**Software Design**

**Document**

**for**

Artificial Intelligence and Data Science for Air Pollution Prediction and Visualization

**Version 2.0 approved**

**Prepared by** Alejandro Alatorre, Emmanuel Cocom, Hector Rucobo, Aldo Ruiz Cruz, Vidal Zazueta NASA and LA City

**December 10th, 2021**

**Table Of Contents:**

Table of Contents.................................................................................................................<pg 2> Revision History................................................................................................................. <pg 4> 1. Introduction.....................................................................................................................<pg 5>

1.1. Purpose......................................................................................................................<pg 5>

1.2. Document Conventions…………………….............................................................<pg 5>

1.3. Intended Audience and Reading Suggestions...........................................................<pg 5>

1.4. System Overview.......................................................................................................<pg 5>

2. Design Considerations..................................................................................................<pg 5>

2.1. Assumptions and dependencies.................................................................................<pg 5>

2.2. General Constraints.....................................................................................................<pg 6>

2.3. Goals and Guidelines...................................................................................................<pg 7>

2.4. Development Methods..................................................................................................<pg 7>

3. Architectural Strategies...................................................................................................<pg 8>

4. System Architecture.........................................................................................................<pg 9>

4.1. .......................................................................................................................................<pg 9>

4.2. .......................................................................................................................................<pg 9>

5. Policies and Tactics..........................................................................................................<pg 10>

5.1. Specific Products Used.................................................................................................<pg 10>

5.2. Requirements traceability..............................................................................................<pg 10>

5.3. Testing the software.......................................................................................................<pg 10>

5.4. Engineering trade-offs..................................................................................................<pg 11>

5.5. Guidelines and conventions...........................................................................................<pg 11>

5.6. Protocols.......................................................................................................................<pg 11>

5.7. Maintaining the software...............................................................................................<pg 11>

5.8. Interfaces....................................................................................................................<pg 11>

5.9. System's deliverables.................................................................................................<pg 11>

5.10. Abstraction...............................................................................................................<pg 11>

6. Detailed System Design

6.x Name of Module..........................................................................................................<pg 12>

6.x.1 Responsibilities.........................................................................................................<pg 12>

6.x.2 Constraints.................................................................................................................<pg 12>

6.x.3 Composition...............................................................................................................<pg 12>

6.x.4 Uses/Interactions.......................................................................................................<pg 12>

6.x.5 Resources...................................................................................................................<pg 12>

6.x.6 Interface/Exports........................................................................................................<pg 12>

7. Detailed Lower level Component Design

7.x Name of Class or File.................................................................................................... <pg 18>

7.x.1 Classification...............................................................................................................<pg 18>

7.x.2 Interface Description...................................................................................................<pg 19>

8. User Interface

8.1. Overview of User Interface..........................................................................................<pg 20>

8.2. Screen Frameworks or Images.....................................................................................<pg 20>

8.3. User Interface Flow Model..........................................................................................<pg 20>

9. Database Design……………………………………………………………………….<pg 21> 10. Requirements Validation and Verification....................................................................<pg 25> 11. Glossary..........................................................................................................................<pg 27> 12. References.......................................................................................................................<pg #>

**Revision History**

| Name | Date | Reason for Changes | Version |
| --- | --- | --- | --- |
| First Draft | 12/7/2021 | Initial Draft of Doc | 2.0.0 |
| Second Draft | 12/8/2021 | Detailed System Design Update | 2.0.1 |

**1. Introduction**

**1.1 Purpose**

This document aims to explain in detail the functions of the application. The document will inform the reader on the purpose, and functions of the application. The purpose of the application is to provide users with an easy to understand, user-friendly website with general air quality , weather, and pollution data. The application will also use machine learning and data science to provide predictions of air pollution and air quality.

**1.2 Document Conventions**

Bullet points will indicate sections being talked about. External links will be underlined blue.

**1.3 Intended Audience and Reading Suggestions (NEEDS UPDATE)**

The intended audience for the Software Requirements Specification document are software developers, project managers, and data scientists.

**1.4 System Overview**

**I. Air Pollution in Los Angeles County Data Visualization (Web App)**

● This application is used to show the measured levels of air pollutants and weather around Los Angeles County. Currency, the utilization of temporal layers in ArcGIS, accompanied by tile layers allows for visualization of air pollution predictions. The utilization of several APIs allows for live data and air pollution to be shown within the United States.

**2. Design Considerations**

The following section details the Assumptions and Dependencies, General Constraints, Goals and Guidelines as well as the Development Method of the applications.

**2.1 Assumptions and Dependencies**

**I. Air Pollution in Los Angeles County Data Visualization (Web App) ●** The following software uses the ArcGIS Online JavaScript API and assumes that any browser allows the proper permissions for it to work.

● The following software is reliant on the ArcGIS Online and assumes that the Feature Layers used remain public and in the same format

● The following software is reliant on using ArcGIS to represent data and assumes developers have an ArcGIS license

● The following software is developed by gathering data from various APIs and assumes that API keys are provided to the developers

**2.2 General Constraints**

Describe any global limitations or constraints that have a significant impact on the design of the system's software (and describe the associated impact). Such constraints may be imposed by any of the following (the list is not exhaustive):

Listed below are the general constraints for each application of this project.

**I. Air Pollution in Los Angeles County Data Visualization (Web App) ● Software Environment**

○ Developers are required to have a functioning ArcGIS Pro license

○ Developers are required to have basic knowledge of HTML

○ Developers are required to have basic knowledge of CSS

○ Developers are required to have basic knowledge of the JavaScriptlanguage

○ Developers are required to have knowledge of the React framework

**● End-User Environment**

○ A web browser is required to use and view the web application

○ A mouse, keyboard, monitor, and desktop/laptop are required for

user inputs and outputs.

**● Interoperability requirements**

○ Data is gathered from ArcGIS Pro

○ Data is gathered from various APIs

**● Performance requirements**

○ Application should load maps and their datasets in less than 5

seconds

○ Application should load graphs in less than 5 seconds

○ Application should load current weather in less than 5 seconds

○ Application should load current weather change in less than 5 seconds

**2.3 Goals and Guidelines**

Describe any goals, guidelines, principles, or priorities which dominate or embody the design of the system's software. For each such goal or guideline, unless it is implicitly obvious, describe the reason for its desirability. Feel free to state and describe each goal in its own subsubsection if you wish. Such goals might be:

**I. Air Pollution in Los Angeles County Data Visualization**

● The data gathered from ArcGIS should be accurate

● The GUI should be user-friendly

● The data gathered from the various APIs should be accurate

● The application should effectively visualize predicted and current data

**2.4 Development Methods**

The developers have split into two teams; one team is responsible for the development of the web application’s maps and the other team is responsible for the APIs and overall design of the website. Both teams split tasks among individual team members in order to complete small updates each week successfully. Regularly, bi-weekly meetings are conducted in order to assess the progress and changes that have been made and to further improve the website by allotting new tasks for the following meeting.

