

# Team Climate Change



## Requirement Specification Document

December 12, 2019

Version 2.0

### Project Sponsors

Doctor Geoffrey Roest

Professor Kevin Gurney

### Team Faculty Mentor

Scooter Nowak

### Team Members

Kiley Jacobs (Team Leader)

Tung Nguyen

Zihang Shen

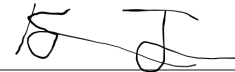
Yisheng Wang

Kiley Jacobs

Team leader

12-12-19

Date



Signature

Geoffrey Roest

Client

Dec 12 2019

Date



Signature

# Table of Content

<b>Introduction</b>	<b>3</b>
<b>Problem statement</b>	<b>4</b>
<b>Solution Vision</b>	<b>5</b>
<b>Project Requirements</b>	<b>6</b>
Functional Requirements	<b>6</b>
Performance requirements	<b>11</b>
Environmental Requirements	<b>13</b>
<b>Potential risks</b>	<b>14</b>
<b>Project Plan</b>	<b>15</b>
<b>Conclusion</b>	<b>17</b>

## Introduction

The Earth is warming at an alarming rate, and the recent growth in CO2 emissions is a major reason for this change. Many countries have signed agreements to reduce CO2 emissions, and many cities in the United States like Los Angeles, are taking the lead in reducing CO2 emissions. However, actions like diplomatic agreements are not enough; Even though many local governments are trying to reduce CO2 emissions, the data they have to work with are often too general to provide a useful solution or effective policies. Also, people don't know whether their actions really mitigate global climate change or not. So, our project is to create a Web Application that can give people a very clear visual map that shows how much CO2 emissions are increasing, as well as being able to see the CO2 emissions level for their own hometown.

The project sponsors are Geoffrey Roest, Postdoctoral researcher in SICCS (School of Informatics, Computing, and Cyber Systems) at NAU (Northern Arizona University), and Kevin Gurney, Professor in SICCS at NAU. Professor Kevin Gurney has recent research on topics in carbon cycle science, climate science, and climate science policy. Also, he is working on the project involving simulation and quantification of US CO2 emissions, the linkages between terrestrial carbon exchange and climate variability, and the impacts of deforestation on climate. He also has worked extensively on climate policy and has been involved for over 25 years with the United Nations Climate Change Framework Convention and the Kyoto Protocol.

For this project, they have collected large amounts of data about CO2 emissions in the United States, and they have been able to create a map over the country that displays this data. The problems that they are facing right now are dissemination and accessibility. Since the data is only available in technical formats, you need specific software to open and analyze them; the general public doesn't have sufficient knowledge or skills to use most of this software. So, our primary goal is to build an

easily accessible interface that allows everyday people to understand and explore CO2 emission findings.

Our solution will be a Web application that can show the CO2 emissions in an interactive graphical interface. Users can access our website and see a map of America displaying relative CO2 emissions. The purpose of this document is to describe the project's requirements, both functional and non-functional, to form the basis of expectations that our team needs to fulfill. Simply put, this document will show the specified features and performances of our final project that our clients want.

## **Problem Statement**

Currently, Professor Gurney and his team has spent several years developing a standardized system as part of the Vulcan Project that quantifies and visualizes greenhouse gases emitted in urban regions, identifying problem areas, and showing whether or not public policy is making a difference. The method that they are using to measure emissions is by tracking activities that generate these emissions.

Our clients want more than just pictures and videos of the map; they want interactive graphics that help people visualize their city and their personal emissions in order to illustrate how their actions are affecting the environment in which they live. Professor Gurney hopes that this will inspire people to become more engaged in policy development.

The three biggest issues with the workflow of our sponsor's goal are as follows:

- Data is only available in technical formats: Our clients would like the general public to access and read these data. However, their data is only available in

technical formats, which require specific software to open and analyze them. Most people don't have technical background to use these softwares.

- Data is hard to interpret: Without knowledge and skills, it's difficult for people to interpret this data, to explore what our client's data is revealing about CO<sub>2</sub> emissions, and to understand the issues more clearly.
- No user interaction: Our sponsors currently only have pictures and videos of CO<sub>2</sub> emission maps. These maps cannot display a specific area or a neighborhood that users request, and are purely static in appearance. So, it limits the information available to the user. This makes it difficult to know how much personal emissions they are contributing to global warming and engage in policy development.

