Project ID: YXM AAVV

Postoperative Telemonitoring of Flap Physiology in Microvascular Free Tissue Transfer

A Major Qualifying Project Report submitted to the faculty of

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the degree of Bachelor of Science

Submitted by:

____

Dennis Giaya

____

Rebeca Stolarczyk

April 28, 2016

____

Professor Yitzhak Mendelson, Ph.D., Advisor

Department of Biomedical Engineering

Dr. Raymond Dunn, M.D., UMass Memorial Medical Center

Dr. Samandar Dowlatshahi, M.D., UMass Memorial Medical Center

Table of Contents

Table of Contents

Table of Figures ... v

Table of Tables ... viii

Table of Symbols & Abbreviations ... ix

Authorship Page ... x

Acknowledgements ... xi

Abstract ... 1

Chapter 1: Introduction ... 2

Chapter 2: Literature Review ... 6

2.1 Microvascular Free Tissue Transfer ... 6

2.2 Flap Failure ... 7

2.3 Flap Salvage ... 8

2.4 Standard Postoperative Monitoring Practices ... 9

2.5 Alternative Postoperative Monitoring Methods ... 11

2.5.1 Temperature ... 11

2.5.2 Doppler Ultrasound Flowmetry ... 13

2.5.3 Laser Doppler Flowmetry ... 15

2.5.4 pH ... 16

2.5.5 Oxygenation ... 20

Chapter 3. Project Strategy ... 24

3.1. Initial Client Statement ... 24

3.2 Previous MQP ... 24

3.3 Design Requirements: Technical ... 25

3.3.1 Objectives ... 25

3.3.2 Constraints ... 29

3.4 Design Requirements: Standards ... 31

3.5 Revised Client Statement 33
3.6 Project Approach.. 34
Chapter 4: Alternative Designs 37
4.1 Functions and Specifications.. 37
4.1.1 Acquire Temperature and Doppler Ultrasound Data... 37
4.1.2 Algorithms to Indicate Vessel Occlusion 42
4.1.3 Transmit the Information Wirelessly to a Database 44
4.1.4 User Interface 47
4.1.5 Alert the User of Occlusion 51
4.2 Preliminary Testing 52
4.2.1 Ultrasound Gel Longevity 52
4.2.2 Thermistor Calibration 53
4.3.3 Preliminary Doppler Ultrasound Data 55
Chapter 5: Design Verification 65
5.1 Doppler Ultrasound 65
5.1.1 Controlled Study.. 65
5.1.2 Patient Trial 70
5.2 Temperature 71
5.3 Color... 72
Chapter 6: Final Design and Validation... 74
6.1 Final Design 74
6.1.1 Temperature... 74
6.1.2 Probe Holder.. 74
6.1.3 Color 75
6.1.4 Case 76
6.1.5 Website 78
6.1.6 Doppler ultrasound 82
6.1.7 System 83

6.2 Impact 83
6.2.1 Economics 83
6.2.2 Environmental Impact 84
6.2.3 Societal Influence 85
6.2.4 Political Ramifications 85
6.2.5 Ethical Concerns 86
6.2.6 Health and Safety Issues 87
6.2.7 Manufacturability 88
Chapter 7: Discussion 90
Chapter 8: Conclusions & Recommendations 93
References 96
Appendix A: Device Code 99
A.1 Arduino Code 99
A.2 patientmonitor.h 110
A.3 patientmonitor.cpp 112
A.4 arduinoInterface.cpp 114
A.5 webInterface.cpp 118
A.6 curlFunctions.cpp 120
A.7 dopplerFunctions.cpp 126
Appendix B: SolidWorks Diagram of Device Case 128
Appendix C: SolidWorks Diagram of Probe Holder 129
Appendix D: Matlab Signal Processing Code 130
Appendix E: Website Code 132
E.1 index.php 132
E.2 dashboard.php 136
E.3 patients.php 139
E.4 patient.php 147
E.5 addPatient.php 159

E.6 removePatient.php 163
E.7 devices.php 168
E.8 device.php 175
E.9 settings.php 201
E.10 resetPassword.php 206
E.11 reset.php 210
E.12 logout.php 214
E.13 stylesheetNew.css 215
E.13 retrieve.php 219
E.14 retrieveStatus.php 220
E.15 temperature.php 221
E.16 updateStatus.php 222
E.17 last.php 223

Table of Figures

Fig. 2.1 Thermal Imaging in Free Tissue Transfer. Thermal images of pedicle flap in rat highlight changes at different elevations and the sensitivity of thermal imaging [18]. 13

Fig. 2.2 Laser Doppler flowmetry in rabbit ear model [26]. 16

Fig. 2.3 pH drop in rat skin flaps subject to arterial and venous clamping [28]. 18

Fig. 2.4 StO_2 in arterial compromise [33]. 22

Fig. 2.5 StO_2 in venous compromise [33]. 22

Fig. 2.6 T.Ox Tissue Oximeter by ViOptix [35]. 23

Fig. 3.1 LabView VI developed by previous MQP for recording Doppler signal. 25

Fig. 3.2 Objectives tree. 26

Fig. 3.3 Work breakdown structure. 35

Fig. 3.4 Gantt chart. 36

Fig. 4.1 Ultrasound probe front view. 39

Fig. 4.2 Ultrasound probe side view. 40

Fig. 4.3 Thermistor side view. 40

Fig. 4.4 Images from SolidWorks model of rectangular suturable probe holder. 41

Fig. 4.5 Login page of GUI. 48

Figure 4.6 Reset password page of GUI. 49

Fig. 4.7 Patient summary page of GUI. 49

Fig. 4.8 Patient monitoring page from GUI. 50

Fig. 4.9 Add patient form from GUI. 50

Fig. 4.10 Device control panel from GUI. 51

Fig. 4.11 Ultrasound gel longevity test setup. 52

Fig. 4.12 Thermistor calibration curve. 54

Fig. 4.13 Calibration of four fully assembled temperature probes. 55

Fig. 4.14 Radial artery spectrogram 0mmHg. 56

Fig. 4.15 Radial artery spectrogram 20mmHg. 57

Fig. 4.16 Radial artery spectrogram 40mmHg. 57

Fig. 4.17 Radial artery spectrogram 60mmHg. 58

Fig. 4.18 Radial artery spectrogram 80mmHg. 58

Fig. 4.19 Radial artery spectrogram 100mmHg. 59

Fig. 4.20 Radial artery spectrogram 120mmHg. 60

Fig. 4.21 Radial artery spectrogram 140mmHg. 60

Fig. 4.22 Radial artery power window 0mmHg. 61

Fig. 4.23 Radial artery power window 20mmHg. 61

Fig. 4.24 Radial artery power window 40mmHg. 62

Fig. 4.25 Radial artery power window 60mmHg. 62

Fig. 4.26 Radial artery power window 80mmHg. 63

Fig. 4.27 Radial artery power window 100mmHg. 63

Fig. 4.28 Radial artery power window 120mmHg. 64

Fig. 4.29 Radial artery power window 140mmHg. 64

Fig. 5.1 Setup for controlled study acquiring Doppler ultrasound and color signals. 66

Fig. 5.2 Subject radial power signal at 0mmHg. 67

Fig. 5.3 Subject radial power signal at 40mmHg. 67

Fig. 5.4 Subject radial power signal at 80mmHg. 68

Fig. 5.5 Subject radial power signal at 120mmHg. 68

Fig. 5.6 Subject radial power signal at deflation. 69

Fig. 5.7 Subject radial power signal after deflation. 69

Fig. 5.8 Power signal from healthy subject. 70

Fig. 5.9 Power signal from postoperative patient. 71

Fig. 5.10 Absorption of white, red, green, and blue light during simulated venous and arterial compromise. 73

Fig. 6.1 Thermistor circuit schematic to Arduino analog input pins. 74

Fig. 6.2 Combination probe holder with flat Doppler probe, thermistor, and ultrasound gel catheter fixed in place. 75

Fig. 6.3 TCS34725 RGB color sensor with white LED. 76

Fig. 6.4 Bottom piece of device case in SolidWorks with mounting for Arduino, Raspberry Pi, and breadboard. 76

Fig. 6.5 Top piece of device case in SolidWorks with holes for electrical components and Doppler panel. 77

Fig. 5.6 Final bedside device with integrated Doppler flow meter and components.......... 78

Fig. 6.7 Login screen on mobile user interface.......... 79

Fig. 6.8 Patient page with dismissible patient alerts of varying priorities displayed.......... 80

Fig. 6.9 Temperature data from website graph with curves from all four probes. 80

Fig. 6.10 Doppler ultrasound audio player and visualizer. 81

Fig. 6.11 Control panel on mobile interface. 81

Fig. 6.12 Flap probe multiplexor. 82

Fig. 6.13 Overview of completed system. 83

Table of Tables

Table 2.1 pH Drop in skin flaps after vascular compromise. ……………………………….. 17

Table 4.1 Evaluation of probe holder designs. 42

Table 4.2 Ultrasound gel longevity test results…..53

Table 5.1 Temperature calibration statistical analysis at 34ºC (N=300)….……………………..72

Table 5.2 Temperature calibration statistical analysis at 38ºC (N=300)..………..……………..72

Table 5.3 Temperature calibration statistical analysis at 42ºC (N=300)..……..………………..72

Table of Symbols & Abbreviations

ATP: Adenosine triphosphate

ISFET: Ion sensitive field effect transistors

SaO_2: Amount of oxygenated hemoglobin in blood

SpO_2: Estimate of arterial oxygen saturation

StO_2: Tissue oxygen saturation

VI: Virtual instrument

FDA: Food and Drug Administration

ISO: International Organization for Standardization

HIPAA: Health Insurance Portability and Accountability Act

GUI: Graphical user interface

WAV: Waveform audio file format

mmHg: Millimeter of mercury

Authorship Page

Abstract [Dennis and Rebecca]

Chapter 1: Introduction [Dennis and Rebecca]

Chapter 2: Literature Review [Dennis and Rebecca]

2.1 Microvascular Free Tissue Transfer [Rebecca]

2.2 Flap Failure [Rebecca]

2.3 Flap Salvage [Rebecca]

2.4 Standard Postoperative Monitoring Practices [Dennis and Rebecca]

2.5 Alternative Postoperative Monitoring Methods [Dennis]

Chapter 3. Project Strategy [Dennis and Rebecca]

3.1. Initial Client Statement [Dennis]

3.2 Previous MQP [Rebecca]

3.3 Design Requirements: Technical [Dennis and Rebecca]

3.4 Design Requirements: Standards [Dennis and Rebecca]

3.5 Revised Client Statement [Dennis]

3.6 Project Approach [Dennis]

Chapter 4: Alternative Designs [Dennis and Rebecca]

4.1 Functions and Specifications [Dennis and Rebecca]

4.2 Preliminary Testing [Dennis and Rebecca]

Chapter 5: Design Verification [Dennis and Rebecca]

5.1 Doppler Ultrasound [Dennis and Rebecca]

5.2 Temperature [Dennis]

5.3 Color [Rebecca]

Chapter 6: Final Design and Validation [Dennis and Rebecca]

6.1 Final Design [Dennis and Rebecca]

6.2 Impact [Dennis and Rebecca]

Chapter 7: Discussion [Dennis and Rebecca]

Chapter 8: Conclusions & Recommendations [Dennis and Rebecca]

Acknowledgements

First, the authors wish to thank their advisor, Professor Yitzhak Mendelson, Ph.D., without whom this project would not have been possible. Professor Mendelson provided technical insight and ongoing support over the course of the project. Additionally, he offered feedback on numerous drafts of the report, poster, and presentation.

The authors would also like to thank Dr. Raymond Dunn and Dr. Sammy Dowlatshahi from the Division of Plastic Surgery at UMass Memorial Medical Center. Dr. Dunn and Dr. Dowlatshahi offered the original concept for this project and helped to refine it over the course of the project. They also provided medical insight that was necessary for development of the device. The authors also would like to thank the nurses and staff in the Division of Plastic Surgery for their support and cooperation over the course of testing the device.

Furthermore, the authors wish to thank the volunteers who participated in the study for their time. Also, they would like to thank John Lovaasen for assisting with 3D printing resources for the probe holders and device case. Additionally, they wish to thank Kryngle Daly for providing server space to host and store the website and data. Finally, they would like to thank Lisa Wall, the lab manager at Worcester Polytechnic Institute, for her assistance with ordering and obtaining components for the device and providing us with all of the necessary tools to carry out the project.

Abstract

Current methods of postoperative monitoring during the recovery stage of microvascular free tissue transfer often prove inadequate in circumstances when complications arise. Acknowledging the room for improvement, a novel system for remote and continuous monitoring of temperature and Doppler ultrasound signals was developed. The system collects data from the transferred tissue flap, analyzes the data using novel signal-processing algorithms, and wirelessly uploads the results to a server where they can be viewed from any mobile device. The system was tested on both healthy subjects in a controlled study and a patient recovering from a bilateral breast reconstruction. The final system is capable of measuring temperature within a range of 30-42°C±0.1°C, recording 44.1kHz, 16bits Doppler ultrasound signal, and uploading the data to a password protected, web-based application. Results suggest the novel algorithms can detect venous and/or arterial occlusion in the transplant from analysis of the Doppler signals alone. Future work will focus on refining the algorithms and integrating color sensing as an additional means of monitoring.

Chapter 1: Introduction

Microvascular free tissue transfer is a type of reconstructive surgery in which a body part, organ, or flap (ie. tissue) is transplanted via detachment of blood vessels at the donor site and re-suturing of vessels at the recipient site. The procedure, which is highly intensive and technically demanding, can be performed as a means of either aesthetic or functional reconstruction. Despite the relatively high success rate of free tissue transfer, failure due to late detection of postoperative complications occurs approximately 5% of the time [1].

Issues that arise and persist for an extended period of time while the patient is recovering can ultimately lead to flap failure, in which the tissue dies and is no longer viable [2]. Blood clot formation is a common complication, occurring in 10-12% of all free flap transplantations [1]. The risk of thrombosis is highest immediately after surgery and can lead to tissue death if left untreated. In order to detect blood clot formation as soon as possible, patients are carefully monitored in the intensive care unit (ICU) for the first 72 hours after surgery [3]. During this time, a nurse performs frequent examinations to evaluate the patient's status and check for signs of flap failure. The color, turgor, and temperature of the transplanted flap are manually determined by the nurse and assessed for abnormalities. Capillary refill and prick tests are also performed to evaluate blood flow to and from the tissue. In the instance of any physiological irregularities, the nurse alerts the surgeon who in turn provides the patient with the necessary treatment or returns the patient to the operating room to correct the issue [4].

Rapid and accurate detection of blood clot formation is critical to recovering tissue and preventing flap failure. Current methods for evaluating flap recovery are both delayed and subjective, relying solely on noninvasive examinations to identify complications occurring within the tissue. Because the physical symptoms of a blood clot may not appear immediately, a clinical evaluation may not detect an occlusion until after the tissue has begun to die.

Acknowledging the room for improvement within the current methods of postoperative free flap monitoring, the team has designed a monitoring and alert system that continuously measures parameters such as temperature, Doppler ultrasound, and color in order to wirelessly alert medical staff to the earliest signs of flap failure. The device is “simple and harmless to the patient and free flap; rapid, repeatable, reliable, recordable, and rapidly responsive; accurate and inexpensive; objective and applicable to all kinds of flaps; and finally, equipped with a simple display or system to alert relatively inexperienced personnel to the development of circulatory impairment.” Creech and Miller outlined these criteria in their paper on the evaluation of circulation in skin flaps [5].

The device is highly advantageous in two areas. The first main advantage is earlier detection. By monitoring the flap continuously, problems can be addressed as they occur. The current protocol at many hospitals is to monitor the patient every hour for the first twenty-four hours, followed by every two hours for the next 48 hours, and finally every four hours until discharge [3]. The described device eliminates the potential time wasted between the onset of a problem and its detection due to infrequent sampling. The second major advantage of the device is an objective interpretation. The subjectivity of the current practices leads to indecisive action because one person’s judgment on the color of a compromised flap may be different from another person’s. Human eyes and ears are not sensitive enough to pick up the subtle details that indicate early signs of flap compromise. Analysis of the data collected can objectively determine whether sufficient criteria have been met to indicate a definitive need for re-exploration of the flap. As such, if the device is successful and reliable in early detection of circulatory problems within flaps then it would no longer be necessary to send medical staff to the patient in order to perform monitoring. This frees up nursing staff to tend to other patients more often and also raises the potential of transferring the patient out of the ICU sooner. The frequency at which nurses must check the condition of the flap requires that the patient be in the ICU. Automating this process with the device could allow for the patient to be transferred to a general floor, which would significantly reduce costs. By satisfying both of these areas the device has potential to become the new gold standard in free flap monitoring.

The device consists of three main stages that differ by function. First are the sensors that interface with the human body at the site of the flap to collect physiological data. The team incorporated existing sensors that could be minimally modified for use in the desired flap monitoring applications. The sensors interface with the second major portion of the device, the body, which is where the data analysis and transmission occurs. The body houses an Arduino Mega board for the purpose of interfacing with all of the sensors and providing analog to digital conversion for the signals. The Arduino communicates with a Raspberry Pi unit that processes the signals to determine whether criteria to raise an alarm have been met and also uploads the information over the Internet to a secure MySQL database for access from the third stage of the device: the application. The web-based application allows surgeons to log in from any mobile device with internet access and view real time data for a patient.

Early development of the device consisted of selecting the appropriate sensors and methods to unobtrusively attach them to the flap and record for extended periods of time. The sensors were evaluated for biocompatibility, sensitivity, and the ability of their target measurand to detect early signs of circulatory insufficiency. This was to ensure that they would be effective in the final device and also that they would not cause any harm to the patient. The primary challenge for the body of the device was to be capable of automatically detecting an insufficiency and initiating the process of alerting medical staff. This required the sensors and framework for data acquisition to be operational so that real data could be acquired for analysis and development. This was completed in tandem with the website application, which effectively delivers the information in a way that is visually appealing and simple to understand. Altogether the device meets and exceeds the standards for a suitable free-flap monitoring device.

Chapter 2: Literature Review

2.1 Microvascular Free Tissue Transfer

Microvascular free tissue transfer is a reconstructive surgical procedure in which a body-part, an organ, or a piece of tissue is transplanted from one area of the body to another. The high success rates of this procedure, along with the superior functional and aesthetic outcomes, have made it the gold standard for a number of surgical reconstructions, including those of the breast, head, and neck [6].

Patients generally undergo free tissue transfer as a means of either aesthetic or functional reconstruction. The former includes procedures that are designed for cosmetic purposes, including post-mastectomy breast reconstruction [7] and facial reconstruction after burns or disfigurement [8]. Conversely, functional reconstruction is needed in cases of trauma that leave the patient with a loss of functionality at the given site. These tissue transfers often include muscle transplantation in order to restore function at the defective location. For example, functional reconstruction may target trauma wounds of the face [9] or a lower extremity [10].

Microvascular free tissue transfer makes use of vascular anastomosis, a surgical technique in which two blood vessels are connected via suturing of the two open ends. In free tissue transfer, the artery and vein of the transplanted flap are anastomosed to the vessels at the recipient site. Initially, the flap is perfused through the pedicle vessel until endothelialization is complete after the first two weeks. As a result, risk of flap compromise is greatest immediately after surgery when there is a sole means of perfusion for the newly transplanted tissue [11].

2.2 Flap Failure

Over the past two decades, the overall success rate of microvascular free tissue transfer has reached the range of approximately 95-99%, making it an effective and reliable option for surgical reconstruction [2]. However, the occurrence of postoperative complications is a prevalent issue that can severely compromise the viability of the flap if left untreated. Complete flap failure, which refers to failure of the tissue on the microvascular level, is the most severe of these outcomes. At this stage, vascular anastomosis has essentially failed and the flap is no longer viable. Less common is partial flap necrosis, which generally occurs as a result of flap planning errors or an underlying hematoma exerting pressure on the flap.

The risk for flap failure is the greatest in the first 48 hours after surgery. There are a number of factors that may put a patient at greater risk of failure than others during this timeframe. These can further be separated into two categories of risk factors: patient-related and surgery related. The former includes both natural and behavioral factors that predispose the patient to vascular complications that could indirectly lead to failure. For example, patients with diabetes have a greater risk of peripheral-vascular disease, which calls for supplemental vascular evaluation post-surgery. Additionally, behavioral factors such as tobacco and alcohol use are associated with vascular spasms and acute alcohol withdrawals, respectively, which put the patient at increased risk of flap failure. Surgery-related factors are those associated with the flap transplant itself. One of the most critical factors in ensuring the success of the operation is the quality of vessels at the recipient site. The artery should be smooth with pulsatile red blood flow, and the vein should be compliant with centripetal blood flow. Another important surgical consideration is the ischemic toleration level of the flap. For example, skin flaps are relatively tolerant of ischemia, whereas muscle flaps are less tolerant and have a lower salvage rate in the instance of any complications. Finally, recipient sites with increased incidence of complications, such as lower limb sites and poorly vascularized sites, can put the patient at greater risk of flap failure [2].

Thrombosis remains the most common cause of flap failure, occurring in approximately 10-12% of all free tissue transfers [1]. Venous occlusion is more prevalent than arterial, although arterial compromise can be significantly more problematic [12]. The salvage rate occluded flaps ranges from 28-90% and depends greatly on origin of thrombosis as well as the risk factors discussed previously [6]. Technical errors are a common cause of occlusion, especially those associated with vascular anastomosis, flap design, and tissue elevation. Compressive forces on the vascular pedicle as a result of haematoma or tight closure can also lead to thrombosis.

Haematoma in the absence of thrombosis is the second most common cause of flap compromise, occurring in approximately 2-4% of all free flap surgeries [13]. This occurs when blood pools outside of the vessels in the tissue itself, causing swelling and other complications. Haematoma-related flap failure that occurs in the absence of vascular pedicle occlusion is distinctly different than pressure-induced thrombosis. In this case, the mechanisms of flap necrosis are biochemical in nature rather than mechanical. There are a number of proposed chemical and cellular processes that may be responsible for subsequent tissue ischemia, including the release of proteolytic enzymes and reactive oxygen species as a result of platelet degradation in the tissue [13].

2.3 Flap Salvage

In most cases of postoperative complications, flap salvage is possible with rapid detection and immediate re-exploration of the compromised tissue. Salvage rates for flap can vary significantly depending on the source of the complication. For example, a study of 2372 head and neck free tissue transfers found that thrombectomy after venous occlusion was successful 60% of the time, whereas arterial thrombectomy had a success rate of only about 15% [14]. Additionally, the location of the tissue can significantly impact the salvage rate of the flap. Another study of 716 free tissue transfers reported a greater failure rate for buried flaps (7%) versus non-buried flaps (2%) due to inadequate postoperative monitoring of buried flaps [15].

It is critical that flap compromise is detected as early as possible for the best chance of salvaging the tissue. In general, the salvage rate is much lower for flaps that undergo late re-exploration. A study of microvascular free tissue transfer in 21 cases of head and neck reconstruction found that the average re-exploration time for non-salvaged flaps was 3.9 days, in comparison with 1.3 days for successfully salvaged flaps [16]. Thus, careful monitoring and early detection of complications is vital for preventing flap failure.

2.4 Standard Postoperative Monitoring Practices

After undergoing microvascular free tissue transfer, patients are frequently monitored in the ICU by in-house nursing staff. Patients are kept on strict bedrest for the first 24 hours to limit the risk of postoperative complications. To prevent vascular occlusion of the newly transplanted flap, additional precautions are taken to prevent anemia and keep the patient well hydrated. In cases of head and neck reconstruction, tracheostomy is a routine procedure used to provide a pathway for breathing [11]. Careful attention must be paid to the positioning of the tracheostomy to ensure that it does not exert excessive pressure on the flap.

Clinical examinations are given every hour for the first 24 hours, followed by every two hours for the next 48 hours, and finally every four hours until discharge [3]. During these examinations, the health of the newly transplanted tissue is evaluated using six parameters: color, turgor, capillary refill, prick test, temperature, and Doppler ultrasound.

Visual examination by a nurse is used to evaluate the flap's color. A healthy, well-perfused flap will appear pink in color. A bluish hue may indicate venous occlusion, whereas pale flap may suggest arterial occlusion [11].

Turgor of the flap is assessed by pinching a portion of the tissue for a few seconds and releasing it to evaluate how quickly it returns to its original state. Healthy flaps will appear minimally edematous, meaning that there is little to no swelling in the tissue. An accumulation of fluid in the flap is one of the first visible signs of thrombosis, and may indicate occlusion of the primary vein [4].

Capillary refill test is performed by applying pressure to the surface of the tissue with a fingertip and releasing the pressure to evaluate the reperfusion of the tissue. A healthy flap will exhibit a capillary refill time of approximately 2-3 seconds. Venous occlusion will result in a shortened capillary refill time (less than 2 seconds), whereas arterial occlusion will have an increased capillary refill time (greater than 3 seconds) [11].

The prick test is conducted daily to evaluate dermal bleeding in the flap. Using a 25-30-gauge needle, the surface of the tissue is punctured and the resulting blood flow is observed. This may be performed at multiple points on the flap to ensure proper perfusion across the entire surface. A healthy flap will show immediate, bright red blood flow. Dark blood is potentially indicative of venous occlusion. A flap that exhibits delayed blood flow beyond twenty seconds or no flow at all may be experiencing arterial thrombosis [11].

Temperature is evaluated by palpation of the flap or temperature strip indicators placed on the surface of the tissue. Healthy, well-perfused tissue should feel warm to the touch. An acute decrease in surface temperature of 3°C may indicate arterial occlusion, whereas a uniform drop of 1-2°C suggests venous occlusion [17].

Finally, Doppler ultrasound is used to evaluate blood flow through the flap. One method for monitoring this parameter is using a handheld Doppler probe, which is placed against the surface of the tissue in order to obtain an acoustic Doppler signal. Immediately after surgery, the locations of the arterial and venous vessels are marked on the surface of the tissue to provide guides for postoperative monitoring measures such as Doppler ultrasound. Additionally, arterial flow can be distinguished from venous flow due to the pulsatile nature of the signal. Arterial occlusion can be detected by loss of the Doppler signal across the flap. However, it takes a highly trained ear to be able to detect the more subtle changes leading up to complete arterial occlusion or to be able to detect venous occlusion, as the arterial pulse may still be present in the signal [11].

2.5 Alternative Postoperative Monitoring Methods

2.5.1 Temperature

Temperature of the newly transplanted flap is a simple and reliable method for determining the health of the tissue. As a transplanted flap acclimates to its new location at the recipient site, the tissue reaches normal body temperature of approximately 37°C. In the event of vascular occlusion, blood perfusion through the flap is compromised and a corresponding drop in temperature is exhibited in the flap [17].

2.5.1.1 Thermistor Probe

Thermistor probes are a simple and cost effective method for measuring temperature in free tissue. A major advantage of them over the currently accepted temperature strip method is that they can be used to continuously measure temperature near the surface of the flap. A thermistor changes its resistance based on temperature so it can be used to transduce a temperature into a corresponding voltage. Their placement at the surface of the flap, however, does subject them to a lot of variation caused by differences in temperature of the room and other factors. This makes them an unlikely candidate for reliable monitoring of the flap but they noninvasive and easy to implement so therefore good to have in addition to measuring other parameters.

2.5.1.2 Thermal Imaging

Thermal imaging is a much more sensitive way of measuring temperature. It is based upon the principle that heat emits energy in the infrared wavelength, which can be picked up by an infrared camera. As shown in Fig. 2.1, a study was done using pedicle flaps in rats and it is possible to thermally see the blood vessels supplying the flap as well as the subtle changes due to variations in flow caused by different angles of elevation [18]. Major disadvantages to thermal imaging for the purposes of continuous monitoring are that it requires bulky and very expensive equipment, especially if each patient required their own separate unit.

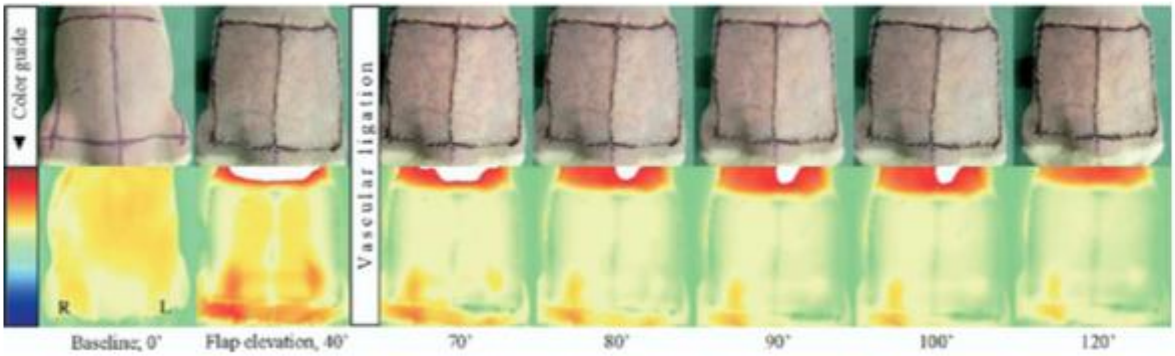

Fig. 2.1 Thermal Imaging in Free Tissue Transfer. Thermal images of pedicle flap in rat highlight changes at different elevations and the sensitivity of thermal imaging [18].

2.5.2 Doppler Ultrasound Flowmetry

Doppler ultrasound flowmetry operates on the principle of the Doppler shift: when a wave of some frequency is reflected by an object travelling at some velocity, the frequency of the reflected wave will be shifted with respect to the frequency of the incident wave by a factor directly proportional to the velocity of the struck object. This principle can be applied in the body to measure the velocity of blood by relying on particles in the blood such as red blood cells to reflect an ultrasonic wave at some shifted frequency. An ultrasound probe uses two piezoelectric crystals to simultaneously emit and receive ultrasonic waves and the difference in frequency between the two can be audibly played to characterize the flow of blood in the region of interest. Ultrasound transducers can be placed on the flap at the surface, for noninvasive measurement, or they can be implanted directly around the blood vessel of interest, as in the case of the Cook-Swartz implantable probe.

2.5.2.1 Handheld External Doppler Probe

External Doppler ultrasound probes are placed on the surface of the flap and directed towards either the vein or artery supplying the flap. They must be placed precisely in order to obtain a good signal. The signal can also be dependent upon the many layers of tissue between the probe and the blood vessel. Every interface between two different kinds of tissue is an opportunity for some of the ultrasound signal to reflect back to the transducer before making it to the blood vessel, hence the use of ultrasound gel to couple the transducer to the surface of the skin so that the entirety of the signal does not get reflected before even penetrating the epidermis. Additionally, different probe frequencies can be selected to adjust the penetrating depth where lower frequency probes will have a greater depth of penetration. The most common standard for monitoring is using a handheld Doppler ultrasound unit and audibly listening to the signal from the flap to diagnose its condition. This can prove unreliable due to the difficulty of recognizing subtleties in the signal. Some advantages of using Doppler ultrasound is that it is already readily available in many hospitals and is a noninvasive method of measurement. However some of the disadvantages are the need for attending physicians to continually check on the patient and manually perform the recording and also that the effectiveness is highly dependent on the particular physician's skill level in listening to flaps.

2.5.2.2 Cook-Swartz Implantable Doppler Probe

Implantable Doppler probes are an effective method for measuring Doppler ultrasound in newly transplanted flaps. Unlike the external Doppler probe, these probes are not as affected by the attenuation of the signal through layers of intermediary tissue as they are wrapped directly around the blood vessel of the flap during surgery and left in for postoperative monitoring. As such, the signal recorded is always from the probe in the same orientation because it does not need to be continually repositioned to search for the signal like probes on the surface of the flap. Implantable Doppler probes have been shown to detect signs of occlusion earlier than clinical methods [20]. Swartz et al have previously shown that when placed on the artery they are capable of immediately detecting arterial occlusion and venous occlusion after 220±40 minutes and when placed on the vein they are capable of immediately detecting venous occlusion and arterial occlusion after 6±2.4 minutes [21]. However, implantable Doppler probes also have a number of disadvantages that limit their use for clinical monitoring. Recent studies have shown false positive rates of approximately 10% amongst flaps that have been monitored using implantable Doppler methods [22]. More significantly, implantable probes have been shown to cause venous congestion in some cases due to excess pressure around vessel. So while the venous placement would seem necessary for accurate detection, it subjects the more delicate venous anastomosis to potential complications [23]. This is a critical limitation of implantable Doppler technology, especially for use in free flap monitoring where vascular flow can be easily compromised.

2.5.3 Laser Doppler Flowmetry

Laser Doppler flowmetry operates on the Doppler shift principle, similar to Doppler ultrasound flowmetry, but it uses waves of light instead of sound. Light being a much higher frequency does not penetrate as deep as sound and laser Doppler flowmetry looks at blood perfusion approximately 1-2mm under the surface instead of blood flowing through larger blood vessels as Doppler ultrasound might. It works on the principle that a single wavelength of light, a laser, can be emitted into the tissue and what is reflected will be a combination of light that has been Doppler shifted due to coming into contact with moving particles and light that has not. The amount of Doppler shifted light and the amplitude of that shift correlate to the perfusion of blood in the subcutaneous layers [24].

Laser Doppler flowmetry is highly effective for detecting both venous and arterial occlusion. A study of free flap monitoring in maxillofacial reconstruction found that laser Doppler flowmetry was able to detect signs of vascular occlusion earlier than traditional clinical assessment. However, there are a number of inherent limitations associated with this method. Namely, laser Doppler signal requires careful interpretation. Generally, arterial occlusion results in an abrupt decline of flow values, while venous occlusion results in a more gradual decline. However, in many cases it can be difficult to distinguish between the two using laser Doppler flowmetry [25]. A healthy flap will have a pulsatile waveform and fluctuations in the waveform caused by normal vasomotor and physiological processes. A failing flap may show a lack of pulsatility or fluctuations. Because laser Doppler flowmetry does not measure in any physically relevant units, all measurements are relative and determinations about the flow in the flap need to be made in reference to baseline values [5].

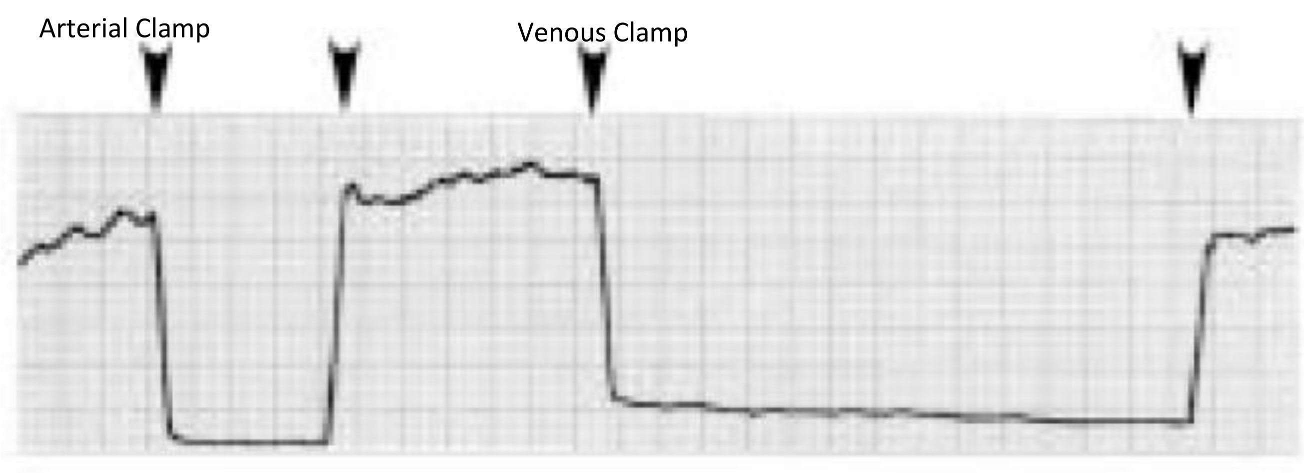

Fig. 2.2 Laser Doppler flowmetry in rabbit ear model [26].

2.5.4 pH

Measuring pH has proven to be an effective method for detecting both venous and arterial occlusion in newly transplanted tissue. Prolonged vascular compromise initiates an ischemic cascade in tissue. Lack of oxygen in the tissue results in a failure of ATP production, causing cells to switch to anaerobic metabolism. This leads to buildup of lactic acid in the cells, causing a drop in pH across the tissue. This can be seen very clearly in Table 2.1and Fig. 2.3 below. Thus, acidosis of the tissue is one of the first indicators of flap failure [27]. By monitoring pH it is possible to determine whether there is a vascular insufficiency in the flap and also even the extent to which the flap has become ischemic and whether it is still viable or not [28]. pH is likely the most promising of all the physiological parameters to measure for an accurate representation of the condition of the flap however none of the currently available sensors meet the requirements necessary for monitoring in microvascular free flap applications.

Table 2.1 pH drop in skin flaps after vascular compromise [28].

Time post occlusion (mins)	*Artery only (6 flaps)*	*Artery and vein (4 flaps)*	*Vein only (4 flaps)*
0	7.43	7.44	7.43
5	7.34	7.35	7.41
10	7.27	7.28	7.38
15	7.18	7.23	7.35
20	7.12	7.16	7.33
30	7.06	7.10	7.29
Overall drop in pH	0.37	0.34	0.14

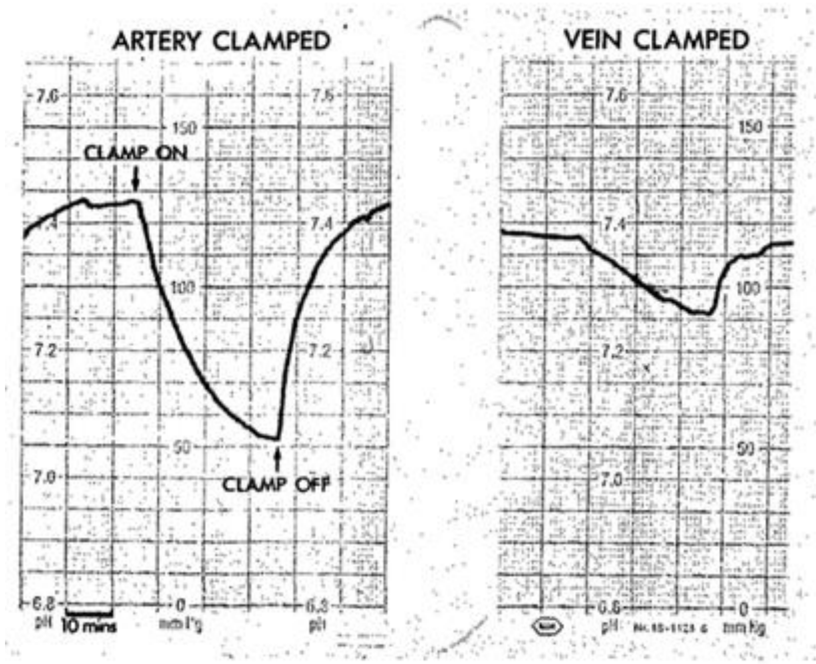

Fig. 2.3 pH drop in rat skin flaps subject to arterial and venous clamping [28].

2.5.4.1 ISFET Probe

Ion sensitive field effect transistors (ISFET) are a solid-state device sensitive to pH. They are not very accurate and readings can vary as much as 0.34pH units due to temperature alone. Drift over extended periods exceeds 1pH unit [29]. Other limitations of this technology are the brittleness of the silicone substrate, which limits applications of ISFET *in-vivo*. Additionally, encapsulation of the sensor has proven to be problematic due to the need to separate the inner wiring from the fluid being measured [30]. In conclusion, they would not be suitable for integration in a continuous free flap-monitoring device.

2.4.5.2 Fiber Optic Probe

Fiber optic pH sensors operate on the principle of measuring hydrogen ion activity or concentration. This is done optically using a pH sensitive dye, often phenol red for its sensitivity from pH 6.8 to 8.2, bound to a fiber optic cable. The dye, absorbs different wavelengths of light depending on the pH and this is detectable by the fiber optic transducer.

Fiber optic pH sensors are advantageous for *in-vivo* use because they do not require an electrical current to be passed and thus are perceived as a safer method. Also the sensor tips can be made very small to be minimally invasive [30]. A major disadvantage is that they are prone to drift and are not accurate over extended period of time and also surgeons can be uneasy about implanting glass into the flap for the potential of it breaking. They would not work for the application of continued postoperative flap monitoring because the five-day period of monitoring would render their results unreliable.

2.5.4.3 Glass Microelectrode

Glass microelectrodes have the necessary accuracy and precision to within 0.01pH units for measuring pH in-vivo and have found to be more stable than the other methods by comparison. A major disadvantage to these electrodes is the necessity for calibration prior to use and multiple reference electrodes to minimize the drift over extended periods of time [29]. This makes set-up cumbersome and time consuming, which is not something the surgeons would desire to do after a long and tedious surgical procedure. Further, delicate glass electrodes become a liability when implanted in the flap if they were to break.

2.5.4.4 NIR Spectroscopy

Near infrared spectroscopy is a noninvasive method for measuring pH that correlates reflectance near infrared spectra to tissue pH. Every molecule reflects a specific and unique spectrum, which makes near infrared spectroscopy an invaluable tool in the identification of chemical species. Using it to measure oxygenation is also fairly straightforward. However, because of the multitude of factors that simultaneously contribute to the pH of tissue it is extremely difficult to correlate across the many confounding factors. There is no single way to claim that a given spectrum correlates to a certain pH. It would be necessary to perform multivariable calibration for each individual species or even subject type being measured because of confounding factors such as penetration depth, tissue color, variations between muscle and fat, and many more. Still, the method has shown some promise in a study utilizing five New Zealand White Rabbits where the pH of rabbits within the population could be determined from the calibrated spectra. It is not currently a feasible method for use in humans until much more data is collected, an endeavor beyond the scope of this project [31].

2.5.5 Oxygenation

Oxygenation is measured by finding the percent oxygen saturation of hemoglobin in the blood. This protein is found in red blood cells and is responsible for transporting carbon dioxide and oxygen to and from the lungs, respectively. Oxygen saturation can be used as a means for evaluating arterial blood flow and tissue perfusion in newly transplanted flaps. Thrombosis results in a decrease of oxygenated blood both in the flap artery and throughout the tissue. This change can be detected using a number of different techniques and oxygenation parameters, including arterial blood gas (SaO_2), pulse oximetry (SpO_2), and NIR spectroscopy (StO_2).

2.5.5.1 SaO_2: Arterial Blood Gas

SaO_2 is an invasive measure of arterial oxygen saturation that involves taking a sample of arterial blood and testing it with a blood gas analyzer. This measures the saturation of O_2 in the arteries, but is not a measure of tissue perfusion or oxygenation [32]. A major disadvantage to this method is that it required invasive sampling of the blood and the analysis is often performed *ex-vivo* on large and expensive laboratory equipment. As such, it is not a feasible method for continuous monitoring.

2.5.5.2 SpO_2: Pulse Oximetry

SpO_2 is a noninvasive measure of arterial oxygen saturation that can be measured using a pulse oximeter. Pulse oximetry uses the principle of light absorbance to determine the oxygen saturation of hemoglobin. A study of pulse oximetry in vascular occlusion of a rabbit ear model found this method to be highly sensitive to arterial occlusion, producing an immediate response in the event of thrombosis. However, changes in SpO_2 were less pronounced in the event of venous complications, exhibiting only a gradual decline and consistent normoxemic levels after venous occlusion [26].

2.5.5.3 StO_2: NIR Spectroscopy

StO_2 is a measurement of tissue hemoglobin oxygen saturation that can provide indications of vascular compromise in newly transplanted tissue. Unlike SaO_2 and SpO_2, StO_2 measures the saturation of O_2 that has diffused into tissue cells. Thus, StO_2 can be used to evaluate tissue perfusion [32]. Lack of oxygenated blood flow through the flap in the event of arterial or venous occlusion will result in a decrease in StO_2 (see Fig. 2.4 and Fig. 2.5). Figure 2.4 depicts StO_2 measurements for arterial compromise in microsurgical breast reconstruction over the course of 4 hours. There is a significant decrease in tissue oxygen saturation (20% decrease over one hour) at time 4:00. Figure 2.5 depicts StO_2 measurements for venous compromise in microsurgical breast reconstruction over the course of 4 hours. There is a significant decrease in tissue oxygen saturation (20% decrease over one hour) at time 3:30, as well as a drop below 30%.

