GM1 Team Stethoscope Final Report

Lauren Luo

June 1, 2017

1 Overview

1.1 Problem

Stethoscope is not only a symbol of medicine, but also an outdated yet imperative instrument for General Physicians. It heavily relies on the hearing of the treating physician to pick up small irregularities. Procuring second opinions are also difficult. Learning to use a stethoscope for up and coming physician can be challenging, since a student can not hear the sounds passing through the stethoscope along side the instructor. The education method heavily relies on onomatopoeia and years of experience.

1.2 Our Solution

There are pre-existing solutions, such as Littmann Electronic Stethoscope 3100 and Thinklabs One Digital, that amplify, filter, and visually display the audio recordings through wireless technology. We hope to, however, adapt the stethoscope to include features such as: audio sound recorded and filtered, live visual display of the sound for analysis, and have all data integrated into the patient's record through L2S2 for easy access.

Currently, we hope to integrate the electric components as an add-on to the existing stethoscope. This allows GP to keep their long-treasured stethoscope and update their equipment for a cheap price.

The product system break down is shown in Figure 1.

We start off with mechanical integration. The entire product must be small and compact, such that it does not hinder original use of the stethoscope. We also hope to keep it light so that doctors will not feel strained wearing a device around their necks. We designed and made a small box that encloses the microphones used to collect sound data, attached at the end of a stethoscope near the bell. Our design also aides with mechanical filtering of noise. Our mechanical expert, David, worked with our electrical team to design mechanical inserts for various places in the stethoscope, and found the connection between the stethoscope bell and tubes to provide the best signal.

Next up is data collection and data transmission. Our electrical team tested various microphones and locations on the stethoscope to produce the cleanest and clearest sound signal. They also tested transmitting data using different connections: Bluetooth and with cable. Bluetooth, with the module we are given, is only capable of transmitting through one channel. With cable, however, we can transmit data through two channels, and import more information.

Once the sound recordings are transmitted to the computer, we conduct signal analysis on the sound files. There are three major steps in our signal analysis algorithms: noise cancellation of ambient sounds, noise reduction, and feature detection. We incorporated two microphones in our system: one inside the tube, and one outside the tube. This allows us to utilize the outside microphone as a base for noises to cancel out. Luckily, our mechanical design has already allowed massive noise cancellation and fairly clear signal to be picked up by the inside microphone, so no specific algorithm was used for noise cancellation. For noise reduction, we segmented the sound recordings and applied overlapadd synthesis and wiener filtering in the frequency domain. For feature detection, average signal power is computed for small segments of the audio and smoothed over time. The average power is then used as an estimate for a threshold. Signal above the threshold is considered heart activity, and signal below the threshold is considered to be noise. The different parts of a heartbeat, specifically "lub" "dub", are matched with local consecutive peaks in each heart beat. Using these peaks, we can also determine the location of one heart cycle. With some calculations, this produces heartbeats per minute and can help us estimate heart cycle stages.

Once the signal has been processed, we can display visual forms of the signal for a physician to further interpret. As our prototype, we designed a MATLAB GUI that can translate the filtered heart sound recordings into spectrograms, and display it real-time. A graph of the sound waveform with the beginning and end of each heartbeat is overlayed on top for a physician as further aide for the physician to analyze the spectrogram quickly and accurately. In future adaptations, we would like to incorporate O2 saturation, CO2 saturation, and simple 2-node EKG, and lung sound mode. These were identified by a consulting GP as the main information required for a physical to make snap judgments on whether further testing is required for a patient in an emergency. We have designed a user interface that incorporates these future additions as well as past data comparison.

In addition to recording, analyzing, and visualizing the data, we worked with L2S2 to store all data into a central database for future access.

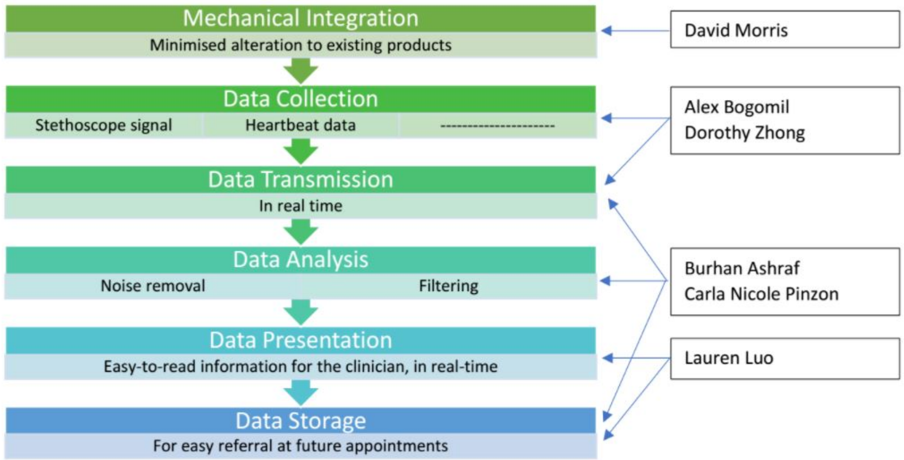


Figure 1: System diagram of product and allocation of work in the team.