Briefly describe the method or approach used for this software design. If one or more formal/published methods were adopted or adapted, then include a reference to a more detailed description of these methods. If several methods were seriously considered, then each such method should be mentioned, along with a brief explanation of why all or part of it was used or not used.

These would be things such as the ‘Water Fall Development’ methods, ‘Agile Development’, ‘Unplanned Mad Scramble Development’, or other development models and variations. Describe how these were applied in the case of your project.

**3. Architectural Strategies**

Describe any design decisions and/or strategies that affect the overall organization of the system and its higher-level structures. These strategies should provide insight into the key abstractions and mechanisms used in the system architecture. Describe the reasoning employed for each decision and/or strategy (possibly referring to previously stated design goals and principles) and how any design goals or priorities were balanced or traded-off. Such decisions might concern (but are not limited to) things like the following:

**I. Air Pollution in Los Angeles County Data Visualization (Web App)** ● Use of a particular type of product (programming language, database, library, etc.)

○ ArcGIS

○ JavaScript

○ HTML

○ CSS

○ React

● Reuse of existing software components to implement various

parts/features of the system

○ This software was based on the 2021 Artificial Intelligence and Data Science for Air Pollution Prediction and Visualization teams software.

● Future plans for extending or enhancing the software

○ Adding predictions of air quality conditions

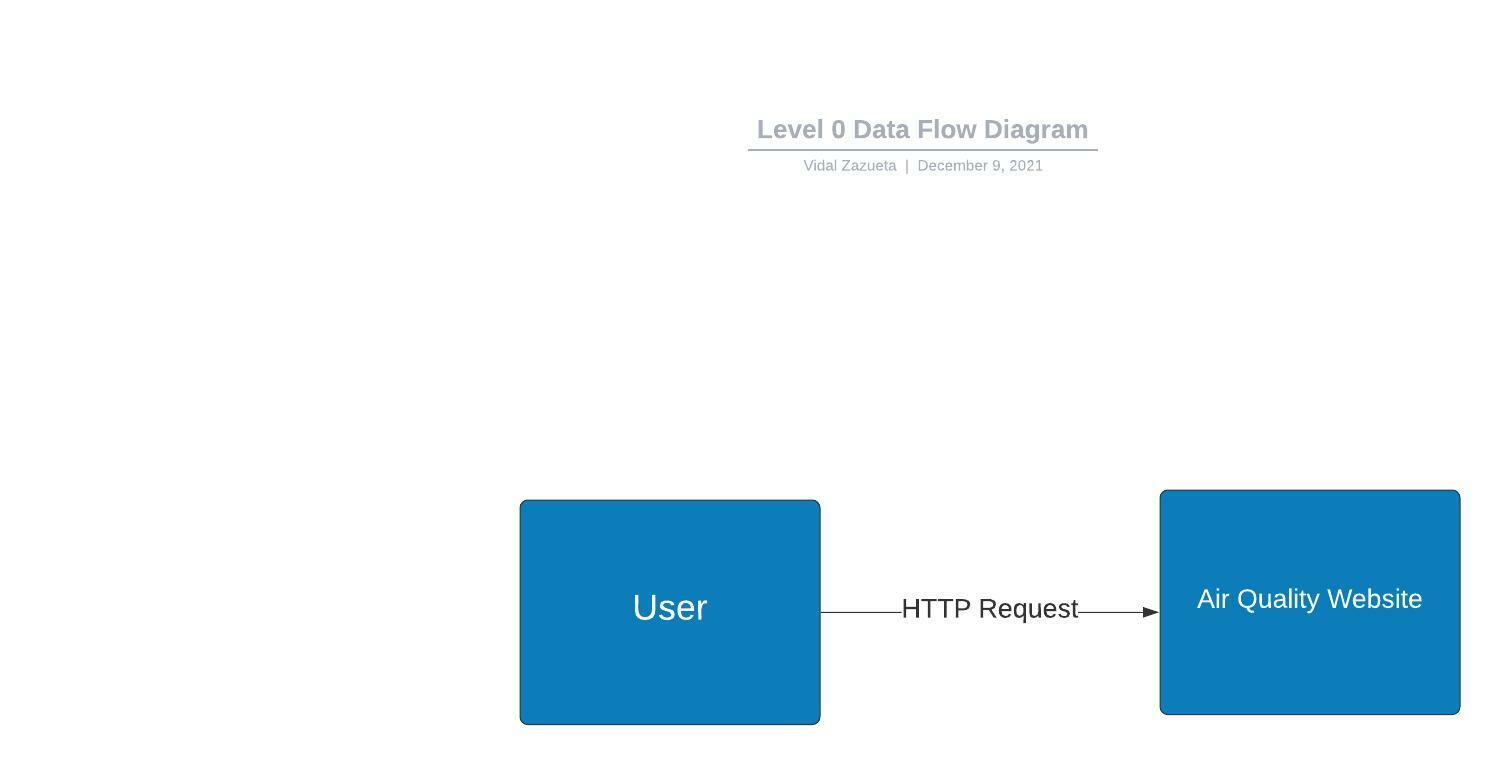
○ Adding updated graphs and gages to pages

