**Project Scope**

Table of Contents

[Introduction 1](#_Toc510348153)

[Tools Used 1](#_Toc510348154)

[High Level Description: overall system functionality 2](#_Toc510348155)

[High Level Description: Reset Button 3](#_Toc510348156)

[High Level Description: Toggle ON / OFF LED and PIR buttons 4](#_Toc510348157)

# Introduction

The scope of this project is to design a Traffic Analyzer app that can establish the number of pedestrians walking across a door or a corridor. To detect the number of pedestrians crossing a certain corridor or door, a series of motion made by the pedestrian must be recorded. This includes a long, short, long and long motions in series. The device only detects and counts a pedestrian crossed if the previously mentioned series of motions are followed, otherwise it would just restart the motion series. Whenever the sensor is enabled, the calculation of the short motions begins immediately, while calculation of the long motions is delayed for 2 seconds, meaning it will start 2 minutes after the PIR motion sensor has been powered on. Both long and short motions have a delay of 4 seconds between each calculation. Thus, eventually the time lag between the long and short motion will be 2 seconds.

# Tools Used

The tools used to develop this system were Beaglebone Black (BBB), a PIR motion sensor and a LED light in the hardware side. The BBB allowed us to establish the communication between the PIR and LED, and as well as perform various algorithms, such as identifying the type of the motion.

Whereas, in the software side, the tools such as socket.io, firebase database, JavaScript, jQuery and HTML were used. Socket.io has been used to establish the connection between the server and the user (client) sides, which allows the system to send the data from the server to client and vice versa. The database from Firebase has being utilized to store the motions recorded (long and short), and as well as the number of pedestrians crossed. JavaScript has been used to read the main code, and execute the commands, while the jQuery allowed us to identify the clicks of the buttons on the HTML website.

# High Level Description: Overall system functionality

# C:\Users\User\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Send%2FRecieve.png

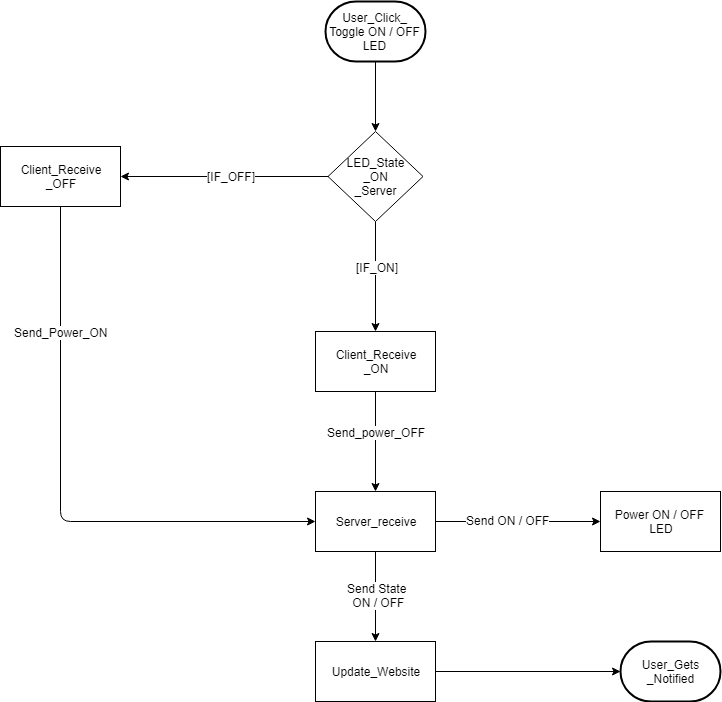
This first diagram will provide a description of how the system checks the motion and retrieves the data from the database. Upon accessing the website, the HTML page retrieves the stored data of the values for long and short motions, and as well as the number of pedestrians crossed from the server, which retrieves the data from the firebase\_databse. The default data that are stored on the firebase\_database are the values of 0 for every counter (long, short, and pedestrian); however, after the steps described below, the data gets updated, and the server is going to retrieve new values, for example 2 short motions, 6 long motions, and 2 pedestrians crossed.

Initially when user powers on the PIR motion sensor in the webpage, the sensor starts checking for the motion series being made. Every short and long motion will be added to the overall counters (count\_long, and count\_short) and will be pushed to the Firebase\_Database. If decider identifies the series of motions, such as long, short, long and long, then the server will add to the overall pedestrian counter (count\_pedestrian), and will push the data to the Firebase\_database. Each counter has its own channel on the database side, which are called /Pedestrian, /Long, and /Short. After the new data has been pushed to the server, the data retrieval process that has been mentioned earlier gets repeated. Server retrieves the data from the firebase\_database and sends it to the webpage from where user can see how many long and short motions have been counted, and how many pedestrians have crossed. This data can be reset, which will be explained below.

# C:\Users\User\AppData\Local\Microsoft\Windows\INetCache\Content.Word\reset.pngHigh Level Description: Reset Button

This explains how a reset works in a firebase, so when the user clicks on the reset the message is captured by the server, gets displayed on the website, and as well as sent to the firebase database. The firebase database then reevaluates and makes the readings of pedestrian, long and short to be 0, and sends it back to the serve, which pushes the data to the webpage.

# High Level Description: Toggle ON / OFF LED and PIR buttons



This explains what happens in the event when a user clicks on the LED to be turned off or on, when the user clicks on the LED to be turned ON / OFF, then client receives ON / OFF and sends ON / OFF to the server, the server then turns ON / OFF the LED and sends it to the website for the user to see.

The same system flow works for the PIR motion sensor turn ON / OFF as well, but in the event of a sensor being turned OFF, the entire execution of the code shuts down (because there is no way to cut the power of the pin on Beaglebone Black), while in the case of a LED, only the LED turns off while the code continues to be executed.