A FINAL REPORT

Department of Electrical and Computer Engineering
Lehigh University
Bethlehem, PA 18015

PROJECT TITLE:

SOUND SENSING SYSTEM WITH POWER OVER ETHERNET TECHNOLOGY

PRINCIPAL INVESTIGATORS:
Samantha Shreck & Zoe Protin

ADVISOR:
Professor Brian Davison

May 2015
Revised 2015
Updated May 2015

Advisor Signature

Date

Abstract

Our team aims to create an efficient and accurate sound monitor system which may be installed in several different rooms. The system functions by taking a reading of a room's noise level with a sound sensor energized by Power over Ethernet technology. The noise level readings from the sound sensor device will be transmitted to a Central Networking Server which will be stored and utilized by the web application. Based on data analysis done by the web application, the user can see for themselves what room is noisy or not without lifting a foot.

Acknowledgements

During this year long project several people have provided many forms of help and support. Firstly we would like to thank Professor Brian Davison for advising us on this project and for continuous guidance throughout the project. Secondly, we would like to thank Ted Bowen and his resources who helped us develop the skills necessary to tackle this project. Without him it is likely we would not have finished this project or we would be facing tremendous struggles. Another thanks to Professor Liang Cheng and Professor William Haller for all the words of wisdom and advice given.

Finally, we would like to thank any faculty members or students at Lehigh University who have provided ideas, been cooperative and made this project happen. There have been no occasions where a conflict of opinion has not been resolved successfully.

Table of Contents

1. Introduction

This report provides a comprehensive documentation of the development of POE sound sensor paired with a web applications for students. Our project identifies students' need for an effective and easy way to use web application by capturing noise levels from the classrooms. Samantha Shreck and Zoe Protin undertook the project under the supervision of Professor Brian Davison from the Computer Science Department at Lehigh University. The development of sound sensors with POE capabilities paired with a web application opened up new ways to utilize POE technology. By using our application, students can easily gain knowledge about study spaces in an interactive and user-friendly way.

This report is organized in the following manner. The Problem Statement introduces the design problem and delineates the requirements and design parameters of the POE sound sensor. The Problem Statement section explains the basic operation of the application, and its development and running environments. The sections of the two different parts of the project illustrate key procedures of each software subsystem of our hardware and our application: POE, decibel conversion, data storage, analysis, user interface (UI). These sections also described engineering challenges in our design, such as the design alternatives we faced as well as the trial and error marathon of coding. The Testing section within each of those then analyzes the results of the accuracy, timing, and functionality tests of our hardware and application's operations. The Project Management section outlines the overall scheduling strategy and the financial cost of our design development. We also describe safety and environmental issues regarding our hardware and our web application design in the following section. Finally, our report will conclude with recommendations for future development of the sound sensor and uses of the web application.

2. Problem Statement

Study time is fundamental to the student experience, campuses need to give students ample appropriate environments. For many, that means quiet environments where they can put their heads down and focus.

Studying in itself is a valuable habit. It builds self-motivation and discipline, and of course is necessary to grasp course material. A recent study completed at University of California found that while an average student studied 24 hours a week in 1961, today that number has dropped to 14 hours a week.[1] The questions that we asked are: Is college too difficult? Could this be related to not finding a quiet place to study? The ultimate question we asked ourselves is how does this correlate with our project?

[1] "Leisure College, USA." *AEI*. N.p., n.d. Web. 02 May 2016.

3. Motivation

College is a time of intellectual growth and development as young adults. At college, study time is fundamental to the student experience and that intellectual growth. Campuses need to give students ample appropriate environments. This means quiet environments where they can put their heads down and focus. On campuses, Students can sometimes view lounges and libraries as spaces for heads-down focus work.

In 2012 Gensler study, "forty-three percent of students told us the library is where they prefer to study/work alone, and 26% reported lab/project/studio space was their favorite place for that heads-down time. But when asked to report where they had actually studied/ worked alone, the numbers flip. Only 22% report studying in the library and 38% in lab/project/studio space. Students want to study at the library, but they are more likely to head elsewhere or simply not find the space they need." [2] This is because of the noise. Those students reported back that the noise levels were too distracting. They also noted that the number of quiet study spaces was limited.

In a study identifying Pontíficia Universidade students' perceptions about noise in the classroom, the authors explain the participants' reactions to noise as a difficulty in concentration and irritation. Noise in the environment is harmful to their performance because it can cause damage and irritability. Noise interferes with learning, grades and health.[3] Noise is distracting and restricts attention level and cognition.

When noise is present, it requires students to make an extra effort and concentrate in order to separate what they are doing from background noise. According to Professor Mark Andrews, "background or low-level noise in the home, work or school often disrupts people's concentration". From research done by Dartmouth College, in many colleges, over 8% of the students report problems concentrating on their studies. Most of these students blame outside distractions for their problems. We have to recognize that noise is coming from everywhere and can affect a student's academic career which can lead to increased levels of stress, poor health and decreased levels of concentration.

---

[2] Gensler, comp. "Changing Course." PsycEXTRA Dataset (2012): 1-12. Gensler. Web. 1 May 2016.

[3] Emilse Aparecida Merlin Servilha, and Marina De Almeida Delatti. "College Students' Perception of Classroom Noise and Its Consequences on Learning Quality." College Students' Perception of Classroom Noise and Its Consequences on Learning Quality. N.p., n.d. Web. 01 May 2016.

4. Background Research

To start our project, we wanted to research what devices and applications like this already exist. We began by researching sound sensing systems that improve quality of life. Through this we were able to find that many hospitals throughout the United States already have systems like this installed in their buildings. Hospitals put sound sensing devices such as QuietyMe in many hallways and patient rooms to make sure that the noise does not get too loud to interfere with patient sleep and comfort.[4] We found that in most cases the system worked by placing a noise sensor in each room and hallway then measuring the noise levels. It then shows graphs of peak noise throughout the day to train nurses to change their behaviors. We can also see how other use sound sensors as a tool for measuring the sound levels and sending an alert to a front desk or using a light outside the room to signal a nurse based on noise level. A nurse, or other hospital staff, would then be able to check on the room and see why there was a spike in the noise level at that time.

We also research home monitoring systems featuring the Raspberry Pi.[5] The Raspberry Pi is very popular for many do it yourself projects and we wanted to see what people had used it for in the past. We had come across many people who wanted to create home monitoring systems for themselves. These systems often featured temperature sensors, humidity sensors, cameras, and motion sensors. Throughout our research, we didn’t find much about people using a Raspberry Pi to monitor sound in their homes. They would send the data they were monitoring to an application they could access from their personal computer, mobile phone, or other device. From this, we decided creating an application to monitor the sound would be best.

The last topic we needed to research before beginning our project was how Power over Ethernet was used. We found that many office buildings use Power over Ethernet to supply power to cameras throughout the buildings. Power over Ethernet was often times used to eliminate additional wiring. Power over Ethernet can be used to supply limited

[4] "Quietyme Unveils Intelligent Sensor For Real-Time Noise Reduction in Hospitals." HIT Consultant. N.p., 28 Apr. 2016. Web. 07 May 2016.

[5] Vujovic, Vladimir, and Mirjana Maksimovic. "Raspberry Pi as a Wireless Sensor Node: Performances and Constraints." 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (2014): n. pag. Web. 7 May 2016.

amounts of power and is best used for small devices like a camera or a Raspberry Pi. We found that Raspberry Pi could not be powered in the traditional sense and needed to be broken down into both ethernet and power.

5. Full Specifications Decisions

Once some research had been carried out, the next stage was to define the technical and user specifications and its purpose. Our team's project implementations both hardware device and web application are detailed next.

5.1 User Specifications

5.1.1 POE Sound Sensor Device

The Linux environment runs on the Raspberry Pi hardware. The system must have at least 990 Megabytes of free disk space to install the program and 512MB-1GB of memory is required to load the application. The user's computer transfers and receives data from a local server using basic networking protocols using either an ethernet cable or a Wi-Fi dongle. All the system's information is stored in the server's' own database which stores the data on the server's disk.

The interface for the sound monitor system has a Raspbian Wheezy OS interface loaded onto it. All code written and run on device is in Python. It is the preferred language of a Raspberry Pi. The code for the program on the device should be launched by typing into the terminal: python sound.py.

All data transferred between the server and the Central Networking Server, depicted in Appendix L, used UDP networking protocol over Power over Ethernet technology. Our project is dealing with Power over Ethernet capabilities which is connected to the Raspberry Pi, shown in Appendix K. Not only does it provide connectivity to the internet, but it powers the device.

5.1.2 Web Application and Data Storage

The Web Application is a HTML5/JavaScript project on NetBeans launched on a Windows 10 operating system. The application is coded using web development languages such as HTML5, CSS, JavaScript, Bootstrap, and jQuery. Using node.js allows us to create our real-time web application.

Data storage takes place on both the hardware device and the Central Networking Server, using a UDP server and client protocol. This is shown on Appendix L. The device has code that collects and sends the sound levels using a specific UDP port. The Central

Networking Server receives those sound levels using the same UDP port; however encodes data in JSON format into a text file on the Central Server. Both the programs on the device and the Central Networking Server are written in Python.

5.2 Technical Specifications

Below represents technical specifications only associated with software applications; however all other technical specifications for our project can be seen in Appendix A.

5.2.1 POE Sound Sensor Device

Technical Specifications	Description
Operating System	
Environment	Unix-like
Version	Raspbian Wheezy OS
Target Market	Raspberry Pi
Code	
Language	Python

5.2.2 Web Application and Data Storage

Technical Specifications	Description
Web Application	
Operating System	Windows 10
Web Development	Javascript, HTML5, CSS, Bootstrap, jQuery
Runtime Server	node.js

Network Protocols	TCP (HTTP)
IDE	NetBeans
Data Storage	
Language	Python
Network Protocols	UDP (unique port)
Communication	Both device and web application

6. POE Sound Sensor Device

6.1 Introduction

This section starts with the hardware designs for our POE adapter and Sound Sensor Device. We then discuss the decibel conversion algorithm. Next follows the section on alternative designs that evolved during our project. Finally the performance of the web application and the data handling is discussed.

