requires information on users' health conditions to assess the air pollutant effect on their bodies.
- The application requires users to share their destination location to provide a route from their current location and additional information concerning the destination location's air quality.
- The application is required to provide information such as an AQI level table and pollution component analysis to enhance users' understanding of relevant knowledge.
- The application is required to provide air quality data via data visualization methods such as tables and graphs.
- The application requires an interactive map, which displays relevant air pollution information and provides advice to users on the safety of their location based on their physical condition information through the map.
- The application requires easy-to-use interface that securely stores user data to facilitate users' safe and convenient use on various platforms.
- All the sensitive user information collection done by the application requires user permission.

3.2 Project Management

Responsible Member: Miao Ruolin
The project is managed in accordance with agile principles.

3.2.1 Roles

The team consists of the following roles:

- Project Manager: While dividing the work of the project, the project manager is also responsible for the diagram designs and data searches and integration and also submits the data to the coder for programming.
- Literature Reviewer: While summarizing literature reviews is the main focus, the literature reviewer is also responsible for diagram design and website interface design.
- Communication Officer: The communications officer is responsible for arrangements and reminders of the start time of each daily scrum and sprint review, submit the work of each stage, and assist the coder in their work whenever necessary.
- Senior Coder: The senior coder is responsible for formulating an efficient way to implement the application features, handling the entire back-end coding of the application to implement those features, and communicating with the junior coder.
- Junior Coder: The junior coder is responsible for the front-end development, including working with the Literature Reviewer on the design, as well as creating and maintaining the team's project website. The junior coder is also in constant communication with the Senior Coder.

3.2.2 Sprints

This project is divided into 4 sprints:

- Determine the basic function of the website and the preliminary design of the website layout.
- Find the relevant datasets and importing it to the website.
- Design the diagram while improving the web site as well as doing the final optimizations.
- Completing the final optimization and final touch ups.

3.2.3 Daily Scrum

The details of the daily scrum meetings is as follows:

- Purpose: To learn about the previous day's work progress and difficulties encountered by team members and to discuss the next day's goals.
- Duration: 15 minutes per session
- Location: CC205 / Online Zoom meetings

3.3 Diagrams

For application design, activity diagrams, flow charts, system architecture diagrams, use case diagrams and state diagrams were initially drawn and is accessible through the project website. The activity diagram and flow chart are drawn based on the process of using each function of the application by the user. The use case diagram and system architecture diagram reflect the functionality and architecture of the entire system.

3.3.1 System Architecture Diagram

Responsible Member: Li Muting
(Figure 1, *Responsible Member: Li Muting*)

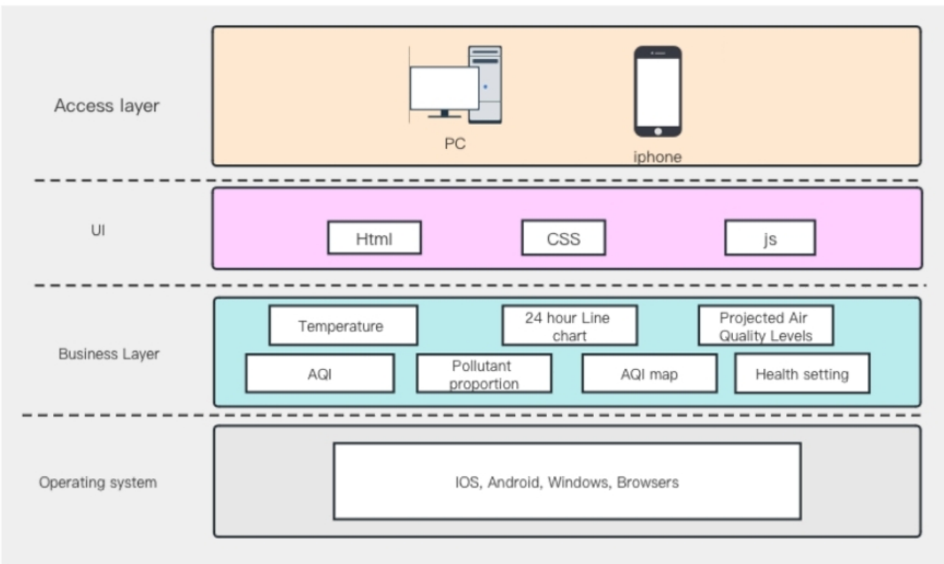

Figure 1: System Architecture Diagram

The system architecture diagram in Figure 1 shows that the application supports both PC and smartphone usage. The UI design uses HTML, CSS, and JavaScript. The functions of the application are mainly divided into seven parts, all of which are to help users understand relevant information about air quality, strengthen the popularization of air quality related knowledge, and help users better maintain physical health. As this application is a web-based application, it supports multiple operating systems and browsers.