2 Data Visualization

My role in the team is to design and create a data visualization interface that can be easily interpreted by users and smoothly integrate the collected data with the L2S2 database.

Data visualization will allow the product to stand out. An important aspect of improving the stethoscope is confirmation and recording of what the doctor is hearing. Out of the five senses in the human body, sight is the most well versed. Humans are well-trained in recognizing patterns and processing information through sight.

We are attempting to offer real-time data visualization and feedback. This allows the doctors to see, and hear, immediately any anomalies in the sound. The doctors can match what they see to what they hear. This also gives the patients some information on what the doctor is hearing, and the doctor can easily pin-point to sources of abnormality. In order to accomplish this, we must develop a user interface that shows the graphs in a easy-to-analyze manner with real-time graphing and data display. The User Interface must also be easy-to-learn and user-friendly, as many doctors are not well-versed in technology.

Ideally, we will offer a real-time update UI and a static UI. The real-time update UI can be used on iPads or phones as well as computers. This lets the doctors to have access to data in crowded emergency rooms. The static UI is useful for physicians who do not have much technological resources. In the case the computer does not have heavy computing powers, the physician can still analyze data after he or she finishes examining the patient.

2.1 Quick-Glance Analysis

Visualizing the sound data allows GPs to look at the vital signs, and quickly, efficiently, and accurately make judgments on whether further, more costly tests is needed. After several interviews with Dr. McLean, we dictated the most useful and vital information were: Heart beats per minute, blood oxygen saturation, carbon dioxide saturation, heart pulse sound from a stethoscope, and a rough EKG. Therefore when designing the overview in the interface, we attempted to incorporate all five sources of data in an easy-to-be-seen manner. Iterations of the Overview UI and be found in Figure 4(d) and Figure 5(a).

2.2 Heart Sound Data

The most efficient and information-ally compact way to represent the sound waves is using a spectrogram. A spectrogram plots time against frequency, with intensity shown in different colors. A spectrogram is obtained by computing fast fourier transform(FFT) on the input signal. Conveniently, Matlab had a "spectrogram" function that takes in an array of audio samples, window size(used to divide audio sample into segments for analysis), number of overlap samples (sample amount which Matlab overlaps with the segment before and after to compute Fast Fourier Transform, must but smaller than window/segment size), frequency of the audio, and axis of display. The array of audio samples and frequency of audio sample can be procured by using "audioread" function on a .wav file. It is important to make sure we take only the first column of the audio array matrix, and "spectrogram" can only analyze signals from one channel. [2]

Of course, not all audio files are best analyzed with the same spectrogram parameters. Depending on the sample frequency and the cleanness of the sound recording, we can adjust the resolution of the spectrogram by trading off time and frequency resolution. As mentioned before, spectrogram cuts the input signal into segments. Longer segments allow better frequency resolution, while shorter segments allow better time resolution. If the signal is strong and has very high power, shorter time segments generates more clarity, especially in locating the signal on a time scale. To make a spectrogram more smooth, increase the overlap. In order to isolate the main features we want to identify, we can add in an additional parameter, "MinThreshold". This parameter essentially creates a threshold, where all signal below the threshold frequency is disregarded. As a result, it highlights the prominent features.

In addition to increasing resolution and applying a threshold, it is also important to maintain an optimal ratio between the sampling frequency and window size of the spectrogram. Ideally, each window size is just large enough to encompass 'a single noise'. As a result, when sampling rate increases, the window size must also increase proportionally to encompass the same 'single noise'. After much testing, we have found the ratio of window size to sampling audio frequency to be 0.023 samples/second.

An example spectrogram plot with sufficient resolution is shown in Figure 2. We can distinctly decipher the diastolic and systolic parts of a heart beat, and the rumbling noise between the two heart sounds.

