

9.17.15 – Capstone Project Report One– TBTB

During the first two weeks of this project we have completed the initial planning stages of our project and began to implement it at a very basic and structural level. We have decided to essentially use a WISA stack for our web application. We will have a Windows machine use IIS and SQL Server 2014 to host our database and then we will use a .NET MVC web application using C# and JavaScript to create the application itself. We will create spatial indexes on the locations with SQL Server 2014 and it's built in spatial indexing functions to ensure our database runs at a sufficiently quick pace to satisfy the need of our end users. Then we will use Entity Framework to connect with the database and a C# backend to serve the data to the JavaScript to be displayed to the client. For the visualization aspect we will be using Leaflet for the map and d3.js for the visualizations on top of the map.

We have divided up our team into two parts to better tackle our obstacles and we will be operating in an agile environment: Tyler and Tom will be working on the SQL server and backend C# code mainly, while Brendan and Bill will be working on the frontend JavaScript. This is only to give the project structure because all members will have to take part in all tasks as the project goes on. This week the teams have been doing some preparation research, testing and small implementations of their respective parts, the specifics follow:

Tyler:

This week I have worked with the team to get them all up to speed on the design we will be following for this project as I have assumed the 'project leader' role because I have used these frameworks before and have the big picture of the project in mind. I have been putting together the initial Visual Studio solution so everyone may use it, and then working out all the kinks on the individual machines. I have worked a lot with this web stack at my current job and my group has never had experience with it, because of this they wanted to learn and I felt it to be a great design for our project. I have also taken the role of writing the documents and keeping track of the changes and the assignments on Github.

I will be setting up a remote personal server to host the groups database and the master branch of the project so we may simply hit a web URL to view the project. I also helped Tom out a little bit with some of the SQL stuff and we have been collaborating when possible to work out any kinks. I have then helped everyone get used to the ideology behind MVC, and how to build a project in it. I will continue to do this and work on the server side code as soon as the database is closer to completion.

Brendan Gray:

During my three weeks in Capstone, I have learned much of a JavaScript library called D3.js (data-driven documents), in addition to another JavaScript library called Leaflet.

D3.js is a library that enables the user to create interactive data visualizations and representations, created in web browsers using mainly HTML. Objects created as a result can be highly customized. It can read data in multiple formats such as CSV files and GeoJSON.

Leaflet is another client-side library that allows the user to build a mapping application using a web browser. One of its strong suits is loading data from a GeoJSON file. Another plus

is that the overhead is smaller, and it has a newer, more modern code base, utilizing HTML5 and CSS3. It can create interactive layers and can be customized.

Using these libraries, I was able to create a rough draft map of New York City, and I was able to place points on the map, such as circles and dots.

Bill:

So far I have been using the leaflet library and understanding how the library works by using the map box to get the map. After getting a map to show up in the leaflet JavaScript library. My next task is to get the Leaflet library to link with the backend which is C# or .Net framework with the SQL database. So far I can get the map to show up a single project but the map from leaflet is not showing up in Visual Studio.

Tom Taylor:

We are using SQL Server 2014 for the database backend. The two most time consuming parts of this so far have been analyzing the raw data and at the same time figuring out how SQL handles geospatial data in general. I have written a very involved query that works on importing the raw data into staging tables, then joining those tables, and finally parsing that join table to remove what is considered invalid data. I have also been researching all aspects of SQL and geospatial data as this whole concept is new to me.

One of the links you provided, related to the raw data, had a document attached to it that talked some about the raw data in general. It stated that perhaps up to 7.5% of the data did not seem to make sense. I am already seeing this as I am working through one of the months of data. In just looking at the latitude and longitude data there are hundreds of thousands of rows that plot points that are nowhere near New York City or even worse, actually contain blanks or zeros or points near zero. I have also seen that there are some issues related to the date and times. As an example I found the latest dropoff date and time on a month of data was several days beyond the end of that month. Looking at that complete row of data it just does not make sense as the trip was just in the hundreds of seconds. So there is more work needing to be done regarding data cleanup

With SQL and geospatial data I have figured out how to properly get the data into the table using the SQL geography data type. This involves taking the existing latitude and longitude data and using built in procedures to change each longitude and latitude pair into the appropriate geography data. I also learned that this data has to be indexed as it is being generated into the table. I have also started learning about creating geospatial queries and what is necessary to return valid results. My first successful attempt at this has been returning distance in meters from a nearest point query for the closest 50 points. On a database that contains only one day of data the results are returned in only a couple of hundred milliseconds, while on a database that is for one month of data the query results take just over five seconds. This length of time is concerning so I have started looking into how perhaps these queries can be done faster.