3.3.2 System Activity Diagram

Responsible Member: Biligsaikhan Khurtsbayar
(Figure 2, *Responsible Member: Biligsaikhan Khurtsbayar*)

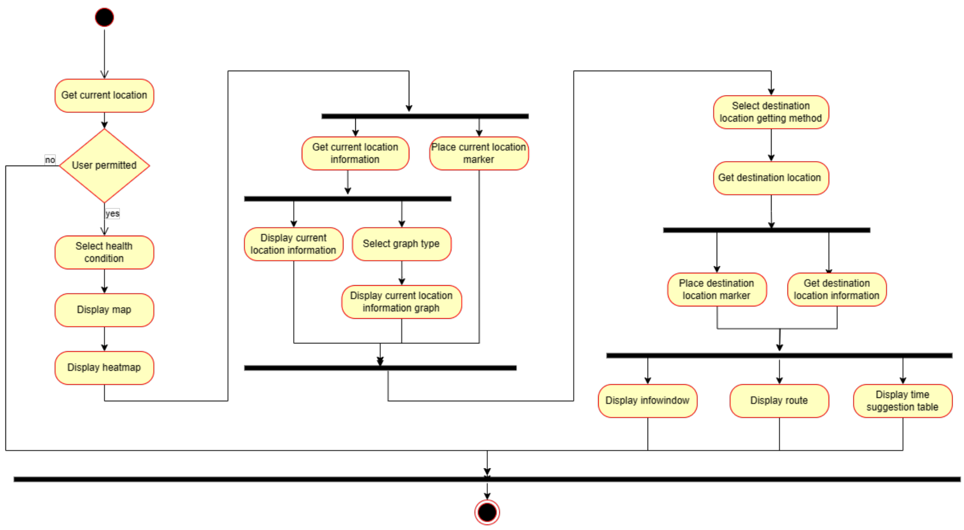

Figure 2: System Activity Diagram

The system activity diagram in Figure 2 illustrates all the activities the prototype application goes through to access all its features. As shown in the diagram, if the user does not permit to access their locational information, the application is unable to access all of its features, thus ending its activity. Explanations regarding the features, shown in Figure 2, will be included in section 3.6.

3.4 Project Website

Responsible Member: Khoo Zhenyu

The project website was written in HTML, CSS, and JavaScript, and was published online via GitHub Pages. It contains a home page, the team's literature reviews, weekly reports and updates, the link to the survey, the list of requirements, the design and concepts, the diagrams, and lastly the application itself. The project website is accessible through this link.

3.5 Application - Front-end

Responsible Member: Khoo Zhenyu

The objective of the application's front end design was to make a simple and information-rich interface. The team agreed that the application should be kept to a one-page design. As such, the elements of the application were kept as compact as possible, without compromising on the information.

The application is written in HTML and CSS, and the interactive aspects were coded in JavaScript.
The user first sees their location's AQI and temperature when they open the application, as well as a map reflecting their current location. From there, they will be able to choose their destination using the search box or via the map. They will also be able to select, if any, their medical condition.
Upon choosing their destination, they will be able to view the location's AQI. The application will also determine, based on the medical condition provided, if the location is safe enough for the user to go to, based on a multiplier made by the team. The map also reflects the AQI of locations using colored zones.
At the bottom of the page, there is a 'More Information' button. Upon clicking, a modal window will pop up, reflecting more detailed information, such as a table explaining the colored zones on the map, projected AQI levels of the destination selected, as well as an interactive graph showing both AQI levels and the components of the air, including PM10, PM2.5, and CO.
The choice of the User Experience/User Interface (UI/UX) design language was based on the existing project website's design elements. The colors selected were a dark gray background and a white font. More muted colors were selected in favor of other colors, so as to not clash with the many colors that would be reflecting on the map.
A modal window is used to display extra information. While the team wanted to include as much information in one page as possible, the extra information were packed away into the modal window to minimize clutter, but at the same time achieve the one-page design that the team wanted.
The website was designed with mobile users in mind. While it is desktop-friendly, the target consumers are mobile users, due to the nature of the application. As such, the application was streamlined in one column to fit mobile devices.

3.6 Application - Back-end (Features)

Responsible Member: Biligsaikhan Khurtsbayar

To implement the application's features, we utilized three different APIs: Google Maps API, Meteo API, and Aqicn API. For displaying the air quality information via an interactive map, Google Maps API was used in development over its counterparts as it provided well-written documentation on its webpage. Google Maps API were crucial in developing the features per our team's wants and needs. For utilizing the locational information provided by the users to provide air quality information, Meteo API was used in development over its counterparts due to its ability to provide hourly updated air quality information and week-long predictions for air quality. As for Aqicn API, it was only chosen for development for its ability to provide daily global air quality estimates through its Tiles API.