6.2 Hardware Design

6.2.1 Our POE Adapter

Our hardware design features our own Power over Ethernet Adapter. Due to the restrictions on campus, we needed to include both a Power over Ethernet injector and a Power over Ethernet splitter to supply power to our device. The first part of our design is a Power over Ethernet injector. A Power over Ethernet injector is necessary when a Power over Ethernet switch is not available or turned on. The Power over Ethernet injector takes the ethernet cable and alternative power supply and produces an ethernet port with Power over Ethernet capabilities. From this we then needed to attach a Power over Ethernet splitter to split this back into a power supply and an ethernet signal. This would not be necessary for something with traditional Power Ever Ethernet capabilities, but due to the limitations of a Raspberry Pi at this point in time it was necessary. A Raspberry Pi cannot be powered just via its ethernet port. It still needs power to be sent in via its traditional Micro USB input port. We produced our own Power over Ethernet splitter that used the first two twisted pairs of cables to provide ethernet and used the second to along, with a step down converter to supply power, but it was not compatible with the PoE injector that we had purchased. For this reason, we also purchased a PoE splitter to supply the power to our Raspberry Pi.

We opted to use PoE technologies because if PoE was turned on throughout the campus, it would eliminate the use of a power outlet to run our system. This would mean less additional wiring was necessary to implement it throughout the university's campus. Using PoE also makes it slightly safer. There is less need to worry about too much power being sent into the device and burning out the hardware. In addition to those two reasons, PoE also is more secure. It is less likely for ethernet to fail than it is for Wi-Fi and it is often faster.

6.2.2 Sound Sensor Device

As shown in Appendix C, our original (prototyped) hardware design features a Raspberry Pi 2 Model B+ attached to a breadboard through a ribbon cable at its GPIO pins. The ribbon cable is then connected to a breakout board that allows us to use the input/output pins. Wired to the breakout board is an analog to digital converter. We chose MCP3008 because it is necessary to convert the analog output of our sound sensor, the DFRobot Sound Sensor, to a digital input the Raspberry Pi can work with. Analog sound sensors are external microphone to enhance power and can be paired with a preamp for improvements of sound quality. The prototyped design also requires a Wi-Fi dongle or an ethernet port because at this stage in our project we hadn't designed our POE adapter yet. To be able to test the code we had written, we needed to connect to internet.

For our final hardware design as shown in Appendix D, we designed and soldered our parts to a printed circuit board (PCB). This eliminated the mess of wires and movability. The final design features our Raspberry Pi 2 Model B+ attached to our custom built PCB through a ribbon cable. The ribbon cable connect the 40 GPIO pins to the breakout board to the analog to digital converter. The converter, MCP3008, converts the analog output to digital input of our sound sensor, DFRobot Sound Sensor. Although not depicted in Appendix D, the final hardware design is powered by Power over Ethernet. This technology is discussed more in the above section; however Raspberry Pi doesn't have POE capabilities. Therefore a POE splitter had to supply both power into the Raspberry Pi's micro USB port and connect it the internet into the Raspberry Pi's ethernet Port.

6.3 Decibel Conversion Algorithm

For our Decibel Conversion Algorithm, we could not find an easy formula to do the conversion accurately due to the outputs of our microphone. To maintain accuracy, we instead used a decibel meter to compare the sound level outputs from 0 - 1023, to the decibel outputs of the decibel meter. Using these numbers we were able to create a function that mapped the sound level measurements being taken into their equivalent decibel values as shown in Appendix H. When the script reads in a value from the sound sensor, it then checks to see where this value falls on a decibel scale and sends that value across the server instead.

Rather than mapping each individual decibel level from 50 to 100, we mapped it in sections. Our sound sensor reads anything as 50 decibels and below as 0 so we were not able to address levels below that cutoff. In our research, we found that 50 decibels was about the equivalent of the humming of a refrigerator. We decided it was ok that our sensor could not detect below 50 decibels as most classrooms have some background noise such as air conditioners, computers, printers, various other devices, etc. It was also hard to produce sounds above 100 decibels in a classroom and accurately measure them. To keep it in perspective, our application is focused on finding quiet study areas and 100 decibels is about the sound of a rock band. For this reason, we chose not to focus on values above 100 decibels. It is unlikely a student, faculty, or other would opt to study in a room above that level. We chose not to map every single decibel measurement possible because we were averaging decibels over 10 minute periods. This meant that the sound in the classroom was likely varying in 5-10 decibels and opted to round the values to the nearest 5 or 10 value.

6.4 Alternative Designs

Our first design originally involved an Arduino Uno. We both had background and familiarity in working with the Arduino. It is simple to code with, having its own platform and the main purpose of an Arduino board is to interface with sensors and devices. Another positive was that it is much easier to connect analog sensors to the Arduino. It can easily interpret and respond. The downfalls were that the Arduino is only a microcontroller, not a full computer. It can’t run a full operating system. Also, the Arduino isn't built for network connectivity directly out of the box.

We moved onto the Raspberry Pi 2 Model B+. A Raspberry Pi in general is a fully functional computer. It has a processor, memory, and graphics driver. It runs a version of Linux operating system. The Pi has built-in Ethernet port, allowing easy access to any network. This gave us more functionality, more reliability, and more speed because it is a fully functionally computer that requires a mouse and keyboard. With our POE adapter, the step down converter outputs a 5VDC.

6.5 Testing

There were multiple parts of our software on the hardware device that required testing. The first being our Decibel Conversion Algorithm. To test our decibel conversion algorithm we installed our system in a private room and played sounds that varied from 20 - 120dB to see what our system would output. We used a decibel meter to ensure that the values being sent to our web application accurately represented the sound levels in the room.

Secondly, we needed to ensure that our system was updating and sending the correct data across the server. We were able to watch the values that the Raspberry Pi was reading and see that they were the same values being sent to the web application at the appropriate times without delay. We also watched the time to make sure that it updated every 10 minutes like we coded the software to do.

We also needed to test that our system worked via Power over Ethernet and that it was sending the appropriate voltages across so as not to overheat the system. We tested our Power over Ethernet system in multiple places to ensure that it was working properly every time we used it. We also used an oscilloscope to measure the voltage being sent into our Raspberry Pi to make sure that it was at an appropriate level. Raspberry Pi requires a voltage from 3.3-5V and the Power over Ethernet splitter we designed outputted 5V when we measured it as it should. In our testing of Power over Ethernet we found that our Raspberry Pi could not be powered via ethernet if the hdmi video output was also plugged in. This is likely due to a circuit loop in the actual Raspberry Pi circuit board and not something we could address in the time given.

7. Web Application and Data Storage

7.1 Introduction

This section starts with the software designs for the web application and the data storage. Next follows the section on alternative designs that evolved during our project. Finally the performance of the web application and the data handling is discussed.

7.2 Software Design

7.2.1 Data Storage

To run the Raspberry Pi, it needed to have an operating system installed on it. The chosen operating system was the Raspbian Wheezy OS. To read input from the sound sensor, the project uses a python code that reads from the input pins every second. It averages the input every 10 minutes and sends that data to the Central Networking Server. The Server takes that data, puts it into JSON format, timestamps it, and dumps the information to a text file on the Server.

7.2.2 Web Application

The Web Application picks up the text file using AJAX request calls from the Central Networking Server. Having the data in the application allows for data analysis and manipulation. The AJAX allows the data such as noise levels, data and time of when the noise of created to be displayed on the web application. This is all directly taken from the file as seen from Appendix I. The code is written in Javascript.

The Web application was written using HTML, CSS, and Javascript in Bootstrap format. Bootstrap allowed for fast development of the web application. Because of its fluidity and ready-made classes, creating the web application with responsiveness and functionality was quick, easy and consistent. The entire page is uniform, making it appealing to the user. On the developer side, using Bootstrap made it customizable to the specifications of our project. We were able to pick and choose the features that we needed. As shown in Appendix J, screenshots of the web application were taken to shown Bootstrap framework was implemented because of its consistent and user-friendly design.

7.3 Testing

Trial and error came into play when trying to figure out how to store the data and upload the information to the web application. After various techniques and errors such as Cross Origins Request, we sought out Professor Femister for help. He is a professor in the Department of Computer Science and was currently teaching Web Apps. Using him a resource, he guided us to restructure our data storage. It was his suggestion to reformat the information going to the Central Networking Server into the file to JSON format. An example of the data in JSON format being uploaded into the file is depicted below:


```
[
  {
    "currentDate": "05/06/2016",
    "currentTime": "17:37:43",
    "soundavg": "70"
  }
]
```


Once we have that information in JSON format in a file, it is easy to exact using jQuery AJAX requests. After some back and forth with the code, the request call worked and the noise levels are displayed on the screen along with the data and time.

It was a struggle to figure out how to display the information once in JSON format because it isn't true JSON. The data is in a text file, not a JSON file.

8. Project Management

This section details some aspects of our project management that took place. Our project management started with splitting the overall task of building the POE sound sensor and web application into smaller sub-systems. These sub-systems were then assigned to the appropriate team members. This is covered under 8.1 Team Management. Then the timing of the project had to be planned so that team members knew about the deadlines involved. This is outlined in section 8.2 Time Management; the timing had to be monitored and revised during the course of the project. Finally, the management of our group's budget required constant attention throughout the project is detailed in section 8.3.

8.1 Team Management

The first task was to split the sound sensor and web application into components. These components had to be treated as separate projects for the purposes of allocating project marks. Many of these components were assigned by default to team members; however sometimes needed to collaborate due to strengths and weaknesses of our team.