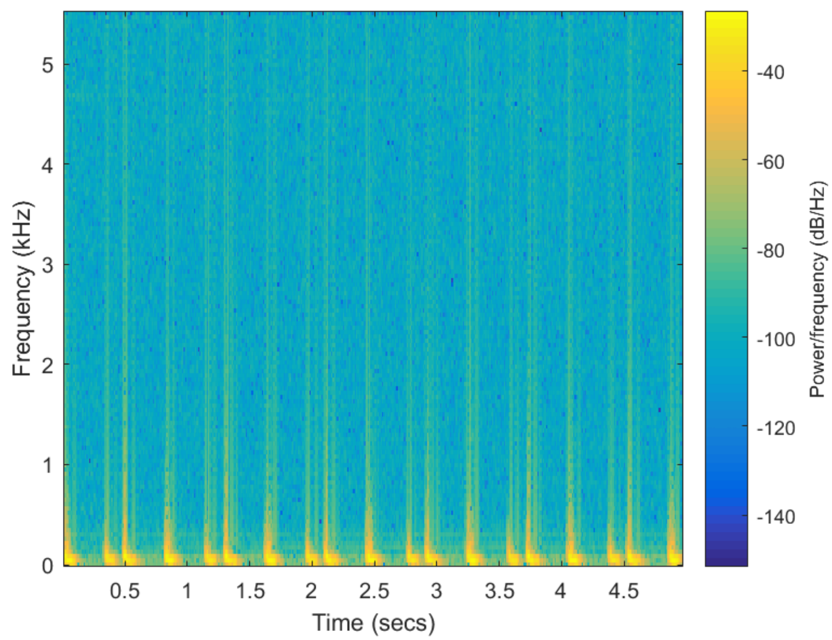


Figure 2: s3 heartbeat present Spectrogram

All in all, after playing around with various parameter for different audio test files, we learned it is essential to give the users of our interface the ability to play with these parameters as well to find the best resolution image and keep a fixed range for the x-axis of the plot. As shown later on in the UI display in Figure 5(b), we incorporated a slider that allows the user to adjust minimum threshold of the viewing spectrogram. After testing with files of various length, the optimal graph display size is 4.5 seconds. If the graph displayed longer than 5 seconds, too much information would be presented, resulting in clusters instead of clear distinct signals. Time range too small did not incorporate enough cycles for spotting abnormalities.

2.3 Lung Sound Data

After a couple interviews with our General Physician consultant Dr. McLean, we learned that the two most important organs GPs use their stethoscope on are the heart and lungs. Although in this iteration of prototype we did not have the time implement lung sound analysis, we did some research on how we would go about the process.

After some research, we found the best way to characterize lung sounds is by expanding a cleaned version of its waveform. Figure 3 clearly displays the different types of lung sounds and how its expanded form is shows the characteristics of each recording.

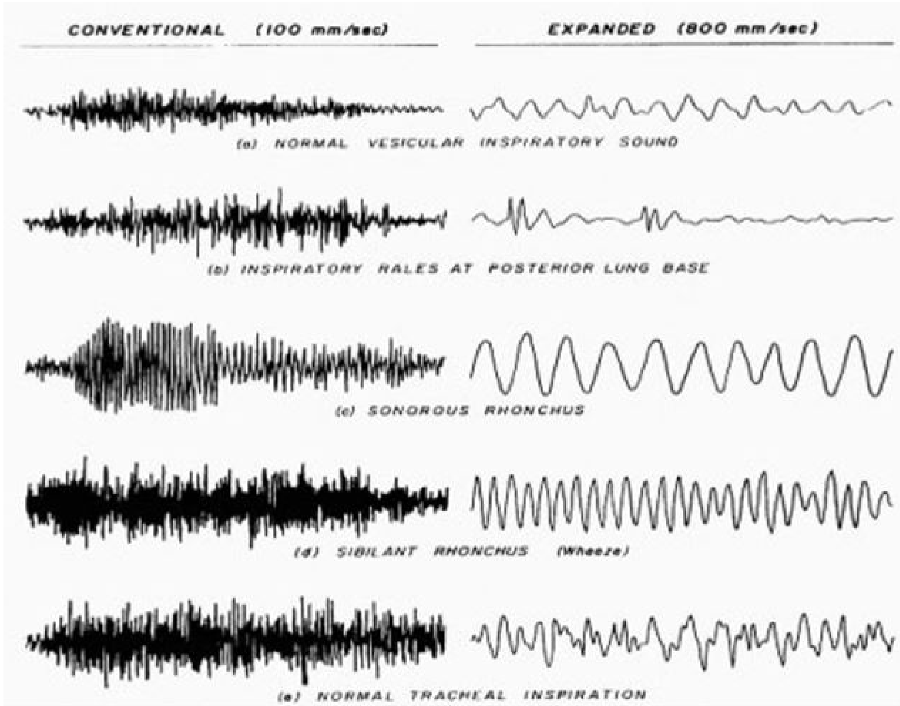


Figure 3: Various lung sound waveform and its expanded waveforms [3]

