

j

CyVital

Design Document

Project 006

sdmay26-06

Dr. Meng Lu

Reza Choudhury

Kate Endersby

Max Tanruther

Claire Haas

Spring 2026

TABLE OF CONTENTS

Executive Summary	5
Learning Summary	5
Development Standards & Practices Used	5
Summary of Requirements	6
New Skills and Knowledge (not taught in courses)	6
Applicable Courses from Iowa State University Curriculum	6
1. Introduction	7
1.1. Problem Statement	7
1.2. Intended Users	7
2. Requirements, Constraints, And Standards	9
2.1. Requirements & Constraints	9
2.2. Engineering Standards	9
3 Project Plan	10
3.1 Project Management/Tracking Procedures	10
3.2 Task Decomposition	11
3.4 Project Timeline/Schedule	13
Figure 1: Semester Gantt Chart	13
Figure 2: Semester 2 Gantt Chart	14
3.5 Risks and Risk Management/Mitigation	14
3.6 Personnel Effort Requirements	15
Table 1: Man-Hour Estimates	16
Table 2: Actual Personnel Effort	16
3.7 Other Resource Requirements	16
Table 3: Bill of Materials	21
4 Design	22
4.1 Design Context	22
4.1.1 Broader Context	22
Table 4: Broader Context	22
4.1.2 Prior Work/Solutions	22
4.1.3 Technical Complexity	23
4.2 Design Exploration	23
4.2.1 Design Decisions	23
4.2.2 Ideation	24
4.2.3 Decision-Making and Trade-Off	24
4.3 Final Design	25
4.3.1 Overview	25
Figure 3: High-Level Design	25
4.3.2 Detailed Design and Visual(s)	26
Figure 4: High-Level Software Diagram	26

Figure 5: Sequence Diagram	27
Figure 6: Sample Plot	28
Figure 7: UI Layout	29
Figure 8: A Sparkfun Iteration of the PCB Board	29
Figure 9: Converts & Regulators Schematic	30
Figure 10: EKG Schematic	31
Figure 11: EMG Schematic	31
Figure 12: Biometric Hub & Pulse Oximeter Schematics	32
Figure 13: Reaction Time Schematic	33
Figure 14: 3D Rendering of PCB	34
4.3.3 Functionality	35
4.3.4 Areas of Challenge	35
4.4 Technology Considerations	35
5 Testing	37
5.1 Unit Testing	37
5.2 Interface Testing	37
5.3 Integration Testing	37
5.4 System Testing	37
5.5 Regression Testing	38
5.6 Acceptance Testing	38
5.7 Security Testing	38
5.8 User Testing	38
5.9 Results	38
Figure 15: Pulse Oximeter Live Test	39
5.9.1 PCB Testing	39
Figure 16: Power Management Assembly Step	40
Table 5: PCB Results	41
Figure 17: Power Management Assembly Step	41
Figure 18: Final Assembly Step	42
Figure 19: Reaction Time Graph	42
Figure 20: EKG Graph	43
5.9.2 Software Testing	43
Table 6: Pytest Code Coverage Report	44
6 Implementation	45
Figure 21: GUI Implementation	45
6.1 Design Analysis	45
Figure 22: Final PCB Case	46
7 Ethics and Professional Responsibility	47
7.1 Areas of Professional Responsibility/Codes of Ethics	47
Table 7: Professional Responsibility	48
7.2 Four Principles	49

Table 8: Four Principles	50
7.3 Virtues	50
8 Conclusion	52
8.1 Summary of Progress	52
8.2 Value Provided	52
8.3 Next Steps	52
9. References	52
10. Appendices	53
10.1 Appendix 1 - Operation Manuals	53
10.1.1 User Manuals	53
10.1.2 Hardware Manuals	53
10.1.3 Software References	54
10.2 Appendix 2 - Code	54
10.3 Appendix 3 - Team Contract	54
10.4 Appendix 4 - Design Analysis From Fall 2025	58

LIST OF FIGURES

Figure 1: Semester Gantt Chart	13
Figure 2: Semester 2 Gantt Chart	14
Figure 3: High-Level Design	25
Figure 4: High-Level Software Diagram	26
Figure 5: Sequence Diagram	27
Figure 6: Sample Plot	28
Figure 7: UI Layout	29
Figure 8: A Sparkfun Iteration of the PCB Board	29
Figure 9: Converts & Regulators Schematic	30
Figure 10: EKG Schematic	31
Figure 11: EMG Schematic	31
Figure 12: Biometric Hub & Pulse Oximeter Schematics	32
Figure 13: Reaction Time Schematic	33
Figure 14: 3D Rendering of PCB	34
Figure 15: Pulse Oximeter Live Test	39
Figure 16: Power Management Assembly Step	40
Figure 17: Power Management Assembly Step	41
Figure 18: Final Assembly Step	42
Figure 19: Reaction Time Graph	42
Figure 20: EKG Graph	43
Figure 21: GUI Implementation	45
Figure 22: Final PCB Case	46

LIST OF TABLES

Table 1: Man-Hour Estimates	16
Table 2: Actual Personnel Effort	16
Table 3: Bill of Materials	21
Table 4: Broader Context	22
Table 5: PCB Results	41
Table 6: Pytest Code Coverage Report	44
Table 7: Professional Responsibility	48
Table 8: Four Principles	50

Executive Summary

CyVital is a hands-on lab kit and simple app for Iowa State's revamped BME 3500: Bioinstrumentation. The kit allows students to use medical sensors on a human volunteer, measuring heart activity, blood oxygen, muscle signals, and reaction time. The user will immediately see volunteer results and introduce the concepts of biomedical signal analysis for health monitoring. The CyVital kit is designed for the whole classroom, and it is easy for students, teaching assistants (TAs), and professors. With additional instruction from the professor and teaching assistants, the student will be able to immediately learn by doing, rather than watch the instructors during the lab section. When students need help, the teaching assistant will be able to easily diagnose and fix issues. Most importantly, the professors have a custom solution that allows them to supplement course instruction without needing to use expensive alternative kits. The CyVital takes a hands-on approach, allowing students to focus on what is truly important about bioinstrumentation.

Prioritizing safety, affordability, and durability, the CyVital team worked on a low-cost custom hardware and software solution that can be maintained easily. The CyVital team worked in two teams, with one making software that collects and displays data, while the other designing the hardware for students to use. The software team utilized the Agile framework and focused on displaying data to the user using existing Python libraries. On the other hand, the hardware team used a conventional design routine, starting with circuit sketching and ending with a physical box containing all of the necessary equipment. This small box reads the sensor signals, while the app features live diagrams and the ability to export data. Within the box, a custom printed circuit board offers six biomedical instruments in a more compact form than other biomedical engineering lab kits. For durability and safety standards, the circuit board has a 3D case to protect the circuitry from potential damage and minimize risk to students. The CyVital app can be used on any Windows device and plugs into the lab kit with a basic USB Type C connector. This lab kit will be utilized by students enrolled in BME 3500 for the first time in the fall of 2026.

The team will work on expanding the CyVital lab and equipment manual to ensure users have access to information on all features of the CyVital. To continue supporting students' learning, the CyVital lab kit may also be offered to students beyond Iowa State. The CyVital is a key solution to helping improve biomedical education for students and instructors.

Learning Summary

DEVELOPMENT STANDARDS & PRACTICES USED

- IEC 62304:2006(en) - Medical device software - Software life cycle processes
- IEEE 11073-10700 - IEEE Standard – Health Informatics – Device Interoperability Part 10700: Point-of-Care Medical Device Communication – Standard for Base Requirements for Participants in a Service-Oriented Device Connectivity (SDC) System
- IEEE 11073-10406 - IEEE Standard – Health Informatics – Device Interoperability Part 10406: Personal Health Device Device Communication – Device Specialization - Basic Electrocardiography (ECG)
- IEEE 1012 - IEEE Standard for System, Software, and Hardware Verification and Validation
- Agile and Waterfall Methodology for Project Management

SUMMARY OF REQUIREMENTS

- Lower cost per kit than competitors
- Parts widely available
- Safe for classroom
- Live diagrams that display information from an Electrocardiogram (ECG), respiratory band (SpO₂), Electromyography (EMG), reaction time, and blood pressure cuff
- Runs on student laptops that support Windows operating systems
- Survives repeated labs
- Teaching Assistant friendly
- Data export for reports
- Responsive User Interface (UI) allowing students to collect data and analyze output plots from their signal processing functions
- Each sensor must be implemented and integrated onto a printed circuit board (PCB)

NEW SKILLS AND KNOWLEDGE (NOT TAUGHT IN COURSES)

- Knowledge of bioinstrumentation sensors
- Understanding ECG/SpO₂/EMG signal analysis

APPLICABLE COURSES FROM IOWA STATE UNIVERSITY CURRICULUM

While CyVital is developed for BME 3500: Bioinstrumentation, the team utilized knowledge from the following courses:

- Software Engineering:
 - COMS 2270: Introduction to Object Oriented Programming
 - COMS 2280: Introduction to Data Structures
 - COM S 3090: Software Development Practices
- Electrical Design
 - EE 2010: Electric Circuits
 - EE 2300: Electronic Circuits and Systems
- Signal Processing
 - EE 2240: Signals and Systems I
 - EE 3240: Signals and Systems II

APPLICABLE COURSES (CONTINUED)

- Embedded Systems
 - CPRE 2810: Digital Logic
 - CPRE 2880: Embedded Systems I: Introduction
- Project Management:
 - COM S 3090: Software Development Practices
 - ENGL 3140: Technical Communication

1. Introduction

1.1. PROBLEM STATEMENT

Many bioinstrumentation labs are expensive, fragile, and pre-built. While these systems can demonstrate physiological signals, they prevent students from interacting with the underlying, leaving students without a thorough understanding of the biomedical devices. With the existing bioinstrumentation kits, students are missing critical hands-on learning that comes from wiring something up and watching their own physiological data appear on a screen. Professors are frustrated with the limitations of regular training equipment as they try to prepare students for the healthcare industry and future coursework without going over their allotted budget. Because these kits can cost thousands of dollars, professors might be forced to choose between affordability and education depth. As healthcare technology advances, there is a growing need for healthcare workers to be proficient in using this equipment. Current lab alternatives fall short in equipping students with these practical, applied skills.

The project, CyVital, addresses this problem by providing classrooms with a low-cost, safe, and truly hands-on method for exploring real body signals, including heart activity, blood oxygen levels, muscle movement, respiration, and reaction times. Students connect the sensors themselves and see a clean, live graph that they can use to read and interpret the biomedical signal data. By emphasizing hands-on learning, this project aims to equip biomedical engineers with the necessary skills, and build confidence, scientific literacy, and practical skills that future biomedical professionals need.

Without tools like CyVital, students won't have the valuable opportunities to explore medical technologies by building the systems and analyzing the data. Our project addresses this issue by creating a platform that enables sensors to be integrated into a simple system. This allows the students to gain firsthand experience building and analyzing these basic medical sensors.

1.2. INTENDED USERS

User 1: Biomedical Engineering Student

Biomedical Engineering students are Iowa State students who want to gain a better understanding of physiological sensors and the data they produce. They aim to maximize the benefits of their education and gain a deeper understanding of biomedical sensors to enhance their future careers in this field.

Biomedical Engineering students need to have a good understanding of physiological sensors and the data they produce. They need to have access to the sensors themselves, rather than being hidden by proprietary protocols and materials. They need to have an easy-to-use GUI that lets them read and analyze the data returned by these sensors.

They will benefit from having increased user interaction with the biomedical sensors. The students will learn about the various sensors and have the opportunity to analyze the signals received from them. They will run the system on live subjects, which allows them to observe the real-world data from biomedical sensors. They will also learn the underlying

principles of medical sensors and data analysis, gaining a deeper understanding of these devices and benefiting from their coursework.

User 2: Lab Teaching Assistant

The Lab TAs will be running the labs for the biomedical engineering students. They will have to have a good understanding of the equipment and software used so that they can introduce the lab to their students, and help them if they have any questions or issues.

The Lab TAs will need to be able to give instructions regarding the equipment, help students with the lab, understand and do the labs themselves, grade lab reports as necessary, and troubleshoot the system if anything goes wrong.

The Lab TAs benefit from a better ability to improve their teaching and communication skills, and being able to better help students learn the labs. The Lab TAs will benefit from having an accessible lab kit as compared to having to deal with proprietary equipment and software. Proprietary equipment and software is hard to debug and the Lab TAs will not be able to make any updates based on student needs or commonly seen issues.

User 3: Course Professor

The course professor is responsible for overseeing and administering the course and making sure the students are meeting learning objectives for the course and major. The professors have a limited budget on course funding, and may not be able to spend a lot of money on expensive equipment.

They need to make sure the labs and activities align with course objectives, ensure the hardware and software are reliable and intuitive to use, and be able to easily interpret and grade student progress in the lab with the assistance of the Lab TAs. The professors need to make sure students are acquainted with common bioinstruments, so they can enter the industry and continue more advanced biomedical engineering coursework.

The professors will benefit from CyVital because it is a hardware and software platform which allows for a simple way to conduct labs and introduce students to the course learning objectives. They can focus on teaching the curriculum with a way to show students the inner workings of the system. The professors can modify the CyVital based on their own course materials and needs, without worrying about students injuring themselves in the lab. The CyVital will remove the most dangerous risks of a bioinstrumentation lab such as exposed power circuitry, while still allowing students to use the sensors to their fullest potential. They will also benefit because the CyVital system is developed in-house, and the cost is much cheaper than market solutions. This will allow the professors to save money and invest in other course material.

2. Requirements, Constraints, And Standards

2.1. REQUIREMENTS & CONSTRAINTS

Functional:

1. The GUI acquires and displays live reaction, EMG, ECG, pulse oximeter, blood pressure, and respiratory data
2. Users should be able to play and pause the data acquisition
3. Users should be able to interact with the collected data on the graph
4. Users should be able to export sections of the collected data to an Excel file

Performance:

5. Graph latency is not slower than the human eye can perceive
6. Export of 5 minutes of data should take no longer than 5 seconds
7. Switching between sensors should take no longer than 3 seconds
8. Data acquisition will not utilize more than 50% CPU usage

Usability:

9. Set-up can be easily completed within 5 minutes
10. GUI can be deployed from a single downloadable executable on Windows devices
11. Sensor selection will be modularized into tabs
12. Hardware will have an accessible open top to allow students to interact with it
13. GUI will contain student-readable labels for errors

Physical/Electrical:

14. Size is no larger than 7 cm x 29 cm x 25 cm – the size of the standard BIOPAC MP36 Four Channel Data Acquisition System [7]
15. Maximum voltage is 12V to prevent electrical-related injuries
16. PCB has a durable plastic casing to protect it from environment
17. The PCB case will be electrically isolated from the desktop and must have a surface resistivity no greater than 900 Ohms

Economic:

18. Hardware costs per unit will not exceed \$200
19. Software will cost \$0 through the use of open source libraries

Reliability

20. Code will be thoroughly tested, with >80% code coverage

2.2. ENGINEERING STANDARDS

The design will adhere to the following standards. A brief description of each standard is provided as well as its relevance to the CyVital project with suggested design implementations to meet the standard.