3.6.1 Current location information

For implementing the functionalities concerning the user's current location, the application uses the navigator.geolocation.getCurrentPosition() method to request their current location. When the user gives the application permission to use it through a web request, the method pulls the current location in latitude and longitude form. After the application receives the locational information of the user, the application passes it to the Meteo API fetch URL to receive various data related to the location in JSON form. From those fetched data, the location's current AQI and temperature are chosen and displayed above the map, as shown in Figure 3.

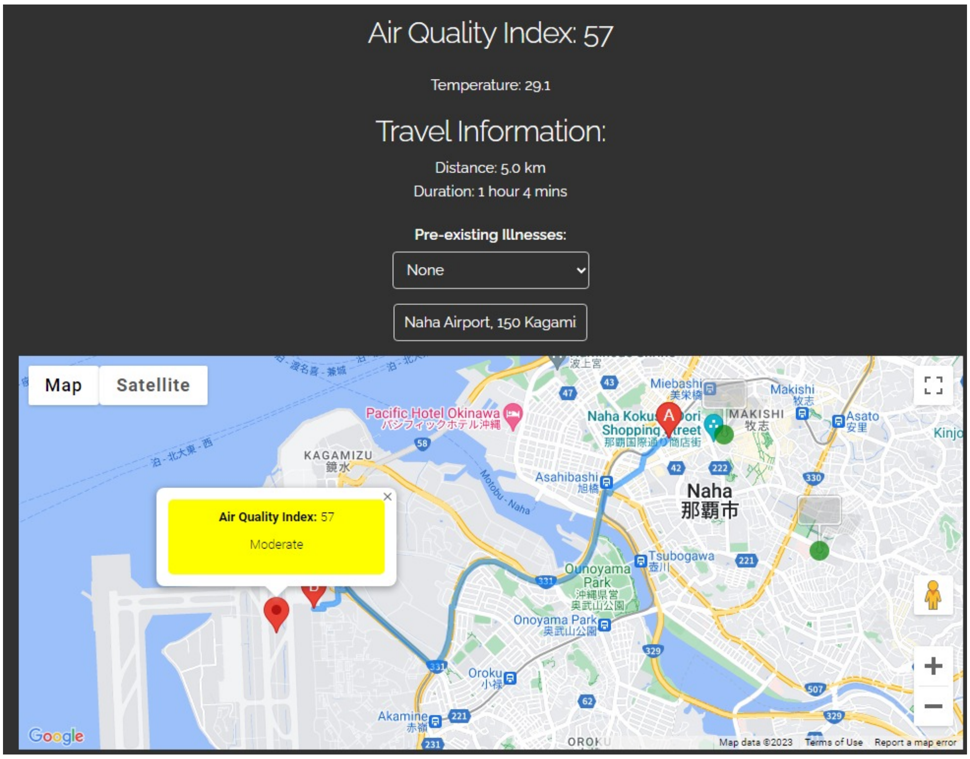

Figure 3: Current location information, route, and infowindow example

3.6.2 Daily prediction graphs

In the application, the daily predictions graph shown in Figure 4, 5 can either show the day's hourly AQI or the component concentration values depending on the values received from the graphs type dropdown list above it. The application implements this feature by utilizing the previously mentioned Meteo API fetched data, as its JSON contains 168 hours of data starting from the current day. By utilizing for loop, the application extracts the first 24 hours of this data and displays it as a graph using plot.ly. The graph-type switch is achieved via a conditional statement passing the necessary values to be plotted depending on the values received from the dropdown list.

Figure 4: AQI graph example

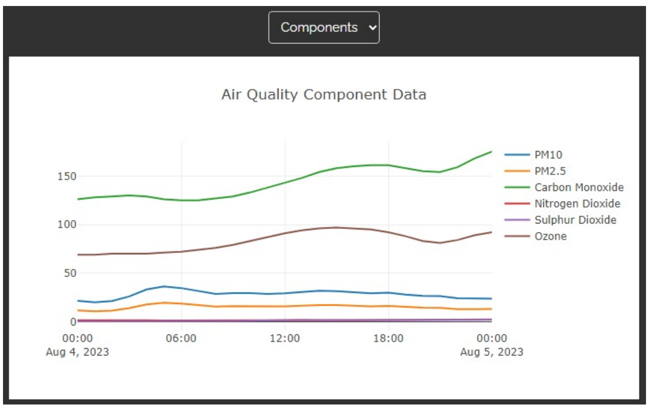

Figure 5: Component graph example

3.6.3 Route calculation

The current location previously received via web request is displayed on the map as an origin marker. As for the destination marker, there are two different ways to put it on the map. The first way to put the destination marker on the map is through an event listener on the map. When a click occurs on the map, the event listener listens and places a marker on the clicked location. The second way to put the destination marker on the map is through the searchbox built-in to Google Maps. When the user search for a place in the searchbox and the place the user searched for is a legitimate place, the searchbox listener places a marker on the searched place location. Through the use of origin and destination marker coordinates, the application uses Google Maps Directions and Routes API to automatically calculate the distance, walking travel time, and the route between the two points. The calculated route between the two points is shown in Figure 3.