8.2 Time Management

This section details the time management that took place for the team individually as well as a whole, examining our various Gantt Charts.

As seen in Appendix F, the Fall Semester was concentrated on setting up our hardware, the Raspberry Pi. We were first learning and researching our topic with basics and starter codes. It was important to learn as much as we could. In the first few months, we faced a series of complications with our Raspberry Pi, but we got back on our feet by overcoming those problems. The first semester of the project was also where our course load was almost overbearing. We were both taking extremely difficult classes that required as much attention as this project. Our dedication to senior design for Fall Semester fell through because we couldn't handle everything.

As Spring Semester rolled in, we implemented a weekly Scrum meeting schedule in addition to our advisor appointments because Fall Semester wasn't too successful. With 5 months to turn it around, our main focus was senior design. During those Scrum meetings, we discussed what we accomplished that week, what problems we had faced, what our goals are for next week. In Spring semester, we also made the project break down more apparent such as who was doing what. This was vital because we didn't need overlap in work and some had strengths in certain areas than others. In Appendix G, the Gantt Chart is broken down between hardware and software because that is the project breakdown as discussed previously. This was our primary focus and goal. With the project assignment and the Scrum meetings, we finished our project.

Both Gantt Charts show us working on our reports and presentations. Multiple drafts had to be submitted in for the reports, poster boards, and the presentations to either our advisor or coursesite.

8.3 Cost Analysis

The project this semester remained under the budget of $500. In the Bill of Materials as shown in Appendix B, the total amount spent is less than $230 not including shipping and tax. The Bill of Materials shows all purchases including distributors and manufacturers. The hardware such as the Arduino Uno and Raspberry Pi 2 we had on hand. This cut a significant of the cost down.

If we were to continue the project and manufacture these systems, the cost would go down tremendously. Buying in bulk would decrease the price of everything we purchased. We also considered using a Raspberry Pi Zero instead of a Raspberry Pi 2 Model B+ as the processor for the sound sensor hardware device. The Raspberry Pi Zero is only $5. A Raspberry Pi 2 is around $35 - $40.

A benefit of our project is the knowledge base for our field. What we have been learning this semester and will be continuing to learn are the stepping stones of our future careers. Project development and management are in high demand and developing these skills well can really make a difference between a project coming in on time and on budget and it being a failure.

9. Ethical Consequences

Anytime you are monitoring your surrounding it is called into question about whether or not you are invading one's personal privacy. When looking into this project, confidentiality and privacy were brought into question. Monitoring only noise levels is the concern of our project. No conversations or video recordings will be taped. No one will be threatened or violated in the progression of this project. No ethical consequences will be crossed. This project is only beneficial to students and their academics life.

10. Industry Standards

The International Standard 9614-1 is the standard for the determination of sound power levels of sources using sound intensity measurements. This relates to our project because our sensor gives the sound intensity measurements and we must take these and determine sound power levels. From this standard, we concluded we need to determine sound power levels in order to make our sound intensity measurements useful to the user of our products. We should follow this standard because it sets forth the ideal for this conversion and will ensure we do it properly.

The ANSI s1.4.1983 is a standard on sound level meters. It states that a sound level meter will consist of a microphone to convert sound pressure signals to electrical signals, an amplifier to raise microphone outputs to a useable level, a level range control to shape the frequency response, and an indicating system to display sound levels. It may also contain a sensitivity control to allow adjustment of amplification. The standard provides frequency weightings and exponential-time-averaging-statistics.

This relates to our project because we are creating a power over ethernet sound sensor. We use everything discussed in the above standard. We have a microphone connected to amplifier and then a device to show us the levels produced. The microphone provides data sheets and stats about the sensitivity and power of the microphone that allow us to convert the sound levels given to useful information for our project. We are able to convert our levels given to decibels because of these stats given. Our sound sensor meter, as far as we can tell, meets all the specifications set forth by this rule and make it a useful piece to our design project.

In the IECC/TR 62652 ED.1.0 EN: 2010 standard, it sets the standard for connecting circuits via power over ethernet. It discusses safe ways to connect and disconnect connections. It discusses cabling for power over ethernet. This relates to our project because we power our sound sensor system via power over ethernet. We should follow this standard because it sets the standard for how to connect things via power over ethernet. It also discusses the safe and right ways to power things via power over ethernet which is very important to our project.

11. Intellectual Property Considerations

Our project's implementation incorporates Power over Ethernet sound monitoring devices. It falls under the classification of Electrical and Audio Signal Processing Systems and Devices because a PoE sound monitoring device is an audio signal processing device. More specifically, it directly monitors sound levels.

Patent 8,194,866 is a sound monitoring, data collection and advisory system. It features one or more sound level indicators that are connected on a network server. Each of the sound level indicators measures the sound level at its specific location. The sound level indicators can also provide visual representation of the level of the sound. The visual representation can be a light that can be controlled through the system. The sound level indicators are monitors remotely on the network server they are connected to. The system provides an application to monitor multiple sound level indicators in real-time while also collecting the data to monitor its history.

The system is used to control noise traffic in the user's selected areas. This is important to guarantee desired noise levels at various locations such as office, a hospital, a school, a museum, and various other buildings where noise control is essential to level of comfort. The system collects characteristic data of the noise levels and provides a representation of it.

The sound levels are measured in decibels (dB) which are a logarithmic unit of measurement that expressed the magnitude of a quantity relative to an implied level. The system provides graphical representation software to visualize sound level history for the user. The devices can feature power over ethernet technology to allow for flexible installation. It uses dynamic host configuration protocol to allow it to operate in an IP network. It features a multimedia card interface. It can work over a network server or on its own. If it is operating over the network server it sends and receives data via the server if it is operating in standalone it collects the data and stores it.

This system relates to our senior design project because it is a sound sensor used to collect and monitor sound levels at various locations in order to improve the quality of life. Our senior project is a POE sound sensor that will monitor sound levels in various classrooms around campus to try to help students find a quiet study spot. The data will be used then to help students on campus academically and mentally.

The system also allows for data to looked at remotely via a server and seen in a graphical form which is what we planned to do for our project. It can also view the data in real-time while also collecting it and compiling it which is what we will be doing with ours. The system can operate without connection to the server and store on a card installed in the sound level indicators. We will be putting an SD card in our raspberry pi to collect data while also sending it over the server.

The system also specified that it will allow for POE technology which is the basis for our project. The implementation of our project does not run over ethernet; however our final product can be solely powered via power over ethernet.

US Patent 20140362999A1 is a system for generating an alarm for a certain level of work in a workspace. It has a threshold level that when passed can communicate with a system and send an alert to the users.

High levels of sound in one area can be disruptive to other areas of a workspace and can prevent people from working fully. This invention features at least one sound sensor connected to an outputting signal device. The output device features some kind of visual representation to show that the sound is at a certain level such as a light. Based on the lighting the individual can adjust their sound level to meet the needs of the workspace.

The system is to be mounted somewhere such as a wall. It is claimed that there is at least one sensor connected to an alert signaling device that is within a workspace. It is to detect when at least one aspect of sound exceeds a certain threshold and further indicate that to the user.

This is similar to the project of this report in that it is a sound monitoring device that can detect levels of sound. The project of this report is used to detect levels of sound in a dorm or workspace to further improve quality of life levels. This patent is intended for use in an office space where high levels of noise could be conducive for productive work. If the system is mounted in one room, it will detect and alert users when it has passed the threshold and could begin to interrupt users in the surrounding rooms so they can adjust their sound levels. The project of this report is similar to ours in the respect that we are collecting data in classrooms or known study spots on Lehigh University’s campus. Sound monitors were placed in those rooms and detected sound levels to allow students to know if it will be too noisy to study or do work.

This patent would be mounted somewhere in the room likely on the wall just as the sounds monitors of this report's will likely be when they are ready to be implemented.

12. Governmental Regulation Impact Study

FCC Title 47 part 15 involves radiators, transmitting devices, and emissions. It states that all radiators must either be licensed or meet the guidelines set forth by the rest of the provision. It states that transmitting devices may not cause interference and instead must accept interference and prohibits the operation of the device if the operator is notified it is causing interference. It prohibits damped wave transmissions and prohibits operating a device for unlawful eavesdropping.

This regulation relates to our project because it regulates the use of transmitters. Our project, a power over ethernet sound sensing device using transmitters in multiple forms. Our device first uses a microphone and amplifier to receive sound levels. Many microphones and amplifiers are required to be licensed under this regulation. Because our microphone and amplifier were purchased from a large company, it is likely they were registered if they needed to be. Our project also submits information via a wireless network. Many wireless devices also require regulation under this title.

13. Health and Safety Study

In an FDA regulations for laser products, manufacturers of electronic radiation emitting products are subject for regulation under the FDA. This rule defines and classifies both lasers and laser pointers. This limits the strength of both visible and invisible laser wavelengths. The FDA requires warning labels on certain classifications of lasers. The rule also defines appropriate and inappropriate use of different lasers and other technology.

This law does not apply to our senior design project as it regards lasers and similar technology. Our project does not use any kind of laser technology or any kind of LED. Our project uses simple power over ethernet connection, a sound sensor and sends information via a wireless connection. It does not do any of the following set forth by this regulation.
The IRB-Health and Human Services states that an institution must have an FWA in order to receive support from Health and Human Services for research on human subjects and each FWA must have at least one Institutional Review Board registered with the office of human research protections. Before an institution may obtain an FWA it must register with an Institutional Review Board.

These regulations do not relate to our project. Our project did not do any research on human subjects. The primary safety concern is with the hardware device. There is potential for a short circuit in the system, which can result in destruction of internal circuitry as well as fire. Another electrical issue is essentially the sensors and POE technology on the hardware device are currently exposed to environmental dangers.

