


Michael Wu

 Michael Wu
Software Engineer Intern, IT Corp Apps

Michael Wu - Software Engineer Intern

Hello! I'm the Software Engineer Intern on the Corporate Applications team. I'm currently enrolled at Emory University pursuing a degree in CS. Last summer I interned at the US Environmental Protection Agency. Don't tell them, but Blizzard is a much better place to intern at! Before that, I was an EMT for Lifeline LLC. I'm blown away that I've found my way here to Blizzard. Last summer I got the amazing opportunity to visit Blizzard's campus as a part of Tespa's Summer Leadership Academy. I think that I've always had a love for the worlds and experiences that Blizzard cultivates but I never thought I was good enough to make it here. My summer here, though remote, has honestly been an amazing experience and I hope that my work here makes you all proud that I interned here too. Please check out my work and reach out to me if you find it interesting!

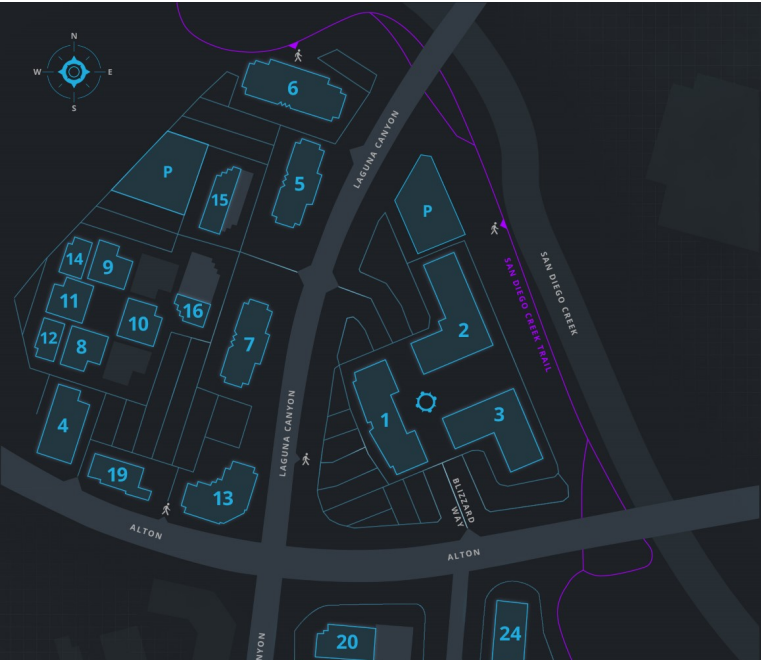


This is me!

```
1 import knowledge from "blizzard";
2 export default {
3   data() {
4     return {
5       name: "Michael Wu",
6       team: "IT Corporate Applications",
7       school: "Emory University",
8       degree: "Computer Science",
9       email: "michael.wu@emory.edu",
10      phone: "781-775-1678",
11      battlenet: "Zenthr#1479",
12      slack: "micwu",
13      linkedIn: "https://www.linkedin.com/in/michaelwu123/",
14    };
15  },
16};
```

Problem Statement

The goal of the ThinkGlobally project is to associate real world locations with their corresponding geographical coordinates. It would be nice to know the exact latitude and longitude of every single conference room so that conference room displays could be used to locate nearby amenities and other points of interest. While placing pins on Google Maps may allow for external mapping, we need to develop a solution that allows us to navigate the internal structure of the buildings. Conveniently, Blizznet has images of building floor plans that we can utilize to help map out each location.



[Skip to main content](#)



How do we map Blizzard's campus?

Image to Map:

https://blizznet.blizzard.net/assets/images/directory

Load

Key:

Map Name:

Altitude:

bldn-4

Building 4

56

X:

25.77

Latitude:

33.6578

Y:

7.76

Longitude:

117.7669

X:

85.22

Latitude:

33.6581

Y:

70.2

Longitude:

117.7672

Create

Problem Solution

To tackle this problem, I worked on creating software which allowed setting the latitudes and longitudes of known locations on an image. By knowing the coordinates of at least two points on a map, we can mathematically calculate the the coordinates at every single location on said map. From there, we can set points of interest with any relevant metadata such as capacity and provided technology (conferencing, monitors, etc). This really powerful for us in the sense that we can utilize this across several different platforms. For example, we could use this map out a HotS map, an area of LA, or anything else that has a layout. Once we have these corresponding points of interest, we can then sort them via relevance or any other form of data we would like. For example, this could be implemented into the Conference Room Displays so that nearby conference rooms could be displayed. Or a filter could be applied such that only conference rooms with a capacity of over 10 are shown. The applications for the software are endless!