3.6.4 Destination AQI infowindow

Same as the searchbox, there is another built-in Google Maps utility called infowindow. The infowindow is customizable and typically used for displaying marked location information on the map. In our application, the infowindow is customized to provide the AQI value, message, and coloring, as shown in Figure 6, instead of the general information concerning the marked location. AQI value for the destination marker is fetched using Meteo API by passing the destination coordinates to its URL. By writing a few conditional according to the information in Figure 6 (The table in the figure is made following Air Quality Index scale as defined by the US-EPA 2016 standard) and passing everything into the infowindow as a content string, the application displays the result shown in Figure 3.

Category and Colour	Range	Health Implications
Good	0 - 50	Air quality is considered satisfactory, and air pollution poses little or no risk
Moderate	51 - 100	Air quality is acceptable; however, for some pollutants there may be a moderate health concern for a very small number of people who are unusually sensitive to air pollution.
Unhealthy for Sensitive Groups	101 - 150	Members of sensitive groups may experience health effects. The general public is not likely to be affected.
Unhealthy	151 - 200	Everyone may begin to experience health effects; members of sensitive groups may experience more serious health effects.
Very Unhealthy	201 - 300	Health warnings of emergency conditions. The entire population is more likely to be affected.
Hazardous	301 - 500	Health alert: everyone may experience more serious health effects.

Figure 6: AQI threshold table

3.6.5 Time suggestion table

The time suggestion table is another feature in our application which utilises the destination marker location. As previously mentioned, the Meteo API JSON data contains 168 hours of AQI information starting from the current day. By writing a simple maximum and minimum finding algorithm, the application finds the best and worst hours for visiting the location for the next seven days. In addition, by initially extracting the starting 24-hour data from the JSON, the application does the same and determines the best and worst hours for visiting the location for the day. The table displays the results as shown in Figure 7.

Projected Air Quality Levels		
Destination AQ Information	**Today**	**Next 7 Days**
Worst AQ Level	12 am: 57	Thursday 11 am: 64
Best AQ Level	3 am: 56	Friday 2 pm: 50

Figure 7: Time suggestion table

3.6.6 Heatmap

The team initially tried implementing the heatmap using the Meteo API by plotting randomly generated points around the map and taking its AQI. However, we searched for a different API for the feature due to the API limitations and inefficiency of the approach. After thorough searching, the heatmap implementation used the Aqicn Tiles API instead of the Meteo API. The Tiles API provides the application with a globally ranged, daily updated overlay for the map. As our image for the feature included a heatmap layer, we formulated an approach which implements the heatmap using the Tiles API. First, the application marks all Tiles API points with the map markers. Then, by using the AQI value provided by the API for the point, it adjusts the marker radius and color according to the index value. An example of this heatmap application can beee sin in Figure 8.

3.6.7 Illness index multiplier

The illness index multiplier is the main feature of our application. Through this feature, the users can assess the potential risk their surroundings possess to them regardless of their pre-existing illness. The multipliers and health condition options set for the illness dropdown list are according to the literature review in section 2.2. The illness multiplier works as an index weight multiplier for all other application feature that utilizes AQI. Each feature that uses AQI value has a multiplier variable called weight attached to it. When the value of the illness dropdown list changes, the listener triggers a conditional that adjusts the value of the multiplier, which consequentially modifies all other features that use the AQI value. An example of this feature modification can be seen from Figure 8, 9.