14. Environmental Impact

Using a Raspberry Pi id low energy, which leads it to have a very small environmental impact, if any environmental impact at all. Looking at our project as a whole, there are again very little environmental impacts due to the low energy used. It is possibly that if equipment is kept on for too long or overused, that may lead to a higher energy consumption.

15. Integration Effort

In this section some of the key integration attempts are discussed. A diary style approach has been avoided as much as possible, but as each attempt builds upon the last there is a requirement for it to be presented in chronological order. As such it has been broken into two sections – one for each semester.

15.1 Fall Semester

Near the end of the fall semester we were still working separately to try and create and build various components of our project. One part was to build the sound sensor system, to fix the decibel conversation code, and then draft a PCB board. Our first sound sensor was used; however due to quality issues, a second sound sensor was being researched. The device was connected to an HDMI cable, powered up, and code was written in Python by Zoe Protin.

The integration phase involved using the UDP socket protocols to communicate between the hardware device system and the Central Networking Server. This proved more complicated and was only partially successful. The system integrated with a Wifi dongle and would receive noise levels of the room using UDP protocols with a unique UDP Port. It was invaluable because it determined problems early and hence they could be fixed by the spring semester. This phase integrated work by the author, Samantha Shreck.

15.2 Spring Semester

The spring semester was when the majority of the integration occurred. The problem was with the Python code and how the Central Networking Server was receiving the data and hence it could be using the Python Standard Library to create JSON formatting into text file. By writing the data into JSON format, the web application can request the information quickly and read it easily. In the end, the Central Networking Server creates a text file, with each entry containing the noise levels and timestamp, all in JSON format.

Zoe Protin had managed to get the sensors along with decibel conversion algorithm working in time for the integration attempt, allowing Samantha Shreck to work on data storage from the device. By the end of the semester, POE technology we developed powered the system and was tested as well.

16. Conclusion

Throughout the course of the year, we were able to achieve our goals in creating a sound sensing system that uses POE technology and sends the sound levels across a server to an easily accessible web application that displays those sound levels. We wanted to create something that would make student's life easier at Lehigh University, solve a problem we may have faced ourselves in our years spent at Lehigh University. Throughout our 4 years here, one of the biggest struggles we faced was finding good, quiet places to study on campus and therefore we sought to solve this problem through our senior design project. We created a device that could be installed in any classroom throughout campus via just an ethernet port with POE capabilities. It would record the sound levels in the classroom or other area and send that data to a server. The sound levels could then be monitored with our web application. The concept of our project is when looking for a quiet place to study a student would not have to use limited studying time to find one. An important issue that we aimed to address was students' privacy on campus. We wanted to make sure that our application would not be an invasion of students' personal lives. For that purpose we used a sound sensor that could only record sound levels and not actually record conversations or other sounds, even if hacked. We achieved everything we hoped to for on our project this year and could see it being of great use if implemented at Lehigh or any other university.

17. Future Works

The sound sensor used with POE technology is not a quality sensor. Future improvements would be investing time and research into a new sound sensor, preferably one that takes in digital inputs of noise levels.

The real sensor readings obtained need further time to determine if they are useful or not. The random samples chosen may have been unrepresentative. Otherwise, the problem needs to be identified and perhaps then the sensor data can be used to help estimate the noise levels in a given classroom.

Another additional feature that should be added is a motion sensor of some sort. This future plan including a motion sensor so that the system knows exactly who is in the room and can be more precise. This would give a better idea of the number of people there are possibly in the room, compared to relying upon how noisy it is. A motion sensor could vastly improve the accuracy and efficiency of the entire system.

18. APPENDICES

Appendix A: TECHNICAL SPECIFICATIONS

POE Sound Sensor Device

Parts	Description
PCB	
Raspberry Pi 2 Model B+	Runs sound sensor system; the computer
Analog Sound Sensor	Reads in noise levels
ADC Chip	Converts analog to digital
Ribbon	Allows for the connection of GPIO pins
Micro SD card	Loaded with Raspbian and code
Code Language	Python
Controlled with	Mouse and Keyboard
Powered by	POE
POE Adapter	
USB Ethernet Adapter	Ethernet to USB
12V-5V Step Down Convertor	Supplied voltage
2x Micro USB Plugs	For power and ethernet adaptor
Thin Wires	For connections
Raspberry Pi 2 Model B+	
Processor Chipset	Broadcom BCM2836 ARMv7 Quad Core
RAM	1 GB SDRAM @ 450 MHz
USB 2.0	Four USB Ports
GPIO	40 Pins
Ethernet	Yes
Input Voltage	5V
Flash Memory Storage	Micro 64 GB SD card
Operating System	Linux distributions (Raspbian Wheezy OS)
IDE	Scratch (anything with Linux support)
HDMI	Yes

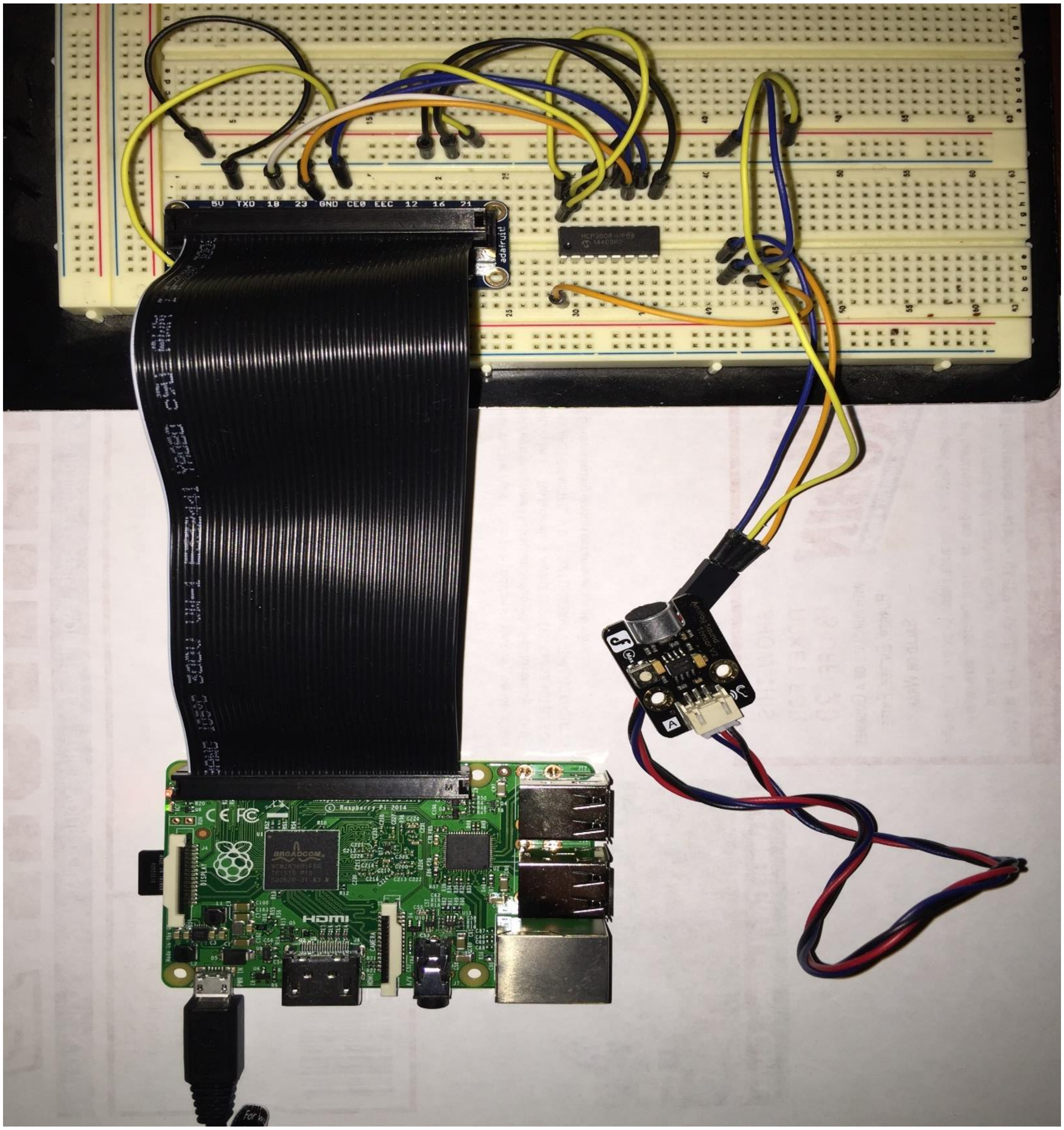
Web Application and Data Storage

Parts	Description
Web Application	
IDE	NetBeans
Target Market	Students
Functional	Yes
Ajax request calls	Displays noise levels
Web Server	Node.js
Code Language	HTML, JavaScript, Bootstrap, CSS, jQuery
Data Storage	
Code Language	Python
Storing a file	Yes
Communicating by	UDP Networking Protocols
Using POE	Yes
IDE	Raspbian, NetBeans