Creating Maps

Here is where we can set up maps to be populated with points of interest. Each map has two pins that can be dragged around and dropped onto the image. By specifying each pins' geographical coordinates, we now

Skip to main content

API.

Editing a Map

Editing a POI

Viewing Populated Maps

Now that we have all of our maps and points of interest, we can pull up any previously created map. Since the points of interests now have their latitudes and longitudes as a part of their information, the map automatically populates with all of the relevant points of interest. This means that if you had recorded the Orc statue on another map, it will be displayed on the current map if it is close enough! That means points of interest are only ever created once and shared across every single map!

Creating Points of Interest

This is the page for creating points of interest. First we load up a map from our list of previously created maps. We pull this information from the back-end API. Since we know the latitude and longitude of every pixel on the map, we can simply just drag a pin around and the coordinates will dynamically update as we move it around. Once we settle on a location, it is as simple as filling out the relevant data and saving it to the database! We can place as many points of interest as we on each map and it will all be saved.

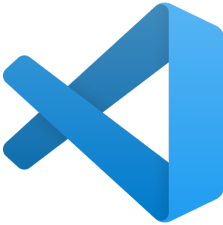
We could even map out a HotS maps!

Curious to see more? Check out the actual website here yourself. Or just visit /dev-thinkglobally!

ThinkGlobally

The Deployment Pipeline

In order to get ThinkGlobally to your computer, I needed to deploy the project from Visual

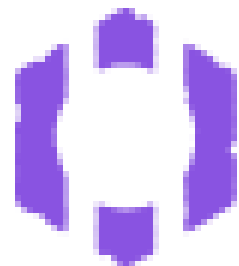


Studio Code to /dev-thinkglobally. Let's see how I did it!



3 - Leeroy

Whenever I merge into the develop branch in Ghosthub, Leeroy will build test and deploy that branch. Specifically, Leeroy is the Corporate Application instance of Jenkins.



6 - Pillar

Now that we have a working build that passes all of our tests, we need a URL for employees to visit. To do so, we submit a DNS request to Pillar who will help us out.

1 - Visual Studio Code

First things first, we have to write the code on visual studio. Here is where I spend the bulk of my time writing code and unit tests.



4 - Octopus

Once Leeroy has built, Octopus will deploy the current build to the virtual machines Blizzard has set up. It will additionally deploy my Docker containers.

2 - Ghosthub

Once I'm satisfied with the qualify of my code, I'll look to push my code to Ghosthub via git. This will let me save my code in the case that I lose anything on my work laptop.



5 - Portainer

If anything breaks Portainer lets me check and manage my Docker containers



Don't be a Lost Viking, make sure to map out your way!

Additional Issues I had to Consider

Unit Testing

My code includes unit tests for each of my components and utility methods. Not only do unit tests help me in writing more solid code, it assists me in refactoring to catch any unexpected break in the code. Beyond that, Leeroy will not let me deploy my current build unless all of my unit test pass without issue.

Back-end API

To persist the information, I set up a back-end API. This lets me not only keep the information all in one location, but it also permits writing validation rules and more.

Cross Site Scripting Attacks

Since my code permits entering custom values into input boxes, I had to be wary of cross site scripting attacks (XSS) in which a user can enter JavaScript code or HTML in attempt to steal user data or more. For this consideration, I made sure to sanitize any user input to guarantee that only proper inputs are permitted. This means that the back-end API includes validations rules alongside front-end validation.

Special Thanks!



John Pombo
Senior Software Engineer



Calen Stivers
Sr. Program Manager



Braulio Estrada
Sr. Mgr. Program Management



Madeline Farrell
UX Design Intern