Requirements Specifications

Team Clean Carbon

December 7, 2022

Sponsor: Allie (Alexander) Shenkin

Mentor: Vahid Nikoonejad Fard

Members: Curtis McHone, Richard McCue, Shayne Sellner,

Justin Stouffer, Jonathan Bloom

Version 2.0

Accepted as baseline requirements for the project:

____
Sponsor Signature

____
Team Lead Signature

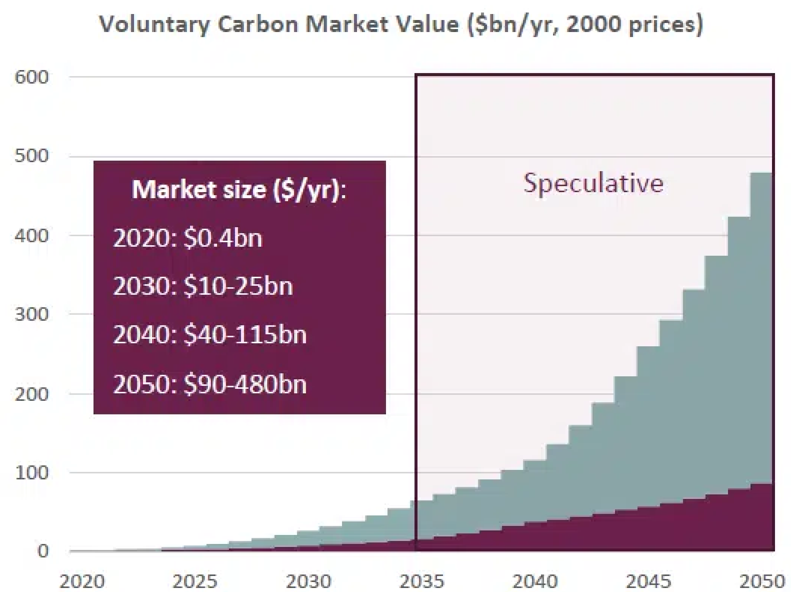
Table of Contents

1. Introduction

In recent decades climate change has become a growing concern as researchers have seen increasing amounts of carbon dioxide and other greenhouse gasses throughout our atmosphere. Carbon dioxide and other greenhouse gasses trap the sun's energy and heat within our atmosphere, helping create a habitable earth. However, when there is an over abundance of carbon dioxide and other greenhouse gasses within our atmosphere, the earth experiences a rise in global temperatures. With these new concerns, big companies have recently started to invest in climate change projects to help offset their carbon footprint. These companies can offset their carbon footprints by purchasing carbon credits. Carbon credits are sold by developers who reforest land and sell the carbon that the forest has taken in. Thus, this uptake of carbon dioxide from the forest offsets the emissions these companies produce.

The carbon market is relatively new, and therefore there are not many tools available to the average consumer or major companies. Being that the carbon market is new and not strictly regulated, the market value is incredibly small at the moment, valuing in at an estimated $500 million in 2020. Since carbon emissions are becoming such a large concern, the market value is predicted to grow exponentially. The market is expected to be valued between $90-480 billion in 2050. In order to get estimates about reforestation statistics such as carbon uptake, costs, etc. a consumer has to contact the individuals who manage the reforestation plot. With the new climate cooling service that our sponsor Allie Shenkin discovered, carbon credits will become an estimated 30% more profitable. With carbon credits being more profitable, developers will be more motivated to invest in reforestation projects in return for carbon credits. Allie's current tool helps solve this accessibility problem by allowing consumers to get estimates on their plot of land using the data from his new scientific finding. However, the tool that is being used is a proof of concept at the moment and will need to be reworked and expanded into a more comprehensive tool for consumers and developers.

Figure 1, Carbon Market Value Graph



The above figure depicts the estimated carbon market value in the upcoming years.
Source: https://carboncredits.com/the-ultimate-guide-to-understanding-carbon-credits/#4

This document will describe the requirements of this software system that have been outlined by our project sponsor, as well as our own team requirements when implementing this software system. Both functional and non-functional requirements will be described in this document.