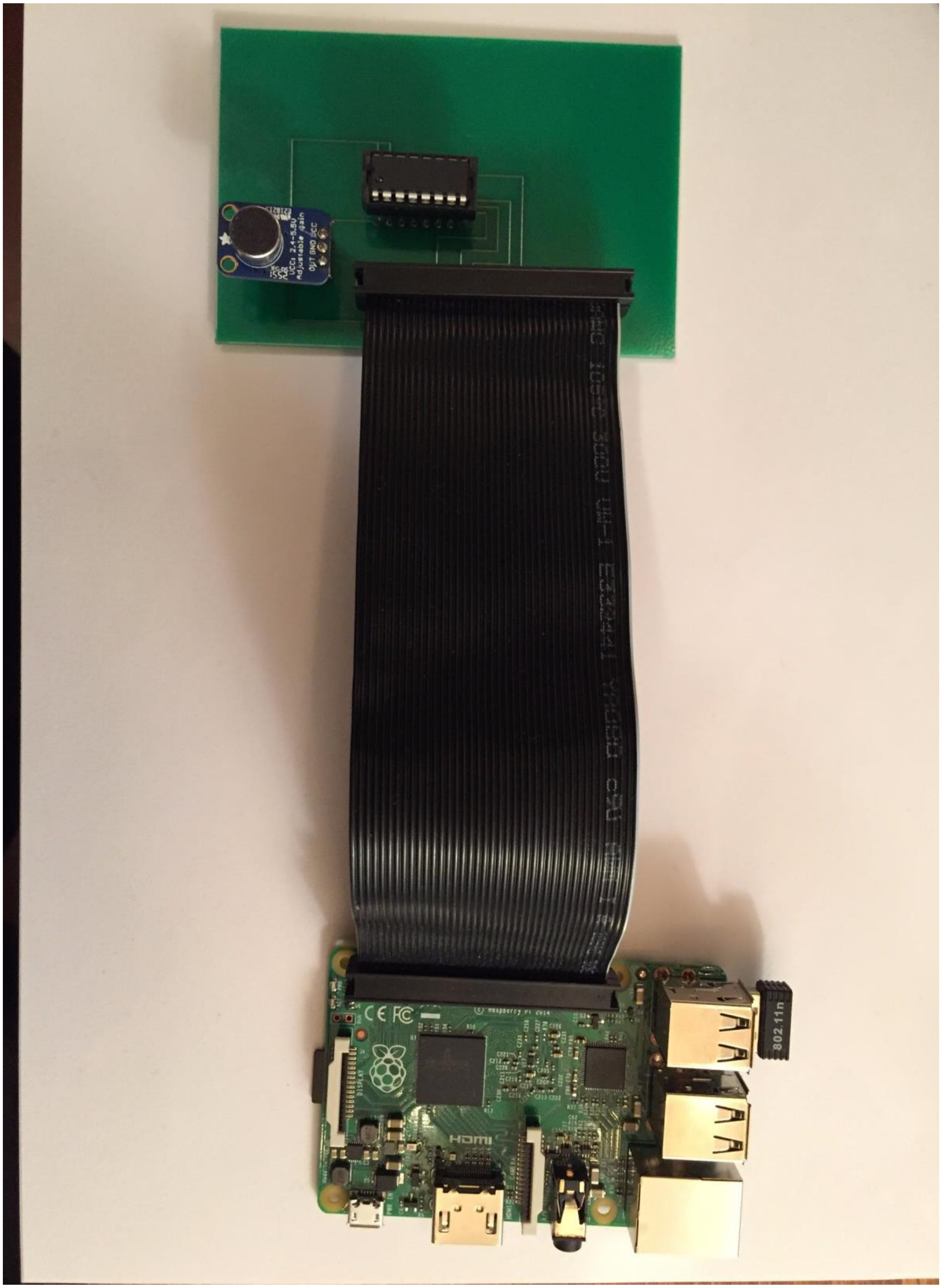
Appendix B: BILL OF MATERIALS

Item	Reference #	Description	Quantity	Manufacturer	Distributer	M. P/N	D. P/N	Price
1	Cobbler26	Assembled Pi 26 Pin Cobbler	1	Adafruit	Mouser Electronics	914	485-914	$6.50
2	A2D	MCP3008	3	Adafruit	Mouser Electronics	865	483-865	$33.71
3	Cobbler40	Assembled Pi 40 Pin Cobbler	1	Amazon	Adafruit	B00Q1T07O8	NS-GPIO-P-01-A	$11.82
4	Wifi	Miniature Wifi Module, Raspberry Pi	1	Amazon	Adafruit	B00GSNBPXA	814	$18.57
5	Sound1	Analog Sound Sensor	2	DFRobot	Adafruit	RB-Dfr-38	DFR0034	$13.80
6	Board	PCB	3	ExpressPCB	Robotshop	ISH-2874103123	32892823	$90.93
7	Sound2	Electret Microphone Amplifier	1	Amazon	Adafruit	B00K9M6S1O	100243	$9.95
8	Stepdown	12V to 5V Step Down Converter	2	SMAKN	Amazon	B00CXKBJI2	0B3627	$13.98
9	EtherAdapt	USB Ethernet Adapter	2	Amazon	Linksys	B0092TRAXK	0A3622	$19.98
10	Splitter	2X USB Splitter	2	Amazon	TP-Link	TL-POE10R	845973030490	$4.56
		Total Quantity:	18	**Total Price:**				$223.80

Appendix C: PROTOTYPED DESIGN



Appendix D: FINAL DESIGN



Appendix E: PATENTS

US 20140362999A1

(19) **United States**

(12) **Patent Application Publication**

Scheper et al.

(10) **Pub. No.: US 2014/0362999 A1**

(43) **Pub. Date: Dec. 11, 2014**

(54) **SOUND DETECTION AND VISUAL ALERT SYSTEM FOR A WORKSPACE**

(71) Applicants: **Robert Scheper**, Grand Rapids, MI (US); **Matthew Warner**, East Grand Rapids, MI (US); **Karl J. Mead**, Grand Rapids, MI (US); **Brett Kincaid**, Ada, MI (US); **Kurt Heidmann**, Grand Rapids, MI (US)

(72) Inventors: **Robert Scheper**, Grand Rapids, MI (US); **Matthew Warner**, East Grand Rapids, MI (US); **Karl J. Mead**, Grand Rapids, MI (US); **Brett Kincaid**, Ada, MI (US); **Kurt Heidmann**, Grand Rapids, MI (US)

(21) Appl. No.: **13/911,654**

(22) Filed: **Jun. 6, 2013**

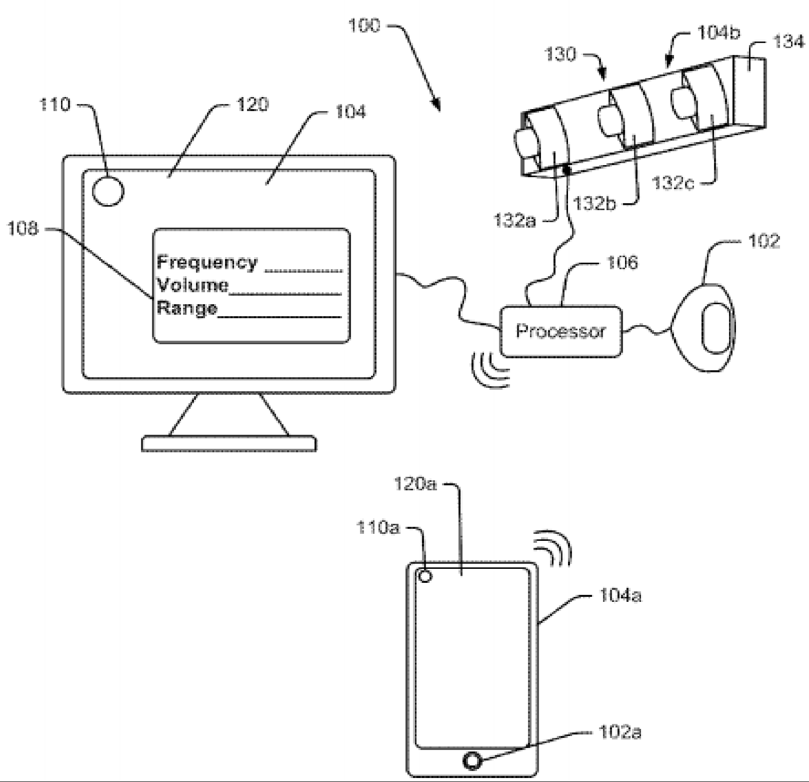
Publication Classification

(51) **Int. Cl.**
G08B 21/18 (2006.01)
H04R 29/00 (2006.01)

(52) **U.S. Cl.**
CPC ***G08B 21/182*** (2013.01); ***H04R 29/00*** (2013.01)
USPC **381/56**

(57) **ABSTRACT**

An alert system for generating a signal indicating at least one aspect of sound within a workspace environment, the system comprising at least a first sensor positioned one of within and proximate a workspace, the at least a first sensor sensing at least one aspect of sound generated within the workspace and generating a signal indicating the at least one aspect, a communication device located within the workspace, the communication device configured to generate a signal perceivable within the workspace and a processor receiving signals from the at least a first sensor and linked to the communication device, the processor programmed to perform the steps of identifying when the at least one aspect exceeds a threshold value and driving the communication device to indicate that the at least one aspect exceeds the threshold value.



US 20090052677A1

(19) **United States**

(12) **Patent Application Publication**

Smith

(10) **Pub. No.: US 2009/0052677 A1**

(43) **Pub. Date: Feb. 26, 2009**

(54) **SOUND MONITORING, DATA COLLECTION AND ADVISORY SYSTEM**

(76) Inventor: **Christopher M. Smith**, New Palestine, IN (US)

Correspondence Address:
BOSE MCKINNEY & EVANS LLP
111 MONUMENT CIRCLE, SUITE 2700
INDIANAPOLIS, IN 46204 (US)

(21) Appl. No.: **12/195,366**

(22) Filed: **Aug. 20, 2008**

Related U.S. Application Data

(60) Provisional application No. 60/965,448, filed on Aug. 20, 2007.