- IEC 62304:2006(en) - Medical device software - Software life cycle processes

- This International Electrotechnical Commission standard describes how to maintain medical device software. The standard defines software life cycle processes that ensure proper usage of the medical device continues throughout software updates and rewrites. This standard will be implemented during the maintenance of existing code bases to maintain reliability and performance of existing medical sensors.
- IEEE 11073-10700 - IEEE Standard – Health Informatics – Device Interoperability Part 10700: Point-of-Care Medical Device Communication – Standard for Base Requirements for Participants in a Service-Oriented Device Connectivity (SDC) System
 - This IEEE standard defines the communication interface protocol that will demonstrate safety and effectiveness in the system. This defines participant key purposes (PKPs) as a set of requirements that must be followed in order to validate and verify the usability and safety of each system function. This standard will be implemented throughout the code integration with hardware to ensure that each medical device is supporting safe interoperability in the classroom, which represents a mock point-of-care environment.
- IEEE 11073-10406 - IEEE Standard – Health Informatics – Device Interoperability Part 10406: Personal Health Device Device Communication – Device Specialization - Basic Electrocardiography (ECG)
 - This IEEE standard describes the communication functionality that must exist for ECG devices to managers such as personal computers. The standards list various extensions of ECG data and its display on an ECG device manager. The standard also includes various profiles based on different ECG models and necessary information that must be conveyed to the manager. This standard will be implemented as the CyVital display will show a heart rate profile and numerical data based on real-time sampling. The different profile and data types adapted will be selected from the IEEE standard in accordance with ECG sensor specifications.
- IEEE 1012 - IEEE Standard for System, Software, and Hardware Verification and Validation
 - This IEEE standard describes ways to validate hardware, software, and system aspects of a project. Before validation and verification, a tasklist is provided for various hardware, software, and system components. This standard will be implemented during each step of the testing process with both hardware and software individually, and then as a combined integration step. The CyVital should be able to meet all of the applicable criteria or answer all of the relevant questions provided in the IEEE 1012 standard.

3 Project Plan

3.1 PROJECT MANAGEMENT/TRACKING PROCEDURES

The team will use a combination of waterfall and agile frameworks. The project has uncertain sensor and wiring risks (hardware) and evolving lab UX needs (software). Agile enables us to iterate quickly on GUI, data models, and analysis, while small waterfall sections minimize risk to hardware. Some steps of the waterfall will be production, wiring, board testing, and finally software integrated

tests. Each sprint ends with a demo-able increment (such as ECG live with R-peaks) that aligns with the team's semester goals and stakeholder reviews with Dr. Lu.

The team will track software progress on GitLab. Hardware progress will be tracked through KiCAD and GitHub's version control features. Communication is conducted through iMessage and Slack, while work reports are shared on Google Drive.

3.2 TASK DECOMPOSITION

The CyVital project is split into four phases: design, development, testing, and iterating. The first three phases will occur in the fall semester, with the spring semester containing phase four. The first three phases must be completed in the fall semester in order to have a working product to utilize in BME3500 - Bioinstrumentation, which is taken during the spring semester. The below tasks will be completed relatively sequentially, but hardware and software tasks will need to be completed simultaneously.

- 1) Phase One - Design
 - a) Git Repository must be created with existing code from prior semesters
 - b) Software plan developed to improve upon existing GUI and add further functionality to existing code
 - c) The existing hardware must be re-assembled
 - d) Create a Bill of Materials (BOM) to document the necessary sensors, components, and other materials that must be ordered
 - e) Request access to the biomedical lab - Coover 1045 for materials and tooling access
 - f) Test the previous senior design teams version
- 2) Phase Two - Development
 - a) Software: Redesign GUI to be more usable
 - b) Software: Reaction data analysis and acquisition
 - c) Software: EMG data analysis and acquisition
 - d) Software: ECG data analysis and acquisition
 - e) Software: Pulse Ox data analysis and acquisition
 - f) Software: Respiratory data analysis and acquisition
 - g) Software: Blood pressure data analysis and acquisition
 - h) Hardware: Test each sensor including those in the BIOPAC to ensure components are working properly
 - i) Hardware: Design a schematic of the circuit to be transferred to KiCAD
 - j) Hardware: Create PCB in KiCAD with all components layed out.
 - k) Hardware: Order PCBs and additional components
 - l) Hardware: Design Housing for the PCB for optimal protection and usability
 - m) Hardware: Assemble PCB (attach sensors and solder components)
- 3) Phase Three - Testing
 - a) Software tested to make sure that it gets real-time data and analyses
 - b) Hardware tested to ensure PCB design is functioning properly
- 4) Phase Four - Iteration
 - a) Design documentation
 - b) Collect feedback from biomedical students
 - c) Adjust design based on student and TA feedback

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

The three milestones for hardware correspond to the different phases for the fall semester as stated in Section 3.2. Progress for completing software milestones will take place during all three phases of the fall semester, but the milestones do not necessarily coordinate to a specific phase as the software is being developed under an Agile framework. The software milestones will be completed relatively sequentially, with necessary changes implemented as determined by the Agile framework. The software and hardware must satisfy the following metrics or include the following features:

- Software Milestone 1: Tool Setup
 - We will acquire the git repository from the previous senior design group
 - Tests will be conducted on the previous version to determine its quality
 - A plan will be developed to convert their software into a better design
- Software Milestone 2: Live GUI Dashboard
 - The dashboard should render ECG/EMG/SpO streams (mock data) smoothly with zoom, filter, start/stop, and save controls.
 - The calculated analyses will be accurate to the 0.1 (tenth) of a measurement
 - The data acquisition software will show live data being gathered at 1 sample per 0.1 seconds
 - The data acquisition will refresh at greater than 30 FPS
 - The dashboard should have latency under than 100 ms
 - The data acquisition will aim to use 40% CPU usage or less, with no crashes
 - The TA will be able to navigate around the GUI in under two minutes to various features
- Software Milestone 3: Record, Replay, Export
 - The GUI will record raw signals, replay them deterministically, and export them as a CSV
 - The CSV will match the specified file as determined in the GUI
 - The GUI must have a timestamp drift under 10 ms/min
 - Dashboard will have replay vs. live correlation greater than 0.98
 - Exports will succeed on long datasets
- Hardware Milestone 1: Document Existing Set-up
 - A detailed drawing and schematic of the existing set-up will be created
 - Upon accidental disassembly of the existing breadboard set-up, the drawing package should provide users with the ability to re-assemble within three minutes
 - A Bill of Materials that establishes manufacturer, purchasing location, and purchasing price will be created
 - An existing inventory will be included in the Bill of Materials
- Hardware Milestone 2: Development of PCB
 - Each sensor will undergo component testing and validation, being within 5% of the BIOPAC sensor values
 - The PCB case will be electrically isolated from the desktop and must have a surface resistivity greater than 900 Ohms
 - Each node on the PCB should measure at or under 5V
 - There will be 10 PCBs assembled with no missing components
- Hardware Milestone 3: PCB Testing
 - The PCB must be able to be connected in under five minutes

- There are no loose connections as verified by the GUI dashboard
- PCB signals are detected in Waveforms software (the default software for the oscilloscope)

3.4 PROJECT TIMELINE/SCHEDULE

GANTT CHART

CyVital

Meng Lu

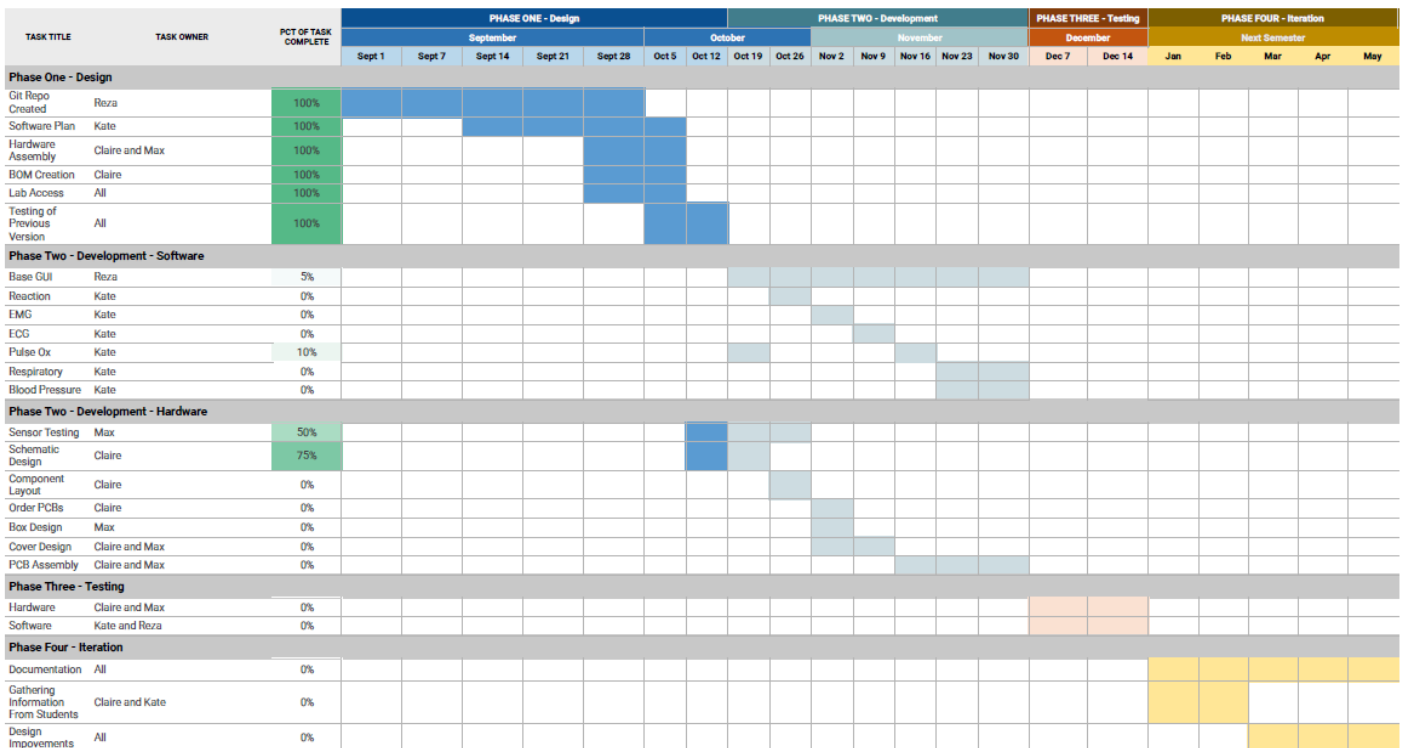


Figure 1: Semester Gantt Chart

GANTT CHART

CyVital

Meng Lu

TASK TITLE	TASK OWNER	PCT OF TASK COMPLETE	PHASE A					PHASE B & PHASE C							
			Feb 6	Feb 13	Feb 20	Feb 27	Mar 6	Mar 13	Mar 20	Mar 27	Apr 3	Apr 10	Apr 17	Apr 24	May 1
Phase A - Finalize															
Finalize Schematic	Claire	100%													
Order PCB	Team	50%													
Fabricate PCB	Claire and Max	0%													
Design and print PCB Housing	Max	0%													
Code Improvements	Kate and Reza	0%													
Phase B - Testing															
Software Testing	Kate	25%													
Hardware Testing	Claire	0%													
Update Software Based on Testing Results	Kate and Reza	0%													
Update PCB, if necessary	Claire and Max	0%													
User Testing	All	0%													
Phase C - Documentation															
Develop Lab Manuals	All	0%													
Final Kit Assembly	All	0%													

Figure 2: Semester 2 Gantt Chart

3.5 RISKS AND RISK MANAGEMENT/MITIGATION

During the fall semester, the team identified the following risks

- Risk 1: Missing physical respiratory sensor or the blood pressure sensor
 - Description: Unable to develop software or complete hardware that accommodates these sensors.
 - Estimated Probability: 50%
 - Risk Mitigation Plan: These will be part of later labs in the semester, so we will be able to move developing software for these until the beginning of next semester. The BME students could also use the already developed BIOPAC if software and hardware is not completed
- Risk 2: Real Time GUI Latency
 - Description: Challenge to get hardware and software to connect seamlessly with little latency. Additional risk factor of possible frame drops between the sensor output and the graph displays.
 - Estimated Probability: 30%
 - Risk Mitigation Plan: Background threading and ring buffer, efficient coding - careful looping, and learning from experts in data display software.
- Risk 3: Signal Processing Algorithms
 - Description: Noisy labs will make it hard to target a 95% signal accuracy.
 - Estimated Probability: 30%
 - Risk Mitigation Plan: Coding will implement acceptance tests with open datasets, allow filters, and the peak pick feature.

The first risk came to pass, as we didn't end up receiving the hardware for the respiratory sensor before the end of the first semester. Therefore, we were unable to develop the correct software to

implement this sensor since we did not know what values it gives off. Similarly, we were unable to get the final PCB itself before the end of the semester. Thus, we had to fall back on our mitigation plan, and had the BME students use the BIOPAC in class during the spring semester rather than our design.

We also had a little trouble with real-time GUI latency, as the longer the code ran, the slower the graphing got. To fix this, the code was adjusted so that it only plotted the current window while the acquisition was running, and plotted everything when it was paused.

3.6 PERSONNEL EFFORT REQUIREMENTS

An estimated personnel effort table has been created in Table 1. The second table documents the actual personnel effort that occurred.

Task	Hours
Milestone 1A - Tool Set-up <ul style="list-style-type: none"> - Git Repository - Software Plan - Lab Access - Testing of Previous Version 	25
Milestone 2A - Live GUI Dashboard <ul style="list-style-type: none"> - Base GUI - Reaction DAQ - EMG DAQ - ECG DAQ - Pulse Ox DAQ 	40
Milestone 3A - Recording, replaying, reporting <ul style="list-style-type: none"> - Base GUI - Reaction DAQ - EMG DAQ - ECG DAQ - Pulse Ox DAQ - Software Testing 	30
Milestone 1B - Document Existing Setup <ul style="list-style-type: none"> - Hardware assembly - BOM Creation - Lab Access - Testing of previous version 	30
Milestone 2B - PCB Development <ul style="list-style-type: none"> - Sensor Testing - Schematic Design - Component Layout - Order PCBs - PCB Box Design - PCB Cover Design 	50

Task	Hours
- PCB Assembly	
Milestone 3B - PCB Testing and Integration - Testing via Waveforms and assembly-time	15

Table 1: Man-Hour Estimates

Task	Hours
Milestone 1A - Buffer Tool Set-up	25
Milestone 2A - Live GUI Dashboard	60
Milestone 3A - Recording, replaying, reporting	30
Milestone 1B - Document Existing Set-up	30
Milestone 2B - PCB Development	70
Milestone 3B - PCB Testing & Integration`	40

Table 2: Actual Personnel Effort

There was a mismatch between our estimated man-hours and actual personnel effort for Milestone 2A because of conflicting scope creep causing duplicate work.