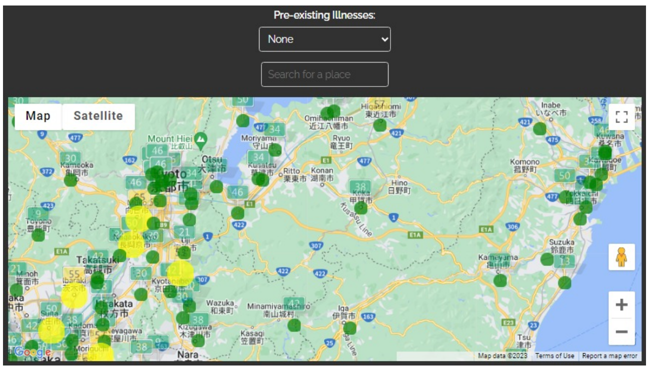

Figure 8: Healthy person heatmap

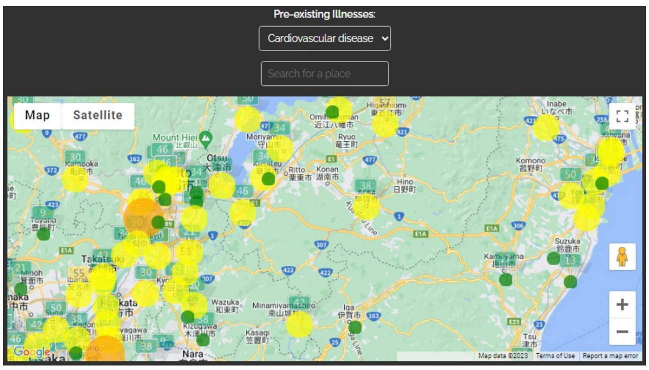

Figure 9: Cardiovascular disease heatmap

4 Analysis and Discussion (Technical Aspects)

4.1 Limitations

Responsible Member: Biligsaikhan Khurtsbayar, Khoo Zhenyu

In the process of making the prototype of our application, we faced and determined many limitations for our project. The following are the ones we have deemed the most limiting factor in reaching our desired results:

- Our team name is Geopath. The team was named as such because we wanted to provide the user with a path which avoids highly air-polluted areas. However, due to time constraints and a lack of programming knowledge, we were unable to implement this feature.
- Due to language barriers and a lack of teamwork between some team members, the team could not accomplish as much as we wanted within the project time limit.
- The Meteo API, the API we used for prototype implementation, was not as accurate as we desired. It only could provide us with a 10 km radius accuracy.
- Google Maps canvas memory was insufficient for displaying the AQI heatmap feature, resulting in lag issues.
- As seen in our prototype implementation, the color of the heatmap point does not diminish with distance from its center, meaning the region affected by the AQI point was not fully researched.
- Using HTML meant that many elements of the application were not as responsive as initially sought out to be.

4.2 Future plans

Responsible Member: Biligsaikhan Khurtsbayar

Considering the limitation stated previously and many other obstacles we discussed amongst the team, we formulated several future steps we can take to improve our application moving forward:

- Search for more accurate paid AQI API, or research for a way to combine multiple AQI API sources seamlessly.
- Research more on health multipliers and areas affected by AQI points.
- Research a way to implement the heatmap more efficiently.
- Research a routing algorithm which avoids the data overlay section of the map.
- Learn a new programming language for mobile app development for more convenient access to our application.

5 Conclusion

Responsible Member: Li Muting

In conclusion, our team have developed an application that helps users understand information related to air quality and provides information on the harm of air pollution to their health based on their physical condition. The purpose is to help more people understand the importance of air quality for health. As a team, we have also understood the importance of communication and teamwork as we have experienced how a lack of it can hinder the process of project progression.

6 Individual Reports

6.1 Report: Li Muting (Project Manager)

First of all, I would like to thank all the team members, they have worked very hard to create our website. Secondly, regarding myself, this is my first time serving as a project manager, so at the beginning, I was not sure what I should do. I just discussed the types and functions of the application with everyone and roughly distributed their respective tasks. After reading and summarizing the literature, I searched for suitable APIs and provided them to two coders. Afterward, I participated in the design of software and the drawing of diagrams. In the middle of software production, through communication with TA, I made clear some of the tasks that project managers should do. Then, the application development and final speech were completed through communication with various team members and workflow integration. Throughout the entire group collaboration process, I was not particularly satisfied with my performance. Although I have participated in various tasks, as a project manager, due to my lack of experience, I did not have timely and sufficient communication with all team members. Although our final application software was completed as scheduled and the speech was relatively smooth, I personally still had space for improvement throughout the entire process. In conclusion, this group work has provided me with valuable experience and ideas for similar work in the future.

6.2 Report: Miao Ruolin (Literature Reviewer)

In general, our group has done a good teamwork, and the division of labor is clear. Everyone can finish his work within the time limit. Collaborating with diverse perspectives taught me valuable communication and teamwork skills. Our project aims to contribute to the protection of the environment. Personally, I now feel a sense of responsibility as a global citizen. Though our project's scope was limited, our collective efforts created awareness and advocacy.

6.3 Report: Ito Daiki (Communications Officer)

First of all I would like to thank my team members who have corporate and created this crucial app and the website within the short period of time. For self assessment as a communication officer I mainly communicated with the project manager and divide the tasks needed to be done and named the app. I have also participated in the speech with the project manager to tell people how our app works. I am not much satisfied with my job and there is some improvements that could be made. However, this project has provided me various experiences that could be useful in the future.