There are a few potential algorithms that can be applied to our program to help a physician discern key features. For instance, depending on the location where the stethoscope is placed, the sound has unique characteristics. For bronchial lung sound, the sound has a tubular quality. It is generally louder in expiration, and can be heard most clearly over the trachea. For Vesicular lung sounds, it is louder in inspiration as opposed to expiration. For crackles in the lung, the noise will generally be discontinuous, higher pitch, less intense, and have shorter durations. To a certain level, some machine learning and analysis can be done to help categorize sounds heard from the lungs. [3]

For our user interface design, we have incorporated the simplest way of analyzing lung sounds, which was expanded the waveform to more clearly display its sound characteristics. Hopefully with this simple solution, physicians can utilize their knowledge to quickly recognize the pattern, confirm their hearing, and make a diagnosis. The evolution of the UI design for lungs are shown in Figure 4(c) and Figure 5(c).

2.4 Data Comparison

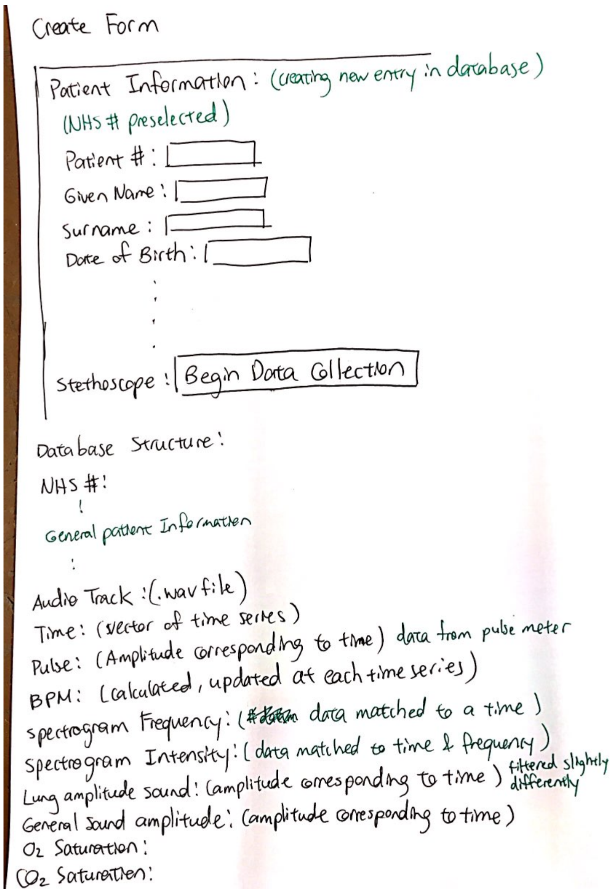
One of the best features of being able to record and stores visual representations of sounds is the ability to compare current data with the past. This gives physicians the ability to keep record of how a patient's conditions have evolved.

Several data comparison methods were considered. One of them was to use a cross power spectral density on the recorded and filtered audio file. Matlab, conveniently, has a function 'cpsd' that produces a cross power spectral density plot instantaneously. The drawback of this method is, for it to work accurately and consistently, much of the data measurement parameters must be consistent. Another method considered would be attempting to envelop detection, finding a way to quantify the "shape" of the signal. However, after some testing with cpsd and further consideration, we found the most efficient, quick, and cheap method of comparing data is by eye. In an technologically out-dated field such as primary medical care, simply the ability to see past data side by side comes a long way.