Notably, the PCB also took much longer than we originally anticipated. The design took longer to finalize than we expected. Then, our design to make the PCB components very small resulted in it taking a large amount of time to assemble and test.

3.7 OTHER RESOURCE REQUIREMENTS

The team has identified the following resources for the Cy-Vital:

Hardware:

Category	Manu. Product Number	Manufacturer	Description	Qty
Capacitor	Co402C102J1GEC7411	KEMET	1 nF, 100 V, CoG, 0402 MLCC	10

Category	Manu. Product Number	Manufacturer	Description	Qty
Capacitor	GRM155R61J334KE01D	Murata	0.33 μ F, 63 V, X5R, 0402 MLCC	10
Capacitor	Co402C103K5RECAUTO	KEMET	10 nF, 50 V, X7R, 0402 MLCC	10
Capacitor	Co402C104K4RACTU	KEMET	0.1 μ F, 16 V, X7R, 0402 MLCC	10
Capacitor	UCDoJ470MCL6GS	Nichicon	47 μ F, 6.3 V, Aluminum Electrolytic SMD	10
Capacitor	CL05A106MP5NUNC	Samsung	10 μ F, 10 V, X5R, 0402 MLCC	10
Capacitor	Co402C105K9PACTU	KEMET	1 μ F, 6.3 V, X5R, 0402 MLCC	10
Capacitor	UWT1E100MCL1GB	Nichicon	10 μ F, 25 V, Aluminum Electrolytic SMD	10
Capacitor	UZG1V100MCL1GB	Nichicon	10 μ F, 35 V, Aluminum Electrolytic SMD	10
Capacitor	UWToJ220MCL1GB	Nichicon	22 μ F, 6.3 V, Aluminum Electrolytic SMD	10
Capacitor	CL05C820JB5NNNC	Samsung	82 pF, 50 V, CoG, 0402 MLCC	10
Capacitor	Co402C223K5REC7411	KEMET	22 nF, 50 V, X7R, 0402 MLCC	10

Category	Manu. Product Number	Manufacturer	Description	Qty
Capacitor	CL05A226MQ5N6J8	Samsung	22 μ F, 6.3 V, X5R, 0402 MLCC	10
LED	SML-M13UTT86	Rohm	Red LED, Clear Lens, 2012 SMD	20
LED	SML-M13YTT86	Rohm	Yellow LED, Clear Lens, 2012 SMD	10
Inductor	B82462G4103M000	TDK	10 μ H, 1.5 A, 62 m Ω , SMD Inductor	10
Inductor	1255AY-4R7M=P3	Murata	4.7 μ H, 4 A, 23 m Ω , SMD Inductor	10
Resistor	RCS04024K70FKED	Vishay	4.7 k Ω , 1%, 0402 SMD Resistor	10
Resistor	RCS0402270RFKED	Vishay	270 Ω , 1%, 0402 SMD Resistor	10
Resistor	RCS04020000ZoED	Vishay	0 Ω Jumper, 0402 SMD	10
Resistor	RCS040210MoFKED	Vishay	10 M Ω , 1%, 0402 SMD Resistor	10
Resistor	CRCW04021K00FKEDHP	Vishay	1 k Ω , 1%, 0402 SMD Resistor	10
Resistor	RCS0402180KFKED	Vishay	180 k Ω , 1%, 0402 SMD Resistor	10

Category	Manu. Product Number	Manufacturer	Description	Qty
Resistor	CRCW0402360KJNEDHP	Vishay	360 k Ω , 5%, 0402 SMD Resistor	10
Resistor	RCA0402100KFKEDHP	Vishay	100 k Ω , 1%, 0402 SMD Resistor	10
Resistor	CRCW04021M00FKEDHP	Vishay	1 M Ω , 1%, 0402 SMD Resistor	10
Resistor	CRCW040210KoFKEDHP	Vishay	10 k Ω , 1%, 0402 SMD Resistor	10
Resistor	RCS040280K6FKED	Vishay	80.6 k Ω , 1%, 0402 SMD Resistor	10
Resistor	RCS0402150KFKED	Vishay	150 k Ω , 1%, 0402 SMD Resistor	10
Switch	1119	Adafruit	12 mm Tactile Pushbutton Switch	1
Diode	1N4448W	Diotec	75 V, 150 mA Standard Diode, SOD-123F	10
Diode	PMEG3020BER-QX	Nexperia	30 V, 2 A Schottky Diode, SOD-123W	2
Diode	MBR120VLSFT1G	onsemi	20 V, 1 A Schottky Diode, SOD-123FL	2

Category	Manu. Product Number	Manufacturer	Description	Qty
IC	TS30042-M033QFNR	Semtech	3.3 V, 2 A Buck Regulator, 16-QFN	1
IC	TS30042-M050QFNR	Semtech	5 V, 2 A Buck Regulator, 16-QFN	1
IC	AD8232ACPZ-R7	Analog Devices	ECG Front-End Amplifier, 20-LFCSP	1
IC	MAX30101EFD+T	Analog Devices/Maxim	Optical Heart Rate/Oximeter Sensor, I ² C	1
IC	MAX32664GTGA+T	Analog Devices/Maxim	Cortex-M4F MCU, 96 MHz, TQFN-24	1
IC	TPS6735IDR	Texas Instruments	-5 V, 200 mA Buck Regulator, SOIC-8	1
IC	AD8221ARMZ-R7	Analog Devices	Instrumentation Amplifier, MSOP-8	1
IC	TLo84IDR	Texas Instruments	Quad JFET Op-Amp, SOIC-14	1
IC	AMS1117-1.8	EVVO	1.8 V, 1 A Linear Regulator, SOT-223	1
Trimmer	3361P-1-103GLF	Bourns	10 k Ω Trimmer Potentiometer, Top Adjust	1

Category	Manu. Product Number	Manufacturer	Description	Qty
Crystal	CFS-20632768DZBB	Citizen	32.768 kHz Crystal, 6 pF Load, THT	1
Connector	PPTCo61LFBN-RC	Sullins	6-Pin Header, 2.54 mm Pitch, THT	1
Connector	ADC-002-1	Adam Tech	DC Power Jack, Right-Angle, Black	1
Connector	2311765-1	TE Connectivity	9-Pin D-Sub Receptacle, Right-Angle	1
Connector	54-00178	Tensility	3.5 mm Stereo Jack, Right-Angle SMD	3
Solder	NC191LTA10	Chip Quik	Low-Temperature Solder Paste	1

Table 3: Bill of Materials

Software Tools:

- Visual Studio Code
- Python
- Tkinter
- AutoCAD
- KiCAD

Facility:

- Lab Access: Coover 1045 for hardware assembly, soldering and testing
- Guidance and feedback from Dr. Lu and TA Jacob Eisenbrenner
- Lab Access: Senior Design Lab to use the reflow oven for assembling SMT components
- ISU Makerspace or Engineering Design Center for cutting acrylic and or 3d Printing

All tools except hardware are either free to use for the public or free to use as a student. Hardware will be acquired using the Cy-Vital funds relating to BME 3500 course development.

4 Design

4.1 DESIGN CONTEXT

4.1.1 Broader Context

Area	Description	Examples
Public health, safety, and welfare	The students will be better biomedical engineers, which will make them better healthcare professionals.	CyVital allows for an interactive approach to sensor learning. CyVital can be shared with other schools in order to better the healthcare training students have.
Global, cultural, and social	This project is designed to improve the skills of prospective biomedical students before they begin work in industry, allowing them to have better skills to apply to their degree program.	CyVital will teach students how to use the sensors and the various engineering motivators. CyVital will teach students how to look for trends or healthcare observations across different ethnicities and genders, as health markers can sometimes vary due to these factors.
Environmental	The project uses rare metals due to it being an electronic project. However, components without tariffs and that did not need to be imported were selected in order to relieve the cost of shipping.	The PCB uses Copper, Silicon, Geranium, and other elements in the procurement process. The case will be plastic, but the goal is to choose a primarily recyclable plastic that is still durable and fits the CyVital.
Economic	It will cost less than the commercial products, allowing it to be more affordable.	It will cost less than \$200 whereas BIOPAC can be up to \$10,000. CyVital could be expanded to other universities, making quality course material accessible.

Table 4: Broader Context

4.1.2 Prior Work/Solutions

CyVital Phase I

The CyVital Phase I system allowed students to read data from the sensors, and displayed it in a gui. However, the hardware was implemented on a breadboard that meant it would be hard for BME students to set up. Their GUI also implemented imgui, which made for a clunky gui that was difficult to use. In addition, there is no functionality to interact with the received data.

[1] BIOPAC Systems Inc., “BIOPAC Student Lab,” [Online]. Available: <https://www.biopac.com/product/bsl-pro/>.

The BIOPAC is the existing product on the market that supports physiological sensor data acquisition and analysis. However, it can cost \$10,000, and the hardware functionality is obstructed inside of a blue box.

[2] OpenBCI, “OpenBCI: Biosensing for Everyone,” [Online]. Available: <https://www.openbci.com/>.

OpenBCI is a notable open-source alternative that provides low-cost biometric amplifiers to measure brain, muscle, and heart activity. It is more affordable than the BIOPAC, but less focused on education.

[3] H. da Silva et al., “BITalino: A novel hardware framework for physiological computing,” Proc. ACM Int. Joint Conf. Pervasive and Ubiquitous Computing Adjunct Publication, 2014, pp. 479–482. DOI: 10.1145/2638728.2641693.

The BITalino platform is another low-cost biosignals platform that is for educational purposes. It provides basic GUI and sensors, but you have to buy everything separately.

4.1.3 Technical Complexity

This project requires both circuit theory and full-stack development engineering. The full-stack development engineering requires developing a user interface, a back-end host for data collection display, and OS deployment. The hardware engineering requires six different sensors. It requires knowledge of analog to digital conversion, signal integrity, and functional testing. On both the software and hardware end, signal processing is required. Low-pass, band-pass, and high-pass filters will be used on the hardware side to eliminate major noise. The software developed will allow for more complex filters such as Chebyshev and Butterworth to be implemented. This project draws upon both electrical engineering and software engineering knowledge in order to create a user-friendly system.

4.2 DESIGN EXPLORATION

4.2.1 Design Decisions

The first key decision the team made was in a printing circuit board (PCB) for teaching. The proposed idea was to have a bread board model with documentation for students to follow. However, the time constraints and electrical background of the students made this less ideal. The PCB board will condense the amount of time students need to learn how to use the bioinstrumentation sensors. The board will be compact and will have all the same sensors as the breadboard iteration from Cy-Vital Phase I as described in Section 4.1.2.

The second key decision that the team made was to use surface mount components, rather than pre-assembled Sparkfun boards. While Sparkfun boards are simple and preassembled, the economic investment for a single sensor like the EMG is nearly five times more expensive than it would be to implement the EMG using purely surface mount components. Using surface mount components requires analog to digital converters, multiplexer lines, and voltage protection circuitry to create a low-noise and functional PCB.

The third decision the team made was to choose a case for the PCB. Knowing that the PCB would be in a classroom environment, there are chances for accidents. Certain accidents like spills and drops should not damage the PCB in any major way. In this case, the team researched common

housings for PCBs that focus on durability while still being cost-efficient. Ultimately, the PCB housing will be made from polylactic acid (PLA) due to its high durability, ease of manufacturing using a 3D-printer, and protection from moisture.

4.2.2 Ideation

The team identified potential PCB housing materials by using past knowledge and researching material properties. The five options that were considered were:

- Polylactic Acid (PLA). This is a common plastic used in 3D printers and is known for its print accuracy and limited warping. PLA is cost effective and easy to use.
- Polyethylene Terephthalate Glycol (PETG). This is another common plastic used in 3D printers. It has high heat resistance but can sometimes be brittle.
- Aluminum. This is a metal option that offers very high resistance to chemicals and stresses. Aluminum can induce additional electromagnetic fields that may affect the circuit.
- Fiberglass. This is a lightweight and fire-resistant material. It is also non-conductive and non-magnetic while providing a high strength-to-weight ratio.
- Wood. This material is easy to work with and is often cheaper. When treated, it can be waterproof and may look more aesthetically pleasing than other materials.

4.2.3 Decision-Making and Trade-Off

The team identified several categories that the housing material must fit. This includes: easy manufacturing, water-resistant, economical, durability, and electrically safe. The team considered wood. While wood can be easily manufactured, it might not be as durable as other materials and could potentially injure students by giving them splinters. Although fiberglass meets a lot of the material properties, it is not economical to get in a desired shape or easily manufacture. Lastly, the team was down to two commonly used 3D-printer plastics. PLA was ultimately chosen due to its price, ease of manufacturing, and accuracy of printing as compared to PETG, since mounting holes and PCB clearances have to be exact. After considering all of the options, the process of elimination allowed the team to choose the best material for the current scope of the project.

4.3 FINAL DESIGN

4.3.1 Overview

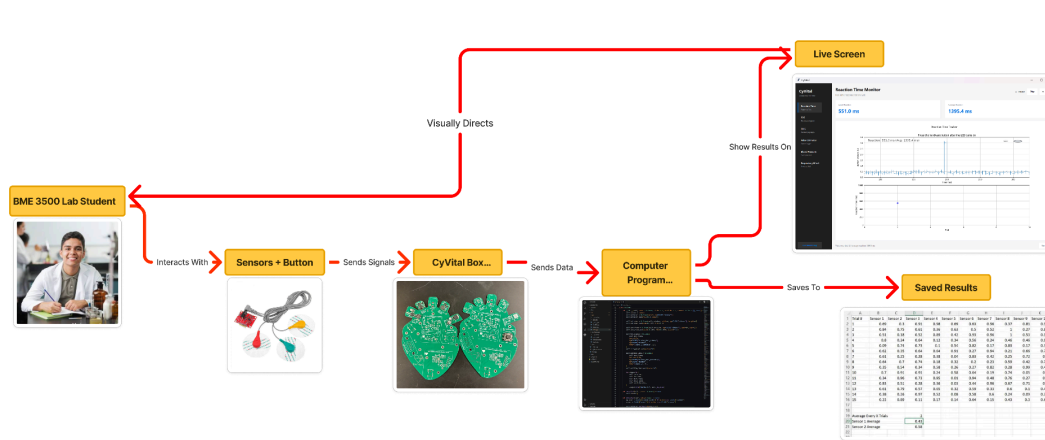


Figure 3: High-Level Design

Students connect the biomedical sensors to the test subjects. These sensors transmit the physiological data to a custom PCB, which routes the signals through a Digilent Analog Discovery 3. This data is then transferred to a computer via a USB Type-C. Once received, the software processes the raw signals to display real-time waveforms, allowing users to select and export specific data sections to Excel for further analysis.