## Solution Vision

Our solution will be make a map to show the CO<sub>2</sub> emission in America. In the map, the different level of CO<sub>2</sub> emission will be represented by different colors, and we will also provide a fair amount of user interaction to help the user get the information they want more easily.

- The data is only available in technical formats, and it is not easy for non technical user to understand. Our solution is using this data to build a visual map so the user can easily understand this data.

- The data will be hard to interpret, so we need a map framework to help us understand and use this data. Mapbox is a very useful framework, so it can help us to create the map by uploading our data and using it.
- In our solution, we need to provide a lot of user interaction to help people be more clear on what the map is exactly showing. We will provide the search location function to allow users to search a specific location and zoom in to see more detail on that area. We will provide a time slider function to let people compare the CO2 emissions over different times. We also will allow the user to show and hide layers of different CO2 emission source types and only look at what they want to see. There are several more UI functions we will implement that are addressed later in this paper.

Figure 1 below is our solution to the picture, these are the features of our website, it includes the time slider, display layers, search location, show information under pointer, and adjust the color of the map.

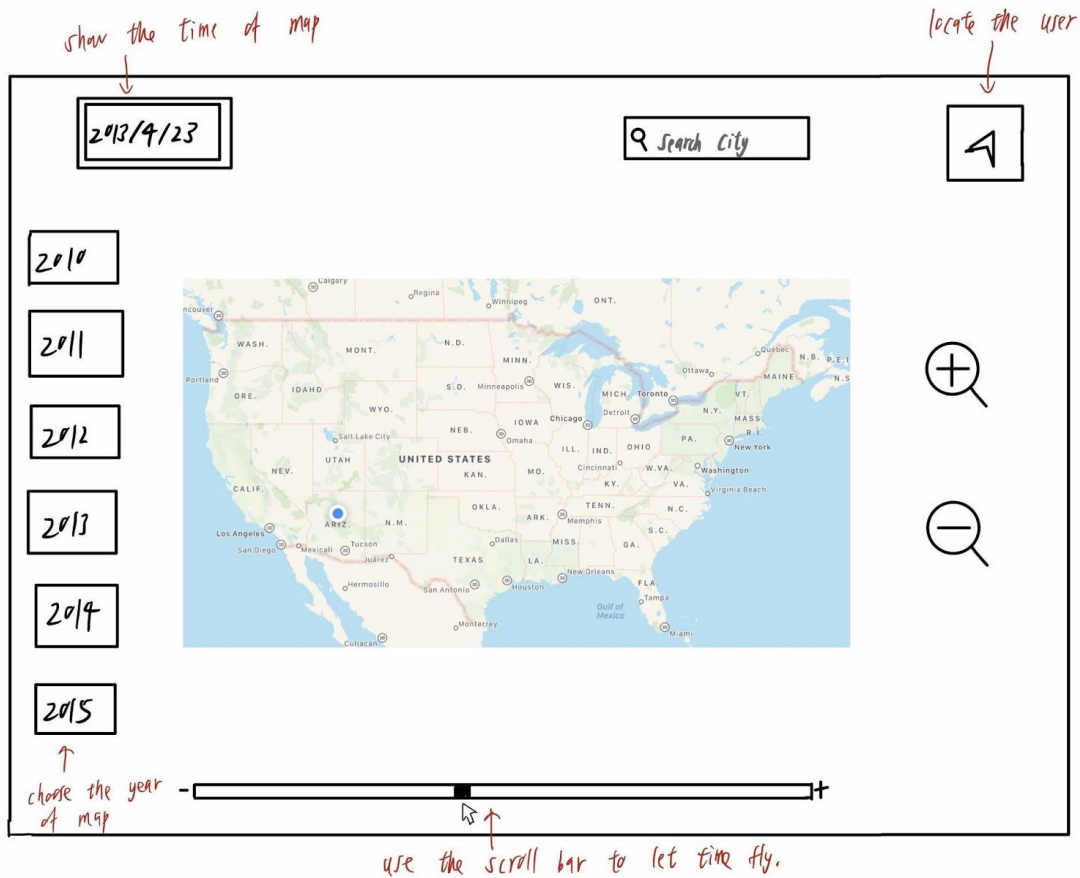


Figure 1

Our system uses raster data, and some other types of data, that are collected from our client. Our main computational processes are data transformation; we upload our client's data into Mapbox's Tilesets and use Tilesets to generate Mapbox's web services API, then this information will be shown on our website.