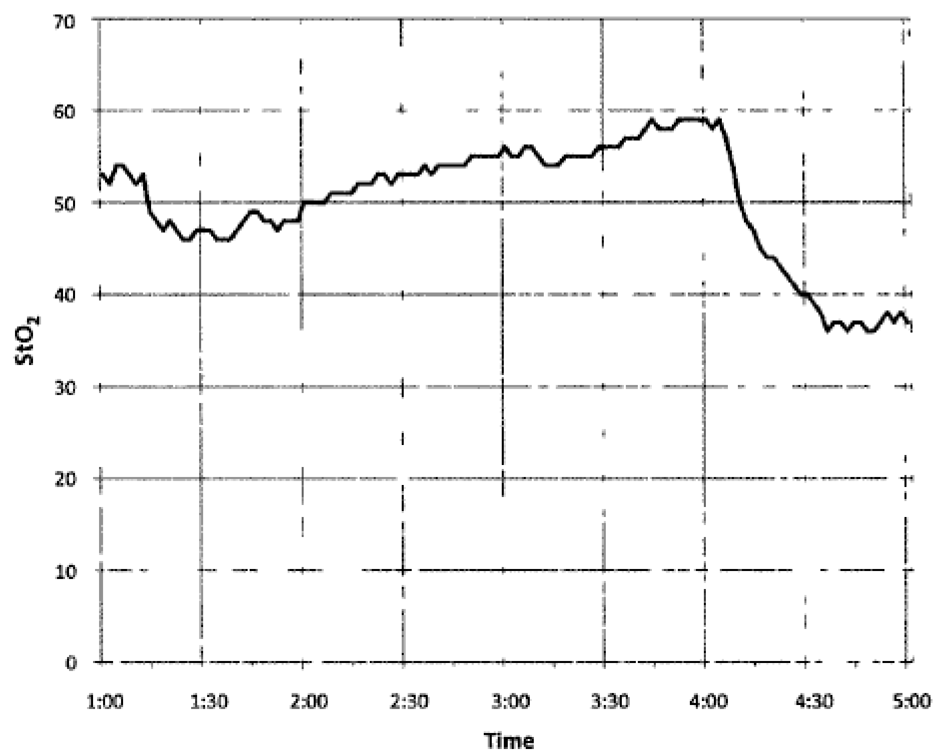

Fig. 2.4 StO_2 in arterial compromise [33].

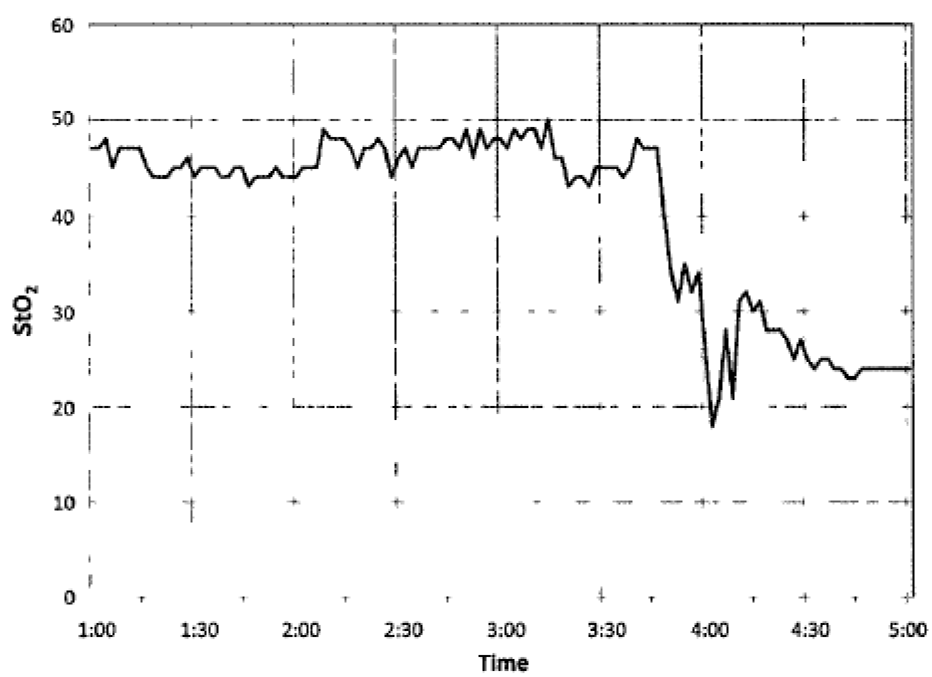

Fig. 2.5 StO_2 in venous compromise [33].

ViOptix currently makes T. Ox, a device for noninvasively measuring StO_2 levels in free flaps using the principle of NIR spectroscopy (see Fig. 2.6). This device provides continuous, real-time StO_2 measurements and wirelessly transmits the data to any Wi-Fi enabled device [34].

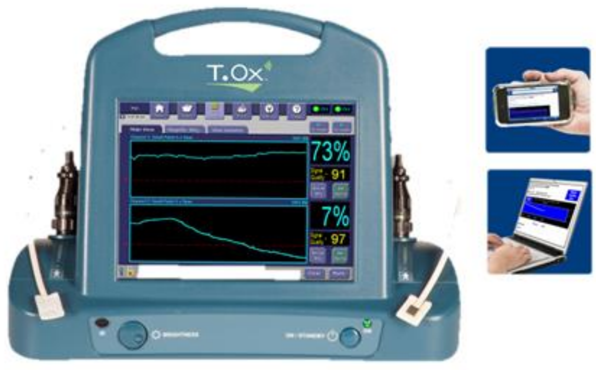

Fig. 2.6 T.Ox Tissue Oximeter by ViOptix [35].

Chapter 3. Project Strategy

The goal of this project is to develop a device for continuous postoperative monitoring of microvascular free tissue transfer. The following chapter outlines the strategy of this project including the revision process of the client statement, the primary and secondary objectives, and the project constraints.

3.1. Initial Client Statement

This MQP project is being done under the sponsorship of Doctors Samandar Dowlatshahi and Raymond Dunn of UMass Medical School. They had initially provided the following statement:

> "This MQP will build on last year's MQP that has already created a device that monitors tissue by means of continuous Doppler and temperature recording, transmitted to a smartphone. The focus this year will be on pH and CO_2 monitoring, using invasive or noninvasive techniques, planning for eventual use on human subjects."

3.2 Previous MQP

This project was adopted as a continuation of work done by a previous MQP, who were also working on monitoring temperature and Doppler ultrasound signal in transplanted flaps [36]. This group developed a proof-of-concept LabView Virtual Instrument (VI) that used a personal computer to record Doppler ultrasound signals (see Fig. 3.1). They also used software called Plotly to be able to plot the temperature data from a thermistor and upload it to an Amazon Web Services Server where it could be viewed from an AppGyver generated iPhone application.

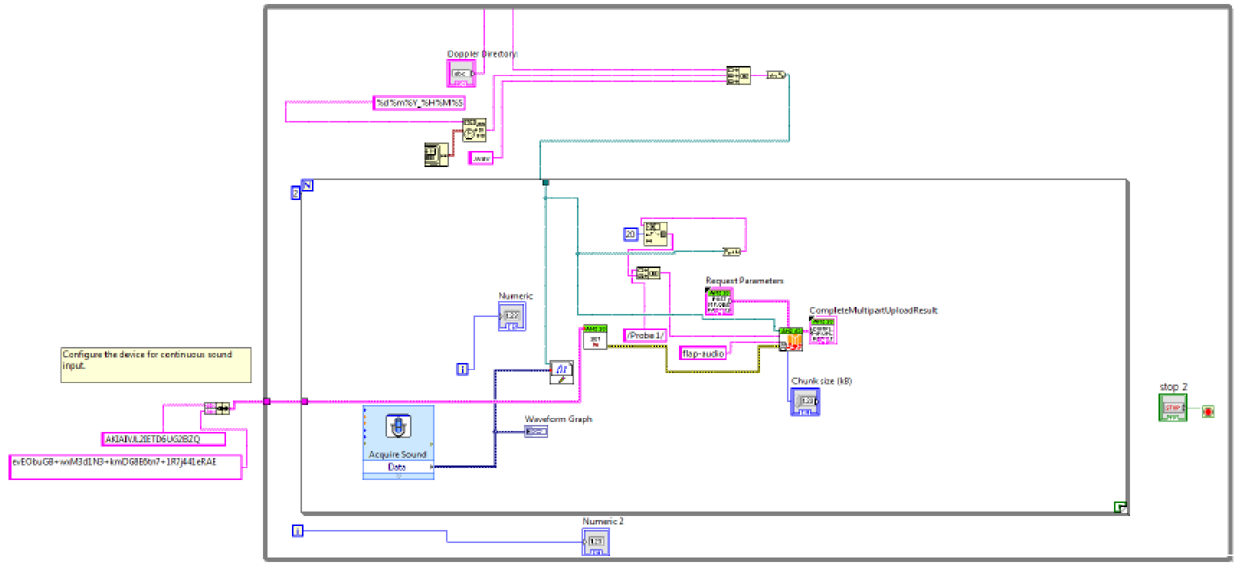

Fig. 3.1 LabView VI developed by previous MQP for recording Doppler signal.

3.3 Design Requirements: Technical

3.3.1 Objectives

Balancing the needs of the clients as well as the needs expressed in literature about postoperative free flap monitoring, the team has come up with the following three primary objectives. They are that the device must be dependable, user friendly, and adaptable. The most important of these is dependable because it directly relates to the safety of the patient and effectiveness of the device, both of which are critical to the purpose of the device. Next is user friendly because it determines how well the device is received by the patient and the physicians that will be using it; this helps ensure that it is actually used. Finally comes adaptable because being able to use the device on different flaps or with different sensors as technology develops will ensure its future success. Under each of these primary objectives is a set of secondary objectives. All of the objectives are ordered in Fig. 3.2 below.

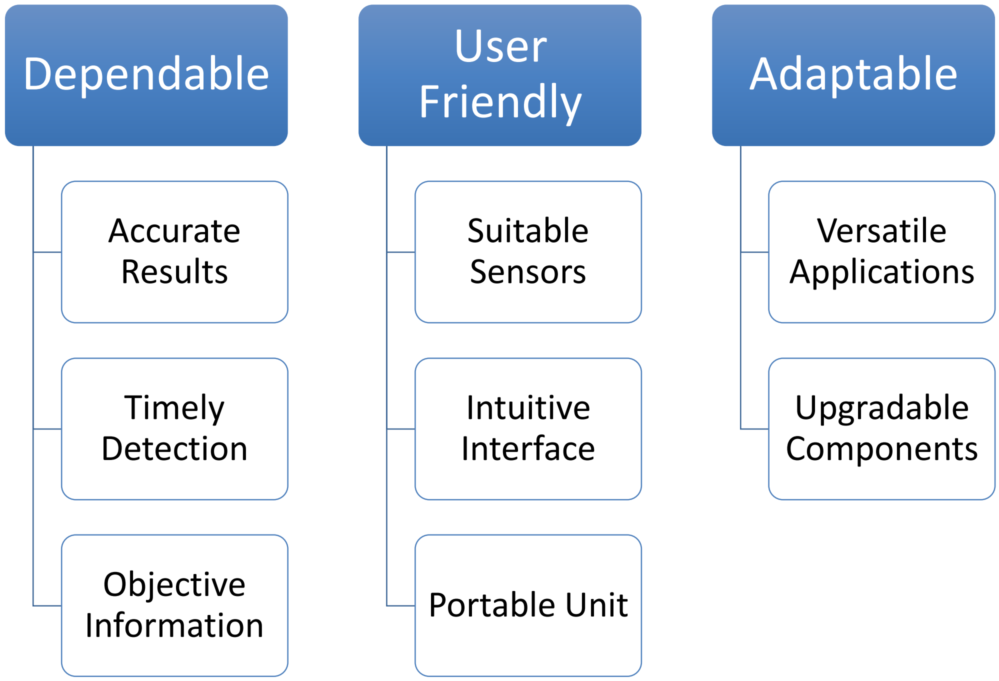

Fig. 3.2 Objectives tree.

3.2.1.1 Dependable

The first primary objective for the device is that it is dependable. This includes the secondary objectives that the device obtains accurate results, responds in a timely fashion, and reports objective information.

It is most important that results from the device are accurate if clinicians are going to be relying on them to make informed decisions about patient care. It cannot falsely indicate a problem with the flap because this could subject the patient to unnecessary exploratory surgery or other measures intended to remedy a nonexistent problem. It also cannot fail to indicate when there truly is a problem with a flap because then it will be overlooked and can quickly lead to a complete loss of the flap.

Next, in order to improve the current methods of monitoring, the device must ensure timely detection and intervention. Detecting signs of occlusion and ischemia as soon as possible is critical. Then, making physicians aware of detected problems immediately and providing them with information in real time will enable timely intervention and recovery of the flap.

It is important for the device to present objective information to clinicians. A significant flaw with current monitoring methods is their subjectivity[1]. Results should require minimal interpretation by the attending staff to know the condition of the flap. Additionally, a very clear threshold should be defined for whether to alert physicians of a problem. This will ensure consistency and minimize any delay in treatment.

All of these things culminate into a device that can be depended on to monitor the status of a free flap. This kind of dependability is critical if the device is to become FDA approved as a standard method for flap monitoring; it insures that its use will always be an asset and never pose additional risks to the patient.

3.2.1.2 User Friendly

The second primary objective of the device is user-friendliness. This objective encompasses all aspects of the device that involve or require user interaction. It includes the secondary objectives of sensor integration, intuitive user interface, and portability.

The most important secondary objective in this category is sensor integration. It is critical that the probes used for monitoring are easy for the surgeon to implant within the tissue during the operative procedure. Sensor attachment must be taken into consideration, namely where in the tissue they will be placed. For example, a probe that requires precise placement around a blood vessel may be difficult to accomplish and compromise the usability of the device. Also, the bulkiness of the sensors will need to be taken into account as it affects the patient's overall comfort level during recovery.

The next secondary objective in this category is to develop an intuitive user interface for monitoring data. The smartphone application should be designed in such a way that it is easy for the surgeon to navigate. It is critical that the data is both accessible and easily interpretable. Consideration must also be given to the notification system to ensure that it alerts the surgeon at proper times with accurate and informative information. Additionally, the smartphone interface should be clean and aesthetically pleasing. All of these factors minimize frustration and ensure a seamless user experience.

A portable device is necessary when patients need to be relocated to receive certain treatments. The goal is to design the device in such a way that it is compact and easily transportable. This will allow for it to move from room-to-room along with the patient. Ideally, the device should be able to move with the patient without interrupting data acquisition.

A user friendly experience for both the patient and the physicians that will be using the device is what makes it something people want to use and could set it apart from the other types of monitors on the market.

3.2.1.3 Adaptable

The third primary objective for the device is that it should be adaptable. This includes the two secondary objectives that the device should be versatile for a number of applications and have upgradable components.

The most important secondary objective in this category is versatility, meaning that the device can be used for a wide range of flap monitoring applications. Namely, the device should be adaptable to flaps of different sizes, tissue types, and locations. In order to achieve versatility, careful consideration must be given to sensor size, type, and attachment. Sensors must be chosen that are applicable to flaps of multiples sizes, different types (skin, muscle, etc.), and different locations (limb, abdomen, etc).

The other secondary objective in this category is to develop the device with upgradable components. Sensor interchangeability is an important aspect of ensuring that the device is applicable to a wide range of flap transplants. Additionally, the device should be designed such that new measurement techniques can be incorporated and improved upon in the future. This is critical to ensuring that the device is current and both meets and exceeds the industry standards for postoperative monitoring.

Overall, the adaptability of this device is critical to ensuring its success both now and in the future. The diversity of free tissue transfer calls for a postoperative monitoring device that can be changed in response to the specifications of the individual transplant and the constant improvements in technology.

3.3.2 Constraints

A set of design constraints was established based on the needs of the client and the technical limitations of the project. These were identified as attributes of the project that absolutely must be accomplished in order for the project to be successful. Taking this into consideration, it was determined that the device must be (1) safe for the patient, (2) compliant with existing protocols, (3) comfortable for the patient, and (4) completed within the given timeframe.

3.2.2.1 The device must be safe for the patient

Safety is of the utmost importance when developing a device for medical applications. The potentially invasive nature of this monitoring system requires thorough consideration of potential complications that could arise and precautions that must be taken to ensure the patient’s wellbeing. It is critical that the device does not induce any negative physiological reactions that could threaten the patient's health. As such, three primary patient safety concerns have been identified: tissue necrosis, inflammation, and infection.

First and foremost, monitoring of the flap must not threaten or disrupt the success of the transplantation. It is critical that the device does not pose a risk of vessel occlusion, hemorrhage, or any other physiological event that could disrupt blood flow or result in tissue death. Second, the biocompatibility of the device must be considered to ensure the implantation of the sensors into the tissue or the attachment of noninvasive sensors does not elicit an inflammatory immune response at the site of contact. Material selection will be key to avoiding rejection of the probes and ensuring the safety of the patient. Finally, the device must be designed in a way such that the probes can be adequately sterilized and implanted to avoid infection.

3.2.2.2 The device must be compliant with existing protocols

Although the device will provide a novel method for postoperative monitoring of the flap, it is important to consider all of the existing protocols that are in place and ensure that this new approach will not disrupt the workflow. This can be further separated into two primary considerations: intraoperative protocols and postoperative protocols.

Fixation of the probes to the tissue during the operation must not interfere with anastomosis, wound closure, or any other procedure of the transplant. It is vital that the device be designed in such a way that it can easily be implemented into the current surgical protocols without compromising the success of the transplant. Similarly, the device must not complicate or interfere with the postoperative monitoring of the patient. The ultimate goal is for the device to become the new gold standard in postoperative free tissue transfer monitoring, however until then it is critical that medical staff can continue to perform their existing monitoring methods while use of the device is early in the adoption phase. Attachment of the device should not obstruct an attending physician's ability to assess the color, temperature, turgor, capillary refill, and Doppler ultrasound of the flap.

3.2.2.3 The device must be comfortable for the patient

In addition to ensuring that the device is safe and suitable, it is necessary that the device is completely comfortable for the patient and does not induce any extraneous pain. To address this constraint, the invasiveness of the sensors must be considered. Fixation of the sensors to the tissue must be achieved is such a way that it is virtually undetectable by the patient. Connection of the sensors to the main unit must also be taken into consideration to ensure that that the patient’s movement is not excessively restricted due to wires or other apparatuses.

3.2.2.4 The project must be completed within the given time frame

An inherent constraint of the project is the timeframe for completion. The project spans approximately eight months in which the team must design, develop, and test the device. This places limitations on the postoperative monitoring methods that are being considered. For example, when investigating methods of flap monitoring the team immediately disregarded those that are in an experimental phase due to a lack of time for testing and validating these approaches. Instead, approaches have been considered which are feasible within the eight-month time frame and have already shown promising signs of clinical validation.

3.4 Design Requirements: Standards

The invasive nature of this monitoring requires careful consideration of the standards in place for medical devices. In order to gain approval from the Food and Drug Administration

(FDA), the device must meet a number of regulations. For example, the system will most likely require a 510(k) premarket submission. This proves that the device is "substantially equivalent" to existing technology in fields such as intended use and performance testing.

Additionally, there are a number of ISO standards in place that must be considered while design and developing the device. ISO 11737-2:2009 identifies the criteria that must be met for medical device sterility using microbiological methods. Because the proposed monitoring device is both invasive and multi-use, sterilization is critical for ensuring patient safety. Such a device must be free of all bacteria and other microorganisms to avoid the transmission of disease and infection.

Another critical ISO standard that must be considered is 10993-1:2009, which evaluates the biocompatibility of medical devices. It is important that the proposed device is compliant with these regulations and does not elicit a negative biological reaction when in contact with the patient. Furthermore, ISO 10993-10:2010, another standard in the class regarding biological evaluation of medical devices, tests for irritation and skin sensitization. It is critical that all probes are compliant with this standard and do not cause any extraneous irritation on the surface of the flap.

The Medical Electrical Equipment General Requirements, IEC60601-1-11:2015, outlines criteria for electrical compliance of the device. It is critical that the device is complaint with such electrical standards in order to be used in a hospital environment on patients. The device will need to undergo formal testing for certification.

In addition to medical regulations, industry standards for technology must also be considered over the course of the project. For example, the website application must be compliant with HIPAA. This will be achieved via a user authentication system to protect patient data, secure data transmission and storage, and separation of identifying information from medical information to ensure that data is anonymous. Furthermore, algorithms developed in MATLAB for analyzing Doppler signal should follow the MATLAB Programming Style Guidelines, which have been established as an industry standard. Similarly, all programming for the mobile interface should meet the W3C standards for application development.

3.5 Revised Client Statement

After reviewing the literature and speaking with the clients, the team established that there exists a significant need for an effective method to detect the earliest possible signs of flap ischemia as a result of anastomotic thrombosis. Along with this, it is necessary to alert medical staff to a potential problem immediately and make measured data available for surgeons to monitor remotely so that they can make an informed decision for the best course of treatment. Another criteria, relating to the user friendliness of the device, which was very important to the clients is that the device is very easy to set up on the patient and does not require extensive preparation or calibration. After extensive research on the methods available for pH measurement in human tissue, the team and the clients made a decision that pH and pCO_2 monitoring were no longer feasible to meet the objectives.

Taking into consideration the different parameters that could be used to evaluate the flap and also some of the existing technology on the market to do so, the team has determined that adapting some of the existing “gold standard” parameters to be continuous and less subjective would be the best course of action. With regards to the previous year’s work, the team will incorporate continuous Doppler and temperature, but intends to take a novel approach in the design of the system to collect, process, and transmit the data. This would result in a novel and cost effective all-inclusive device that could be delivered to the clients at the completion of the project. Additionally, the device will make use of some of the research regarding variations in light absorbance in the tissue of the flap under differing vascular conditions and will include a light reflectance transducer to quantitatively and objectively evaluate the color of the flap to a degree of sensitivity unachievable by the human eye. As such, the team has come up with the following revised client statement to better reflect the goals of the project.

> "Develop a postoperative monitoring device for early ischemic detection in microvascular free tissue transfer. The device shall continuously collect information from temperature, Doppler ultrasound, and light reflectance sensors and process this information to alert clinical staff to signs of complications as well as transmit this information wirelessly so that it can be accessed by the surgeons through a smart-phone application or web portal."

3.6 Project Approach

Once the initial aspects of the project were defined, a concise system was developed for identifying necessary tasks and tracking their completion. After analyzing the project from the broader perspective of objectives and constraints, a more detailed work-breakdown structure was developed (see Fig. 3.3). This divides the project into several primary tasks, each with secondary tasks that outline the phase in greater detail.

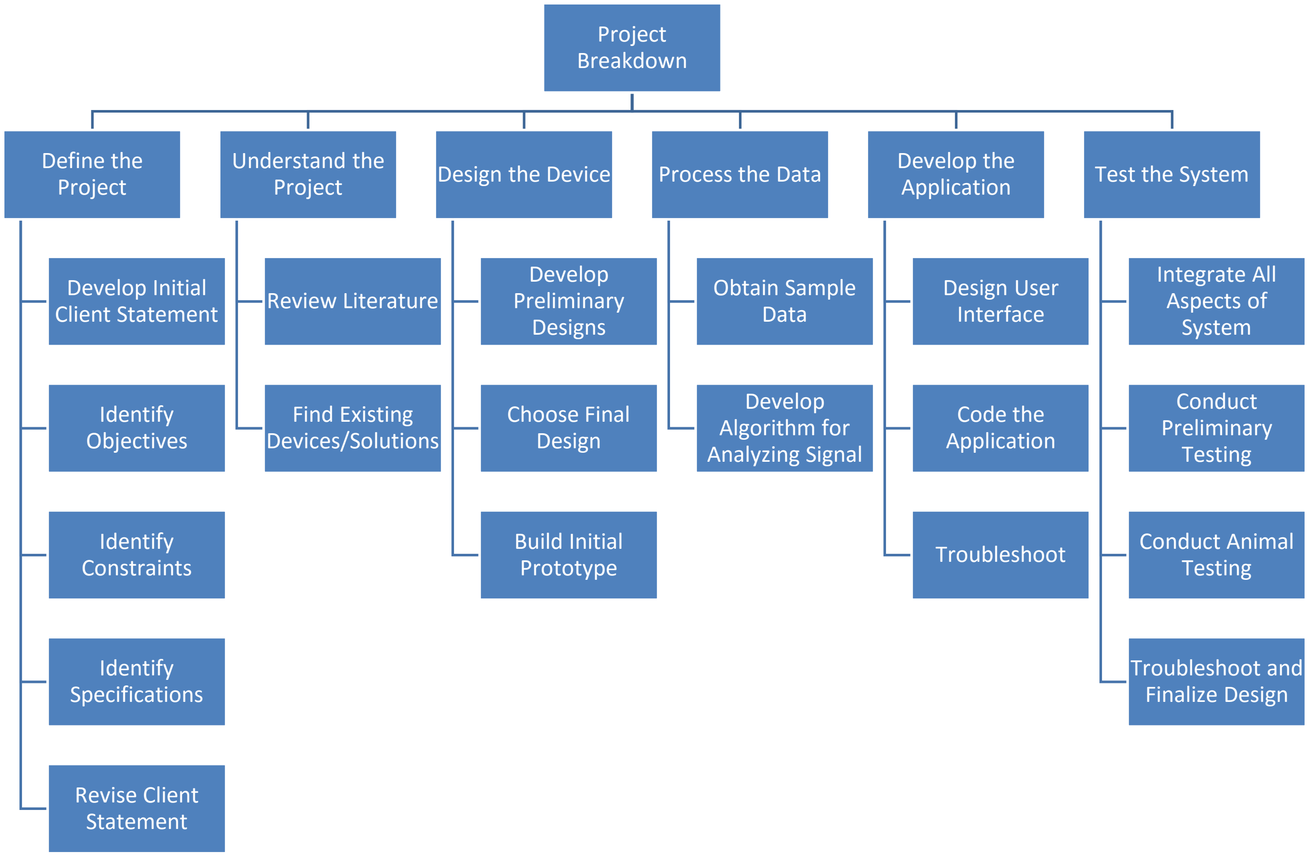

Fig. 3.3 Work breakdown structure.

After developing a work breakdown structure, the tasks were assigned internal completion dates using a Gantt chart (see Fig. 3.4). This schedule outlines the project from start to completion and ensures that the necessary tasks are completed in a timely manner.

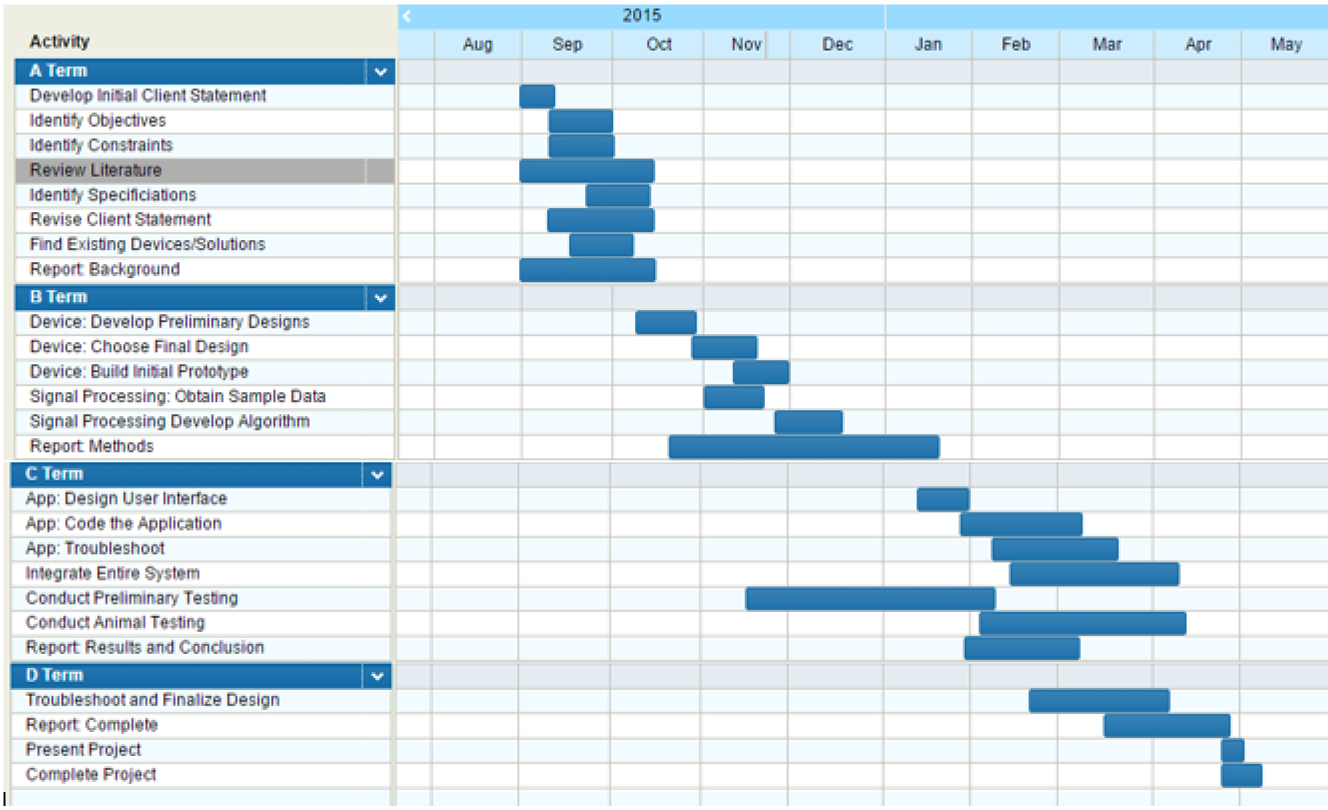

Fig. 3.4 Gantt chart.

Chapter 4: Alternative Designs

4.1 Functions and Specifications

The following set of functions were established to ensure the device meets the objectives and constraints of the project; it was determined that the system will (1) acquire temperature and Doppler ultrasound data from several probes, (2) apply algorithms to process the data in order to detect variance indicative of vessel occlusion, (3) transmit the information wirelessly and store it remotely, (4) display the data on an graphical user interface (GUI), (5) and alert the clinicians of occlusion. The following chapter will discuss each function in greater detail as it pertains to the overall system, including the needs analysis, specifications, and alternative designs considered for each.

4.1.1 Acquire Temperature and Doppler Ultrasound Data

The first functional aspect of the device is collecting periodic data from a set of thermistors and Doppler ultrasound probes. This involves the following components: (1) temperature data acquisition, (2) Doppler ultrasound data acquisition, and (3) probe holder design.

4.1.1.1 Temperature Data Acquisition

Needs Analysis

One of the major functional aspects of the system relies on the ability to collect consistent temperature recordings for the duration of the monitoring period. According to the literature, the device must be able to detect a temperature differential of 1°C or 3ºC in order to detect venous or arterial occlusion respectively [17]. Therefore, the temperature measurements should be accurate to within 0.1°C in order to ensure that variations in temperature are reported accurately.

Additionally, the monitoring method must be relatively inexpensive and sensitive to changes in temperature at the surface of the skin.

Conceptual Design

The device uses thermistors to measure the temperature at the different probe locations. Configured in a simple voltage divider it was not possible to get temperature measurements resolved to 0.1ºC with 10kΩ thermistors and the 10-bit analog to digital converter built in to the Arduino that the device uses for data acquisition. These will be changed to 100kΩ thermistors in the final design for higher sensitivity in the range of temperatures that will be measured. Additionally, the team will opt for 1% tolerances for the most consistency across different probes.

4.1.1.2 Doppler Ultrasound Data Acquisition

Needs Analysis

Doppler ultrasound data acquisition is a key component of the device. It is necessary to be able to collect signals from a minimum of two locations on the flap so that an arterial and a venous signal can be captured. Also, the device must be capable of switching between audibly playing the real time Doppler signal for probe placement and silently capturing the Doppler signal for remote monitoring.

Conceptual Design

The device will make use of an existing Parks Medical model 811-B Doppler ultrasound flow meter and interface with the probe input channels and headphone output port to record signals from multiple Parks Medical infant flat Doppler probes. The reason for using an existing unit it that the Parks Medical flow meter is already approved for medical diagnostics and is currently the model used by UMass Medical Center. Probe signal wires will plug into a multiplexing circuit controlled by the device, which selects which probe signal is fed to the Doppler flow meter. The device then records the corresponding Doppler flow meter output from the headphone output port. Using this method, the device can collect signals from at least two Doppler probes and this could be extended to allow for more probes as well. In order to be able to switch between silently and automatically capturing the Doppler signal for remote monitoring and playing the signal out load on demand in real time, a uxcell HK19F Relay is used to switch the signal between the headphone output or microphone output of the Doppler flow meter.

4.1.1.3 Probe Holder Design

Needs Analysis

Probe holder design was critical for ensuring consistent and effective data acquisition. A number of factors were considered to optimize the design of the holder. First, the holder must be able to effectively hold a flat Doppler ultrasound probe (see dimensions in Fig. 4.1 and Fig. 4.2), and/or a single thermistor (see dimensions in Fig. 4.3). In addition to housing the probes, the holder should be as small and lightweight as possible so as to not create excessive stress on the newly transplanted flap. The holder must also be securable to the skin for long-term monitoring (24-72 hours). Finally, the probe holder must incorporate a method for preserving or replenishing ultrasound gel over the course of its use.

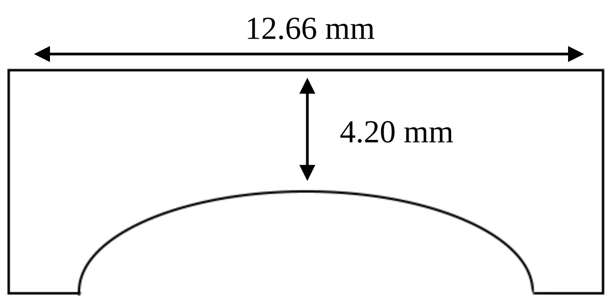

Fig. 4.1 Ultrasound probe front view.

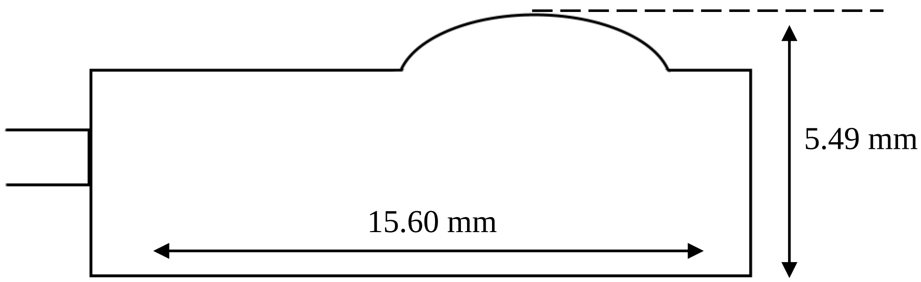

Fig. 4.2 Ultrasound probe side view.

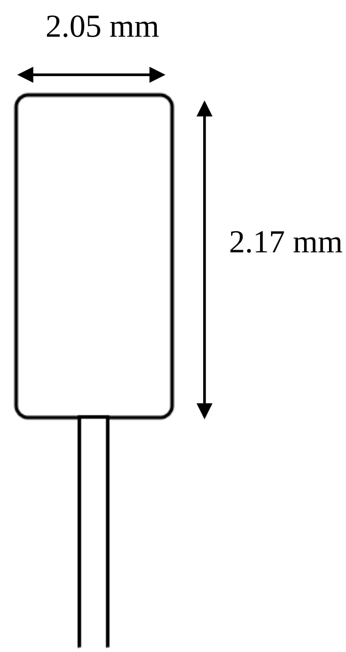

Fig. 4.3 Thermistor side view.

<u>Alternative Designs</u>

Adhesive Patches

The first alternative design that was considered for fixing the probes to the skin consists of adhesive patches. These patches would be designed in two different sizes; either to adhere a Doppler probe to the skin or a thermistor. Several patches could then be placed across the flap in appropriate positions. A hypoallergenic material such as cotton, coated in a biocompatible adhesive, would be used to create the small probe holders. The adhesive portion would then be placed to the skin, fixing the probe against the skin.

Rectangular Suturable Holder with Gel Delivery System

The rectangular suturable probe holder is compact and houses a single flat infant Doppler probe and a thermistor probe for temperature measurement. In addition to housing the probes, the holder has a small port for a 17-gauge catheter to be inserted to allow for the replenishment of ultrasound gel between the skin surface and the ultrasound probe. The gel is necessary to couple the probe to the tissue for adequate acoustic transmission and it can dry out or be absorbed by the skin over time. This design can be seen depicted in Fig. 4.4 below.

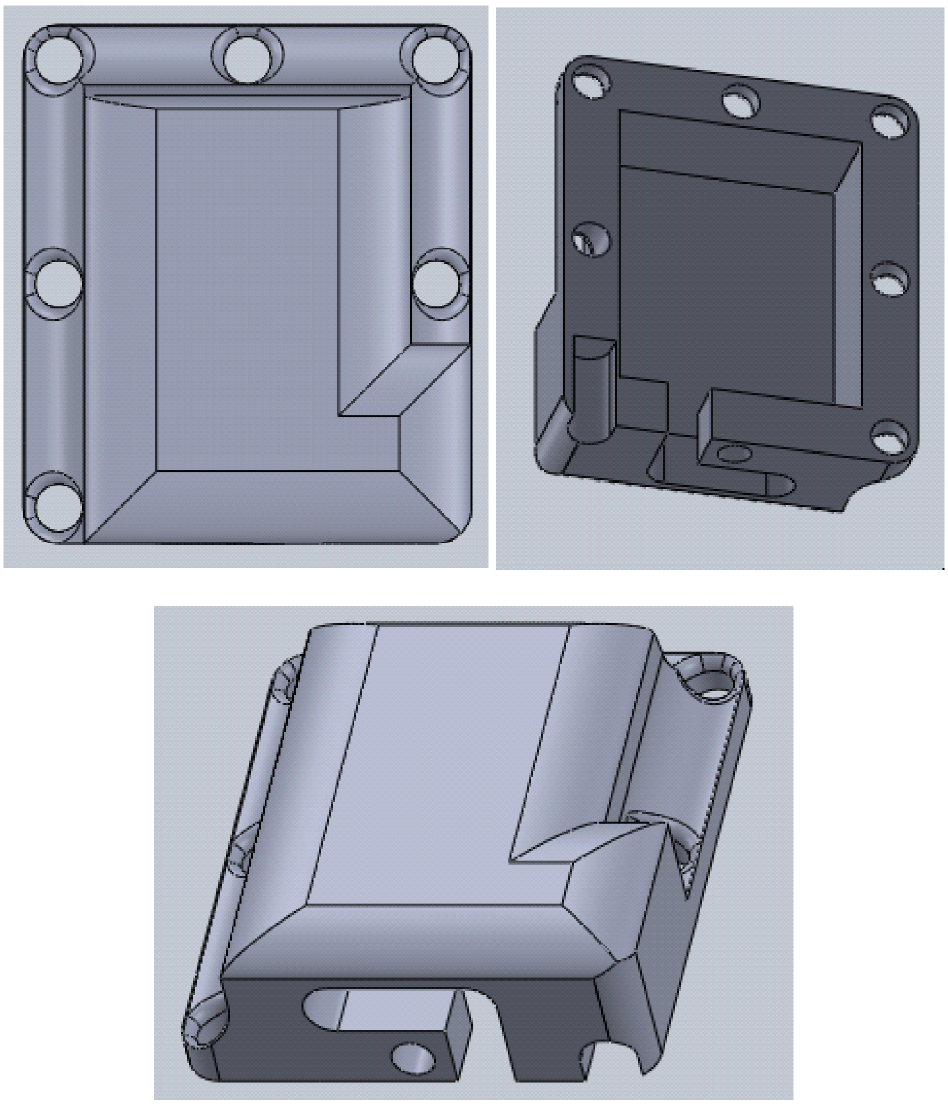

Fig. 4.4 Images from SolidWorks model of rectangular suturable probe holder.

Evaluation of Alternative Designs

Table 4.1 was used to evaluate the probe holder alternative designs in relation the needs that were established for this component. Ultimately, it was determined that the adhesive patches, while meeting some aspects of the design criteria, were not fit for long-term use. This is primarily because of the ultrasound gel interfering with the adhesive, both during initial application and subsequent delivery. This issue was solved with the gel delivery system incorporated into the final design. This alternative design minimized the size and weight of the holder, while maintaining the ability to secure the probes for an extended period of time. The gel delivery system enables the user to easily replenish the ultrasound gel beneath the Doppler probe, optimizing the holder for long-term use.

Table 4.1 Evaluation of probe holder designs.

	Adhesive Patches	Rectangular Suturable Holder with Gel Delivery System
Ability to secure probes for extended time?	Ultrasound gel may interfere with adhesion	Yes
Compact?	Greater surface area because thermistor and Doppler probes adhered separately	Yes
Lightweight?	Yes	Yes
Biocompatible?	Yes	Yes
Ultrasound gel preservation/delivery?	No, adhesive may be affected by additional ultrasound gel	Yes

4.1.2 Algorithms to Indicate Vessel Occlusion

In addition to collecting data and transmitting it the device will continuously be performing analysis on collected data to detect any potential problems with flap perfusion. Using information from both the temperature probes and Doppler probes, the goal is for the device to be able to detect problems before traditional clinical signs become readily apparent.

4.1.2.1 Temperature

According to the literature, a surface temperature differential greater than 1-2ºC between the surface of the flap and that of proximal tissue can be indicative of venous occlusion and a differential greater than 3ºC can be indicative of arterial occlusion [17]. Therefore, in addition to plotting the temperature values for all of the probes, the device also computes the difference in temperature between the probes on the flap and off the flap. When this difference becomes greater than 2ºC it sends an alert to the clinician’s cellular phone via text message and also displays an alert for that patient in the alerts section of the web interface. To assess the difference, the device uses the mean temperature from all of the probes on the flap in the last five minutes and compares it to the mean temperature from all of the probes proximal to the flap in the last five minutes.

4.1.2.2 Doppler ultrasound

For Doppler ultrasound, the team is looking to incorporate metrics such as the average power in a low frequency band indicative of venous flow, the area under the power curve for arterial occlusion, the presence of biphasic or triphasic components, or possibly using a least mean squares approach to compare measured signals to reference signals indicative of different levels of occlusion. The decision for the final design will be based on whichever method is most robust at detecting arterial or venous occlusion from arterial, venous, or mixed signals across the range of sample signals collected during preliminary testing.

4.1.3 Transmit the Information Wirelessly to a Database

The remote monitoring aspect of the device primarily involved two components: (1) wireless transmission of the acquired data and (2) organization of the information in a database.

4.1.3.1 Wireless Transmission of Data

The device connects to the Internet over a WPA2 secured wireless local area network in the hospital. There are two methods the device uses to communicate information with the remote server. The HTTP protocol's built in POST request method is used for simple transactions such as user interface events, temperature values, and Doppler recording filenames. The secure copy protocol (SCP) is used to transfer actual Doppler recording data files in uncompressed audio (.wav) format from the device to a directory on the remote server.

4.1.3.2 Database Design

Needs Analysis

The database must hold the information necessary to associate the device with a particular patient, remotely update the monitoring status of the device, relay temperature and Doppler data to the graphic interface, and secure the graphic interface in order to protect patient information. Also, in the database structure any patient identifying information (name) must be kept separate from relevant medical information (temperature and Doppler recordings).

Conceptual Design

The first database design consists of seven tables, identified by the following titles: patients, deviceRequest, deviceStatus, alerts, Doppler, temperature, and members. The contents and purpose of each table will be discussed in this section. For the binary attributes in each of the following tables, a value of 0 indicates the negative and a value of 1 indicates the affirmative.

The patients table consists of three attributes: patientID (a unique identifier for each patient in the table), lastname (the last name of the patient), and firstname (the first name of the patient). This table is used to store patient information for each individual registered with the monitoring system.

The deviceRequest table consists of five attributes: deviceID (a unique identifier for each device), patientID (the patientID corresponding to the patient who is assigned to the device with the given deviceID), temperature (a binary indicator of whether there is a request to turn on continuous temperature monitoring; 0 indicates no request, 1 indicates a request), Doppler (a binary indicator of whether there is a request to turn on continuous Doppler monitoring; 0 indicates no request, 1 indicates a request), and Dopplernow (a binary indicator of whether there is a request to turn on immediate Doppler monitoring; 0 indicates no request, 1 indicates a request). This table is used to pass information to the device regarding pending changes in monitoring status.

