Wifi signal strength HeatMap of Cleveland State University Campus

Developer's Guide

Proposed by

Anubhuti Dayal – 2824826
Suman Kumar – 2861070

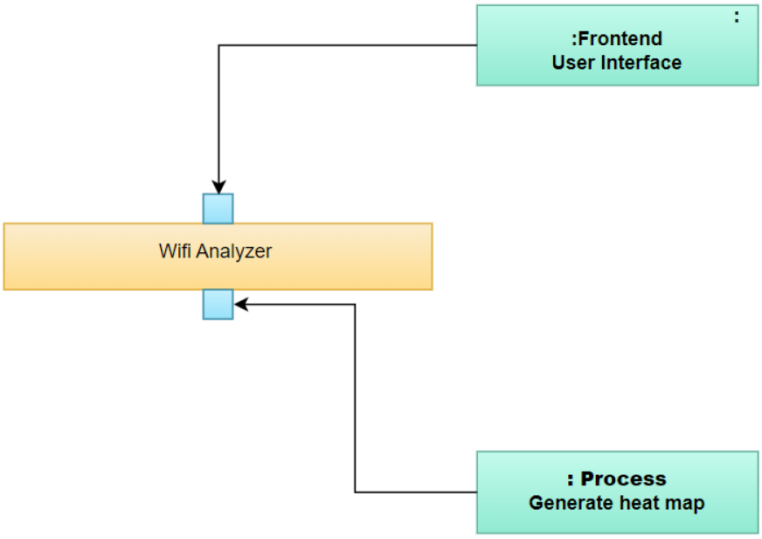
Introduction

This is developers guide for the WIFI Analyzer of Cleveland State University.

We have created “WIFI Analyzer” website, which represent a Wi-Fi access points (APs) coverage heat map in the city of Cleveland at the periphery of the Cleveland State University. With the help of the heat map, we can easily identify the specific position with poor signal reception and determine effective router positions.

We have showed the results for the college campus; highlight the weakest and strongest coverage area through a heat map. Data collection for input was done via measuring WIFI signal strength around the college building including basement, where we have car parking. The smart phone uses application to measure Wi-Fi signal strength at a specific area.

There is user interfaces for the frontend and for generating heatmap backend logical process