4.3.2 Detailed Design and Visual(s)

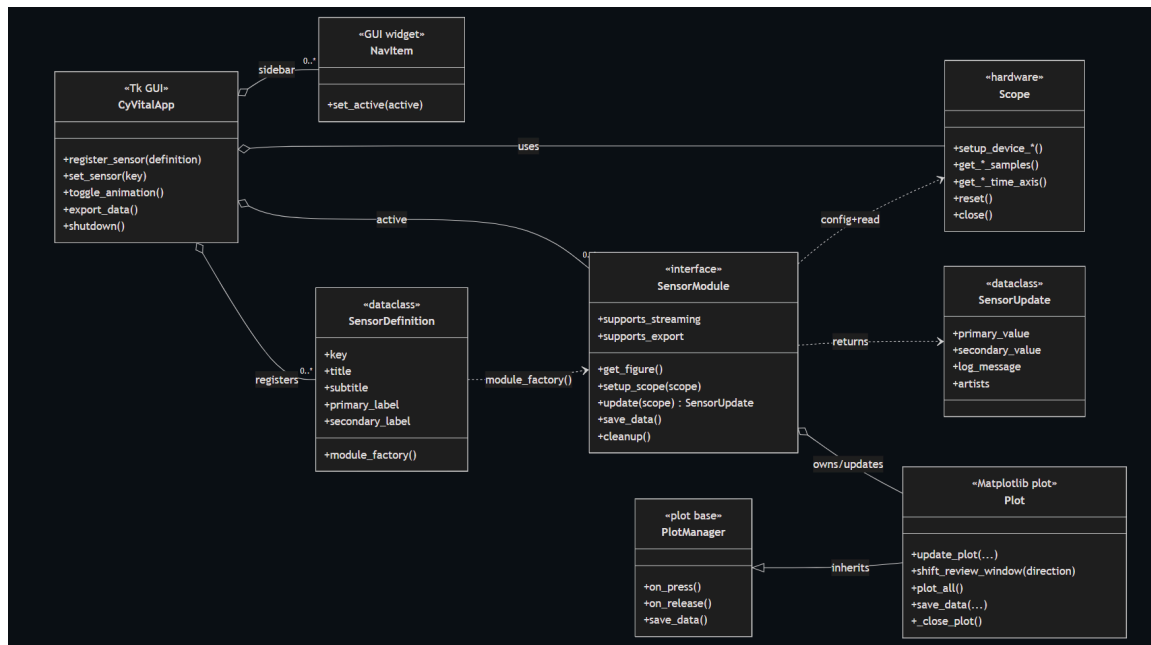


Figure 4: High-Level Software Diagram

The CyVital application is organized around a Tkinter GUI that manages layout, navigation, and the active sensor workflow. Available sensors are described by `SensorDefinition` objects that the app registers and uses to instantiate a concrete `SensorModule` when the user selects a sensor. Each `SensorModule` acts as the integration layer: it configures and reads from the hardware abstraction (`Scope`), updates a corresponding Matplotlib-backed `Plot`, and returns a normalized `SensorUpdate` containing UI-ready metric strings, log/status text, and the updated Matplotlib artists for efficient redraw. Plotting code is structured with `Plot` classes that inherit shared interaction/export behavior from `PlotManager`, and each sensor module owns its plot instance to keep plotting logic isolated from the GUI while still enabling features like shifting the review window, exporting data, and cleanup during shutdown. Overall, this separation allows adding a new sensor by introducing a new `Plot` and `SensorModule`, then registering a new `SensorDefinition`, without changing the core GUI or hardware interface.

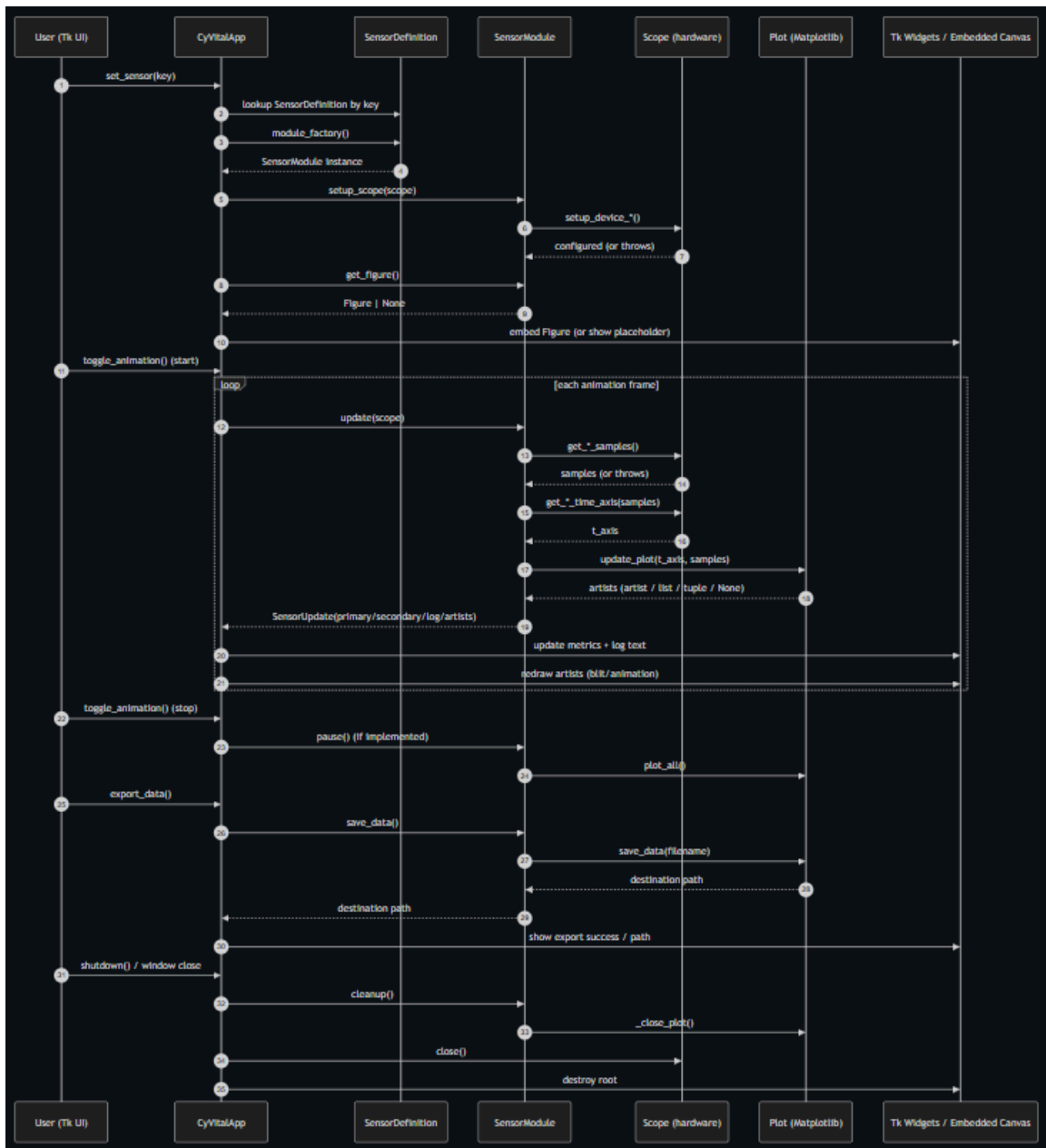


Figure 5: Sequence Diagram

When the user selects a sensor, `CyVitalApp.set_sensor(key)` looks up the matching `SensorDefinition`, uses its `module_factory()` to instantiate a `SensorModule`, calls `setup_scope(scope)` to configure the hardware via `Scope.setup_device_*`, and then embeds the module's Matplotlib figure into the UI. When the user starts streaming (`toggle_animation()`), the app enters a per-frame loop where it calls `SensorModule.update(scope)`; the module reads fresh samples from the `Scope` (`get_*_samples()`), computes or requests the time axis (`get_*_time_axis()`), updates the plot via `Plot.update_plot(...)`, and returns a `SensorUpdate` containing formatted metric strings, a log message, and the updated Matplotlib artists. The app then applies that update to the Tk widgets and triggers a redraw of the returned artists for efficient animation. When streaming is paused, the app stops the frame loop and typically asks the module/plot to render a full-history view. Export is

initiated by the user through `export_data()`, which calls `SensorModule.save_data()` and delegates to the plot layer to write an Excel file, returning the destination path for the GUI to display. Finally, on shutdown or window close, the app calls the module's `cleanup()` to close figures and release resources, closes the Scope device connection, and destroys the Tk root window.

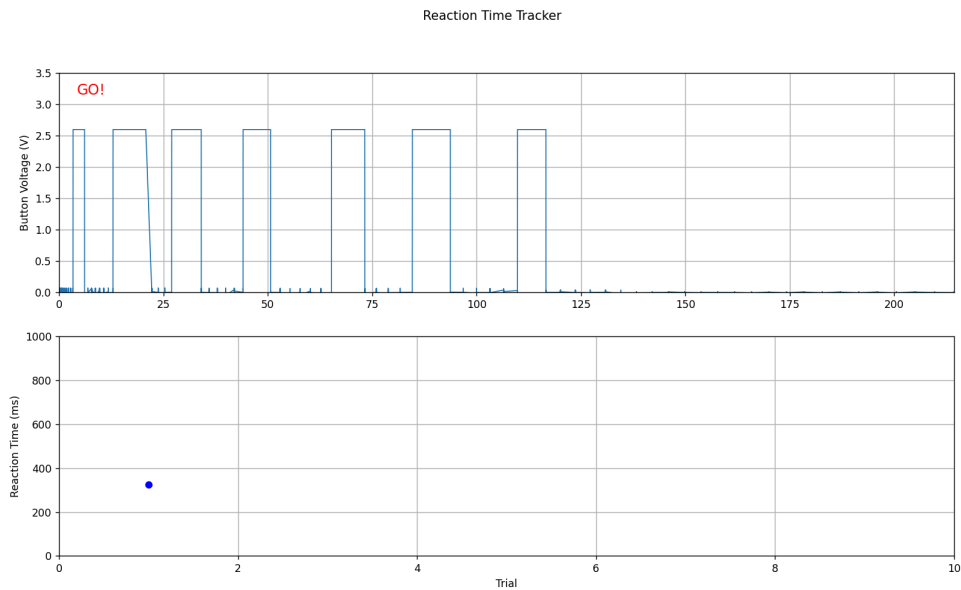


Figure 6: Sample Plot

For the plotting functions, there will be two plots displayed on the screen. The top plot will be the raw signal coming from the sensors – digital for the pulse oximeter and analog for the rest of the sensors. The bottom plot will display the processed data. Each plot will be interactable using `mpl_interactions` that allow the user to zoom, pan, and select sections of the data.

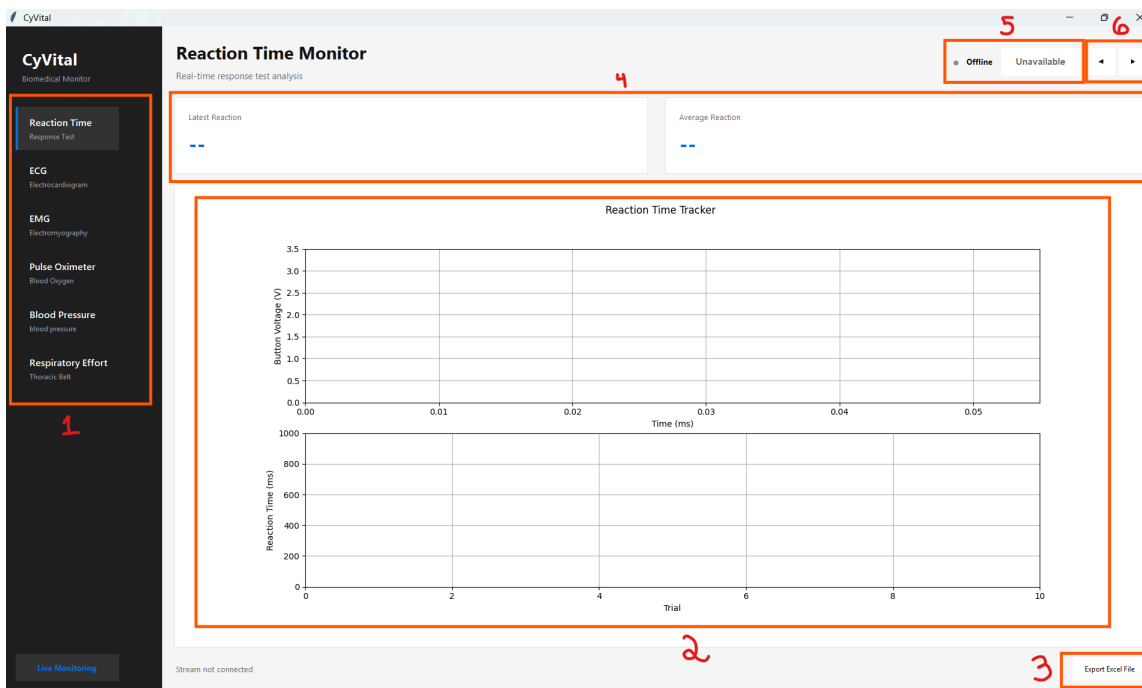


Figure 7: UI Layout

The plot will be integrated into a Tkinter GUI (2). The left panel (1) will display tabs to select between the various sensors. There will be a button in the lower right corner (3) that allows the user to export the graph to an excel file. Above the graph will be sections displaying current calculated measurements of various vital signs (4). In the upper right, it will have functionality to start/stop the data acquisition (5) if the scope is available. The user will also be able to move the graph around, with both matplotlib interactions on the graph itself, as well as with buttons in the upper right (6).