2. Problem Statement

The current issue plaguing reforestation projects centered around the generation of carbon credits is the high upfront costs, long development and investment return times, and the uncertainty of the regulations in the carbon market. Currently there is no regulation in the voluntary carbon market which is encouraging many large companies to create or invest in reforestation projects that are not fair or ethical, as these carbon credits are cheaper than ethically sourced carbon credits. Some examples of the negatives of unethical reforestation projects would be poor management leading to a reduction in biodiversity of the area. These reforestation plots if done improperly could introduce potentially invasive and non-native species. These issues could lead to a lower number of available habitats for the native inhabitants of the forest. Another issue that can arise in reforestation plots is if the native population is adversely affected by the creation and upkeep of the plot.

Our sponsor's vision is to create a new tool that is more user friendly and more efficient than his current tool and the alternative tools that exist in the current carbon market. At its core, Allie plans to incorporate this into a business model by implementing usage based billing for this newly provided service. This will allow reforestation project managers to decrease the long investment return times and the overall profitability of the project as they will benefit overall from the 30% increase in carbon credits from Allie's research findings. The increased profitability of this new solution will also drive more people to use the service and increase the number of reforestation projects around the world.

3. Solution Vision

The system that we as a group are going to implement has four main software components to it. The first component is a user-friendly web-interface that allows users to draw on a map describing the plot of land that they wish to receive estimates. The second component will be a high performance back end that will be responsible for running the prediction system on the user supplied data. The third component will be a REST API that allows communication between the front end web interface and the high performance back end. The fourth and final component will be a simple database management system that will be responsible for storing user information. Below is a bulleted list of features that will be implemented throughout this project:

- Front End, Web Interface
    - An easy to use map that will allow the user to draw a polygon that designates the plot of land that they want to get estimates on
    - Map will be zoomable and scalable
    - Interface must be responsive and friendly to use
    - The interface will be able to upload the user supplied data to the back end

- The interface will be able to receive the results from the back end, then display them to the user
- Back End, High Performance Computer
    - Backend must be able to receive uploaded files and polygons from the web interface
    - Will run an efficient and accurate global prediction system that estimates the carbon credit count of the given plot of land
    - The prediction's runtime will be kept to acceptable times, which is determined by our project sponsor
    - Backend must be able to communicate the computed results back to the web interface for display
- Communication, REST API
    - The backend must be able to communicate with the web interface and vice versa through the API
    - The API will be able to handle large data files efficiently
    - The API will be fast and efficient when transferring data
    - Secure Encryption for security purposes
- Database, User Account Control
    - We need to store user access information such as usernames and password
    - All information stored in the database must be secured, as much of it is sensitive

While our new software system is going to be similar to our sponsor's current prototype, there are a few main differences. The main problem with our sponsor's prototype is that it is simply too slow. As it is currently running on the Google Earth Engine, it is lacking the efficiency that our sponsor desires. Another factor that makes this prototype slower is the fact that it is computing with very large datasets. Our software system is going to essentially be splitting up the prototype into multiple different software components, which will increase the efficiency of the overall system. Having a dedicated high performance back end will allow the global prediction system to run faster and more efficiently, leading to faster response times on the front end interface. The front end interface that we are going to create will be more professional and user friendly than the current prototype as well.

Our software system will allow any user to get estimates on the number of carbon credits that they can expect to get from any plot of land. With easier access to tools in the carbon market, we hope that more reforestation projects will be funded, especially considering that this new finding will increase the number of carbon credits per project by up to 30%. With this increase in carbon credits, the hope is that it will entice more developers to invest into reforestation efforts, which will slow down and possibly reverse climate change.

4. Project Requirements

While there are many requirements for this project, there are four main software pieces to our system. These four main pieces are the frontend, the backend, and a REST API that will allow communication between the frontend, backend and the database.

The general frontend requirements for our project are as follows:

- The frontend should host a map that the user is able to draw a polygon on, designating the plot of land that they wish to get estimates on.
- The frontend should be able to send this polygon to the backend for computing.
- The frontend should be able to display the computed results to the user. All of these requirements will be met while maintaining a user friendly interface.

The general backend requirements for the system are as follows:

- The backend should run a global prediction system, written in python, that is able to estimate the number of carbon credits for any given plot of land.
- The backend will be able to run these predictions in a reasonable amount of time, which is determined by our project sponsor.
- The backend will finally return the computed results back to the frontend for display.

There are not many requirements for the REST API, however the general API requirements for the system are as follows:

- The REST API must be able to allow communication between the front and backend.

