**Project scope statement**

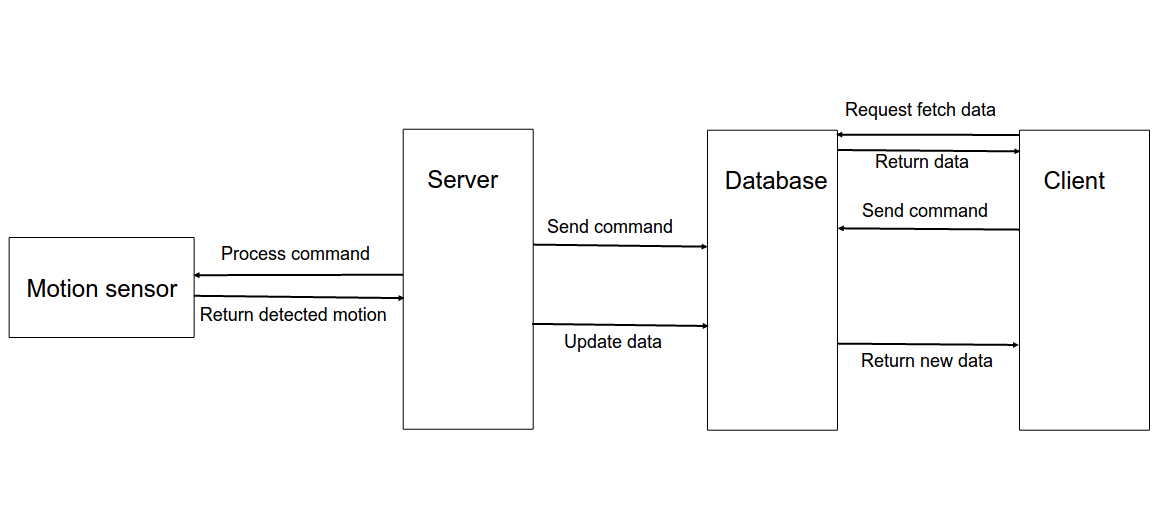
Project vision

This project is a intruder counter program, it detect motions through a motion sensor, store motions in a database and reports intruder detected when a particular motion sequence occurred.

Project requirements

1. Server can listen and read motion sensor’s data.
2. Server can send database new data.
3. Server can listen to commands from client.
4. Database can store all motions detected by motion sensor until it gets reseted.
5. Client can fetch data from database if exists.
6. Client can update displays when new data detected by server.
7. Client displays numbers of long and short motion and number of intruders.
8. Client can send LED on/off, motion sensor on/off and reset data commands to the server.

Working system diagram



Project operation outline

**Database**

The database used in this project is Firebase, formally known as Google Cloud Messaging, it is used to development high-quality applications. The database is in charge of storing data about motions detected by motion sensor and the commands came from the client. It has two channels, one is for the commands sent from the client, in this channel, there are variables for all available commands, server will always listen to this channel to make responds to new commands. The other channel is for motions detected, this channel stores variables that keeps count of different types of motions, also an array for calculating intruders.

**Server**

The server is a node.js file, node.js is a platform used to build application using javascript. The server communicates with the motion sensor, LED and the database, as stated in database section, server will calculate motion duration whenever the sensor detects one. The server communicates with the motions sensor and the LED using “johnny-five”, which is used to interact with the Arduino board, then the server not only classifies long/short motions, it will also check the array stored in the database, whenever there is a long-short-long-long sequence of motions, server will define it as an intruder and update both the array and the intruder count. The way server controls the LED and motion sensor is also through the database, however, all the commands in the database are sent from the client, so it is listening to the client but through the database and turning LED on/off or turning the motion sensor on/off. It only updates database counts when the motion sensor is being turned on.

**Client**

The client is a html file, HTML is a language used to build the basis of a web page, there are also javascript used in this file to communicate with the database just like the server. The client is always listening to the motions channel to update its display information. Client also updates the corresponding attribute in the commands channel when a button on the website is clicked, this is achieved by using jQuery, which is a library used in html file for event handling. One special function is ‘reset’, when this button is clicked, client updates the database and server will reset the database, then client can reset all counts, this makes it more secure.

Project attributes/constraints

The project can perform as desired most of the time, but there are some constraints that limits the performance of the project:

1. The motion sensor has a delay of 5 seconds, so when defining long/short motions, the middle point has to be longer than 5 seconds.

2. The database might slow down when there are too many data stored, which might also slow down the overall performance of the project.