● User interface paradigms

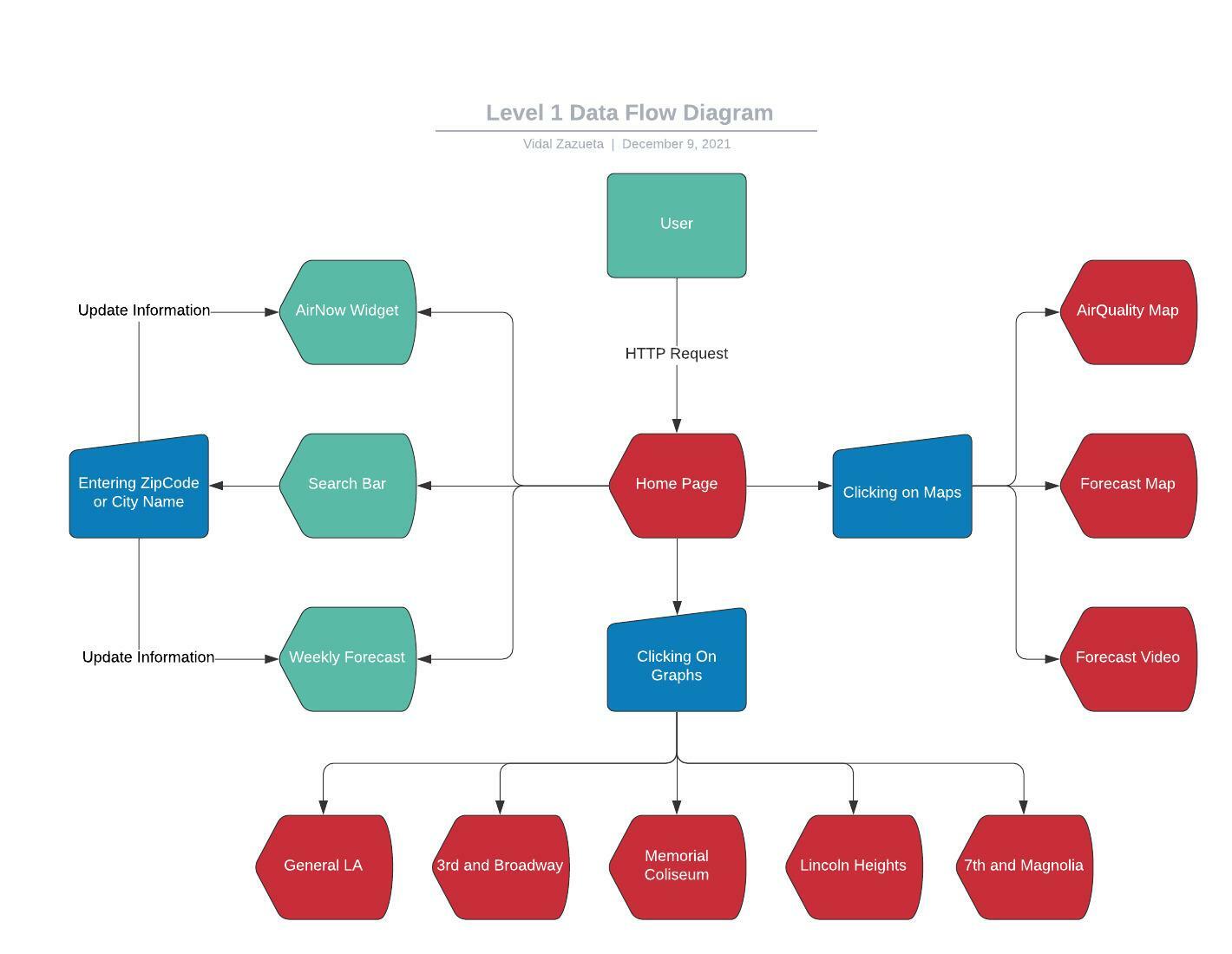
○ A computer and internet connection is required to use the

application as well as a mouse and keyboard to navigate the web application

**4. System Architecture**

****

Level 0 Data Flow Diagram (DFD)

****

Level 1 Data Flow Diagram (DFD)

**5. Policies and Tactics**

**5.1 Choice of which specific products used**

● Programming languages used:

○ JavaScript

○ HTML

● ArcGIS

**5.2 Plans for ensuring requirements traceability**

The requirements of the project are traceable via GitHub and the base code versions. Documents will follow System Design Description.

…Describe…

**5.3 Plans for testing the software**

The software will be continuously tested as new components and features are added and modified.

**5.4 Plans for maintaining the software**

The plans to maintain software are to fix bugs as they arise, and to collect and live update data as it becomes available.

**5.5 Engineering trade-offs**

* Due to issues pertaining to parent child information sharing a temporary search bar had to be implemented.
* A video had to be implemented temporarily due to problems with ArcGIS time slider systems.

**5.# Coding guidelines and conventions**

…Describe…

5.# The protocol of one or more subsystems, modules, or subroutines

…Describe…

**5.# The choice of a particular algorithm or programming idiom (or design pattern) to implement portions of the system's functionality**

* A simple segmented system with individual components was chosen to make it easier to move components from page to page
  + This system was also chosen to make web-app development simpler

**5.# Plans for maintaining the software**

* Routine updates will update data as it is received.
* Update product keys for weather APIs in order to maintain function.
* Renew licensing for ArcGIS to keep maps accessible.

5.# Interfaces for end-users, software, hardware, and communications

…Describe…

5.# Hierarchical organization of the source code into its physical components (files and directories).

…Describe…

5.# How to build and/or generate the system's deliverables (how to compile, link, load, etc.) …Describe…

5.# Describe tactics such as abstracting out a generic DatabaseInterface class, so that changing the database from MySQL to Oracle or PostGreSQL is simply a matter of rewriting the DatabaseInterface class.

For this particular section, it may become difficult to decide whether a particular policy or set of tactics should be discussed in this section, or in the System Architecture section, or in the Detailed System Design section for the appropriate component. You will have to use your own "best" judgement to decide this. There will usually be some global policies and tactics that should be discussed here, but decisions about interfaces, algorithms, and/or data structures might be more appropriately discussed in the same (sub) section as its corresponding software component in one of these other sections.

**6. Detailed System Design**

**6.1 Data Visualization**

**6.1.1 Responsibilities**

The primary responsibility of this module is to visualize data on a map and translate the data into UI elements such as symbols or polygons. This module should also ensure that all symbols are unique and that all polygons are colored based on intensity.

**6.1.2 Constraints**

Some datasets used for this module are not up to date or do not update in real-time which may lead to inaccurate representation. Datasets that are up to date may also have null fields which may cause certain data to not appear on the map.

**6.1.3 Composition**

The list below details the use and meaning of the subcomponents used in this module

● **Legend**: shows the different symbols and their meanings as well as the different colors for the polygons and their meanings

**● Toggle on/off:** allows for datasets to be toggled on/off and refreshes the map **● Add dataset:** allows for custom datasets to be added to the map given they are in the right format

**● Switch map:** allows for multiple maps with different datasets and different usage to be shown in the same application

**6.1.4 Uses/Interactions**

This module will be used to visualize the common causes of air pollution, the measured levels of air pollutants, the current demographics of the populus, as well as statistics on the effects of air pollution surrounding Los Angeles county.

**6.1.5 Resources**

The resources this module uses are datasets of Feature Layers that are present within ArcGIS’s Living Atlas as well as data gathered from various sensors surrounding Los Angeles county.

**6.1.6 Interface/Exports**

The visualized data will be displayed on the Air Pollution in Los Angeles County Data Visualization application via the use of the ArcGIS Online JavaScript API.

**6.2 Navigation Bar**

**6.2.1 Responsibilities**

Allows users to navigate between the 3 different options: Home, Graphs, Maps with a work in progress search bar

**6.2.2 Constraints**

Users must have an internet connection and an I/O device

**6.2.3 Composition**

Compositions consists Javascript for the syntax of the code, React for the functions, and CSS for the styling

**6.2.4 Uses/Interaction**

Consists of our logo, motto, and a search bar to search temperature and air quality by zipcode. It is a work in progress. Also contains 3 tabs: Home, Graphs, and

Maps.