6.4 Report: Biligsaikhan Khurtsbayar (Senior/Back-end coder)

During the latter half of the PBL3 course, I was assigned to Group G and was chosen to become the senior/back-end coder of the team. As I was the senior coder of the team, I expected from the start that I would be handling most of the team's work. However, the extent to which I am disappointed in some of my team members' performance was not within my expectations. Although the group had five members in total, either because of differences in priorities or just a lack of handling of responsibilities, the work amount managed by the members was not equal. To reach a point in the group project that I was satisfied with, I had to compensate and overwork myself despite having many other classes, similar to my team members.

Although I was aware of the language barrier between me and half of my group members, I thought I was getting my progression on the back end of the program across to my teammates during the feature explanation speeches I would give to the TA during our team meetings. I could indeed have been more assertive and demanding in my communication with them, but I failed to do so, as I was unfamiliar with such scenarios.

With all this considered, I want to thank my junior coder for implementing the application UI, making the project website exceptionally and being a communicator for me to the rest of the team who lacks English language expertise. As for myself, I implemented the entirety of the back-end/features of the application as I was responsible and ended up becoming responsible for most of the report sections and its finalization.

During this project, I have learned many things. I learned to solve programming issues through critical thinking, spot bugs in the code through testing and even report writing skills of mine have improved. Additionally, I learned to only put complete trust in myself and be more assertive in communication. I think the skills and experiences I have gained during the project will be helpful to me in the future, and thankful to my team for working with me.

6.5 Report: Khoo Zhenyu (Junior/Front-end coder)

I would like to thank my team members for their personal contributions to this project. For myself, I think that I fulfilled my job scope and purpose, which was to aid my senior coder, as well as constantly updating the project website, and being responsible for the UI/UX of the project. While UI/UX was a new thing for me, I personally enjoyed myself doing it and learned a lot from it.

References

[1] Wieczorek, W. F., Delmerico, A. M. (2009). *Geographic information systems. In The Geographic Information Science and Technology Body of Knowledge. Association of American Geographers.*

[2] Lin, M.-D. and Lin, Y.-C. (2001) *The application of GIS to air quality analysis in Taichung City, Taiwan, ROC. Environmental Monitoring and Assessment, 75(3), 273-291 .*

[3] G. P. Bălă et al. (2021). *Air pollution exposure—the (in)visible risk factor for respiratory diseases, Environmental Science and Pollution Research, vol. 28, pp. 19615-19628.*

[4] D. E. Newby et al. (2015) *Expert position paper on air pollution and cardiovascular disease, European heart journal, vol. 36, no. 2, pp. 83-93.*

[5] E. Thiering and J. Heinrich (2015) *Epidemiology of air pollution and diabetes, Trends in Endocrinology and Metabolism, vol. 26, no. 7, pp. 384-394.*

[6] S. Charkaoui, Z. Adraoui and E. H. Benlahmar (2014) *"Cross-platform mobile development approaches," 2014 Third IEEE International Colloquium in Information Science and Technology (CIST), Tetouan, Morocco, 2014, pp. 188-191, DOI: 10.1109/CIST.2014.7016616.*

[7] J. Bishop and N. Horspool (2006) *"Cross-Platform Development: Software that Lasts," in Computer, vol. 39, no. 10, pp. 26-35, Oct. 2006, doi: 10.1109/MC.2006.337.*

[8] J.H. Seinfeld, S.N. Pandis, 'Atmospheric Chemistry and Physics: From Air Pollution to Climate Change' https://books.google.co.jp/books?hl=zhCN&lr=&id=n_RmCgAAQBAJ&oi=fnd&pg=PR5&ots=gSNdELocfM&sig=lhmvglFua8qh0sbt12DVAov0Ss&redir_esc=y#v=onepage&q&f=false, Last accessed: July 2023.

[9] E. E. Agency, Air quality in Europe, 2016.

[10] İbrahim Kök; Mehmet Ulvi Şimşek; Suat Özdemir, 2017. *A deep learning model for air quality prediction in smart cities* https://ieeexplore.ieee.org/abstract/document/8258144/references#references, Last accessed: July 2023.

Appendix

The application prototype is accessible through this link.