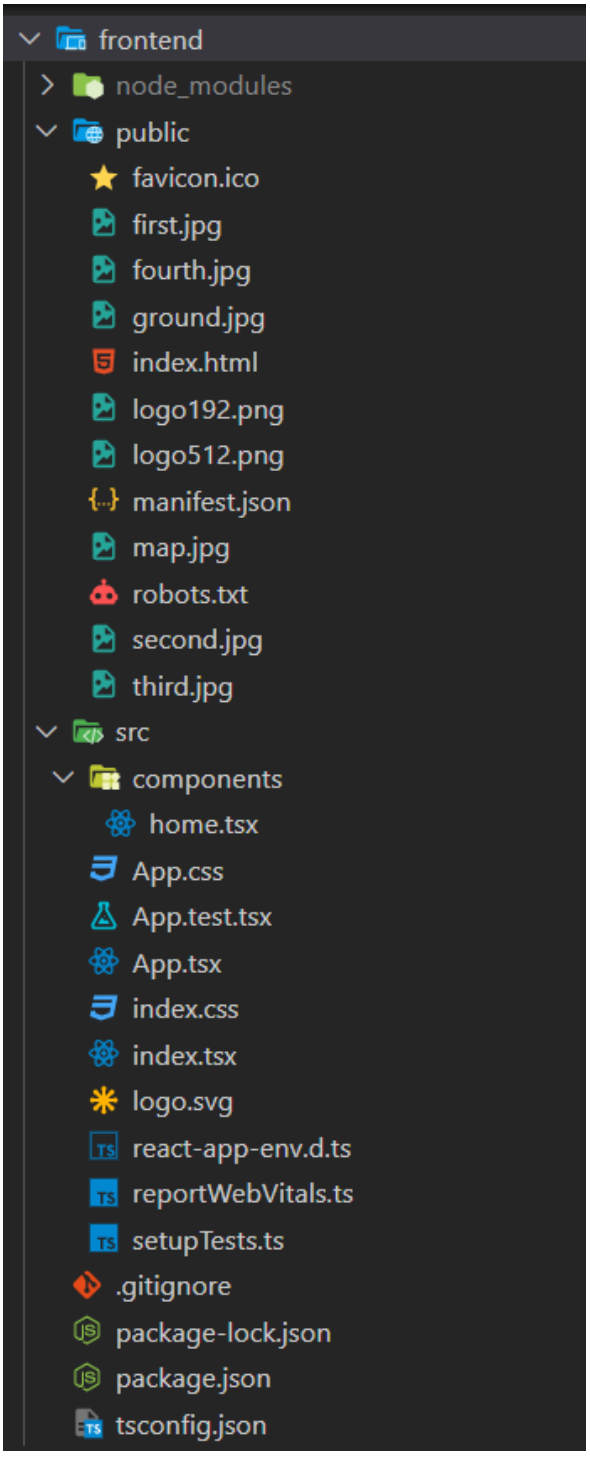


Technology, software, and hardware used

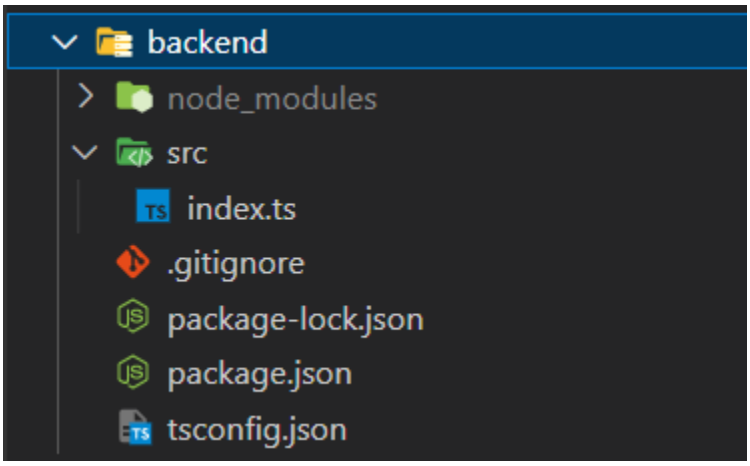
- **React with Typescript** for the frontend UI designing
- **Node.js with Express in Typescript** for the API development
- **Matlab** to generate the heat map and verify the data quickly
- **Airport WiFi scanner** to measure the WIFI signal strength
- **Excel** to capture the generated data

The Project Structure

The entire project has been kept in the Github for version controlling. The project contains Frontend (React), Backend (Node), Matlab code and additionally documentations that has been prepared in every stage of development as to follow the best practices of Software Engineering.



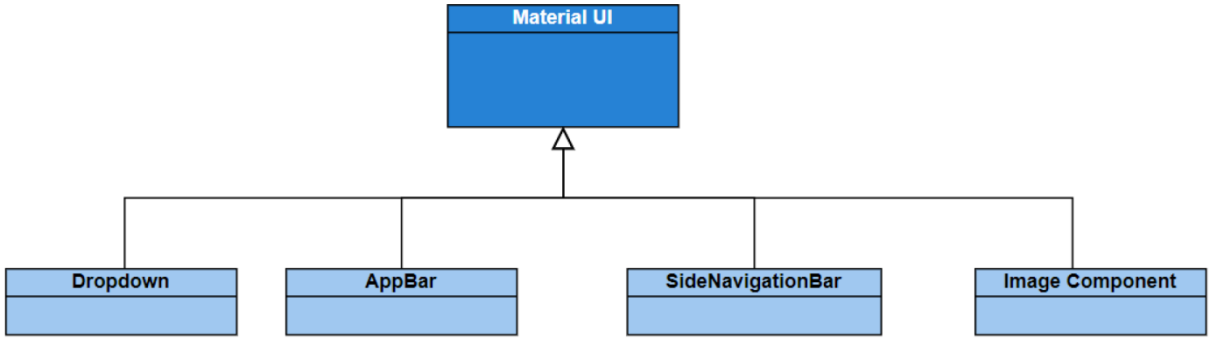
This is the Frontend project structure of the application; this follows the Create-React-App react boilerplate project architecture with typescript.



This is the backend structure for the project which uses Node.js and Express for the API development. This also uses Typescript to enable the type safety in the project.