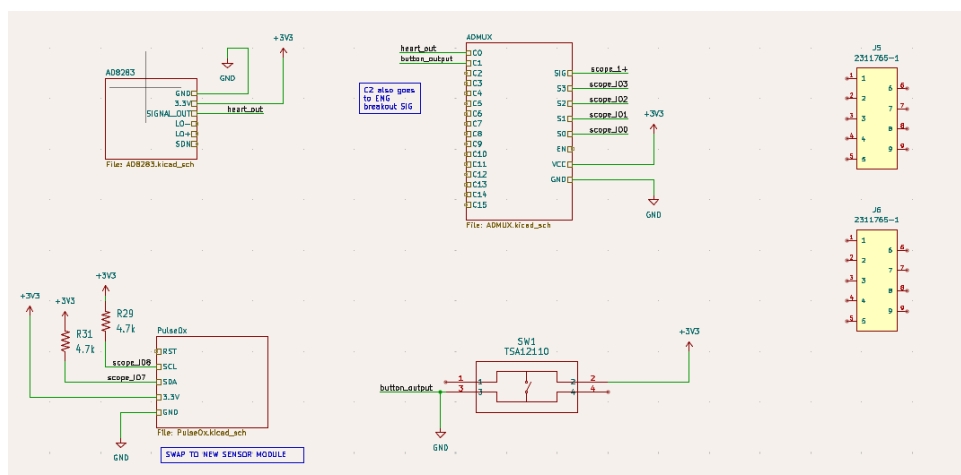


Figure 8: A Sparkfun Iteration of the PCB Board

The PCB was designed in KiCAD, a free and open source software, which aligns with the team's commitment to financial responsibility. A Sparkfun iteration of the board is shown in Figure 6. The electrical schematic shows how the sensors will be connected logically and abstracts some of the voltage protections and filters that are present in the surface mount version.

Next, the team worked on creating the surface mount PCB. The detailed design includes power management circuits, the EMG circuit, the Pulse Oxygen and Heart Rate Sensor, the Reaction Time Sensor, and the Electrocardiogram Amplifier Circuit. Going from the Sparkfun version to the PCB version required the team to implement updated signal filtering, voltage and current protections, and visual labels that will enhance student's learning.

The circuit board takes in a standard 12V supply through a barrel connector. The 12V supply then is broken into busses of 5V, -5V, 3.3V, and 1.8V based on the integrated circuit requirements.

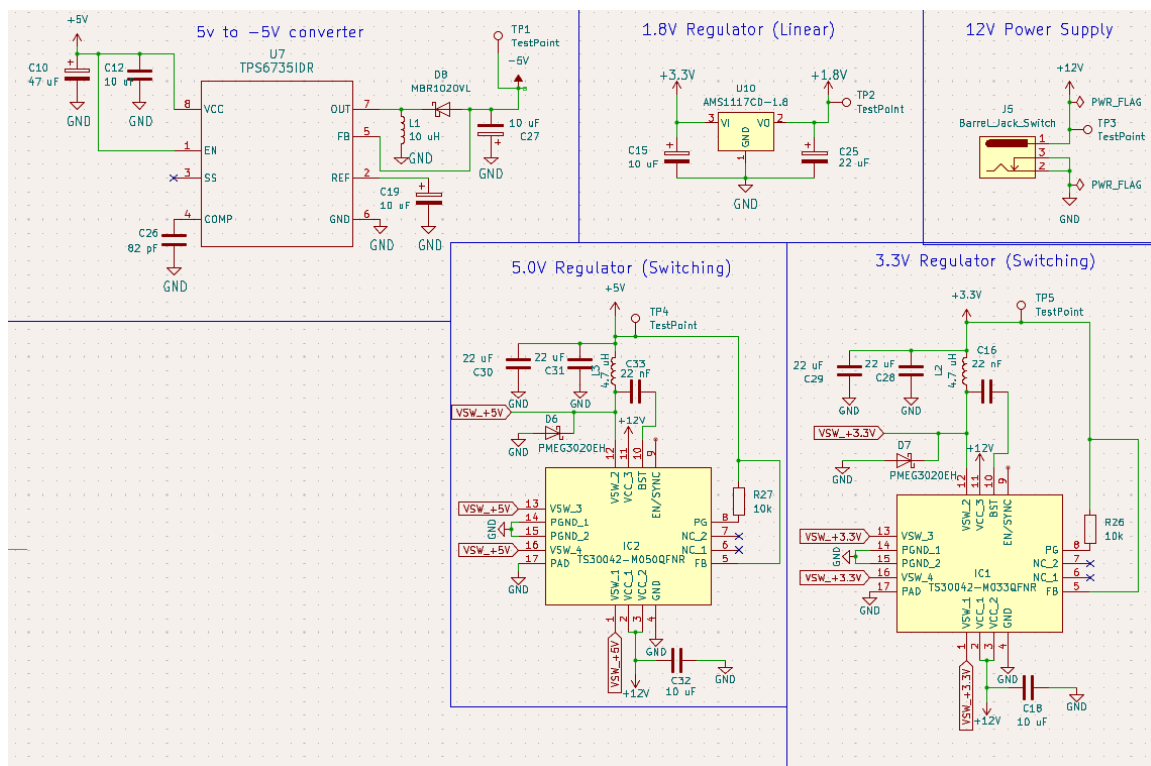


Figure 9: Converts & Regulators Schematic

Figure 10 shows the schematic for the electrocardiogram. There is an input jack for the electrode pad connector that takes activity from the heart and feeds it into the IC chip. The integrated circuit is designed to have an increased signal to noise ratio and outputs an analog voltage signal that is shown on the software through a standard 2.54mm connector.

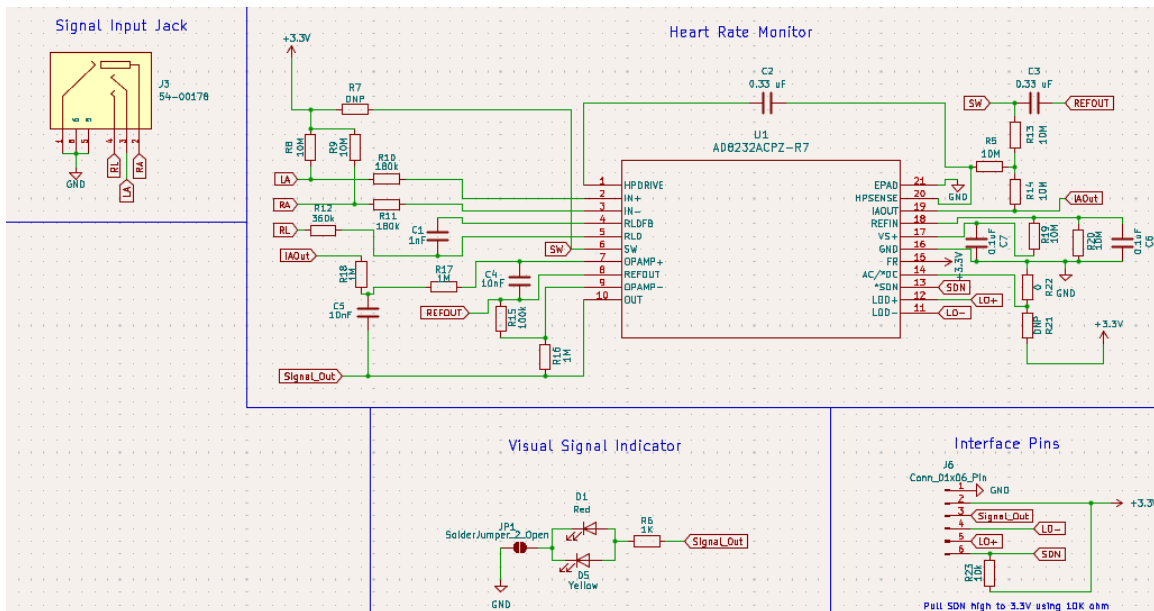


Figure 10: EKG Schematic

In Figure 11, the EMG schematic is shown where a differential and instrumentation operational amplifier has been designed with variable gain. The electrode pad connector feeds in signal into u8 which is a precision operational amplifier. This signal is then rectified and smoothed. The custom PCB allows for two places where the software can take readings, one after the instrumentation op-amp and another after the differential op-amp. The options for this allows the software team to do further debugging and implement other signal processing features if needed.

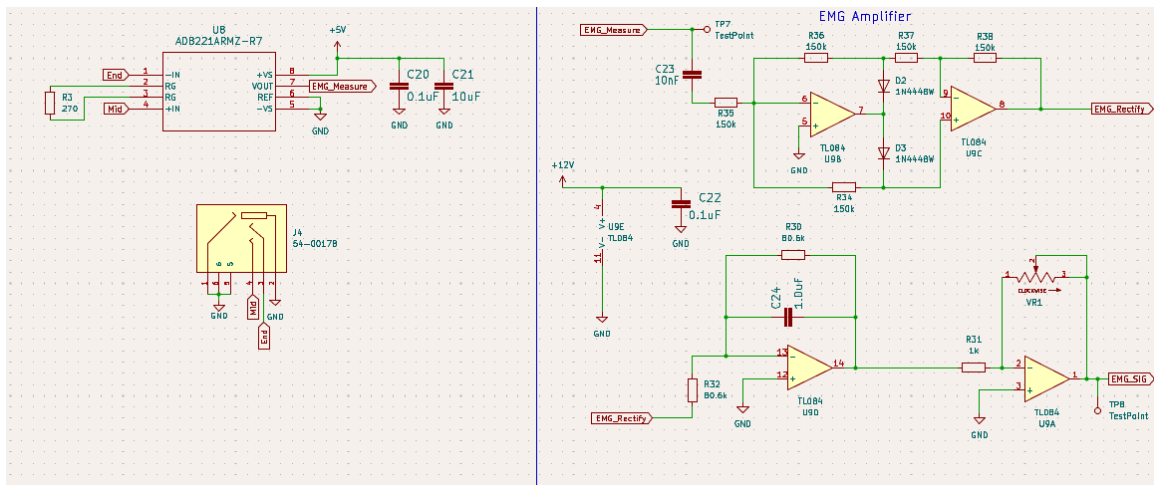


Figure 11: EMG Schematic

In Figure 12, the pulse oximeter and biometric hub is shown. The pulse oximeter and heart rate sensor data is fed into the biometric hub. Utilization of the biometric hub allows for the I2C communication protocol to be used with a 32.768kHz real time clock. The software uses these I2C lines to read and process the data.

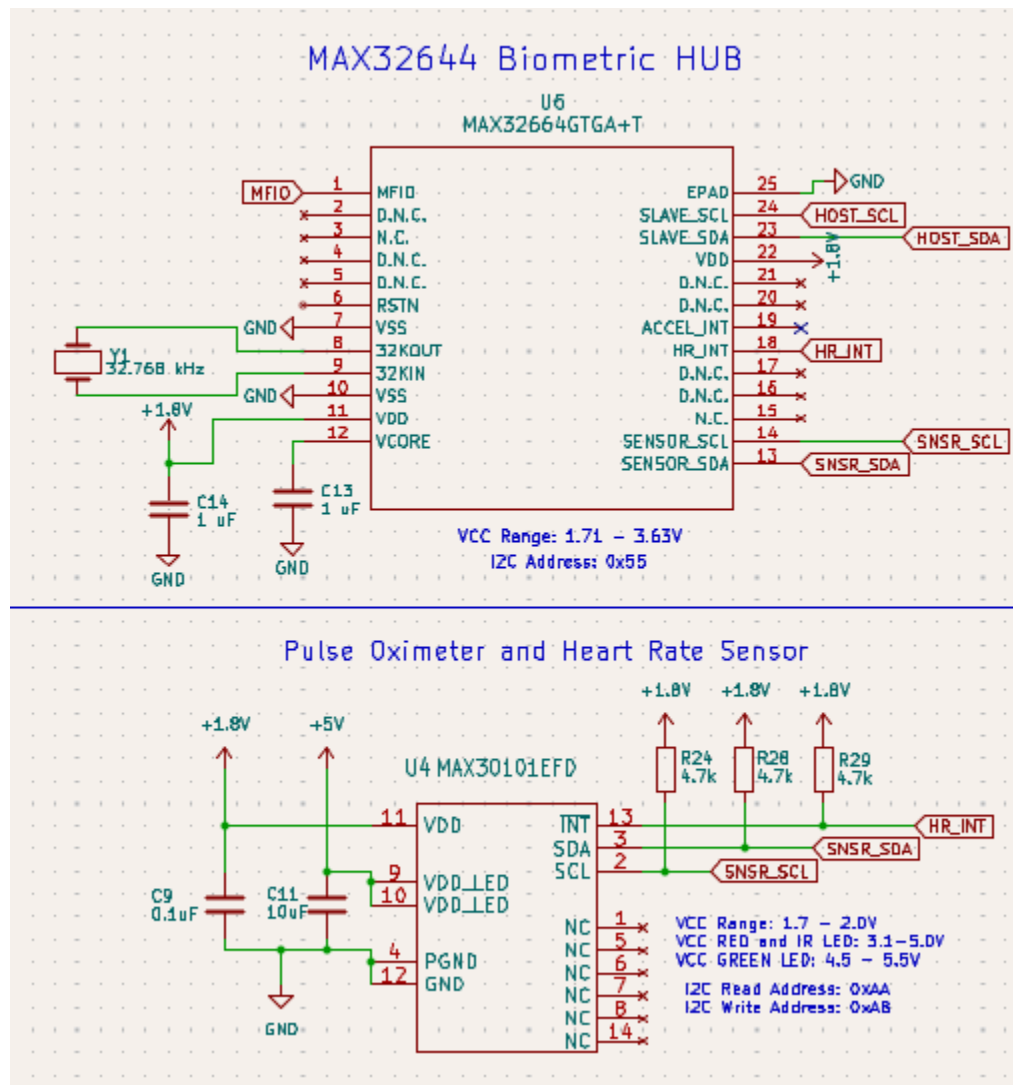


Figure 12: Biometric Hub & Pulse Oximeter Schematics

Figure 13 shows the connection to existing BIOPAC sensors and the reaction time sensor. The reaction time sensor uses a standard push button switch that reads 3.3V when pressed. Existing BIOPAC sensors currently use D-sub connectors to output an analog reading or I2C readings. The PCB allows for the integration of old systems (BIOPAC) with our own software by connecting the signals from the D-sub to a 2.54mm connector that works with the Digilent Analog Discovery.