- This communication should be efficient and fast to ensure that there are no bottlenecks in our system.

Finally, there are also not many requirements for the database. The general database requirements are as follows:

- The database will be used to allow for a simple user access control system.
- Stretch goals could add more requirements to the database, but at the moment, a simple user access control system is all that is required from the database.

Functional Requirements

The following section will describe all of the functional requirements that are necessary for our software system. Functional requirements are specific functions in the software that we as a team are obligated to fulfill. The functional requirements describe the technical aspects to our requirements, rather than the quality or usability of our software, which are non-functional requirements.

Frontend

Our front end will be hosted via a web interface. This user-oriented interface will present a global map that can be zoomed in and out of, as well as rotated and moved around. The goal of the front end is to present a user-friendly, easy to use, and responsive global map similar to that of Google Earth, but with the addition of more tools for layers and drawing. The user will be able to manipulate the map to their liking to view the area of their choosing. From there they will be able to draw a polygon over the area they are researching for a reforestation plot. Once the polygon is drawn on the map there will be several options to choose from. The user can run the prediction on various raster layers corresponding to previous years data. The user interface must be able to accept user supplied data and send that to the back end.

Once the data is computed, the front end will receive the data and carbon credit prediction from the back end and display that information to the user. The user can then use that information or make another prediction on a different area.

Backend

The backend system for our program will be one of the main components of our software system. Originally our sponsor and team were planning to use NAU's super computer, Monsoon, for our backend; however, due to recent complications that have arisen, we have started to shift our research and focus onto Amazon Web Services (AWS). The most crucial requirement on our backend is that the global prediction system should be able to run fast while still providing users with accurate information. This entails many other major concerns, such as ensuring that data in the raster files are being stored and read in the correct format. The global prediction system will need to be able to compute the desired results in acceptable times. These times will be determined by our project sponsor and might vary throughout the project. The backend in conjunction with the database must also be able to support a simple user access control system. This system will be able to allow our sponsor to give users access to the program. The backend will also be able to support a REST API that allows for communication between the frontend and the backend of our program. The performance of this backend system will be discussed more in depth further in the performance requirements section of this document.

As a team we have thought of multiple stretch goals that we would want to implement if and when we finish the main requirements of the project. If we reach our stretch goals, we would like to develop an ArcGIS plugin for ESRI's ArcGIS tool so that other developers could easily add our program into a portion of theirs. Another stretch goal as a team we would like to develop is the ability to read and calculate information given from a raw shape file. This feature would be helpful to our sponsor and users because they would no longer have to use the entire GUI functionality in order to receive specific calculations.

Overall, the backend of this project is one of the largest components. In order for our team to be successful during the remainder of this project. We will need to create an efficient backend that has the ability to hold and run the other features of our program.

REST API

While the REST API being an important and necessary part of our system, the relatively straightforward nature of the API leads it to not have as many requirements as the rest of the system. The most crucial requirement is that the REST API should facilitate communication between the front end and the backend. This will aid in ensuring a smooth and efficient experience for the user. However, in order to facilitate the communication effectively the API needs to be able to send over the necessary information of a polygon so that we can run the calculations in the backend. After running the calculations in the backend the API needs to be able to return all of the necessary results back to the front end for display to the user. Finally, the API will be interacting with the database to ensure that each interaction is tied to a specific user. This ensures that only certain users are using the service and allow for the future scalability of teaching usage of the system for billing purposes.

Database

Our database will be used for storing user information and possibly user queries and billing information. A user access control system is the main use for our database backend, but future additions may be added as well. These future additions include the storage of prior user spatial queries, which need to be accessed within the database. The database must also be able to extract information from these spatial queries using spatial object mapping. Along with those, the database must be hosted on a secure server and must implement encryption of the data along with other security measures. Sending and receiving data to and from the server must be fast, with quick response times, and it also must be able to access the information in a structured way. The database will need to be able to be scaled up to accommodate future increase in users, as well as provide information and statistics about how the users use our system. The database backend must also be able to be compatible with our web front end, specifically with the Django framework that will be used to build our website. The implementation of our database with the framework must be efficient and simple to work with, with communication between the front end and the database being easily facilitated by the framework. Our database must also be inexpensive, with the option to upgrade later on for more features as well as more storage.

