Software Design Document

Team Clean Carbon

February 16th, 2023

Sponsor: Allie (Alexander) Shenkin

Mentor: Vahid Nikoonejad Fard

Members: Curtis McHone, Richard McCue, Shayne Sellner, Justin Stouffer, Jonathan Bloom

Version 2.0

Accepted as a baseline design for the project:

______________________________ Sponsor Signature

______________________________ Team Lead Signature

Table of Contents

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 habitable earth. However, when there is an overabundance 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 it 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. 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 plots 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. This document will outline the major design features and decisions that will be present in this software system. By the end of the document, the architecture and design of this system will be apparent.

Implementation Overview

Frontend

As an overview of our frontend interface, we will host a web application on an AWS instance. This web application will display a world map with multiple raster layers to select from. The user will be able to select a raster layer and a polygon drawing tool from a toolbar we will have on the side of the map. With this, the user can draw a polygon on the map that they would like to receive an estimation for the number of carbon credits they can sell for that designated plot of land. The application will then display the estimation in an easy-to-understand way for the user. We will be implementing a javascript library to handle all the features of the map and drawing the polygon. Once the user draws the polygon the web interface will send the coordinates to the back end so the prediction can be made. Here are the key features of the front end:

- 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
- The map will be zoomable and scalable
- The interface must be responsive and friendly to use
- The interface will be able to upload the user-supplied data to the back end

Backend

The back-end section of this software system is not the most complex. There are a few key requirements that our backend must adhere to, these requirements are:

- The ability to support and/or host the frontend interface and database
- The ability to communicate with the frontend interface and database
- Able to store and access the raster files correctly
- Must be able to correctly run the global prediction system with user-supplied input

To achieve these requirements, key design features are necessary. As of now, the front end and back end are going to be hosted on the same AWS EC2 server, because of this, communication between the two will be very much simplified. Since these components are located on the same server, we will not need to implement a REST API or any external communication pipeline in order for our frontend to communicate to our backend. The backend is going to be constructed and organized in a way that allows for correct and easy raster storage/retrieval. To complete the backend, the global prediction system is going to be written in python, with the guidance of our project sponsor to ensure accuracy.

Architectural Overview

Frontend

The frontend interface will consist of one main component that will display the map to the user. Some of the sub-components that will need to be included in the web interface are the toolbar with tools, a user login, a place to select a raster layer, and sending the polygon coordinates to the back end.

The web interface will be hosted on an AWS instance and will implement OpenLayers, a javascript library. This library will handle all the big challenges of using and showing a map to the user. This library will give us the functionality to implement zoom, pan, scale, and rotate, which are all important features of great maps such as Google Maps and Bing Maps.

We will also implement a place for users to login. This will allow users to pick up where they left off and in the future as a stretch goal implement billing. The usage based billing would allow our sponsor to be able to charge for this software depending on how much the user is actually using this tool.

Along with the display features, OpenLayers will take care of the tools we will implement as well. We will use this library to include a polygon tool to draw over an area of land that the user would like to get a prediction for. While the polygon tool is most likely our most important tool we will have a full toolbar of options such as polygon shape, years to run the prediction off of, a country selection dropbox, and what raster to run the prediction on.

Backend

The backend system will have a few main components that will work together to create a seamless backend. These components consist of a global prediction system, the ability to communicate to the front end, a database, user login with security, and correctly storing large scale raster files.

The global prediction system is one of the most critical components of the backend system, as this is the core of the entire software system. The global prediction system is going to be responsible for accurately estimating the number of carbon credits that a user can expect from a given plot of land. With our project sponsor's new research findings, we are hoping that this global prediction system will convince more developers to invest in reforestation, as there will be a massive increase in profitability. Many of the fine details surrounding the global prediction system are still confidential though, so the details may not be discussed further in this document.

The communication between the front end and the back end is also very important, as communicating the user input to the back end and the results to the front end is a crucial aspect of this system. As of now, we are no longer implementing an API as our core communication method since our frontend, backend, and database are all going to be on the same server. This simplifies the communication aspect, as we do not need to communicate with any other systems outside of our own.

Another component of our backend system is creating a secure login network for users. This feature will be connected to many other parts of our program including the frontend system and the database. In order for this feature to work and be efficient we need to incorporate a secure encryption formula or API when sending and receiving information from our database. This ensures that if there is ever a breach in our system, sensitive user information will remain safe.