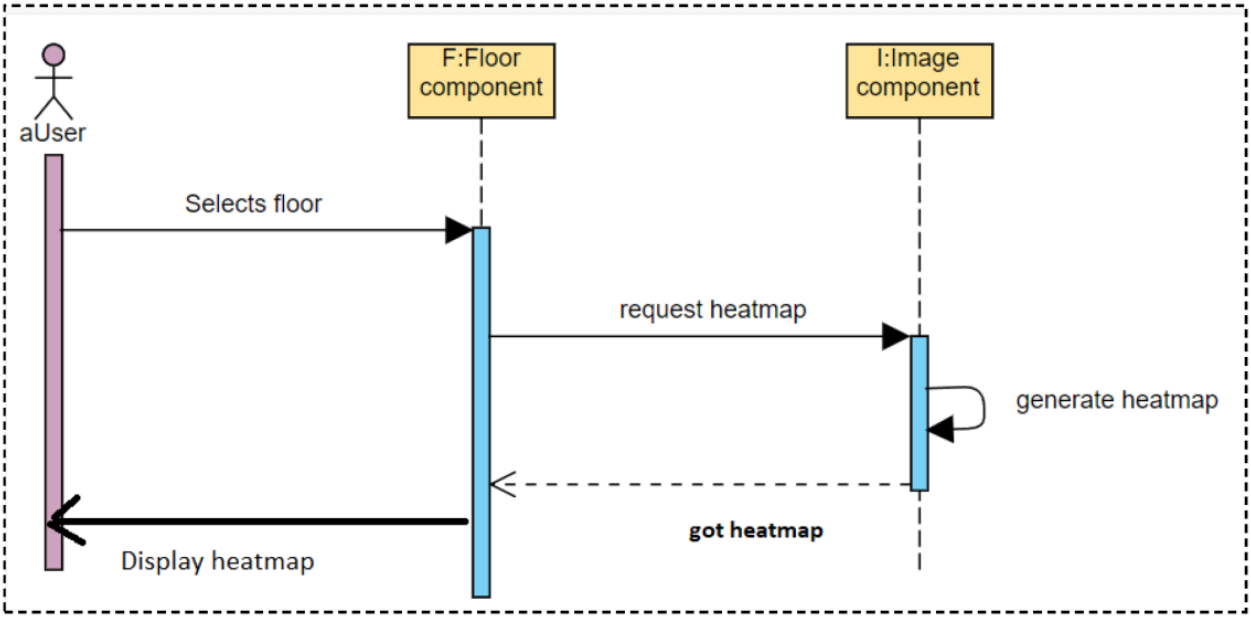
Components design

We are using Material UI for the component designing and we have segregated our design view in a various functional views. As the application has been developed using React, this follows Functional Component conventions for the front-end. Below are the parts that are involved as reusable functional components which are inherited from Material UI:
AppBar, SideNavigationBar or Drawer, Dropdown, Image Component



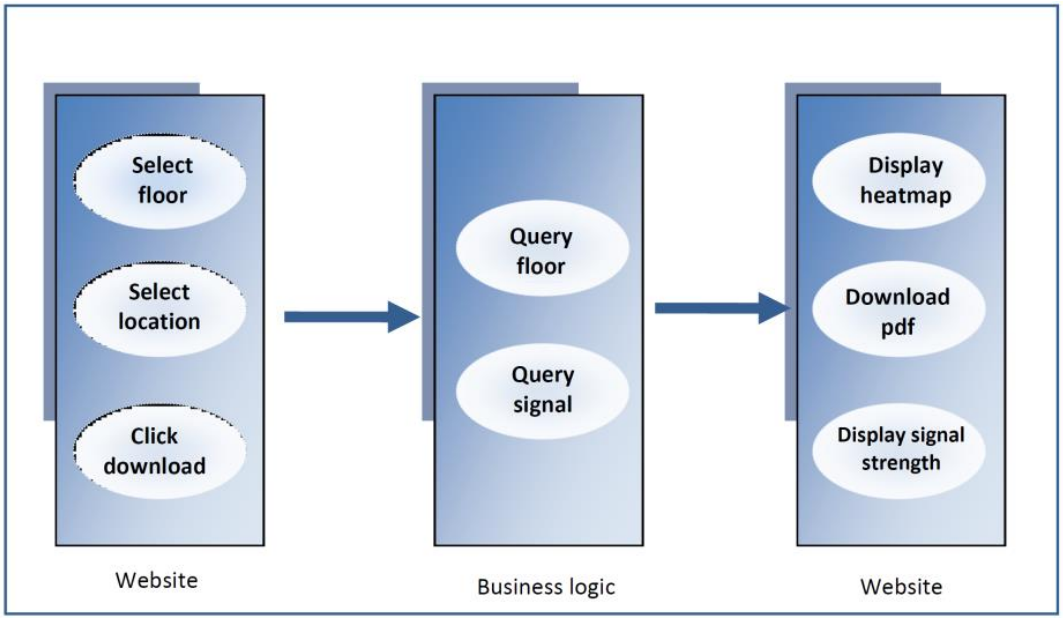
Apart from this every Floor components are inherited from the same class where the values get changed based on the Floor type

