**Senior Design Final Report**

Artificial Intelligence and Data Science for Air Pollution Prediction and Visualization

Version 1.0 - 05/014/2021

Team Members:

Eric Chan

Micky Chan

Peter Gatsby

Larry Guiterrez

Jose Landa

Faculty Advisor:

Dr. Mohammad Pourhomayoun

Liaisons:

NASA

LA City

**Table of Contents**

 1. Introduction

 1.1. Background

 1.2. Design Principles

 1.3. Design Benefits

 1.4. Achievements

 2. Related Technologies

 2.1. Existing Solutions

 2.2. Reused Products

 3. System Architecture

 3.1. Overview

 3.2. Data Flow

 3.3. Implementation

 4. Conclusions

 4.1. Results

 4.2. Future

 5. References

**1. Introduction**

**1.1. Background**

Air Pollution in Los Angeles continues to increase every year and it puts those with respiratory illnesses at risk. Our senior design project attempts to address this problem by visualizing air pollution in Los Angeles County with a web and mobile application. Both of these applications serve a different purpose. The web application is used to provide more general information of air pollution in Los Angeles while the mobile application is more personalized towards the users needs.

The web and mobile application utilize a map component that will reflect the current state of air pollution in different parts of Los Angeles county. We use this approach because visualizing the data is a more meaningful method of relaying information to the user. The Air Pollution Prediction and Visualization App will be developed for web and Android devices, with the plan to expand and make available to other devices.

**1.2. Design Principles**

There are two main deliverables for this project: a web application and a mobile application. The goal for the web application is to be an all-in-one dashboard that provides useful information pertaining to air quality and air pollution. The goal for the mobile application is to be a personalized app that offers more flexibility and allows for user customization in terms of the content they want to view. Both of these applications need to operate efficiently and display information in an effective way while being intuitive and user-friendly. Both of these applications are still in their development phase so their design needs to be simple for future maintenance and expansions.

**1.3. Design Benefits**

By having a system architecture that allows for future features and components to be added with ease in both the web and mobile application, it allows for demands of both the client and the developer to be met in an efficient manner. By splitting each module of our applications into components, it allows for developers of all levels of expertise to be able to understand and work on these applications and also allows for these applications to be easily fixed and managed. These applications were designed with simplicity in mind and provides a nice user interface that is easy to understand and easy to use.

**1.4. Achievements**

Over the course of the academic year, our team was able to develop preliminary applications for both web and mobile that provide useful air quality and air pollution information. The web application is currently built using the ReactJS framework and is being hosted on an AWS Amplify server. Currently, the web application is capable of gathering data pertaining to air quality or air pollution and display them via maps, graphs, and articles. The mobile application is currently built using the Java language and the Android SDK. Currently, the mobile application is capable of maintaining a list of existing users or adding new users to a Firebase database we are using. The mobile application is also capable of allowing users to customize their location and providing useful air quality and air pollution information in that location. Although these two applications are very simple, the frameworks we’ve chosen to work with allow for features to be added with ease and are capable of transforming these two applications into more complex applications.

**2. Related Technologies**

**2.1. Existing Solutions**

Some existing solutions that we took note of were websites like airnow.gov, iqair.com, epa.gov, and who.int.

The IQAir and AirNow website both have the mapping, search, and air quality index features that are common amongst most of the prevalent existing solutions that we’ve discovered. On the flip side, websites like EPA and WHO have information that are shown in either the form of a graph or graphics.

While websites like these have excellent information, they are often hard to navigate and these features are generally separate in that websites have the maps but no graphs or vice versa. Our goal for both the web and mobile application is to combine all these features in a way that is efficient and user friendly.

**2.2. Reused Products:**

The web application used ReactJS in order to build the framework of the website. ArcGIS was used in order to build the maps that are used to show air quality index, causes of air pollution, and individuals who have asthma in the United States. NewsAPI was used in order to obtain news articles in our web application. As well as two weather API’s called WeatherBit API and the OpenWeatherMap API; which were used in order to obtain the current forecast and the current air quality index of certain locations.

**3. System architecture**

The architecture for both the web application and the mobile application differ from each other and thus will be explained separately.

**3.1.1. Web Application Overview**

This is a diagram that displays how the architecture of our web application works at a high level. It is also our DFD Level-0.



* **The User:** This is an important factor in the architecture because considerations need to be made on how the user will interact with the application and what components may seem out of place or unnecessary. To that end, one of our team members, Jose Landa, has acted as the user for this application and has worked on fixing components that may not seem intuitive to other users.

**3.1.2 Mobile Application Overview**

This is a diagram that displays how the architecture of our mobile application works at a high level. It is also our DFD Level-0.

****

**3.2.1 Web Application Data Flow**

Here is a more indepth look into our web application as a system. This diagram is our DFD Level-1.