**6.2.5 Interface/Exports**

**-**Javascript, CSS, and React

**6.3 Home**

**6.3.1 Responsibilities**

Allows users to see the current air quality at the specific hour and location and 5 week weather forecast

**6.3.2 Constraints**

Users must have an internet connection as well as an I/O device.

**6.3.3 Composition**

The composition process consists of javascript, html, css, and react.

**6.3.4 Uses/Interaction**

A user-friendly menu which consists of a widget from AirNow and a 5 day weather forecast with temperature.

**6.3.5 Interface/Exports**

**6.4 Search**

**6.4.1 Responsibilities**

Allows users to search for the air quality and weather forecast based on the zipcode input

**6.4.2 Constraints**

Requires a user input of the locations zip code or city name

**6.4.3 Composition**

The composition of the process required React and JavaScript

**6.4.4 Uses/Interaction**

A search bar above the weekly forecast. Entering a valid zip code will display the air quality through the gauge and the weekly forecast. The user can also input the city name which will also change the gauge and weekly forecast.

**6.4.5 interface/Exports**

-React

-JavaScript

**6.5 Graphs**

**6.5.1 Responsibilities**

Provides the user graphical data on 5 locations around Los Angeles regarding air pollution

**6.5.2 Constraints**

Requires the user to navigate between the locations

**6.5.3 Composition**

The composition of the process used JavaScript, React , and CanvasJs.

**6.5.4 Uses/Interaction**

A user-friendly page that displays graphs on 5 different locations. The graphs include an air quality graph, the air quality for the past 3 days, and an air quality trend from 1980. Users can click on the locations to view the graphs of that location.

**6.5.5 interface/Exports**

-JavaScript

-React

-CanvasJS

**6.6 Maps**

**6.6.1 Responsibilities**

Allows users to choose between 2 different maps and a forecast video. The maps include an air quality map and a forecast map.

**6.6.2 Constraints**

User must click on the drop down menu to choose between the 3 items

**6.6.3 Composition**

-React

-Javascript

-ArcGIS

**6.6.4 Uses/Interaction**

Clicking on the drop down menu will prompt the user with 3 options that the user must choose from

**6.6.5 interface/Exports**

-JavaScript

-ArcGIS

-React

**6.7 Air Quality Map**

**6.7.1 Responsibilities**

Displays an interactive air quality map with various sites around the globe but mainly in ‘

the United States. Additionally, includes a forecast air quality map to show machine learning generated sensor values.

**6.7.2 Constraints**

Users must have an internet connection and an I/O device to interact with the map.

**6.7.3 Composition**

The composition uses React, JavaScript, and ArcGIS

**6.7.4 Uses/Interaction**

Displays an interactive map built with ArcGIS. Users can zoom in and out of the map and navigate around the map. Users can click on any of the sites, identified by a colored circle, for more information such as current air quality, location, site name, etc. There is a legend on the bottom left to show what each site’s air quality is based on the color. Users can also use the search bar and input a location to be taken to that location.

**6.7.5 interface/Exports**

-React

-JavaScript

-ArcGIS

**6.8 Forecast Map**

**6.8.1 Responsibilities**

Users can see and interact with a map that is an air quality forecast respective to their

location. Currently, the forecast is limited to Los Angeles County and is a work in progress.

**6.8.2 Constraints**

Users must have an internet connection, javascript enabled and an I/O device.

**6.8.3 Composition**

Composition is made up of ArcGIS, React, and JavaScript

**6.8.4 Uses/Interaction**

The user can interact with the map by zooming in and out as well as moving around the map. To the bottom left is a slider with a play control scheme. Moving the slider will change the map by displaying the current air quality per hour. The map will change showing a colored layer above the map, from green to red. Clicking on play will show the forecast automatically per hour, as well as clicking on next and previous.

**6.8.5 interface/Exports**

-ArcGIS

-JavaScript

-React

**6.9 Forecast Video**

**6.9.1 Responsibilities**

Shows users an example of what the forecast page will look like

**6.9.2 Constraints**

Users must have an internet connection

**6.9.3 Composition**

Composition consists of JavaScript, React, CanvasJS and ArcGIS

**6.9.4 Uses/Interaction**

The page will display a video that displays the air quality per hour for October 14, 2021

**6.9.5 interface/Exports**

-React

-JavaScript

-ArcGIS

-CanvasJS

**7. Detailed Lower level Component Design**

Other lower-level Classes, components, subcomponents, and assorted support files are to be described here. You should cover the reason that each class exists (i.e. its role in its package; for complex cases, refer to a detailed component view.) Use numbered subsections below (i.e. “7.1.3 The ABC Package”.) Note that there isn't necessarily a one-to-one correspondence between packages and components.

**7.1 Airqualitylinechart**

**7.1.1 Classification**

This component is a .jsx file.

**7.1.2 Interface Description**

This component visualizes different levels of pollutants within the Los Angeles area on a line graph.

**7.2 Barchart**

**7.2.1 Classification**

This component is a .jsx file.

**7.2.2 Interface Description**

This component visualizes 3 different pollutant levels on a bar graph for the Los Angeles area.

**7.3 Broadway-live**

**7.3.1 Classification**

This component is a .jsx file.

**7.3.2 Interface Description**

This component draws live data for visualization specifically from the sensor on 3rd and Broadway.

**7.4 Lincolnheights-live**

**7.4.1 Classification**

This component is a .jsx file.

**7.4.2 Interface Description**

This component draws live data for visualization specifically from the sensor in Lincoln Heights.

**7.5 Magnolia-live**

**7.5.1 Classification**

This component is a .jsx file.

**7.5.2 Interface Description**

This component draws live data for visualization specifically from the sensor on 7th and Magnolia.

**7.6 Memorialcoliseum**

**7.6.1 Classification**

This component is a .jsx file.

**7.6.2 Interface Description**

This component draws live data for visualization specifically from the sensor in Memorial Coliseum.

**8. Database Design**

* Current Version 2.0 of PWWB’s APLAC does not utilize a database on a server since it is mainly a visualization of live sensor readings and manually and locally generated machine learning forecasts. Future Versions will include a database to keep track, manage, and display such data and revisions to include these changes will be made as needed.

**9. User Interface**

**9.1 Overview of User Interface**

When a user enters our website’s web address, they will be introduced to the home page. At the top of the website page is the Navigation Bar that holds 3 different tabs; Home, Graphs, and Maps. The Navigation Bar also holds our team logo and our project name, “Predict What We Breathe.” There is also a search bar that is a work in progress; the user can type in it but it will not return any results yet.

The home page will be the first tab any users will see when they visit our website, but can also be accessed when you are in a different page from the Navigation Bar. To the left of the page we have a widget from AirNow. The gauge displays the current air quality at the hour at the location it states, Los Angeles will be the default location. The gauge will also display what air pollutant it is reading and the gauge will move from left to right depending on the condition of the air quality. In the center is another search bar and a weekly forecast. The weekly forecast will show the weather as well as the average, maximum, and minimum temperature for the following days. Inputting a city name or zip code in the second search bar will update the AirNow widget and the weekly forecast, displaying the information for the specified location.