Performance Requirements

The performance requirements for our project describe the metrics that will be used to measure how efficient and effective our software system is. Each section of our software system will be efficient enough to ensure a friendly and positive user experience.

Frontend

The front end of our program is expected to run much faster than the prototype supplied by our sponsor. The current program is very sluggish and not user friendly at all. It is very difficult to upload raster data or change the layers on which the prediction will be based. Besides, the time it takes to receive the prediction is annoyingly slow. The front end will get data from the user via the polygon and send it to the back end. Separating the front and back end will speed up the computation time tremendously and will improve the user experience.

Backend

There are not many performance requirements for our backend, as there are not too many moving parts associated with the backend. The main performance requirement for the backend system is that the global prediction system should be able to compute the amount of carbon credits for any given plot of land within acceptable times. The time limit that was given to us by our project sponsor is 10 seconds for every 1000 hectares of land. This means that if a user gives the system a 1000 hectare plot of land, they can expect the results back to them within at the maximum 10 seconds. Once the software is implemented, this time limit may change, depending on our project sponsors wishes.

As we do not have as strict requirements for the other sections of our software system at this moment, there are no hard guidelines on the performance of these sections in conjunction with the backend. All of the other sections of the software system will be able to communicate and collaborate with the backend efficiently. The backend will be able to retrieve user information from the database quickly so as not to delay user access to our software. The backend will be able to communicate the computed results to the API quickly so that there are no bottlenecks when giving the users back their computed results.

If our stretch goals are met, further performance upgrades will be implemented such as tiling rasters. Other performance improvements may be implemented if there is time within the scope of this project.

REST API

The REST API of our program is expected to run in an efficient and tentative manner. The API only needs to be ready whenever a user is on the system to transfer information back and forth between the front and back ends of the application. The major component of the API that we can optimize is having it ready for the user when needed as once the files that are being transferred are away from the API it is out of our control until it reaches their destination. Finally the API will communicate with the database to store user queries to ensure that optimal time and speed has been met.

Database

Overall, the database is relatively straightforward in terms of nonfunctional/performance requirements. The database will be required to be a stable, fast mode of storing user data and that also allows data to be changed and updated efficiently. Another requirement for the database includes being able to rapidly respond to user requests, and also being able to calculate spatial queries if desired. Along with this, the database must be able to handle and store multiple different user accounts, all while keeping the data secure and reliable.

Environmental Requirements

The environmental requirements for this project are the requirements that describe the specific technologies and softwares that will be used, which were determined by our project sponsor.

Front End

As specified by our sponsor our program should be a web application. This allows users across almost every platform to access it. Whether it be a desktop, laptop, mobile, or any operating system, our application will be able to be used. Our web application needs to have some elements as specified by our sponsor and us, to make the application dynamic, user friendly, and responsive. We will implement many design elements, such as menu bars and pop up windows to display important information such as coordinates, selected layers, tools, carbon credit prediction, and other display elements. This will allow the user to customize their screen and display the information they want to. Changing this layer will change what the back end will run its prediction off of. These changes will be a huge upgrade from the prototype because there currently are not any options to display different information or any customizability in how the screen interacts with the user.

Backend

There are only two main environmental requirements for our project when discussing the backend. The first requirement is that the backend should be hosted on a linux based operating system, that is able to crunch large datasets efficiently. At the beginning of the project, this linux based server was going to be the high performance computing cluster, Monsoon, here at NAU. However, within the past few weeks, we have discovered that we might not be able to use Monsoon, so this is still blackboxed at the moment. The other main environmental requirement for our backend is that the global prediction system must be written in python.

REST API

In general, there are not any environmental constraints imposed on the REST API. The only constraint set by our sponsor is that the API needed to exist so that this project would work without human intervention and could be seen as a cohesive project with the frontend and backend being connected.

Database

In general, the environmental requirements for our database are straightforward. We will need a server that will be able to host our database, whether that be AWS or another service that can run our database software effectively. Along with the server, we will need a software package to actually store the user data, along with other possible data forms. This database will need to be able to be smoothly run on the server of our choice. The last environmental requirement is the language we will be using in our database system. This language will go hand in hand with the database software we choose, but it will need to be easily understandable as well as powerful.