Below is the sequential model for Wi-Fi Analyzer for one scenario, where user is requesting for heat map for one of the floor –



High-level Architecture

- **Input**: Input section will be web portal, where a user can access the website link and user will have option to either select the floor, or select the location or else user can click download button.
- **Process**: Process section is only for internal processing of the user requests. , if user has selected the floor or location, system will generate heatmap for the specific floor or system will get the WIFI signal strength of the specific location and send the result to the portal.
- **Output**: Once again this time, frontend web port6al will serve as output as it will display the results to the user. If user has requested for the heatmap, it will show heatmap. Else if user has selected the current location, then it will display the real time WIFI signal strength of the heatmap. Or else if user has requested for the download pdf , it will generate pdf and download on the user's machine.


Select floor
Select location
Click download
Website
Query floor
Query signal
Business logic
Display heatmap
Download pdf
Display signal strength
Website

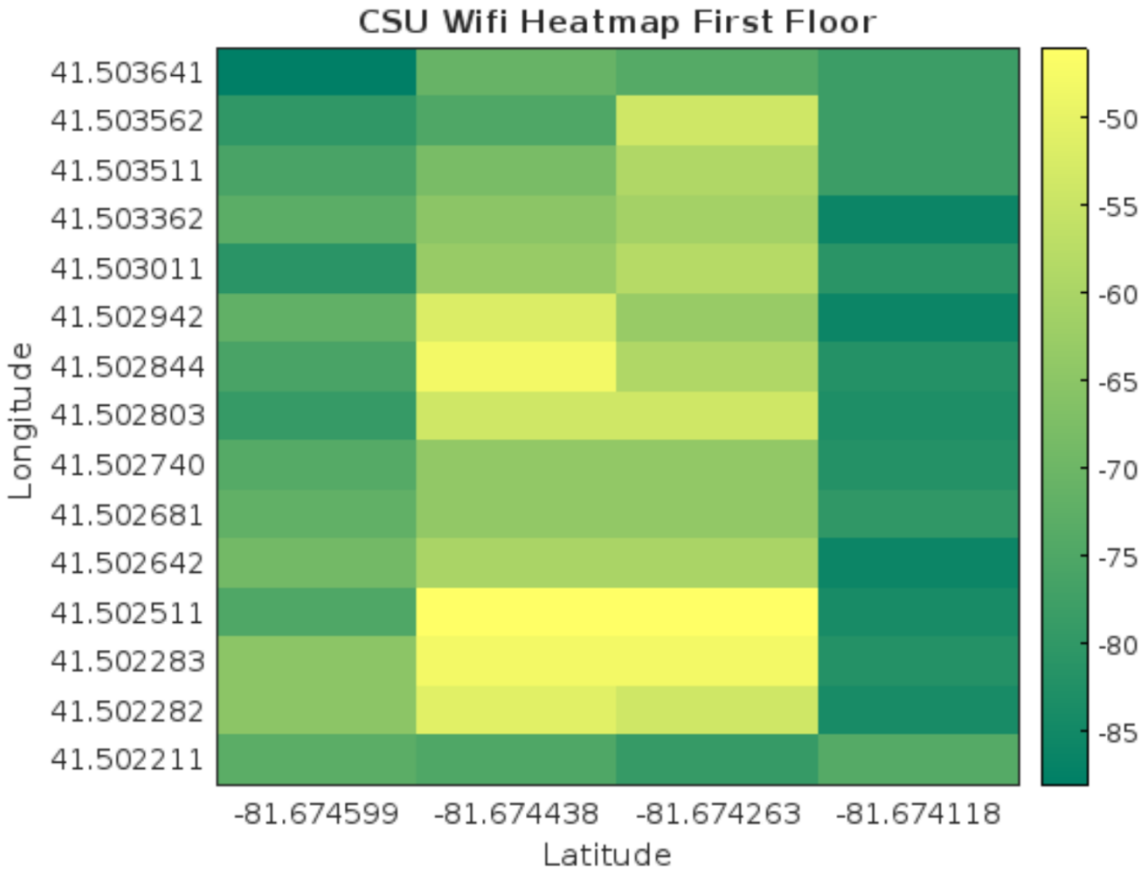
MATLAB Code

Heatmaps are a way to visualize data using color. Heatmap function in matlab is used to generate WIF Heatmap. Heatmap Chart properties control the appearance and behavior of a Heatmap Chart object. By changing property values, you can modify certain aspects of the heatmap chart.

In below code cdata is wifi signal strength : received signal power, which is measured in decibels, or dBm on a logarithmic scale , Latitude valuse observed from GPS , Longitude valuse observed from GPS. To plot the heat map used MATLAB function heatmap.

```
%CSU WI-FI Heatmap - First Floor
% Written by Anubhuti Dayal
% Project - Wifi Heat Map

%Close figures and clear everything before starting
close All;
clc;

%Initialize variables

%cdata is wifi signal strength : received signal power, which is measured in
 decibels, or dBm on a logarithmic scale
%closer to 0 dBm, the better the signal is i.e. -90 dBm Unusable while -67 dBm
 Very Good
cdata = [-88 -71 -74 -78 ; -80 -75 -54 -78 ; -76 -68 -59 -78 ; -73 -65 -61
 -86 ; -81 -63 -58 -81 ; -72 -52 -63 -86 ; -76 -48 -59 -82 ; -79 -54 -54 -83 ;
 -74 -64 -64 -82 ; -72 -64 -64 -80 ; -69 -60 -60 -86 ; -75 -46 -46 -84 ; -65
 -48 -48 -82 ; -65 -51 -54 -84 ; -73 -75 -79 -74 ];