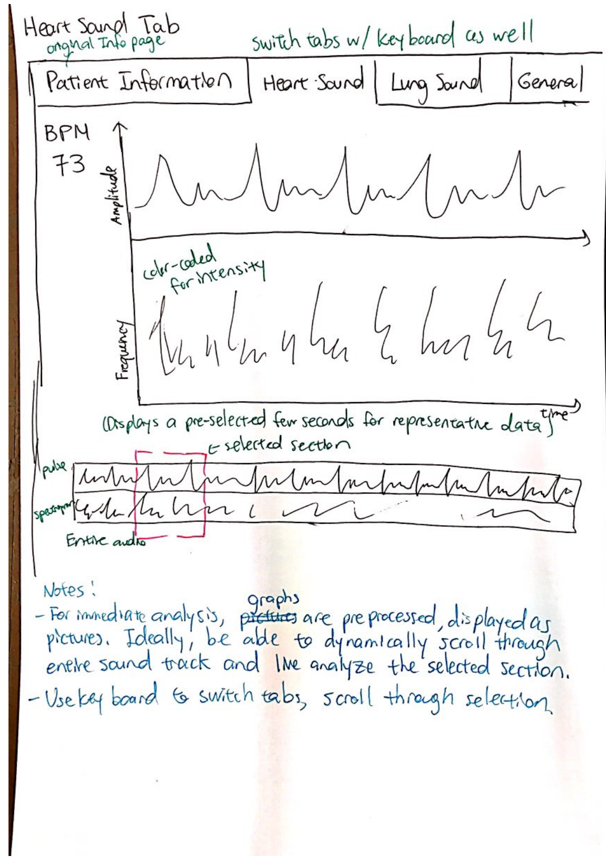
We reflected this feedback in our UI by adding a "Past Data" comparison page, as shown in Figure 5(c) and 5(d). The page simply displays all past dates where related data had been collected. To compare data of various dates, simply select for those dates, and data visuals from each of the selected dates will be displayed on the same screen.

2.5 Designed UI

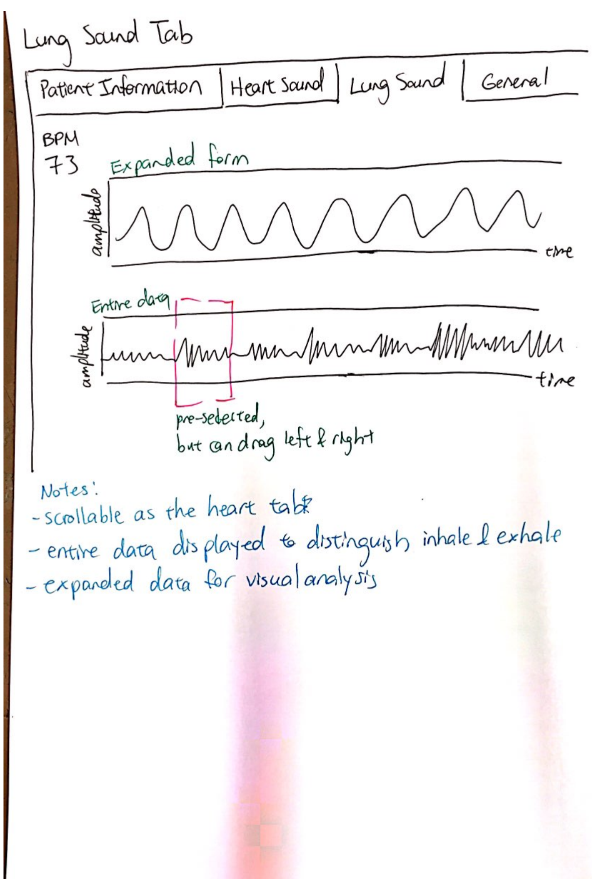
Throughout the project, our design of the idea user interface has evolved. As you can see from Figure 4 to Figure 5, we have added key features such as adjustable analysis parameter sliders, data comparison tabs, an overview that utilizes more sensor inputs, and upload and download buttons. Most of the features mentioned are morem carefully explained in sections above. Upload button is used to upload data into the L2S2 central database. Download button is for downloading files onto the current computer. These two button allows more flexibility in data storage and transfer.



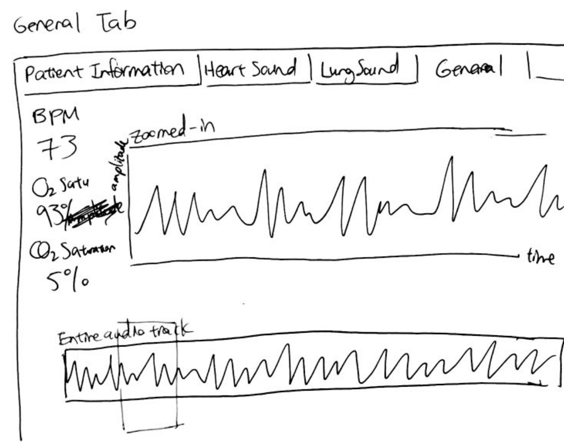
(a)