The deviceStatus table consists of five attributes identical to those found in the deviceRequest table. This table is updated once a request is passed to the device and the device monitoring status is updated accordingly. The reason for two identical tables is so a feedback loop can be created between the user interface controls and the device itself. When the device detects a request in the deviceRequest table, it completes the loop by changing its status and pushing that to the deviceStatus table. The user interface pulls status information from the deviceStatus table so if a request is made but for whatever reason does not reach the device, this would be reflected by no change in the devices status.

The alerts table consists of five attributes: alertID (a unique identifier for each alert), patientID (the patientID of the patient that the alert pertains to), epoch (the timestamp for when the given alert was issued), priority (the level of importance of the alert, from 0 to 2, 0 being the least important), and message (a string containing the message associated with the given alert). This table is used to store alerts that are displayed on the user interface.

The Doppler table consists of four attributes: patientID (the patientID of the patient that the recording belongs to), epoch (the timestamp for the given Doppler recording), probe (the number of the probe from which the Doppler recording was obtained), and filename (the name of the file under which the recording is saved in the directory). This table is used to store and organize the periodic Doppler recordings that are collected.

The temperature table consists of four attributes: patientID (the patientID of the patient that the recordings belong to), epoch (the timestamp for the given temperature recording), probe (the number of the probe from which the temperature recording was obtained), and value (the temperature value at the given instant). This table is used to store and organize periodic temperature recordings that are collected.

Finally, the members table consists of seven attributes: memberID (the unique member identifier), username (the username the member uses to login with), password (the password the member uses to login with, stored in encrypted form with SHA-512), email (the e-mail address associated with the member for activating accounts and resetting forgotten passwords), active (whether the member has activated their account), resetToken (a password reset token), and resetComplete (whether the password has been reset). This table keeps track of the authorized members who have access to the site. Members need to log-in through the welcome page in order to gain access to any of the device control or patient monitoring pages of the site. Their credentials are checked against those in the members table and if they match an active session is opened and each subsequent page checks whether the user if authorized before displaying any contents.

4.1.4 User Interface

It is necessary for the end user to be able to start and stop the collection of data, specify certain parameters about the probes installed, and specify which patient they are currently monitoring. Also, once the data is acquired and processed, it must be displayed in such a way that it is easily accessible and interpretable by the clinician. Namely, this includes visual and audio representation of temperature and Doppler ultrasound recordings. The following section will discuss the design of both the physical user interface of the device as well as the graphical user interface associated with it.

Needs Analysis

First and foremost, it is critical that the graphical user interface is secure and protects all patient information that it contains. Next, the graphical user interface must be visually appealing, easy to interpret, and easy to use. The physical user interface must be simple to use and provide immediate access to any functionality that a clinician at the patient's bedside would want to activate such as switching the Doppler signal between silent recording and audible playback on the speaker.

Alternative Designs

Some options the team considered for a graphical user interface were an iPhone application, a built in touch screen display on the device, and a web based user interface. Due to time constraints of the project, it was decided not to pursue the iPhone application. Similarly, the touch screen display would be a desirable feature but not a priority. A web based graphical user interface was decided upon as a good compromise because it can make the data acquired available remotely, it is easily customizable, and also it is portable and able to be viewed not only from a computer but also from the web browser of any smartphone. Also it is possible to easily port a "web application" to a native smartphone application using available online services. To complement the web user interface, which may have some minor delay in transmitting and receiving data and commands, the device will include a series of switches and status LEDs for immediate control at the bedside.

Conceptual Design

The graphical user interface was coded in HTML5, along with a CSS stylesheet for managing all of the stylistic aspects of the layout. All web pages of the GUI are initially redirected to a login page, where the user is prompted to enter their credentials: a username and password chosen by the user at the time of registration (see Fig. 4.5). Additionally, a forgotten password option is provided on the login page, which redirects to a form in which the user can opt to send an email to reset their account password (see Fig. 4.6).

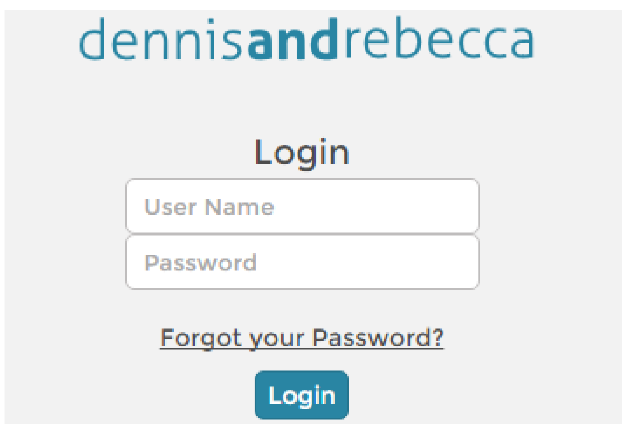

Fig. 4.5 Login page of GUI.

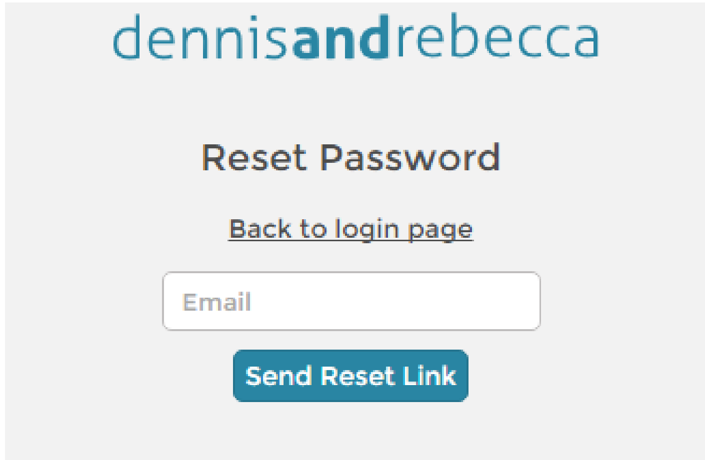

Figure 4.6 Reset password page of GUI.

Upon entering their credentials and logging in, the user is redirected to their Dashboard, a platform for monitoring and managing the patients and devices associated with the user's account. The Patients section contains a table summarizing each patient's temperature and Doppler monitoring status (Fig. 4.7), as well as a list of each patient currently in the database.

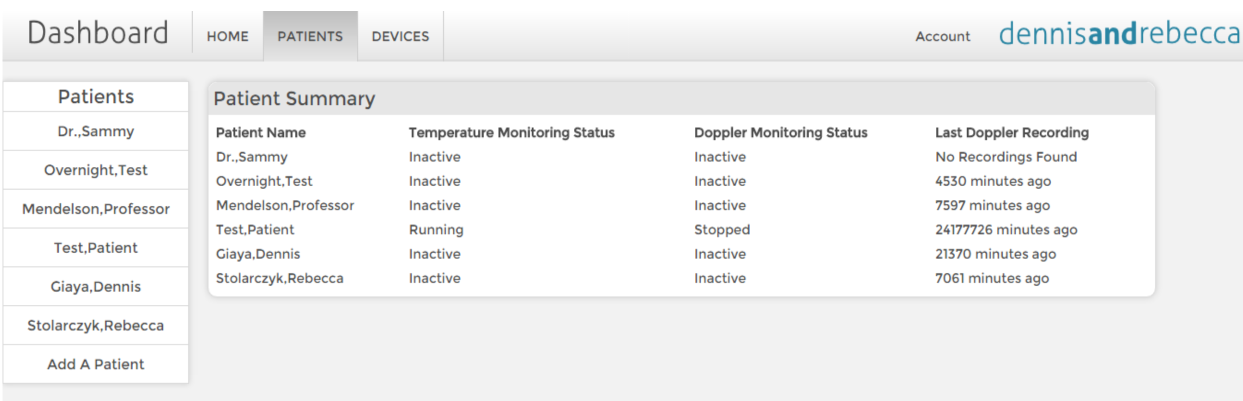

Fig. 4.7 Patient summary page of GUI.

Upon selecting a patient from the list, the user is redirected to the monitoring page associated with the particular patient (see Fig. 4.8). This page contains the following sections: patient alerts (with prioritized messages pertaining to the patient's temperature and Doppler data), temperature data (a graph with real-time curves for each probe), and Doppler data (a list of time stamped Doppler recording files with a built-in player to listen to the recordings).

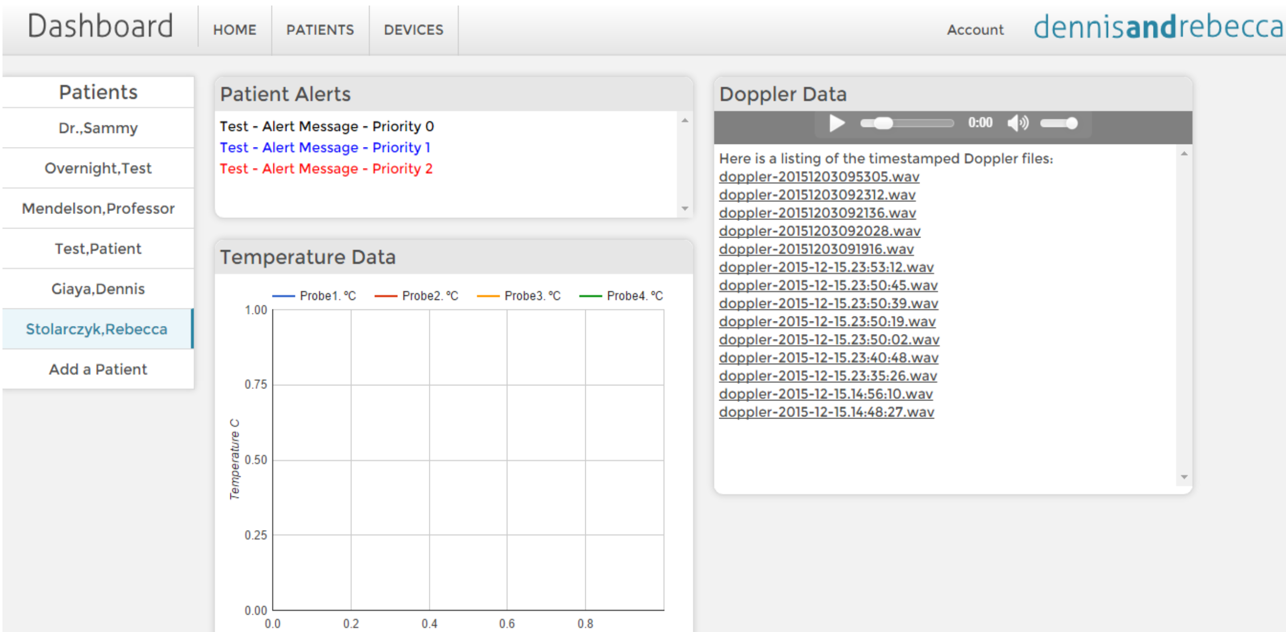

Fig. 4.8 Patient monitoring page from GUI.

An option is also included directly in the "Patients" navigation list for adding a patient to the database. This redirects the user to a form (see Fig. 4.9) that prompts them for the first and last name of the patient. Upon completing the form, a patient with the given information is automatically added to the database and a new page is generated for them displaying their data.

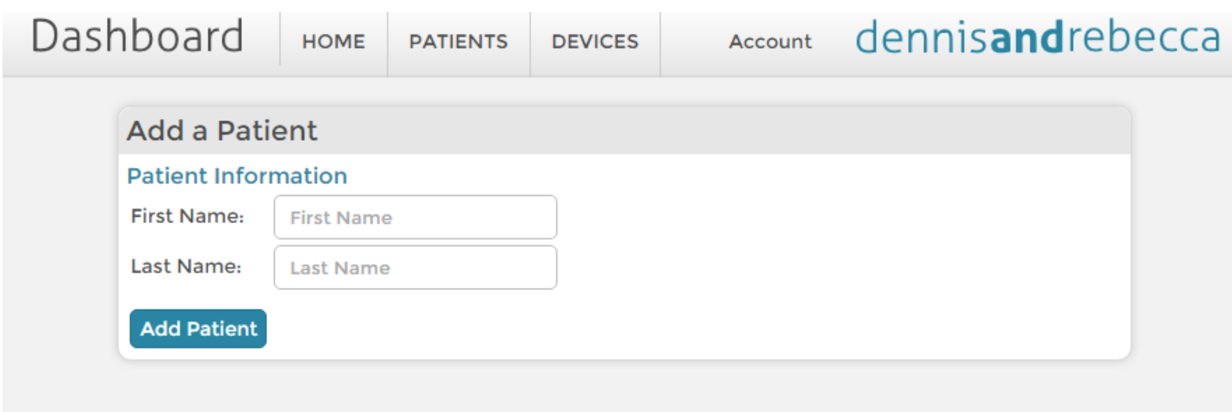

Fig. 4.9 Add patient form from GUI.

Each device also has an individual page containing a "Control Panel" (see Fig. 4.10) for managing the device. Here, the user can assign the device to a particular patient, as well as control the temperature and Doppler recording status. These changes are reflected in the column on the far right of the Control Panel, which displays the current status of each function (patient assignment, continuous temperature monitoring, continuous Doppler monitoring, and current Doppler monitoring).

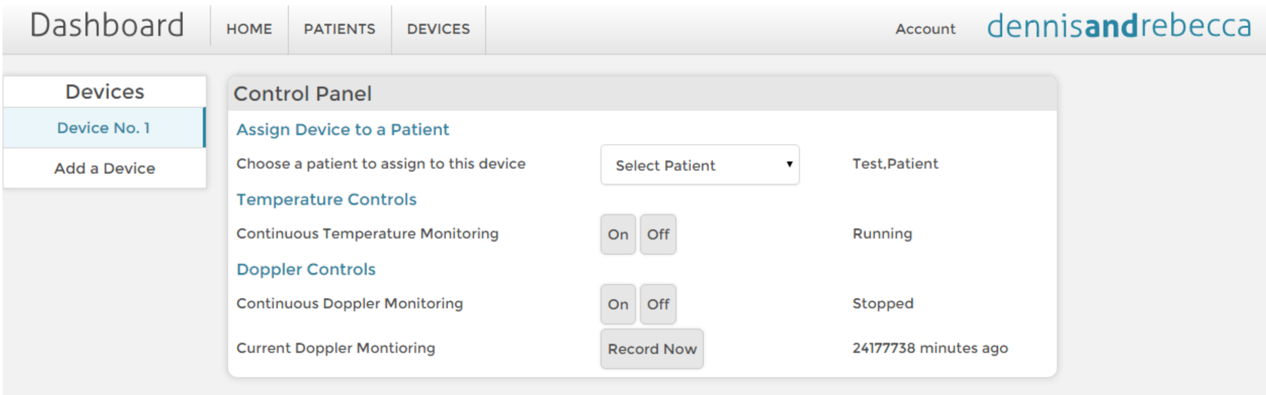

Fig. 4.10 Device control panel from GUI.

4.1.5 Alert the User of Occlusion

Needs Analysis

It is necessary for the device to promptly alert clinicians of any problems developing with the flap. Alerts must quickly reach clinicians and provide actionable information.

Conceptual Design

The device utilizes two different approaches for alerts. The first is a text message alert system to send a text message to the clinician's phone immediately following the detection of any problems such as a temperature discrepancy detected in the flap or the loss of certain characteristics of the Doppler signal. The second system is to log all alerts in the patient monitoring section of the web user interface. Alerts are color coded by priority and time stamped. Low priority alerts are simply time records of whenever recording is started or paused for whatever reason. High priority alerts are when the system detects an abnormality in the data being collected. The clinician will then know if the problem detected is related to the Doppler signal or the temperature probes. They can then check the web user interface for more information such as the graph of all the temperature probes for the patient or directly listen to the most recent Doppler recording or request another recording be taken to listen to. From this information the clinician can make the decision whether the patient will need a more thorough clinical examination or in some cases determine that the patient will need immediate revision surgery.

4.2 Preliminary Testing

4.2.1 Ultrasound Gel Longevity

Preliminary testing was conducted to test the longevity of ultrasound gel on the skin. A dime-sized amount of hypoallergenic ultrasound gel was applied to the skin and covered in a piece of plastic wrap. The plastic was secured to the skin with medical tape, ensuring that the gel was completely sealed against the skin (see Fig. 4.11). The "gel patch" was then worn over the next 24 hours to determine how long the gel would last before absorbing into the skin.

Fig. 4.11 Ultrasound gel longevity test setup.

The presence of the gel under the plastic was checked every two hours and the results were recorded in a table detailing whether the gel was still present or not at the given time interval (see Table 4.2). It was found that the gel had been mostly absorbed into the skin after 8 hours of application, and completely absorbed into the skin after 9 hours.

Table 4.2 Ultrasound gel longevity test results.

Time	Time Since Initial Application (hours)	Gel Status
3:30pm	0	Present (initial application)
5:30pm	2	Present
7:30pm	4	Present
9:30pm	6	Present
11:30pm	8	Mostly absorbed into skin
12:30am	9	Fully absorbed into skin

4.2.2 Thermistor Calibration

Thermistor calibration was completed by measuring the resistance of the thermistors while submerged in a water bath at various temperatures. The 10kΩ thermistor was wrapped in plastic wrap and placed in a water bath alongside a digital thermometer. Both were positioned such that they were fully submerged in the water and not in contact with the sides or bottom surface of the water bath. The terminals of the thermistor was connected to a digital ohmmeter and resistance values were obtained for temperature values for every 0.1°C ranging from 31°C to 38.5°C. The values were plotted and an equation was obtained for the calibration curve (see Fig. 4.12).

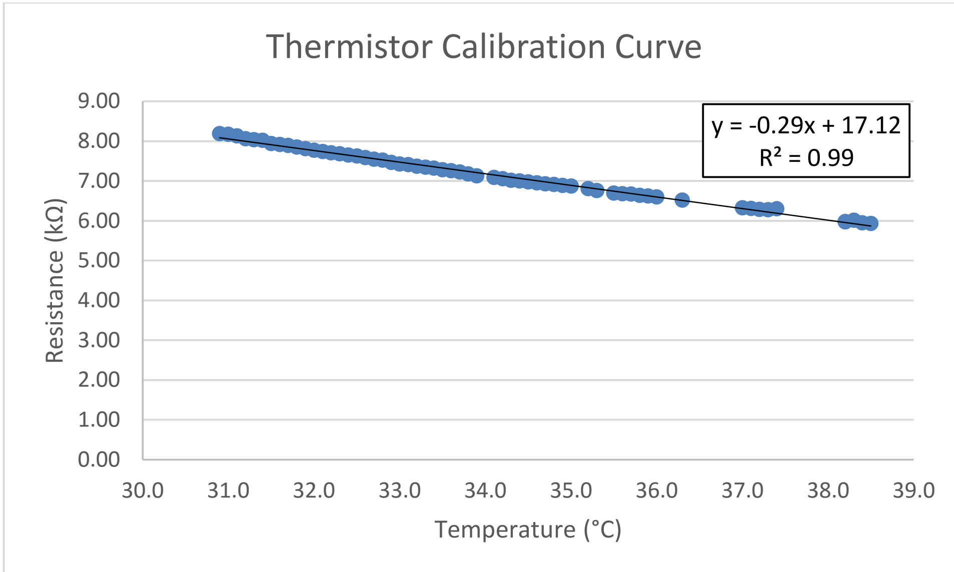

Fig. 4.12 Thermistor calibration curve.

Unable to achieve the desired sensitivity with the 10kΩ thermistor, the device was modified to 100kΩ thermistors. Once all four temperature probes were assembled, another calibration was performed by wrapping the thermistor end of the probes tightly in plastic such that they were beside each other and submerging the set of probes in a water bath. The temperature of the water bath was decreased from 42°C to 30°C and measurements of the ADC value were recorded for each increment of 0.1°C (e.g. 42.0°C, 41.9°C, 41.8°C, etc.). Fig. 4.13 depicts the results of the calibration for each of the four probes, along with the corresponding equation derived from each probe. Each point corresponds to a value measured and the dotted line depicts the linear least squares trend line fitted to the data. According to the slopes of the trend lines, the end sensitivity of this set-up is approximately 10-bits per ºC, or 0.1ºC per bit.

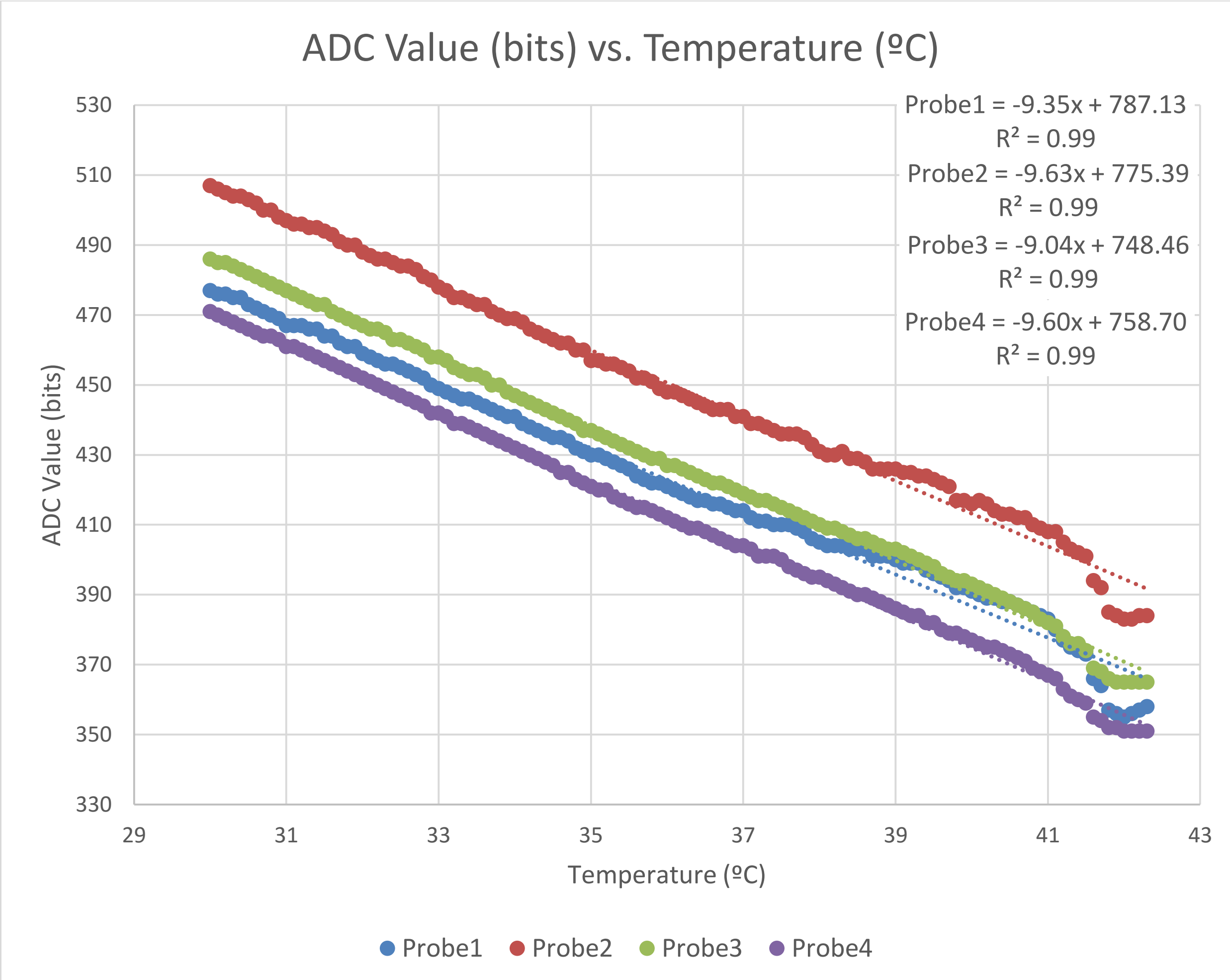

Fig. 4.13 Calibration of four fully assembled temperature probes.

4.3.3 Preliminary Doppler Ultrasound Data

Preliminary testing was conducted with the prototype of the device to test the Doppler ultrasound recording capabilities and to collect initial data to use for the development of signal processing algorithms. Participants were instructed to lie in the supine position in an examination chair and a standard blood pressure cuff was secured around the participant's bicep. Parks Medical flat Doppler probes were used to obtain radial artery and cephalic vein signals. Recordings were first collected from the radial artery. The cuff pressure was increased to the participant's systolic blood pressure (SBP) until there was a complete loss of arterial pulse, and a 5-second recording of the Doppler signal was collected. The cuff pressure was then decreased in increments of 20 mmHg to 0 mmHg and a 5-second recording was taken at each point. Cephalic vein recordings were done similarly from 40 mmHg down to 0 mmHg in increments of 10 mmHg with a 5-second recording at each point.

The spectrograms of these recordings for the first patient are depicted in Fig. 4.14-4.21 below to highlight the changing time-frequency components of these signals as occlusion increases. In the first recording in Fig. 4.14 below with no occlusion it is possible to see the venous signal present in the low frequency range from 0Hz to approximately 50Hz across the duration of the recording. Then the periodic spikes with each heartbeat are the prominent feature of the arterial signal. At 20mmHg in Fig. 4.15 it is possible to see this venous component becoming less pronounced.

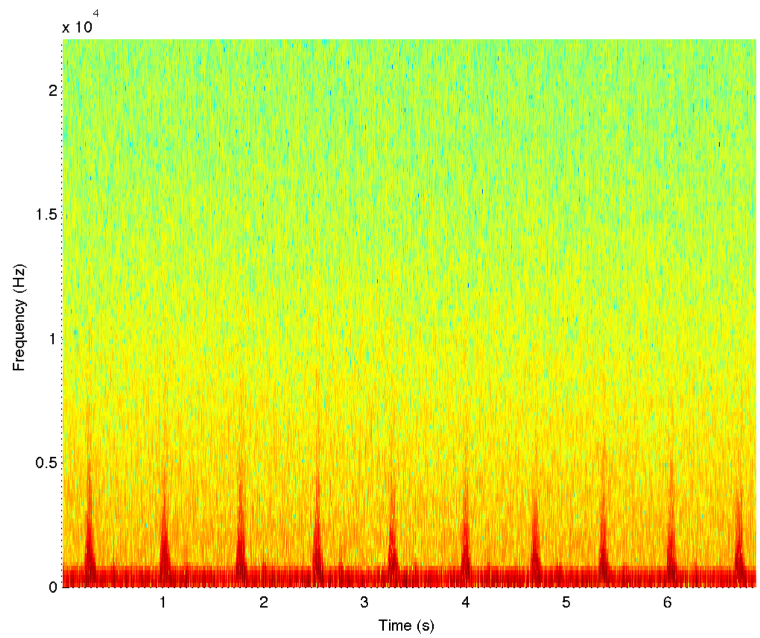

Fig. 4.14 Radial artery spectrogram 0mmHg.

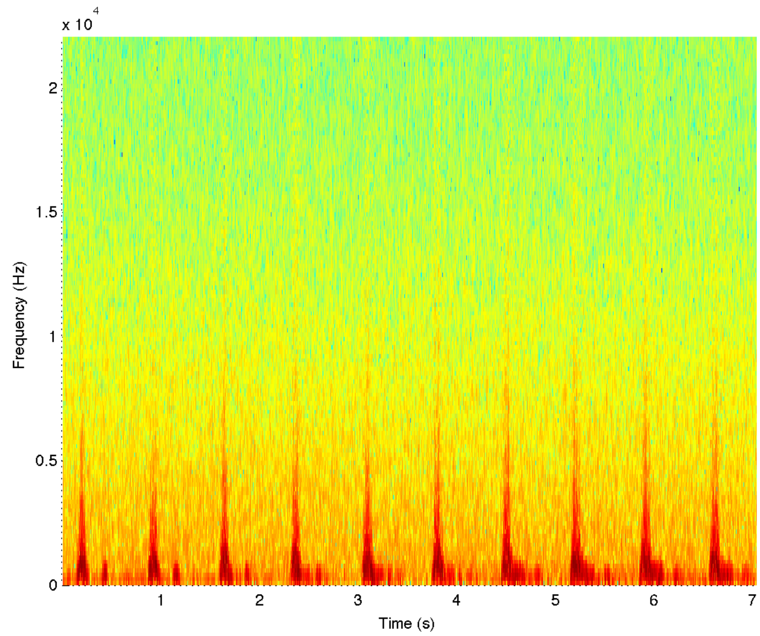

Fig. 4.15 Radial artery spectrogram 20mmHg.

In Fig. 4.16 and Fig. 4.17 below the arterial signal is gone but it is still possible to see the triphasic component of the arterial signal as pressures have not yet gotten close to systolic blood pressure to impede arterial flow.

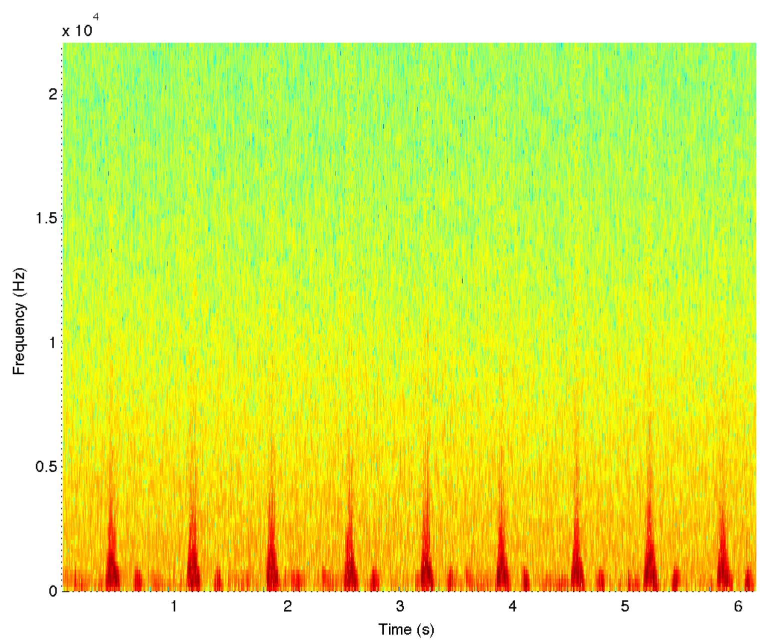

Fig. 4.16 Radial artery spectrogram 40mmHg.

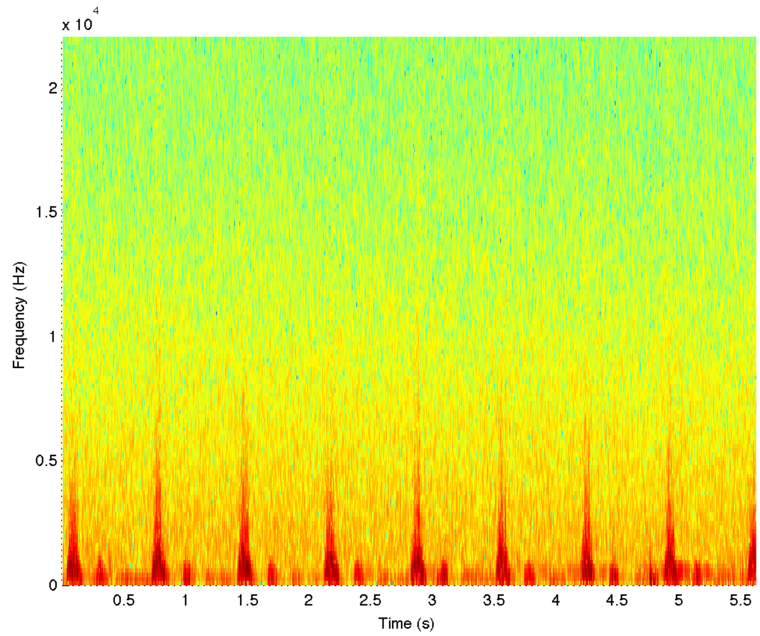

Fig. 4.17 Radial artery spectrogram 60mmHg.

In Fig. 4.18 and Fig. 4.19 below the signal is biphasic still but only the first pulse is really pronounced as the arterial flow is impeded at pressures nearing systolic blood pressure.

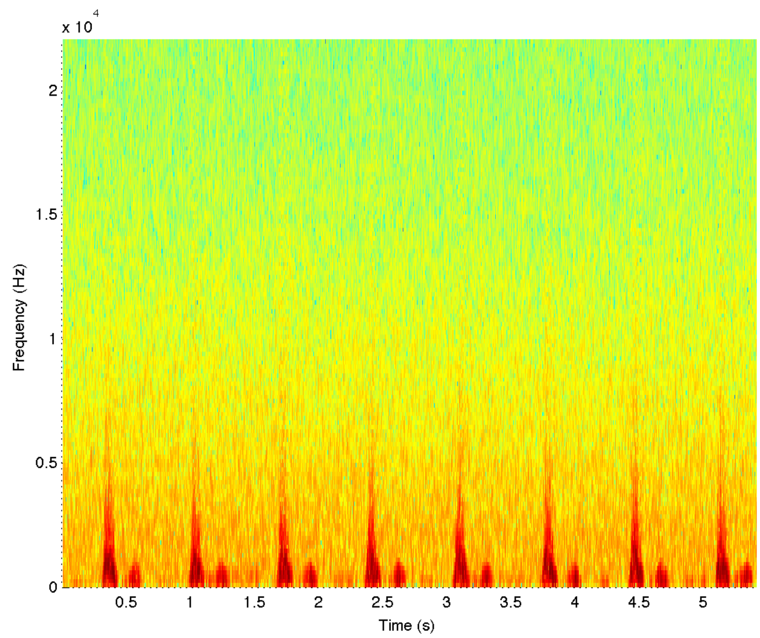

Fig. 4.18 Radial artery spectrogram 80mmHg.

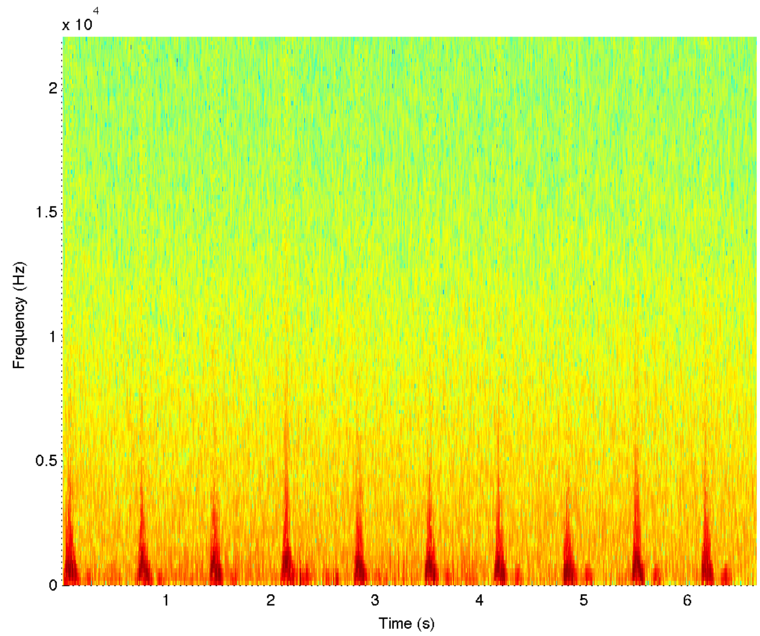

Fig. 4.19 Radial artery spectrogram 100mmHg.

In Fig. 4.20 and Fig. 4.21 below it is evident that the flow is severely impeded. Figure 4.20 still shows some flow where the signal is monophasic but the intensity of the peak is not as high in frequency as some of the less occluded signals. The frequency here corresponds to the velocity of the flow so it is expected that a lower velocity flow would be observed. In Fig. 4.21 the signal is completely lost at pressures above the total occlusion pressure.

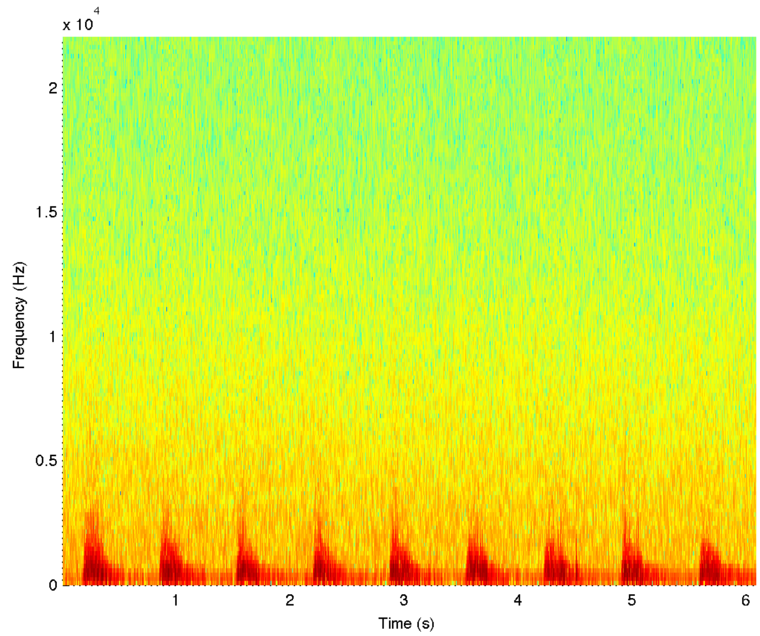

Fig. 4.20 Radial artery spectrogram 120mmHg.

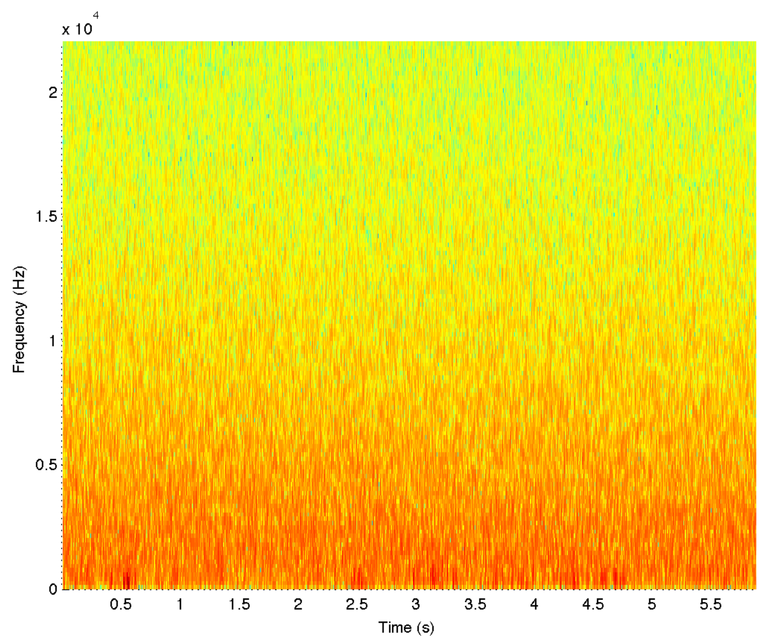

Fig. 4.21 Radial artery spectrogram 140mmHg.

Figures 4.22-4.29 below depict the power window signal computed for the same radial artery Doppler signals at varied levels of occlusion. In Fig. 4.22 below it is possible to see both the prominent arterial pulses of the signal and the venous component of the signal depicted as a baseline greater than zero between arterial pulses. In Fig. 4.23 where the cuff pressure has increased to 20mmHg the venous component of the signal is no longer present however the arterial pulses remain relatively unchanged.

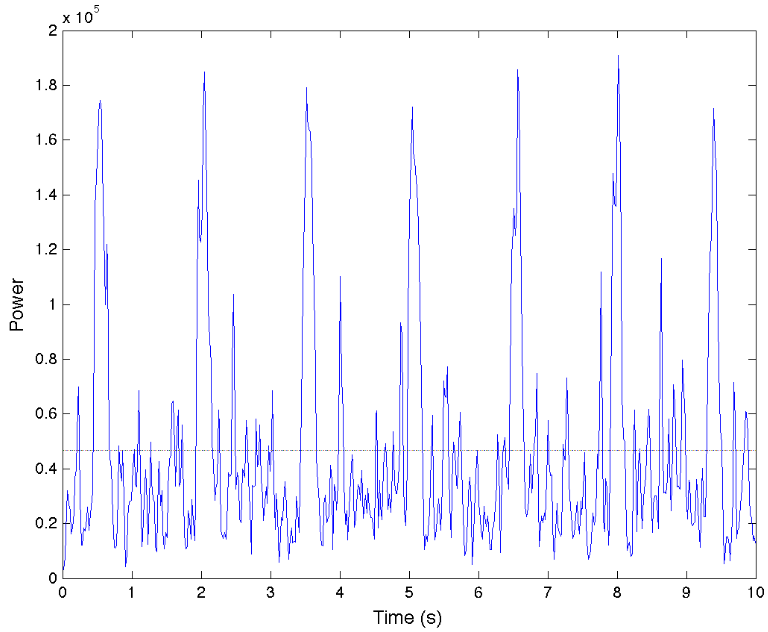

Fig. 4.22 Radial artery power window 0mmHg.

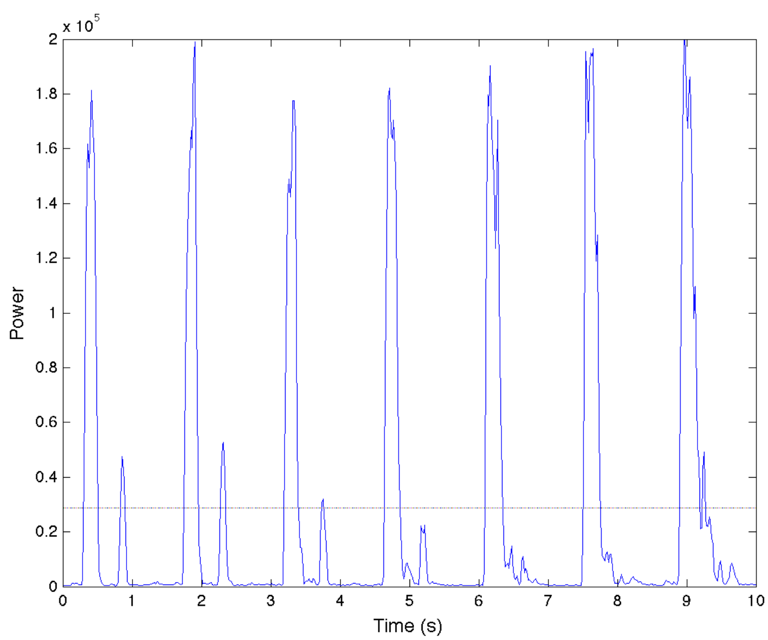

Fig. 4.23 Radial artery power window 20mmHg.

In Fig. 4.24-4.26 below the power window depicts the characteristic biphasic nature of the arterial signal at pressure levels from 40-80mmHg. In Fig. 4.27 the signal clearly switches from biphasic to monophasic at 100mmHg.

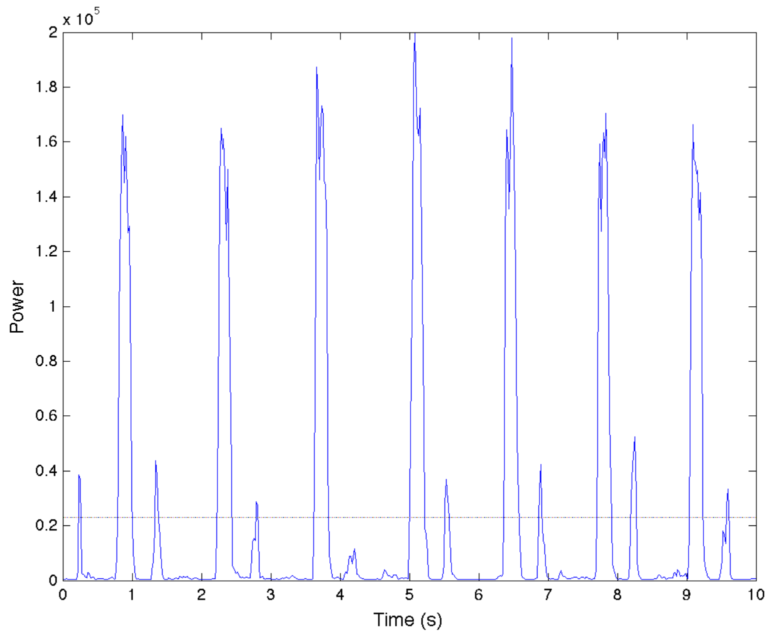

Fig. 4.24 Radial artery power window 40mmHg.