5. Potential Risks

There are many risks associated with developing a software system for the public. The main risks in our project include account security, user privacy, and accurate data representation. In order for users to have access to our product they first would have to create an account and become verified. This includes storing sensitive information, such as official names, birthdays, emails, and passwords, within our databases. As a team we need to make sure that a secure encryption technique is implemented in order to ensure that user information remains safe. To do this we plan to implement an existing encryption package within the database, or by using encryption methods that the API already offers. This encryption API will be on both sides of our program so we can support pre and post encryption. All or most of the sensitive information we receive will be sent through our API, meaning not only will our database be secure, but our REST API will be as well.

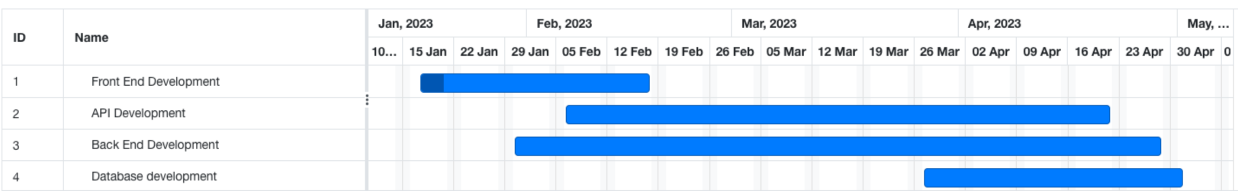
Another major risk we have to address is user privacy. This somewhat goes along with account security but in our opinion fits two different categories. Once a user creates an account and begins to run searches, our program will keep track of what that user is searching for and the parameters along with it. We want to ensure our users that their information will remain private. To do this we will implement another type of encryption as well as administrator passwords to add additional layers of protection.

The last major risk we must consider is to guarantee that our calculations are accurate. This is extremely important because if our calculations are wrong or less than expected, investors would in turn lose money and time. This is a crucial step of our project because without these calculations, our program would have no use. To minimize this risk we will be making sure the calculations are returning expected results and talking with our sponsor extensively.

There aren't many risks that come along with our project however, the risks we do have are fairly major. As a team we want to ensure users that their information and themselves are secure and protected and that the information they are receiving is up to date and correct.

6. Project Plan

Figure 2, Gantt Chart



The above figure is the gantt chart made by the team depicting the project milestones for the Spring 2023 Semester.
Source: Team Clean Carbon

Throughout the Spring 2023 semester we plan to reach several milestones for our project. We have already started the front end development where we have already created a basic web application that hosts an interactive global map as well as a polygon tool that can highlight specific areas on the map. With the development we have completed already this puts us at 10% progress for the front end development. We plan on breaking up our semester to focus on development of the different parts. We have designated the first part of the semester to focusing on the front end. This will ensure that we will have a user friendly interface that we have something to show for. We will then focus our attention on API development as well as the backend. Once we have made good progress on the backend and API we will work on the database alongside those. This will ensure that everything communicates properly and is functioning the way we would like it to.

7. Conclusion

Developing a solution to climate change is no easy task; however, our team is dedicated to developing a product that will allow different people, companies, organizations, and others to be a part of the broader solution. Our product has the potential to kickstart large scale reforestation around the world, something that is desperately needed to reduce the amount of carbon dioxide in the atmosphere. Currently, companies looking to offset their carbon footprint have many options, one of them being the purchase of carbon credits. Carbon credits act as a certification that allows a company to produce a certain amount of carbon, in exchange for their purchase of land to be reforested. However, this is not very attractive to many companies due to the time it takes them to see a return on their investment. Our solution to this issue comes with help from our sponsor, Allie Shenkin, who has discovered a way to produce 30% more carbon credits per plot of land. Using this discovery we will create a product that allows companies to calculate how many credits they will earn per plot of land, all through an easy and streamlined process.

The requirements outlined in this document provide the structure that our design process will follow. Following these functional, nonfunctional, and environmental requirements will ensure that our product maintains the level of professionalism that our team and our sponsor expects. The risks that are also outlined in this document will be continuously reviewed by our team; this will make sure the proper safeguards are being followed to prevent any undesirable outcomes. Our team is very excited about the future development of our product, and we are very grateful for the opportunity to work on something of this importance. We have made a great deal of progress so far in preparation for the development process and will continue to finalize our research in the coming weeks.