%Latitude valuse observed from GPS
xvalues = {'-81.674599', '-81.674438', '-81.674263', '-81.674118'};
%Longitude valuse observed from GPS
yvalues =
 {'41.503641', '41.503562', '41.503511', '41.503362', '41.503011', '41.502942', '41.502844


%Plotting heatmap
h = heatmap(xvalues,yvalues,cdata, 'Colormap',summer);

%Setting up Heatmap variables
h.Title = 'CSU Wifi Heatmap First Floor';
h.XLabel = 'Latitude  ';
h.YLabel = 'Longitude ';
h.GridVisible = 'off';
h.CellLabelColor="none";
```

Output


CSU Wifi Heatmap First Floor
41.503641
41.503562
41.503511
41.503362
41.503011
41.502942
41.502844
41.502803
41.502740
41.502681
41.502642
41.502511
41.502283
41.502282
41.502211
Longitude
-81.674599
-81.674438
-81.674263
-81.674118
Latitude
-50
-55
-60
-65
-70
-75
-80
-85

Guideline to run the application

In the project directory, you can run:

npm start
Runs the app in the development mode.
Open http://localhost:3000 to view it in your browser.

The page will reload when you make changes.
You may also see any lint errors in the console.

npm test
Launches the test runner in the interactive watch mode.
See the section about running tests for more information.
https://facebook.github.io/create-react-app/docs/running-tests

npm run build
Builds the app for production to the build folder.
It correctly bundles React in production mode and optimizes the build for the best performance.

The build is minified, and the filenames include the hashes.
Your app is ready to be deployed!

See the section about deployment for more information.
https://facebook.github.io/create-react-app/docs/running-tests

npm run eject
Note: this is a one-way operation. Once you eject, you can't go back!

If you aren't satisfied with the build tool and configuration choices, you can eject at any time. This command will remove the single build dependency from your project.

Instead, it will copy all the configuration files and the transitive dependencies (webpack, Babel, ESLint, etc) right into your project so you have full control over them. All the commands except eject will still work, but they will point to the copied scripts so you can tweak them. At this point you're on your own.

You don't have to ever use eject. The curated feature set is suitable for small and middle deployments, and you shouldn't feel obligated to use this feature. However, we understand that this tool wouldn't be useful if you couldn't customize it when you are ready for it.