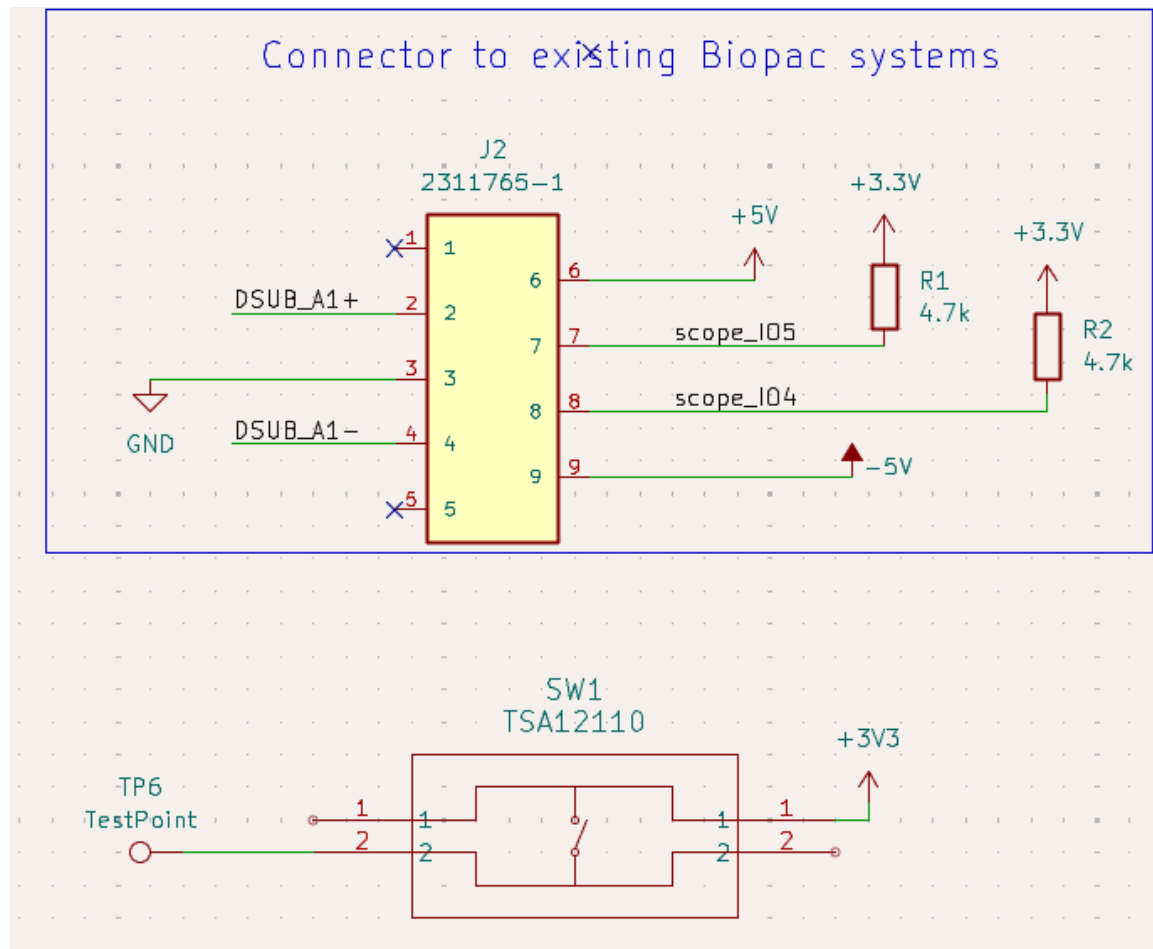


Figure 13: Reaction Time Schematic

Figure 14 shows the PCB layout as a 3D rendering. It is based on an anatomical heart to be more engaging and fit with the theme of medicine. The power supplies are tucked into the arteries, with a spot for the 12V connector. Data can be read using the nearby connectors which are compatible with the Digilent Analog Discovery oscilloscope.

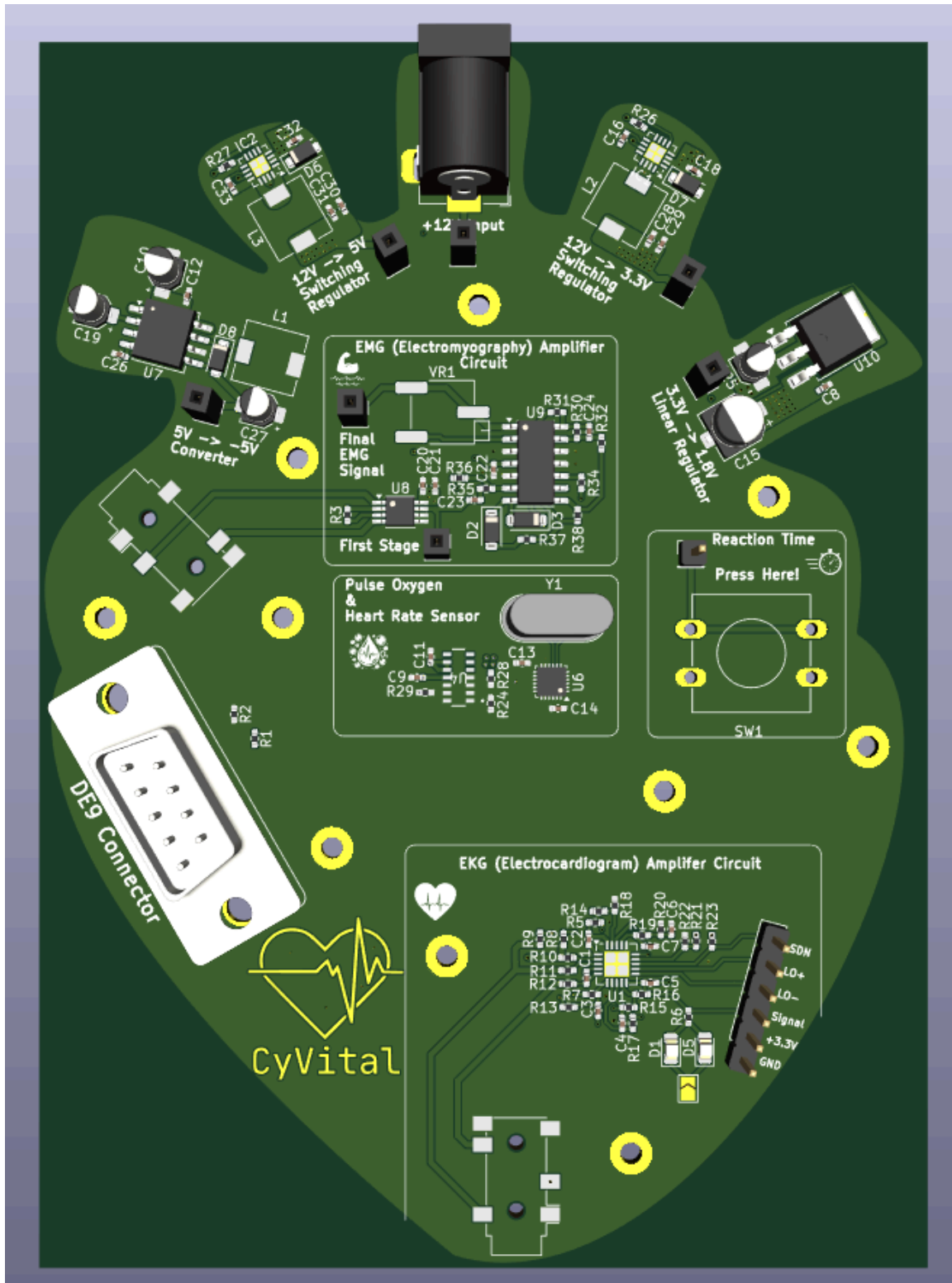


Figure 14: 3D Rendering of PCB

4.3.3 Functionality

Students will first connect the sensors to the volunteer patient. Students will use the detailed silkscreen and other lab materials to determine which sensor is which. They may observe other components on the PCB that are not part of the sensor signals but are necessary for the board to operate. This board teaches the students that there are numerous inner-systems when it comes to a medical module. Each inner-system might provide power, noise reduction, or test indicators that all contribute to a smooth device. After properly hooking up the volunteer patient and ensuring that the Analog Discovery Oscilloscope USB is plugged into the lab computers, the students will be able to use the GUI.

Students will be able to select what sensor to start reading data from. The program will read this information through the oscilloscope to display in a graph to the user. Once it is displayed on the screen, users will be able to zoom, pan, and select sections of the data to be exported to an excel file for later analysis.

4.3.4 Areas of Challenge

Because the BME 3500 class we are designing this system for was supposed to be taken in Spring 2026, we were time constrained to get a product working by the end of the first semester. However, we didn't end up succeeding. This means that the system might not be as complex as would be ideal. Thus, the team is focusing on making the product simple, basic, and easy-to-use.

As the system is developed in-house, it is up to the BME3500 TA or other people that provide course support to fix any issues that might arise when used in the lab environment. The team hopes that the system will be able to be kept up for use in later years.

4.4 TECHNOLOGY CONSIDERATIONS

PCB (Surface Mount)

- *Strengths:* It is cheaper than using pre-made Sparkfun boards. It also will be easier and safer for students to use than a breadboard with exposed circuitry. The PCB will be able to be smaller than if using the pre-made Sparkfun boards, as well as integrate new sensors such as a blood pressure cuff. The PCB will be more compact and could have less noise than Sparkfun boards due to a more electrically-efficient design.
- *Weaknesses:* The assembly for the PCB will take longer. Due to cost constraints, each component must be manually placed. In addition, there is a higher chance for error than with Sparkfun boards due to manufacturing errors.
- *Trade-Offs:* The team decided that it is better to save money and custom design the PCB. This will give the team more control over the types of filters implemented and components chosen to avoid supply-chain issues. In addition, the team will not be limited by potential firmware issues.

Python

- *Strengths:* It is good for creating simple GUIs and has easy integration with libraries like matplotlib. There is an established library – dwfpy – to read the waveforms from the scope. The previous group’s code already used python.
- *Weaknesses:* A weakness is that one sensor’s setup requires us to use C code to set the mode. It can also be slower than Java or C++.
- *Trade-Offs:* The team figured that prioritizing simplicity and ease-of-use for students was more important than the weaknesses.

Tkinter

- *Strengths:* It integrates well with Matplotlib, which is being used to display the graphs of the signals. It can run across operating systems. It is a lightweight system that is stable.
- *Weaknesses:* It isn’t very complex, with limited customization and features, and lacks interesting styling or design.
- *Trade-Offs:* The team decided to prioritize functionality over styling to make sure the program would be stable and easy-to-use.

5 Testing

5.1 UNIT TESTING

The units that need to be tested are the six different sensors, both their physical hardware as well as their software representation. Physically, each sensor hardware has to be tested to make sure that it is accurate using an oscilloscope or multimeter. The team will compare that data with what is given by the original BIOPAC sensors. Then, the software has to test each sensor separately to make sure the signal processing and data display works. Before adding to the main program, each unit test was written as a separate function before being merged to the main branch.

5.2 INTERFACE TESTING

The first interface relates hardware to software data transfer processes. The Scope class is the interface between the oscilloscope and the plots. This interface by making sure the data is displayed in the graphs when hooked up to the physical scope as well as testing with an abstracted scope.

The software interface connects between the user interaction of the GUI, the raw input gathered from the scope, and the plotting. The PlotManager class is the main interface between the user input and making stuff happen on the graphs. Each functionality of the GUI (buttons, scrolls, etc.) will be tested by all members to make sure they perform the correct functions and account for any differences in operating system and personal computers.

5.3 INTEGRATION TESTING

The team identified the following paths and tools to test the relationships between each of the pieces in the CyVital project:

Sensor hardware → scope → data analysis → GUI

The sensors send the data to the oscilloscope, the code reads the data from that oscilloscope, performs data analysis on it, and then displays those values to the user in the GUI. This relates to the functional requirements of sensor data displaying in real time as stated in .

User interaction → PlotManager → Sensor plot display

This integration path specifies that the user interaction on the GUI (buttons, clicks, scroll etc.) affect the plot graphs. An important requirement was our UI requirements, that included having the UI functionality operate.

5.4 SYSTEM TESTING

Each sensor module will be considered separately. For each module, the different requirements of functionality, performance, and UI/UX will be tested. To determine functionality of hardware, the team will record a reasonable amount of sample data points as specified by the manufacturer based on failures in time (FIT), mean time between failures (MTBF), and defective parts per million (DPPM). Many manufacturers provide calculators to determine the number of sample data points to test [4]. To determine performance, measurements such as rise time and fall time for signals will be recorded and compared against the benchmarks provided in sensor datasheets or in the code libraries for additional software packages. To test the UI/UX, each sensor module will be tested by all members of the team.

5.5 REGRESSION TESTING

The critical features to monitor include data acquisition from sensors, the accuracy of the data analysis, and the GUI responsiveness. Since each sensor has its own dedicated plot module, ideally adding a new sensor module should have no effect on the old functionality. However, when adding a new sensor plot module, the team will then iterate through all of the existing sensor modules, determining that the buttons are still working, the selection of data is user-friendly, and the data acquisition is accurate. The software team does this by utilizing GitLab for version control. A branch is created for each feature to be added and only merge with the main code after having tested the new feature.

After assembly of the PCB, each sensor will be probed by an oscilloscope and the Digilent Analog Discovery scope. If the sensor is determined to be a failure, the sensor can be desoldered from the board and all connections from before will remain intact. As the process of soldering is tedious, it is expected that the sensor component went through prior testing such as unit tests and interface tests beforehand.

5.6 ACCEPTANCE TESTING

The acceptance testing process is to perform a complete, live demo of the physiological sensors. The performance is evaluated by comparing it to the standard BIOPAC sensors. Latency times will be recorded and compared against the requirements as determined in Section 2. Despite the CyVital using less commercialized hardware, each component was selected for high precision without being too expensive, producing quality tools that help students learn. It is expected that there will be some deviation from the CyVital performance and the BIOPAC sensor, but each value should be within 95% of the expected value.

The team plans to document the acceptance testing, allowing the client to go back and repeat the acceptance tests and will give further confidence in the project. Each requirement will be given a test case ID. The steps to perform the acceptance test will be associated with each ID. The expected and actual outcomes will be recorded along with a file of measurements during the testing session.

5.7 SECURITY TESTING

One vulnerability in the system is with writing data to a file. It is important to determine that the export path is safe. This will be done by reviewing export handling procedures to ensure files only land in expected directories determined by the student or the default “Downloads” area. This vulnerability can be tested by writing invalid filenames or duplicate file names to see if data is overwritten.

5.8 USER TESTING

The plan for user testing involves the BME 3500 students taking the class in future semesters. The team will design surveys and discuss with the students about their experience using the product. The hope is to gather ideas on what students like or think could be improved about the product. Some user activities that test a wide range of the CyVital capabilities are listed below:

1. Download the CyVital App. In this case,

5.9 RESULTS

The display and integration of the sensor plots on CyVital Phase I from the previous senior design group has been tested. Since this is a prior hardware version, the team has focused more on

qualitative testing. This was so that we were able to focus on basic interface and integration testing to make sure the connection between scope and GUI was sound. Our tests were successful for all four sensors on the Phase I breadboard – reaction, ECG, EMG, and pulse oximeter.



Figure 15: Pulse Oximeter Live Test

5.9.1 PCB Testing

To test, the PCB power management was first assembled. This allows the team to test for any major design defects that could cause the system to fail such as current and voltage overloads. The team tested basic electrical connection (continuity) and voltage, current, and resistance using a multimeter. The measured resistance of the board is around 10 Ohms, meeting the design criteria of having the board's resistance to be under 900 Ohms.

