**Lumi Progress Report**

**Week 5**

Richard Dip

[richarddip@hotmail.com](mailto:richarddip@hotmail.com) | N00653804

Raymond Dang

[Raymonddang96@gmail.com](mailto:Raymonddang96@gmail.com) | N010

RJ Cootauco

[rjcootauco@gmail.com](mailto:rjcootauco@gmail.com) | N01114847

**Lumi Physical Format**

Emailed Austin about this cuz I have no idea what this is about

//more stuff here

**Progress In-Depth**

**Mobile Application**

In the past two weeks, Richard has been able to condense the mobile app to functionalities specifically to compliment the hardware. The UI has also been modified to mirror the changes in the exclusion of unnecessary functionalities. Lumi’s mobile app is now able to enable/disable lights from the NeoPixel ring, enable/disable sound activation, light pattern options for the NeoPixel including rainbow, strobe, and solid RGB. These functionalities have been tested up to the mobile app and further integration with work by Ruel will continue the data input testing to the hardware.

Currently the progress on the application primarily involves sending and fetching input data to the Firebase Database as a means of testing for future full implementation.

**Hardware Python Coding**

The python code that runs on the Raspberry Pi is to control the activity of the sensors that are being used. The code is used to change the RGB values on the NeoPixel. The Lumi also implements the sound sensor that will trigger the NeoPixel ring and light up according to the RGB values set. The way that the hardware works with the python code created is that the program takes both of the sensors together and has an output of the NeoPixel ring lighting up due to the sound sensor picking up the baby crying.

The current progress with the hardware is just having the ability to have all of the components of the Lumi such as the raspberry pi, neopixel ring sensor and the sound sensor to be incorporated into one shell.

**Google Firebase Database**

The real-time database will be evaluated through three separate python codes. These actions were achieved through installing and implementing python-firebase 1.2 which allowed Ruel John access to a library of firebase methods. The python code provided will record the RGB color coding, a timestamp, and/or a brightness value. The progress completed by Ruel John as of week 6 concludes to developing a working python code that posts a timestamp every time sound is detected. As for the RGB color values, the database will automatically record the value of the light when idle. When triggering the sound test, the RGB value will temporarily change while the sensor is still active, meaning the value on the database will shortly update until it is back in idle form. As for the light (photocell sensor), there has been no progress done yet.

**Project Obstacles & Difficulties**

**Lumi 3D Shell**

The softworks shell provided by our collaborator David Neumann has been found to be too small to house our Raspberry Pi 3. The current progress on this aspect of the hardware is to enlarge the shell in a 3D software designing platform. We expect it to be 200-250% larger than the original. The difficulty lies with learning the platform in order to successfully complete this task with minimal to no design flaws during 3D printing.

**Lumi Python Code**

**NeoPixel Ring RGB -** The python code developed requires more thorough coding as it roughly performs what it is supposed to be doing. During the time when the sound sensor is activated, the RGB lighting is to light up all 12 LED as red at the same. In this case, the light only triggers one LED every half a second which doesn’t give enough time to fully display a whole red LED. As for the photocell sensor, there is still some work that needs to be done with detecting a value of the room’s brightness level. Once accomplished, that value can be stored into the database.

**Sound Sensor** – The python code developed for the sound sensor triggers sounds very clearly. However, during the test run for detecting sounds, it would trigger the if statement more than once, creating more instances of data records than expected. In this case, Ruel has set the firebase to PATCH records only, meaning it will record an instance of data and will remain the same until the python code is being run again.

**Project Financials**

No real changes has been made to the hardware that required deviations from the original financial plans. The modifications made to the 3D Lumi shell has fortunately been provided for free through Humber College.