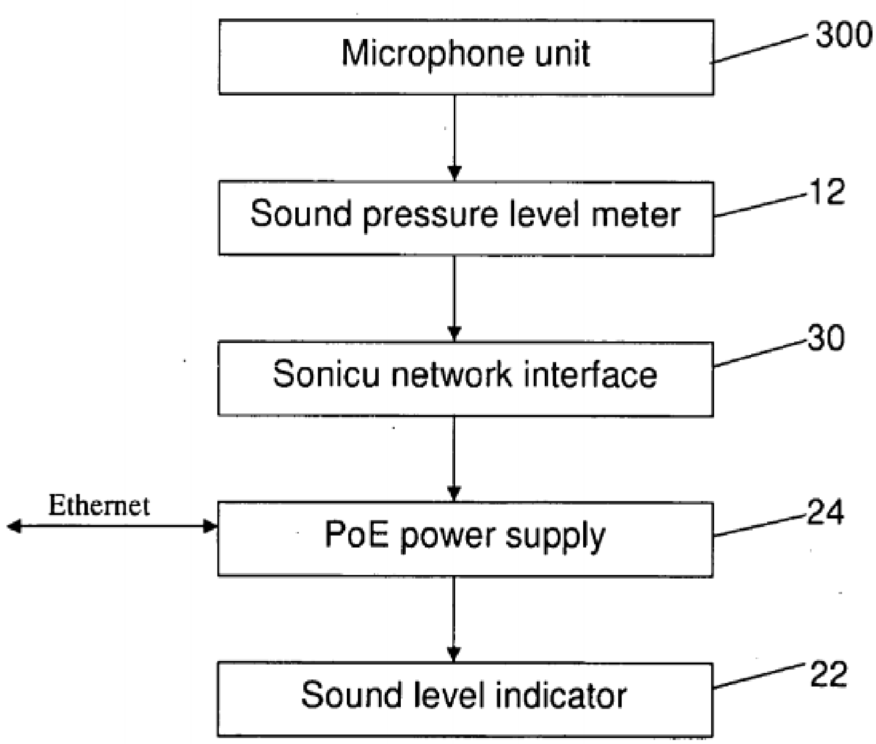
Publication Classification

(51) **Int. Cl.**
H04R 29/00 (2006.01)

(52) **U.S. Cl.** **381/56**

(57) **ABSTRACT**

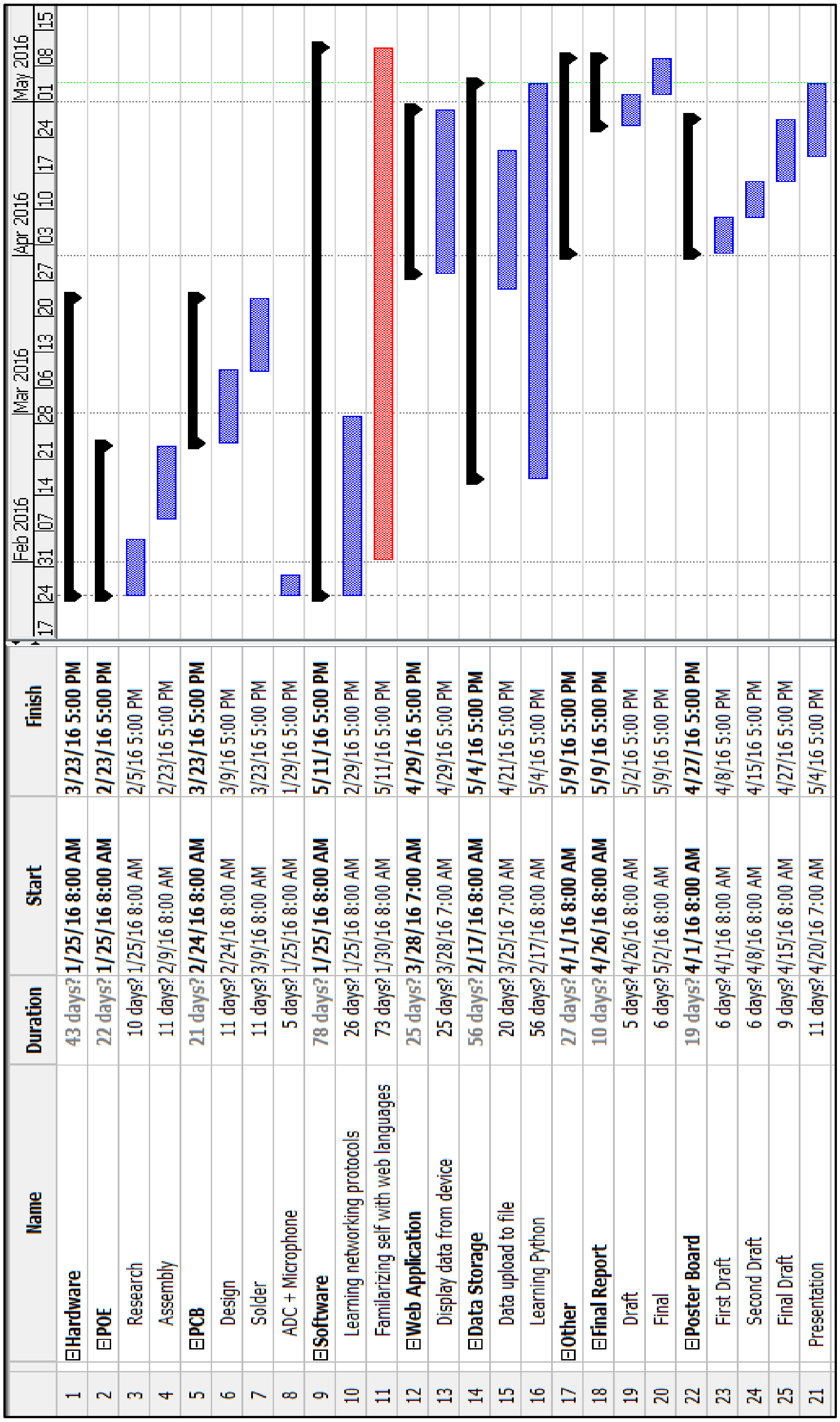
A sound monitoring system and method. The system can include a plurality of sound pressure level meters, a plurality of sound level indicators and a server connected by a network. The sound pressure level meters measuring a sound level at their location, and the sound level indicators providing a visual indication of the sound level measured by at least one of the sound pressure level meters. The system devices can be powered, as well as monitored and controlled remotely over the network. A user interface enables constructing and monitoring multiple zones and groups in a monitored area, as well as reviewing real-time and historical sound data. The system can also control lighting in the monitored area, and use lighting as a visual indicator of noise level.



Appendix F: FALL SEMESTER GANTT CHART

ID	Task Name	Duration	Start	Finish
1	**<First Semester>**	**76 days**	**Mon 8/24/15**	**Mon 12/7/15**
2	Acquire Raspberry Pi 1	4 days	Tue 9/1/15	Fri 9/4/15
3	Make Gantt Chart	5 days	Mon 9/7/15	Fri 9/11/15
4	Purchase Sound Sensor	2 days	Fri 9/11/15	Mon 9/14/15
5	Design starter code with sensor	6 days	Mon 9/14/15	Sun 9/20/15
6	Re-familiarize selves with hardware	6 days	Fri 9/18/15	Fri 9/25/15
7	Complete User Specification	5 days	Mon 9/28/15	Fri 10/2/15
8	Purchase 26 pin cobbler and ADC chip	4 days	Thu 10/1/15	Tue 10/6/15
9	Patent search	5 days	Fri 10/2/15	Thu 10/8/15
10	Install linux OS on personal laptop	3 days	Wed 10/14/15	Fri 10/16/15
11	Acquire Raspberry Pi 2	3 days	Thu 10/15/15	Sat 10/17/15
12	**Setup of Raspberry Pi(s)**	**74 days**	**Wed 8/26/15**	**Mon 12/7/15**
13	Load linux onto Raspberry Pi(s)	11 days	Fri 10/23/15	Fri 11/6/15
14	Install Node.js on Raspberry Pi(s)	3 days	Fri 11/6/15	Tue 11/10/15
15	Complete Standards and Regulations	6 days	Sun 11/1/15	Fri 11/6/15
16	Purchase 40 pin cobbler and USB wifi	13 days	Thu 11/12/15	Mon 11/30/15
17	Work on Interim Presentation	14 days	Sun 11/15/15	Wed 12/2/15
18	Work on Interim Report	17 days	Sun 11/15/15	Mon 12/7/15
19	**Other**	**70 days**	**Tue 9/1/15**	**Mon 12/7/15**
20	Research	70 days	Tue 9/1/15	Mon 12/7/15
21	Title Change	46 days	Mon 10/5/15	Mon 12/7/15

Appendix G: SPRING SEMESTER GANTT CHART



	Name	Duration	Start	Finish
1	⊟**Hardware**	43 days?	**1/25/16 8:00 AM**	**3/23/16 5:00 PM**
2	⊟**POE**	22 days?	**1/25/16 8:00 AM**	**2/23/16 5:00 PM**
3	Research	10 days?	1/25/16 8:00 AM	2/5/16 5:00 PM
4	Assembly	11 days?	2/9/16 8:00 AM	2/23/16 5:00 PM
5	⊟**PCB**	21 days?	**2/24/16 8:00 AM**	**3/23/16 5:00 PM**
6	Design	11 days?	2/24/16 8:00 AM	3/9/16 5:00 PM
7	Solder	11 days?	3/9/16 8:00 AM	3/23/16 5:00 PM
8	ADC + Microphone	5 days?	1/25/16 8:00 AM	1/29/16 5:00 PM
9	⊟**Software**	78 days?	**1/25/16 8:00 AM**	**5/11/16 5:00 PM**
10	Learning networking protocols	26 days?	1/25/16 8:00 AM	2/29/16 5:00 PM
11	Familiarizing self with web languages	73 days?	1/30/16 8:00 AM	5/11/16 5:00 PM
12	⊟**Web Application**	25 days?	**3/28/16 7:00 AM**	**4/29/16 5:00 PM**
13	Display data from device	25 days?	3/28/16 7:00 AM	4/29/16 5:00 PM
14	⊟**Data Storage**	56 days?	**2/17/16 8:00 AM**	**5/4/16 5:00 PM**
15	Data upload to file	20 days?	3/25/16 7:00 AM	4/21/16 5:00 PM
16	Learning Python	56 days?	2/17/16 8:00 AM	5/4/16 5:00 PM
17	⊟**Other**	27 days?	**4/1/16 8:00 AM**	**5/9/16 5:00 PM**
18	⊟**Final Report**	10 days?	**4/26/16 8:00 AM**	**5/9/16 5:00 PM**
19	Draft	5 days?	4/26/16 8:00 AM	5/2/16 5:00 PM
20	Final	6 days?	5/2/16 8:00 AM	5/9/16 5:00 PM
22	⊟**Poster Board**	19 days?	**4/1/16 8:00 AM**	**4/27/16 5:00 PM**
23	First Draft	6 days?	4/1/16 8:00 AM	4/8/16 5:00 PM
24	Second Draft	6 days?	4/8/16 8:00 AM	4/15/16 5:00 PM
25	Final Draft	9 days?	4/15/16 8:00 AM	4/27/16 5:00 PM
21	Presentation	11 days?	4/20/16 7:00 AM	5/4/16 5:00 PM

