Milestone 4

Group 111-4 | 11/7/19

Revised List of Features

1. Current Sound(db) Reading at All Locations

This feature of our software will display each location where sound(db) data is being collected and the corresponding noise level of each respective location represented by a status tag (“busy”, “moderate”, “calm”).

2. Average Sound(db) Reading at Specific Location

This feature of our software will promptly ask the user to choose a location from our interface and we will display the average sound(db) of that location.

3. Current Best Location to Study

This feature of our software will determine the best (quietest) location for a user to study at the current time and display this location.

4. Quietest Time at Location

This feature of our software displays the user specific times during the open hours of a location on campus that we have tested sound(db) measures thoroughly and that the sound(db) tested is at its lowest for the day. The user will promptly pick the location from our interface and then will be shown a time interval in graph format.

5. Loudest Time at Location

This feature of our software displays the user specific times during the open hours of a location on campus that we have tested sound(db) measures thoroughly and that the sound(db) tested is at its highest for the day. The user will promptly pick the location from our interface and then will be shown a time interval in graph format.

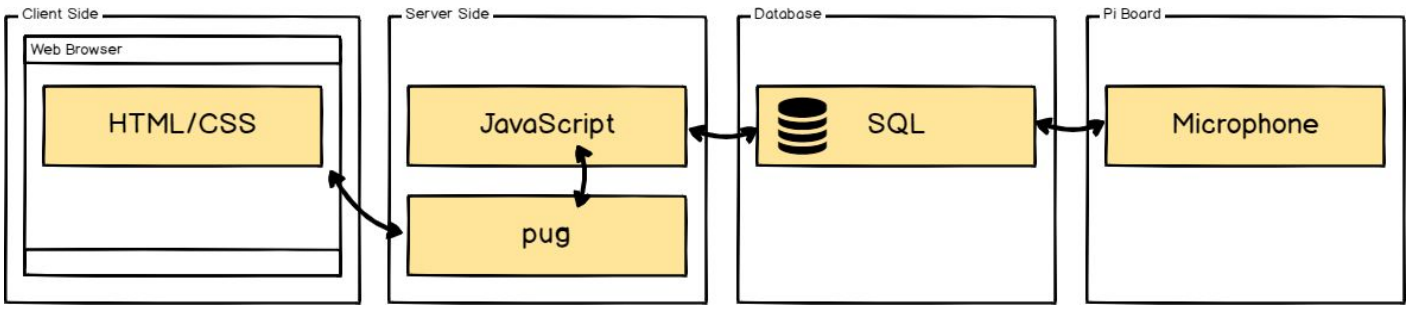
6. Display Data for Specific Time and Location

This feature of our software promptly asks the user for a location and a specific time interval that they want to see displayed. We will show the user in graph format the sound(db) measures of that location during the given time interval.

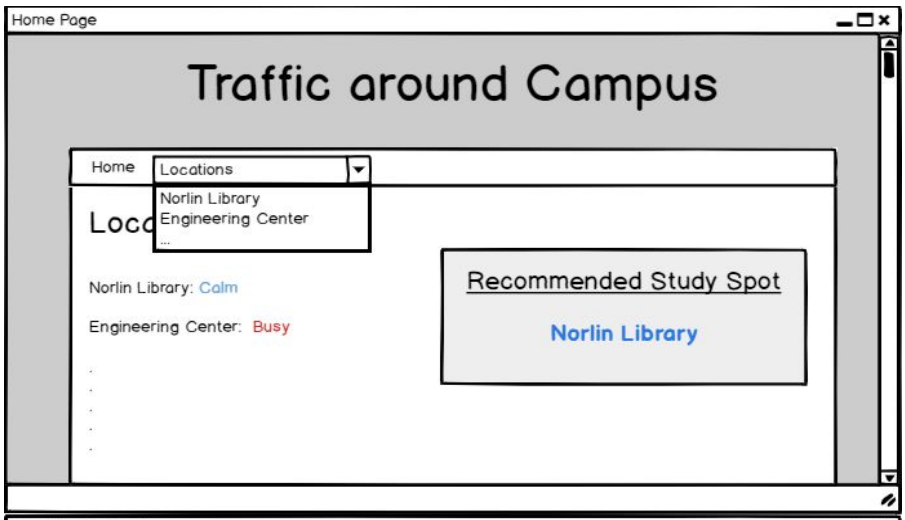
7. Display all Data at Specific Location