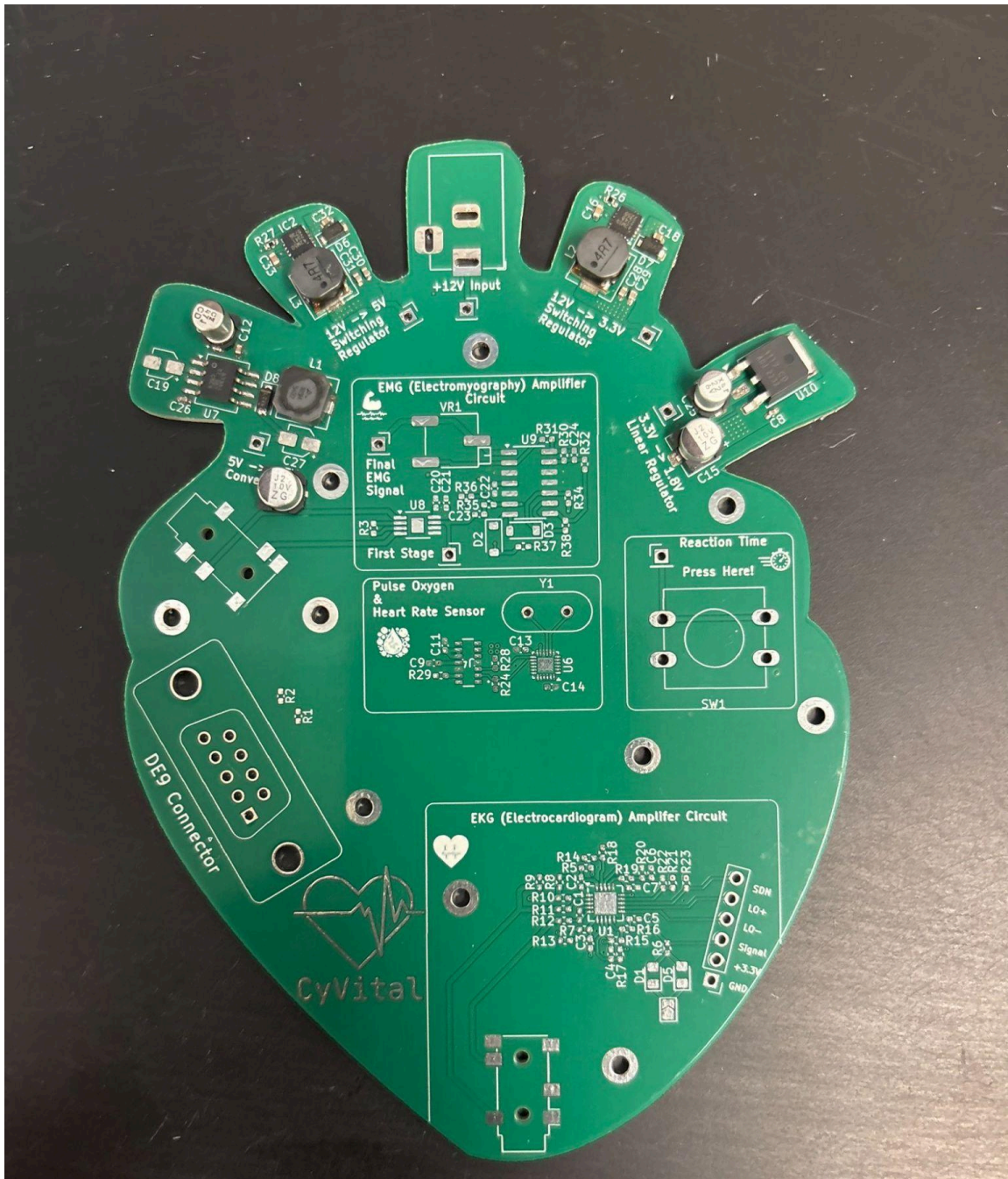


Figure 16: Power Management Assembly Step

As the Reaction Time button is a through-hole component, it could be tested simultaneously during the power management assembly process as shown in Figure 17. In this case, the button was inserted into the board and 3.3V, the button operating voltage, was delivered to the board. When testing the Reaction Time button, the voltage measured from the button is around 2.9V which indicates a slight voltage drop. This makes sense though as there are other parasitic materials that have been introduced onto the board.

To test the other components (PCB assembly shown in Figure 18), the test points were connected to a standard oscilloscope (non-Digilent Analog Discovery) and the waveforms were analyzed. In this case, there were detected signals for the EKG, EMG, and pulse oximeter.

	Desired Result	Result
Reaction Time Button	3.3V	2.95 V
PCB Surface Resistivity	<900 Ohms	10 Ohms
		Signal detected?
EKG		x
EMG		x
Pulse Oximeter		x

Table 5: PCB Results

After initial signal readings, the PCB was hooked up to the software and some sample waveforms for the reaction time sensor and the EKG are shown in Figures 19 and 20. For the EKG, additional electrode pads are needed for more accurate BPM results, but the readings show the desired interfacing.

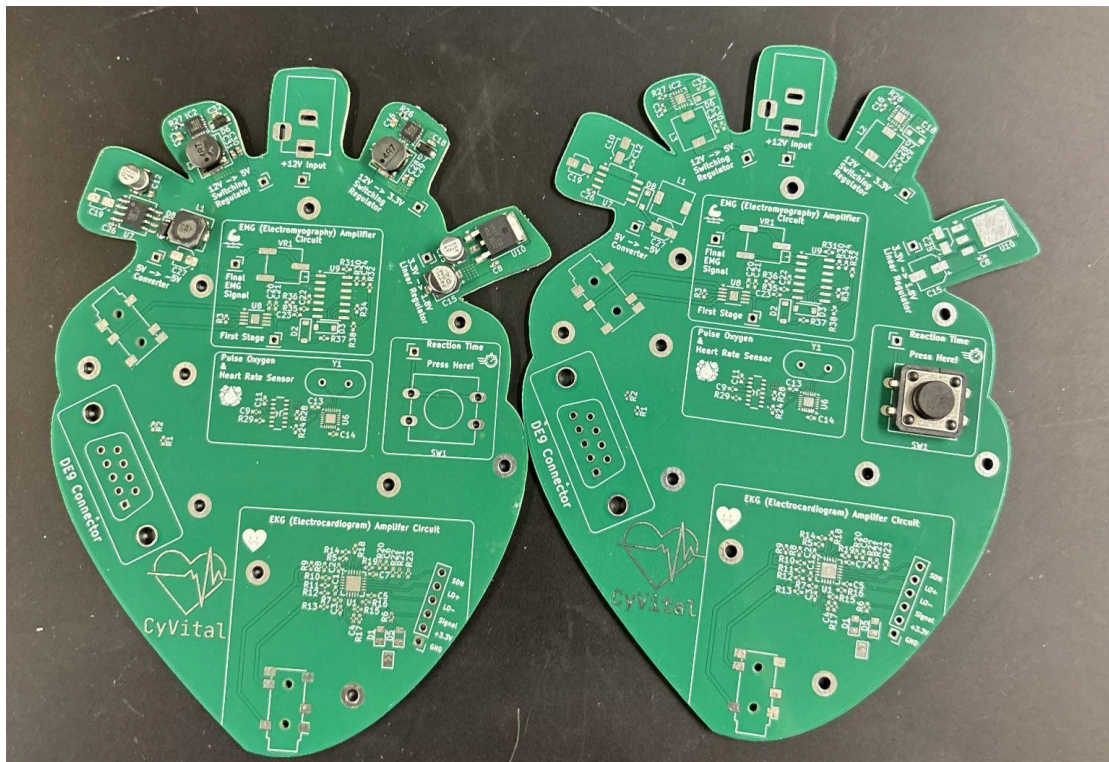


Figure 17: Power Management Assembly Step

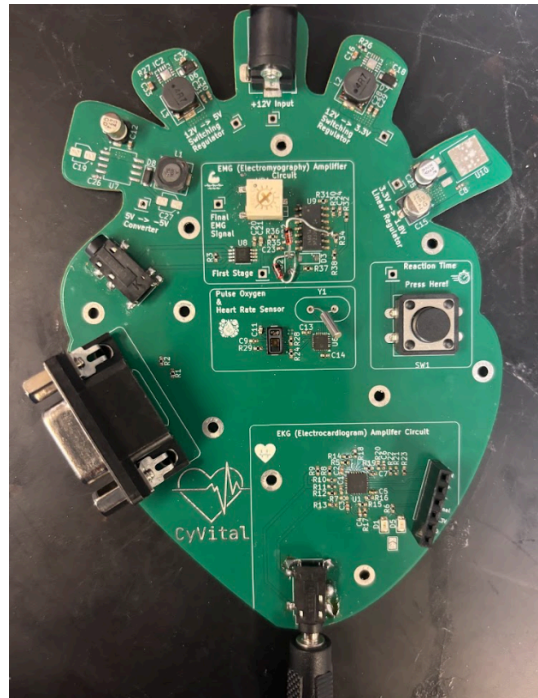


Figure 18: Final Assembly Step

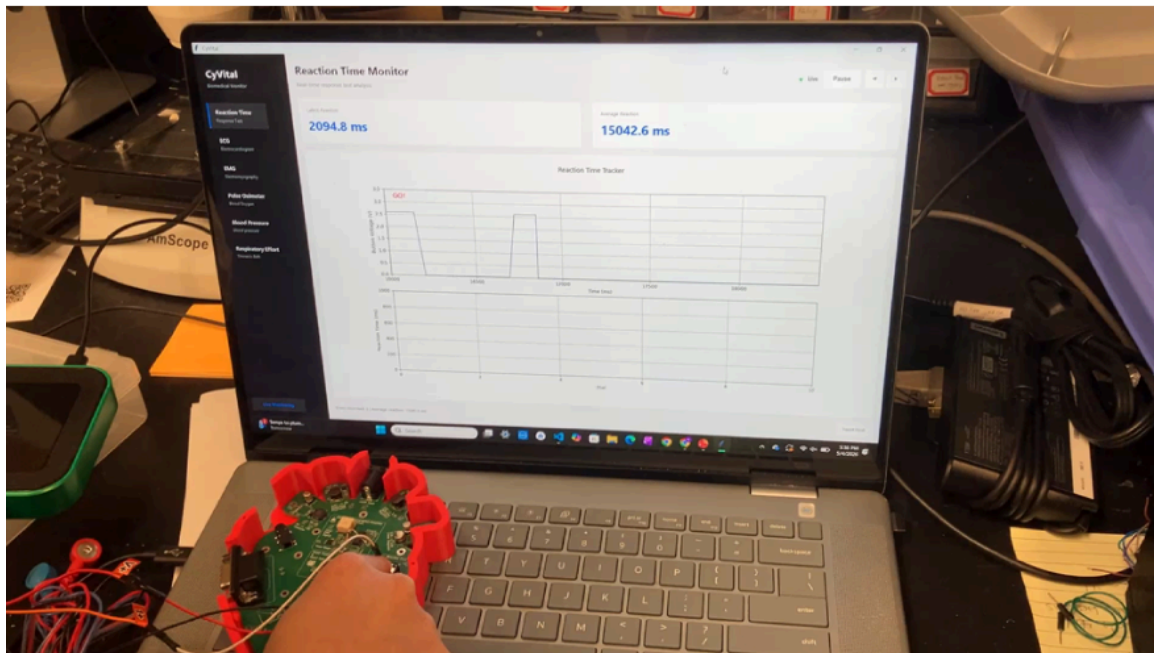


Figure 19: Reaction Time Graph

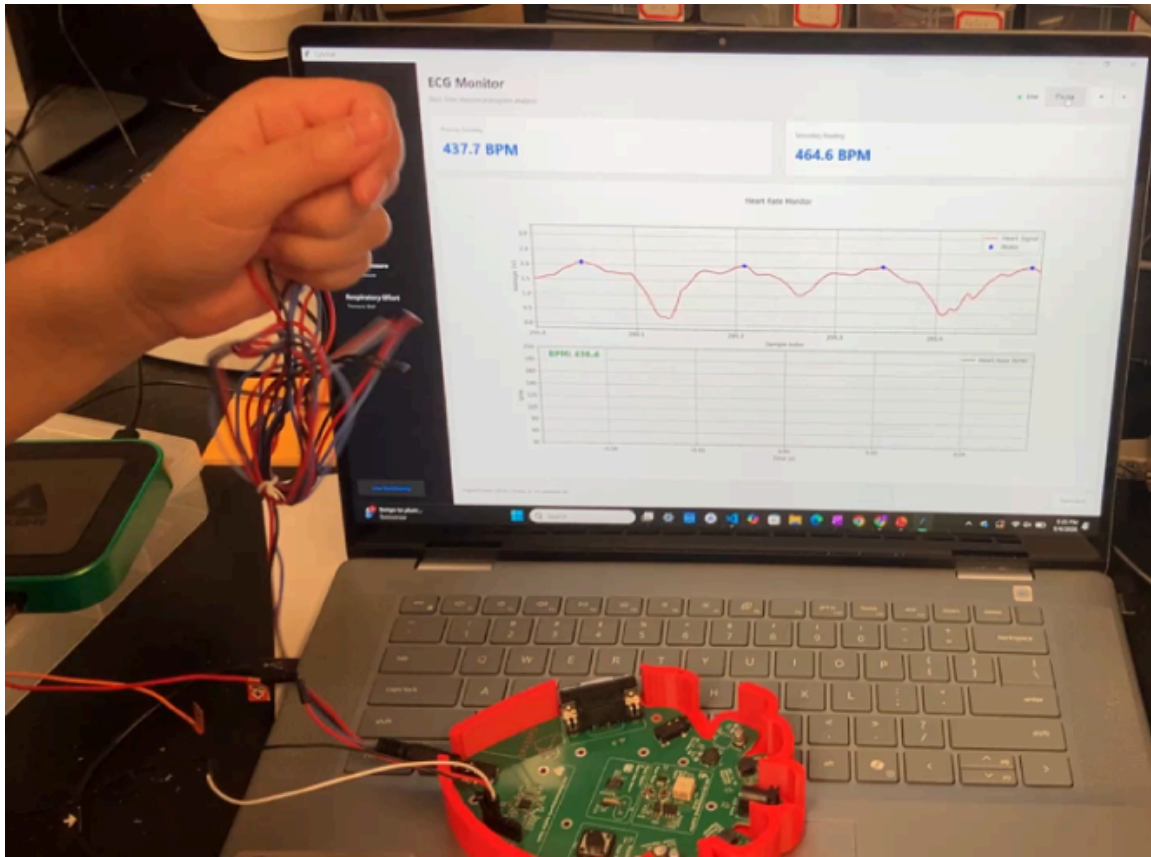


Figure 20: EKG Graph

5.9.2 Software Testing

Utilizing pytest, 491 test cases were created to test the functionality of the software in order to achieve 99% code coverage. This sufficiently satisfied our requirement of having >80% code coverage. The only statements not tested are unavoidable exclusions such as the sensor except ImportError fallbacks, thus 99% is the practical ceiling.