## Project Requirements

For an interactive map-based website, users' experience are the most important aspect. Our website must be a highly functional website that is easy to use with no frustration

on the user's end. This will allow us to create several different functions to allow the user to interact and customize the website's map in different ways.

## 1. Functional Requirements

Functional Requirements needs to layout all of the functions that the application will have, as well as how various users can use our application.

### 1.1. Our Product will Upload Various Data Formats

Our clients already possesses, and will provide us with, sufficient CO2 emissions data for America. Accessing this data using our maps will be one of the main requirements. Our clients have been helped previously by google programmers to create an interactive map using GIS, but the lack of maintenance has caused the collapse of that interactive map. Most of the data the clients currently has is GIS raster data, but also has some vector data and Hestia data. We need to upload these data into the Mapbox, which is the map API we have chosen to utilize for our project.

#### 1.1.1. Our Product will upload Raster Data Formats

Raster tilesets are used in Mapbox Tools to display map data as a grid of images that can be loaded onto the map. We can upload this raster data to the Mapbox account in GeoTIFF format, which is the specific data format the client currently has right now.

##### 1.1.1.1. Our Product will upload GeoTIFF Format as Raster Data Formats

GeoTIFF is a standard. It is based on the TIFF format and is used as an interchange format for georeferenced raster imagery. tif or image file format that includes additional spatial (georeferencing) information embedded in the tif file as tags. This data is in the form of an image with certain information attached to it, rather than a raw table of data. The



Mapbox Raster Tiles API also allows us to request tiles from a Mapbox-hosted raster tileset if need be.

#### 1.1.2. Our Product will upload Vector Data Formats

The Mapbox Vector Tiles API serves vector tiles. Each API request results in a single tile from a vector tileset. Vector data differs from Raster data as it can come in many different polygon shapes. We can upload vector data to Tilesets by Shapefile, but it must be uploaded as a compressed file.

Vector tiles can be used as sources for styles, or queried directly to create interactive experiences with the Mapbox Mobile SDKs and Mapbox GL JS.

##### 1.1.2.1. Our Product will upload Shapefile as Vector Data Formats

The shapefile is a simple, nontopological format for storing the geometric location and attribute information of geographic features. Geographic features in a shapefile will be represented by three ways: points, lines, or polygons.

#### 1.1.3. Our Product will upload Other Data Types

Mapbox also accepts other file types. CSV or GeoJSON files can be uploaded up to 5 Mb in datasets, or 1 GB in tilesets, MBTiles up to 25 GB, KML up to 260 Mb with 15 layers, or GPX up to 260 Mb into tilesets.

These are the other file formats we may possibly utilize if our current data type does not work or is not performing at the level at which we want it to.

##### 1.1.3.1. Our product will upload CSV files

A CSV is a comma-separated values file, which allows data to be saved in a tabular format.

##### 1.1.3.2. Our product may will upload GeoJSON files

GeoJSON is an open standard geospatial data interchange format that represents simple geographical elements and their non-spatial attributes.

#### 1.1.3.3. Our product may will Upload MBTiles files

MBTiles is a file format for storing map tiles in a single file. It is, technically, a SQLite database.

#### 1.1.3.4. Our product will Upload KML files

KML file extension is a Keyhole Markup Language file. KML files use XML to express geographic annotation and visualization by storing locations, image overlays, video links and modeling information like lines, shapes, 3D images and points.

#### 1.1.3.5. Our product will Upload GPX files

GPX (the GPS eXchange Format) is a data format for exchanging GPS data between programs, and for sharing GPS data with other users.

