Software Requirements Specification

for

Climate Trace

(Central America)

Version 1.0

Prepared by Vitaliy Stepanov
Naoki Lucas
Andre Weertman
Max Ayala

WattTime

November 11, 2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Vitaliy Stepanov	11/11/20	New Use Case Diagram	2.1
Naoki	11/11/20	Added RTM	2.1

1. Introduction

1.1 Purpose

Climate Trace is an initiative to gather all the available greenhouse gas emissions and energy production data from across the globe into a single centralized source. The project will use a combination of satellites and AI to develop a real-time model that will help pinpoint major sources of air pollution from the electrical grid. This version update will include new data scrubbers for various countries in Central America including: Panama, Nicaragua, Honduras, and El Salvador. Additionally, this SRS will cover a new beta implementation of a visualization tool for the energy data and its accompanying forecasting program.

1.2 Product Scope

A major problem we face today is the inability to verify the multitudes of self-reported data. Climate Trace will utilize new technologies to better track the sources of pollution. In return we empower investors to green. Furthermore, better data verification will mean accountability to industries that may not be up to code of the Paris Climate Agreements.

The overall goal of this project is to add functionality to Climate Trace. Additionally, these updates will directly improve the availability and capability of various eco-conscious technologies like WattTime's Automatic Emission Reduction (AER). In the long run, we endeavor to make Climate Trace a reliable authority on the most current emissions data. This tool will continue to be publicly open and free source to promote a greener future. We aim to inspire companies to develop energy efficient software, institutions to study the effects of our energy consumption, and even individuals to promote public awareness.

For a more detailed description you can check out this article co-authored by Al Gore, former vice president, and Gavin McCormick, founder and executive director of WattTime. If you have questions, please see the Press Release Frequently Asked Questions.

1.3 References

Gore, Al and Gavin McCormick. "We Can Solve the Climate Crisis by Tracing Pollution Back to it's Sources. A New Coalition Will Make it Possible." Medium, 15 July 2019. Online. <https://medium.com/@algore/we-can-solve-the-climate-crisis-by-tracing-pollution-back-to-its-sources-4f535f91a8dd>

Roberts, David. "We'll soon know the exact air pollution from every power plant in the world. That's huge." Vox, 27 May 2019. Online. <https://www.vox.com/energy-and-environment/2019/5/7/18530811/global-power-plants-real-time-pollution-data>

WattTime. WattTime, 2020. <http://watttime.org>

2. Overall Description

2.1 Product Perspective

As an extension of an existing product, we are adding and utilizing scraper resources with WattTime. The application will make use of a database and create visual models for both real and forecasted data. This whole application, while in it of itself is new, does extends from the idea planted by WattTime. Our divergence from WattTime as our own product include features like being able to view our own Central American scrapped data further back in time and being provide a forecast for that data. With this comes our own database and tools that will be custom tailored to our pursuit, making this a self-contained product. The database will be made available to others with read-only access to specific points, and the application will have the ability to export data. This lets our users feel free to simply use or even develop off our platform.

2.2 Product Functions

Major functions the product must perform:

- Scrape websites for emissions data.
- Analyze satellite images for emissions amount.
- Verify the accuracy of reported website data with satellite images and realistic values.
- Store website data and satellite images in a database.
- Forecast emissions
- GUI will display 3D map of Central America and information such as Air Quality Index, Power Plants, demand, and more.
- GUI will have background music and play audio effects from user inputs.

Major functions the user must perform:

- Export data from our database.
- Select the past, present, and future visualize emission images and data.
- Filter by pollution type, hour, and day.
- Click to see location of satellites.
- Display cost of different energy uses depending on time.
- Display costs by using different energy sources at different time of day.
- Compare other sources of emission generators like from cars.

2.3 User Classes and Characteristics

We have two predominant user classes. Primarily, our country specific scrapers will be used by WattTime affiliates as an expanded arm of their global reach for energy emission data collection. As a subclass to that group, AER compatible devices within the scope of our scrapers will implement our data into their reduction algorithms, lowering energy peak (dirty) times, therefore smart-device users within our country list by lineage will be users.

