***Vision***

This project is basically trying to mimic a real world traffic analyzing app.

***High Level Description of What is traffic analyzing***

Traffic can be people / cars and many other things. In other words, anything moving from one point to another point (Maybe from a source to a destination). So a traffic analyzer is basically recording these traffic when they pass a certain point during their movement. That certain point can be a traffic light in case of analyzing vehicle traffic or some corridor in case of analyzing people movement through that corridor. After we record this traffic, we can make sense of this traffic by processing the recorded data using some data analytics tools and then visualize it in a human readable way like bar charts to find useful information like “Time of the day which has the most traffic”.

***Outline of the System***

******

**Note: The flowchart gives a high level view of how the system function.**

***Tools used to make this project***

*For the client side interface*

* **HTML** – this is used to describe the structure of the web page that the client will see when they visit the website.
* **CSS** – this is used to give styles to the different elements of the webpage so that the website looks nicer to look at and also makes it more usable in terms of the appearance and positioning of the different elements on the webpage.
* **JavaScript** – this is used to dynamically update the elements on a web page to provide more user interaction on the website like making something happen when user clicks a button. Also it adds a dynamic feature to all elements on the webpage so that they can change at the proper time as needed.
* **jQuery** – this is used for easy manipulation of document object elements in the browser. It is built on top of JavaScript.

*For the server side*

* **Express** – used as the server for the website. Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It is used to provide small, robust tooling for HTTP servers, making it a great solution for single page applications, web sites, hybrids, or public HTTP APIs.

*For the database*

* **Google Firebase** – it is used to store all motion sensor data. Firebase gives us the functionality like analytics, databases, messaging and crash reporting so we can move quickly and focus on our users. Firebase is built on Google infrastructure and scales automatically, for even the largest apps.

*For communication between client and Server*

* **Socket.io** – this is used because it provides us with a bi-directional communication channel. So for example, if the client wants to close the LED, it can easily tell the server to do the work by emitting an event. Same goes for the server, if the server wants to inform the client to update the message count, it can send an event as well.

*Libraries Used*

* **Bonescript** – The BoneScript library provides several functions useful for interacting with the beaglebone hardware. So functions it provides are we can see if the LED is on or off or any other device attached to the beaglebone can be checked to see if it is on or off.

Finally, **Node.js** is used which is basically JavaScript code running on the server side.

***Features of the system***

1. Get total number of short messages detected.
2. Get total number of Long messages detected.
3. Get total number of visitors detected.
4. Delete button to delete all records from the database.
5. LED on/off button.
6. Motion sensor on/off button.

***Additional Features I have Added.***

1. Get total number of consecutive visitors.
2. Display all record from the database on the client side in a readable manner.

***Constraints***

1. The motion sensor is inaccurate. Sometimes it is able to detect motion but sometimes it just does not detect any motion.
2. When there will be two consecutive visitors, the on time for the LED needs to be extended by 5 more seconds. So for the first visitor the LED is on for 20 seconds and then it becomes off. Then again the LED is switched on for further 5 seconds. So for the LED to switch back from off to on takes some milliseconds. To the human eye it may seem that the LED is actually on for 25 seconds but in reality it wasn’t the case. So this is because the electronic components on the board have a gate delay to turn something from on to off or vice versa.

***Screen Mockup***

**SENSOR ON/OFF**

**DELETE DATABASE DATA**

**LED ON/OFF**

TOTAL NUMBER OF VISITORS

10

TOTAL NUMBER OF SHORT MESSAGE

14

10

TOTAL NUMBER OF LONG MESSAGE

**TRAFFIC ANALYZER**