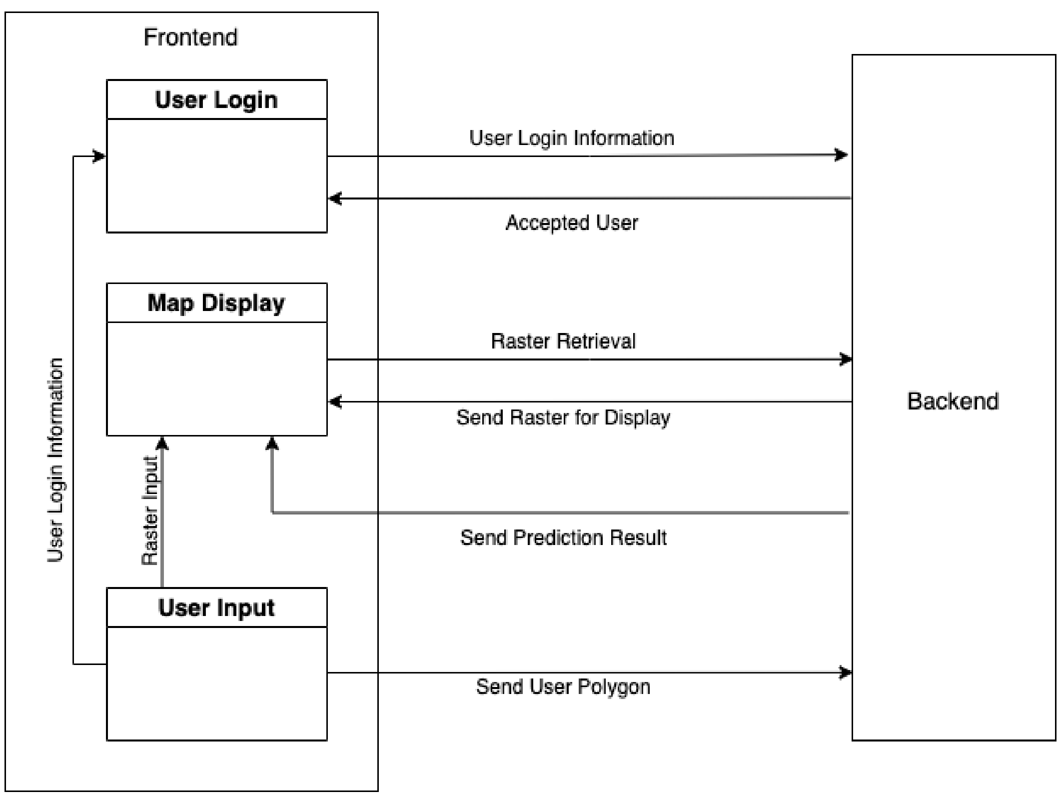
The last major component of the backend is the storage of raster files. Our raster files will be stored in the correct format so that they can be easily accessible by the global prediction system for predictions. We may also look into pre-computing some rasters to speed up our prediction system, so this also correlates with this component. Many aspects of this component are still up in the air, and we will be working with our project sponsor to ensure that the rasters are stored the way that they wish.

Module and Interface Descriptions

Frontend

The frontend of our project will be mainly responsible for grabbing the user input, sending results, and displaying a map and results to the user. As previously mentioned we will have a map displayed as our main website, a toolbar for user input and a place for the user to log in. The following UML diagram shows how everything will communicate with each other as well as sending the necessary information to the back end.

