

Background:

This project was requested by the Networking and Telecommunications department of Oregon State University. They wanted an application to aid in the representation of their wireless network's data, in order to expedite the network's service and analysis. The suggested a way to do accomplish this task through a heatmap of the campus, displaying the desired information in an efficient and meaningful way.

The current data has no way of being graphically represented and this is the problem our application wishes to solve. Having the ability to quickly analyze and report different statistics for each building on the OSU campus is what our application strives to achieve.

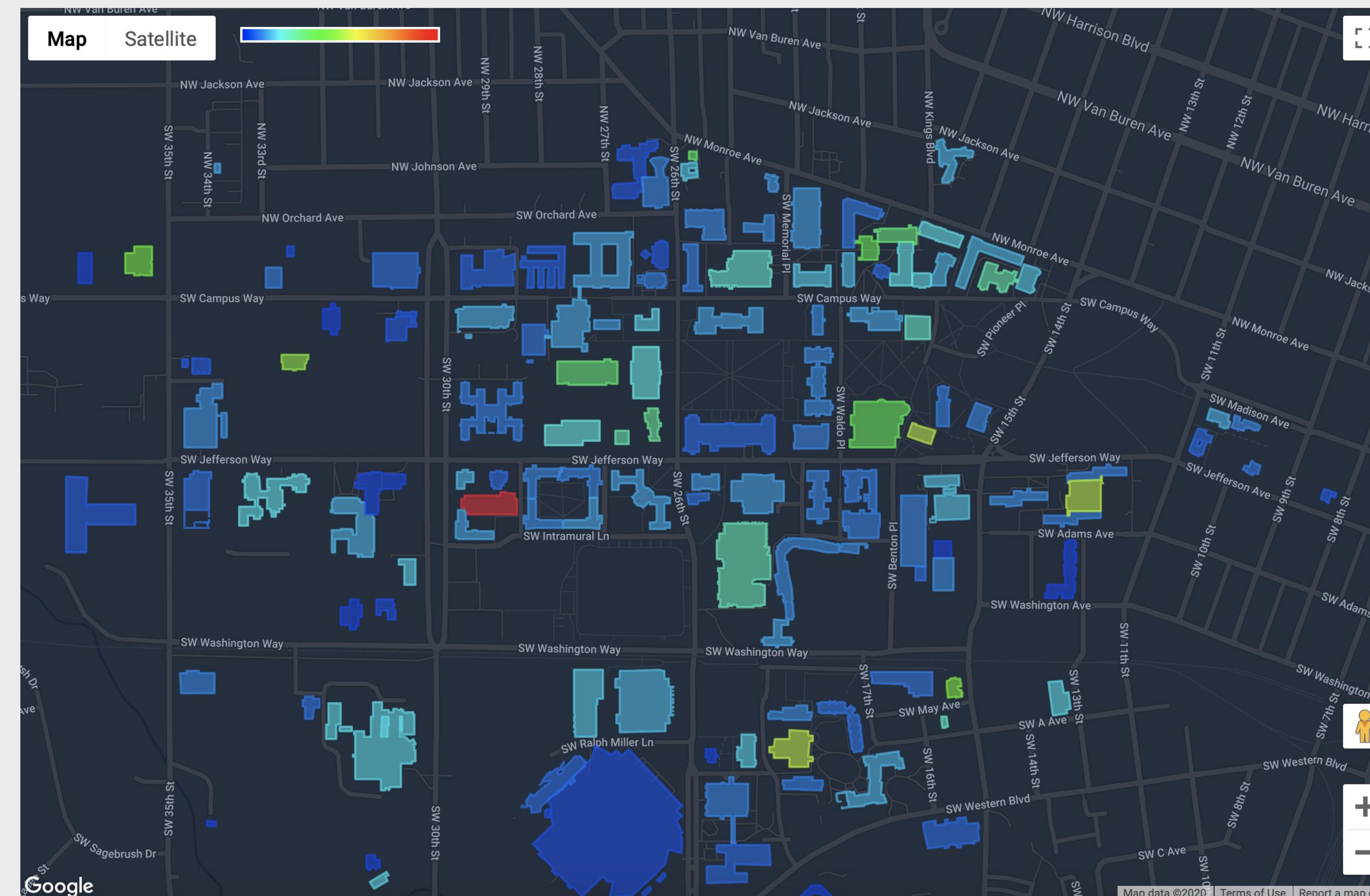
Our application can be utilized in a variety of ways for the visualization of the wireless data. One of the applications functions is to graphically display the usage of the network for each building. This would aid the Networking and Telecommunications team is quickly analyzing an issue with their network's usage.

Another is to display the number of users currently connected to a wireless network's access points. This could be used by students or staff to determine where on campus there is likely a large number of people.



OSU WIRELESS HEATMAP

An application designed to graphically display statistics of the wireless usage on Oregon State University's campus.



Description

Using a combination of tools we were able to create an application that would aid the Networking and Telecommunications department at Oregon State University. These tools included Django, the Google Maps API, the OSU Developers API, the InfluxDB database, and the SQLite database. We got them all working in conjunction to form the foundation of our application. The OSU Developers API returned to us information about each of the OSU buildings such as the coordinates for the building's perimeter. These coordinates are then parsed and implemented to the Google Maps API to become a polygon layer. We then took the newly generated polygon layer and used a function to determine the corresponding "heat" of the building.

Representation of Statistics

- The current statistics being represented on the map are the number of student currently connected to a building's wireless access points. The data is displayed as a ratio between the building maximum occupancy and the current number of people connected to the wireless access point. The ratio is then mapped to a hex color and used to represent the heat signature of a building.
- Due to limitations the data is not perfectly accurate. For example, people may have more than one device connected to the wireless access point skewing the number of total users. Other limitations include how often the Airwave API updates, as well as changing baselines.

Data Handling:

- The bulk of the project revolves around the data that we retrieve from the Aruba Airwave API that is provided by the university. This API provides information on the usage of the wireless network on campus in XML format, which we then parse and store in the time series database InfluxDB. The raw data, which is collected every 15 minutes due to limitations with the Airwave API is stored for 48 hours, and is down sampled every 24 hours to archive the data for the day. This down sampled data is stored for one year, and is viewable via the web interface.
- The other data that we had to work with was the information that we used to translate the wireless usage data into a meaningful heatmap. This data included the polygon representation of the buildings to display which we retrieved from a development API provided by the university, the relationships between rooms, floors, and buildings location wise, which are based on the data organization from Airwave, and the baselines for each individual location, which is based on the number of wireless access points in each location. This information was stored in Django's built in database which created using Django models. This database was created with SQLite3 for development purposes, but can be migrated to a different database using Django's built in migration feature for use in production.