### 1.2. Show Data on Map

After we upload the data into datasets and/or tilesets, the next step is to use this data and show it on our map. We will need a Mapbox access token to use any of Mapbox's tools, APIs, or SDKs, as Mapbox uses access tokens to associate the account with the requests to Mapbox API resources. We can find all the access tokens, create new ones, or delete existing ones on the Access tokens page. We can also use the Mapbox Tokens API to programmatically create, update, and delete access tokens.

### 1.3. Our Product will Provide User interaction in map

We are using maps to display the United States' CO2 emissions, making it easier for ordinary people to understand the changes in CO2 emissions

throughout different regions at varying times. Through various user interactions, users can easily gather the information they want about the current state of CO2 emissions in the country.

#### 1.3.1. Our Product will Provide a Time Slider

The time slider will allow the user to drag the slider to view CO2 emissions from different times spanning across several years.. You can also press play on the time slider to make the map automatically play. Our first goal is to create a time slider in year or month increments. A later iteration would have this interval be brought down to hour by hour. This feature makes it more intuitive for users to truly see how CO2 emissions in American are rising or falling over time.

#### 1.3.2. Our Product will Provide Hide or show layer

Since on our map we expect to have several different CO2 emission source layers such as vehicle exhaust emissions, emissions from factory production, etc., our map will allow users change how many layers of the map they are looking at at one time.. Users can use this function to hide sources of CO2 emissions that they are not interested in, and only observe the changes of the CO2 emission sources that they are interested in so they can have a better understanding of that particular set of data rather than being overwhelmed by all of them.

#### 1.3.3. Our Product will Provide the Search Location Function

Allow users to search for a location, or directly locate the location of the user's IP, making it easier for users to find the specific place they want to see on the map. Searching the location can be searched by address or keyword, and Mapbox Studio comes with such a search function already implemented, which entails using Geolocate to locate the user and then

track their current location on the map using the GeolocateControl.

#### 1.3.4. Our Product will Provide Function to Show Information Under Pointer

When the user clicks somewhere on the map, it will display the information of the area under the mouse pointer. Such as the total CO2 emissions in this area, different CO2 emissions from different sources, or the latitude and longitude coordinates. It will give the user a summary of the local basic information to allow for more context.

#### 1.3.5. Our Product will Provide Function to Adjust Color or Transparency

Users can adjust the layer display color and/or transparency to their liking. When the transparency is reduced, the user can more easily view multiple types of CO2 sources at a time.

### 1.4. Our Product will Provide Ranking

#### 1.4.1. Our Product will Provide the Emission Ranking

Show the rank of the total CO2 emissions and other different sources of CO2 emissions each year or each month, to see which states or cities has high CO2 emissions rates.

#### 1.4.2. Our Product will Provide the Per Capita Averages Ranking

We can show the rank of CO2 emissions by emissions per person averages in different states or cities.

### 1.5. Our Product will allow user Downloading Data

Place a link under the map or somewhere within the website to allow users to download all of our CO2 emission data.

### 1.6. Our Product will allow Different Site Access

Users should be able to have a good interaction with the website on a variety of different devices.

#### 1.6.1. Our Product will allow Desktop Website to Access

Mapbox provides a few tools to build maps into the website or web-based application. Mapbox GL JS and Mapbox.js are both open source JavaScript libraries that we can use to display the Mapbox maps, add interactivity, and customize the map experience in the application. Also Mapbox has provided many plugins for extending the web map's functionality, with drawing tools and interfaces to Mapbox web services APIs, like the Mapbox Geocoding API or Mapbox Directions API.

#### 1.6.2. Our Product will allow user to Access from IOS system

The Mapbox Maps SDK for iOS is an open-source library for embedding highly customized maps within iOS applications. It has custom styling, offline maps, data visualizations, and is open source. It compatible with applications built in Xcode 8 or above and deployed on iOS 9 and above. This is available to us just in case the client wants us transplant this map into some IOS application.

#### 1.6.3. Our Product will allow user to Access from Android system

The Mapbox Maps SDK for Android is an open source toolset for displaying maps inside of Android applications. For the version v8.5.0, it has these features: Map styles, Run-time styling, Data clusters, Camera manipulation, Querying the map, Gestures, and Map images. This is available to us just in case the client wants us transplant this map into some Android application.