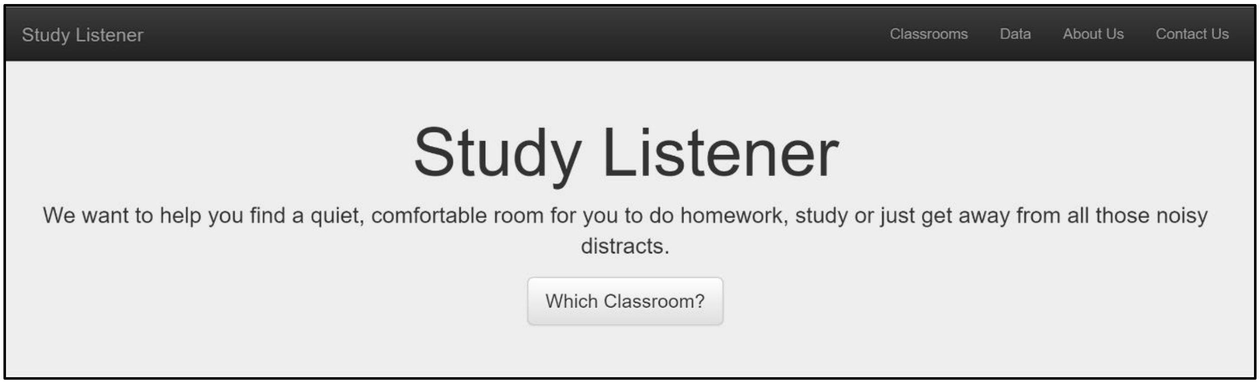
Appendix H: DECIBEL CONVERSION ALGORITHM CODE

```
soundsensor_adc = 0

last_read = 0
tolerance = 5


UDP_IP = "192.168.2.9"
UDP_PORT = 12345

while True:
  while True:
    tenmins = []
    i = 0
    while i < 60:
      sound_sensor = readadc(soundsensor_adc, SPICLK, SPIMOSI, SPIMISO, SPICS)

      sound = sound_sensor
      tenmins.append(sound)
      i = i + 1
      time.sleep(1)
    tensum = sum(tenmins)
    tenavg = tensum/60
    decibel = 50
    if tenavg >= 60:
      decibel = 100
    elif tenavg >= 50:
      decibel = 90
    elif tenavg >= 30:
      decibel = 80
    elif tenavg >= 20:
      decibel = 75
    elif tenavg >= 10:
      decibel = 70
    elif tenavg >= 5:
      decibel = 65
    elif tenavg > 0:
      decibel = 60
    else:
      decibel = 50

    print 'Sound = {sound_sensor}' .format(sound_sensor = tenavg)
    print 'Decibel = {decibel}' .format(decibel = decibel)
```

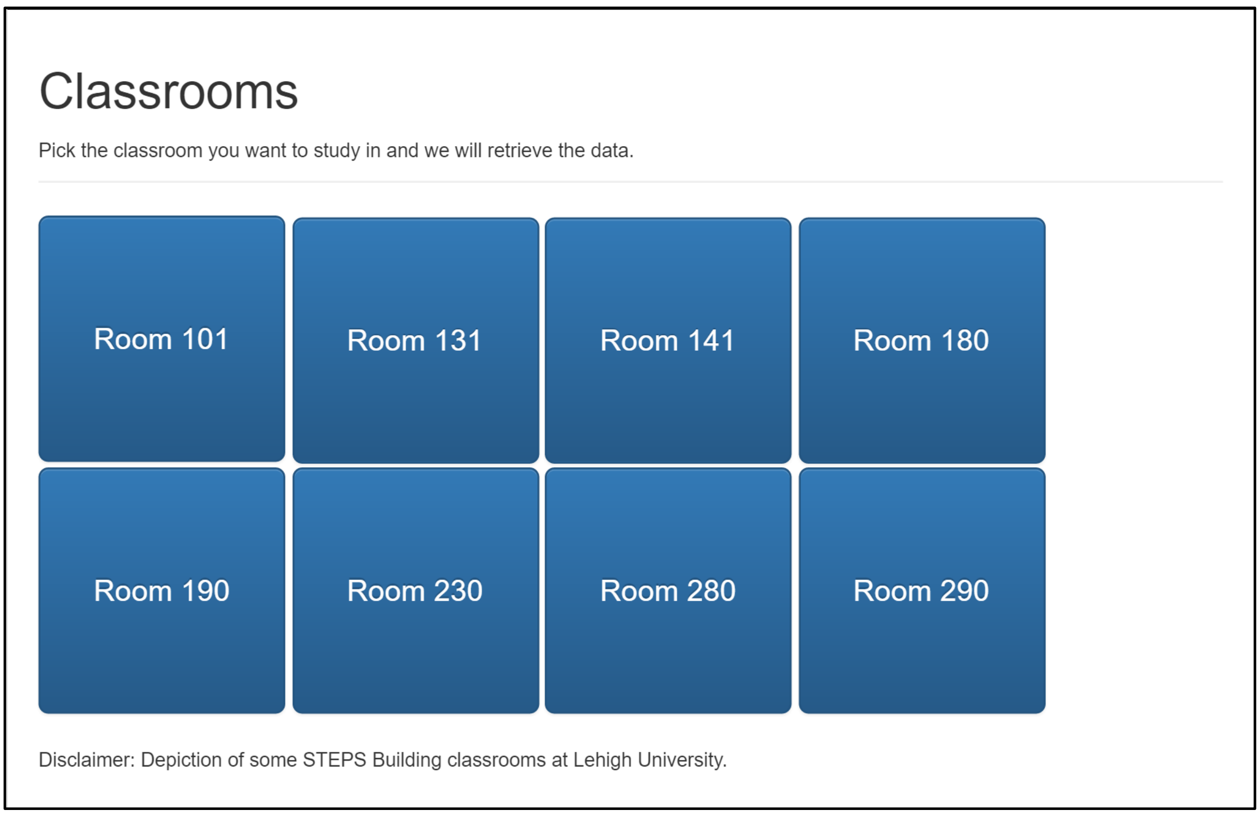
Appendix I: AJAX REQUEST CODE

```
var container = $('div.info');
$(document).ready(function () {
  $.ajax({
    url: "SoundLevels/raspdata.txt",
    dataType: 'json',
    type: "GET",
    success: function (data) {
      $.each(data, function(index, item){
      //    container.append(index + ' : ' + item + '</br>');
      //});
        $('div.info').append($('<div>',{
          text: "The Noise Level: " + item.soundavg
        }));
        $('div.info').append($('<div>',{
          text: "Last Refreshed: " + item.currentDate + "  " + item.currentTime
        }));
      });
    },
    error: function(xhr, status, text) {
      console.log(status + ' ' + text);
      $("div.info").html("<p>An error has occurred.");
    }
  });
});
```

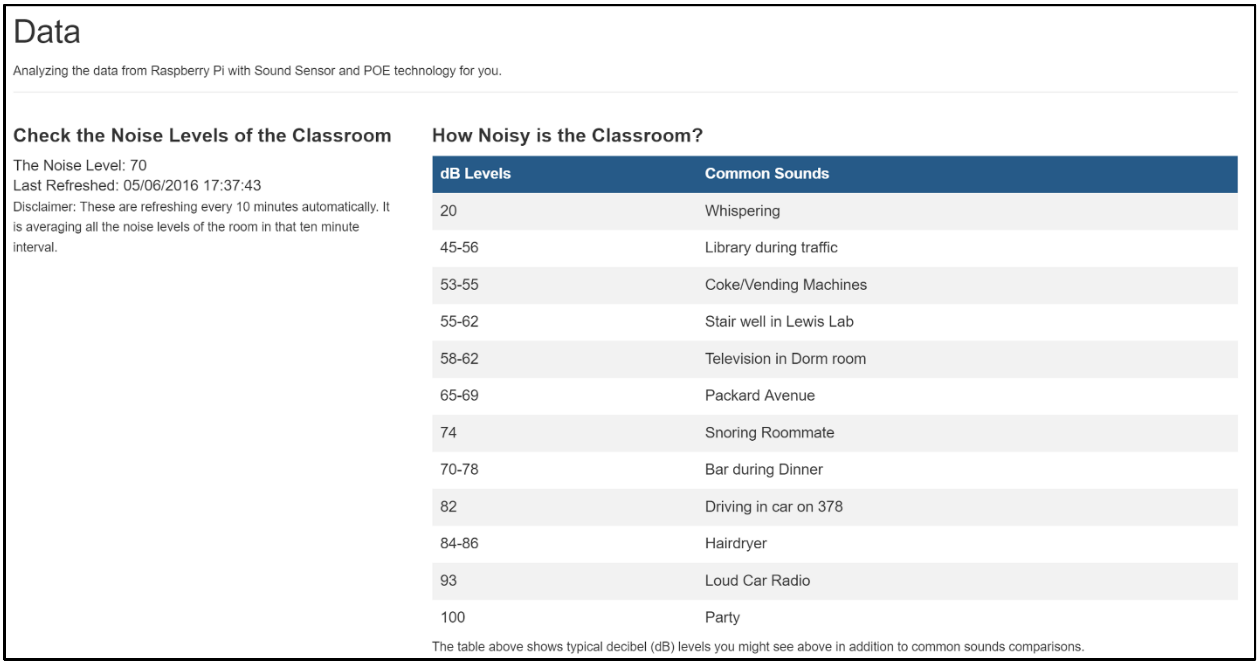
Appendix J: WEB APPLICATION SCREENSHOTS



Menu bar of the web application



User chooses classroom they wish to study by clicking a button



Data

Analyzing the data from Raspberry Pi with Sound Sensor and POE technology for you.

