



Testing Plan

April 3, 2020

Version 1.0

Project Sponsors

Doctor Geoffrey Roest

Doctor Kevin Gurney

Team Faculty Mentor

Scooter Nowak

Team Members

Kiley Jacobs (Team Leader)

Tung Nguyen

Zihang Shen

Yisheng Wang

Table of Contents

1. Introduction	3
2. Unit Testing	4
3. Integration Testing	8
4. Usability Testing	9
5. Conclusion	11

1. Introduction

The Earth is warming at an alarming rate, and the growth of 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. Without an easy to use way for everyday people to see environmental changes, people cannot observe and understand the changes of regional CO2 emissions. Also, people don't know whether their actions really mitigate global climate change or not.

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. The project aims to provide people with a free, interactive map of U.S. CO2 emissions to observe and compare U.S. CO2 emissions from various sources between 2010 and 2015. With the high-precision raster layer, users can more accurately observe the carbon dioxide emissions in their living areas and even their homes. So, it can make people realize that the growth of carbon dioxide emissions is closely related to their lives.

In order to ensure that the final submitted software does not have any problem, we need a good testing plan. The project mainly focuses on the CO2 emission interactive maps. The other two web pages, ranking page and download page are not difficult, just need to take a while to make it look good and the style fit our client's website. The whole test plan is divided into three parts; Unit Testing, Integration Testing, and Usability Testing. Unit Testing is the most basic test, which mainly tests whether each basic component works as well as required. In this part, we will test that all the Tilesets in the mapbox work well in Styles, and whether all the layers are numbered and named as required. The API for each Styles are in public states and available for use. Each interactive function needs to work as required, the data in the ranking page must be

accurate, and from the download page users can easily find the data they want to download and can download it. Integration Testing is to test if all the components can still work well after they are combined together, and there will be no mistakes under any circumstances. Lastly, the Usability Testing is about Testing the user's experience, convenience, and functionality. This way we can find some small shortcomings and improve the user experience.

2. Unit Testing

Unit testing is mainly used to test each single component and that all the code of functional modules can perfectly represent the functions in different situations that meet customer requirements. This is to ensure that when all the components are combined, the system will not collapse due to a single component or a single functional module error.

Unit testing has four parts:

1. Mapbox component: The test for all the Tilesets in the mapbox work well in Styles, all the layers are numbered and named as required, and the API for each Styles are in public states and available for use.
2. Interactive function: Each interactive function can work as required.
3. Ranking page: The accuracy of the data in the ranking page.
4. Download page: Can easily find the data users want to download and be available to download.

2.1 Mapbox Component:

This is the basis of our interactive map and as also are our online database, and we have to manage all the layers and maps well to make sure that the API for the layers can be correctly used in our Style and the API for the map is used correctly in our code.

Unit Name: Tilesets

Description: all the layers are named and numbered in the right way. Because Mapbox Tilesets doesn't allow classification management, the layers must be uniformly named as source name plus the year.

Successful: All layers named as format: <source_name><year>.

Expected Outcome: Easy to manage, and if clients want to add more layers in feature, it will not mess up.

Unit Name: Styles.

Description: All the source layers should be put in the right place, because the basic map already has 45 layers, in all of the Styles should put the source layer in the same place. Otherwise, the different year's map will look different.

Successful: In each Styles, all the source layers are put in the same place with the same order in the basic layer.

Expected Outcome: All the different year's maps look the same.

Unit Name: API.

Description: Each Styles will have a unique API, since our client has no data confidentiality requirements, so for each Styles we can set states be public just to make sure everyone can use it.

Successful: All the Styles are usable.

Expected Outcome: All the Styles can be used in HTML code.

2.2 Interactive Function:

We need to test each interaction function independently first to make sure it works well in all cases without errors.

Unit Name: Search function.

Description: Allows the user to search for the exact address and locate it accurately. The secondary system can help the user type, enter a part of the address to give some associated address options that the customer might want to search for.

Successful: Accurate positioning, response time in one second.

Successful Expected Outcome: Accurate location and fast response speed plus auxiliary search can give users a good user experience.

Unsuccessful: Nonexistent address or garbled code.

Unsuccessful Expected Outcome: Auxiliary search box will show 'No results found' and doing nothing.

Unit Name: Show information function.

Description: Create a clickable point for each states, after click the point, display all kinds of information about the state, such as population, etc., to help users get a preliminary understanding of that state.

Successful: All the points can be clicked and respond quickly within a second, showing accurate information.

Expected Outcome: When users want to learn something about a state, they can click on the black point to get some preliminary information about that state.

Unit Name: Change year function.

Description: Users should be able to change the map by year. Click the button representing different years to change the map.

Successful: When a user clicks the year, the map can be changed in two to three seconds under normal network conditions. And the year button should work any time whatever each source user chooses and whatever the show information function is showing the information or not.

Expected Outcome: A set of year buttons. Simple, quick response and no error reporting under any circumstances.

Unit Name: Change source function.

Description: User should be able to change the layer to a different emission source.

Click the button representing different emission sources to change the layer. A total layer and ten other emission sources layer.