The graphs page will show 3 different graphs for each of the locations. The first graph is the current air quality for that location. The second graph is the air quality in the past 3 days for that location. The final graph is the change in air quality from 1980 to today. There are 5 different locations to select from: General Los Angeles, 3rd and Broadway, Memorial Coliseum, Lincoln Heights, and 7th and Magnolia. Clicking on the name of either of these locations will display the 3 graphs previously described.

The Maps tab will show a drop down menu displaying 3 items; Air Quality Map, Forecast Map, Forecast Video.

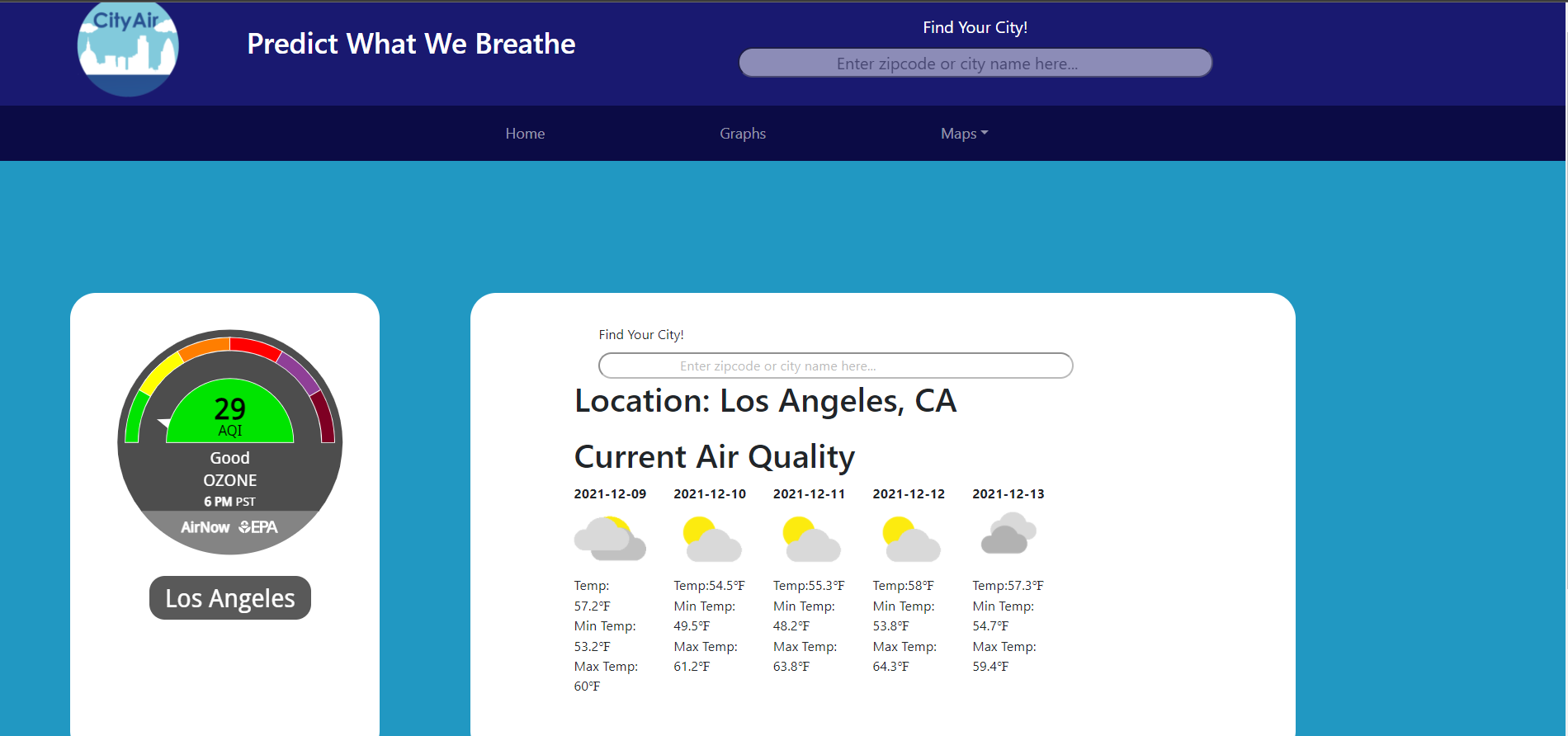
The Air Quality Map is an interactive map built with ArcGIS. Users can do basic functions such as zooming in and out as well as moving the map around. Users can also click on any of the sites, identified by a colored circle, to view more information from that site. There is also a legend on the bottom left to identify the sites. Users can use the search bar at the top right to find any sites around the specified location.

The Forecast map is another interactive map built with ArcGIS. The map will display the air quality per hour to the users. There is a control scheme on the bottom left that includes a slider and a play, previous, and next button. Moving the slider will show the air quality at that hour. Clicking on play will show the change in air quality for each hour automatically. Clicking on next will go forward 1 hour while clicking on previous will go back 1 hour.