Check the Noise Levels of the Classroom

The Noise Level: 70
Last Refreshed: 05/06/2016 17:37:43

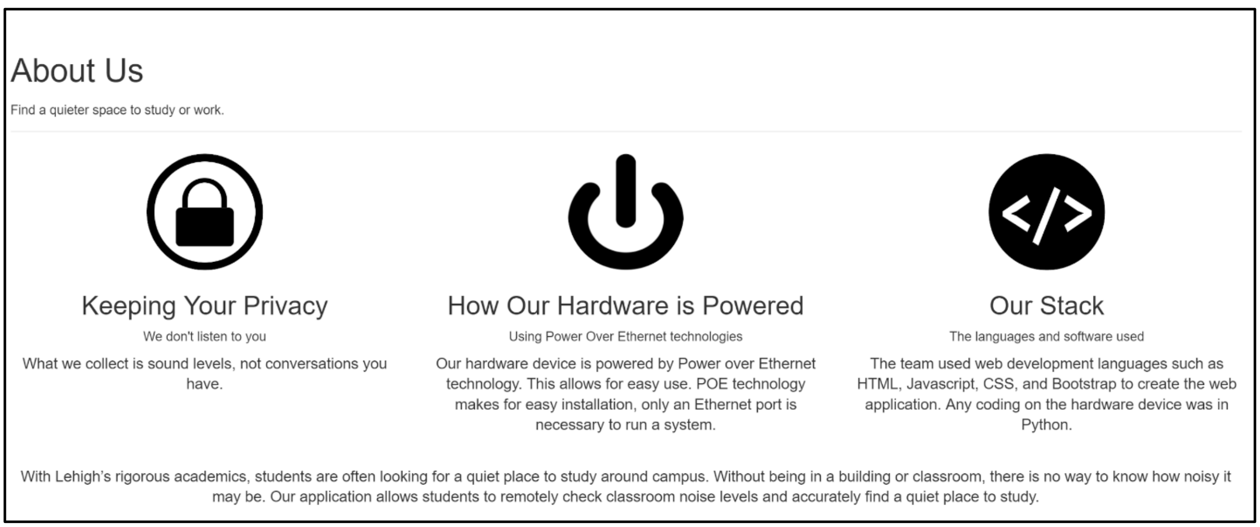
Disclaimer: These are refreshing every 10 minutes automatically. It is averaging all the noise levels of the room in that ten minute interval.

How Noisy is the Classroom?

dB Levels	Common Sounds
20	Whispering
45-56	Library during traffic
53-55	Coke/Vending Machines
55-62	Stair well in Lewis Lab
58-62	Television in Dorm room
65-69	Packard Avenue
74	Snoring Roommate
70-78	Bar during Dinner
82	Driving in car on 378
84-86	Hairdryer
93	Loud Car Radio
100	Party

The table above shows typical decibel (dB) levels you might see above in addition to common sounds comparisons.

User is able to check the noise level of the classroom



About Us

Find a quieter space to study or work.

Keeping Your Privacy

We don't listen to you

What we collect is sound levels, not conversations you have.

How Our Hardware is Powered

Using Power Over Ethernet technologies

Our hardware device is powered by Power over Ethernet technology. This allows for easy use. POE technology makes for easy installation, only an Ethernet port is necessary to run a system.

Our Stack

The languages and software used

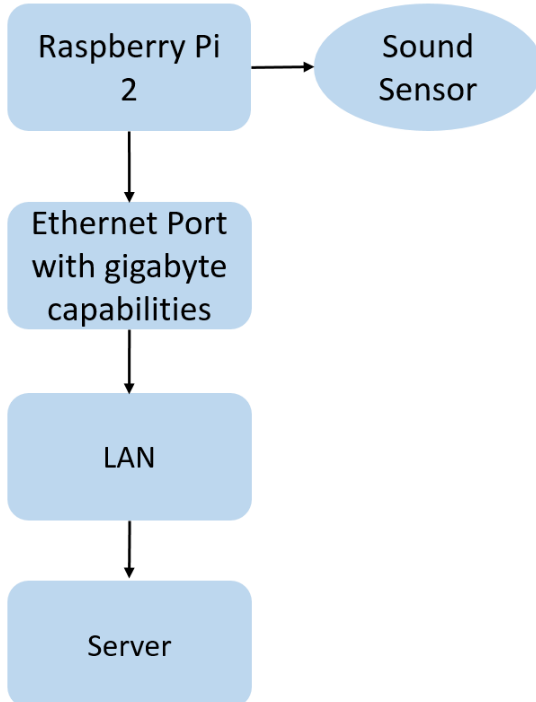
The team used web development languages such as HTML, Javascript, CSS, and Bootstrap to create the web application. Any coding on the hardware device was in Python.

With Lehigh's rigorous academics, students are often looking for a quiet place to study around campus. Without being in a building or classroom, there is no way to know how noisy it may be. Our application allows students to remotely check classroom noise levels and accurately find a quiet place to study.

Information about the web application and our project

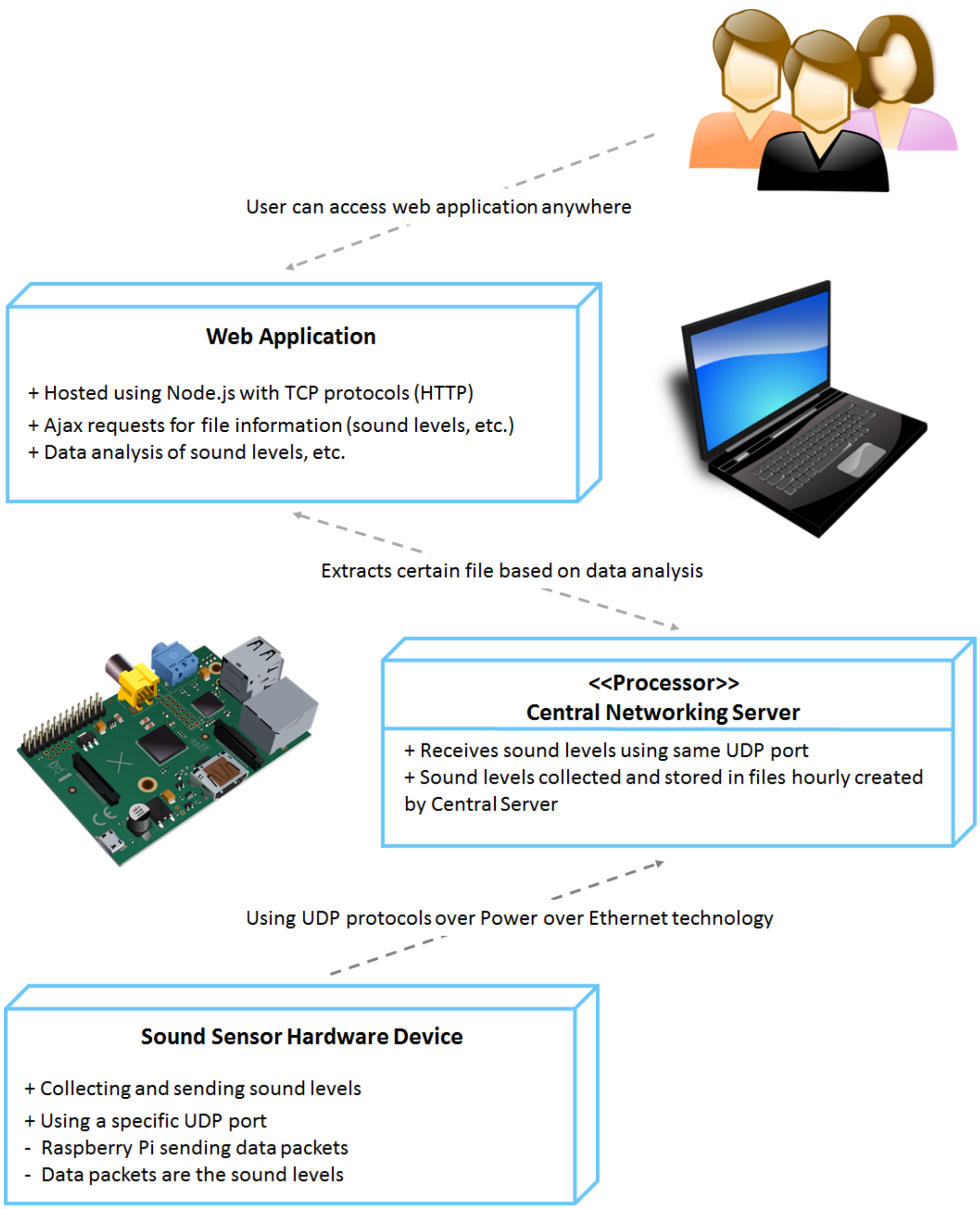
A functional contact us page on the web application

Appendix K: HARDWARE DESIGN



The initial block diagram showing the components of the POE Sound Sensor Device

Appendix L: SOFTWARE DESIGN



Deployment diagram from the POE Sound Sensor device to the Web Application showing the connectivity