(b)

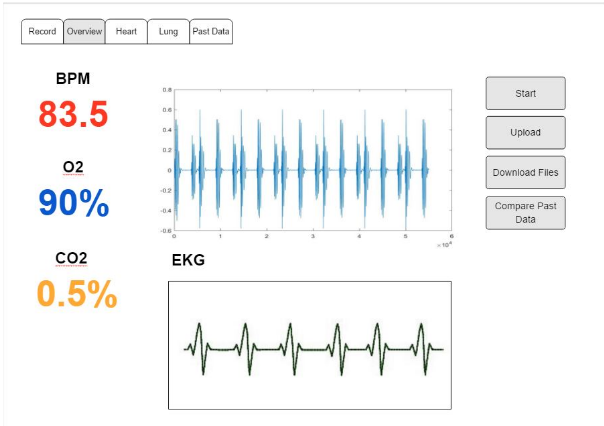


(c)

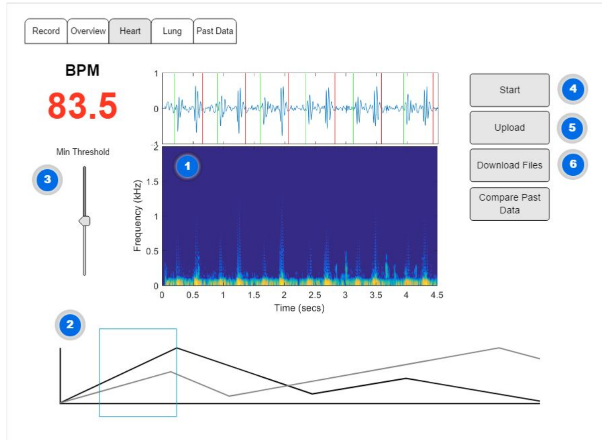


(d)

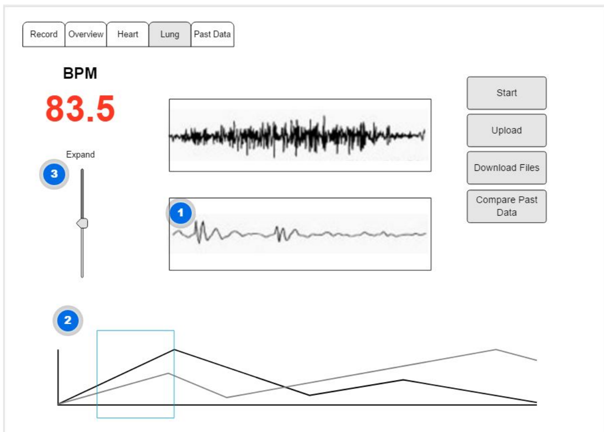
Figure 4: Hand drawn user interfaces



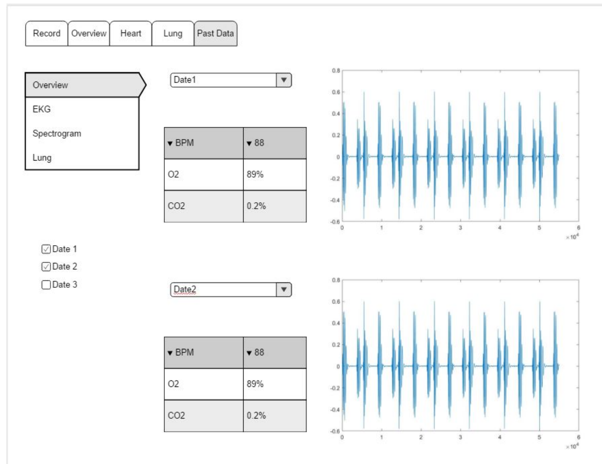
(a)



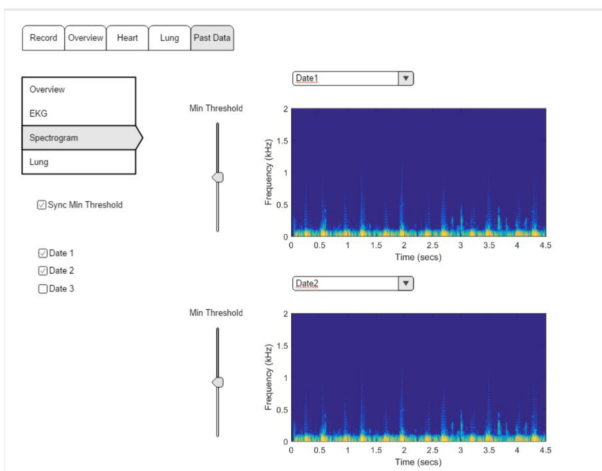
(b)



(c)



(d)



(e)

Figure 5: Web-based drawn user interface design.