The second predominant class of our users will be the general public within the scope of our scrapers that have internet access. Even without AER compatible devices, anyone can reduce peak “dirty” energy hours by simple habit changes. Our API will give tools to the general public that allow them to see and forecast their own local energy power plants demand. This information can empower simple changes to personal energy consumption (such as starting dishwashers, and washing machines, or battery charging devices and the lowest energy hours). We assume this will predominantly be green energy enthusiasts, and those who care about climate change (which should be everyone by the way), but depending on local energy consumption charge rates, we may also offer money saving capabilities for those within power plants who charge differing rates across the cleanliness spectrum of their energy demand.

As a smaller class, users will be climate and energy researchers, those not specific to WattTime and its affiliates. Climate change is a world-wide effort, and our data addition to Climate trace may end up being a part of anyone’s effort to continue an energy map and forecast of the world. We make our data completely exportable in multiple formats, giving any research team free access to essential information in the continued fight against the most existential threat to us as we know it, climate change.

2.4 Design and Implementation Constraints

The main constraints for this project involve the difficulties in gathering available emissions data. Not every country has formal regulations for the reporting of this data, and as such many areas go unreported, or perhaps worse, report false data. Furthermore, if the data exists, it can come in a variety of languages and formats, such as spreadsheets, csv files, or just tables in a webpage. Some data will update in real-time (say maybe every 5 minutes), while other data will only update every hour or perhaps even every day. In the end, it will be up to the AI software and satellite imaging to help fill in the holes left by the global data.

This project is also non-profit and has realistically no budget. As such we will utilize free technologies, such as MongoDB. This will however constrain our space for storing historical data which will further limit the number of viable machine learning algorithms we can use. Also, this project is under the jurisdiction of WattTime and their policies. As such, we are constrained to developing in pep-8 format, and the output of the scrapers will have to be a list of dictionaries.

Lastly, as for hardware constraints, we will be using raspberry pi’s to run the database.

2.5 Assumptions and Dependencies

Currently this project assumes the existence of viable machine learning algorithms that can be trained frequently with data that is then no longer necessary to store. In other words, we have assumed that low storage space will not impact the capabilities of our forecasting tool. Additionally, this project assumes that all data gathered from “reputable” sources is accurate. We know however, that this will not always be the case. Both assumptions will impact our tools’ accuracy.

The data scrapers naturally assume that the format of the data will not change over time. A scraper will effectively be rendered obsolete whenever the data moves to a new website, comes in a new form, and is updated to include more information. Sometimes this will mean a minor fix and other times it will mean starting again from scratch.

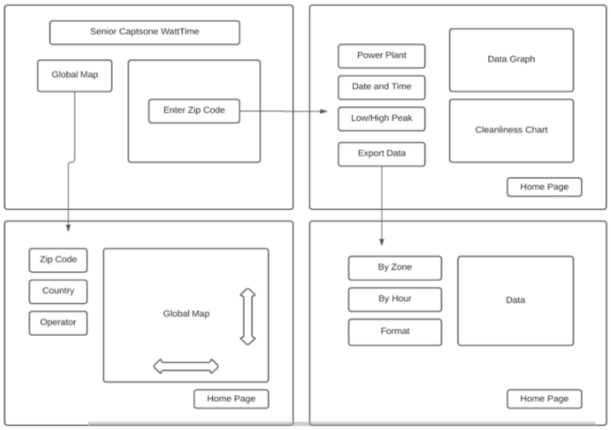
Lastly this project will have various outside dependencies. Primarily these are the database used for storing historical data (MongoDB), any satellite imaging software provided by WattTime, and unity for developing the visualization tool. Furthermore, there will be dependencies to each individual website from which data is being scraped.