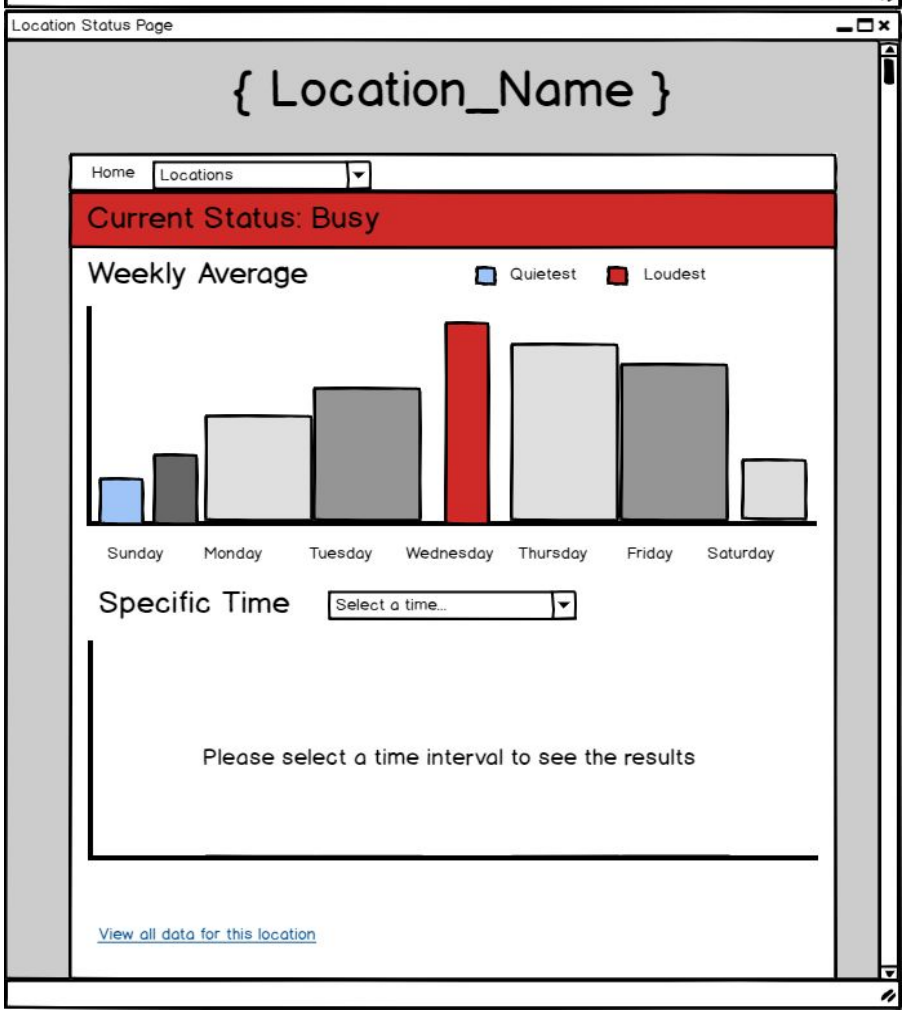
This feature of our software will have the user pick a specific location in our interface and then we will display all of the given sound(db) data in graph format to the user.

Architecture Diagram



Front End Design





Web Service Design

Our project will not be using any web services.

Database Design

- The microphone will be connected to a RPI via a standard GPIO. GPIO stands for general purpose input/output. Meaning that this is a standard analog pin that is reading in an ADC voltage to the RPI. The RPI will be doing some standard math conversions- converting the ADC voltage to a digital number based off the size of our ADC. In most cases this means that our ADC will have 10 bits of resolution. Which in turn means that we can distinguish 2^10 different discrete levels of sound. Of course dB's are not linear- they are logarithmic so there will be some calculations that will be made on board of the RPI. Usually this consists of using a known calibrator(most likely our phone). We will take around 10 data points to create a function that closely matches the way our microphone works. Of course this will lead to some error in our calculation, but it will be negligible for this project.
- The RPI will be used as a transfer server- meaning that the RPI will transfer the data into an SQL database and likewise create meaningful information from our data.
- After this stage javascript will be used in conjunction with PUG and HTML to create a website to show our cool statistics.
- Data storage:
    - Stage 1: Raw ADC values converted to dB's will be stored via a text document on the RPI
    - Stage 2: The data will be parsed in order to be used by SQL and stored in an SQL query. Similarly this will just have a timestamp with a dB value attached to it
    - Javascript and pug will interpret this data in a meaningful way
    - Lastly HTML will display our data in conjunction with the previous stage.