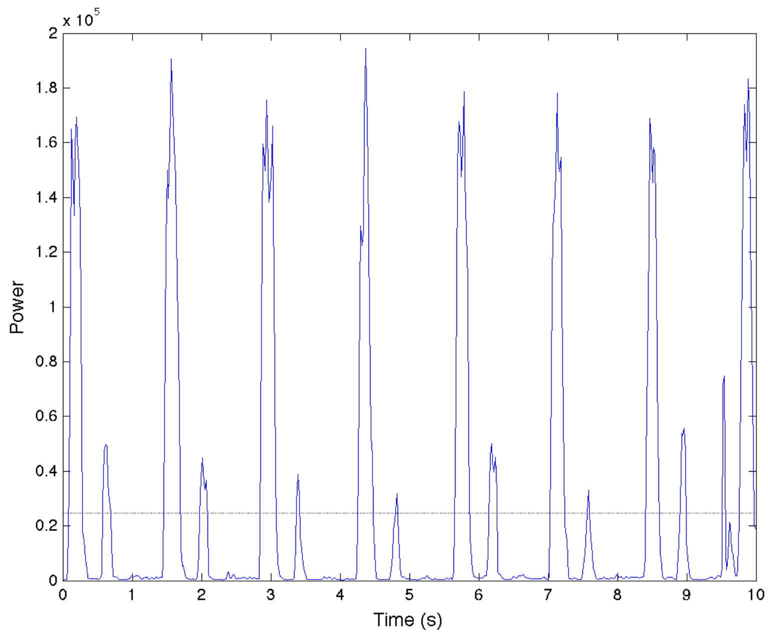

Fig. 4.25 Radial artery power window 60mmHg.

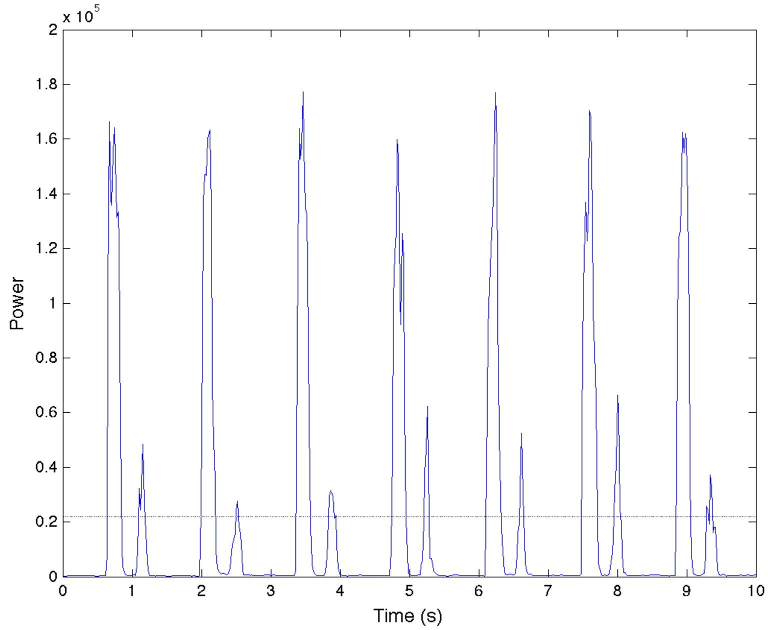

Fig. 4.26 Radial artery power window 80mmHg.

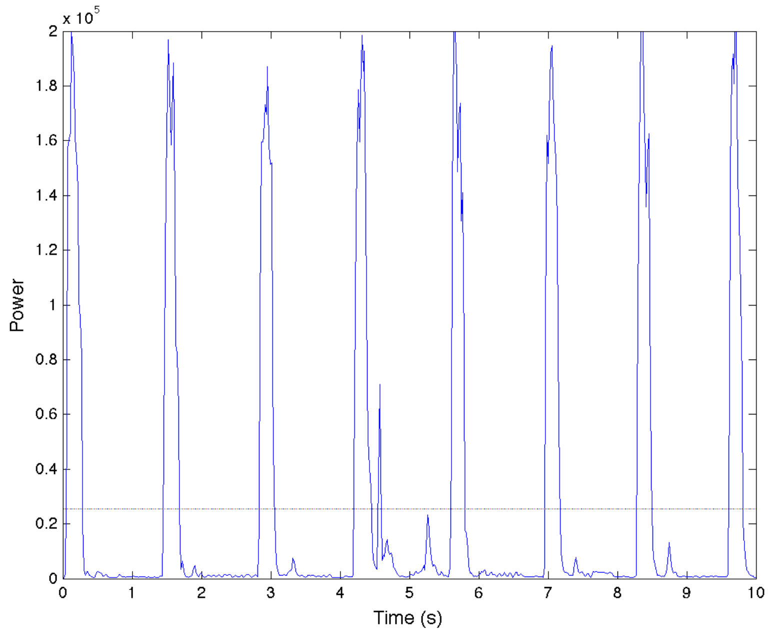

Fig. 4.27 Radial artery power window 100mmHg.

In Fig. 4.28 below the signal is monophasic and the peak width has slightly increased. In Fig. 4.29 there is total occlusion and the signal is no longer visible on the same scale.

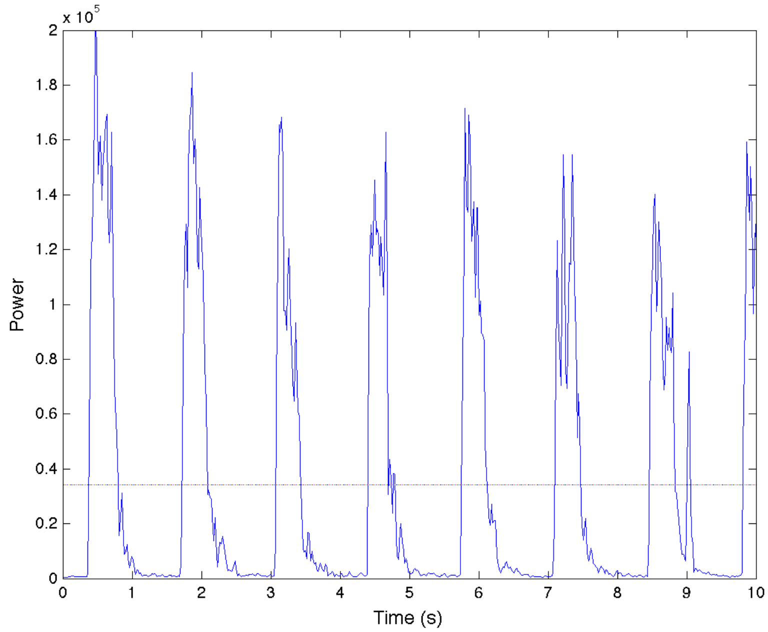

Fig. 4.28 Radial artery power window 120mmHg.

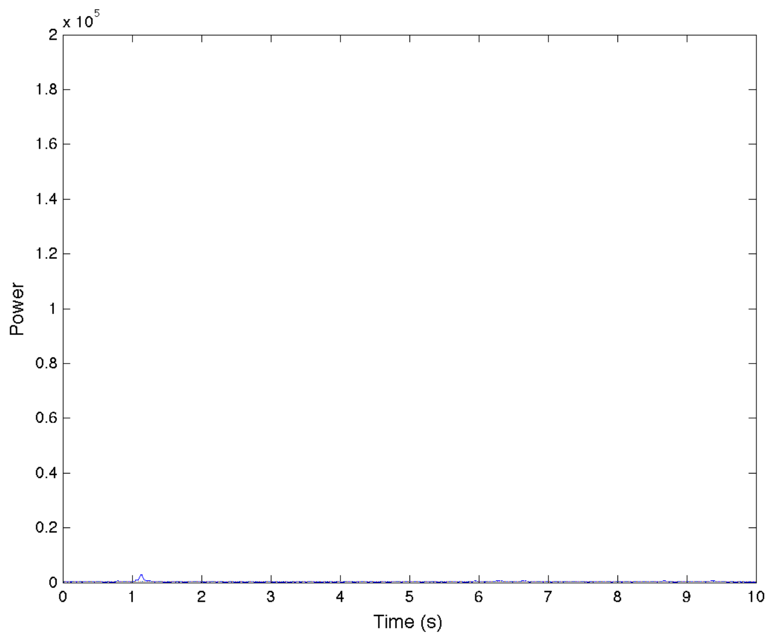

Fig. 4.29 Radial artery power window 140mmHg.

Chapter 5: Design Verification

This chapter presents the outcomes of experimentation that was used to evaluate the performance of the Doppler ultrasound, temperature, and light monitoring functions of the device. The final section for overall design verification includes results pertaining to the overall function of the device and associated components and their compliance with relevant medical device standards.

5.1 Doppler Ultrasound

5.1.1 Controlled Study

To verify that the device was capable of recording adequate Doppler ultrasound signals a study was conducted with healthy volunteer subjects. Participants were instructed to lie in the supine position in an examination chair. A standard blood pressure cuff was fixed around the participant’s upper arm. Combined signal from the radial artery and cephalic vein were collected using a Parks Medical flat Doppler probe housed in the probe holder (see Fig. 5.1). After confirming the quality of the signal at each placement location, the probes were taped in place against the skin using medical tape. Recordings were first collected from the radial artery. The cuff pressure was increased to 200mmHg in increments of 40mmHg and a 5-second recording of the Doppler ultrasound signal was collected for each pressure. The cuff pressure was then released upon reaching 200mmHg. A 5-second Doppler ultrasound recording was taken as the cuff was released and once the pressure had returned to 0mmHg.

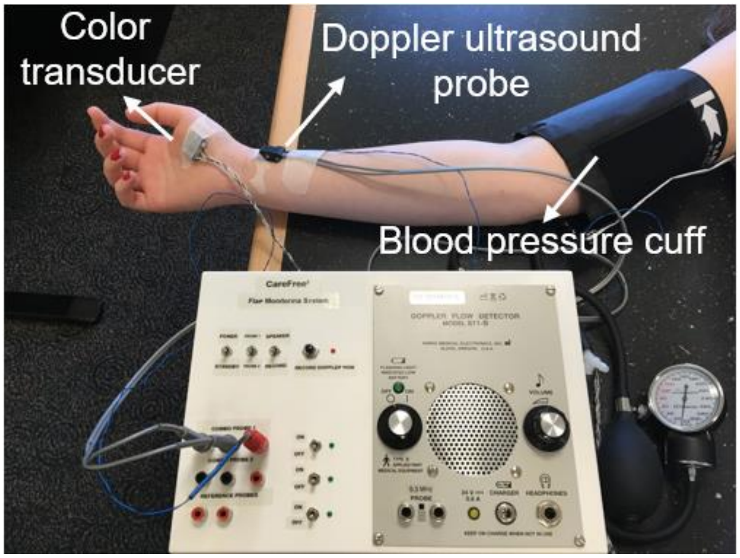

Fig. 5.1 Setup for controlled study acquiring Doppler ultrasound and color signals.

Figures 5.2-5.7 below depict the results of the controlled Doppler ultrasound study. The data has undergone novel signal processing methods that allow for detection of and differentiation between venous and arterial occlusion. The power signals were plotted for each of the pressures (from 0mmHg to 200mmHg and subsequent deflation).

Prior to inflation (Fig. 5.2), the signal is unobstructed with a high baseline indicative of steady venous flow and healthy arterial pulses. At 40mmHg (Fig. 5.3), venous flow has become occluded, which corresponds to the significant drop in the baseline signal. Arterial pulses are still present with biphasic peaks (the first peak is followed by a second, smaller peak), which indicates healthy arterial signal.

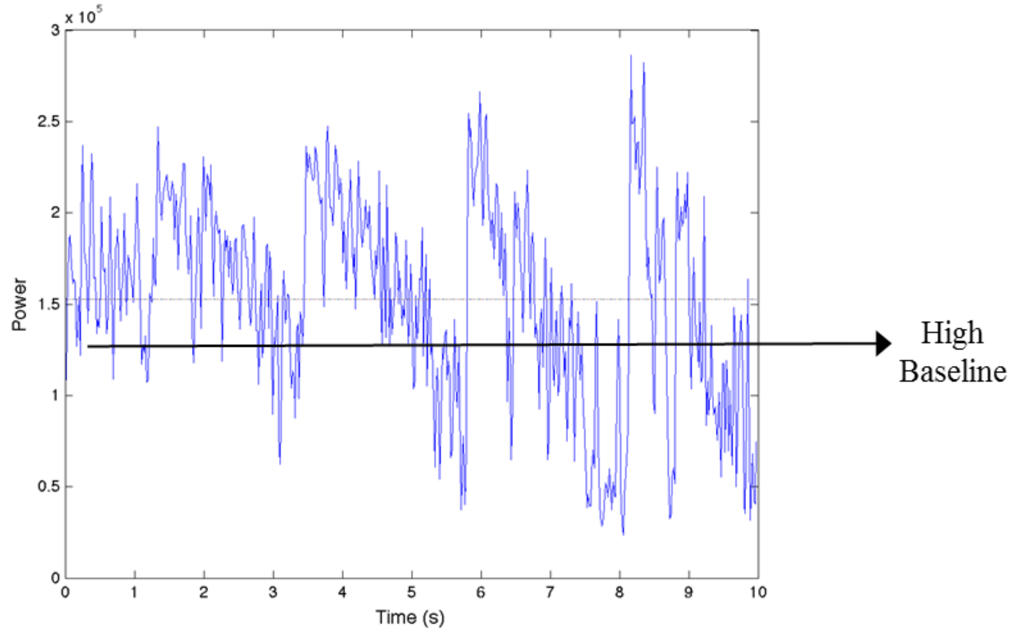

Fig. 5.2 Subject radial power signal at 0mmHg.

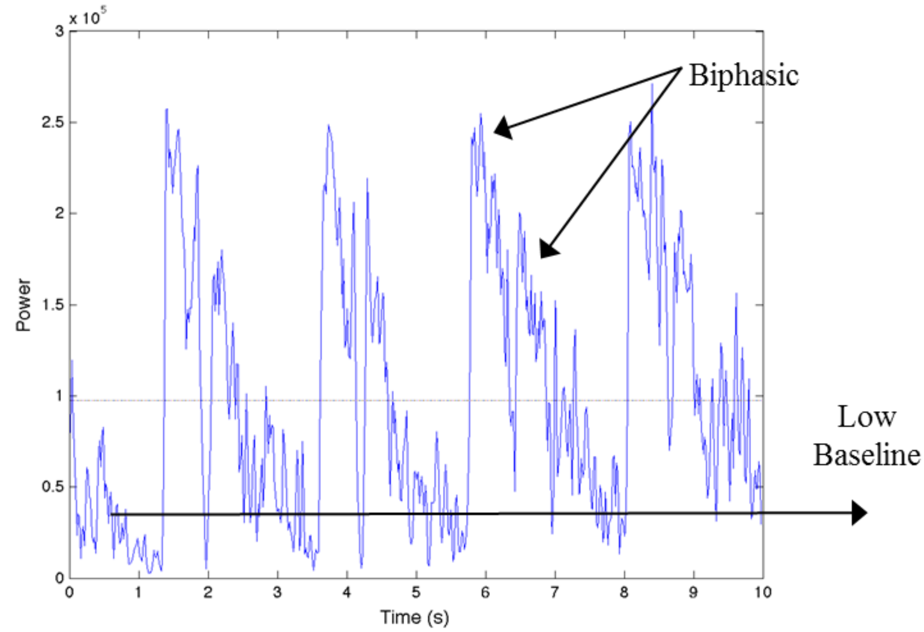

Fig. 5.3 Subject radial power signal at 40mmHg.

At 80mmHg (Fig. 5.4), arterial flow starts to become compromised as the cuff reaches systolic blood pressure. The monophasic peaks (single peak with little/no secondary peak) are indicative of weak arterial signal. At 120mmHg (Fig. 5.5), blood flow to the arm is entirely occluded and the signal disappears entirely. This persists for all pressures above 120mmHg, where the cuff is above systolic blood pressure.

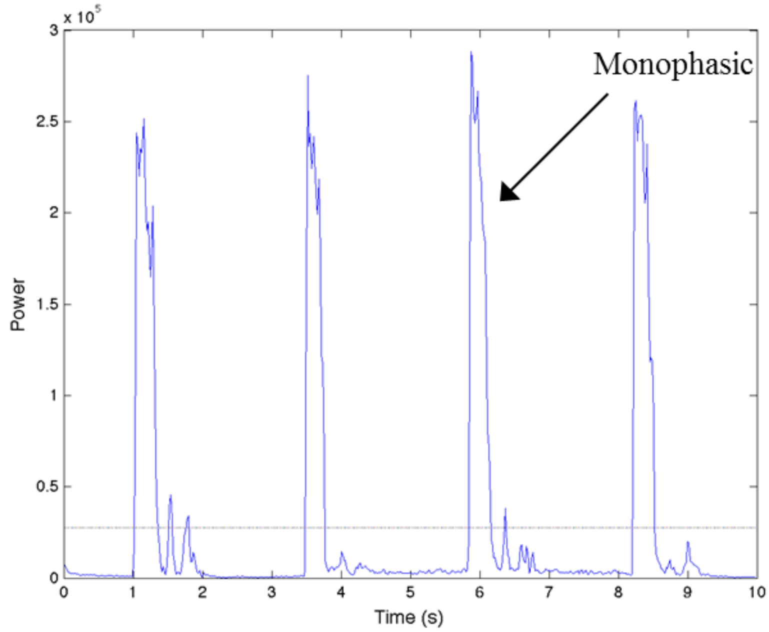

Fig. 5.4 Subject radial power signal at 80mmHg.

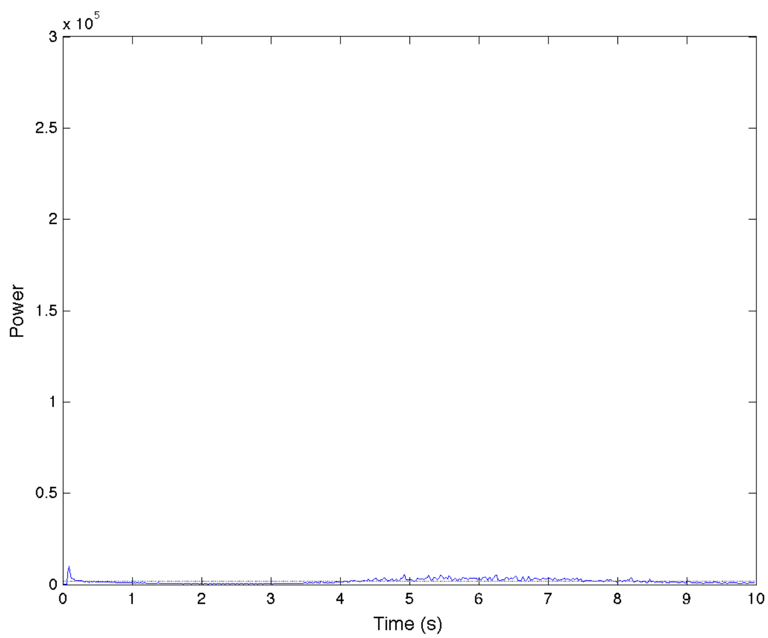

Fig. 5.5 Subject radial power signal at 120mmHg.

Upon releasing the pressure in the cuff, there is a return of arterial pulse (Fig. 5.6) and a subsequent return of venous flow (Fig. 5.7).

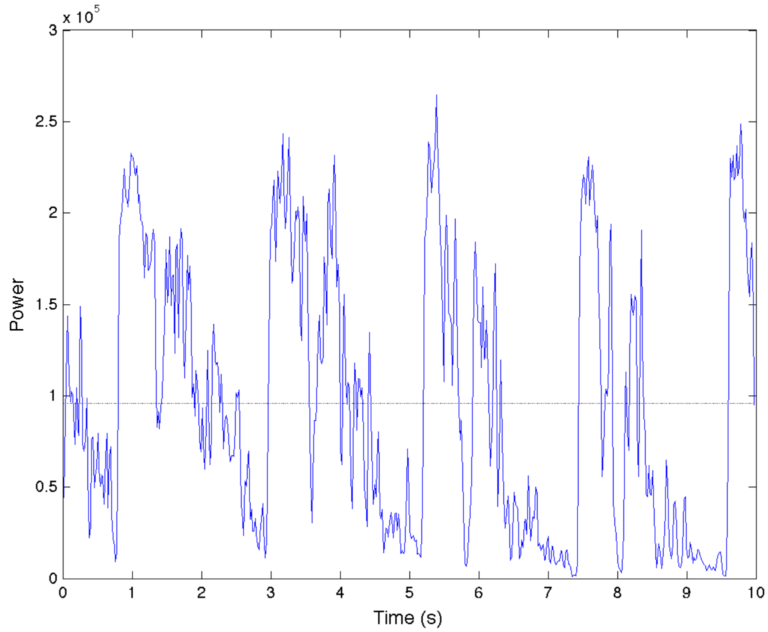

Fig. 5.6 Subject radial power signal at deflation.

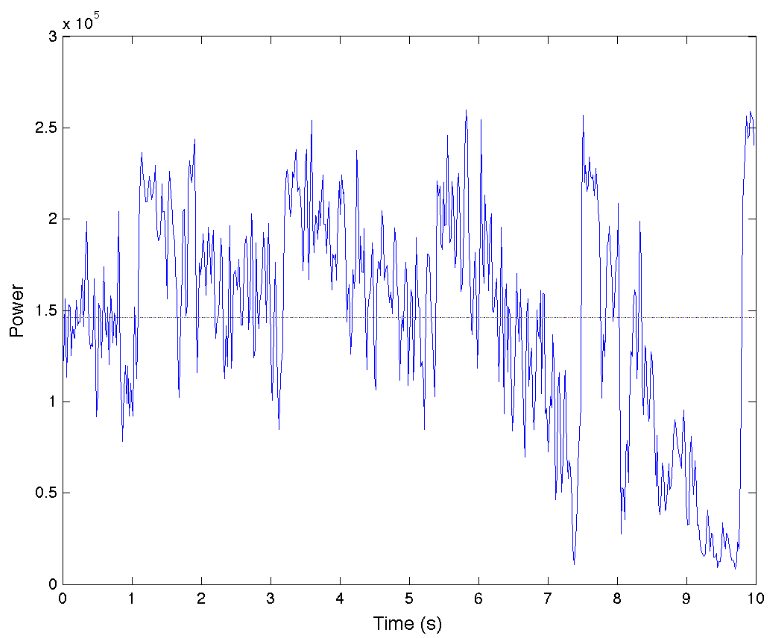

Fig. 5.7 Subject radial power signal after deflation.

5.1.2 Patient Trial

After successfully completing Doppler ultrasound trials on healthy volunteer subjects, the device was tested on a patient who had undergone a bilateral breast reconstruction. A flap Doppler ultrasound probe was placed over the breast flap, and a mixed arterial-venous signal was obtained. When compared to the signal from a healthy volunteer (Fig. 5.8), the signal collected from the patient had the same characteristic shape, including the high baseline corresponding to venous flow and the biphasic peaks corresponding to arterial pulse (Fig. 5.9). The ability of the system to obtain similar, high quality Doppler signal from flaps of different sizes and locations indicates that the detection algorithms can be applied to numerous types of flaps.

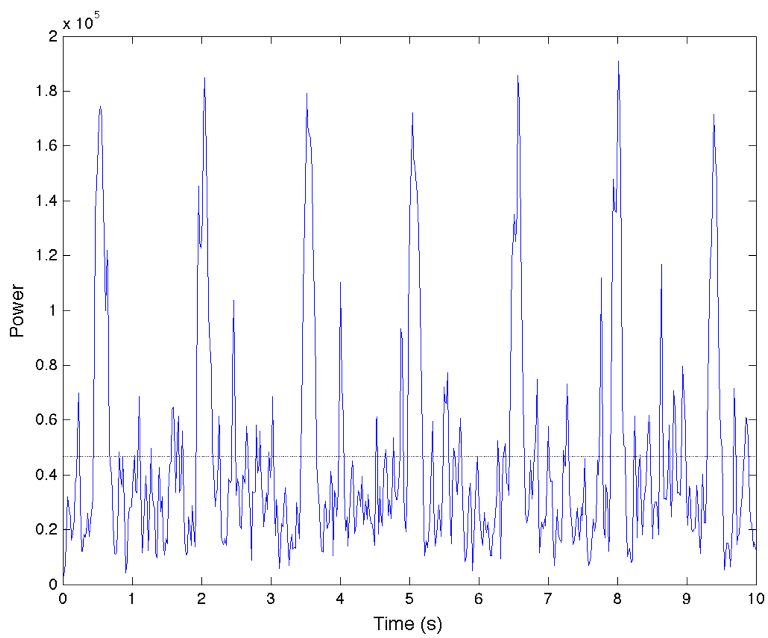

Fig. 5.8 Power signal from healthy subject.

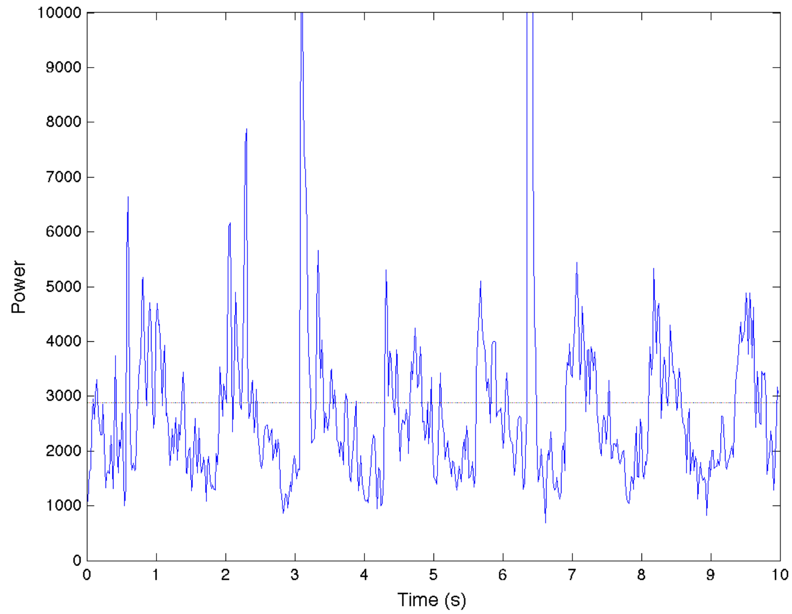

Fig. 5.9 Power signal from postoperative patient.

5.2 Temperature

The device must accurately measure small differences in temperature between probes that are located on the transplanted tissue and those that are located as a reference on nearby healthy well-perfused tissue. To verify that the probes reported the temperature within 0.1ºC agreement, extended recordings were taken with all four probes in a temperature controlled water bath at three different temperatures. Recording times were five minutes in duration and the bath was set to 34ºC, 38ºC, and 42ºC according to an independent measurement made by the Fischer Scientific ISOTEMP205 water bath incubator and AcuRite digital thermometer. The results of the data collected are summarized in Table 5.1-5.3 below.

Table 5.1 Temperature calibration statistical analysis at 34ºC (n=300).

Target 34ºC		
	Average	STDEV
Probe 1	34.17	0.006
Probe 2	34.11	0.0174
Probe 3	34.15	0.0096
Probe 4	34.19	0.0094

Table 5.2 Temperature calibration statistical analysis at 38ºC (n=300).

Target 38ºC		
	Average	STDEV
Probe 1	37.68	0.033
Probe 2	37.77	0.0314
Probe 3	37.75	0.0294
Probe 4	37.74	0.0263

Table 5.3 Temperature calibration statistical analysis at 42ºC (n=300).

Target 42ºC		
	Average	STDEV
Probe 1	41.92	0.0747
Probe 2	41.97	0.0705
Probe 3	42.05	0.0679
Probe 4	41.98	0.0679

5.3 Color

An identical procedure to that used for Doppler ultrasound was used to evaluate the effectiveness of the light transducer's ability to detect sensitive changes in the tissue reflectance of red, blue, and green wavelengths with varying levels of perfusion. One transducer was placed over the thenar of the subject, as depicted by Fig. 5.1, and continuous data was collected for the duration of the cuff inflation and subsequent deflation. Fig. 5.10 depicts the four raw channels recorded from the transducer indicating the reflected amounts of white, red, green, and blue light. The transducer was able to detect sensitive pulsatile changes in blood volume corresponding to each heartbeat. Furthermore, it was capable of detecting changes in red and green absorption when the flow is occluded that are not visible to the naked eye.

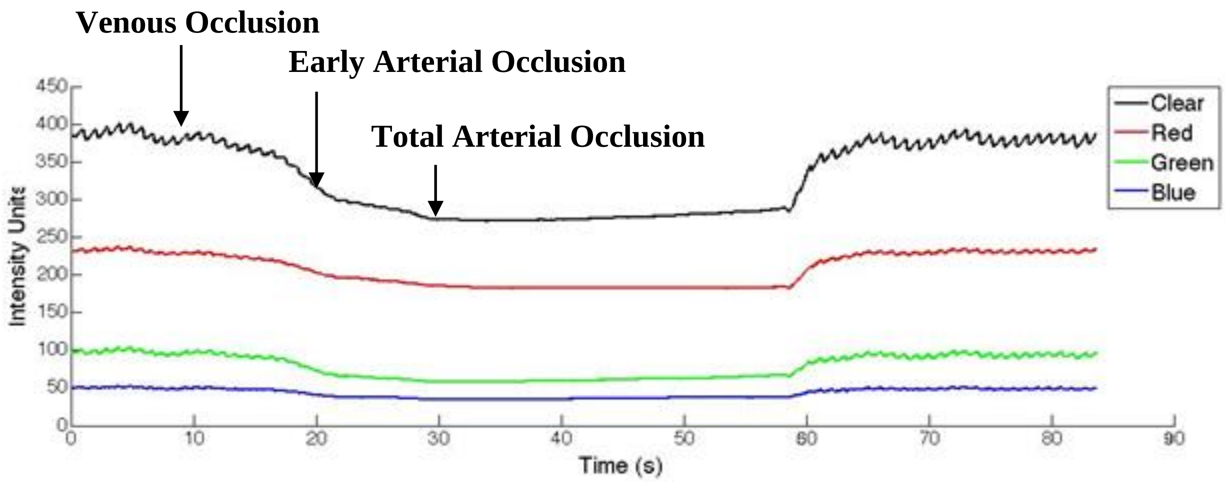

Fig. 5.10 Absorption of white, red, green, and blue light during simulated venous and arterial compromise.

Chapter 6: Final Design and Validation

6.1 Final Design

6.1.1 Temperature

Analog temperature signals are acquired from 100kΩ NTC thermistors and digitized by an ADC on-board the Arduino Uno with a 10-bit resolution for temperature recordings sensitive to 0.1ºC (see Fig. 6.1). The temperature probes are also carefully calibrated for agreement across all four probes to 0.1ºC so that an accurate determination of the temperature differential between the transplant and surrounding tissue can be made.

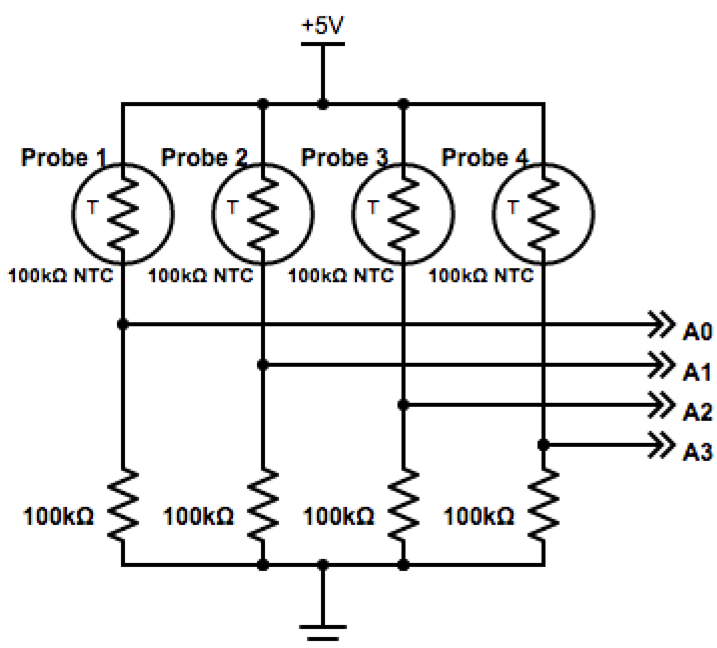

Fig. 6.1 Thermistor circuit schematic to Arduino analog input pins.

6.1.2 Probe Holder

The final holder for the combination Doppler-temperature probes was 3D printed from the design outlined in Section 4.1.1.3. The piece was printed from ABS plastic and fixed with the

Doppler probe and thermistor. The ultrasound-gel catheter was also fixed in place in the holder (see Fig. 6.2).

Fig. 6.2 Combination probe holder with flat Doppler probe, thermistor, and ultrasound gel catheter fixed in place.

6.1.3 Color

Color of the flap is assessed using the TCS34725 RGB color sensor and a white LED (see Fig. 6.3). The two components are packaged together along with signal amplification and digital to analog converters in a small PCB board component available from Adafruit. The Arduino communicates with this unit over the I2C connection and takes readings of the amounts of light through red, green, blue, and clear filters being received by the sensor after being emitted from the LED and reflected by the tissue.

Fig. 6.3 TCS34725 RGB color sensor with white LED.

6.1.4 Case

The final case for the device was drawn in SolidWorks and 3D printed. The case consists of two separate pieces that fit together to form a box around all of the components. The bottom piece of the case was developed to hold all of the circuitry in place, including spots for the Arduino, Raspberry Pi, and breadboard (see Fig. 6.4).

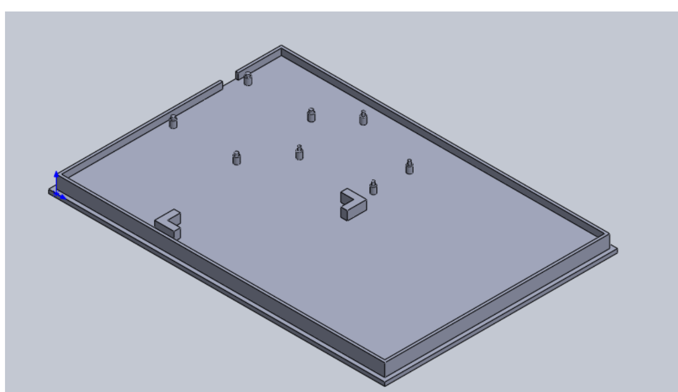

Fig. 6.4 Bottom piece of device case in SolidWorks with mounting for Arduino, Raspberry Pi, and breadboard.

The top piece of the case was printed with a series of holes for the jacks, switches, buttons, and lights (see Fig. 6.5). Additionally, a large space was left for the panel of the Doppler flow meter. The top face of this piece of the case serves as a control panel for the user, containing all of the necessary components for managing the status of the device. Using these controls, the user can turn on monitoring for each of the probes, record an instant Doppler signal, and switch on the speaker of the Doppler flow meter.

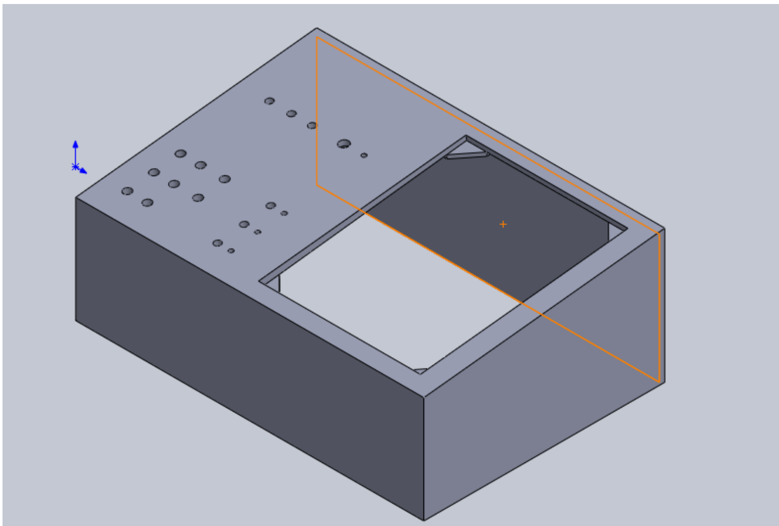

Fig. 6.5 Top piece of device case in SolidWorks with holes for electrical components and Doppler panel.

Fig. 6.6 depicts the final version of the device case with all of the components integrated into the control panel. The switches and buttons that control general device function can be seen on the left side of the panel, and the Doppler flow meter component can be seen on the right.

Fig. 5.6 Final bedside device with integrated Doppler flow meter and components.

6.1.5 Website

The final website was developed for mobile in HTML5 with the Bootstrap framework (v3.3.6) and an additional CSS stylesheet for managing other stylistic aspects of the interface. A functional login system was developed that restricts access to registered members. This is managed by a user MYSQL table in the system database (see Fig. 6.7).

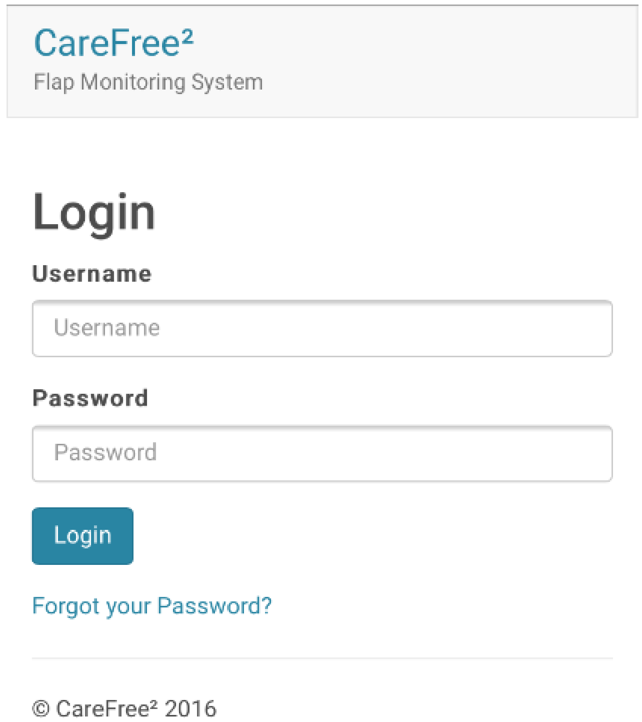

Fig. 6.7 Login screen on mobile user interface.

From the main portal, the user can access both patient and device information. The patients section of the application lists all of the patients currently enrolled in the system (which can be added and deleted from the database directly from the app). Each patient has a page associated with him or her that displays data from the device in real time. From this page, the user can access patient alerts (Fig. 6.8), temperature graphs (Fig. 6.9), and Doppler ultrasound recordings (Fig. 6.10) that have remotely been uploaded to the interface via the device. Furthermore, the user can check the status of the device and control the system remotely from the app (see Fig. 6.11).

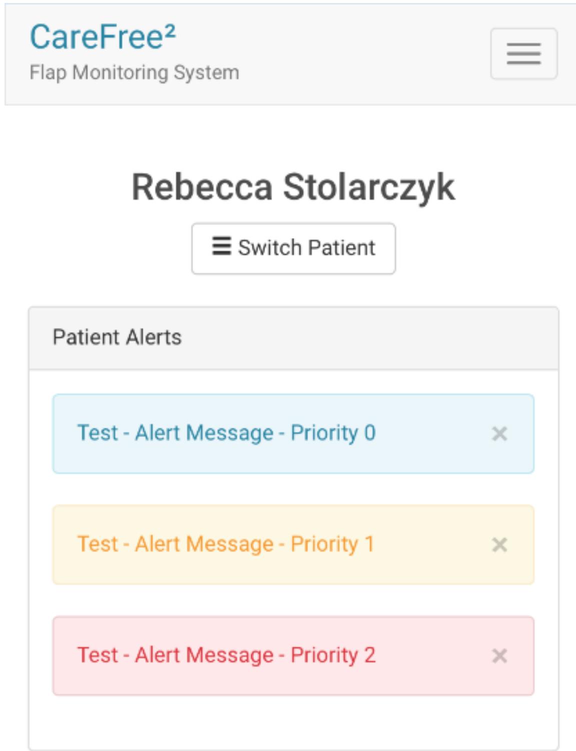

Fig. 6.8 Patient page with dismissible patient alerts of varying priorities displayed.

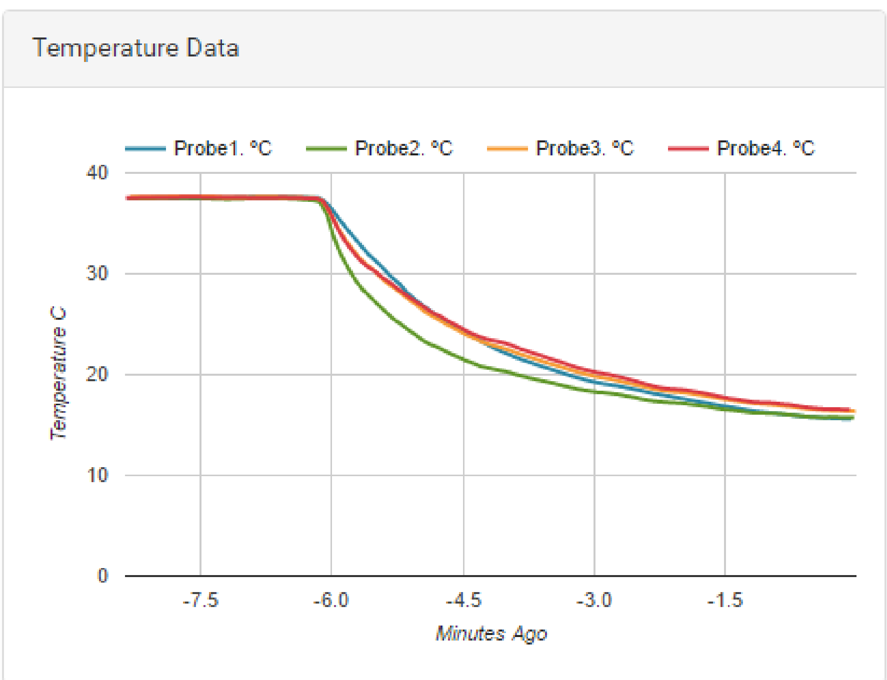

Fig. 6.9 Temperature data from website graph with curves from all four probes.

Fig. 6.10 Doppler ultrasound audio player and visualizer.

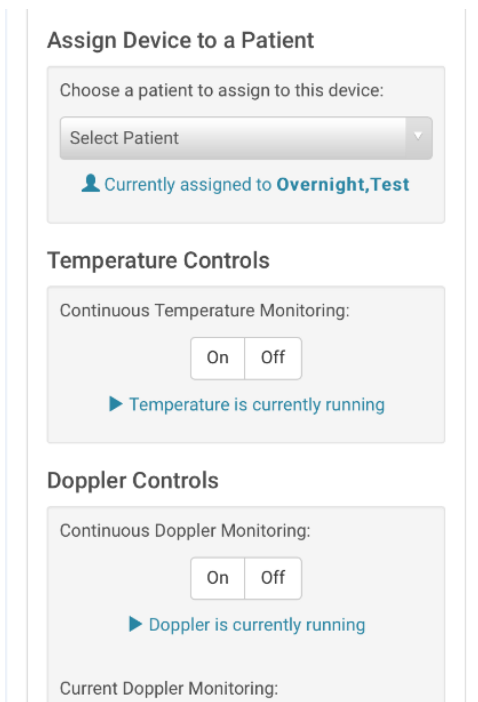

Fig. 6.11 Control panel on mobile interface.

6.1.6 Doppler ultrasound

In the final system, Doppler ultrasound signals are acquired from the Parks Medical 811-B Doppler flow meter. In order to monitor two probes from the single input of the Doppler flow meter, the system contains the multiplexing circuitry comprising two DPDT-Relays controlled from digital output pin 6 on the Arduino via a current sinking transistor (see Fig. 6.12) The status of digital pin 6 allows the system to select between Flap Probe 1 or Flap Probe 2 when making recordings. Signals are recorded directly from the audio output of the Doppler flow meter through a Syba C-Media USB sound card interfacing with the Raspberry Pi.