Here is a brief overview of the five major modules in this system

**3.2.1.1 Main Module:** This module is responsible for taking in user input and passing it to the other modules if necessary. This module is also responsible for interacting with the other modules in this system in order to gather information to display to the user. The user input that this module takes in may affect the information that is being displayed.

**3.2.1.2 Map Module:** This module is responsible for grabbing data from the ArcGIS Living Atlas and transforming the data into data points on a map which will then be passed to the Main Module to be displayed. This module is also responsible for taking the user input that the Main Module has collected and changing certain features of the map based on what the user input is. This means that certain datasets may be omitted or specific information for a datapoint can be inquired for.

**3.2.1.3 Conditions Module:** This module is responsible for grabbing air quality and weather data from various APIs and transforming that data into usable bits for the Main Module to display. This module is also responsible for taking the user input that the Main Module has collected and changing the air quality and weather data. This means that different datasets will be gathered based on the user input.

**3.2.1.4 Graphs Module:** This module is responsible for grabbing air quality and air pollution data from various APIs and transforming the data into data points for a graph that will be used by the Main Module to display to the user.

**3.2.1.5 Articles Module:** This module is responsible for grabbing article data pertaining to air quality and air pollution from various APIs which will then be used to construct article cards that will be used by the Main Module to display to the user.

**3.2.2 Mobile Application Data Flow**

Here is a more indepth look into our web application as a system. This diagram is our DFD Level-1.



Here is a brief overview of the five major modules in this system

**3.2.2.1 Login Module:** This module is responsible for the login or the registration of the user. When registered the user’s credentials are stored within a database so that the user can be authenticated when logging in

**3.2.2.2 Map Module:** This module is responsible for grabbing data from the ArcGIS database and transforming the data into data points on a map which will then be displayed on a map. This module is also responsible for taking user input and directing the user to the specified address of the user’s choosing.

**3.2.2.3 Dashboard Module:** This module is responsible for displaying cards of specified cities in a lift for the user. Each card will display the specified city and its air quality information along with certain weather data for that city. The module is also responsible for keeping track of which cities are in the dashboard for each user

**3.2.2.4 News Module:** This module is responsible for grabbing article data pertaining to air quality and air pollution from the WebSearch API which will then be displayed in a list to the user

**3.2.2.5 Profile Module:** This module is responsible for getting the permission to use the user’s location. If the user grants permission for their location to be used, then this module will display a card and minimap with information based on the user’s current location

**3.3. Implementation**

As this project is split into two applications, we will be discussing how each application was implemented separately.

**Web Application**

The web application was implemented by splitting the progress into two sections: air quality/pollution data and user interface.

**3.3.1. Air Quality and Air Pollution Data**

Air quality and air pollution data was gathered using JavaScript fetch requests. The data is gathered from various APIs that we have found and is transformed into usable bits of data that we can use to indicate certain conditions pertaining to air quality and air pollution.

**3.3.2. User Interface**

A user interface was designed for this application to visualize the data that we have gathered in **section 3.3.1**. The user interface allows for some interaction that may alter the data that has been gathered either through filtering or requesting a new dataset altogether.

**Mobile Application**

The mobile application was implemented by splitting the progress into two sections: air quality/pollution data and user interface.

**3.3.3. Mobile Air Quality and Air Pollution Data**

Air quality and air pollution data was gathered using methods from the ArcGIS runtime SDK. The data is gathered from a feature layer that we have displayed on the map in our map tab.

**3.3.4. Mobile app User Interface**

A user interface was designed for this application to visualize the data that we have gathered in **section 3.3.3**. The user interface allows the switching of several tabs to get different sets of data whether it be general data from a specific city or an overview on the map

**4. Conclusions**

**4.1. Results**

We were able to create two amazing applications that are capable of showing information on air quality and air pollution in various locations. The web application is capable of showing air quality index of multiple locations, weather conditions, as well as graphs and articles pertaining to air quality. The mobile application is capable of personalizing a profile that is tailored to the user’s location and showing relevant air quality and air pollution information pertaining to that area. Both of these applications are easy to navigate through and are extremely user friendly.

**4.2. Future**

Although there is already a lot of work done for these applications, there are still some features or expansions that are desirable and may be implemented in the future. For the web application and mobile application we plan on using machine learning models and algorithms to be able to generate a more accurate and live reading of how good or poor the air quality is in certain locations. The mobile application is also limited to Android users so an extension to the iOS system is heavily desired.

**5. References**

React Esri Documentation: <https://github.com/Esri/jsapi-resources>

ArcGIS JavaScript Documentation: <https://developers.arcgis.com/javascript/latest/>

CanvasJS Graphs Documentation: <https://canvasjs.com/>

ReactJS Documentation: <https://reactjs.org/docs/getting-started.html>

ArcGIS SDK Reference: <https://developers.arcgis.com/android/api-reference/reference/>