Use Case	Functional Requirements Met
----------	-----------------------------

Client wants us to change data to CSV	1.1.3.1
User wants to see emissions over entire year	1.3.1
User wants to see only road emissions	1.3.2, 1.3.5
User wants to search own address	1.3.3
User wants to see emissions per capita in their zip code	1.3.4, 1.4.2
User has a tough time telling apart colors on map and wants to change them to another	1.3.5
User wants to see the city that produces the most emissions	1.4.1, 1.3.4
User wants to look at the raw data themselves	1.5
User wants to view map on a larger screen	1.6.1
User wants to view map on their phone	1.6.2, 1.6.3

Table 1

In the table above we lay out several use cases showing practical needs for a large number of our functional requirements.

## 2. Performance Requirements

In this section we will be evaluating the performance requirements for our system. These non-functional requirements serve as benchmark goals for navigation times and data accuracy. The performance requirements for our system are as follows:

2.1 The map (1.2) shall be responsive and implement good design principles

2.1.1 The front end (1.3) shall respond within 10 seconds of the user input

2.1.2 Information should appear in under 5 seconds when user clicks on a label (1.3.4).

2.2 Usability:

2.2.1 The system must include a design that enables users to navigate from one page to the next in less than 10 seconds.

2.2.2 The map should take less than 10 seconds to display different layers of the map (1.3.2) when a user changes them.

2.2.3 The map should take 5 seconds to search or modify a location(1.3.3)

2.2.4 Users must have the ability to get expected results from each click. The results should return in less than 5 seconds.

2.3 Reliability

Given the amount of expected traffic and simultaneous use of the map application (1.6), it is extremely important that the system is always accessible to users at all times.

2.3.1 The map should take 10 seconds or less to load at all times.

2.4 The system shall verify data for correctness (1.1)

2.4.1 The system shall display accurate data 100% of the time.

2.4.2 The system shall explicitly validate the data after receiving it from the database before displaying it on the map.

2.5 The system is easily modified by non-members of our team

2.5.1 Code will include blocks of comments that are deciphered by non-technical users in less than 10 seconds and able to be modified by minimally experienced programmers

2.5.2 There be leway given in areas the client has expressed the intention to expand upon and comments indicating this.

### **3. Environmental Requirements**

Environmental Requirements are constraints put on us by either the tools we are using or restraints given to us directly by the clients.

#### **3.1. Using Client's Data**

For our project, we have to use the data the client has already produced.

##### **3.1.1. Data Size**

The client has a large amount of data as they have six years worth of emissions data for the entire United States. The client's data will be around 100 gb, so our solution needs to be able to store that amount of data.

##### **3.1.2. Data Type**

Our client's data is currently in GeoTIFF format. This gives us a limited range of manipulation with said data. However, our clients are capable of converting this data to other formats, such as Netcdf. Our solution must be able to use GeoTIFF data, or some derivative of it that the client can



produce for us.

### 3.2. Integrating Into Client's Website

Our client already has an existing website so we must be able to move whatever website we create over into theirs

#### 3.2.1. Limited Freedom

Due to us not creating a website entirely from the ground up, we will be limited with how we make our site look as it should ideally not clash with their current design. Our solution must fit the design already on the client's website.

#### 3.2.2. Intact Website

We could run the risk of causing our portion of their website to not work, or making their entire site not work once we integrate over. This will require us to check that we aren't going to cause issues such as this from happening. Our solution must be able to properly function when put into the client's website.

## Potential Risks

As with all software, there are many potential risks that are involved. During our requirements gathering and technological feasibility studies, we have determined a few key risks to this project's success and overall impact. These were documented as we went through both processes and conducted general research, including reading about software engineering (namely the risks and mitigation strategies). It is important to outline the risks and ways to overcome them, because they can ultimately lead to the project not functioning, or completely failing.

We have summarized our risks in the table below:

Risks	Probability	Severity	Description	Solution
Memory leak	60%	High	The application can crash anytime.	Remove unused layers
Meeting our client's vision	20%	Medium	We may miss some crucial requirements	Keep client updated with our progress
Slow loading time	80%	High	Internet connection speed influences greatly to the map performance	Use generalized version of layers at smaller scale

Table 2: Potential risks summary

In table 2, We have assigned a percentage of occurrence at the worst case (probability of happening without mitigation), and a general low, medium, or high impact rating. We will go into more details for each risk.