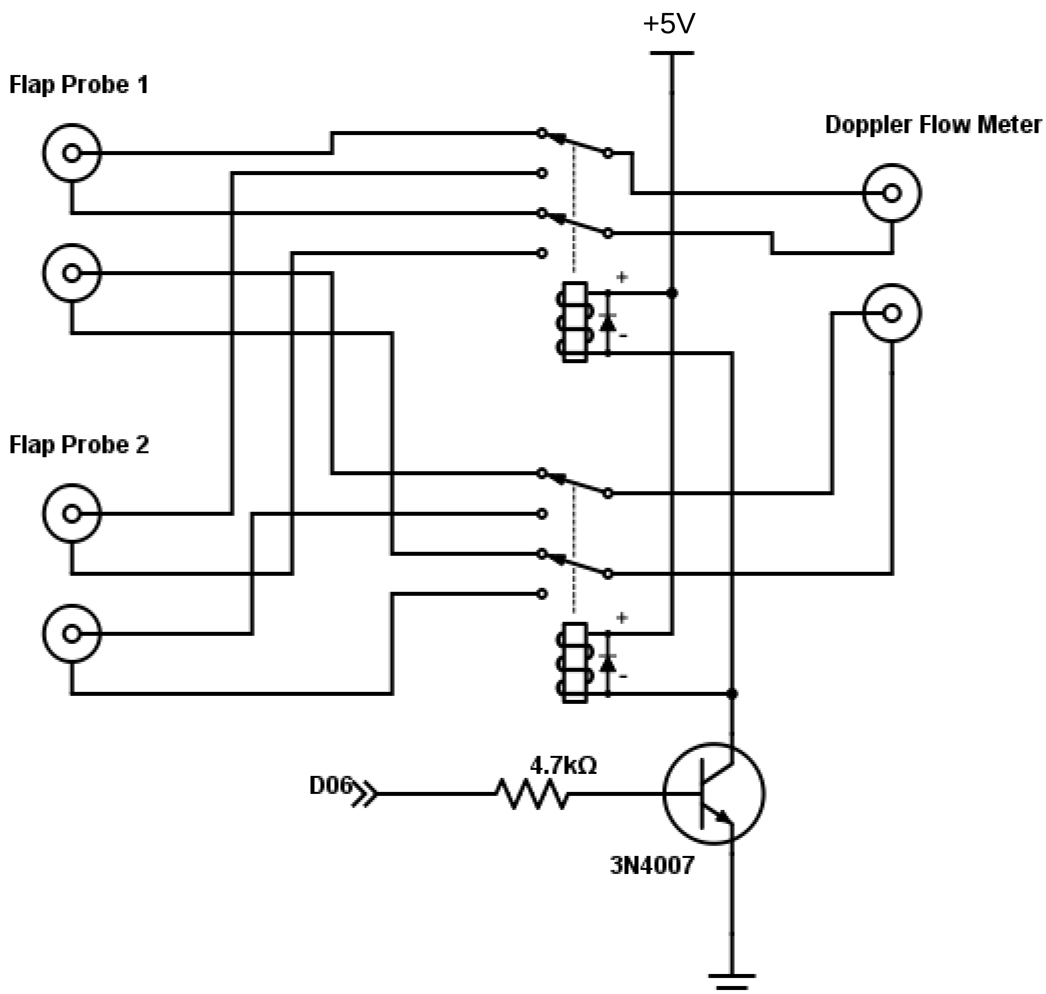

Fig. 6.12 Flap probe multiplexor.

6.1.7 System

The coordination of the system is performed by a Raspberry Pi embedded computer. This unit coordinates the data collection polling sensors from the Arduino as well as making recordings of the Doppler signal through a USB sound card. The Raspberry Pi also performs the data analysis and generates alerts for a given patient and transmits data to the remote server to be accessible from the web application (see Fig. 6.13). The code for this system is contained within Appendix A.

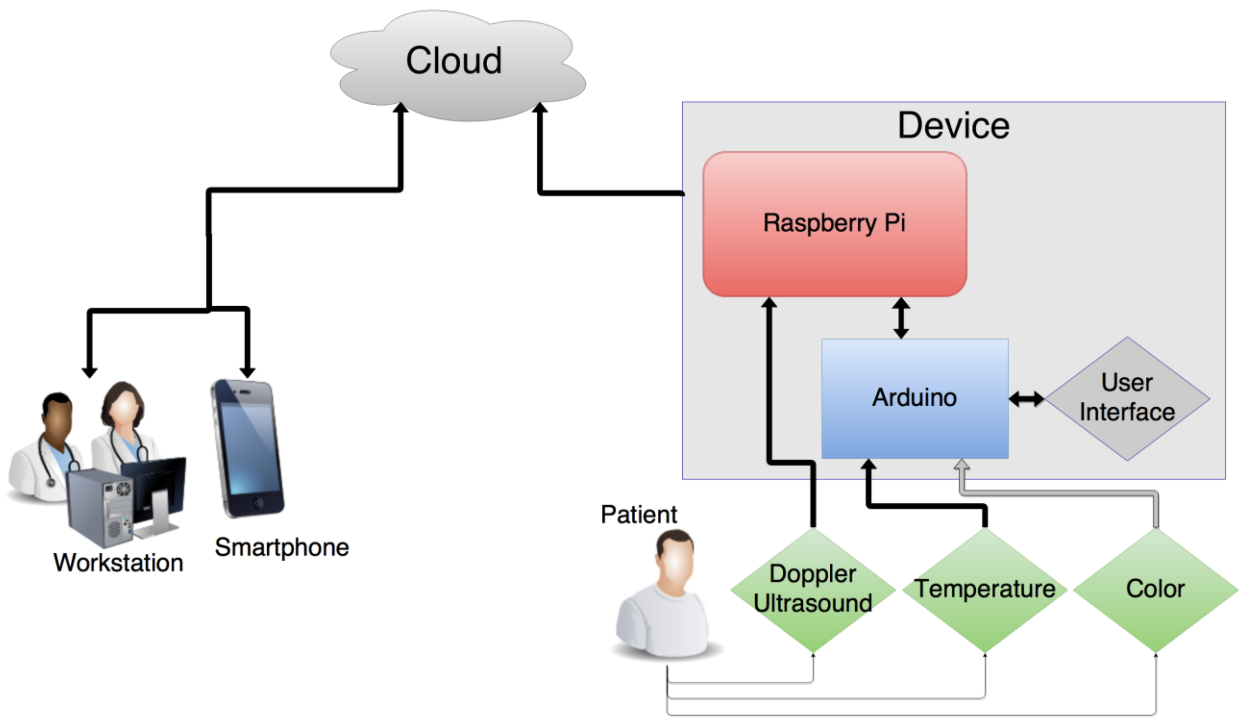

Fig. 6.13 Overview of completed system.

6.2 Impact

6.2.1 Economics

The economic impact of this device would primarily affect three groups: hospitals, patients, and insurance companies. The use of the device in the ICU would eliminate the need for hourly clinical examinations by nursing staff, thereby freeing up time for the staff to monitor other patients. By automating processes that were previously required to be completed manually by clinicians, hospitals could potentially save money in four ways: reduction of equipment, reduction of time spent in the ICU, reduction of staff, and reduction of overtime payment. While the latter two pose considerable ethical issues (see Section 6.5), from a financial standpoint the device would serve to benefit the hospital's budget. If such a system was capable of continuous, remote monitoring with minimal-to-no need for regular clinical examination, existing monitoring equipment such as bedside Doppler ultrasound components and supplemental flap monitoring systems could be replaced with a single, reusable device that effectively monitors multiple parameters at once. Furthermore, the portability of the device would enable patients to be transferred out of the ICU faster, thus saving the hospital costs associated with ICU-based care.

The device also has the potential to reduce out-of-pocket payments from patients who spend time recovering in the ICU. Although there will be significant variation in these costs from person-to-person, a reduction of time spent in the ICU will directly correspond to a decrease in costs for ICU-based care. Similarly, insurance companies would see a reduction in costs for the patient's visit at the hospital.

6.2.2 Environmental Impact

The device and its components are intended for multiple uses. All probes and probe-holders can be sterilized in an autoclave after clinical use and reused on the next patient. This significantly reduces the waste that would be produced if the components were designed for one-time use. Furthermore, the Doppler unit is powered by a rechargeable battery, thereby reducing the waste that would be generated by a single-use battery component. It should be noted, however, that this rechargeable lithium battery should be recycled at the end of its life cycle in order to minimize the impact on the environment. Overall, the device is environmentally friendly. Over the entirety of its lifetime, the device will produce a minimal amount of waste, most of which can be recycled.

6.2.3 Societal Influence

Although the device is intended for use by a relatively small subset of the population (those who have undergone microvascular free tissue transfer), the influence extends beyond this group of individuals. The device is designed to improve the quality of care for patients, ultimately providing a patient with around the clock observation. It may allow for patients to be released from hospitals sooner to finish their recovery from the comfort and convenience of their own homes, especially in cases such as free tissue transfer where the likelihood of complications is so minimal yet severe that all patients are kept for an extended stay in the hospital for postoperative monitoring. This device could allow patients to return to their homes sooner while still ensuring that a medical professional has access to streaming information about the status of the transplant that would inform them of any complications. This would have a significant impact on patients, especially young children, who may undergo several surgeries for a procedure such as a hand reconstruction.

6.2.4 Political Ramifications

The potential political ramifications of this device stem from making patient health and status remotely available to clinicians. Many underdeveloped countries lack the trained medical professionals and resources to treat certain conditions. For many of these places, a procedure such as a free tissue transfer is beyond the realm of possibility for burn victims, people with congenital deformities impeding their daily lives, and people who have suffered severe trauma requiring reconstructive surgery. In some cases, surgeons will make trips to such locations to set up impromptu clinics and perform these surgeries. However, the duration of their stay is limited and the period of postoperative monitoring is critical. A device such as this would allow key information about the status of a patient to be relayed to trained clinicians who may be continents away and this would enable them to further and more appropriately advise the local attending clinicians as to what action to take. The scope of telemonitoring in this fashion transcends the realm of free tissue transfers and can be applied to a variety of medical conditions where it would be highly beneficial if patients in developing countries had a means to provide actionable, objective, and reliable information to trained physicians far away. This includes the possibility to make western medicine accessible to undeveloped communities.

6.2.5 Ethical Concerns

As technology rapidly develops in a world that is still very much dependent on human interaction, the question of using devices to replace human labor becomes one of ethics when considering the successive impact on jobs and the economy. One must consider that, for every new piece of technology that is developed with the capability of replacing human labor, there is a substantial impact, ranging from the personal to the institutional level. The device developed over the course of this project is intended for the purpose of continuous and remote flap monitoring with intentions of providing rapid, accurate data to the clinician so that decisions can be made quickly and objectively. The nature of such a device reduces, and in the most extreme case, eliminates the need for hourly clinical examinations by nursing staff in the ICU. Use of the device on a small scale (for example, one or two patients) would not lead to a significant impact in hospital workflow: the nurses who are assigned to a patient being monitored by the device would have availability to tend to other patients in the ICU. However, if the device were to be used on a large scale, with several patients at any given time being monitored using the system, it is possible that the ICU could become overstaffed. This would vary significantly from hospital to hospital, depending on the number of free tissue transfers that are performed regularly and how many patients are recovering from this procedure at a given time. In the most extreme of cases, replacement of manual clinical examinations with automated monitoring could lead to a reduction in the nursing staff at the ICU. As previously mentioned, this is an ongoing concern as the world, and more specifically the medical industry, enters a new age of significant technological advancements. Such consideration should be given to any medical device with similar implications on personnel productivity.

6.2.6 Health and Safety Issues

The ultimate goal of this device is to provide patients and clinicians with an improved system for postoperative monitoring in microvascular free tissue transfer. As such, it is absolutely necessary that the device is safe for all users. The device is designed to improve the quality of care for patients who are recovering from surgery. The objective is not only to meet current monitoring standards that are in place at hospitals, but to exceed these without compromising patient safety. Proper measures have been taken to ensure that the device meets the safety requirements that are in place for medical devices of this nature (see Section 3.4).

Amongst these concerns are electrical safety, biocompatibility, and integrity of the transplant. Electrical safety will be addressed by obtaining approval from the Clinical Engineering Department at UMASS Medical Center. This approval will involve documenting all aspects of the device to ensure that it does not pose a physical threat to the patient or user. Similar approval could be obtained from any hospital where the device was implemented into the postoperative monitoring routine. Biocompatibility of the device is ensured through research and experimental methods. The components that come directly in contact with the patient (namely, the probes) were selected to ensure long-term compatibility with the skin. For example, the Doppler probes were obtained from a medical device company that markets the products for clinical use. Additional aspects of the device, such as the ultrasound-gel delivery system, were researched extensively and specifically chosen for biocompatibility to ensure that no reaction was elicited from prolonged exposure to the skin. Finally, it is critical that the device does not compromise the integrity of the newly transplanted flap in any way. To minimize the possibility of the device interfering with the patient's recovery, several measures were taken. First, non-invasive measures of flap monitoring (Doppler ultrasound, temperature, color sensing) were chosen for the device. This eliminated the risk of invasive components interfering with the vascular anastomosis or any other aspects of the recovering flap. Furthermore, lightweight wires were chosen for the connections between the surface probes and device connectors to minimize the mechanical stress placed on the tissue.

6.2.7 Manufacturability

The device has excellent manufacturing potential. The materials used to develop the device are readily accessible and could be reproduced either manually or automatically with the appropriate preparation. The device case and probe holders could be 3D printed from the existing design files (see Appendix B and Appendix C), or the files could be used to make a mold for mass produced and low cost injection molding. The device components (such as circuitry, switches, buttons, lights, etc.) could be purchased in bulk and assembled in the case using the same techniques that were implemented for the original device. Furthermore, the Doppler, temperature, and color-sensor probes could be purchased in bulk and wired to interface with the device. This may require some modification so that the plugs will be compatible with the jacks on the device case, but it is achievable using the original device as a guideline. Ultimately, the physical aspects of the device could be reproduced, either individually or in bulk.

Chapter 7: Discussion

Ultimately, the described device meets the objectives that were originally established for this project and provides an innovative solution to continuous and remote flap monitoring. Validation of the final design confirms its dependability and overall capability to detect vascular compromise accurately, quickly, and objectively. All components of the device also meet the objective of user-friendliness. All sensors were designed to be compact and comfortable for the patient. Furthermore, the graphical user interface provides clinicians with an intuitive, easy-to-use platform for accessing patient data from any mobile device. The entire unit, along with the probes, is also portable so that it can be transported throughout the hospital with the patient as necessary. Finally, adaptability was achieved by using probes that are compatible with a variety of flap sizes and types, as well as utilizing components that can be upgraded in future iterations of the device, such as the Doppler ultrasound probes and color transducer.

Along with meeting the objectives of the project, the device was developed within the constraints that were initially established. The device was designed in compliance with industry standards for medical equipment and technology, and thus prioritizes patient safety. In order to gain certification for use in the hospital, the device will undergo formal testing. The device was also designed to be comfortable for the patient. All patient probes are compact (<2.5x2.5cm) and suturable with thin wiring so as to not tug on the flap or create excess stress at the wound site. Finally, the placement of the probes is unobtrusive and allows for regular clinical examinations to be carried out on the patient. In addition to being designed in compliance with industry standards, the device effectively measures three of the six parameters that are currently used clinically for evaluating flap perfusion.

The device meets the necessary specifications described previously. It is capable of measuring temperature within the range of 30°C-42°C with a precision of ±0.1°C. This effectively enables the device to detect both venous and arterial compromise by a change in a surface temperature. The device is also capable of recording high quality (44.1 kHz, 16-bits) Doppler ultrasound signals. This provides clinicians with the ability to remotely listen to the signal without compromising the sound quality. Furthermore, the device is capable of data acquisition over a minimum period of 72 hours, which allows for continuous monitoring over the course of the patient's recovery. Finally, the device utilizes small (<2.5x2.5) and suturable probes that can easily be fixed to the flap and left in place as the patient begins their recovery.

The results obtained from initial testing of the system components are promising and will be used to further improve and optimize the device. The device has been shown capable of continuous temperature monitoring over an extended period of time. Furthermore, these data are accessible from the user interface in real time. The website application has the capability of displaying data from numerous probes at a single time, allowing temperature readings from both individual and combination probes to be plotted on a single graph. This enables the surgeon to easily compare the temperature at the reference sites and on the flap surface to ensure they are consistent.

The device has also been shown capable of both continuous and immediate Doppler monitoring. The recordings are effectively uploaded to the server in real-time and organized by the patient from whom they originate. The surgeon can then immediately listen to the time-stamped recording directly from the user interface. This enables the surgeon to make decisions regarding the status of anastomosis without needing to be present at the patient's bedside or to rely on subjective descriptions of the Doppler signal from the clinician who performed the examination.

The data collected from the color transducer also demonstrates the capability of the system to detect changes in light absorption that are not noticeable to the naked eye. Furthermore the transducer is sensitive enough to detect pulsatile changes corresponding to the patient's heart rate. The pulsatile waveform is present with a healthy arterial signal and absent after arterial occlusion. The overall intensity of the signal also varies with the different levels of occlusion tested. With further optimization, color sensing could be directly integrated into the system as a third parameter for monitoring flap status.

Chapter 8: Conclusions & Recommendations

The described system is capable of continuously monitoring temperature, Doppler ultrasound and color in subjects for the duration of their recovery. It is an improvement upon the already established practice of observing these physiological parameters by monitoring them continuously, objectively, and with a higher degree of sensitivity. Through signal processing methods, the device is capable of alerting clinicians, via SMS or visual alerts on the web interface, to potential complications even during periods of time between the hourly checks typically performed by a nurse. By checking for the presence of a non-zero baseline of the Doppler ultrasound power signal, the device can determine venous status, and by checking for the presence of the secondary peaks corresponding to a biphasic signal the device can determine arterial perfusion status. The team has filed a provisional patent for these novel algorithms to determine venous and/or arterial insufficiency from the Doppler ultrasound signal alone. Furthermore, the device can detect surface temperature fluctuations to 0.1°C and distinguish between possible venous or arterial complication using the aforementioned criteria from literature regarding temperature.

An additional benefit is that the data recorded to the web based application makes it feasible to remotely inform clinicians regarding the status of a patient and provides a record of the Doppler signals and temperature of the transplant that can later be referred to. Previously, Doppler signals were evaluated audibly and subjectively and no recording existed to be referred to. In several cases when a physician had to make a determination as to whether the transplant required revision surgery, only second hand descriptions of the Doppler signal from other clinicians who had been performing the monitoring were available.

After a more thorough validation, this system has the potential to replace the current gold standard manual evaluation by providing the same information to clinicians continuously, objectively, and remotely. This could not only improve the quality of patient care and success rate of free tissue transfers but also allow patients to transfer out of the ICU sooner and therefore free up hospital resources.

Many of the improvements that could be made relate to the portability and convenience of using the device. One major area for improvements is the incorporation of the outdated, heavy, and bulky analog Doppler flow meter technology. Because the end result of the acquired signals will involve digitizing them regardless and allowing surgeons to hear them remotely, the clunky Doppler flow meter should be removed entirely and replaced with small and efficient integrated circuit solutions for Doppler ultrasound front-end modules that are available from a variety of suppliers such as Analog Devices and Texas Instruments. This would allow for a drastic miniaturization of the associated circuitry and could allow for a miniature device to be worn on the patient and interface with the sensors and transmit that over Bluetooth or WLAN to a base unit that performs the signal processing and communication with the remote server.

For further miniaturization, custom Doppler probes could be manufactured so that the piezoelectric crystals responsible for transmitting and receiving the ultrasonic signals were built directly into the probe case design. The possibility to 3D print objects with electrical conduits makes this a promising option and then the light sensor could also be integrated into the very same probe holder along with its associated circuitry and then a single cable could connect the three transducers to whatever unit was responsible for acquiring the data.

For signal processing, the collection of much more data over a large population of patients over the course of using this device would allow for the refinements of the signal processing algorithms and possibly even new discoveries being made about the relationships between Doppler, temperature, and color and their early prediction of the loss of flap perfusion. These improved algorithms should continue to be incorporated into the device. It is possible to visualize the Doppler signal and use it to get a visual representation of the velocity profile of blood travelling through the vessels. However, apparatuses already exist that serve this purpose. Furthermore, the cost and complexity that this would add to the system would remove it from the realm of accessibility that makes it practical for at home patient monitoring or deploying in underdeveloped areas, which the team recommends as being the focus for further improvements to the device.

Regarding the web interface and data storage, this aspect of the device would need to be scaled up accordingly with usage. Currently it is more than suitable for a dozen or so devices and users to be able to manage the devices as well as the patients and data associated with them. However, for this to be offered on a greater scale, the data storage would need to be scaled up accordingly as well as the administrative features of the website in order to ensure a seamless user experience.

References

1. Hanasono, M. M. (2013). Prevention and treatment of free flap thrombosis (and what to do after the flap fails). Kaiser Permanente Plastic Surgery Symposium. MD Anderson Cancer Center.
2. Koul, A. R., Patil, R. K., & Nahar, S. (2013). Unfavourable results in free tissue transfer. Indian Journal of Plastic Surgery : Official Publication of the Association of Plastic Surgeons of India, 46(2), 247–255.
3. Fox, P.M., Zeidler, K., Carey, J., & Lee, G. K. (2013). White light spectroscopy for free flap monitoring. Microsurgery, 33(3), 198-202. doi:10.1002/micr.22069
4. Porter, G.T., Thompson, C. (2004). Microvascular Free Tissue Transfer. Grand Rounds Presentation, UTMB, Dept. of Otolaryngology.
5. Whitaker, I. S., Karoo, R. O. S., Oliver, D. W., Ganchi, P. A., Gulati, V., & Malata, C. M. (2005). Current techniques in the post-operative monitoring of microvascular free-tissue transfers. European Journal of Plastic Surgery, 27(7), 315-321.
6. Novakovic, D., Patel, R. S., Goldstein, D. P., & Gullane, P. J. (2009). Salvage of failed free flaps used in head and neck reconstruction. Head & Neck Oncology, 1, 33. http://doi.org/10.1186/1758-3284-1-33.
7. Yu, S. C., Kleiber, G. M., & Song, D. H. (2013). An Algorithmic Approach to Total Breast Reconstruction with Free Tissue Transfer. Archives of Plastic Surgery, 40(3), 173–180. http://doi.org/10.5999/aps.2013.40.3.173.
8. Rose, E. H. (2015). Aesthetic reconstruction of the severely disfigured burned face: A creative strategy for a “natural” appearance using pre-patterned autogenous free flaps.Burns & Trauma, 3(1) doi:10.1186/s41038-015-0014-8.
9. Futran, N. D., Farwell, D. G., Smith, R. B., Johnson, P. E., & Funk, G. F. (2005). Definitive management of severe facial trauma utilizing free tissue transfer. Otolaryngology - Head and Neck Surgery, 132(1), 75-85. doi:10.1016/j.otohns.2004.08.013.
10. Buncke, H. H. (1984). Functional reconstruction of an extremity by free tissue transfer of the latissimus dorsi.Journal of Bone and Joint Surgery - Series A, 66(5), 806.
11. Salgado, C. J., Chim, H., Schoenoff, S., & Mardini, S. (2010). Postoperative Care and Monitoring of the Reconstructed Head and Neck Patient. Seminars in Plastic Surgery, 24(3), 281–287. http://doi.org/10.1055/s-0030-1263069.
12. Bui, D. T., Cordeiro, P. G., Hu, Q., Disa, J. J., Pusic, A., & Mehrara, B. J. (2007). Free flap reexploration: Indications, treatment, and outcomes in 1193 free flaps. Plastic and Reconstructive Surgery, 119(7), 2092-2100. doi:10.1097/01.prs.0000260598.24376.e1.
13. Glass, G. E., & Nanchahal, J. (2012). Why haematomas cause flap failure: An evidence-based paradigm. Journal of Plastic, Reconstructive and Aesthetic Surgery, 65(7), 903-910. doi:10.1016/j.bjps.2011.12.014.
14. Nakatsuka, T., Harii, K., Asato, H., Takushima, A., Ebihara, S., Kimata, Y.. . Ichioka, S. (2003). Analytic review of 2372 free flap transfers for head and neck reconstruction following cancer resection. Journal of Reconstructive Microsurgery, 19(6), 363-368.
15. Hidalgo, D. A., Disa, J. J., Cordeiro, P. G., & Hu, Q. (1998). A review of 716 consecutive free flaps for oncologic surgical defects: Refinement in donor-site selection and technique. Plastic and Reconstructive Surgery, 102(3), 722-732.

16. Hyodo, I., Nakayama, B., Kato, H., Hasegawa, Y., Ogawa, T., Terada, A., & Torii, S. (2007). Analysis of salvage operation in head and neck microsurgical reconstruction.The Laryngoscope, 117(2), 357-360. doi:10.1097/mlg.0b013e3180312380.
17. Chiu, E. S., Altman, A., & Allen, R. J. (2008). Free flap monitoring using skin temperature strip indicators: Adjunct to clinical examination. Plastic and Reconstructive Surgery,122(5), 144E-145E. doi:10.1097/PRS.0b013e318186caf3.
18. Tenorio, X., Mahajan, A. L., Wettstein, R., Harder, Y., Pawlovski, M., & Pittet, B. (2009). Early detection of flap failure using a new thermographic device. Journal of Surgical Research, 151(1), 15-21. doi:10.1016/j.jss.2008.03.001.
19. Garg, S., & Deschler, D. (2013). Saving a free flap with close clinical postoperative monitoring. Journal of the American Academy of Physician Assistants, 26(1), 47-49. doi:10.1097/01720610-201301000-00010.
20. Frost, M. W., Niumsawatt, V., Rozen, W. M., Eschen, G. E. T., Damsgaard, T. E., & Kiil, B. J. (2015). Direct comparison of postoperative monitoring of free flaps with microdialysis, implantable cook-swartz Doppler probe, and clinical monitoring in 20 consecutive patients. Microsurgery, 35(4), 262-271. doi:10.1002/micr.22331.
21. W.M. Swartz, R. Izquierdo, M.J. Miller. Implantable venous doppler microvascular monitoring: laboratory investigation and clinical results. Plast Reconstr Surg, 93 (1994), pp. 152–163.
22. Poder, T. G., & Fortier, P. -. (2013). Implantable Doppler in monitoring free flaps: A cost-effectiveness analysis based on a systematic review of the literature. European Annals of Otorhinolaryngology, Head and Neck Diseases, 130(2), 79-85. doi:10.1016/j.anorl.2012.07.003.
23. Smit, J. M., Klein, S., De Jong, E. H., Zeebregts, C. J., De Bock, G. H., & Werker, P. M. N. (2012). Value of the implantable doppler system in free flap monitoring. Journal of Plastic, Reconstructive and Aesthetic Surgery, 65(9), 1276-1277. doi:10.1016/j.bjps.2012.03.016.
24. Swiontkowski, M. F. (1991). Laser Doppler Flowmetry—Development and Clinical Application. The Iowa Orthopaedic Journal, 11, 119–126.
25. Hölzle, F., Loeffelbein, D. J., Nolte, D., & Wolff, K. (2006). Free flap monitoring using simultaneous non-invasive laser doppler flowmetry and tissue spectrophotometry. Journal of Cranio-Maxillofacial Surgery, 34(1), 25-33. doi:10.1016/j.jcms.2005.07.010.
26. Hallock, G. G., & Rice, D. C. (2003). A comparison of pulse oximetry and laser Doppler flowmetry in monitoring sequential vascular occlusion in a rabbit ear model. The Canadian Journal of Plastic Surgery = Journal Canadien De Chirurgie Plastique, 11(1), 11.
27. Xing, C., Arai, K., Lo, E. H., & Hommel, M. (2012). Pathophysiologic cascades in ischemic stroke. International Journal of Stroke, 7(5), 378-385. doi:10.1111/j.1747-4949.2012.00839.x.
28. Dickson, M. G., & Sharpe, D. T. (1985). Continuous subcutaneous tissue pH measurement as a monitor of blood flow in skin flaps: An experimental study. British Journal of Plastic Surgery, 38(1), 39-42. doi:10.1016/0007-1226(85)90084-0.
29. Muller, P., Peura, R. A., Mendelson, Y., Kun, S., & Dunn, R. M. (1998). In-vitro and in-vivo comparative analysis of four tissue pH monitoring systems. Biomedical Instrumentation and Technology, 32(6), 656-667.

30. Korostynska, O., Arshak, K., Gill, E., & Arshak, A. (2008). Review paper: Materials and techniques for in vivo pH monitoring. IEEE Sensors Journal, 8(1), 20-28. doi:10.1109/JSEN.2007.912522.
31. Soller, B. R., Soller, B., Micheeh, R. H., Micheels, R. H., Coen, J., Parikh, B.. . Hsi, C. (1996). Feasibility of non-invasive measurement of tissue pH using near-infrared reflectance spectroscopy. Journal of Clinical Monitoring, 12(5), 387-395. doi:10.1007/BF02077636.
32. Bloos F, Reinhart K. Venous oximetry. Int Care Med. 2005;31(7):911–913.
33. Lin, S. et al. (2011). Tissue Oximetry Monitoring in Microsurgical Reconstruction Decreases Flap Loss and Improves Rate of Flap Salvage. Journal of the American Society of Plastic Surgeons, 127(3), 1080-1085.
35. ViOptix (2015). T.Ox Tissue Oximeter. http://vioptix.com.
36. DiStephano, D., Kostopoulos, T., Mathie, R., Phelan, M., & Schattschneider, E. (2015). Physiological Remote Monitoring of Free Tissue Transfer. Worcester Polytechnic Institute.

Appendix A: Device Code

A.1 Arduino Code

```
#include <Arduino.h>

#include <avr/wdt.h>

#include <Wire.h>

#include "Adafruit_MCP23017.h"



//Interface Command Types

#define RUNNING 0

#define TEMPERATURE 1

#define ULTRASOUND 2

#define LIGHT 3

#define FLAPPROBE1 0

#define FLAPPROBE2 1

#define REFTEMP 2


//Initialize MCP object

Adafruit_MCP23017 mcp;


//MCP Pins

      //LED outputs

#define ULTRASOUND_CURRENTRECORDING_LED 8

#define FLAPPROBE1_STATUS_LED 9

#define FLAPPROBE2_STATUS_LED 10

#define REFTEMP_STATUS_LED 11

      //Switch inputs

#define DOPPLER_POWER 0

#define PROBE_SELECT 1
```

```
#define DOPPLER_SPEAKER 2
#define FLAPPROBE1_SW 3
#define FLAPPROBE2_SW 4
#define REFTEMP_SW 5

//Digital Pins

//Analog Input Locations for Thermistor Probes
#define THERMISTOR0 0
#define THERMISTOR1 1
#define THERMISTOR2 2
#define THERMISTOR3 3

float n = 0;

const long temperatureInterval = 5000;
unsigned long previousTemp = 0;

//last status of the switches from mcp digital read
int lastDopplerPower, lastDopplerSpeaker, lastProbeSelect, lastFlapProbe1SW,
lastFlapProbe2SW, lastRefTempSW;


//Whether or not to be recording temperature or ultrasound based on user
input
//0 means reference temperature is off
volatile int recordTemperature = 0;
//0 means both probes off
//1 means flap probe 1
//2 means flap probe 2
//3 means flap probe 1 and 2
volatile int recordUltrasound = 0;
```

```
//Some variables to keep track of debouncing times
long debouncing_time = 500; //Debouncing Time in Milliseconds
volatile unsigned long last_micros;

void sendTemperature(double probe, double value){
	Serial.print("temperature ");
	Serial.print(probe);
	Serial.print(" ");
	Serial.println(value);
}

void sendpH(double probe, double value){
	Serial.print("pH ");
	Serial.print(probe);
	Serial.print(" ");
	Serial.println(value);
}

void sendInterface(int type, int value){
	Serial.print("Interface ");
	Serial.print(type);
	Serial.print(" ");
	Serial.println(value);
}



void setup(){

	//Initialize Serial
```

```
Serial.begin(115200);
//Initialize MCP chip address 0x00
mcp.begin(0);
//Initialize MCP pins as outputs
mcp.pinMode(ULTRASOUND_CURRENTRECORDING_LED, OUTPUT);
mcp.pinMode(FLAPPROBE1_STATUS_LED, OUTPUT);
mcp.pinMode(FLAPPROBE2_STATUS_LED, OUTPUT);
mcp.pinMode(REFTEMP_STATUS_LED, OUTPUT);
//Initialize MCP pins at inputs
mcp.pinMode(DOPPLER_POWER, INPUT);
mcp.pullUp(DOPPLER_POWER, HIGH);
mcp.pinMode(DOPPLER_SPEAKER, INPUT);
mcp.pullUp(DOPPLER_SPEAKER, HIGH);
mcp.pinMode(PROBE_SELECT, INPUT);
mcp.pullUp(PROBE_SELECT, HIGH);
mcp.pinMode(FLAPPROBE1_SW, INPUT);
mcp.pullUp(FLAPPROBE1_SW, HIGH);
mcp.pinMode(FLAPPROBE2_SW, INPUT);
mcp.pullUp(FLAPPROBE2_SW, HIGH);
mcp.pinMode(REFTEMP_SW, INPUT);
mcp.pullUp(REFTEMP_SW, HIGH);
//Initialize pin 7 for speaker/record
pinMode(7, OUTPUT);
//Initialize pin 6 for probe select
pinMode(6, OUTPUT);
//Initialize pin 5 for doppler power
pinMode(5, OUTPUT);
//Initialize interrupt pin for record ultrasound now
pinMode(2, INPUT_PULLUP);
attachInterrupt(0, debounceRecordUltrasoundNow, RISING);
```

```
	//Turn on and off all the LEDs Initially
	for (int i = ULTRASOUND_CURRENTRECORDING_LED; i<=REFTEMP_STATUS_LED; i++){
		mcp.digitalWrite(i,HIGH);
	}
	delay(100);
	for (int i = ULTRASOUND_CURRENTRECORDING_LED; i<=REFTEMP_STATUS_LED; i++){
		mcp.digitalWrite(i,LOW);
	}

	//Initialize current state of user interface
	lastFlapProbe1SW = mcp.digitalRead(FLAPPROBE1_SW);
	lastFlapProbe2SW = mcp.digitalRead(FLAPPROBE2_SW);
	recordUltrasound = lastFlapProbe1SW+2*lastFlapProbe2SW;
	sendInterface(ULTRASOUND,recordUltrasound);

	lastRefTempSW = mcp.digitalRead(REFTEMP_SW);
	recordTemperature = lastRefTempSW;
	sendInterface(TEMPERATURE, recordTemperature);


	lastDopplerPower = mcp.digitalRead(DOPPLER_POWER);
	digitalWrite(5, lastDopplerPower);
	lastProbeSelect = mcp.digitalRead(PROBE_SELECT);
	digitalWrite(6, !lastProbeSelect);
	lastDopplerSpeaker = mcp.digitalRead(DOPPLER_SPEAKER);
	digitalWrite(7, lastDopplerSpeaker);


	//recordTemperature = digitalRead(4);
	//sendInterface(TEMPERATURE, recordTemperature);
```

```
	//recordUltrasound = digitalRead(3);
	//sendInterface(ULTRASOUND,recordUltrasound);

	//Initialize current state of LEDs based on user interface
	//digitalWrite(ULTRASOUND_STATUS_LED, recordUltrasound);
	//digitalWrite(TEMPERATURE_STATUS_LED, recordTemperature);
	wdt_enable (WDTO_8S);  // reset after 8 second, if no "pat the dog" 
received
}



void loop(){
	wdt_reset();
	//Check to see if status of switches has changed. If so, update the new 
status to RaspberryPi
	//and await instructions
	if ( 
(!(lastFlapProbe1SW==mcp.digitalRead(FLAPPROBE1_SW)))||(!(lastFlapProbe2SW==m
cp.digitalRead(FLAPPROBE2_SW))) ){
		lastFlapProbe1SW = mcp.digitalRead(FLAPPROBE1_SW);
		lastFlapProbe2SW = mcp.digitalRead(FLAPPROBE2_SW);
		recordUltrasound = lastFlapProbe1SW + 2*lastFlapProbe2SW;
		sendInterface(ULTRASOUND,recordUltrasound);
	}
	if (!(lastRefTempSW==mcp.digitalRead(REFTEMP_SW))){
		recordTemperature = mcp.digitalRead(REFTEMP_SW);
		sendInterface(TEMPERATURE, recordTemperature);
		lastRefTempSW = recordTemperature;
	}
	if (!(lastProbeSelect==mcp.digitalRead(PROBE_SELECT))){
		lastProbeSelect = mcp.digitalRead(PROBE_SELECT);
		digitalWrite(6, !lastProbeSelect);
```

```
}
if (!(lastDopplerSpeaker==mcp.digitalRead(DOPPLER_SPEAKER))){
	lastDopplerSpeaker = mcp.digitalRead(DOPPLER_SPEAKER);
	digitalWrite(7, lastDopplerSpeaker);
}
if (!(lastDopplerPower==mcp.digitalRead(DOPPLER_POWER))){
	lastDopplerPower = mcp.digitalRead(DOPPLER_POWER);
	digitalWrite(5, lastDopplerPower);
}

//Variable to store incoming serial byte
byte byteRead;
//Check if serial command is available from Pi
//Serial commands from Pi follow this protocol
//"a" 97 is turn off ultrasound current recording LED
//"b" 98 is turn on ultrasound current recording LED
//"c" 99 is send state of ultrasound switch
//"d" 100 is both flap probes off
//"e" 101 is flap probe 1 only on
//"f" 102 is temperature status led low
//"g" 103 is temperature status led high
//"h" 104 is Flap Probe 2 only on
//"i" 105 is both flap probes on
//"j" 106 turn doppler box on
//"k" 107 return doppler box to standby
//"l" 108 activate probe 1
//"m" 109 activate probe 2
//"n" 110 return probe to standby
if (Serial.available()){
	byteRead = Serial.read();
	if (byteRead == 97){
```

```
		mcp.digitalWrite(ULTRASOUND_CURRENTRECORDING_LED, LOW);
		digitalWrite(7, mcp.digitalRead(DOPPLER_SPEAKER));
		digitalWrite(5, lastDopplerPower);
	}
	else if (byteRead == 98){
		mcp.digitalWrite(ULTRASOUND_CURRENTRECORDING_LED, HIGH);
		digitalWrite(7, LOW);
		digitalWrite(5, HIGH);
	}
	else if (byteRead == 99){
		sendInterface(ULTRASOUND, recordUltrasound);
	}
	else if (byteRead == 100){
		mcp.digitalWrite(FLAPPROBE1_STATUS_LED, LOW);
		mcp.digitalWrite(FLAPPROBE2_STATUS_LED, LOW);
		recordUltrasound = 0;
	}
	else if (byteRead == 101){
		mcp.digitalWrite(FLAPPROBE1_STATUS_LED, HIGH);
		mcp.digitalWrite(FLAPPROBE2_STATUS_LED, LOW);
		recordUltrasound = 1;
	}
	else if (byteRead == 102){
		mcp.digitalWrite(REFTEMP_STATUS_LED, LOW);
		recordTemperature = 0;
	}
	else if (byteRead == 103){
		mcp.digitalWrite(REFTEMP_STATUS_LED, HIGH);
		recordTemperature = 1;
	}
	else if (byteRead == 104){
```

```
			mcp.digitalWrite(FLAPPROBE1_STATUS_LED, LOW);
			mcp.digitalWrite(FLAPPROBE2_STATUS_LED, HIGH);
			recordUltrasound = 2;
		}
		else if (byteRead == 105){
			mcp.digitalWrite(FLAPPROBE1_STATUS_LED, HIGH);
			mcp.digitalWrite(FLAPPROBE2_STATUS_LED, HIGH);
			recordUltrasound = 3;
		}
		else {
			Serial.println("Arduino received unrecognized byte.");
		}
	}

	unsigned long currentMillis = millis();
	double temp1, temp2, temp3, temp4;
	double raw1, raw2, raw3, raw4;
	//Need to update this to check if time threshold is met first
	//then within time check whether each probe is on and analog read
acordingly
	if ( ((recordTemperature==1)||(recordUltrasound>0)) && (currentMillis-
previousTemp>=temperatureInterval)){
		previousTemp = currentMillis;

		if (recordUltrasound == 1){
			raw1 = 0;
			for (int i = 0; i < 1000; i++){
				raw1 += analogRead(THERMISTOR0);

			}
			temp1 = ((raw1/1000)-748.4460285728)/-9.0428638867;
			sendTemperature(1, temp1);
```

```
        delay(100);
    }
    if (recordUltrasound == 2){
        raw2 = 0;
        for (int i = 0; i < 1000; i++){
            raw2 += analogRead(THERMISTOR1);
        }
        temp2 = ((raw2/1000)-787.1128263932)/-9.34851298;
        sendTemperature(2, temp2);
        delay(100);
    }
    if (recordUltrasound == 3){
        raw1 = 0;
        raw2 = 0;
        for (int i = 0; i < 1000; i++){
            raw1 += analogRead(THERMISTOR0);
            raw2 += analogRead(THERMISTOR1);
        }
        temp1 = ((raw1/1000)-748.4460285728)/-9.0428638867;
        temp2 = ((raw2/1000)-787.1128263932)/-9.34851298;
        sendTemperature(1, temp1);
        delay(100);
        sendTemperature(2, temp2);
        delay(100);
    }
    if (recordTemperature == 1){
        raw3 = 0;
        raw4 = 0;
        for (int i = 0; i < 1000; i++){
            raw3 += analogRead(THERMISTOR2);
            raw4 += analogRead(THERMISTOR3);
```

```
                }
                temp3 = ((raw3/1000)-775.3926594177)/-9.6307474430;
                temp4 = ((raw4/1000)-758.7047741935)/-9.5974193548;
                sendTemperature(3, temp3);
                delay(100);
                sendTemperature(4, temp4);
                delay(100);
            }
        n++;
        if (n>1000){
            n=0;
        }
        }


}

//Debouncer for recording now
void debounceRecordUltrasoundNow(){
        if((long)(micros() - last_micros) >= debouncing_time * 1000) {
        recordUltrasoundNow();
        last_micros = micros();
        }
}

//INTERRUPT ROUTINE to record ultrasound immediately. Currently attached to
pin2.
void recordUltrasoundNow(){
        sendInterface(ULTRASOUND, 4);
}
```

A.2 patientmonitor.h

```
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <string>
#include <iostream>
#include <pthread.h>
#include <thread>
#include <sstream>
#include <vector>
#include <curl/curl.h>
#include <wiringSerial.h>
#include <time.h>


using namespace std;

struct Measurement
{
	string type;
	double probe;
	double value;
};

struct Status
{
	int patientID;
	int temperature;
	int doppler;
	int dopplernow;
};
/*********GLOBAL DECLARATIONS*******************
**************************************************/
extern const int deviceID;
extern Status deviceStatus;

/***********FUNCTION PROTOTYPES******************
**************************************************/

//curlFunctions
int status_Write(Status deviceStatus);

size_t curl_get_status(void *ptr, size_t size, size_t nmemb, void *buffer);
Status status_Get(int deviceID);

int mysql_write_doppler (int patientID, string type, string sensorProbe,
string sensorValue);
```

```
int mysql_write (int patientID, string type, string sensorProbe, string
sensorValue);

size_t curl_write_measurement( void *ptr, size_t size, size_t nmemb, void
*buffer);
Measurement mysql_getMeasurement(string typesend, int id);

size_t curl_write_last( void *ptr, size_t size, size_t nmemb, void *buffer);
int lastRow (string table);

//signalProcessing
double averageLastN(string type, int n);

//dopplerFunctions
//both static
void* recordNow(void *arg);
void* dopplerRecord(void *arg);

//arduinoInterface
//both static
void* getRequests(void *arg);
void* readRequests(void *arg);

void identifyRequest(string request);
void sendByte(char byte);

//webInterface
//static
void* webInterface(void *arg);

//datetime
const std::string currentDateTime();
```

A.3 patientmonitor.cpp

```
#include "patientmonitor.h"

/**********GLOBAL VARIABLES FOR INTERFACE***********
****************************************************
***************************************************/
//Device ID Number
const int deviceID = 1;

Status deviceStatus = {};

/***********MAIN*******************
**********************************
*********************************/

int main (){
	//Global variable deviceStatus initialization
	///////////////////////////////////////////////

	// Curl Initialization
	curl_global_init(CURL_GLOBAL_ALL);

	//Patient ID Number Associated with Device
	deviceStatus = status_Get(deviceID);
	deviceStatus.dopplernow = 0;
	deviceStatus.temperature = 0;
	deviceStatus.doppler = 0;
	status_Write(deviceStatus);
	////////////////////////////////////////////////

	cout <<"Starting Program. Current date time is: " << currentDateTime()
<< std::endl;
	//Vector queue of requests coming from arduino
	vector<string> pendingRequests;

	//Threads being used in the program
	pthread_t thread1; //thread running "getRequests" listening for
requests from arduino
	pthread_t thread2; //thread running "readRequests" parsing requests
from getRequests
	pthread_t thread3; //thread running "dopplerRecording" recording 20
seconds of doppler every 5 minutes
	pthread_t thread4; //
	std::cout << currentDateTime() << " - " << "Starting Threads" <<
std::endl;
	int s;
	// COMMENT OUT THIS OTHER THREAD OF CURL STUFF
	s=pthread_create(&thread4, NULL, webInterface, NULL);
```

```
            if (s != 0){
                  std::cout << "PThread Create Failure" << std::endl;
            }
      s=pthread_create(&thread1, NULL, getRequests, &pendingRequests);
            if (s != 0){
                  std::cout << "PThread Create Failure" << std::endl;
            }
      s=pthread_create(&thread2, NULL, readRequests, &pendingRequests);
            if (s != 0){
                  std::cout << "PThread Create Failure" << std::endl;
            }
      s=pthread_create(&thread3, NULL, dopplerRecord, NULL);
            if (s != 0){
                  std::cout <<  "PThread Create Failure" << std::endl;
            }


      for(;;)
      {
      std::cout << currentDateTime() << " - " << pendingRequests.size() <<
std::endl;
      sleep(300);
      //std::cout << "Dopplernow status: " << deviceStatus.dopplernow <<
std::endl;
      }


      //Curl global cleanup
      curl_global_cleanup();
      return 0;
}
```