Some additional notes to the UI design:
Figure 5(b)

- 1. The Spectrogram is a 3D graph, and can be zoomed in with a mouse or cursor
- 2. A representation of the entire audio recording. The selected section is the section displayed in the graphs above.
- 3. The min. threshold slider allows adjustments to spectrogram for a higher resolution graph.
- 4. Start button should be pressed to start data collection. It is there as a safety mode, to prevent excess data collection.
- 5. Upload button uploads recorded sound file, analyzed sound file, heart pulse graph, spectrogram, and heart beats per minute into L2S2 database.

- 6. Downloads the files listed above onto the current workstation.

Figure 5(c)

- 1. The expanded graph is zoomable with a mouse and/or cursor.
- 2. A representation of the entire sound recording. Selected section is displayed in the graphs above.
- 3. the 'Expand' slider controls how much of the x-axis is expanded to allow flexibility for optimal analysis.

3 Progress

The past section described mostly our idealized user interface and its key features. In this prototype, I was able to integrate with the signal analysis team to produce a MATLAB GUI that, given an heart audio file, real-time filters and graphs the spectrogram for analysis. When the recording is finished, feature detection is applied and overlay on top of the spectrogram, clearly displaying the beginning and end of each heart beat cycle. The GUI is shown in Figure 6, and the code for developing the MATLAB GUI is in the stored team USB.

The biggest challenge coding the GUI was making it real-time without significant delay, and consistently display only 4.5 seconds of all the data. As the incoming signal was analyzed in small segments, re-graphing each small segment is very computationally consuming. It was important to find the optimal rate and method to re-graph the spectrogram without overloading the program. In addition, since data came in segments, it was challenging coming up with an efficient method of data storage for keeping the already analyzed segments, but hide the analyzed data that does not fit in the time window.

Our solution was to create an original array of zeros that popular enough data to fill from 0 to 4.5 seconds, similar to equation 1. Once segments of data is streamed in, I append the new data to the end of the array, and take out enough data from the beginning of the array such that the length of the array stays constant, as shown in equation 2. Another variable outside the segment generating loop keeps track of all the filtered segments and stitches it back into one array, which will be returned as a process .wav audio file.

What helped with graphing real-time was adding a "pause" function after each segment was appended. This allots time for the function to process. Although the pause causes a slight delay, the delay is only 20ms, so it is barely noticeable.