### **Risk 1: Memory Leak | 60 % probability | High**

Memory leak is when objects are stored in the memory and not being removed when they are no longer in use, overtime this builds up until the application crashes. To avoid memory leaks, memory allocated on heap should always be freed when no longer needed. What we can do is to develop a backend program to remove any unused layers and data frames that are unrelated to the current display.

### **Risk 2: Meeting Our Client's Vision | 20 % probability | Medium**

While we are working with our clients, there is always an amount of risk that we may have missed a crucial requirement for the system. As we develop the requirements, we will hopefully diminish this risk, but there is always a possibility that the client mentions

something that may have been a big requirement that we did not catch. To minimize this risk, we have to get in touch with client either face to face or through email frequently. Keep them updated about our plans, as well as what we have done, and what we plan to do in the next few weeks.

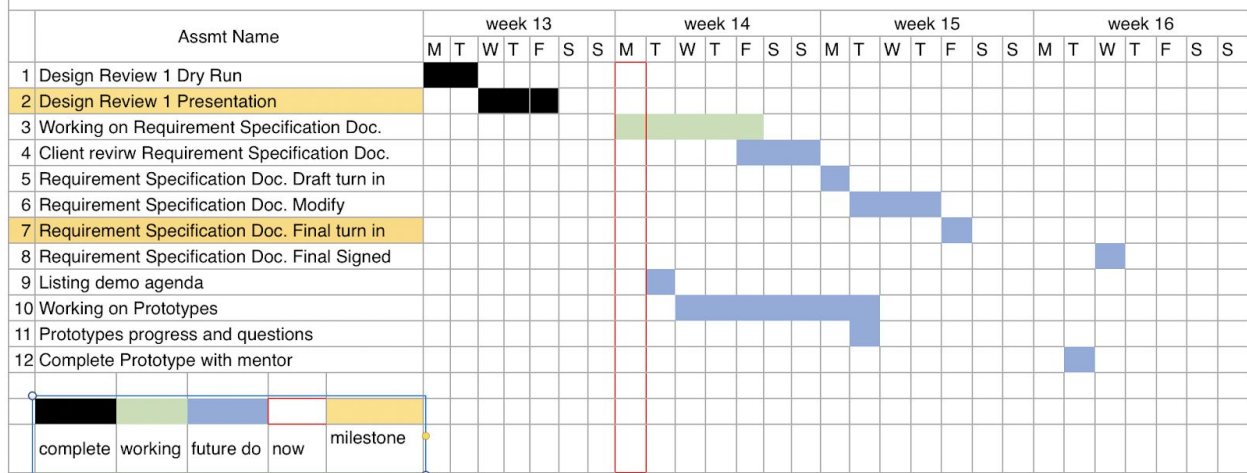
### **Risk 3: Slow Loading Time | 80 % probability | High**

This risk concerns about users who have slow internet speed which will make the map have longer loading time. Since our map will contain a large amount of information, it could take a while to get the map loaded, depending on user's internet connection speed and their device specifications. What we can do to minimize the risk is to use generalized versions of layers when displaying at smaller scales to have fewer vertices and line segments. This will help the map draw much faster with little difference in appearance at a small scale. In addition, simplified data can improve performance for querying and identify operations.