3. Specific Requirements

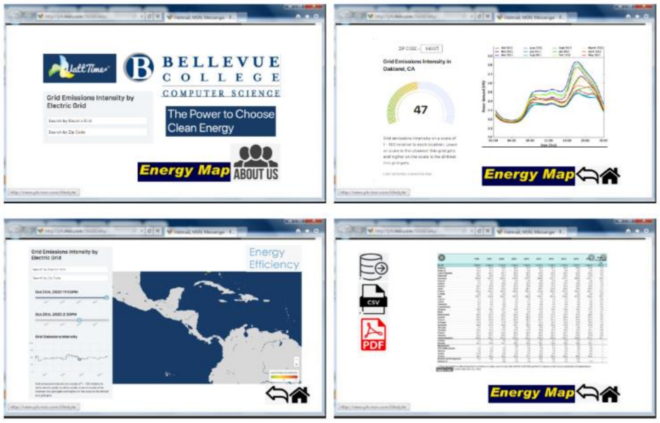
3.1 User Interfaces

Our online API will be accessible globally to anyone with internet access. It will consist of 4 main pages: a home, map, data graphics, and data export. Links between them will include a home screen and map button for direct access, and a return button. A visualization is included below and as a linked lucid chart in GUI_model, updated from the Wireframe_UX diagram.

WireFrame_UX



Unimodal


WattTime
BELLEVUE COLLEGE
COMPUTER SCIENCE
Grid Emissions Intensity by Electric Grid
The Power to Choose Clean Energy
Energy Map
ABOUT US
Grid Emissions Intensity in Oakland, CA
47
Energy Map
Grid Emissions Intensity by Electric Grid
Energy Efficiency
CSV
PDF
Energy Map

The top left page being our home page, with basic information about us and our project, and an entry box for a zipcode/country name that would lead you to our graphical representation page on the top right. All pages include a link to the energy map page in the bottom left, as well as our data export page in bottom right. On the energy map page, users can scroll or search specifically by zipcode/country, and from our data export page, users can export the dictionary list of data in a variety of formats.

3.2 Functional Requirements

Functional requirements are provided in the following use cases, and graphically represented in our Use Case Diagram. Generally, our scrappers (that are also used by WattTime) continuously create raw data that fills our database. Our GUI, accessible online has 4 main user functions for the user, that lead to their specific operation, that ends up grabbing data from our database to meet their request.

3.2.1 Data Scrapers

Data Scrapers are specific to individual country API websites. They are meant to find hourly based energy usage. Some have entire day records provided and others we will run hourly timed scrapers across their data.

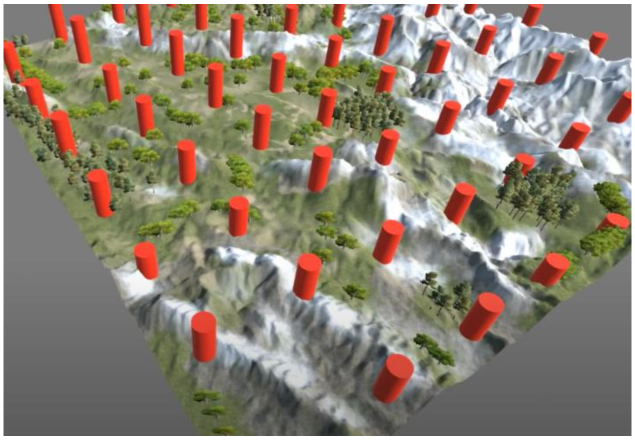
1. The data scraper must retrieve energy data from specific country websites.
2. The data scraper must return the data in dictionary format.

3.2.2 Visualization tool

This application will be able to display data in 3D with audio and be interactive.

1. The user can filter data by pollution type.
2. The user can view specific power plants.
3. The user can see past, present, and future data.
4. Any user can enter a zip code, or country, and access our cleanliness rating, meter, and hourly graph of the energy usage at that specific location
5. Any user can click our map feature, and scroll to any country in the world, and find our data cleanliness data mapped to the countries our scrapers pertain.
6. Our map feature will highlight countries by a ‘green’, ‘yellow’, or ‘red’ indicated their overall energy cleanly gauge.
7. Any user will be able to return to the home page from our map page, or by clicking on a country, go to our data page for that country
8. From our data page of a specific country/zip code, any user can export our raw data in dictionary format or a graphical representation of the hourly rating throughout the day.