A.4 arduinoInterface.cpp

```
#include "patientmonitor.h"

/******Sending User Interface Feedback to Arduino****
*****************************************************
****************************************************/

void sendByte(char byte){
      int fd;
      if ((fd = serialOpen ("/dev/ttyACM0", 115200)) < 0){
      fprintf (stderr, "Unable to open serial device: %s\n", strerror
(errno)) ;
      //return 1 ;
      }
      serialPutchar(fd, byte);
}

/**************Thread to listen and get requests from
arduino*****************
**********Thread to read the requests in the queue and handle
them*************
*************************is further
down*************************************/

void* getRequests(void *arg){ //listens on serial port to get requests from
arduino then adds them to the queue of pendingRequests
      vector<string> *pendingRequests = (vector<string> *)arg;
      int fd;
      if ((fd = serialOpen ("/dev/ttyACM0", 115200)) < 0){
      fprintf (stderr, "Unable to open serial device: %s\n", strerror
(errno)) ;
      //return 1 ;
      }
      serialFlush(fd);
      for(;;){
                  string incomingRequest;
                  //for (int i=1;i=serialDataAvail(fd);i++)
                  char nextChar;
                  if (serialDataAvail(fd)>0){
                        do{
                              nextChar = (char)serialGetchar(fd);
                              incomingRequest += string(1,nextChar);
                        } while (nextChar!= '\n');
                  pendingRequests->insert(pendingRequests-
>begin(),incomingRequest); //adds the string received to the end of the
vector of strings in queue
                  //cout << incomingRequest;
                  incomingRequest.clear();
```

```
			}
			usleep(250);
		}
	}

	//Scans queue of pending requests and takes care of them
	void* readRequests(void *arg) {
		vector<string> *pendingRequests = (vector<string> *)arg;
		for(;;){
			if (!pendingRequests->empty()){
				identifyRequest(pendingRequests->back());
				pendingRequests->pop_back();
			}
			usleep(250);
		}
	}

	/***********Parsing requests in the queue************
	*****************************************************
	****************************************************/

	void identifyRequest(string request){
		/*
		cout << "enter thread" << endl;
		string* request = reinterpret_cast<string*>(arg);
		cout << "casted" << endl;
		cout << *request << endl;
		*/
		//cout << "Identifying request: " << request << endl;
		//string &request = *(static_cast<string*>(arg));
		//std::cout << request << endl;
		string type;
		double probe;
		double value;
		stringstream ss(request);
		ss >> type >> probe >> value;
		//cout << request << endl;
		if ((type.compare("temperature")==0) &&
	((deviceStatus.temperature)||(deviceStatus.doppler>0))){
			//cout<<"Temperature Reading\n";
			//cout<<"Probe: "<<probe<<'\n';
			//cout<<"Value: "<<value<<'\n';
			mysql_write(deviceStatus.patientID, type, to_string(probe),
	to_string(value));
			sleep(1);
			//cout << "called mysqlwrite for probe " << probe << endl;
			}
		else if (type.compare("Interface")==0){
			/*In this case probe refers to the user interface control element
			0 - running
```

```
            1 - temperature
            2 - ultrasound
            3 - pH
            Then value is to set the value of boolean 0 for false 1 for true
            */
            switch((int)probe){
                  case 0:
                        if (value==0){
                              deviceStatus.temperature = 0;
                              deviceStatus.doppler = 0;
                              sendByte('f');
                              sendByte('d');
                        }
                        else if (value==1){
                              deviceStatus.temperature = 1;
                              deviceStatus.doppler = 1;
                              sendByte('g');
                              sendByte('e');
                        }
                        else
                              cout << "Error Detected Invalid Input User
Interface" << endl;
                        break;
                  case 1:
                        if (value==0){
                              deviceStatus.temperature = 0;
                              sendByte('f');
                        }
                        else if (value==1){
                              deviceStatus.temperature = 1;
                              sendByte('g');
                        }
                        else
                              cout << "Error Detected Invalid Input User
Interface" << endl;
                        break;
                  case 2:
                        //both probes off
                        if (value==0){
                              deviceStatus.doppler = 0;
                              sendByte('d');
                        }
                        //flap probe 1 on
                        else if (value==1){
                              deviceStatus.doppler = 1;
                              sendByte('e');
                        }
                        //flap probe 2 on
                        else if (value==2){
                              deviceStatus.doppler = 2;
```

```
                              sendByte('h');
                        }
                        //both probes on
                        else if (value==3){
                              deviceStatus.doppler = 3;
                              sendByte('i');
                        }
                        else if (value==4){
                              if (deviceStatus.dopplernow){
                                    break;
                              }
                              else
                                    cout << "Ultrasound Record Now" << endl;
                                    pthread_t recordthread; //thread for
ultrasound recording
                                    int s;
                                    s=pthread_create(&recordthread, NULL,
recordNow, NULL);
                                    if (s != 0){
                                          std::cout << "Ultrasound on Demand
Thread Create Failure" << std::endl;
                                    }
                                    s=pthread_detach(recordthread);
                                    if (s != 0){
                                          std::cout << "Error detaching on
demand ultrasound thread" << std::endl;
                                    }
                        }
                        else
                              cout << "Error Detected Invalid Input User
Interface" << endl;
                        break;
            }
            status_Write(deviceStatus);
            //std::cout << "wrote device status after interface" <<
std::endl;
      }
      else {
            cout << "Did not understand request:";
            cout << request << endl;
      }

}
```

A.5 webInterface.cpp

```
#include "patientmonitor.h"

/***********************************************************
**************Thread to read interface data from website*******
**********************************************************/
void* webInterface(void *arg){
      for (;;){
            bool dopplerflag;
            if (deviceStatus.dopplernow){
                  dopplerflag = false;
                  //std::cout << "Doppler flag false" << std::endl;
            }
            else {
                  dopplerflag = true;
                  //std::cout << "doppler flag true" << std::endl;
            }
            deviceStatus = status_Get(deviceID);
            if ((deviceStatus.dopplernow==1)&&dopplerflag){
                  //std::cout << "Starting new doppler thread for recording"
<< std::endl;
                  pthread_t recordthread2; //thread for ultrasound recording
                  int s;
                  s=pthread_create(&recordthread2, NULL, recordNow, NULL);
                  if (s != 0){
                        std::cout << "Ultrasound on Demand Thread Create
Failure" << std::endl;
                  }
                  s=pthread_detach(recordthread2);
                  if (s != 0){
                        std::cout << "Error detaching on demand ultrasound
thread" << std::endl;
                  }
            }
            std::cout << deviceStatus.doppler << std::endl;
            if (deviceStatus.temperature == 0){
                  sendByte('f');
            }
            if (deviceStatus.temperature == 1){
                  sendByte('g');
            }
            if (deviceStatus.doppler == 0){
                  sendByte('d');
            }
            if (deviceStatus.doppler == 1){
                  sendByte('e');
            }
            if (deviceStatus.doppler == 2){
```

```
			sendByte('h');
		}
		if (deviceStatus.doppler == 3){
			sendByte('i');
		}
		status_Write(deviceStatus);
		sleep(10);
	}
}
```

A.6 curlFunctions.cpp

```
#include "patientmonitor.h"

/***********CURL STUFF IN HERE************
*******************************************
******************************************/


int status_Write(Status deviceStatus){
      string query = "deviceID="+ to_string(deviceID)+"&patientID="+
to_string(deviceStatus.patientID) +"&temperature="+
to_string(deviceStatus.temperature) +"&doppler="+
to_string(deviceStatus.doppler) +"&dopplernow="+
to_string(deviceStatus.dopplernow);
      CURL *curl3;
      CURLcode res;


      /* get a curl handle */
      curl3 = curl_easy_init();

      if(curl3) {
            /* First set the URL that is about to receive our POST. This URL
can
            just as well be a https:// URL if that is what should receive the
            data. */
            curl_easy_setopt(curl3, CURLOPT_URL,
"http://dennis.kbiophysics.com/updateStatus.php");
            /* Now specify the POST data */
            curl_easy_setopt(curl3, CURLOPT_POSTFIELDS, query.c_str());
            curl_easy_setopt(curl3, CURLOPT_NOSIGNAL, 1);

            /* Perform the request, res will get the return code */
            res = curl_easy_perform(curl3);
            /* Check for errors */
            if(res != CURLE_OK)
            fprintf(stderr, "1 curl_easy_perform() failed: %s\n",
                        curl_easy_strerror(res));

            /* always cleanup */
            curl_easy_cleanup(curl3);
      }

      return 0;
}

size_t curl_get_status( void *ptr, size_t size, size_t nmemb, void *buffer){
    ((std::string*)buffer)->append((char*)ptr, size * nmemb);
```

```
    return size * nmemb;
}

Status status_Get(int deviceID){
      Status receivedStatus;

      string query = "deviceID="+ to_string(deviceID);
      CURL *curl4;
      CURLcode res;

      /* get a curl handle */
      curl4 = curl_easy_init();

      /* declare string for the curl data recieved */
      string curlbuffer;

      /* First set the URL that is about to receive our POST. This URL can
      just as well be a https:// URL if that is what should receive the
      data. */
      curl_easy_setopt(curl4, CURLOPT_URL,
"http://dennis.kbiophysics.com/retrieveStatus.php");
      /* Now specify the POST data */
      curl_easy_setopt(curl4, CURLOPT_POSTFIELDS, query.c_str());
      // Tell curl to send the incoming bytes to this function to form a
measurement type
      curl_easy_setopt(curl4, CURLOPT_WRITEFUNCTION, curl_get_status);
      curl_easy_setopt(curl4, CURLOPT_WRITEDATA, &curlbuffer);
      curl_easy_setopt(curl4, CURLOPT_NOSIGNAL, 1);

      /* Perform the request, res will get the return code */
      res = curl_easy_perform(curl4);
      /* Check for errors */
      if(res != CURLE_OK)
    fprintf(stderr, "2 curl_easy_perform() failed: %s\n",
            curl_easy_strerror(res));

      /* always cleanup */
      curl_easy_cleanup(curl4);

      int patientID;
      int temperature;
      int doppler;
      int dopplernow;
      stringstream ss(curlbuffer);
      ss >> patientID >> temperature >> doppler >> dopplernow;

      receivedStatus.patientID = patientID;
      receivedStatus.temperature = temperature;
      receivedStatus.doppler = doppler;
      receivedStatus.dopplernow = dopplernow;
```

```
	return receivedStatus;
}



//Funcitons to add data to database
int mysql_write_doppler (int patientID, string type, string sensorProbe,
string sensorValue) {
	string query = "patientID=" + to_string(patientID) + "&type="+ type
+"&probe="+ sensorProbe +"&value="+ sensorValue;
	//string query = "patientID=" + to_string(patientID) + "&type="+ type
+"&probe="+ to_string(sensorProbe) +"&value="+ to_string(sensorValue);
	CURL *curl5;
	CURLcode res;

	/* get a curl handle */
	curl5 = curl_easy_init();

	if(curl5) {
		/* First set the URL that is about to receive our POST. This URL
can
		just as well be a https:// URL if that is what should receive the
		data. */
		curl_easy_setopt(curl5, CURLOPT_URL,
"http://dennis.kbiophysics.com/insertdoppler.php");
		/* Now specify the POST data */
		curl_easy_setopt(curl5, CURLOPT_POSTFIELDS, query.c_str());
		curl_easy_setopt(curl5, CURLOPT_NOSIGNAL, 1);

		/* Perform the request, res will get the return code */
		res = curl_easy_perform(curl5);
		/* Check for errors */
		if(res != CURLE_OK)
		fprintf(stderr, "3 curl_easy_perform() failed: %s\n",
				curl_easy_strerror(res));

		/* always cleanup */
		curl_easy_cleanup(curl5);
	}

	return 0;
}




int mysql_write (int patientID, string type, string sensorProbe, string
sensorValue) {
	//cout << "Inside mysql write for probe " << sensorProbe << endl;
```

```
	//cout << "Time in " << time(0) << endl;
	string query = "patientID=" + to_string(patientID) + "&type="+ type
+"&probe="+ sensorProbe +"&value="+ sensorValue;
	//string query = "patientID=" + to_string(patientID) + "&type="+ type
+"&probe="+ to_string(sensorProbe) +"&value="+ to_string(sensorValue);
	CURL *curl;
	CURLcode res;

	/* get a curl handle */
	curl = curl_easy_init();

	if(curl) {
		/* First set the URL that is about to receive our POST. This URL
can
		just as well be a https:// URL if that is what should receive the
		data. */
		curl_easy_setopt(curl, CURLOPT_URL,
"http://dennis.kbiophysics.com/insert.php");
		/* Now specify the POST data */
		curl_easy_setopt(curl, CURLOPT_POSTFIELDS, query.c_str());
		curl_easy_setopt(curl, CURLOPT_NOSIGNAL, 1);

		/* Perform the request, res will get the return code */
		res = curl_easy_perform(curl);
		/* Check for errors */
		if(res != CURLE_OK)
		fprintf(stderr, "4 curl_easy_perform() failed: %s\n",
				curl_easy_strerror(res));

		/* always cleanup */
		curl_easy_cleanup(curl);
	}
	//cout << "Done with probe " << sensorProbe << endl;

	return 0;
}


	//Functions to get data from database

size_t curl_write_measurement( void *ptr, size_t size, size_t nmemb, void
*buffer){
    ((std::string*)buffer)->append((char*)ptr, size * nmemb);
    return size * nmemb;
}
Measurement mysql_getMeasurement(string typesend, int id){
	Measurement gotMeasurement;

	string query = "type="+ typesend +"&id="+ to_string(id);
	CURL *curl1;
```

```
	CURLcode res;

	/* get a curl handle */
	curl1 = curl_easy_init();

	/* declare string for the curl data recieved */
	string curlbuffer;

	/* First set the URL that is about to receive our POST. This URL can
	just as well be a https:// URL if that is what should receive the
	data. */
	curl_easy_setopt(curl1, CURLOPT_URL,
"http://dennis.kbiophysics.com/retrieve.php");
	/* Now specify the POST data */
	curl_easy_setopt(curl1, CURLOPT_POSTFIELDS, query.c_str());
	// Tell curl to send the incoming bytes to this function to form a
measurement type
	curl_easy_setopt(curl1, CURLOPT_WRITEFUNCTION, curl_write_measurement);
	curl_easy_setopt(curl1, CURLOPT_WRITEDATA, &curlbuffer);
	curl_easy_setopt(curl1, CURLOPT_NOSIGNAL, 1);

	/* Perform the request, res will get the return code */
	res = curl_easy_perform(curl1);
	/* Check for errors */
	if(res != CURLE_OK)
    fprintf(stderr, "5 curl_easy_perform() failed: %s\n",
		curl_easy_strerror(res));

	/* always cleanup */
	curl_easy_cleanup(curl1);

	string type;
	double probe;
	double value;
	stringstream ss(curlbuffer);
	ss >> type >> probe >> value;

	gotMeasurement.type = type;
	gotMeasurement.probe = probe;
	gotMeasurement.value = value;

	return gotMeasurement;
}

size_t curl_write_last( void *ptr, size_t size, size_t nmemb, void *buffer){
    ((std::string*)buffer)->append((char*)ptr, size * nmemb);
    return size * nmemb;
}

int lastRow (string table){
```

```
	int last;

	string query = "type="+ table;
	CURL *curl2;
	CURLcode res;

	/* get a curl handle */
	curl2 = curl_easy_init();

	/* declare string for the curl data recieved */
	string curlbuffer;

	/* First set the URL that is about to receive our POST. This URL can
	just as well be a https:// URL if that is what should receive the
	data. */
	curl_easy_setopt(curl2, CURLOPT_URL,
"http://dennis.kbiophysics.com/last.php");
	/* Now specify the POST data */
	curl_easy_setopt(curl2, CURLOPT_POSTFIELDS, query.c_str());
	// Tell curl to send the incoming bytes to this function to form a
measurement type
	curl_easy_setopt(curl2, CURLOPT_WRITEFUNCTION, curl_write_last);
	curl_easy_setopt(curl2, CURLOPT_WRITEDATA, &curlbuffer);
	curl_easy_setopt(curl2, CURLOPT_NOSIGNAL, 1);

	/* Perform the request, res will get the return code */
	res = curl_easy_perform(curl2);
	/* Check for errors */
	if(res != CURLE_OK)
    fprintf(stderr, "6 curl_easy_perform() failed: %s\n",
		curl_easy_strerror(res));

	/* always cleanup */
	curl_easy_cleanup(curl2);

	last = stoi(curlbuffer);

    return last;
}
```

A.7 dopplerFunctions.cpp

```
#include "patientmonitor.h"

/*********Doppler Recording*********
*************************************
**********************************/

void* dopplerRecord(void *arg){
	for (;;){
		if (deviceStatus.doppler&&(!deviceStatus.dopplernow)){
			deviceStatus.dopplernow = 1;
			std::string dopname = std::string("doppler-
")+currentDateTime()+std::string(".wav");
			//std::cout << "Dopname" << dopname << std::endl;
			std::string const command =
std::string("/home/pi/development/fourth_iteration/dopplerrecord.sh ") +
dopname + std::string(" &");
			sendByte('b');
			system( command.c_str() );

	//system("/home/pi/development/third_iteration/dopplerrecord.sh &");
			sleep(5);
			deviceStatus.dopplernow = 0;
			sendByte('a');
			double probe = 1.0;
			mysql_write_doppler(deviceStatus.patientID,
std::string("doppler"), to_string(probe), dopname);
		}
		sleep(280);
	}
}

/*******Record 5 Seconds of Doppler NOW*********
***************************************************
**************************************************/
void* recordNow(void *arg){
	/*
	if (deviceStatus.dopplernow){
		std::cout << "Recording in Progress. Terminating Request." <<
std::endl;
		pthread_exit(NULL);
	}
	else {
	*/
		deviceStatus.dopplernow = 1;
		std::string dopname = std::string("doppler-
")+currentDateTime()+std::string(".wav");
		//std::cout << "Dopname " << dopname << std::endl;
```

```
            string command =
std::string("/home/pi/development/fourth_iteration/dopplerrecord.sh ") +
dopname + std::string(" &");
            sendByte('b');
            system( command.c_str() );
            //system("/home/pi/development/third_iteration/dopplerrecord.sh
&");
            sleep(5);
            deviceStatus.dopplernow = 0;
            //cout << "Dopplernow writing device status" << endl;
            status_Write(deviceStatus);
            sendByte('a');
            double probe = 1.0;
            //cout << "Dopplernow writing doppler filename to server" <<
endl;
            mysql_write_doppler(deviceStatus.patientID,
std::string("doppler"), to_string(probe), dopname);
      //}
      pthread_exit(NULL);
}

/*********TIMESTAMP STUFF************
***********************************/
const std::string currentDateTime() {
    time_t     now = time(0);
    struct tm  tstruct;
    char       buf[80];
    tstruct = *localtime(&now);
    // Visit http://en.cppreference.com/w/cpp/chrono/c/strftime
    // for more information about date/time format
    strftime(buf, sizeof(buf), "%Y-%m-%d.%X", &tstruct);

    return buf;
}
```

Appendix B: SolidWorks Diagram of Device Case

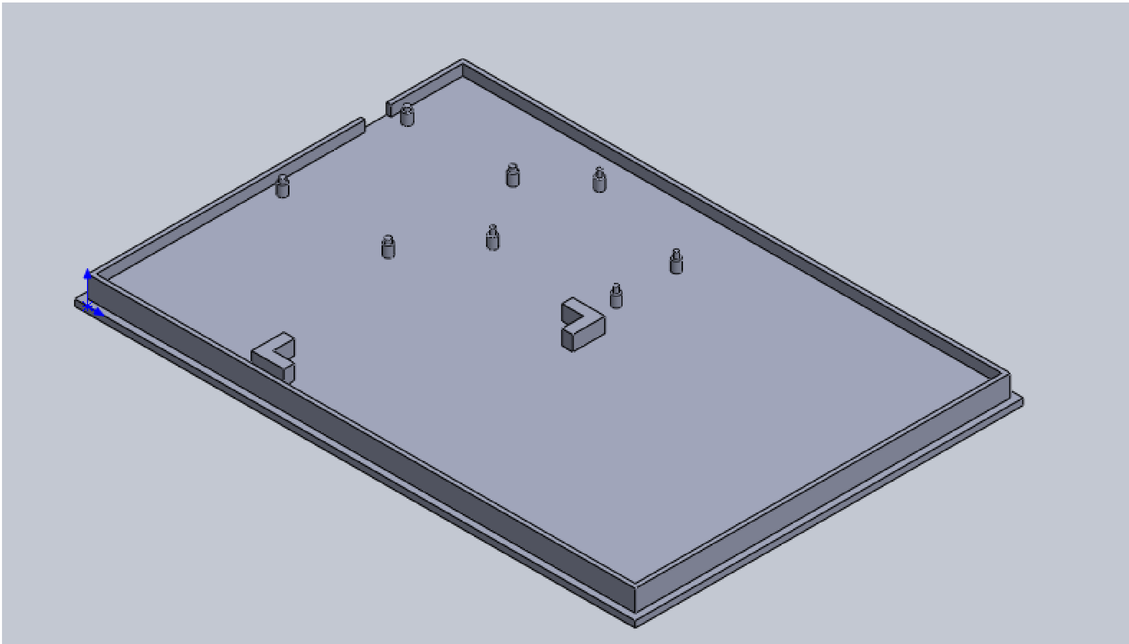

Fig. B1. SolidWorks diagram of bottom piece of device case.

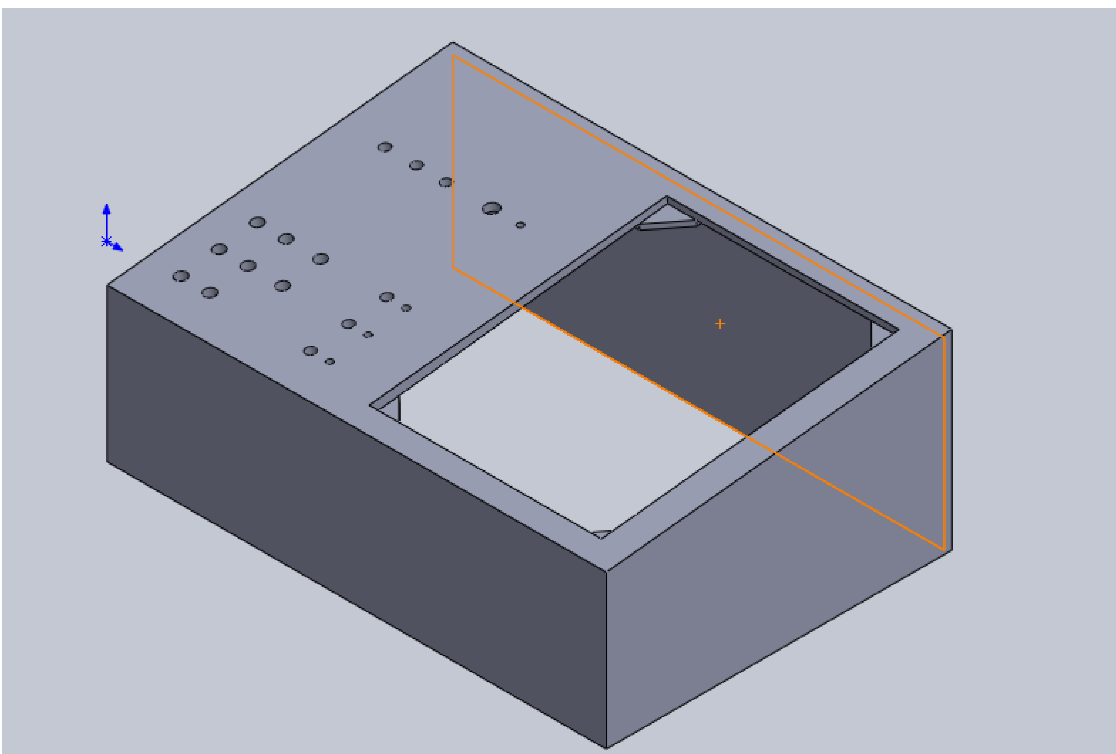

Fig. B2. Solidworks diagram of top piece of device case.

Appendix C: SolidWorks Diagram of Probe Holder

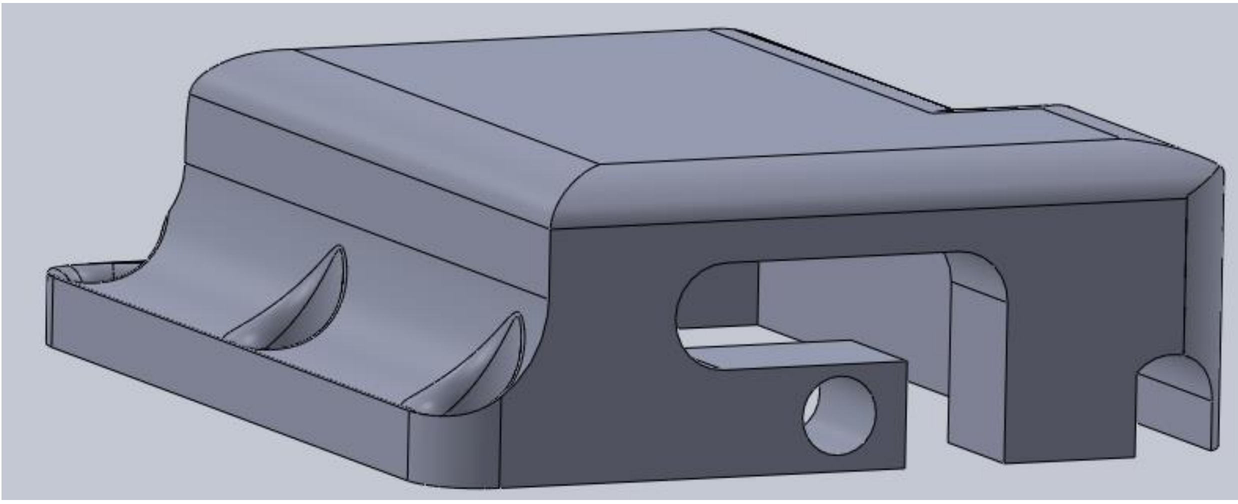

Fig. C1.Side view of probe holder diagram.

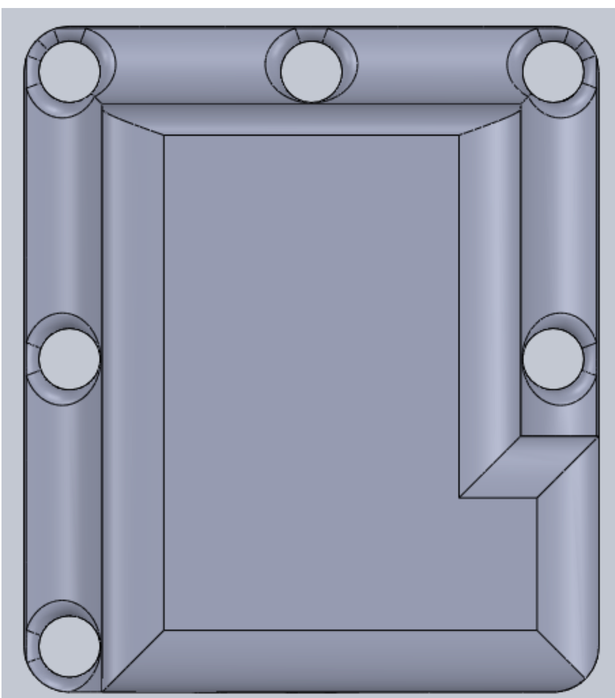

Fig. C2.Top-down view of probe holder diagram.

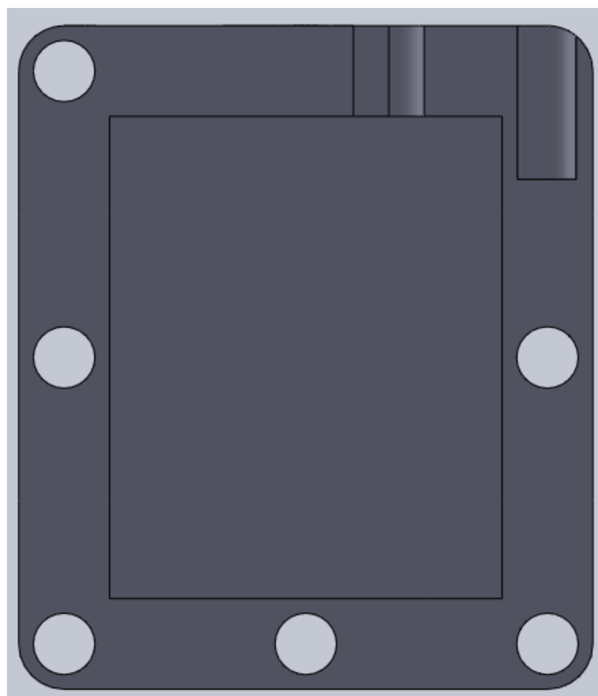

Fig. C3. Bottom-up view of probe holder diagram.

Appendix D: Matlab Signal Processing Code

```
% Power visualization of ultrasound signals
%% Initialize everything load in the wav files
clear all;
Fs = 44100;
cleans = [1,2];
blocks = [1,3];
for i = 1:1
    cleanfilename = sprintf('aparajit_1/doppler_8.wav');
    cleantemp = audioread(cleanfilename);
    clean{i} = cleantemp;
end

%% Filter Data before Computing Power
b = fdesign.bandpass(10/(Fs/2), 12/(Fs/2), 4000/(Fs/2), 4200/(Fs/2),
60,0.1,60);
filt1 = design(b,'butter');

for i = 1:1
    % filter the data
    clean{i} = filter(filt1, clean{i});
end

%% Compute power using window of signal
for j = 1:1
    % Initialize vector to contain clean signal power
    cleanpower{j} = [];
% Set window size
    windowsize = 800;
    for i = (windowsize/2)+1:windowsize/2:length(clean{j})-(windowsize/2)
        tempclean = clean{j}(i-windowsize/2:i+windowsize/2);
        cleanpower{j} = [cleanpower{j}
sum(tempclean.^2)*(1/2*length(tempclean)+1)];
    end
end


%% Generate plots of power
minrange = 0;
maxrange = 300000;

for i = 1:1
    %subplot(2,2,(i*2-1))

plot((1:length(cleanpower{i}))/(Fs/windowsize),cleanpower{i},(1:length(cleanp
ower{i}))/(Fs/windowsize), mean(cleanpower{i}))
    title('Subject Radial Power Signal Recovery')
    xlabel('Time (s)')
```

```
    ylabel('Power')
    axis([0 10 minrange maxrange])

end
```

Appendix E: Website Code

E.1 index.php

```
<?php
//include config
require_once('includes/config.php');

//check if already logged in move to dashboard
if( $user->is_logged_in() ){ header('Location: dashboard.php'); }

//process login form if submitted
if(isset($_POST['submit'])){

        $username = $_POST['username'];
        $password = $_POST['password'];

        if($user->login($username,$password)){
                $_SESSION['username'] = $username;
                header('Location: dashboard.php');
                exit;

        } else {
                $error[] = 'Wrong username or password or your account has
not been activated.';
        }

}//end if submit


?>


<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <!-- The above 3 meta tags *must* come first in the head; any other head
content must come *after* these tags -->
 <title>CareFreeÂ²</title>

 <!-- Bootstrap -->
 <link href="css/bootstrap.min.css" rel="stylesheet">

 <link href="stylesheetNew.css" rel="stylesheet">
```

```
 <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
queries -->
 <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
 <!--[if lt IE 9]>
 <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
 <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
 <![endif]-->
 </head>
 <body>
         <nav class="navbar navbar-default">
         <div class="container-fluid">

         <!-- Brand and toggle get grouped for better mobile display -->
         <div class="navbar-header">

          <a class="navbar-brand" href="index.php"><span
id="logoText1">CareFree&#178;</span><br><span id="logoText2">Flap Monitoring
System</span></a>
         </div>

         </div><!-- /.navbar-collapse -->
         </div><!-- /.container-fluid -->
         </nav>

         <div class="container">
         <div class="row">
         <div class="col-md-12">
        <div class="row">
                 <div class="col-md-3">
                <h2>Login</h2>
                <form role="form" method="post" action="" autocomplete="off">
                                                <?php
                                //check for any errors
                                if(isset($error)){
                                        foreach($error as $error){
                                                echo '<p class="bg-
danger">'.$error.'</p>';
                                        }
                                }
                                if(isset($_GET['action'])){
                                        //check the action
                                        switch ($_GET['action']) {
                                                case 'active':
                                                        echo "<h2 class='bg-
success'>Your account is now active you may now log in.</h2>";
                                                        break;
                                                case 'reset':
                                                        echo "<h2 class='bg-
success'>Please check your inbox for a reset link.</h2>";
```

```
                                                                break;
                                                        case 'resetAccount':
                                                                echo "<h2 class='bg-
success'>Password changed, you may now login.</h2>";
                                                                break;
                                                }
                                        }
                                        ?>
                                <div class="form-group">
                                <label for="exampleInputEmail1">Username</label>
                                <input type="text" name="username" id="username"
class="form-control" placeholder="Username" value="<?php if(isset($error)){
echo $_POST['username']; } ?>" tabindex="1">
                                </div>

                                <div class="form-group">
                                <label for="exampleInputPassword1">Password</label>
                                <input type="password" name="password" id="password"
class="form-control" placeholder="Password" tabindex="3">
                                </div>

                                <div class="form-group">
                                        <input id="buttonCustom" class="btn btn-default"
type="submit" name="submit" value="Login"/>
                                </div>

                                <div class="contentText">
                                         <a id="activeLink" href='reset.php'>Forgot your
Password?</a>
                                </div>

                                </form>

                                </div>
                </div>
                </div>
                </div>
        <hr>

        <footer>
        <p>Â© CareFreeÂ² 2016</p>
        </footer>
        </div>



 <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
 <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></scr
ipt>
```

```
 <!-- Include all compiled plugins (below), or include individual files as
needed -->
 <script src="js/bootstrap.min.js"></script>
 </body>
</html>
```

E.2 dashboard.php

```
<?php require('/var/www/html/dennis/includes/config.php');

//if not logged in redirect to login page
if(!$user->is_logged_in()){ header('Location: index.php'); }

?>

<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <!-- The above 3 meta tags *must* come first in the head; any other head
content must come *after* these tags -->
 <title>CareFreeÂ²</title>

 <!-- Bootstrap -->
 <link href="css/bootstrap.min.css" rel="stylesheet">

  <!--Additional Stylesheet -->
 <link href="stylesheetNew.css" rel="stylesheet">

 <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
queries -->
 <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
 <!--[if lt IE 9]>
 <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
 <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
 <![endif]-->
 </head>
 <body>
         <nav class="navbar navbar-default">
         <div class="container-fluid">

         <!-- Brand and toggle get grouped for better mobile display -->
         <div class="navbar-header">
         <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-
expanded="false">
         <span class="sr-only">Toggle navigation</span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
         </button>
```

```
<a class="navbar-brand" href="index.php"><span
id="logoText1">CareFreeÂ²</span><br><span id="logoText2">Flap Monitoring
System</span></a>

         </div>

         <!-- Collect the nav links, forms, and other content for toggling --
>
         <div class="collapse navbar-collapse" id="bs-example-navbar-
collapse-1">
         <ul class="nav navbar-nav" id="nav">
                 <li class="active"><a href="dashboard.php"><span
class="glyphicon glyphicon-home"></span> Home <span class="sr-
only">(current)</span></a></li>
                 <li><a href="patients.php"><span class="glyphicon glyphicon-
user"></span> Patients</a></li>
                 <li><a href="devices.php"><span class="glyphicon glyphicon-
phone"></span> Devices</a></li>
         </ul>

         <!-- code for the right side of the navbar -->
         <ul class="nav navbar-nav navbar-right" id="nav">
                <li class="dropdown">
                        <a href="#" class="dropdown-toggle" data-
toggle="dropdown" role="button" aria-haspopup="true" aria-
expanded="false"><span class="glyphicon glyphicon-cog"></span> Account<span
class="glyphicons glyphicons-user"></span><span class="caret"></span></a>
                                <ul class="dropdown-menu">
                                <li><a href="settings.php">Settings</a></li>
                                <li><a href="logout.php">Logout</a></li>
                                </ul>
                </li>
         </ul>
         </div><!-- /.navbar-collapse -->
         </div><!-- /.container-fluid -->
         </nav>

         <div class="container">
         <div class="row">
         <div class="col-md-12">
         <div class="jumbotron">
         <h1>Welcome</h1>
         <p>This is the main landing screen. Patients can be added, managed,
and monitored under the patients tab. Devices can be assigned to patients and
controlled under the devices tab.


         </p>
         </div>
         </div>
```

```
          </div>
         <hr>

         <footer>
         <p>Â© CareFreeÂ² 2016</p>
         </footer>
         </div>


 <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
 <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></scr
ipt>
 <!-- Include all compiled plugins (below), or include individual files as
needed -->
 <script src="js/bootstrap.min.js"></script>
 </body>
</html>
```

E.3 patients.php

```
<?php require('includes/config.php');

//if not logged in redirect to login page
if(!$user->is_logged_in()){ header('Location: index.php'); }

$patientAdded = $_GET['patientAdded'];

?>

<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <!-- The above 3 meta tags *must* come first in the head; any other head
content must come *after* these tags -->
 <title>Patients- CareFreeÂ²</title>

 <!-- Bootstrap -->
 <link href="css/bootstrap.min.css" rel="stylesheet">

 <!--Additional Stylesheet -->
 <link href="stylesheetNew.css" rel="stylesheet">

 <!-- Checks current page -->
<script src="http://code.jquery.com/jquery-1.8.3.min.js"></script>
<script>
$(function(){
  $('a').each(function() {
    if ($(this).prop('href') == window.location.href) {
      $(this).addClass('current');
    }
  });
});
</script>


 <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
queries -->
 <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
 <!--[if lt IE 9]>
 <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
 <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
 <![endif]-->
 </head>
```

```
 <body>
        <nav class="navbar navbar-default">
        <div class="container-fluid">

        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-
expanded="false">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        </button>

<a class="navbar-brand" href="index.php"><span
id="logoText1">CareFreeÂ²</span><br><span id="logoText2">Flap Monitoring
System</span></a>

        </div>

        <!-- Collect the nav links, forms, and other content for toggling --
>
        <div class="collapse navbar-collapse" id="bs-example-navbar-
collapse-1">
        <ul class="nav navbar-nav" id="nav">
                <li><a href="index.php"><span class="glyphicon glyphicon-
home"></span> Home</a></li>
                <li class="active"><a href="patients.php"><span
class="glyphicon glyphicon-user"></span> Patients<span class="sr-
only">(current)</span></a></li>
                <li><a href="devices.php"><span class="glyphicon glyphicon-
phone"></span> Devices</a></li>
        </ul>

        <!-- code for the right side of the navbar -->
        <ul class="nav navbar-nav navbar-right" id="nav">
               <li class="dropdown">
                       <a href="#" class="dropdown-toggle" data-
toggle="dropdown" role="button" aria-haspopup="true" aria-
expanded="false"><span class="glyphicon glyphicon-cog"></span> Account<span
class="glyphicons glyphicons-user"></span><span class="caret"></span></a>
                               <ul class="dropdown-menu">
                               <li><a href="settings.php">Settings</a></li>
                               <li><a href="logout.php">Logout</a></li>
                               </ul>
               </li>
        </ul>
        </div><!-- /.navbar-collapse -->
        </div><!-- /.container-fluid -->
```

```
        </nav>

        <div class="container">
        <div class="row">
                <div class="col-md-12">
                <div class="row">
                        <div class="col-md-2">
                        <!-- Left Column -->
                        <div class="list-group">
                                        <center><h2>Patients</h2></center>
                                        <ul id="pat">
                                        <?php
                                        //$stmt = $db->prepare('SELECT
patients.patientID,patients.lastname,patients.firstname,alerts.patientID FROM
patients,alerts FULL OUTER JOIN alerts ORDER BY patientID DESC');
                                        $stmt = $db->prepare('SELECT
patientID,lastname,firstname FROM patients ORDER BY patientID DESC');
                                        $stmt->execute(array());
                                        while ($row = $stmt-
>fetch(PDO::FETCH_ASSOC)) {
                                        //echo "<a class='list-group-item'
href='patient.php?patientID=". $row["patientID"]."'>".
$row["lastname"].",".$row["firstname"]." <br><span class='label label-info'>3
Alerts</span></a>";
                                        echo "<a class='list-group-item'
href='patient.php?patientID=". $row["patientID"]."'>".
$row["lastname"].",".$row["firstname"]."</a>";
                                        }
                                        ?>
                                        <a href="../addpatient.php"
class="list-group-item"><span class="glyphicon glyphicon-plus"></span> Add a
Patient</a>
                                        <a href="../removepatient.php"
class="list-group-item"><span class="glyphicon glyphicon-minus"></span>
Remove a Patient</a>
                                        </ul>
                                </div>
                        </div>

                        <!--Mobile Table -->
                        <div class="visible-xs col-md-10">
                                <center>
                                <button data-toggle="collapse" data-
target="#patSumTable" class="btn btn-default"> <span class="glyphicon
glyphicon-th-list"></span> Show Patient Summary Table</button>
                                </center>
                                <br>
                                <div id="patSumTable" class="collapse">
                                        <div class="panel panel-default">
```

```
                                                  <div class="panel-
heading">Patient Summary</div>
                                                  <div class="panel-body">
                                                 <table class="table
summaryTable">
                                                 <thead>
                                                          <tr>
                                                                <th>Patient
Name</th>

        <th>Temperature Monitoring Status</th>
                                                                <th>Doppler
Monitoring Status</th>
                                                                <th>Last
Doppler Recording</th>
                                                          </tr>
                                                  </thead>
                                                  <tbody>
                                                  <?php
                                                                $stmt = $db-
>prepare('SELECT a.patientID,a.lastname,a.firstname,b.temperature,b.doppler
FROM patients a LEFT JOIN deviceStatus b ON a.patientID = b.patientID ORDER
BY patientID DESC');
                                                                $stmt-
>execute(array());

                                                                $results =
$stmt->fetchAll(PDO::FETCH_ASSOC);

        foreach($results as $row){
                                                                        echo
"<tr>";
                                                                        echo
"<td>". $row["lastname"].",".$row["firstname"]."</td>";

        if($row[temperature]==NULL){

        echo "<td> Inactive </td>";
                                                                        }
                                                                        else
if ($row[temperature]){

        echo "<td> Running </td>";
                                                                        }
                                                                        else
{

        echo "<td> Stopped </td>";
                                                                        }
```

```
        if($row[doppler]==NULL){

        echo "<td> Inactive </td>";
                                                                        }
                                                                        else
if ($row[doppler]){

        echo "<td> Running </td>";
                                                                        }
                                                                        else
{

        echo "<td> Stopped </td>";
                                                                        }
                                                                        $stmt
= $db->prepare('SELECT epoch FROM doppler WHERE patientID = :patientID ORDER
BY epoch DESC LIMIT 1');

        $stmt->execute(array(':patientID' => $row['patientID']));
                                                                        $row2
= $stmt->fetch(PDO::FETCH_ASSOC);
                                                                        if
($row2['epoch']==NULL){

        echo "<td>No Recordings Found</td>";
                                                                        }
                                                                        else
if (round((time()-$row2['epoch'])/60)){

        echo "<td>".round((time()-$row2['epoch'])/60)." minutes ago</td>";
                                                                        }
                                                                        echo
"</tr>";
                                                                }
                                                    ?>
                                                  </tbody>
                                                  </table>
                                          </div>
                                          </div>
                                          <br>
                                  </div>
                         </div>

                         <!--Desktop Table -->
                         <div class="hidden-xs col-md-10">
                         <!-- Right Column -->
                         <div class="panel panel-default">
                                  <div class="panel-heading">Patient
Summary</div>
```

```
                                <div class="panel-body">
                               <table class="table">
                               <thead>
                                        <tr>
                                              <th>Patient Name</th>
                                              <th>Temperature Monitoring
Status</th>
                                              <th>Doppler Monitoring
Status</th>
                                              <th>Last Doppler
Recording</th>
                                        </tr>
                                </thead>
                                <tbody>
                                <?php
                                              $stmt = $db->prepare('SELECT
a.patientID,a.lastname,a.firstname,b.temperature,b.doppler FROM patients a
LEFT JOIN deviceStatus b ON a.patientID = b.patientID ORDER BY patientID
DESC');
                                              $stmt->execute(array());

                                              $results = $stmt-
>fetchAll(PDO::FETCH_ASSOC);
                                              foreach($results as $row){
                                                      echo "<tr>";
                                                      echo "<td>".
$row["lastname"].",".$row["firstname"]."</td>";

        if($row[temperature]==NULL){
                                                              echo "<td>
Inactive </td>";
                                                      }
                                                      else if
($row[temperature]){
                                                              echo "<td>
Running </td>";
                                                      }
                                                      else {
                                                              echo "<td>
Stopped </td>";
                                                      }

        if($row[doppler]==NULL){
                                                              echo "<td>
Inactive </td>";
                                                      }
                                                      else if
($row[doppler]){
                                                              echo "<td>
Running </td>";
```

```
                                                        }
                                                        else {
                                                                echo "<td>
Stopped </td>";
                                                        }
                                                        $stmt = $db-
>prepare('SELECT epoch FROM doppler WHERE patientID = :patientID ORDER BY
epoch DESC LIMIT 1');
                                                        $stmt-
>execute(array(':patientID' => $row['patientID']));
                                                        $row2 = $stmt-
>fetch(PDO::FETCH_ASSOC);
                                                        if
($row2['epoch']==NULL){
                                                                echo "<td>No
Recordings Found</td>";
                                                        }
                                                        else if
(round((time()-$row2['epoch'])/60)){
                                                                echo
"<td>".round((time()-$row2['epoch'])/60)." minutes ago</td>";
                                                        }
                                                        echo "</tr>";
                                                }

                                  ?>

                                </tbody>
                                </table>
                                </div>
                                </div>
                        </div>
                </div>
                </div>
         </div>
        <hr>

        <footer>
        <p>Â© CareFreeÂ² 2016</p>
        </footer>
        </div>



 <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
 <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></scr
ipt>
 <!-- Include all compiled plugins (below), or include individual files as
needed -->
```

```
 <script src="js/bootstrap.min.js"></script>
 </body>
</html>
```