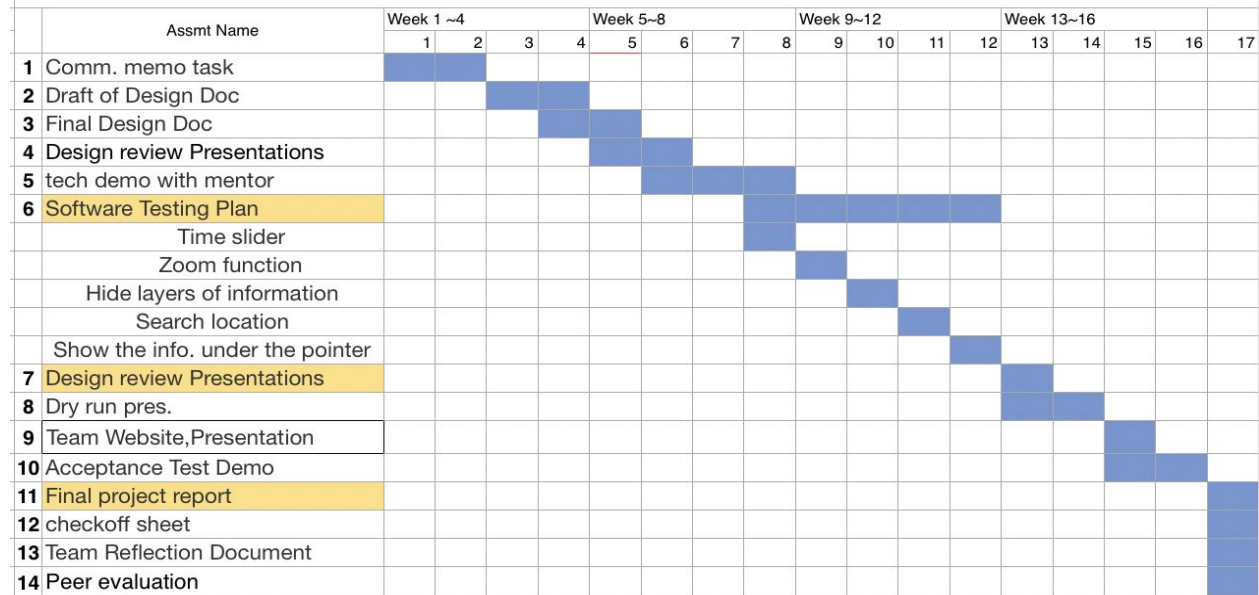
## **Project Plan**

Our project execution plan was developed with milestones in mind. We created the Gantt charts below to help illustrate our development schedule. We separated our development schedule into phases, each phase represents a milestone in our development.

## Fall Semester Schedule



## Spring Semester Schedule



The Gantt chart above serves as a high level overview illustrating our development schedule going into the spring semester. Our Prework phase consists of tasks we were carrying out this semester and will conclude with our technology demo showcasing each

top level requirement. The five milestones are marked by yellow in the graph, each one corresponds to our functional requirements. The first 2 milestones have been completed in the Fall semester.

In January, we will begin working on map features. To develop the interactive map, we plan to implement at least a core feature every two weeks for the map. This will allow the client ample time to review the map application and request any changes and report any bugs during development. In February, our milestone will be testing the features of our application. We divided the testing plan into 6 phases, each phase will have a week duration for testing.

When Spring break is over, we should have all major features of the website completed, and the product will enter into a production environment as soon as possible for the client to help determine the existing flaws with the product. Then, in March we will test our interactive map on client's website. Our final milestone will have the project completed and ready for our client's use. This Gantt chart is high level and is tentative; Certain phases may be reduced or extended, or even overlap with one another.

## **Conclusion**

The project given to us by our client is to help create a web application that people can use to see the CO2 emissions in the United States that doesn't require technical knowledge of the data. So, our solution to this is to interpret this data in a database that will allow the website visitor to view the data in a number of ways simply by clicking buttons and sliders, such as a time slider that allows users to view emissions overtime.

This requirements specification document has provided overall clarity of the project requirements and details the expectations that are to be met. It has projected future dated highlights that will be faced next semester as well as what must be done to materialize the vision. This document aims to clearly define our sponsor's problem

statement, our proposed vision for the solution, our Functional, Performance, and Environmental requirements, our project plans, and any potential risks associated with our project. We are confident with our planning and development schedule that we will build an interactive map that satisfies our sponsor's needs.

In conclusion, in order to let people know more about CO2 emissions in the USA, our project's goal is to develop a website showing CO2 emissions at each city in America, we hope our map can be highly interactive with our customers and can be easily read. Now our problem is the build of framework, the good news is that we already finish choosing the framework, Mapbox will be our solution to the framework, the picture below is the result of we input client's data into Mapbox, it's a great achievement in our project and we believe our team will develop a satisfying product in the final.