Successful: When a user clicks one of the emission source buttons, the layer will be changed in one second under normal network conditions. And the source button should work any time whatever year map users visit and whatever the show information function is showing the information or not. Also, no more than one layer is allowed to overlap.

Expected Outcome: A set of emission source buttons. Simple, quick response and no error reporting under any circumstances.

2.3 Ranking Page:

Unit Name: Ranking Page

Description: A webpage to let users look at the ranking of U.S. carbon dioxide emissions from different sources.

Successful: The data must be accurate and reliable, the presentation intuitive and clear, the page style similar to the client's website.

Expected Outcome: When the user clicks from the client's website to this page, there is no big visual difference. And the data or charts are clear and easy to understand.

2.4 Download Page:

Unit Name: Download Page

Description: The webpage to allow users to download our data.

Successful: The classification of the data is clear, with a sentence introduction. The file corresponding to each download link is accurate, and the page style similar to the client's website.

Expected Outcome: When the user clicks from the client's website to this page, there is no big visual difference. And can easily find links to download data of interest.

3. Integration Testing

Another important factor of assuring product quality and integrity is to conduct integration testing. This makes sure that all of the systems - the map framework and the datasets are all working properly.

Goal and Overall Approach:

Our goal for integration testing is to make sure that all the essential technologies and dependencies are holding together nicely in development. Our overall approach for this style of testing will include pushing all available features to our development server. This will allow us to have an idea of how the product will be served to the end user and provide us with valuable feedback.

Focus:

The main focus of the integration testing will be making sure our map framework and datasets are connected properly. Since these two components are the largest and most significant portions of our end product, there will be a big focus on how well they integrate with each other.

Features that will be tested:

To make sure the map page, ranking page and download page put in the client's website services will work fine without any error.

The function need to test after it put in client's website:

Information display:

Create a clickable point for each state. The information for each state will be displayed when the user hovers the mouse over point to help the user get an understanding of that state's specific emissions.

Search location:

Search function still working and can find the current location.

Switch layers:

Display the correct layer when the user chooses an option for that specified source. When the user chooses another source, the previous source layer will be removed.

Switch Map:

Display the correct map when the user chooses an option for that specified year.

Map color:

The color will represent the quantity of emission for different ranges. Highest is black and lowest is white.

Ranking page:

The ranking image will show fine as Unit Testing doing.

Download page:

The download link works fine as UnitTesting doing.

Confirming results:

HTTP responses, redirects, and user interfaces, and all functions work correctly.

Conclusion:

Based on the information that will be gathered from the integration testing, our team will be able to analyze the viability of our product and how well the technologies fit together. These integration tests will help ensure proper operation of the platform but is not enough to determine adequate functionality and limitations. Even with extensive unit and integration testing, bugs and issues are still likely to arise. The most reliable way to verify the stability, functionality, and ease of use, is to incorporate user testing alongside these previous tests. The next section will discuss usability testing.

4. Usability Testing

Usability testing is about making sure users have the desired experience, by making sure all our functions work during front end interaction as well. Our usability testing will be more straightforward, as our GUI will consist of a number of different buttons and menus for users to interact with. Testing will consist of making sure all these tools work properly, as well as trying to do things the users shouldn't be able to in order to make sure the functions handle those end cases properly as well. All of the following tests will

be carried out by ourselves, as well as asking average users to test if all these things work properly.

4.1 Initial Load

The first part of usability we will be testing is what the website looks like on initial load. We will be testing load times to make sure they are appropriate, and make whatever adjustments we can to make them as fast as possible.

4.2 Button to Change Year

We have a list of buttons that allow users to change the year being displayed. We will test that these change to the proper year and show the proper emissions maps, as well as having a smooth transition between the two years.

4.3 Button to Change Layer

Our other main set of buttons is to change which emissions layer is being shown. There are 11 different ones we have currently, so we will test to make sure that when the button is clicked it gets changed to the proper layer, as well as getting rid of the previous one.

4.4 Data points

Each state will have a data point that when clicked will show the population, total emissions, emissions per capita, as well as the information for whichever layer is being shown. We will test that all these points are present, clickable, and displaying the correct data.

4.5 Search Function

The last main thing to test in our main map is the search functionality. The user is able to search anywhere in the world, so we have to limit this to not show map data outside

the United States. We also must test that searching takes the user to the correct location on the map.

4.6 Emissions Ranking

Testing on this page of our project will mainly be about making sure it looks readable in a way that is easy for users to understand. There is less user interaction on this page, so there is less usability testing to be done.

4.7 Data Download

Our third and final page will have all the data we used on it available for download. Our testing will be to make sure it looks pleasing and easy to understand which file is which, as well as making sure all the files download properly.

5. Conclusion

To review the problem, our clients currently have pictures and videos of CO2 emissions. However, they want an interactive way to help people visualize the current state of emissions in the country, as well as view their own city and personal emissions to show how their actions are affecting the environment in where they live. 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 create an interactive map that contains all the information from our clients.

In this document we have covered our plans for Unit Testing, Integration Testing, and Usability Testing. These series of tests will ensure that our system is ready for a stable release at the end of the development period. By thoroughly testing the

main components of our system and gathering user feedback we will be able to deliver a maximally error-free, functional, and highly usable software product.