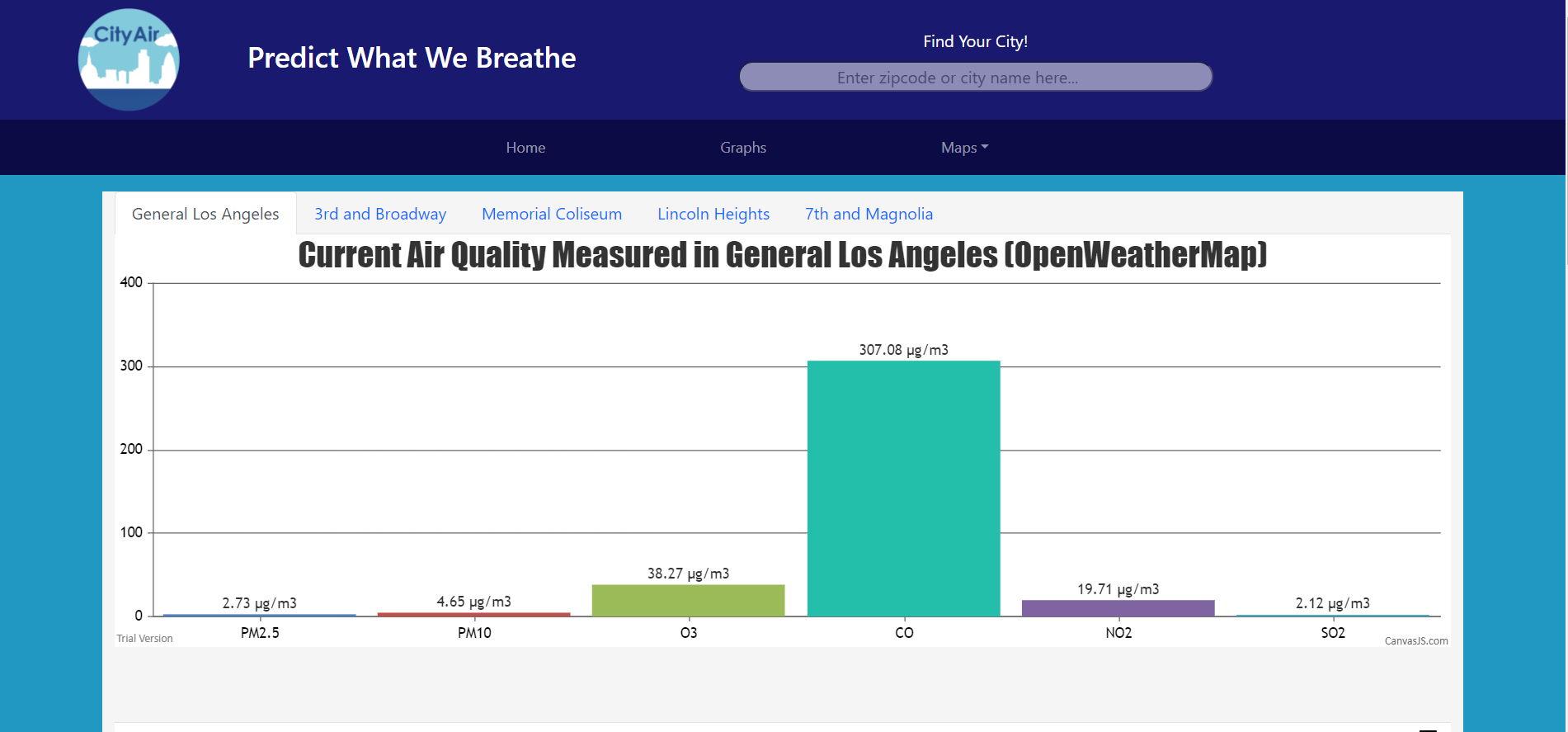
The forecast video is a video showing the change in air quality per hour for 24 hours around Los Angeles on October 14, 2021.