E.4 patient.php

```
<?php require('/var/www/html/dennis/includes/config.php');

//if not logged in redirect to login page
if(!$user->is_logged_in()){ header('Location: ../index.php'); }

$patientID = $_GET['patientID'];

//$deviceID = 1;


$stmt = $db->prepare('SELECT deviceID FROM deviceStatus WHERE patientID =
:patientID');
$stmt->execute(array(':patientID' => $patientID));



$stmt = $db->prepare('SELECT epoch, value FROM dennis_stuff.temperature WHERE
patientID = :patientID AND probe = 1 ORDER BY epoch DESC LIMIT 100');
$stmt->execute(array(':patientID' => $patientID));


$currentTime = time();

$table = array(
    'cols' => array(
        array('label' => 'Time', 'type' => 'number'),
        array('label' => 'Probe1. ÂºC', 'type' => 'number'),
        array('label' => 'Probe2. ÂºC' , 'type' => 'number'),
        array('label' => 'Probe3. ÂºC' , 'type' => 'number'),
        array('label' => 'Probe4. ÂºC' , 'type' => 'number')
    ),
    'rows' => array()
);

while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
if (($row['epoch']-$currentTime)/60 > -120){
    $table['rows'][] = array(
        'c' => array(
            array('v' => ($row['epoch']-$currentTime)/60),
            array('v' => $row['value']),
            array('v' => NULL),
            array('v' => NULL),
            array('v' => NULL)
        )
    );
}
}
```

```
$stmt = $db->prepare('SELECT epoch, value FROM dennis_stuff.temperature WHERE
patientID = :patientID AND probe = 2 ORDER BY epoch DESC LIMIT 100');
$stmt->execute(array(':patientID' => $patientID));

while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
if (($row['epoch']-$currentTime)/60 > -120){
    $table['rows'][] = array(
        'c' => array(
            array('v' => ($row['epoch']-$currentTime)/60),
            array('v' => NULL),
            array('v' => $row['value']),
            array('v' => NULL),
            array('v' => NULL)
        )
    );
}
}

$stmt = $db->prepare('SELECT epoch, value FROM dennis_stuff.temperature WHERE
patientID = :patientID AND probe = 3 ORDER BY epoch DESC LIMIT 100');
$stmt->execute(array(':patientID' => $patientID));

while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
if (($row['epoch']-$currentTime)/60 > -120){
    $table['rows'][] = array(
        'c' => array(
            array('v' => ($row['epoch']-$currentTime)/60),
            array('v' => NULL),
            array('v' => NULL),
            array('v' => $row['value']),
            array('v' => NULL)
        )
    );
}
}

$stmt = $db->prepare('SELECT epoch, value FROM dennis_stuff.temperature WHERE
patientID = :patientID AND probe = 4 ORDER BY epoch DESC LIMIT 100');
$stmt->execute(array(':patientID' => $patientID));

while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
if (($row['epoch']-$currentTime)/60 > -120){
    $table['rows'][] = array(
        'c' => array(
            array('v' => ($row['epoch']-$currentTime)/60),
            array('v' => NULL),
            array('v' => NULL),
            array('v' => NULL),
            array('v' => $row['value'])
```

```
        )
    );
}
}


$jsonTable=json_encode($table);


?>
<!DOCTYPE html>
<html lang="en" ng-app="mdWavesurferApp">
 <head>
 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <!-- The above 3 meta tags *must* come first in the head; any other head
content must come *after* these tags -->
 <title>Patients- CareFreeÂ²</title>

 <!-- Bootstrap -->
 <link href="css/bootstrap.min.css" rel="stylesheet">

 <!--Additional Stylesheet -->
 <link href="stylesheetNew.css" rel="stylesheet">

<!-- Checks current page -->
<script src="http://code.jquery.com/jquery-1.8.3.min.js"></script>
<script>
$(function(){
  $('a').each(function() {
    if ($(this).prop('href') == window.location.href) {
      $(this).addClass('current');
    }
  });
});
</script>

 <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
queries -->
 <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
 <!--[if lt IE 9]>
 <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
 <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
 <![endif]-->


 <!--Load the AJAX API-->
```

```
    <script type="text/javascript"
src="https://www.google.com/jsapi"></script>
    <script type="text/javascript"
src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
    <script type="text/javascript">

      // Load the Visualization API and the piechart package.
      google.load('visualization', '1.0', {'packages':['corechart']});

      // Set a callback to run when the Google Visualization API is loaded.
      google.setOnLoadCallback(drawChart);

      // Callback that creates and populates a data table,
      // instantiates the pie chart, passes in the data and
      // draws it.
      function drawChart() {



        // Create the data table.
        var data = new google.visualization.DataTable(<?=$jsonTable?>);



        // Set chart options
        var options = {
                                   hAxis: {title: 'Minutes Ago'},
                                   vAxis: {title: 'Temperature C'},
                                   //legend: 'none',
                                   'legend':'top',
                                   colors: ['#2985a3', '#5E9527',
'#F69A33', '#DA3842'],
                       //'width': (0.5 * w),
                       //'height':(0.5 * h),
                       chartArea:{left:50,top:30,width:'90%',height:'75%'}
                       //'chartArea': {'width': '90%', 'height': '90%'},
                       };

        // Instantiate and draw our chart, passing in some options.
        var chart = new
google.visualization.LineChart(document.getElementById('chart_div'));
        chart.draw(data, options);
      }
    </script>


    <!-- Audio Player-->
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
    <link href='http://fonts.googleapis.com/css?family=Roboto:400,500'
rel='stylesheet' type='text/css'>
    <link href='https://cdnjs.cloudflare.com/ajax/libs/material-design-
iconic-font/2.2.0/css/material-design-iconic-font.min.css'
          rel='stylesheet' type='text/css'>
    <link
href="https://cdnjs.cloudflare.com/ajax/libs/highlight.js/8.9.1/styles/github
.min.css" rel="stylesheet">
    <link href="https://cdnjs.cloudflare.com/ajax/libs/angular-
material/1.0.0-rc3/angular-material.min.css"
          rel="stylesheet" type="text/css"/>

    <link rel="stylesheet" href="main.css"/>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.4.7/angular.min.js">
</script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.4.7/angular-
animate.min.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.4.7/angular-
aria.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/angular-
material/1.0.0-rc3/angular-material.min.js"></script>
    <script type="text/javascript" src="wavesurfer.min.js"></script>
    <script type="text/javascript" src="wavesurfer.directive.js"></script>
    <script type="text/javascript" src="main.js"></script>

            <!-- App -->
        <script src="app.js"></script>

 </head>
 <body>
         <nav class="navbar navbar-default">
         <div class="container-fluid">

         <!-- Brand and toggle get grouped for better mobile display -->
         <div class="navbar-header">
         <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-
expanded="false">
         <span class="sr-only">Toggle navigation</span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
         </button>

         <a class="navbar-brand" href="index.php"><span
id="logoText1">CareFreeÂ²</span><br><span id="logoText2">Flap Monitoring
System</span></a>
```

```
        </div>

        <!-- Collect the nav links, forms, and other content for toggling --
>
        <div class="collapse navbar-collapse" id="bs-example-navbar-
collapse-1">
        <ul class="nav navbar-nav" id="nav">
                <li><a href="index.php"><span class="glyphicon glyphicon-
home"></span> Home</a></li>
                <li class="active"><a href="patients.php"><span
class="glyphicon glyphicon-user"></span> Patients<span class="sr-
only">(current)</span></a></li>
                <li><a href="devices.php"><span class="glyphicon glyphicon-
phone"></span> Devices</a></li>
        </ul>

        <!-- code for the right side of the navbar -->
        <ul class="nav navbar-nav navbar-right" id="nav">
               <li class="dropdown">
                       <a href="#" class="dropdown-toggle" data-
toggle="dropdown" role="button" aria-haspopup="true" aria-
expanded="false"><span class="glyphicon glyphicon-cog"></span> Account<span
class="glyphicons glyphicons-user"></span><span class="caret"></span></a>
                               <ul class="dropdown-menu">
                               <li><a href="settings.php">Settings</a></li>
                               <li><a href="logout.php">Logout</a></li>
                               </ul>
               </li>
        </ul>
        </div><!-- /.navbar-collapse -->
        </div><!-- /.container-fluid -->
        </nav>


         <div class="container">
        <div class="row">
                <div class="col-md-12">
                <div class="row">
                        <!--Patient Name Menu Collapses on Mobile-->
                        <div class="visible-xs col-md-2">
                                <center><h3>
                                        <?php
                                        $stmt = $db->prepare('SELECT
patientID,lastname,firstname FROM patients WHERE patientID = :patientID');
                                        $stmt-
>execute(array(':patientID'=>$patientID));
                                        while ($row = $stmt-
>fetch(PDO::FETCH_ASSOC)) {
                                        echo $row["firstname"]."
".$row["lastname"];
```

```
                                        }
                                        ?>

                                </h3></center>
                                <center>
                                <button data-toggle="collapse" data-
target="#patMenu" class="btn btn-default"> <span class="glyphicon glyphicon-
menu-hamburger"></span> Switch Patient</button>
                                </center>
                                <br>
                               <div id="patMenu" class="collapse">
                                        <ul id="pat">
                                        <?php
                                        $stmt = $db->prepare('SELECT
patientID,lastname,firstname FROM patients ORDER BY patientID DESC');
                                        $stmt->execute(array());
                                        while ($row = $stmt-
>fetch(PDO::FETCH_ASSOC)) {
                                                $myPatientID =
$row["patientID"];
                                                echo "<a class='list-group-
item' href='patient.php?patientID=". $row["patientID"]."'>".
$row["lastname"].",".$row["firstname"];
                                                        //$stmt = $db-
>prepare('SELECT COUNT(patientID) FROM alerts WHERE patientID = :patientID');
                                                        //$stmt-
>execute(array('patientID'=>$myPatientID));
                                                        //while ($row =
$stmt->fetch(PDO::FETCH_ASSOC)) {
                                                        //      if
($row['COUNT(patientID)'] > 0) {
                                                        //              echo
"hello";
                                                        //      }
                                                        //}
                                        }
                                        ?></a>


                                        <a href="../addpatient.php"
class="list-group-item"><span class="glyphicon glyphicon-plus"></span> Add a
Patient</a>
                                        <a href="../removepatient.php"
class="list-group-item"><span class="glyphicon glyphicon-minus"></span>
Remove a Patient</a>
                                        </ul>
                                        <br>
                               </div>
                        </div>
```

```
                         <div class="hidden-xs col-md-2">
                         <!-- Left Column -->
                         <div class="list-group">
                                        <center><h2>Patients</h2></center>
                                        <ul id="pat">
                                        <?php
                                        $stmt = $db->prepare('SELECT
patientID,lastname,firstname FROM patients ORDER BY patientID DESC');
                                        $stmt->execute(array());
                                        while ($row = $stmt-
>fetch(PDO::FETCH_ASSOC)) {
                                        echo "<a class='list-group-item'
href='patient.php?patientID=". $row["patientID"]."'>".
$row["lastname"].",".$row["firstname"]."</a>";
                                        }
                                        ?>
                                        <a href="../addpatient.php"
class="list-group-item"><span class="glyphicon glyphicon-plus"></span> Add a
Patient</a>
                                        <a href="../removepatient.php"
class="list-group-item"><span class="glyphicon glyphicon-minus"></span>
Remove a Patient</a>
                                        </ul>
                                </div>
                         </div>

                         <div class="col-md-5">
                         <!-- Right Column -->
                                 <div class="panel panel-default">
                                  <div class="panel-heading">Patient
Alerts</div>
                                  <div class="panel-body">
                                                <?php
                                                $stmt = $db->prepare('SELECT
message,priority FROM alerts WHERE patientID=:patientID');
                                                $stmt-
>execute(array(':patientID' => $patientID));
                                                while ($row = $stmt-
>fetch(PDO::FETCH_ASSOC)) {
                                                        if
($row["priority"]==0){
                                                                echo "<div
class='alert alert-info' id='alertBlue'><a href='#' class='close' data-
dismiss='alert' aria-label='close'>&times;</a>";
                                                        }
                                                        else if
($row["priority"]==1){
```

```
                                                                echo "<div
class='alert alert-warning' id='alertYellow'><a href='#' class='close' data-
dismiss='alert' aria-label='close'>&times;</a>";
                                                        }
                                                        else if
($row["priority"]==2){
                                                                echo "<div
class='alert alert-danger' id='alertRed'><a href='#' class='close' data-
dismiss='alert' aria-label='close'>&times;</a>";
                                                        }
                                                        echo
$row["message"]."</div>";
                                                }

                                                ?>
                                  </div>
                                </div>


                                <div class="panel panel-default">
                                  <div class="panel-heading">Temperature
Data</div>
                                  <div class="panel-body">
                                        <div id="chart_div"
style="width:100%; height:50vh"></div>
                                  </div>
                                </div>
                         </div>

                         <div class="col-md-5">
                         <!-- Right Column -->



                                        <div ng-controller="MainController">


                                          <md-wavesurfer-audio player-wave-
color="gray" player-progress-color="black"

           player-backend="MediaElement">
                                                <!-- This is analogous to
HTML <source> element -->
                                                <!-- Sample recording -->
                                                <md-wavesurfer-source
src="demo.wav" title="doppler-2016-04-12.20:10:53.wav"></md-wavesurfer-
source>
                                                <md-wavesurfer-source
src="demo2.wav" title="doppler-2016-04-12.20:18:42.wav"></md-wavesurfer-
source>
```

```
                                                  <!--Doppler Recordings for
this patient -->
                                        <?php

                                                  $files = array();
                                                  $path="../dopplerrecordings";
                                                  $dir = opendir($path); //
open the cwd..also do an err check.
                                                  while(false != ($file =
readdir($dir))) {
                                                          if(($file != ".") and
($file != "..") and ($file != "index.php") and ($file != "filenames.txt")) {

        $files[] = $file; // put in array.
                                                          }
                                                  }

                                                  rsort($files, SORT_NATURAL);
// sort.

                                                  // print.
                                                  foreach($files as $file) {
                                                          //echo("<a
href='dopplerwindow.php?address=$path/$file'>$file</a><br/>");
                                                  }

                                                          $stmt = $db-
>prepare('SELECT value FROM doppler WHERE patientID=:patientID ORDER BY value
DESC');
                                                          $stmt-
>execute(array(':patientID' => $patientID));
                                                          while ($row = $stmt-
>fetch(PDO::FETCH_ASSOC)) {
                                                                  //echo "<a
id='activeLink'
href='dopplerwindow.php?patientID=$patientID&address=$path/".$row['value']."'
>".$row['value']."</a><br/>";
                                                                  echo "<md-
wavesurfer-source src='".$path."/".$row['value']."'
title='".$row['value']."'></md-wavesurfer-source>";
                                                          }
                                                  ?>

                                            </md-wavesurfer-audio>
                                        </div>
```

```
                                <div>
                                <center>

                                        <?php
                                                $stmt = $db->prepare('SELECT
deviceID FROM deviceStatus WHERE patientID = :patientID');
                                                $stmt-
>execute(array(':patientID'=>$patientID));
                                                while ($row = $stmt-
>fetch(PDO::FETCH_ASSOC)) {
                                                echo "<a
href='device.php?deviceID=". $row["deviceID"]."' class='btn btn-default'
role='button'>Go to Device Control Panel</a>";
                                                }
                                        ?>
                                <br>
                                <?php
                                                $stmt = $db->prepare('SELECT
COUNT(patientID) FROM alerts WHERE patientID = :patientID');
                                                $stmt-
>execute(array('patientID'=>$patientID));
                                                while ($row = $stmt-
>fetch(PDO::FETCH_ASSOC)) {
                                                        if
($row['COUNT(patientID)'] > 0) {
                                                                echo "**For
testing**: <h5><span class='label label-danger' style='font-weight:
normal;'>". $row['COUNT(patientID)'] ."</span></h5>";
                                                        }
                                        }
                                ?>


                                </center>
                                </div>

                        </div>
                </div>
         </div>
         </div>
        <hr>

        <footer>
        <p>Â© CareFreeÂ² 2016</p>
        </footer>
        </div>
```

```
 <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
 <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></scr
ipt>
 <!-- Include all compiled plugins (below), or include individual files as
needed -->
 <script src="js/bootstrap.min.js"></script>
 </body>
</html>
```

E.5 addPatient.php

```
<?php require('includes/config.php');

//if not logged in redirect to login page
if(!$user->is_logged_in()){ header('Location: index.php'); }

$patientAdded = $_GET['patientAdded'];

?>

<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <!-- The above 3 meta tags *must* come first in the head; any other head
content must come *after* these tags -->
 <title>Patients- CareFreeÂ²</title>

 <!-- Bootstrap -->
 <link href="css/bootstrap.min.css" rel="stylesheet">

 <!--Additional Stylesheet -->
 <link href="stylesheetNew.css" rel="stylesheet">

 <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
queries -->
 <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
 <!--[if lt IE 9]>
 <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
 <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
 <![endif]-->
 </head>
 <body>
         <?php
        if($patientAdded==1){
        ?>
        <div class = "alert alert-success alert-dismissable" id="alertGreen">
           <button type = "button" class = "close" data-dismiss = "alert"
aria-hidden = "true">
                  &times;
           </button>
           Patient was successfully added. <a href="patients.php"><font
color="#5E9527"><b>Go back to Patients Page?</b></font></a>
        </div>
```

```
        <?php
        }
        ?>

                 <nav class="navbar navbar-default">
         <div class="container-fluid">

         <!-- Brand and toggle get grouped for better mobile display -->
         <div class="navbar-header">
         <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-
expanded="false">
         <span class="sr-only">Toggle navigation</span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
         </button>

<a class="navbar-brand" href="index.php"><span
id="logoText1">CareFreeÂ²</span><br><span id="logoText2">Flap Monitoring
System</span></a>

         </div>

         <!-- Collect the nav links, forms, and other content for toggling --
>
         <div class="collapse navbar-collapse" id="bs-example-navbar-
collapse-1">
         <ul class="nav navbar-nav" id="nav">
                 <li><a href="index.php"><span class="glyphicon glyphicon-
home"></span> Home</a></li>
                 <li class="active"><a href="patients.php"><span
class="glyphicon glyphicon-user"></span> Patients<span class="sr-
only">(current)</span></a></li>
                 <li><a href="devices.php"><span class="glyphicon glyphicon-
phone"></span> Devices</a></li>
         </ul>

         <!-- code for the right side of the navbar -->
         <ul class="nav navbar-nav navbar-right" id="nav">
                <li class="dropdown">
                        <a href="#" class="dropdown-toggle" data-
toggle="dropdown" role="button" aria-haspopup="true" aria-
expanded="false"><span class="glyphicon glyphicon-cog"></span> Account<span
class="glyphicons glyphicons-user"></span><span class="caret"></span></a>
                                <ul class="dropdown-menu">
                                <li><a href="settings.php">Settings</a></li>
                                <li><a href="logout.php">Logout</a></li>
                                </ul>
                </li>
```

```
        </ul>
        </div><!-- /.navbar-collapse -->
        </div><!-- /.container-fluid -->
        </nav>

        <div class="container">
        <div class="row">
               <div class="col-md-6 col-md-offset-3">
                       <div class="panel panel-default">
                                 <div class="panel-heading">Add a
Patient</div>
                                 <div class="panel-body">
                                        <p>Please complete the following
patient information:</p>
                                        <form method="POST"
name="addPatient" action="addpatient.php?patientAdded=1" autocomplete="off"/>
                                        <div class="form-group">
                                        <label>First Name</label>
                                        <input type="text" class="form-
control" name="firstName" class="input" placeholder="First Name" tabindex="1"
required>
                                        </div>

                                       <div class="form-group">
                                       <label>Last Name</label>
                                       <input type="text" name="lastName"
class="form-control" placeholder="Last Name" tabindex="1" required>
                                       </div>

                                       <div class="form-group">
                                       <input class="btn btn-default"
id="buttonCustom" type="submit" name="Submit" value="Add Patient">
                                       </div>

                                       <?php
                                       $firstName = $_POST['firstName'];
                                       $lastName = $_POST['lastName'];
                                       $stmt = $db->prepare('INSERT INTO
patients (firstname,lastname) VALUES (:firstname,:lastname)');
                                       $stmt-
>execute(array(':firstname'=>$firstName,':lastname'=>$lastName));
                                       ?>

                                       </form>
                                 </div>
               </div>

       </div>
       </div>
       <hr>
```

```
        <footer>
        <p>Â© CareFreeÂ² 2016</p>
        </footer>
        </div>



 <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
 <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></scr
ipt>
 <!-- Include all compiled plugins (below), or include individual files as
needed -->
 <script src="js/bootstrap.min.js"></script>
 </body>
</html>
```

E.6 removePatient.php

```
<?php require('includes/config.php');

//if not logged in redirect to login page
if(!$user->is_logged_in()){ header('Location: index.php'); }

$patientRemoved = $_GET['patientRemoved'];

?>

<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <!-- The above 3 meta tags *must* come first in the head; any other head
content must come *after* these tags -->
 <title>Patients- CareFreeÂ²</title>

 <!-- Bootstrap -->
 <link href="css/bootstrap.min.css" rel="stylesheet">

 <!--Additional Stylesheet -->
 <link href="stylesheetNew.css" rel="stylesheet">

 <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
queries -->
 <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
 <!--[if lt IE 9]>
 <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
 <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
 <![endif]-->
 </head>
 <body>
         <?php
        if($patientRemoved==1){
        ?>
        <div class = "alert alert-success alert-dismissable" id="alertGreen">
           <button type = "button" class = "close" data-dismiss = "alert"
aria-hidden = "true">
                  &times;
           </button>
           Patient was successfully removed. <a href="patients.php"><font
color="#5E9527"><b>Go back to Patients Page?</b></font></a>
        </div>
```

```
        <?php
        }
        ?>

                 <nav class="navbar navbar-default">
         <div class="container-fluid">

         <!-- Brand and toggle get grouped for better mobile display -->
         <div class="navbar-header">
         <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-
expanded="false">
         <span class="sr-only">Toggle navigation</span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
         </button>

<a class="navbar-brand" href="index.php"><span
id="logoText1">CareFreeÂ²</span><br><span id="logoText2">Flap Monitoring
System</span></a>

         </div>

         <!-- Collect the nav links, forms, and other content for toggling --
>
         <div class="collapse navbar-collapse" id="bs-example-navbar-
collapse-1">
         <ul class="nav navbar-nav" id="nav">
                 <li><a href="index.php"><span class="glyphicon glyphicon-
home"></span> Home</a></li>
                 <li class="active"><a href="patients.php"><span
class="glyphicon glyphicon-user"></span> Patients<span class="sr-
only">(current)</span></a></li>
                 <li><a href="devices.php"><span class="glyphicon glyphicon-
phone"></span> Devices</a></li>
         </ul>

         <!-- code for the right side of the navbar -->
         <ul class="nav navbar-nav navbar-right" id="nav">
                <li class="dropdown">
                        <a href="#" class="dropdown-toggle" data-
toggle="dropdown" role="button" aria-haspopup="true" aria-
expanded="false"><span class="glyphicon glyphicon-cog"></span> Account<span
class="glyphicons glyphicons-user"></span><span class="caret"></span></a>
                               <ul class="dropdown-menu">
                               <li><a href="settings.php">Settings</a></li>
                               <li><a href="logout.php">Logout</a></li>
                               </ul>
                </li>
```

```
        </ul>
        </div><!-- /.navbar-collapse -->
        </div><!-- /.container-fluid -->
        </nav>

        <div class="container">
        <div class="row">
               <div class="col-md-6 col-md-offset-3">
                       <div class="panel panel-default">
                                 <div class="panel-heading">Remove a
patient</div>
                                 <div class="panel-body">
                                        <p>Please choose a patient to
remove:</p>
                                        <form method="POST"
name="removePatient" action="removepatient.php?patientRemoved=1"
autocomplete="off"/>
                                        <div class="form-group">
                                         <select class="form-control"
name="var" required>
                                               <option value="" disabled
selected hidden>Select Patient</option>
                                                       <?php
                                                       $submittedValue;

                                                       $stmt = $db-
>prepare('SELECT patientID,lastname,firstname FROM patients ORDER BY
patientID DESC');
                                                       $stmt-
>execute(array());
                                                       while ($row = $stmt-
>fetch(PDO::FETCH_ASSOC)) {
                                                       echo "<option
class='patientDropdownSub' value=". $row["patientID"]."> ".
$row["lastname"].",".$row["firstname"]."</option>";
                                                       }
                                                       ?>
                                         </select>
                                        </div>

                                        <!-- Trigger the modal with a button
-->

                                       <button type="button" class="btn btn-
default" id="buttonCustom" data-toggle="modal" data-target="#myModal">Remove
Patient</button>
                                       <!-- Modal -->
                                       <div id="myModal" class="modal fade"
role="dialog">
                                         <div class="modal-dialog">
```

```
                                                <!-- Modal content-->
                                                <div class="modal-content">
                                                  <div class="modal-header">
                                                        <button type="button"
class="close" data-dismiss="modal">&times;</button>
                                                        <h4 class="modal-
title"><span class="glyphicon glyphicon-alert"></span> Remove Patient</h4>
                                                  </div>
                                                  <div class="modal-body">
                                                        <p>Are you sure that
you want to remove this patient? This action cannot be undone.</p>
                                                  </div>
                                                  <div class="modal-footer">
                                                                <input
class="btn btn-default" id="buttonCustom" type="submit" name="Submit"
value="Yes, Remove Patient"/>
                                                        <button type="button"
class="btn btn-default" data-dismiss="modal">No</button>
                                                  </div>
                                                </div>

                                          </div>
                                        </div>


                                        <?php
                                        if (isset($_POST['var']))
                                        {
                                        $patientID =
$_POST['var'];
                                        $stmt = $db->prepare('DELETE FROM
patients WHERE patientID = :patientID' );
                                        $stmt-
>execute(array(':patientID'=>$patientID));
                                        }
                                        ?>

                                        </form>
                                  </div>
                  </div>

        </div>
        </div>
        <hr>

        <footer>
        <p>Â© CareFreeÂ² 2016</p>
        </footer>
        </div>
```

```
 <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
 <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></scr
ipt>
 <!-- Include all compiled plugins (below), or include individual files as
needed -->
 <script src="js/bootstrap.min.js"></script>
 </body>
</html>
```

E.7 devices.php

```
<?php require('includes/config.php');

//if not logged in redirect to login page
if(!$user->is_logged_in()){ header('Location: index.php'); }

?>

<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <!-- The above 3 meta tags *must* come first in the head; any other head
content must come *after* these tags -->
 <title>Devices- CareFreeÂ²</title>

 <!-- Bootstrap -->
 <link href="css/bootstrap.min.css" rel="stylesheet">

 <!--Additional Stylesheet -->
 <link href="stylesheetNew.css" rel="stylesheet">

 <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
queries -->
 <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
 <!--[if lt IE 9]>
 <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
 <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
 <![endif]-->
 </head>
 <body>
         <nav class="navbar navbar-default">
         <div class="container-fluid">

         <!-- Brand and toggle get grouped for better mobile display -->
         <div class="navbar-header">
         <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-
expanded="false">
         <span class="sr-only">Toggle navigation</span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
         </button>
```

```
<a class="navbar-brand" href="index.php"><span
id="logoText1">CareFreeÂ²</span><br><span id="logoText2">Flap Monitoring
System</span></a>

         </div>

         <!-- Collect the nav links, forms, and other content for toggling --
>
         <div class="collapse navbar-collapse" id="bs-example-navbar-
collapse-1">
         <ul class="nav navbar-nav" id="nav">
                 <li><a href="index.php"><span class="glyphicon glyphicon-
home"></span> Home</a></li>
                 <li><a href="patients.php"><span class="glyphicon glyphicon-
user"></span> Patients</a></li>
                 <li class="active"><a href="devices.php"><span
class="glyphicon glyphicon-phone"></span> Devices<span class="sr-
only">(current)</span></a></li>
         </ul>

         <!-- code for the right side of the navbar -->
         <ul class="nav navbar-nav navbar-right" id="nav">
                <li class="dropdown">
                        <a href="#" class="dropdown-toggle" data-
toggle="dropdown" role="button" aria-haspopup="true" aria-
expanded="false"><span class="glyphicon glyphicon-cog"></span> Account<span
class="glyphicons glyphicons-user"></span><span class="caret"></span></a>
                                <ul class="dropdown-menu">
                                <li><a href="settings.php">Settings</a></li>
                                <li><a href="logout.php">Logout</a></li>
                                </ul>
                </li>
         </ul>
         </div><!-- /.navbar-collapse -->
         </div><!-- /.container-fluid -->
         </nav>

         <div class="container">
         <div class="row">
                 <div class="col-md-12">
                 <div class="row">
                         <div class="col-md-2">
                         <!-- Left Column -->
                         <div class="list-group">
                                        <center><h2>Devices</h2></center>
                                        <ul id="pat">
                                        <?php
                                                $stmt = $db->prepare('SELECT
deviceID FROM deviceStatus');
                                                $stmt->execute(array());
```

```
                                                while ($row = $stmt-
>fetch(PDO::FETCH_ASSOC)) {
                                                        echo "<a class='list-
group-item' href='device.php?deviceID=". $row["deviceID"]."'> Device No.
".$row["deviceID"]."</a>";
                                                }
                                        ?>
                                        <a href="../adddevice.php"
class="list-group-item"><span class="glyphicon glyphicon-plus"></span> Add a
Device</a>
                                        </ul>
                                </div>
                        </div>

                        <!--Mobile Table -->
                        <div class="visible-xs col-md-10">
                                <center>
                                <button data-toggle="collapse" data-
target="#patSumTable" class="btn btn-default"> <span class="glyphicon
glyphicon-th-list"></span> Show Device Summary Table</button>
                                </center>
                                <br>
                                <div id="patSumTable" class="collapse">
                                        <div class="panel panel-default">
                                 <div class="panel-heading">Device
Summary</div>

                                 <div class="panel-body">
                        <table class="table">
                                <thead>
                                 <tr>
                                        <th>Patient Name</td>
                                        <th>Temperature Monitoring
Status</th>
                                        <th>Doppler Monitoring Status</th>
                                        <th>Last Doppler Recording</th>
                                 </tr>
                                </thead>
                                <tbody>
                         <?php
                                        $stmt = $db->prepare('SELECT
a.patientID,a.lastname,a.firstname,b.temperature,b.doppler FROM patients a
LEFT JOIN deviceStatus b ON a.patientID = b.patientID ORDER BY patientID
DESC');
                                        $stmt->execute(array());

                                        $results = $stmt-
>fetchAll(PDO::FETCH_ASSOC);
                                        foreach($results as $row){
                                                echo "<tr>";
```

```
								echo "<td>".
$row["lastname"].",".$row["firstname"]."</td>";
								if($row[temperature]==NULL){
									echo "<td> Inactive
</td>";
								}
								else if ($row[temperature]){
									echo "<td> Running
</td>";
								}
								else {
									echo "<td> Stopped
</td>";
								}
								if($row[doppler]==NULL){
									echo "<td> Inactive
</td>";
								}
								else if ($row[doppler]){
									echo "<td> Running
</td>";
								}
								else {
									echo "<td> Stopped
</td>";
								}
								$stmt = $db->prepare('SELECT
epoch FROM doppler WHERE patientID = :patientID ORDER BY epoch DESC LIMIT
1');
								$stmt-
>execute(array(':patientID' => $row['patientID']));
								$row2 = $stmt-
>fetch(PDO::FETCH_ASSOC);
								if ($row2['epoch']==NULL){
									echo "<td>No
Recordings Found</td>";
								}
								else if (round((time()-
$row2['epoch'])/60)){
									echo
"<td>".round((time()-$row2['epoch'])/60)." minutes ago</td>";
								}
								echo "</tr>";
							}

				?>

					</tbody>
				</table>
				</div>
```

```
                        </div>
                        </div>
                        </div>


                         <div class="hidden-xs col-md-10">
                         <!-- Right Column -->
                         <div class="panel panel-default">
                                  <div class="panel-heading">Device
Summary</div>
                                  <div class="panel-body">
                         <table class="table">
                                 <thead>
                                  <tr>
                                        <th>Patient Name</td>
                                        <th>Temperature Monitoring
Status</th>
                                        <th>Doppler Monitoring Status</th>
                                        <th>Last Doppler Recording</th>
                                  </tr>
                                 </thead>
                                </tbody>
                          <?php
                                        $stmt = $db->prepare('SELECT
a.patientID,a.lastname,a.firstname,b.temperature,b.doppler FROM patients a
LEFT JOIN deviceStatus b ON a.patientID = b.patientID ORDER BY patientID
DESC');
                                        $stmt->execute(array());

                                        $results = $stmt-
>fetchAll(PDO::FETCH_ASSOC);
                                        foreach($results as $row){
                                                echo "<tr>";
                                                echo "<td>".
$row["lastname"].",".$row["firstname"]."</td>";
                                                if($row[temperature]==NULL){
                                                        echo "<td> Inactive
</td>";
                                                }
                                                else if ($row[temperature]){
                                                        echo "<td> Running
</td>";
                                                }
                                                else {
                                                        echo "<td> Stopped
</td>";
                                                }
                                                if($row[doppler]==NULL){
                                                        echo "<td> Inactive
</td>";
```

```
}
else if ($row[doppler]){
        echo "<td> Running
</td>";
}
else {
        echo "<td> Stopped
</td>";
}
$stmt = $db->prepare('SELECT
epoch FROM doppler WHERE patientID = :patientID ORDER BY epoch DESC LIMIT
1');
$stmt-
>execute(array(':patientID' => $row['patientID']));
$row2 = $stmt-
>fetch(PDO::FETCH_ASSOC);
if ($row2['epoch']==NULL){
        echo "<td>No
Recordings Found</td>";
}
else if (round((time()-
$row2['epoch'])/60)){
        echo
"<td>".round((time()-$row2['epoch'])/60)." minutes ago</td>";
}
echo "</tr>";
    }

?>

            </tbody>
        </table>
        </div>
        </div>
         </div>
      </div>
   </div>
   </div>
  <hr>

  <footer>
  <p>Â© CareFreeÂ² 2016</p>
  </footer>
  </div>


 <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
```

```
 <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></scr
ipt>
 <!-- Include all compiled plugins (below), or include individual files as
needed -->
 <script src="js/bootstrap.min.js"></script>
 </body>
</html>
```