We will use Unity 3D visualization data tools



3.3 Performance Requirements

3.3.1 Load

1. Up to 1000 users may use the product, database, concurrently.

3.3.2 Storage

1. Database must store 1 year of all data including satellite images
2. Robust. Back-up data.

3.3.3 Speed

1. Scrape data for each country in 10 seconds
2. User actions will change GUI under 0.5 seconds.
3. Requests for data will be fetched and displayed within 2 seconds.
4. Retraining ML algorithm daily will take a maximum of 3 minutes
5. Export of currently displayed data should be in less than 5 seconds
6. Load GUI in 5 seconds

3.3.4 Accuracy

1. Prediction is at least 90% accurate

3.4 Logical Database Requirements

The database will be used in a variety of ways by several machines. The database will be on its own machine and on another machine the scrapers; both currently being Raspberry Pi's. Calling one a database server and another a scraper server, the scraper server will have scrapers constantly running on custom timers to collect hourly data. Once this information is acquired, this scraper server then sends this data to our database. The scraper server will also do most of the maintenance for the database, such as removing old and unused data. The database will hold these data entries and should be consistent in both time and information when queried.

On the scraper server, since there will be a lot of idle time, the continuous machine learning algorithm will be computed on this machine as well. Once the algorithm is computed and the forecast is created, both will get sent to the database along with other useful data pertaining the training model. The next time a forecast/algorithm is computed, it will pull this data and any new datapoints from the scrapers for re-training. The server will also compare these forecasts to actual data as they eclipse and upload accuracy data per each power grid. The database will have portions of it made available with read-only access. The database is expected to run 24/7 and handle hundreds of queries a day at a minimum. As the project scales up, possibly a newer machine to handle thousands of queries an hour, however reliability amongst system components come first and this requirement is not severely bottlenecked by hardware.

3.5 Software Quality Attributes

Our system, with expansion in mind, is a flexible set of tools meant to apply to most/all countries. The database should readily accept data from any nation and the application should easily access this data. The only thing that might need extra work is how these objects are placed on a map. This means that as scrapers are built, the only concern should really be where on the map this data is presented. Scrapers will easily be interoperable and therefore maintainable. Anything presented to the user should be clear, consistent, and have useful links where applicable. Data should be presented to the user promptly when requested and the correctness and availability of data should be reliable.

The database will be on its own dedicated machine since it will hold a crucial role for this project and should focus on being reliable and available 24/7. The scrapers and the GUI depend on this. The database should also hold any data that should serve useful/necessary by either the application or the scraper tools such as user export requests or an update to the iteration of forecasting models.

3.6 RTM

Req. #	Requirement	Design Specification	Program Module	Test Specification	Test Case(s) Numbers	Successful Test Verification	Modification of Requirement	Remarks
1	Scraper must retrieve data and in dictionary format.	3.2.1 Data Scrapers	3.2.1.1					

2	User can filter data by pollution type	3.2.2 Visualization Tool	3.2.1.1					
3	User can view specific power plants	3.2.2 Visualization Tool	3.2.2.2					
4	User can see past, present, and future data	3.2.2 Visualization Tool	3.2.2.3					
5	Users can enter a zip code, or country, and access our cleanliness rating, meter, and hourly graph of the energy usage at that specific location	3.2.2 Visualization Tool	3.2.2.4					
6	Users can click our map feature, and scroll to any country in the world, and find our data cleanliness data mapped to the countries our scrapers pertain to.	3.2.2 Visualization Tool	3.2.2.5					
7	Map features will highlight countries by a 'green', 'yellow', or 'red' indicated their overall energy cleanly gauge	3.2.2 Visualization Tool	3.2.2.6					
8	Users will be able to return to the home page from our map	3.2.2 Visualization Tool	3.2.2.7					

	page, or by clicking on a country, go to our data page for that country							