**9.2 Screen Frameworks or Images**

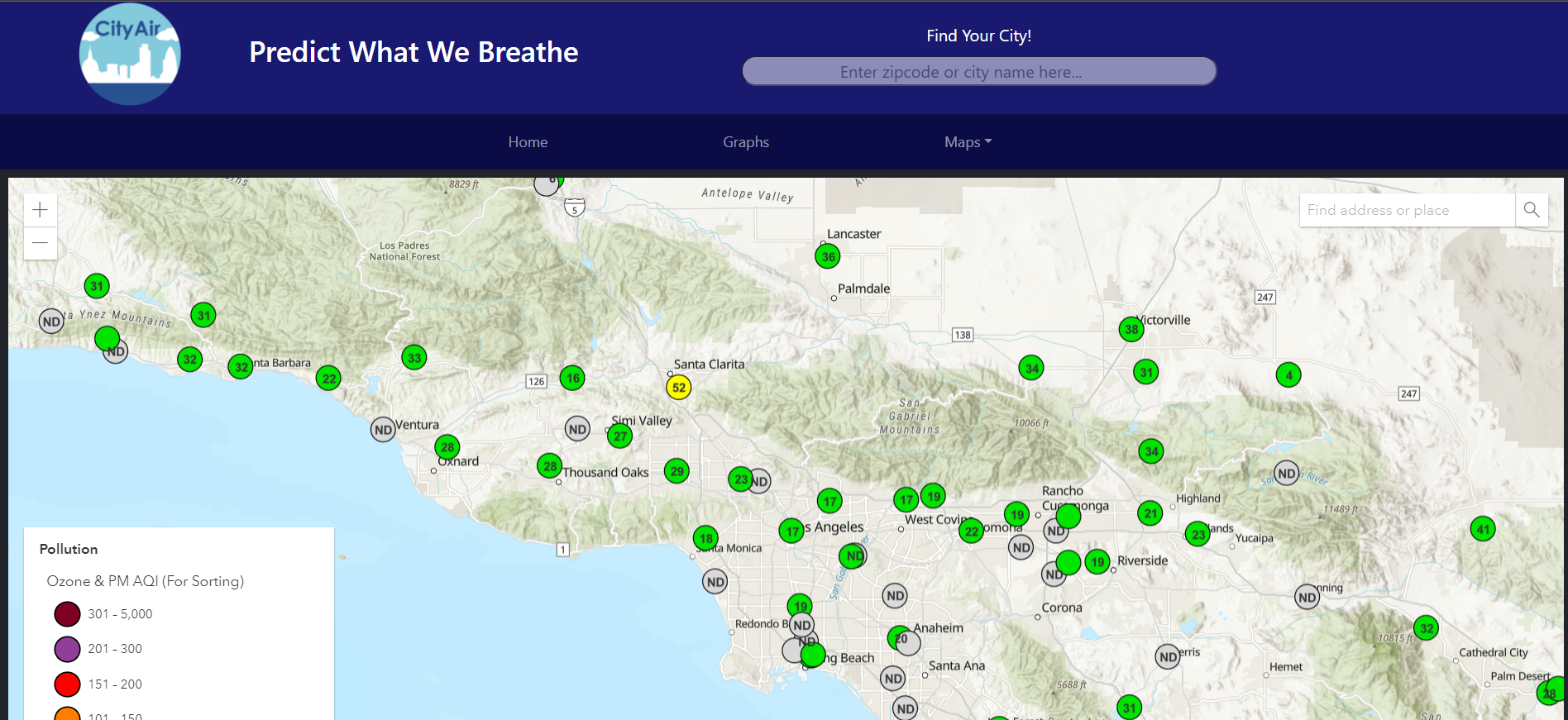
**Home Page:**



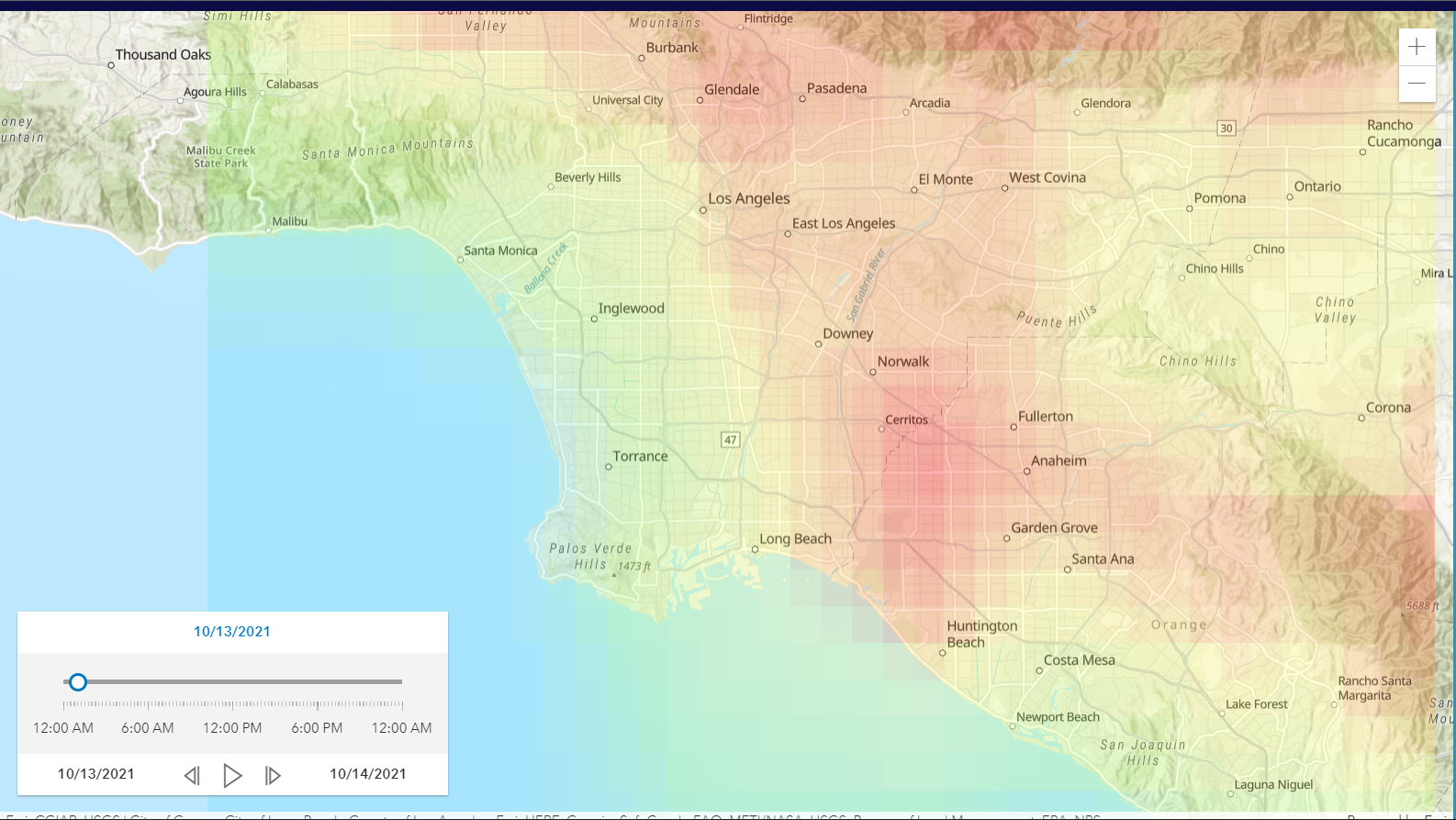
**Graphs Page:**



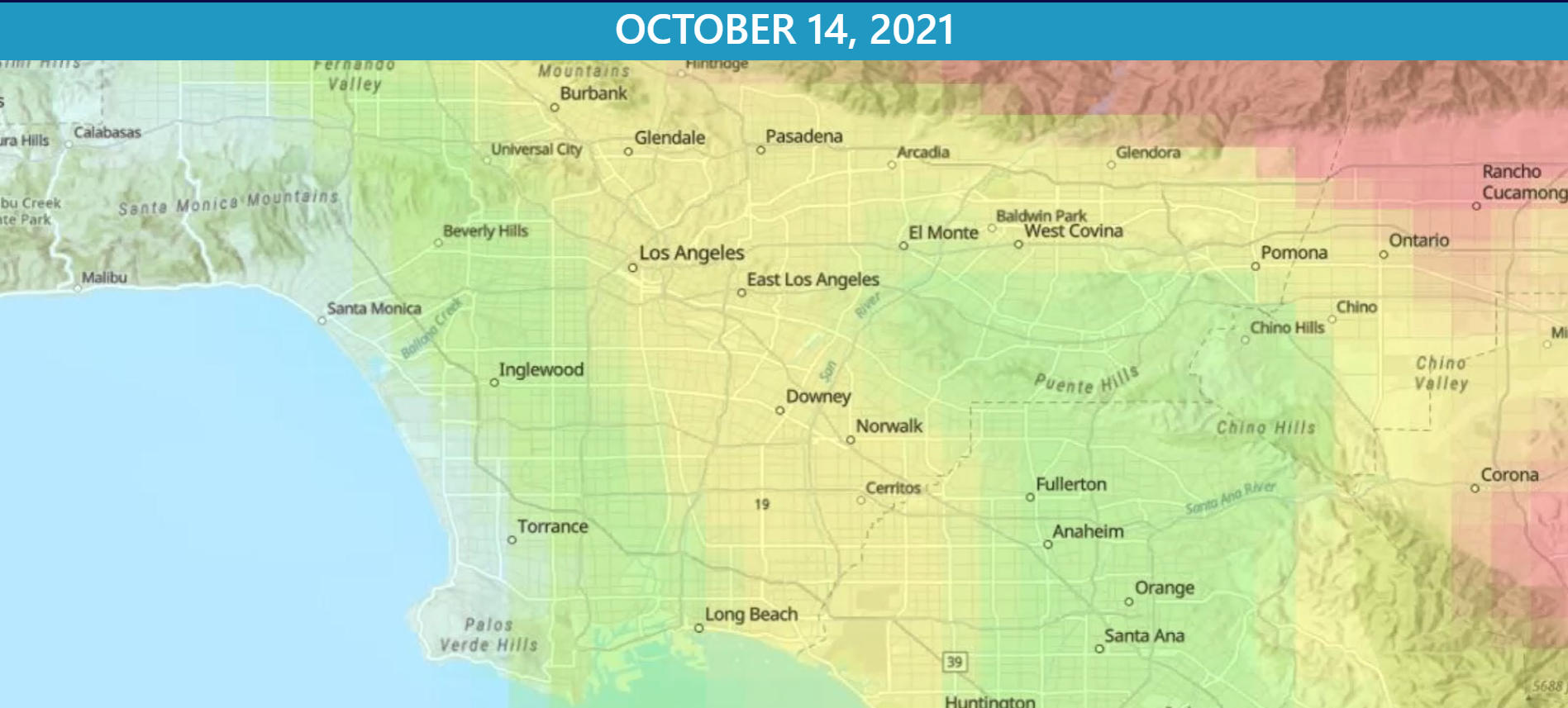
**Air Quality Map:**



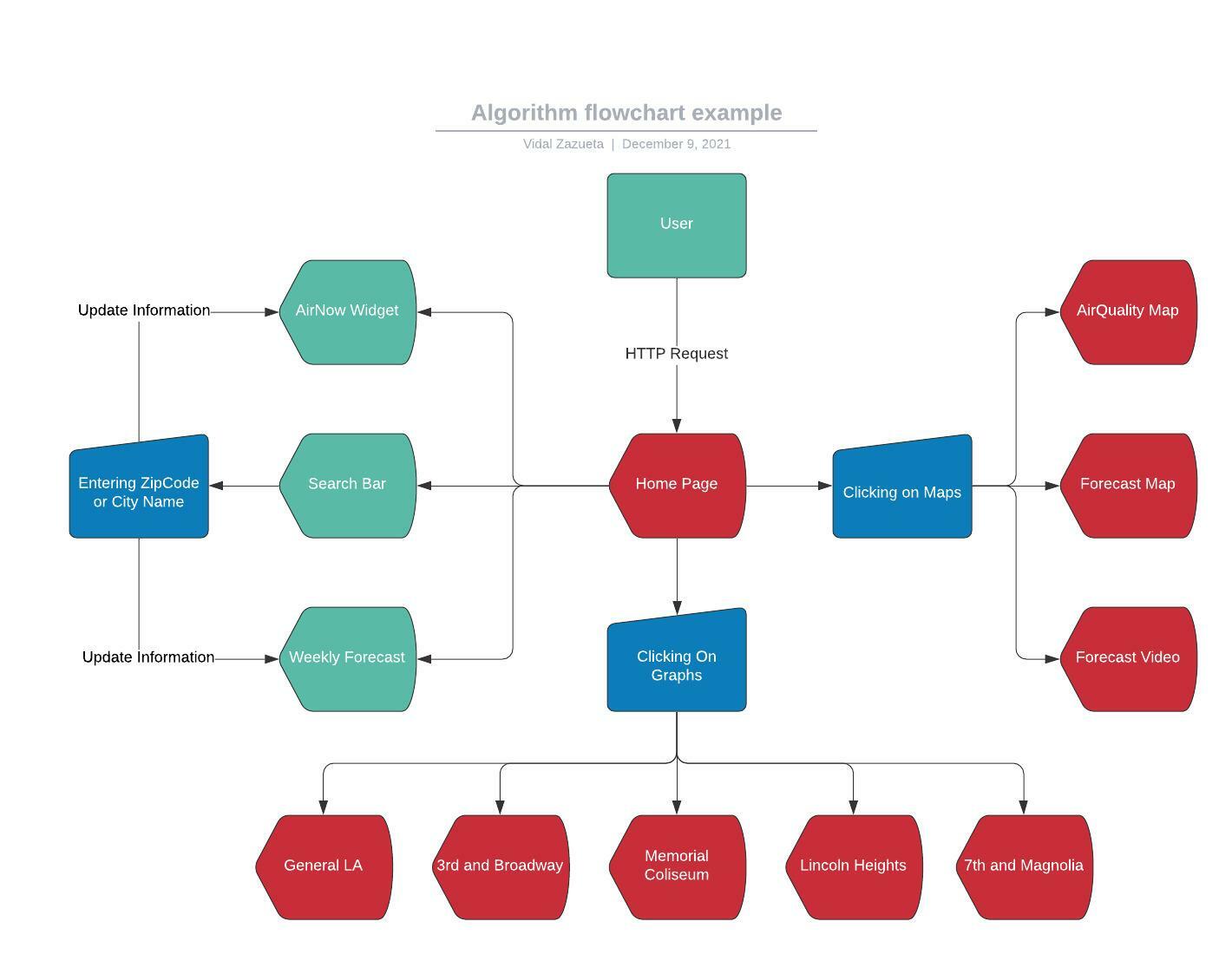
**Forecast Map:**



**Forecast Video:**

****

**9.3 User Interface Flow Model**



**10. Requirements Validation and Verification:**

**I. Air Pollution in Los Angeles County Data Visualization Module**

| 4.1.1 | The system shall retrieve data Layers from ArcGIS’s Live Atlas | Air Quality Map |
| --- | --- | --- |
| 4.1.2 | The system should retrieve data for custom Feature Layers from air quality sensors and fill in data for census tracts | Air Quality Map |
| 4.1.3 | The system shall use the retrieved data to display symbols on the map | Air Quality Map |
| 4.1.4 | The system shall use the retrieved data to display polygons on the map | Air Quality Map |
| 4.1.5 | The system shall visualize measured levels of air pollution using colors based on air quality severity harm | Air Quality Map |
| 4.1.6 | The system shall visualize demographic statistics using colors based on air quality | Air Quality Map |
| 4.1.7 | The system shall display a popup window upon clicking on a symbol on the map | Air Quality Map |
| 4.1.8 | The system shall display a popup window upon clicking on a highlighted polygon on the map | Air Quality Map |
| 4.1.9 | The system shall remove the popup window upon clicking away from the info window | Air Quality Map |
| 4.1.10 | The system shall add Layers to the map, Forecast Map prediction on air quality | Air Quality Forecast Map |
| 4.1.11 | Forecast Map has slider to show predicting machine learning model layers that predict each hour (still in process) | Air Quality Forecast Map |