Listing 1: Modal Window of GeoPath

```
//This is a snippet of the code showing how the modal window containing the extra information is written.
<div class="container">
  <button class="show-modal">More Information</button>
  <!-- Bottom sheet modal -->
  <div class="bottom-sheet">
    <div class="sheet-overlay"></div>
    <div class="content">
      <div class="header">
        <div class="drag-icon"><span></span></div>
      </div>
      <div class="body">
        <h3>More Information</h3>
        <div class="container">
          <h2>AQI Level Table</h2>
          <table id = "data-tablex" style="margin-bottom: 25px;">
            <tr>
              <th>Category and Colour</th>
              <th>Range</th>
              <th>Health Implications</th>
            </tr>
            <tr style = "background-color: green;">
              <td>Good</td>
              <td>0 - 50</td>
              <td>Air quality is considered satisfactory, and air pollution poses little or no risk</td>
            </tr>
            <tr style = "background-color: yellow; color: black;">
              <td>Moderate</td>
              <td>51 - 100</td>
              <td>Air quality is acceptable; however, for some pollutants there may be a moderate health concern for a very small number of people who are unusually sensitive to air pollution.</td>
            </tr>
            <tr style="background-color: orange; color: black;">
              <td>Unhealthy for Sensitive Groups</td>
              <td>101 - 150</td>
              <td>Members of sensitive groups may experience health effects. The general public is not likely to be affected.</td>
            </tr>
            <tr style="background-color: red; color: black;">
              <td>Unhealthy</td>
```

```
                  <td>151 - 200</td>
                  <td>Everyone may begin to experience health effects; members of sensitive
                      groups may experience more serious health effects.</td>
                </tr>
                <tr style="background-color: purple; color: black;">
                  <td>Very Unhealthy</td>
                  <td>201 - 300</td>
                  <td>Health warnings of emergency conditions. The entire population is more
                      likely to be affected.</td>
                </tr>
                <tr style="background-color: maroon; color: white;">
                  <td>Hazardous</td>
                  <td>301 - 500</td>
                  <td>Health alert: everyone may experience more serious health effects.</td>
                </tr>
              </table>
            </div>
            <div class="float-container">
              <h2>Projected Air Quality Levels</h2>
              <div class = "float-child">
                <table id="data-table" style="margin-top: 25px;">
                  <tr>
                    <th>Destination AQ Information</th>
                    <th>Today</th>
                    <th>Next 7 Days</th>
                  </tr>
                  <tr>
                    <td>Worst AQ Level</td>
                    <td>Row 1, Cell 2</td>
                    <td>Row 1, Cell 3</td>
                  </tr>
                  <tr>
                    <td>Best AQ Level</td>
                    <td>Row 2, Cell 2</td>
                    <td>Row 2, Cell 3</td>
                  </tr>
                </table>
              </div>
          </div>
          <div class="container">
            <div id="graph-type">
              <label for="graph-dropdown">Graph types:</label>
              <select id="graph-dropdown" onchange="onGraphChange()">
                <option value="AQI">AQI</option>
                <option value="Components">Components</option>
              </select>
            </div>
            <div class = "container" style="margin-bottom: 425px;">
              <div id="us-aqi-plot"></div>
            </div>
          </div>
          <div class = "container" style="margin-bottom: 50px;">
            <p2>Main particulate matter: PM2.5 particles are small enough to enter the
                bloodstream, usually originating from wildfires, chimneys, bacteria, or small
                dust particles.</p2>
          </div>
          </div>
        </div>
      </div>
    </div>
```

Listing 2: GraphChange function JavaScript Code

```
// This is the code responsible for handling the current location AQI and component graphs


  function onGraphChange() {
            var graphSelect = document.getElementById("graph-dropdown");
            var selectedGraph = graphSelect.value;

            if (navigator.geolocation) {
                navigator.geolocation.getCurrentPosition(function (position) {
                    var latitude = position.coords.latitude;
                    var longitude = position.coords.longitude;
```

```
                    if (selectedGraph === "AQI") {
                        fetch(`https://air-quality-api.open-meteo.com/v1/air-quality?latitude=
                            ${latitude}&longitude=${longitude}&hourly=pm10,pm2_5,
                            carbon_monoxide,nitrogen_dioxide,sulphur_dioxide,ozone,us_aqi`)
                            .then(response => response.json())
                            .then(jsonResponse => {
                                console.log(jsonResponse);
                                // Extracting the relevant data from the JSON
                                const usAqiData = jsonResponse.hourly.us_aqi;
                                const timeData = [];
                                for (let i = 0; i <= 24; i++) {
                                    timeData.push(jsonResponse.hourly.time[i]);
                                }
                                // Creating a div element for the US AQI value
                                const plotGraphDiv1 = document.getElementById('us-aqi-plot');
                                const plotData1 = [
                                    { x: timeData, y: usAqiData, name: 'AQI' },
                                ];
                                const layout1 = { title: 'Air Quality Index Data' };
                                Plotly.newPlot(plotGraphDiv1, plotData1, layout1);
                            })
                            .catch(error => {
                                console.error('Error:', error);
                            });
                    }
                    if (selectedGraph === "Components") {
                        fetch(`https://air-quality-api.open-meteo.com/v1/air-quality?latitude=
                            ${latitude}&longitude=${longitude}&hourly=pm10,pm2_5,
                            carbon_monoxide,nitrogen_dioxide,sulphur_dioxide,ozone,us_aqi`)
                            .then(response => response.json())
                            .then(jsonResponse => {
                                // Extracting the relevant data from the JSON
                                const pm10Data = jsonResponse.hourly.pm10;
                                const pm2_5Data = jsonResponse.hourly.pm2_5;
                                const coData = jsonResponse.hourly.carbon_monoxide;
                                const no2Data = jsonResponse.hourly.nitrogen_dioxide;
                                const so2Data = jsonResponse.hourly.sulphur_dioxide;
                                const ozoneData = jsonResponse.hourly.ozone;
                                const timeData = [];
                                for (let i = 0; i <= 24; i++) {
                                    timeData.push(jsonResponse.hourly.time[i]);
                                }

                                // Creating a div element for the plot graph
                                const plotGraphDiv = document.getElementById('us-aqi-plot');
                                const plotData = [
                                    { x: timeData, y: pm10Data, name: 'PM10' },
                                    { x: timeData, y: pm2_5Data, name: 'PM2.5' },
                                    { x: timeData, y: coData, name: 'Carbon Monoxide' },
                                    { x: timeData, y: no2Data, name: 'Nitrogen Dioxide' },
                                    { x: timeData, y: so2Data, name: 'Sulphur Dioxide' },
                                    { x: timeData, y: ozoneData, name: 'Ozone' },
                                ];
                                const layout = { title: 'Air Quality Component Data' };
                                Plotly.newPlot(plotGraphDiv, plotData, layout);

                            })
                            .catch(error => {
                                console.error('Error:', error);
                            });
                    }
                });
            } else {
                console.error('Geolocation is not supported by this browser.');
            }

        }
        window.addEventListener('DOMContentLoaded', onGraphChange);
```

