**Spike Outcomes**

**Name:** Serverless

**Goals:**

A serverless implementation of the following:

* A node.js based program that reads the data from the Arduino using a motion detector and pushes it to the firebase. The data that the program pushes should be in the json format with 3 attributes:
  + Timestamp
  + Motion start time
  + Motion end time
* A program that does the following:
  + Listens to firebase
  + For every new object:
    - Send email to pre-defined address if motion is long
    - Send email to pre-defined address that shows how many long and short motions have been detected

**Technologies, Tools, and Resources used:**

* Nodemailer, to allow use of an email to send emails from the node app
* Firebase, to host our data and allow updating of values
* Firebase functions, to allow back end code to automatically run on updates
* Motion sensor, to allow updating of time and to be listened to

**Tasks undertaken**:

The first component would listen to the motion sensor, and update the start, end and timestamp values, as well as determine whether these were long or short, and update them. This would trigger the second component, that would activate the firebase function to send an email automatically

**What we found out:**   
The first issue we ran into was deploying the firebase. After some research, it was discovered simply using “firebase deploy” would not be enough, so we needed to select our specific project to deploy. To do this, we used “firebase deploy –project project\_name”. We also found the only way to stop the automated listening of the firebase function was to comment it out and re-deploy it.

**Open issues/risks:**

There is no way to time off the listener in the serverless function as it is all back end, the only option would be to deploy the function each time you want it on/ off.

**Recommendations:**

The only recommendation to be given would be that to include an on/ off ability of the serverless function to get rid of that issue.