| 4.1.12 | Forecast Map Video shall be easier to see air quality prediction data on a certain hour | Air Quality Forecast Map Video |

**11. Glossary**

**Appendix A: Glossary**

| **AIDSAPPV** | Artificial Intelligence and Data Science for Air Pollution Prediction and Visualization |
| --- | --- |
| **Air Pollution** | the presence of substances in the atmosphere that are harmful to the health of humans and other living beings or cause damage to the climate |
| **APLAC** | Air Pollution in Los Angeles County Data Visualization |
| **API** | a computer interface that defines interactions between software immediaries |
| **ArcGIS** | Esri’s all-in-one solution to work with geographic information |
| **Artificial Intelligence** | intelligence demonstrated by machines |
| **Data Science** | an interdisciplinary field that uses scientific methods, processes, algorithms, and systems to extract knowledge from data |
| **Feature Layer** | a grouping of similar geographic features that are used for visualizing data |
| **HTTPS** | application layer protocol that is used for secure communication over a computer network |
| **Machine Learning** | an application of artificial intelligence that provides systems the ability to automatically learn and improve |

**Appendix B: To Be Determined List**

The following requirements are conditions are to be updated:

● Add more Air Quality gauges of different Air Quality Pollutants

* Implement Mobile Application Using ReactNative for Air Quality Information
* Forecast tiff layers to be updated every hour or so

**12. References**

SRS document:

https://docs.google.com/document/d/1Nz4\_kllBgWuZvKEjCWoi8G7RZJNRBggr5WM29zdvvAE/edit?usp=sharing