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]   (1)
[0 0 0 0 0 0 0 0 0 0 0 0 [segment]]   (2)
```

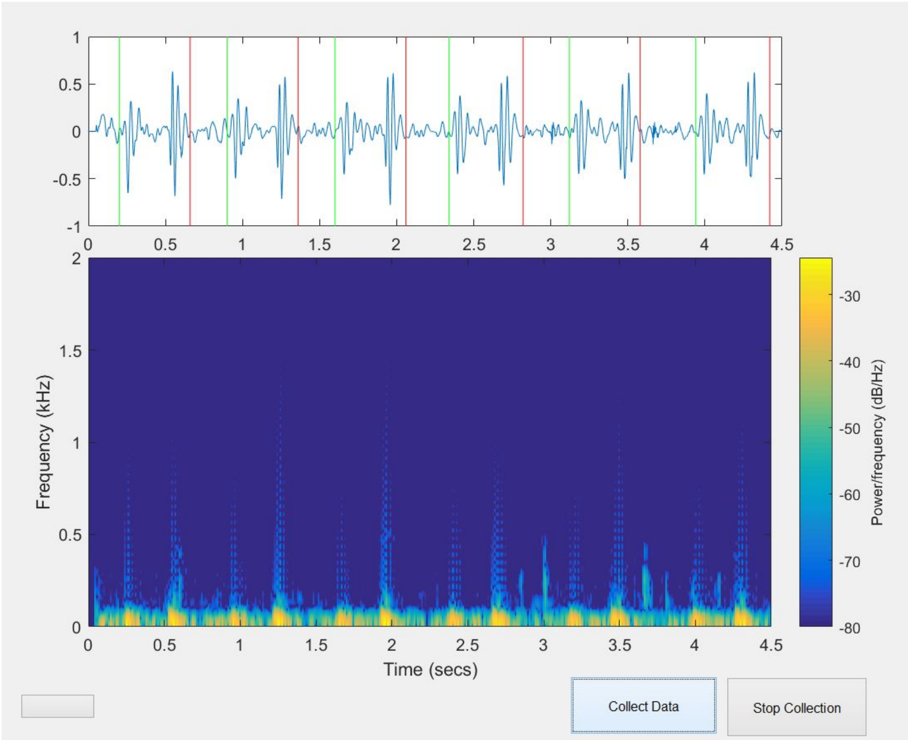


Figure 6: Matlab GUI prototype

I am currently attempting to communicate with Dr. Gaffney on potential GUI design interfaces that can be incorporated in L2S2. I have designed and drew rough sketches of potential interfaces as well as the data structure that would be incorporated into L2S2 patient database.

4 Data Storage

As I mentioned throughout the report, one of the key features of our product is the ability to store and extract from a central database, L2S2. This capability not only allows physicians to keep record of all the analysis, but also compare current data with past data. Our team had some issues in communication with the L2S2 team, therefore we only had 3 days to learn about the Web-based L2S2 API, and how to use a SOAP client to use the API. I only had enough time to learn to create a new user, record, and automate uploads of analyzed data into the database, but not necessarily with patient record association. However, we were able to have a plate template designed and made, where we can manually add in the files produced from MATLAB code, and upload it into the system. The plate, as seen on the L2S2 system, is shown in Figure 7. The manual method does associate all of uploaded data to a specific patient and date. Given more time and guidance, we would be able to automate the process.

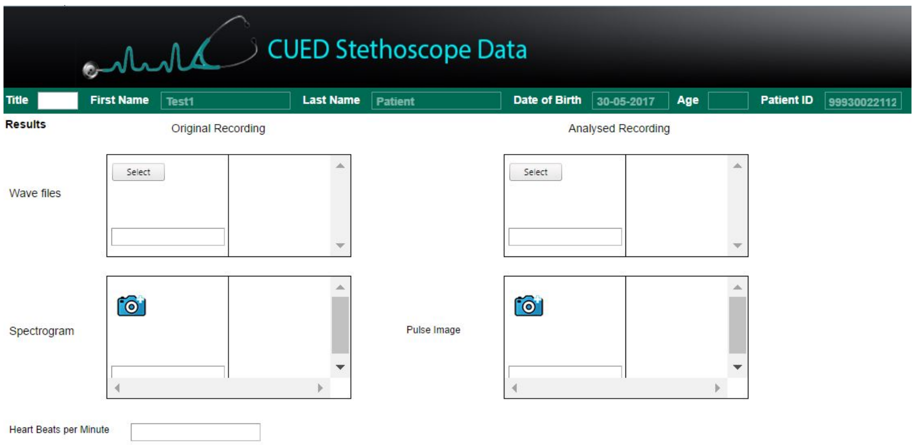


Figure 7: L2S2 Plate

5 Conclusion and Learning Process

This project has a lot of potential for further development. With more time and resources, we can work closely with L2S2 to develop a more sophisticated user interface, potentially embedded into their system. We can also look more into machine learning methods of detecting heart sound patterns to give instant diagnosis. Incorporating more sensors will also contribute greatly.

Overall, this project has been a learning experience. In the technical aspects, I researched various ways of analyzing heart and lung signals, learned in more details about the workings of a spectrogram, how to code a Matlab GUI, how to code a graph to display real-time data, and how to use python to interact with a SOAP client API. In terms of presentation, I gained with user interface and presentation design, paying attention to details such as font and visibility. As far as product design and development process goes, I learned the importance of testing, rapid prototyping, and iterations. Before we can go very deep into the project, it was crucial for us to test various equipment and various locations on the stethoscope where our product would produce the best results.

I learned to communicate better in a team: it is crucial to be patient and communicate your specific wants and needs. Everyone comes from a different background and have different areas of expertise. When you are able to communicate your knowledge at a very basic level, such that your teammates can understand, it helps to keep your team on the same page. Looking back, I enjoyed working with my teammates and found this project a productive and learning experience.

6 References

[1]Documentation. (n.d.). Retrieved May 31, 2017, from
https://uk.mathworks.com/help/signal/examples/practical-introduction-to-time-frequency-analysis.html

[2]Mathworks. (n.d.). Retrieved May 31, 2017, from
https://uk.mathworks.com/help/signal/ref/spectrogram.html

[3]Stethographics. (n.d.). Retrieved May 31, 2017, from
http://www.stethographics.com/mainphysiology_ls_introduction.html