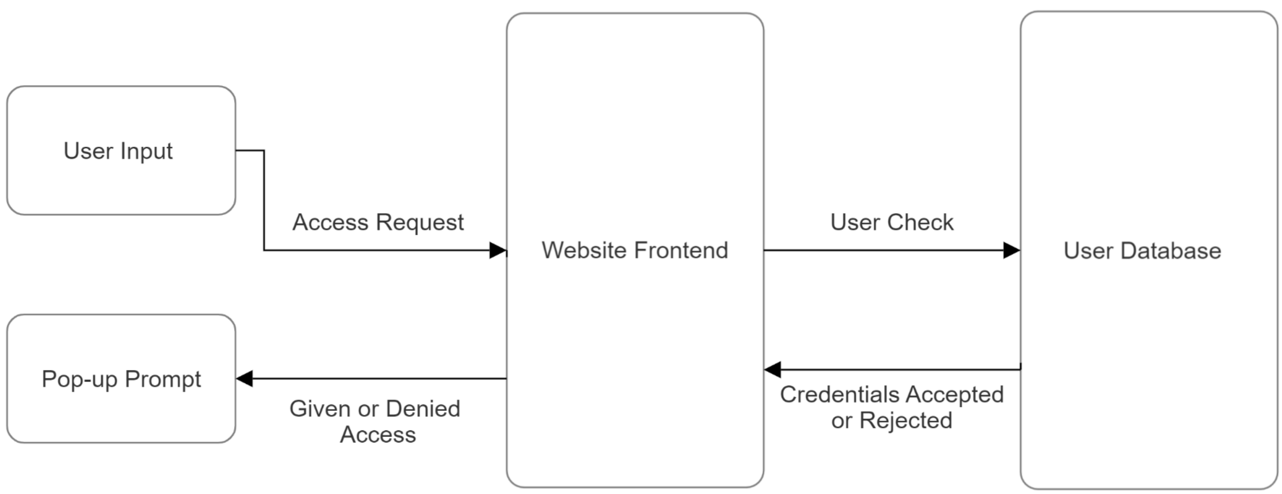
Figure 1. UML Diagram of our frontend



Frontend: Login Function

Along with the aspect of having the results returned to the web interface, and as previously mentioned before, the user will be able to log in with credentials and be able to use the system. This requires a database to store user information, as well as having a predefined set of users who are given permission to access the website and its tools. Our sponsor will initially determine who will be given permission to interact with the system, and once the user is added to the database, they will have access to the system. The following UML diagram illustrates how the user login module will function.

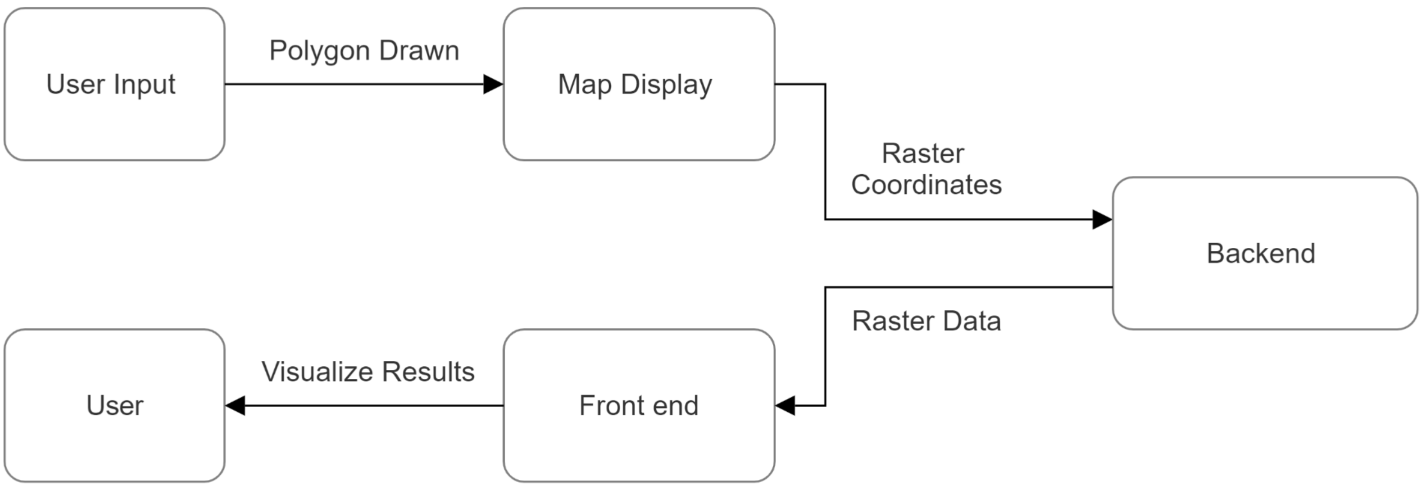
Figure 2. UML Diagram of Login Function



Frontend: Map Function

Another key part of our frontend includes our map display and its functions, of which the user will be interacting with primarily. The user will be able to scroll around the map, and zoom in when desired. Once the user has found a location they are interested in, they will be able to draw a polygon around that area and send it to the backend for computation. The front end needs to perform all of these functions up until the data is sent to the backend to process. Below is a UML diagram showing how this will be implemented.

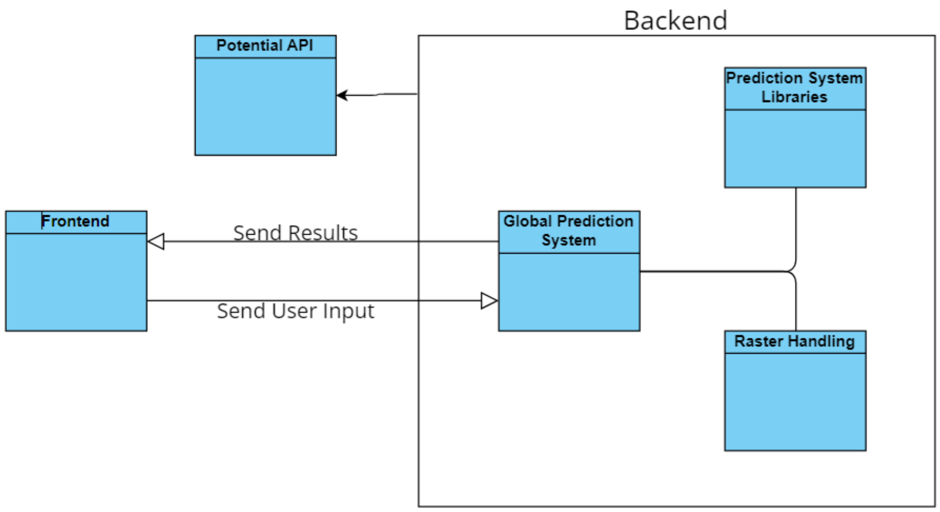
Figure 3. UML Diagram of Map display function