E.8 device.php

```
<?php require('/var/www/html/dennis/includes/config.php');

//if not logged in redirect to login page
if(!$user->is_logged_in()){ header('Location: ../index.php'); }

$deviceID = $_GET['deviceID'];
?>

<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <!-- The above 3 meta tags *must* come first in the head; any other head
content must come *after* these tags -->
 <title>Devices- CareFreeÂ²</title>

 <!-- Bootstrap -->
 <link href="css/bootstrap.min.css" rel="stylesheet">

 <!--Additional Stylesheet -->
 <link href="stylesheetNew.css" rel="stylesheet">

 <!-- Checks current page -->
<script src="http://code.jquery.com/jquery-1.8.3.min.js"></script>
<script>
$(function(){
  $('a').each(function() {
    if ($(this).prop('href') == window.location.href) {
      $(this).addClass('current');
    }
  });
});
</script>

 <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
queries -->
 <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
 <!--[if lt IE 9]>
 <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
 <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
 <![endif]-->
 </head>
 <body>
         <nav class="navbar navbar-default">
```

```
        <div class="container-fluid">

        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-
expanded="false">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        </button>

<a class="navbar-brand" href="index.php"><span
id="logoText1">CareFreeÂ²</span><br><span id="logoText2">Flap Monitoring
System</span></a>

        </div>

        <!-- Collect the nav links, forms, and other content for toggling --
>
        <div class="collapse navbar-collapse" id="bs-example-navbar-
collapse-1">
        <ul class="nav navbar-nav" id="nav">
                <li><a href="index.php"><span class="glyphicon glyphicon-
home"></span> Home</a></li>
                <li><a href="patients.php"><span class="glyphicon glyphicon-
user"></span> Patients</a></li>
                <li class="active"><a href="devices.php"><span
class="glyphicon glyphicon-phone"></span> Devices<span class="sr-
only">(current)</span></a></li>
        </ul>

        <!-- code for the right side of the navbar -->
        <ul class="nav navbar-nav navbar-right" id="nav">
               <li class="dropdown">
                       <a href="#" class="dropdown-toggle" data-
toggle="dropdown" role="button" aria-haspopup="true" aria-
expanded="false"><span class="glyphicon glyphicon-cog"></span> Account<span
class="glyphicons glyphicons-user"></span><span class="caret"></span></a>
                               <ul class="dropdown-menu">
                               <li><a href="settings.php">Settings</a></li>
                               <li><a href="logout.php">Logout</a></li>
                               </ul>
               </li>
        </ul>
        </div><!-- /.navbar-collapse -->
        </div><!-- /.container-fluid -->
        </nav>
```

```
        <div class="container">
        <div class="row">
                <div class="col-md-12">
                <div class="row">
                        <!--Patient Name Menu Collapses on Mobile-->
                        <div class="visible-xs col-md-2">
                                <center><h3>
                                        <?php
                                        $stmt = $db->prepare('SELECT deviceID
FROM deviceStatus WHERE deviceID = :deviceID');
                                        $stmt-
>execute(array(':deviceID'=>$deviceID));
                                        while ($row = $stmt-
>fetch(PDO::FETCH_ASSOC)) {
                                        echo "Device No. ".$row["deviceID"];
                                        }
                                        ?>

                                </h3></center>
                                <center>
                                <button data-toggle="collapse" data-
target="#patMenu" class="btn btn-default"> <span class="glyphicon glyphicon-
menu-hamburger"></span> Switch Device</button>
                                </center>
                                <br>
                               <div id="patMenu" class="collapse">
                                        <ul id="pat">
                                        <?php
                                                $stmt = $db->prepare('SELECT
deviceID FROM deviceStatus');
                                                $stmt->execute(array());
                                                while ($row = $stmt-
>fetch(PDO::FETCH_ASSOC)) {
                                                        echo "<a class='list-
group-item' href='device.php?deviceID=". $row["deviceID"]."'> Device No.
".$row["deviceID"]."</a>";
                                                }
                                        ?>
                                        <a href="../adddevice.php"
class="list-group-item"><span class="glyphicon glyphicon-plus"></span> Add a
Device</a>
                                        </ul>
                                        <br>
                               </div>
                        </div>


                        <div class="hidden-xs col-md-2">
                        <!-- Left Column -->
                           <div class="list-group">
```

```
                                        <center><h2>Devices</h2></center>
                                        <ul id="pat">
                                        <?php
                                                $stmt = $db->prepare('SELECT
deviceID FROM deviceStatus');
                                                $stmt->execute(array());
                                                while ($row = $stmt-
>fetch(PDO::FETCH_ASSOC)) {
                                                        echo "<a class='list-
group-item' href='device.php?deviceID=". $row["deviceID"]."'> Device No.
".$row["deviceID"]."</a>";
                                                }
                                        ?>
                                        <a href="../adddevice.php"
class="list-group-item">Add a Device</a>
                                        </ul>
                                </div>
                        </div>



                        <!--Desktop Control Panel -->
                        <div class="hidden-xs col-md-10">
                        <!-- Right Column -->
                        <div class="panel panel-default">
                                <div class="panel-heading">Control
Panel</div>
                                <div class="panel-body">
                                        <h4>Assign Device to a Patient</h4>
                                        <div class="well well-sm">
                                        <table class="deviceControlTable">
                                                <thead>
                                                        <tr>
                                                                <th
class="row-1 row-descrip"></th>
                                                                <th
class="row-2 row-button"></th>
                                                                <th
class="row-3 row-icon"></th>
                                                                <th
class="row-4 row-status"></th>
                                                        </tr>
                                                </thead>
                                                <tbody>
                                                        <tr>
                                                                <td>Choose a
patient to assign to this device</td>
                                                                <td>
                                                                        <form
role="form" method="POST" action="" autocomplete="off">
```

```
        <select class="form-control" name="var"
onchange="this.form.submit();">

                <option value="" disabled selected hidden>Select
Patient</option>

                <?php

                $submittedValue;


                                                                        $stmt
= $db->prepare('SELECT patientID,lastname,firstname FROM patients ORDER BY
patientID DESC');

        $stmt->execute(array());
                                                                        while
($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
                                                                        echo
"<option class='patientDropdownSub' value=". $row["patientID"]."> ".
$row["lastname"].",".$row["firstname"]."</option>";

                }

                ?>



        </select>



                                                                        <?php
                                                                        if
(isset($_POST['var']))
                                                                        {

        $patientID = $_POST['var'];



        $stmt = $db->prepare('UPDATE deviceRequest SET patientID = :patientID
WHERE deviceID = :deviceID' );

        $stmt-
>execute(array(':patientID'=>$patientID,':deviceID'=>$deviceID));

                                                                        }
                                                                        ?>
                                                                </form>
```

```
                                                                </td>
                                                                <td><font
color="2985a3"><span class="glyphicon glyphicon-user"></span></font></td>

                                                                <td>
                                                                        <font
color="2985a3">
                                                                <?php

        //$deviceID=1;
                                                                        $stmt
= $db->prepare('SELECT
a.patientID,a.lastname,a.firstname,b.temperature,b.doppler FROM patients a
LEFT JOIN deviceStatus b ON a.patientID = b.patientID WHERE deviceID =
:deviceID');

        $stmt->execute(array(':deviceID'=>$deviceID));
                                                                        while
($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
                                                                        echo
"Currently assigned to <a href='patient.php?patientID=". $row["patientID"]."'
id='activeLink'><b>". $row["lastname"].",".$row["firstname"]."</b></a>";
                                                                        }

                                                                ?>

        </font>
                                                                </td>
                                                        </tr>
                                                </tbody>
                                                </table>
                                                </div>
<h4>Combination Doppler-Temperature Probes</h4>
                                                <div class="well well-sm">
                                                <table
class="deviceControlTable">
                                                        <thead>
                                                        <tr>
                                                                <th
class="row-1 row-descrip"></th>
                                                                <th
class="row-2 row-button"></th>
                                                                <th
class="row-3 row-icon"></th>
                                                                <th
class="row-4 row-status"></th>
                                                                </tr>
                                                        </thead>
                                                <tbody>
                                                        <tr>
```

```
                                                                <td>Combo
Probe 1:</td>
                                                                <td>
                                                                        <form
method="POST" action="">

        <div class="btn-group">

        <button class="btn btn-default btn-md" type="submit" name="dop1"
value="1">On</button>

        <button class="btn btn-default btn-md" type="submit" name="dop1"
value="0">Off</button>

        </div>

        </form>

                                                                        <?php
                                                                        if
(isset($_POST['dop1']))
                                                                        {

        $dopRequest = $_POST['dop1'];

                                                                        $stmt
= $db->prepare('SELECT
a.patientID,a.lastname,a.firstname,b.temperature,b.doppler FROM patients a
LEFT JOIN deviceStatus b ON a.patientID = b.patientID WHERE deviceID =
:deviceID');

        $stmt->execute(array(':deviceID'=>$deviceID));


        $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
                                                                        //ON
                                                                        if
($dopRequest==1){

        foreach($results as $row){

                if(($row[doppler]==2)or($row[doppler]==3)){

                        $stmt = $db->prepare('UPDATE deviceRequest SET
doppler = :dopRequest WHERE deviceID = :deviceID' );

                        $stmt-
>execute(array(':dopRequest'=>3,':deviceID'=>$deviceID));

                }
```

```
                else {

                        $stmt = $db->prepare('UPDATE deviceRequest SET
doppler = :dopRequest WHERE deviceID = :deviceID' );

                        $stmt-
>execute(array(':dopRequest'=>1,':deviceID'=>$deviceID));

                }

        }
                                                                }
                                                                //OFF
                                                                else
if ($dopRequest==0){

        foreach($results as $row){

                if(($row[doppler]==2)or($row[doppler]==3)){

                        $stmt = $db->prepare('UPDATE deviceRequest SET
doppler = :dopsRequest WHERE deviceID = :deviceID' );

                        $stmt-
>execute(array(':dopsRequest'=>2,':deviceID'=>$deviceID));

                }

                else {

                        $stmt = $db->prepare('UPDATE deviceRequest SET
doppler = :dopsRequest WHERE deviceID = :deviceID' );

                        $stmt-
>execute(array(':dopsRequest'=>0,':deviceID'=>$deviceID));

                }

        }
                                                                }
                                                        }
                                                                ?>
                                                        </td>
                                                        <td><font
color="2985a3">
                                                        <?php

        //$deviceID=1;
```

```
                                                                                    $stmt
= $db->prepare('SELECT
a.patientID,a.lastname,a.firstname,b.temperature,b.doppler FROM patients a
LEFT JOIN deviceStatus b ON a.patientID = b.patientID WHERE deviceID =
:deviceID');

        $stmt->execute(array(':deviceID'=>$deviceID));

        $results = $stmt->fetchAll(PDO::FETCH_ASSOC);

        foreach($results as $row){

        if($row[doppler]==NULL){

                echo "<span class='glyphicon glyphicon-remove'></span>";

        }

        else if ($row[doppler]==1){

                echo "<span class='glyphicon glyphicon-play'></span>";

        }

        else if ($row[doppler]==3){

                echo "<span class='glyphicon glyphicon-play'></span>";

        }

        else {

                echo "<span class='glyphicon glyphicon-stop'></span>";

        }
                                                                                    }
                                                                             ?>
                                                                             </font></td>
                                                                            <td>
                                                                                    <font
color="2985a3">Combo probe 1 is currently
                                                                            <?php

        //$deviceID=1;
                                                                                    $stmt
= $db->prepare('SELECT
a.patientID,a.lastname,a.firstname,b.temperature,b.doppler FROM patients a
LEFT JOIN deviceStatus b ON a.patientID = b.patientID WHERE deviceID =
:deviceID');
```

```
        $stmt->execute(array(':deviceID'=>$deviceID));

        $results = $stmt->fetchAll(PDO::FETCH_ASSOC);

        foreach($results as $row){

        if($row[doppler]==NULL){

                echo "inactive";

        }

        else if ($row[doppler]==1){

                echo "running";

        }

        else if ($row[doppler]==3){

                echo "running";

        }

        else {

                echo "stopped";

        }
                                                                }
                                                         ?>
                                                         </font>
                                                        </td>
                                                </tr>
                                                <tr>
                                                        <td>Combo
Probe 2:</td>
                                                        <td>
                                                                <form
method="POST" action="">

        <div class="btn-group">

        <button class="btn btn-default btn-md" type="submit" name="dop2"
value="2">On</button>

        <button class="btn btn-default btn-md" type="submit" name="dop2"
value="0">Off</button>
```

```
        </div>

        </form>
                                                                        <?php
                                                                        if
(isset($_POST['dop2']))
                                                                        {

        $dopRequest = $_POST['dop2'];

                                                                        $stmt
= $db->prepare('SELECT
a.patientID,a.lastname,a.firstname,b.temperature,b.doppler FROM patients a
LEFT JOIN deviceStatus b ON a.patientID = b.patientID WHERE deviceID =
:deviceID');

        $stmt->execute(array(':deviceID'=>$deviceID));


        $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
                                                                        //ON
                                                                        if
($dopRequest==2){

        foreach($results as $row){

                if(($row[doppler]==1)or($row[doppler]==3)){

                        $stmt = $db->prepare('UPDATE deviceRequest SET
doppler = :dopRequest WHERE deviceID = :deviceID' );

                        $stmt-
>execute(array(':dopRequest'=>3,':deviceID'=>$deviceID));

                }

                else {

                        $stmt = $db->prepare('UPDATE deviceRequest SET
doppler = :dopRequest WHERE deviceID = :deviceID' );

                        $stmt-
>execute(array(':dopRequest'=>$dopRequest,':deviceID'=>$deviceID));

                }

        }
                                                                        }
```

```
                                                                      //OFF
                                                                      else
if ($dopRequest==0){

        foreach($results as $row){

                if(($row[doppler]==1)or($row[doppler]==3)){

                        $stmt = $db->prepare('UPDATE deviceRequest SET
doppler = :dopRequest WHERE deviceID = :deviceID' );

                        $stmt-
>execute(array(':dopRequest'=>1,':deviceID'=>$deviceID));

                }

                else {

                        $stmt = $db->prepare('UPDATE deviceRequest SET
doppler = :dopRequest WHERE deviceID = :deviceID' );

                        $stmt-
>execute(array(':dopRequest'=>$dopRequest,':deviceID'=>$deviceID));

                }

        }
                                                                      }
                                                              }
                                                                      ?>
                                                              </td>
                                                              <td><font
color="2985a3">
                                                              <?php

        //$deviceID=1;
                                                                      $stmt
= $db->prepare('SELECT
a.patientID,a.lastname,a.firstname,b.temperature,b.doppler FROM patients a
LEFT JOIN deviceStatus b ON a.patientID = b.patientID WHERE deviceID =
:deviceID');

        $stmt->execute(array(':deviceID'=>$deviceID));

        $results = $stmt->fetchAll(PDO::FETCH_ASSOC);

        foreach($results as $row){

        if($row[doppler]==NULL){
```

```
                echo "<span class='glyphicon glyphicon-remove'></span>";

        }

        else if($row[doppler]==2){

                echo "<span class='glyphicon glyphicon-play'></span>";

        }

        else if($row[doppler]==3){

                echo "<span class='glyphicon glyphicon-play'></span>";

        }

        else {

                echo "<span class='glyphicon glyphicon-stop'></span>";

        }
                                                                        }
                                                                 ?>
                                                                 </font></td>
                                                                <td>
                                                                        <font
color="2985a3">Combo probe 2 is currently
                                                                <?php

        //$deviceID=1;
                                                                        $stmt
= $db->prepare('SELECT
a.patientID,a.lastname,a.firstname,b.temperature,b.doppler FROM patients a
LEFT JOIN deviceStatus b ON a.patientID = b.patientID WHERE deviceID =
:deviceID');

        $stmt->execute(array(':deviceID'=>$deviceID));

        $results = $stmt->fetchAll(PDO::FETCH_ASSOC);

        foreach($results as $row){

        if($row[doppler]==NULL){

                echo "inactive";

        }

        else if ($row[doppler]==2){
```

```
                echo "running";

        }

        else if ($row[doppler]==3){

                echo "running";

        }

        else {

                echo "stopped";

        }
                                                                  }
                                                            ?>
                                                            </font>
                                                           </td>
                                                    </tr>
                                            </tbody>
                                            </table>
                                            </div>

                                            <h4>Reference Temperature
Probes</h4>
                                            <div class="well well-sm">
                                            <table
class="deviceControlTable">
                                                    <thead>
                                                    <tr>
                                                            <th
class="row-1 row-descrip"></th>
                                                            <th
class="row-2 row-button"></th>
                                                            <th
class="row-3 row-icon"></th>
                                                            <th
class="row-4 row-status"></th>
                                                            </tr>
                                                    </thead>
                                            <tbody>
                                            <tr>
                                                    <td>Reference
Temperature Probes:</td>
                                                    <td>
                                                            <form
method="POST" action="">
```

```
                                                                        <div
class="btn-group">

        <button class="btn btn-default btn-md" type="submit" name="temp"
value="1">On</button>

        <button class="btn btn-default btn-md" type="submit" name="temp"
value="0">Off</button>

        </div>
                                                                </form>

                                                                <?php
                                                                if
(isset($_POST['temp']))
                                                                {

        $tempRequest = $_POST['temp'];

                                                                        $stmt
= $db->prepare('UPDATE deviceRequest SET temperature = :tempRequest WHERE
deviceID = :deviceID' );

        $stmt-
>execute(array(':tempRequest'=>$tempRequest,':deviceID'=>$deviceID));
                                                                }
                                                                ?>
                                                        </td>
                                                        <td><font
color="2985a3">
                                                                <?php

        //$deviceID=1;
                                                                $stmt = $db-
>prepare('SELECT a.patientID,a.lastname,a.firstname,b.temperature,b.doppler
FROM patients a LEFT JOIN deviceStatus b ON a.patientID = b.patientID WHERE
deviceID = :deviceID');
                                                                $stmt-
>execute(array(':deviceID'=>$deviceID));
                                                                $results =
$stmt->fetchAll(PDO::FETCH_ASSOC);

        foreach($results as $row){

        if($row[temperature]==NULL){

        echo "<span class='glyphicon glyphicon-remove'></span>";
                                                                        }
                                                                        else
if ($row[temperature]){
```

```
        echo "<span class='glyphicon glyphicon-play'></span>";
                                                                }
                                                                else
{
        echo "<span class='glyphicon glyphicon-stop'></span>";
                                                                }

                                                        }
                                                 ?>
                                                 </font>
                                                 </td>
                                                <td>
                                                        <font
color="2985a3">Reference probes are currently
                                                <?php

        //$deviceID=1;
                                                        $stmt = $db-
>prepare('SELECT a.patientID,a.lastname,a.firstname,b.temperature,b.doppler
FROM patients a LEFT JOIN deviceStatus b ON a.patientID = b.patientID WHERE
deviceID = :deviceID');
                                                        $stmt-
>execute(array(':deviceID'=>$deviceID));
                                                        $results =
$stmt->fetchAll(PDO::FETCH_ASSOC);

        foreach($results as $row){

        if($row[temperature]==NULL){

        echo "inactive";
                                                                }
                                                                else
if ($row[temperature]){

        echo "running";
                                                                }
                                                                else
{

        echo "stopped";
                                                                }

                                                        }
                                                 ?>
                                                 </font>
                                                </td>
                                        </tr>
```

```
                                                </tbody>
                                                </table>
                                                </div>

                                        </div>
                                </div>

                                 </div>

                                <!--Mobile Control Panel -->
                                 <div class="visible-xs col-md-10">
                                 <!-- Right Column -->
                                 <div class="panel panel-default">
                                         <div class="panel-heading">Control
Panel</div>
                                         <div class="panel-body">
                                               <h4>Assign Device to a
Patient</h4>
                                               <div class="well well-sm">
                                                       <p>Choose a patient to assign
to this device:</p>
                                                       <p><form role="form"
method="POST" action="" autocomplete="off">

        <select class="form-control" name="var"
onchange="this.form.submit();">

                <option value="" disabled selected hidden>Select
Patient</option>

                <?php

                $submittedValue;



                $stmt = $db->prepare('SELECT patientID,lastname,firstname
FROM patients ORDER BY patientID DESC');

                $stmt->execute(array());

                while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {

                echo "<option class='patientDropdownSub' value=".
$row["patientID"]."> ". $row["lastname"].",".$row["firstname"]."</option>";

                }

                ?>
```

```
        </select>

                                                                        <?php
                                                                        if
(isset($_POST['var']))
                                                                        {

        $patientID = $_POST['var'];



        $stmt = $db->prepare('UPDATE deviceRequest SET patientID = :patientID
WHERE deviceID = :deviceID' );

        $stmt-
>execute(array(':patientID'=>$patientID,':deviceID'=>$deviceID));

                                                                        }
                                                                        ?>
                                                                </form>
                                                </p>
                                                <p><center><font
color="#2985a3"><span class="glyphicon glyphicon-user"></span> Currently
assigned to <?php

        //$deviceID=1;
                                                                        $stmt
= $db->prepare('SELECT
a.patientID,a.lastname,a.firstname,b.temperature,b.doppler FROM patients a
LEFT JOIN deviceStatus b ON a.patientID = b.patientID WHERE deviceID =
:deviceID');

        $stmt->execute(array(':deviceID'=>$deviceID));
                                                                        while
($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
                                                                        echo
"<a href='patient.php?patientID=". $row["patientID"]."' id='activeLink'><b>".
$row["lastname"].",".$row["firstname"]."</b></a>";
                                                                        }

                                                                ?>
                                                </font>
                                                </center>
                                                </p>
                                                </div>
```

```
                                                            <h4>Combination Doppler-
Temperature Probes:</h4>
                                                            <div class="well well-sm">
                                                                    <p>Combo Probe 1:</p>
                                                                    <p><center><form
method="POST" action="">

        <div class="btn-group">

        <button class="btn btn-default btn-md" type="submit" name="dop1"
value="1">On</button>

        <button class="btn btn-default btn-md" type="submit" name="dop1"
value="0">Off</button>

        </div>

        </form></center>
                                                                                        <?php
                                                                                        if
(isset($_POST['dop1']))
                                                                                        {

        $dopRequest = $_POST['dop1'];

                                                                                        $stmt
= $db->prepare('SELECT
a.patientID,a.lastname,a.firstname,b.temperature,b.doppler FROM patients a
LEFT JOIN deviceStatus b ON a.patientID = b.patientID WHERE deviceID =
:deviceID');

        $stmt->execute(array(':deviceID'=>$deviceID));


        $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
                                                                                        //ON
                                                                                        if
($dopRequest==1){

        foreach($results as $row){

                if(($row[doppler]==2)or($row[doppler]==3)){

                        $stmt = $db->prepare('UPDATE deviceRequest SET
doppler = :dopRequest WHERE deviceID = :deviceID' );

                        $stmt-
>execute(array(':dopRequest'=>3,':deviceID'=>$deviceID));
```

```
                }

                else {

                        $stmt = $db->prepare('UPDATE deviceRequest SET
doppler = :dopRequest WHERE deviceID = :deviceID' );

                        $stmt-
>execute(array(':dopRequest'=>$dopRequest,':deviceID'=>$deviceID));

                }

        }
                                                                        }
                                                                        //OFF
                                                                        else
if ($dopRequest==0){

        foreach($results as $row){

                if(($row[doppler]==2)or($row[doppler]==3)){

                        $stmt = $db->prepare('UPDATE deviceRequest SET
doppler = :dopRequest WHERE deviceID = :deviceID' );

                        $stmt-
>execute(array(':dopRequest'=>2,':deviceID'=>$deviceID));

                }

                else {

                        $stmt = $db->prepare('UPDATE deviceRequest SET
doppler = :dopRequest WHERE deviceID = :deviceID' );

                        $stmt-
>execute(array(':dopRequest'=>$dopRequest,':deviceID'=>$deviceID));

                }

        }
                                                                        }
                                                                }
                                                                        ?>
                                                        </p>
                                                        <p><center><font
color="#2985a3">
                                                                <?php
```

```
        //$deviceID=1;
                                                                                    $stmt
= $db->prepare('SELECT
a.patientID,a.lastname,a.firstname,b.temperature,b.doppler FROM patients a
LEFT JOIN deviceStatus b ON a.patientID = b.patientID WHERE deviceID =
:deviceID');

        $stmt->execute(array(':deviceID'=>$deviceID));

        $results = $stmt->fetchAll(PDO::FETCH_ASSOC);

        foreach($results as $row){

        if($row[doppler]==NULL){

                echo "<span class='glyphicon glyphicon-remove'></span> Combo
probe 1 is currently inactive";

        }

        else if ($row[doppler]==1){

                echo "<span class='glyphicon glyphicon-play'></span> Combo
probe 1 is currently running";

        }

        else if ($row[doppler]==3){

                echo "<span class='glyphicon glyphicon-play'></span> Combo
probe 1 is currently running";

        }

        else {

                echo "<span class='glyphicon glyphicon-stop'></span> Combo
probe 1 is currently stopped";

        }
                                                                                }
                                                                         ?>

</font></center></p>
                                                                         <br>

                                                              <p>Combo Probe 2:</p>
                                                              <p><center><form
method="POST" action="">
```

```
        <div class="btn-group">

        <button class="btn btn-default btn-md" type="submit" name="dop2"
value="1">On</button>

        <button class="btn btn-default btn-md" type="submit" name="dop2"
value="0">Off</button>

        </div>

        </form></center>
                                                                        <?php
                                                                                if
(isset($_POST['dop2']))
                                                                                {

        $dopRequest = $_POST['dop2'];

                                                                                $stmt
= $db->prepare('SELECT
a.patientID,a.lastname,a.firstname,b.temperature,b.doppler FROM patients a
LEFT JOIN deviceStatus b ON a.patientID = b.patientID WHERE deviceID =
:deviceID');

        $stmt->execute(array(':deviceID'=>$deviceID));


        $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
                                                                                //ON
                                                                                if
($dopRequest==2){

        foreach($results as $row){

                if(($row[doppler]==1)or($row[doppler]==3)){

                        $stmt = $db->prepare('UPDATE deviceRequest SET
doppler = :dopRequest WHERE deviceID = :deviceID' );

                        $stmt-
>execute(array(':dopRequest'=>3,':deviceID'=>$deviceID));

                }

                else {

                        $stmt = $db->prepare('UPDATE deviceRequest SET
doppler = :dopRequest WHERE deviceID = :deviceID' );
```

```
                        $stmt-
>execute(array(':dopRequest'=>$dopRequest,':deviceID'=>$deviceID));

                }

        }
                                                                        }
                                                                        //OFF
                                                                        else
if ($dopRequest==0){

        foreach($results as $row){

                if(($row[doppler]==1)or($row[doppler]==3)){

                        $stmt = $db->prepare('UPDATE deviceRequest SET
doppler = :dopRequest WHERE deviceID = :deviceID' );

                        $stmt-
>execute(array(':dopRequest'=>1,':deviceID'=>$deviceID));

                }

                else {

                        $stmt = $db->prepare('UPDATE deviceRequest SET
doppler = :dopRequest WHERE deviceID = :deviceID' );

                        $stmt-
>execute(array(':dopRequest'=>$dopRequest,':deviceID'=>$deviceID));

                }

        }
                                                                        }
                                                                }
                                                                        ?>
                                                        </p>
                                                        <p><center><font
color="#2985a3">
                                                                <?php

        //$deviceID=1;
                                                                        $stmt
= $db->prepare('SELECT
a.patientID,a.lastname,a.firstname,b.temperature,b.doppler FROM patients a
LEFT JOIN deviceStatus b ON a.patientID = b.patientID WHERE deviceID =
:deviceID');
```

```
        $stmt->execute(array(':deviceID'=>$deviceID));

        $results = $stmt->fetchAll(PDO::FETCH_ASSOC);

        foreach($results as $row){

        if($row[doppler]==NULL){

                echo "<span class='glyphicon glyphicon-remove'></span> Combo
probe 2 is currently inactive";

        }

        else if ($row[doppler]==2){

                echo "<span class='glyphicon glyphicon-play'></span> Combo
probe 2 is currently running";

        }

        else if ($row[doppler]==3){

                echo "<span class='glyphicon glyphicon-play'></span> Combo
probe 2 is currently running";

        }

        else {

                echo "<span class='glyphicon glyphicon-stop'></span> Combo
probe 2 is currently stopped";

        }
                                                                  }
                                                               ?>

</font></center></p>
                                                               <br>

                                                              </div>


                                                <h4>Reference Temperature
Probes</h4>
                                                <div class="well well-sm">
                                                        <p>Reference
Temperature Probes:</p>
                                                        <p>
```

```
                                                                 <center><form
method="POST" action="">
                                                                        <div
class="btn-group">

        <button class="btn btn-default btn-md" type="submit" name="temp"
value="1">On</button>

        <button class="btn btn-default btn-md" type="submit" name="temp"
value="0">Off</button>

        </div>

        </form></center>
                                                                 <?php
                                                                 if
(isset($_POST['temp']))
                                                                 {

        $tempRequest = $_POST['temp'];

                                                                        $stmt
= $db->prepare('UPDATE deviceRequest SET temperature = :tempRequest WHERE
deviceID = :deviceID' );

        $stmt-
>execute(array(':tempRequest'=>$tempRequest,':deviceID'=>$deviceID));
                                                                 }
                                                                 ?>
                                                         </p>
                                                         <p><center><font
color="#2985a3">
                                                         <?php

        //$deviceID=1;
                                                                 $stmt = $db-
>prepare('SELECT a.patientID,a.lastname,a.firstname,b.temperature,b.doppler
FROM patients a LEFT JOIN deviceStatus b ON a.patientID = b.patientID WHERE
deviceID = :deviceID');
                                                                 $stmt-
>execute(array(':deviceID'=>$deviceID));
                                                                 $results =
$stmt->fetchAll(PDO::FETCH_ASSOC);

        foreach($results as $row){

        if($row[temperature]==NULL){

        echo "<span class='glyphicon glyphicon-remove'></span> Reference
probes are currently inactive";
```

```
                                                                        }
                                                                        else
if ($row[temperature]){

        echo "<span class='glyphicon glyphicon-play'></span> Reference probes
are currently running";
                                                                        }
                                                                        else
{

        echo "<span class='glyphicon glyphicon-stop'></span> Reference probes
are currently stopped";
                                                                        }

                                                                }
                                                         ?>
                                                        </font></center></p>
                                                </div>



                </div>

         </div>
         </div>
        <hr>

        <footer>
        <p>Â© CareFreeÂ² 2016</p>
        </footer>
        </div>



 <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
 <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></scr
ipt>
 <!-- Include all compiled plugins (below), or include individual files as
needed -->
 <script src="js/bootstrap.min.js"></script>
 </body>
</html>
```

E.9 settings.php

```
<?php require('includes/config.php');

//if not logged in redirect to login page
if(!$user->is_logged_in()){ header('Location: index.php'); }



//if form has been submitted process it
if(isset($_POST['submit'])){

	//basic validation
	if(strlen($_POST['password']) < 3){
		$error[] = 'Password is too short.';
	}

	if(strlen($_POST['passwordConfirm']) < 3){
		$error[] = 'Confirm password is too short.';
	}

	if($_POST['password'] != $_POST['passwordConfirm']){
		$error[] = 'Passwords do not match.';
	}

	//if no errors have been created carry on
	if(!isset($error)){

		//hash the password
		$hashedpassword = $user->password_hash($_POST['password'],
PASSWORD_BCRYPT);

		try {

			$stmt = $db->prepare("UPDATE members SET password =
:hashedpassword, resetComplete = 'Yes'  WHERE resetToken = :token");
			$stmt->execute(array(
				':hashedpassword' => $hashedpassword,
				':token' => $row['resetToken']
			));

			//redirect to index page
			header('Location: login.php?action=resetAccount');
			exit;

		//else catch the exception and show the error.
		} catch(PDOException $e) {
		    $error[] = $e->getMessage();
		}
```

```
        }

}



?>

<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <!-- The above 3 meta tags *must* come first in the head; any other head
content must come *after* these tags -->
 <title>Settings- CareFreeÂ²</title>

 <!-- Bootstrap -->
 <link href="css/bootstrap.min.css" rel="stylesheet">

 <!--Additional Stylesheet -->
 <link href="stylesheetNew.css" rel="stylesheet">

 <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
queries -->
 <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
 <!--[if lt IE 9]>
 <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
 <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
 <![endif]-->
 </head>
 <body>

                 <nav class="navbar navbar-default">
         <div class="container-fluid">

         <!-- Brand and toggle get grouped for better mobile display -->
         <div class="navbar-header">
         <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-
expanded="false">
         <span class="sr-only">Toggle navigation</span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
         </button>
```

```
<a class="navbar-brand" href="index.php"><span
id="logoText1">CareFreeÂ²</span><br><span id="logoText2">Flap Monitoring
System</span></a>

         </div>

         <!-- Collect the nav links, forms, and other content for toggling --
>
         <div class="collapse navbar-collapse" id="bs-example-navbar-
collapse-1">
         <ul class="nav navbar-nav" id="nav">
                 <li><a href="index.php"><span class="glyphicon glyphicon-
home"></span> Home</a></li>
                 <li><a href="patients.php"><span class="glyphicon glyphicon-
user"></span> Patients</a></li>
                 <li><a href="devices.php"><span class="glyphicon glyphicon-
phone"></span> Devices</a></li>
         </ul>

         <!-- code for the right side of the navbar -->
         <ul class="nav navbar-nav navbar-right" id="nav">
                <li class="dropdown">
                        <a href="#" class="dropdown-toggle" data-
toggle="dropdown" role="button" aria-haspopup="true" aria-
expanded="false"><span class="glyphicon glyphicon-cog"></span> Account<span
class="sr-only">(current)</span><span class="caret"></span></a>
                                <ul class="dropdown-menu">
                                <li><a href="settings.php">Settings</a></li>
                                <li><a href="logout.php">Logout</a></li>
                                </ul>
                </li>
         </ul>
         </div><!-- /.navbar-collapse -->
         </div><!-- /.container-fluid -->
         </nav>

         <div class="container">
         <div class="row">
                <div class="col-md-6 col-md-offset-3">
                        <div class="panel panel-default">
                                 <div class="panel-heading">Account
Settings</div>
                                 <div class="panel-body">

                                         <?php if(isset($stop)){

                        echo "<p class='bg-danger'>$stop</p>";

                } else { ?>
```

```
                                <form role="form" method="post" action=""
autocomplete="off">
                                        <h4>Change Password</h4>
                                        <hr>

                                        <?php
                                        //check for any errors
                                        if(isset($error)){
                                                foreach($error as $error){
                                                        echo '<p class="bg-
danger">'.$error.'</p>';
                                                }
                                        }

                                        //check the action
                                        switch ($_GET['action']) {
                                                case 'active':
                                                        echo "<h2 class='bg-
success'>Your account is now active you may now log in.</h2>";
                                                        break;
                                                case 'reset':
                                                        echo "<h2 class='bg-
success'>Please check your inbox for a reset link.</h2>";
                                                        break;
                                        }
                                        ?>

                                        <div class="row">
                                                <div class="col-xs-6 col-sm-6
col-md-6">
                                                        <div class="form-
group">
                                                                <input
type="password" name="password" id="password" class="form-control"
placeholder="New Password" tabindex="1">
                                                        </div>
                                                </div>
                                                <div class="col-xs-6 col-sm-6
col-md-6">
                                                        <div class="form-
group">
                                                                <input
type="password" name="passwordConfirm" id="passwordConfirm" class="form-
control" placeholder="Confirm Password" tabindex="1">
                                                        </div>
                                                </div>
                                        </div>

                                        <div class="row">
```

```
                                                <div class="col-xs-6 col-md-
6"><input type="submit" name="submit" value="Change Password" class="btn btn-
default" id="buttonCustom" tabindex="3"></div>
                                        </div>
                                </form>

                        <?php } ?>

                                  </div>
                </div>

        </div>
        </div>
        <hr>

        <footer>
        <p>Â© CareFreeÂ² 2016</p>
        </footer>
        </div>



 <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
 <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></scr
ipt>
 <!-- Include all compiled plugins (below), or include individual files as
needed -->
 <script src="js/bootstrap.min.js"></script>
 </body>
</html>
```

E.10 resetPassword.php

```
<?php require('includes/config.php');

//if logged in redirect to members page
if( $user->is_logged_in() ){ header('Location: dashboard.php'); }

$stmt = $db->prepare('SELECT resetToken, resetComplete FROM members WHERE
resetToken = :token');
$stmt->execute(array(':token' => $_GET['key']));
$row = $stmt->fetch(PDO::FETCH_ASSOC);

//if no token from db then kill the page
if(empty($row['resetToken'])){
        $stop = 'Invalid token provided, please use the link provided in the
reset email.';
} elseif($row['resetComplete'] == 'Yes') {
        $stop = 'Your password has already been changed!';
}

//if form has been submitted process it
if(isset($_POST['submit'])){

        //basic validation
        if(strlen($_POST['password']) < 3){
                $error[] = 'Password is too short.';
        }

        if(strlen($_POST['passwordConfirm']) < 3){
                $error[] = 'Confirm password is too short.';
        }

        if($_POST['password'] != $_POST['passwordConfirm']){
                $error[] = 'Passwords do not match.';
        }

        //if no errors have been created carry on
        if(!isset($error)){

                //hash the password
                $hashedpassword = $user->password_hash($_POST['password'],
PASSWORD_BCRYPT);

                try {

                        $stmt = $db->prepare("UPDATE members SET password =
:hashedpassword, resetComplete = 'Yes'  WHERE resetToken = :token");
                        $stmt->execute(array(
                                ':hashedpassword' => $hashedpassword,
```

```
                                ':token' => $row['resetToken']
                        ));

                        //redirect to index page
                        header('Location: index.php?action=resetAccount');
                        exit;

                //else catch the exception and show the error.
                } catch(PDOException $e) {
                    $error[] = $e->getMessage();
                }

        }

}

?>
<!DOCTYPE html>
<html>
<head>
<title>Reset Account</title>
<link href="stylesheet.css" rel="stylesheet" type="text/css">

</head>
<div class="container">

        <div class="row">

            <div class="col-xs-12 col-sm-8 col-md-6 col-sm-offset-2 col-md-
offset-3">


                <?php if(isset($stop)){

                        echo "<p class='bg-danger'>$stop</p>";

                } else { ?>

                                <form role="form" method="post" action=""
autocomplete="off">
                                        <h2>Change Password</h2>
                                        <hr>

                                        <?php
                                        //check for any errors
                                        if(isset($error)){
                                                foreach($error as $error){
                                                        echo '<p class="bg-
danger">'.$error.'</p>';
                                                }
```

```
                                        }

                                        //check the action
                                        switch ($_GET['action']) {
                                                case 'active':
                                                        echo "<h2 class='bg-
success'>Your account is now active you may now log in.</h2>";
                                                        break;
                                                case 'reset':
                                                        echo "<h2 class='bg-
success'>Please check your inbox for a reset link.</h2>";
                                                        break;
                                        }
                                        ?>

                                        <div class="row">
                                                <div class="col-xs-6 col-sm-6
col-md-6">
                                                        <div class="form-
group">
                                                                <input
type="password" name="password" id="password" class="form-control input-lg"
placeholder="Password" tabindex="1">
                                                        </div>
                                                </div>
                                                <div class="col-xs-6 col-sm-6
col-md-6">
                                                        <div class="form-
group">
                                                                <input
type="password" name="passwordConfirm" id="passwordConfirm" class="form-
control input-lg" placeholder="Confirm Password" tabindex="1">
                                                        </div>
                                                </div>
                                        </div>

                                        <hr>
                                        <div class="row">
                                                <div class="col-xs-6 col-md-
6"><input type="submit" name="submit" value="Change Password" class="btn btn-
primary btn-block btn-lg" tabindex="3"></div>
                                        </div>
                                </form>

                        <?php } ?>
                </div>
        </div>


</div>
```

```
</body>
</html>
```

E.11 reset.php

```
<?php require('includes/config.php');

//if logged in redirect to members page
if( $user->is_logged_in() ){ header('Location: dashboard.php'); }

//if form has been submitted process it
if(isset($_POST['submit'])){

        //email validation
        if(!filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)){
            $error[] = 'Please enter a valid email address';
        } else {
                $stmt = $db->prepare('SELECT email FROM members WHERE email = 
:email');
                $stmt->execute(array(':email' => $_POST['email']));
                $row = $stmt->fetch(PDO::FETCH_ASSOC);

                if(empty($row['email'])){
                        $error[] = 'Email provided is not on recognised.';
                }

        }

        //if no errors have been created carry on
        if(!isset($error)){

                //create the activasion code
                $token = md5(uniqid(rand(),true));

                try {

                        $stmt = $db->prepare("UPDATE members SET resetToken = 
:token, resetComplete='No' WHERE email = :email");
                        $stmt->execute(array(
                                ':email' => $row['email'],
                                ':token' => $token
                        ));

                        //send email
                        $to = $row['email'];
                        $subject = "Password Reset";
                        $body = "<p>Someone requested that the password be 
reset.</p>
                        <p>If this was a mistake, just ignore this email and 
nothing will happen.</p>
                        <p>To reset your password, visit the following 
address: <a
```

```
href='".DIR."resetPassword.php?key=$token'>".DIR."resetPassword.php?key=$toke
n</a></p>";

                        $mail = new Mail();
                        $mail->setFrom(SITEEMAIL);
                        $mail->addAddress($to);
                        $mail->subject($subject);
                        $mail->body($body);
                        $mail->send();

                        //redirect to index page
                        header('Location: index.php?action=reset');
                        exit;

                //else catch the exception and show the error.
                } catch(PDOException $e) {
                    $error[] = $e->getMessage();
                }

        }

}

?>



<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <!-- The above 3 meta tags *must* come first in the head; any other head
content must come *after* these tags -->
 <title>CareFreeÂ²</title>

 <!-- Bootstrap -->
 <link href="css/bootstrap.min.css" rel="stylesheet">

 <link href="stylesheetNew.css" rel="stylesheet">

 <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
queries -->
 <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
 <!--[if lt IE 9]>
 <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
 <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
 <![endif]-->
```

```
 </head>
 <body>
        <nav class="navbar navbar-default">
        <div class="container-fluid">

        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">


<a class="navbar-brand" href="index.php"><span
id="logoText1">CareFreeÂ²</span><br><span id="logoText2">Flap Monitoring
System</span></a>

        </div>

        </div><!-- /.navbar-collapse -->
        </div><!-- /.container-fluid -->
        </nav>

        <div class="container">
        <div class="row">
        <div class="col-md-12">
       <div class="row">
                <div class="col-md-3">
                        <h2>Reset Password</h2>
                        <form role="form" method="post" action=""
autocomplete="off">
                                <?php
                                //check for any errors
                                if(isset($error)){
                                        foreach($error as $error){
                                                echo '<p class="bg-
danger">'.$error.'</p>';
                                        }
                                }
                                if(isset($_GET['action'])){

                                        //check the action
                                        switch ($_GET['action']) {
                                                case 'active':
                                                        echo "<h2 class='bg-
success'>Your account is now active you may now log in.</h2>";
                                                        break;
                                                case 'reset':
                                                        echo "<h2 class='bg-
success'>Please check your inbox for a reset link.</h2>";
                                                        break;
                                        }
                                }
                                ?>
```

```
                        <div class="form-group">
                        <label for="exampleInputEmail1">Email</label>
                        <input type="email" name="email" id="email"
class="form-control" placeholder="Email" value="" tabindex="1">
                        </div>

                        <div class="form-group">
                        <input class="btn btn-default" id="buttonCustom"
type="submit" name="submit" value="Send Reset Link"/>
                        </div>

                       </form>

                       <p><a id="activeLink" href='index.php'>Back to login
page</a></p>
                </div>
         </div>
         </div>
         </div>
        <hr>

        <footer>
        <p>Â© CareFreeÂ² 2016</p>
        </footer>
        </div>



 <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
 <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></scr
ipt>
 <!-- Include all compiled plugins (below), or include individual files as
needed -->
 <script src="js/bootstrap.min.js"></script>
 </body>
</html>
```

E.12 logout.php

```
<?php require('includes/config.php');

//logout
$user->logout();

//logged in return to index page
header('Location: index.php');
exit;
?>
```

E.13 stylesheetNew.css

```
body{
        color: #4C4C4C;
}

#pat {
        width: 100%;
    list-style: none;
        margin: 0px;
        padding: 0px;
        font-size: 15px;
        }

#pat li {
        float: center;
}

#pat li a {
        text-align: center;
        display: block;
        padding: 10px 0px;
        text-decoration: none;
        color: #4C4C4C;
        border-top: 1px solid #d9d9d9;
}

#pat li a:hover {
        background: #F5F5F5;
}

#pat a.current {
        background: #ebf6fa;
        color: #2985a3;
        border-right: 3px solid #2985a3;
}


#patTitle{
        font-weight: bold;
}


.patientTitle {
        border-bottom: 1px solid #ccc;
        padding-bottom: 5px;

}
```

```
.patient{
        padding-top: 10px;
        padding-bottom: 10px;

}



.form-control input-lg{
    background-color: #FFFFFF;
    border:solid 1px #BFBDBD;
    border-radius: 5px;
    height: 28px;
    margin-top:10px;
    padding-left:10px;
    width: 191px;
}

.button
{
    color: #FFFFFF;
    font-family: 'Montserrat-Regular';
    font-size: 14px;
    border:solid 1px #24748e;
    border-radius: 5px;
    margin-top:10px;
    background-color: #2985a3;
    padding: 7px;
    -webkit-appearance: none;

}

.button:hover {
        background: #80bf40;
        font-family: 'Montserrat-Regular';
        border-radius: 5px;
        border:solid 1px #90ac39;
        -webkit-appearance: none;

}

/*Device Control Panel Table*/
.deviceControlTable{
        border: none;
        width: 100%;
}

.deviceControlTable th, td{
        //padding-top: 5px;
        padding-bottom: 5px;
```

```
        padding-right: 10px;
}

/* Column widths are based on these cells */
.row-descrip {
  width: 40%;
}
.row-button {
  width: 30%;
}

.row-icon {
  width: 3%;
}
.row-status {
  width: 25%;
}


/*Patient and Device List Links*/
#pat{
        text-align: center;
}
#pat a.current {
        background: #ebf6fa;
        color: #2985a3;
        //border-right: 3px solid #2985a3;
}


/*Nav bar*/

#nav li a {
        color: #4C4C4C;
}

#nav li a:hover {
        color: #2985a3;
        //border-bottom: 1px solid #2985a3;
}

#activeLink{
        color: #2985a3;
}

#buttonCustom{
        color: white;
        background: #2985a3;
        border-color: #237794;
}
```

```
#buttonCustom:hover{
        color: white;
        background: #206B85;
        border-color: #1C5E75;
}

#alertRed{
        color: #DA3842;
        background: #FFE8E9;
}

#alertYellow{
        color: #F69A33;
        //background: #FFE5E7;
}

#alertBlue{
        background: #ebf6fa;
        color: #2985a3;
}

#alertGreen{
        background: #E7FCD1;
        color: #5E9527;
}

#logoText1{
        color: #2985A3;
        font-size: 22px;
        //font-weight: bold;
        padding: 0px;
}

#logoText2{
font-size: 13px;

}
```

E.13 retrieve.php

```
<?php
if ($_POST)
{
        $con = mysql_connect("localhost","dennis","donkeytree");

        if (!$con)
        {
                die('Could not connect: ' . mysql_error());
        }

mysql_select_db("dennis_stuff", $con);


$type = $_POST['type'];
$id = $_POST['id'];


$id = mysql_real_escape_string($id);
$type = mysql_real_escape_string($type);


$query = "SELECT probe, value FROM dennis_stuff.$type WHERE id=$id;";

$result = mysql_query($query);
if (!$result) {
    echo 'Could not run query: ' . mysql_error();
    exit;
}
$row = mysql_fetch_row($result);
echo $type;
echo " ";
echo $row[0]; // Probe
echo " ";
echo $row[1]; // Value
echo "\n";


//echo "Requested Measurement\n";
//var_dump($_POST);

mysql_close($con);
}
?>
```

E.14 retrieveStatus.php

```
<?php
if ($_POST)
{
        $con = mysql_connect("localhost","dennis","donkeytree");

        if (!$con)
        {
                die('Could not connect: ' . mysql_error());
        }

mysql_select_db("dennis_stuff", $con);


$deviceID = $_POST['deviceID'];

$deviceID = mysql_real_escape_string($deviceID);


$query = "SELECT patientID, temperature, doppler, dopplernow FROM
dennis_stuff.deviceRequest WHERE deviceID=$deviceID;";

$result = mysql_query($query);
if (!$result) {
    echo 'Could not run query: ' . mysql_error();
    exit;
}
$row = mysql_fetch_row($result);
echo $row[0]; //patientID
echo " ";
echo $row[1]; // temperature
echo " ";
echo $row[2]; // doppler
echo " ";
echo $row[3]; // dopplernow
echo "\n";


//echo "Requested Measurement\n";
//var_dump($_POST);

mysql_close($con);
}
?>
```

E.15 temperature.php

```
<?php
$patientID = $_GET['patientID'];
$db_host = 'localhost';
$db_database = 'dennis_stuff';
$db_user = 'dennis';
$db_password = 'donkeytree';

include('../../inc/db.php');
$db = mysql_connect($db_host, $db_user, $db_password);
mysql_select_db($db_database);

$sqlQuery = "SELECT epoch, value FROM dennis_stuff.temperature WHERE
patientID = $patientID ORDER BY epoch DESC LIMIT 100;";
$sqlResult = mysql_query($sqlQuery);
$currentTime = time();

$table = array(
    'cols' => array(
        array('label' => 'Time', 'type' => 'number'),
        array('label' => 'Temp. Âº C', 'type' => 'number')
    ),
    'rows' => array()
);

while ($row = mysql_fetch_assoc($sqlResult)) {
   if (($row['epoch']-$currentTime)/60 > -120){
    $table['rows'][] = array(
        'c' => array(
            array('v' => ($row['epoch']-$currentTime)/60),
            array('v' => $row['value'])
        )
    );
    }
}


echo json_encode($table);



?>
```

E.16 updateStatus.php

```
<?php
$patientID = $_GET['patientID'];
$db_host = 'localhost';
$db_database = 'dennis_stuff';
$db_user = 'dennis';
$db_password = 'donkeytree';

include('../../inc/db.php');
$db = mysql_connect($db_host, $db_user, $db_password);
mysql_select_db($db_database);

$sqlQuery = "SELECT epoch, value FROM dennis_stuff.temperature WHERE
patientID = $patientID ORDER BY epoch DESC LIMIT 100;";
$sqlResult = mysql_query($sqlQuery);
$currentTime = time();

$table = array(
    'cols' => array(
        array('label' => 'Time', 'type' => 'number'),
        array('label' => 'Temp. ÂºC', 'type' => 'number')
    ),
    'rows' => array()
);

while ($row = mysql_fetch_assoc($sqlResult)) {
   if (($row['epoch']-$currentTime)/60 > -120){
    $table['rows'][] = array(
        'c' => array(
            array('v' => ($row['epoch']-$currentTime)/60),
            array('v' => $row['value'])
        )
    );
    }
}


echo json_encode($table);



?>
```

E.17 last.php

```
<?php
if ($_POST)
{
        $con2 = mysql_connect("localhost","dennis","donkeytree");

        if (!$con2)
        {
                die('Could not connect: ' . mysql_error());
        }

mysql_select_db("dennis_stuff", $con2);


$type = $_POST['type'];

$type = mysql_real_escape_string($type);


$query = "SELECT id FROM dennis_stuff.$type ORDER BY id DESC LIMIT 1;";

$result = mysql_query($query);
if (!$result) {
    echo 'Could not run query: ' . mysql_error();
    exit;
}
$row = mysql_fetch_row($result);

echo $row[0]; // id
echo "\n";

mysql_close($con2);
}
?>
```