Listing 3: Heatmap layer JavaScript Code

```
// This is the code responsible for creating the heatmap layer for the map

      var tileLayer = new google.maps.ImageMapType({
                getTileUrl: function(coord, zoom) {
```

```
                    var url = 'https://tiles.waqi.info/tiles/usepa-aqi/' + zoom + '/' + coord.
                        x + '/' + coord.y + '.png?token=4
                        c514b8dc8a3b3aa25b3557580c995c57b783b18';
                    return url;
                },
                tileSize: new google.maps.Size(256, 256),
                maxZoom: 18,
                opacity: 0.5 // Adjust the tile layer opacity as desired
            });

            map.overlayMapTypes.push(tileLayer);

            var markers = []; // Array to store the markers

            function updateMarkers() {
                var bounds = map.getBounds();
                if (!bounds) {
                    return; // Return if bounds are not available yet
                }

                var ne = bounds.getNorthEast();
                var sw = bounds.getSouthWest();
                var url = 'https://api.waqi.info/map/bounds/?latlng=' + sw.lat() + ',' + sw.
                    lng() + ',' + ne.lat() + ',' + ne.lng() + '&token=4
                    c514b8dc8a3b3aa25b3557580c995c57b783b18';

                fetch(url)
                    .then(response => response.json())
                    .then(data => {
                        // Clear previous markers
                        markers.forEach(function(marker) {
                            marker.setMap(null);
                        });
                        markers = [];

                        // Generate markers with colors based on the AQI tiles
                        data.data.forEach(function(tile) {
                            var latLng = new google.maps.LatLng(tile.lat, tile.lon);
                            var weight = tile.aqi || 0; // Use AQI as weight (or 0 if
                                undefined)
                          console.log(weight + "previous");
                            // Customize the marker color based on weight range
                           multiplier = onIllnessChange();
                           weight = weight * multiplier;
                           console.log(weight);
                            var markerColor = getMarkerColor(weight);

                            // Create a marker for each data point and set the color
                            var marker = new google.maps.Marker({
                                position: latLng,
                                map: map,
                                icon: {
                                    path: google.maps.SymbolPath.CIRCLE,
                                    fillColor: markerColor,
                                    fillOpacity: 0.7, // Set the marker opacity to 70%
                                    strokeWeight: 0,
                                    scale: getMarkerSize(weight) // Adjust the marker size
                                        based on weight
                                }
                            });
                            markers.push(marker);
                        });
                    })
                    .catch(error => console.error('Error updating markers:', error));
            }
```

Listing 4: Illness index multiplier JavaScript Code

```
//Sample code for illness index multiplier modifying the AQI color class conditional

  function getAirQualityColorClass(index,multiplier) {
    index *=multiplier;
    if (index <= 50) {
      return "green";
    } else if (index <= 100) {
```

```
      return "yellow";
    } else if (index <= 150) {
      return "orange";
    } else if (index <= 200) {
      return "red";
    } else if (index <= 300) {
      return "purple";
    } else if (index <= 500) {
      return "brown";
    } else {
      return "brown";
    }
  }
```