Backend

As previously mentioned, the backend of our software system is going to be mainly responsible for running the global prediction system with the user-supplied input. This aspect of the backend is one of the main components of this software system. Many of the other components of this software system are supportive features of this component, as this component cannot stand on its own. For example, our front-end system is simply an interface that the user can use to interact with the backend global prediction system.

Figure 4. UML Diagram of our backend

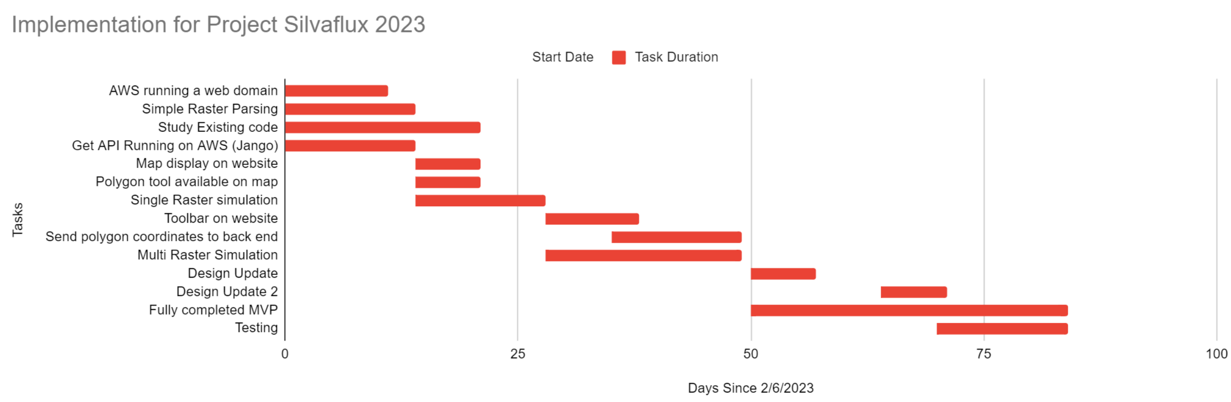


The diagram above depicts a rough UML diagram of our backend system and how it will be communicating to the front end. Our global prediction system is going to be the main component inside of the backend, but it will be accessing other components. The first component it will access is the front end so that it may receive the user input for processing. Our global prediction system will also be going to access outside libraries for computing, as well as our raster handling/storage component. The global prediction system will then return the computed results to the front end. Our project sponsor has also wished that we create an API to support further development of this system if this goal can be reached. Although we do not need the API, setting an API up would be very beneficial to our sponsor and their development after we move on from this project. There will be no public-facing interface for our backend.

Implementation Plan

Team Clean Carbons' development cycle will take place over this semester and will use an iterative development approach to make the final product as reliable and efficient as possible. Development has begun with our initialization of the AWS server and a pre-alpha tech demo of our front-end web interface. Shown below is our teams' overall schedule with various milestones outlined for us to maintain our production goal in time. Throughout this project, we will test each section thoroughly and then set up a large final testing suite at the end of production. Additionally, throughout the development process, we will garner feedback from our client and work on quickly implementing any changes needed through our testing phases. This will ensure that our project will be robust and efficient through its processes.

Figure 5. Gantt Chart Project Plan



As shown, our development plan contains many tasks that run parallel to each other and that is because we plan on allotting the backend work and frontend work to different members of the group based on their preferences. With this allocation will have two front-end developers (Curtis McHone and Justin Stouffer), two backend developers (Richard McCue and Shayne Sellner), and one full-stack developer working on both ends of the project (Jonathan Bloom). This separation of development allows the members of our team to work in small pairings throughout the project, ensuring that we maintain the benefits of paired programming. At this point, our team has completed two initial tasks: Map display on the website and the Polygon tool available on the map. Currently, the team is focused on the important task of studying the initial code of our project's tech demo supplied by our sponsor. We are making certain that we understand every aspect of the tech demo so that we can work to improve what was given and so we can implement it on a more robust and scalable platform.

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, and organizations to be a part of the broader solution. Our product has the potential to kickstart large-scale reforestation projects around the world, something that is desperately needed in order to minimize the effects of climate change. Currently, companies looking to offset their carbon footprint have many options, one of them being to purchase 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 and monitored 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 final 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.