Name	Stmts	Miss	Cover	Missing
src\Main.py	17	0	100%	
src__init__.py	3	0	100%	
src\gui__init__.py	3	0	100%	
src\gui\models.py	17	0	100%	
src\gui\sensors__init__.py	9	0	100%	
src\gui\sensors\base.py	23	2	91%	13-14
src\gui\sensors\blood_pressure.py	45	2	96%	16-17

src\gui\sensors\ecg.py	57	2	96%	16-17
src\gui\sensors\emg.py	45	2	96%	16-17
src\gui\sensors\helpers.py	10	0	100%	
src\gui\sensors\message.py	9	0	100%	
src\gui\sensors\pulse_ox.py	53	2	96%	16-17
src\gui\sensors\reaction.py	57	2	96%	17-18
src\gui\sensors\registry.py	12	2	83%	7-8
src\gui\sensors\respiratory.py	68	2	97%	17-18
src\gui\theme.py	4	0	100%	
src\gui\tkGui.py	339	7	98%	32, 40-44, 628
src\oscilloscope\Scope.py	165	0	100%	
src\oscilloscope__init__.py	3	0	100%	
src\plots\BloodPressurePlot.py	63	0	100%	
src\plots\ECGPlot.py	152	0	100%	
src\plots\EMGPlot.py	79	0	100%	
src\plots\PlotManager.py	73	0	100%	
src\plots\PulseOxPlot.py	132	0	100%	
src\plots\ReactionPlot.py	132	0	100%	
src\plots\RespiratoryPlot.py	105	0	100%	
src\plots__init__.py	3	0	100%	
TOTAL	1676	23	99%	

Table 6: Pytest Code Coverage Report

6 Implementation

The CyVital team implemented a GUI that reads data from sensors, performs analysis, and displays that information in a graph to the user. It displays a list of buttons on the left to select between the different sensors, data analysis values in the boxes at the top, and an embedded matplotlib graph below those containing the raw and analyzed data plots. Using a cursor, it allows the user to select sections of the graphs to save to a data file for later use.



Figure 21: GUI Implementation

6.1 Design Analysis

After using the PCB with the software, there were some adjustments to be noted. The PCB lacks quality access points for the pulse oximeter sensor and the D-sub connector. While data is able to be read, it is not easy to use with the oscilloscope channels. More minor adjustments need to be made in the sizing of the aluminum capacitors and diodes to better fit supply chain availability. The final PCB assembly and case with the headers and connectors is shown in Figure 22.

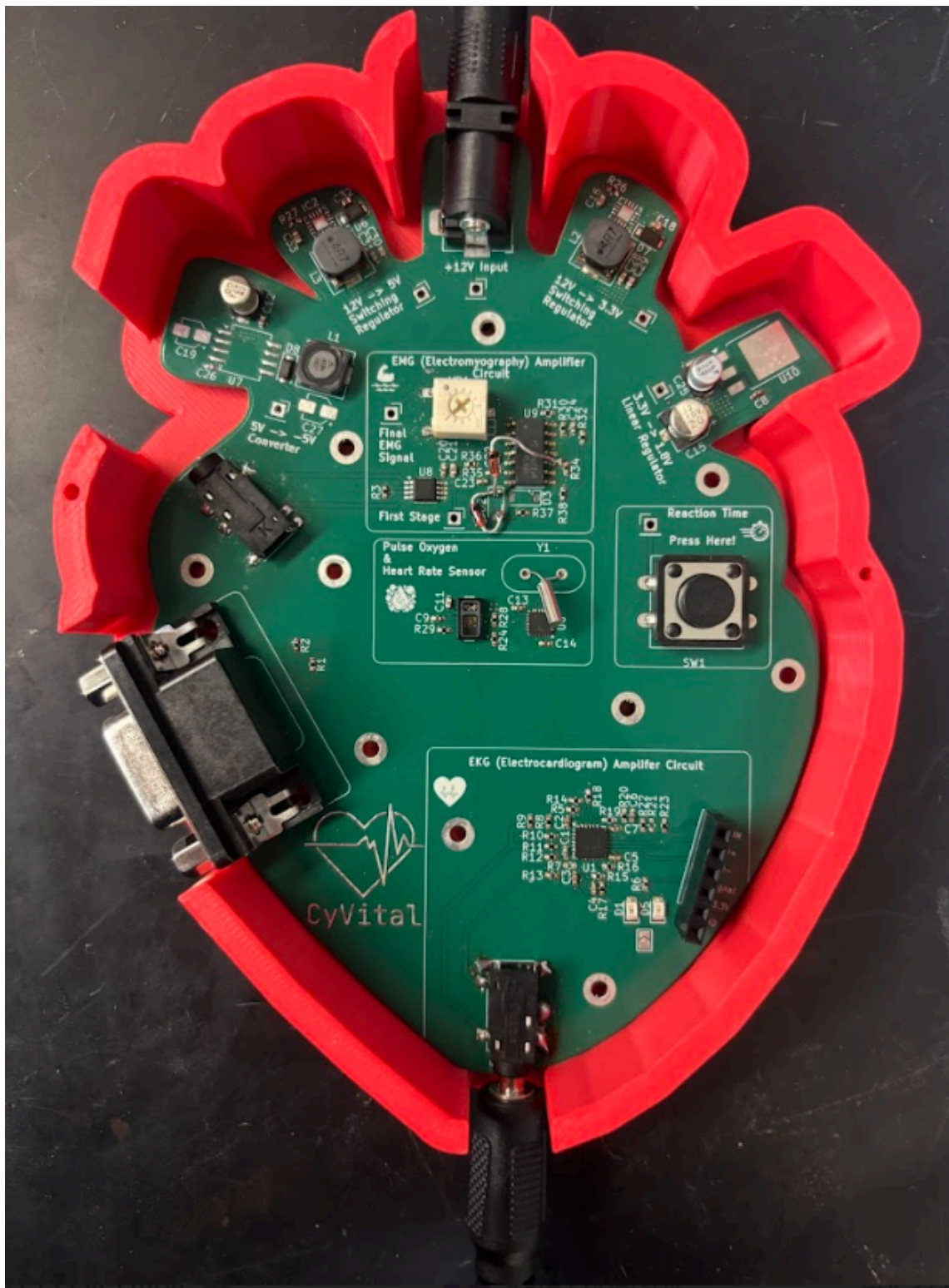


Figure 22: Final PCB Case

7 Ethics and Professional Responsibility

During the course of the second semester, we looked into a potential future business plan. In transitioning this system toward a commercial business model, we prioritized our professional responsibilities and the ethical implications of our design. Our commitment to excellence in these areas is driven by three core pillars: raising industry standards, market accessibility and equity, and educational integrity.

We believe it is important to provide an alternative to the BIOPAC – the aging, unintuitive system currently used by Iowa State Biomedical Engineering students. By developing a high-quality system, we ensure that students are no longer forced to rely on obsolete technology to learn more about these systems. In having access to a modern system, they will be better prepared for real systems they will encounter in their professional careers.

Ethical consideration also requires addressing the barrier of cost. While each BIOPAC system can cost thousands of dollars, with the addition of more money spent for additional sensors, our system is available at less than \$200 per unit, as well as free software. This drastic reduction in price ensures that this technology is accessible to a broader range of institutions.

Because this system is intended for biomedical engineering education, it is essential that it supports a high standard of learning before students begin designing real-world healthcare technologies. We have a social responsibility to ensure that the tools we provide are accurate, as well as meaningful and reflective of current industry standards so that students can develop the competence required in their future careers.

7.1 AREAS OF PROFESSIONAL RESPONSIBILITY/CODES OF ETHICS

Area of Responsibility	Definition (In Our Own Words)	Relevant IEEE Code of Ethics Item	How Our Team Has Practiced This
Work Competence	Deliver high-quality work on time	“Maintain and improve technical competence” (IEEE Code of Ethics, Item 6); [6]	We strive to work hard at making a good system.
Financial Responsibility	Deliver cost-effective solutions	“Avoid real or perceived conflicts of interest” (Item 4). [6]	We tried to keep our budget low by using cost-effective parts for the PCB. We tried to keep our budget low by continuing to use free and open-source software.
Communication Honesty	Being clear, accurate, and upfront about what works, what	IEEE #3: Be honest and realistic in	Throughout the project, we’ve kept open communication both within our group and the instructor via

Area of Responsibility	Definition (In Our Own Words)	Relevant IEEE Code of Ethics Item	How Our Team Has Practiced This
	doesn't work, and what still needs fixing.	stating claims or estimates based on available data [6]	iMessage and Slack, when problems arise we let each other know and make sure everyone is fully informed of the situation. We report all actions taken and obstacles faced in our weekly reports and meetings with Dr. Lu. This ensures everyone is on the same page and makes the project move smoother.
Health, safety, and well-being	Ensure the solution prioritizes health, safety, and well-being	"Hold paramount the safety, health, and welfare of the public" (Item 1). [6]	We are teaching future biomedical professionals, so we have to make sure that our data is accurate to give them the best education.
Property ownership	Respect information, data, property, and ideas of others	"Avoid injuring others, their property, or reputation" (Item 8). [6]	We must ensure that user data isn't compromised.
Sustainability	Protect natural resources and the environment	"Strive to comply with sustainable development practices" (Item 10). [6]	We want to minimize waste created from our product
Social responsibility	Produce solutions that benefits society	"Enhance the quality of life of humankind" (Item 1). [6]	We developed the solution to align with educational goals, ensuring usability and accessibility

Table 7: Professional Responsibility

Our team has done a solid job staying open and honest with each other about our progress. When someone is ahead, stuck, confused, or needs help, we've been pretty good about saying so instead of keeping it to ourselves. This has made it easier to divide up tasks and know what to work on next. It also helps our meetings with Dr. Lu go more smoothly because we're upfront about what we've finished and what still needs work. Overall, keeping communication open has helped us avoid a lot of misunderstandings.

One thing we could improve is being a little more consistent with updates. Occasionally, someone finishes a task or hits a small delay but doesn't mention it until later, which slows the team down a bit because we aren't always working with the most current information. Being sure to alert the team of any mishaps or delays immediately, would help keep the team on track.

In the second semester, we made efforts to improve on our collaboration consistency. We made sure to incorporate methods like utilizing GitHub issues in order to make sure that people we assigned the correct tasks to implement.

7.2 FOUR PRINCIPLES

	Beneficence	Nonmaleficence	Respect for Autonomy	Justice
Public health, safety, and welfare	Helps students learn physiology more easily by giving them a clear, easy to use tool during the lab.	Make sure the device is safe to use in a classroom and doesn't cause issues or confusion.	Lets students explore data at their own pace using features like zooming and saving.	Every lab pair gets their own device instead of the whole class sharing a few BIOPAC units, giving everyone equal hands-on time.
Global, cultural, and social	Supports a positive learning experience for students from many backgrounds.	Avoids creating a confusing or overly complex tool that could disadvantage some users.	Gives instructors flexibility in how they run labs or demonstrations.	Ensures the system is simple and approachable regardless of a student's previous experience.
Environmental	A reusable PCB with built-in components reduces waste by replacing fragile breadboards and loose parts.	Avoids building something fragile that breaks easily and needs constant replacement.	Makes it possible to repair or adjust the setup rather than having it shipped for repair.	Reusable design keeps equipment in good shape so future students have the same quality setup.
Economic	Saves the department money by creating hardware instead of buying expensive systems.	Reduces accidental costs from needing last-minute replacements as it can be repaired easily.	Students can export and use their data freely without paying for extra tools.	Keeps the learning experience fair as all students will use our product instead of having to share a few BIOPAC units.

Table 8: Four Principles

One area that stands out for our project is Public Health & Safety × Beneficence. Since the device will be used by students in a lab setting, our main goal is to make the experience easier, clearer, and more hands-on. By creating a tool that's simpler to understand than the BIOPAC system and more accessible for every lab pair, we're helping future students get more value out of the course.

One area where our project could improve is Environmental × Nonmaleficence. Even though our PCB design is reusable, the development process still creates some waste through extra prototype boards, spare parts, and materials we test along the way. Reusing parts when possible, limiting unnecessary redesigns, and being careful about how many boards we order are small steps that could help reduce the overall environmental impact.

We improved on our environmental nonmaleficence by making our PCB components really small so they would use less materials and be less wasteful.

7.3 VIRTUES

1. Communication

We actively share updates through Slack and iMessage, and we don't hesitate to ask for help or clarification. Regular check-ins during meetings help everyone stay on the same page and reduce confusion.

2. Responsibility

We divide tasks based on strengths, track assignments through GitLab and GitHub, and make sure responsibilities are clearly understood. If someone falls behind, we discuss it as a group and redistribute tasks when needed.

3. Collaboration

We combine our hardware and software efforts, review each other's work, and make joint decisions when choosing designs or approaches. Everyone has a voice in discussions, which keeps the environment supportive and productive.

From the collaboration process, each member identified a virtue to demonstrate and a virtue that they will continue to work on:

Kate Endersby

- Virtue Demonstrated: Diligence
 - It is important to me because it's important to work hard and put a lot of effort into the project. I have demonstrated this by putting in a lot of labor into making sure this project will be successful.
- Virtue to Demonstrate: Timeliness
 - It is important to me because it's best if things get done in a timely manner. To demonstrate this virtue, it would be good to get assignments done early rather than procrastinating until the last minute.

Claire Haas

- Virtue Demonstrated: Attention to detail
 - This virtue is important to me because careful design involves great attention to detail. I have demonstrated this by considering the gains and drawbacks of each design decision before making a decision.
- Virtue to Demonstrate: Optimism
 - This virtue is important to me because it keeps me focused on the task at hand and what can be done. I can demonstrate this virtue by listing more wins of the tasks at hand instead of dwelling on the negative.

Reza Choudhury

- Virtue Demonstrated: Honesty
 - It is important to be honest so that people can trust you. This was demonstrated in senior design work by sharing with the others what had been accomplished and what wasn't.
- Virtue to Demonstrate: Clear and thorough documentation
 - It is important because it ensures that complex systems, designs, and processes can be understood, replicated, and maintained over time. A good way to demonstrate this virtue in the senior design work is to contribute to this design document.

Max Tanruther

- Virtue Demonstrated: Trust
 - It is important to trust your teammates to do good work because without it, cooperation collapses. This was demonstrated by trusting my teammates to do their work.
- Virtue to Demonstrate: Excellence
 - Doing exceptional work is important because it makes for a better product. I can demonstrate this better by going above and beyond in doing further work on the project.

8 Conclusion

8.1 SUMMARY OF PROGRESS

We have made significant progress in developing our solution. We have designed and developed a prototype to be implemented next semester. The GUI allows students to visualize the sensor data in graphs, see the results of data analysis, and save the raw data to a file for their own later analysis. The PCB allows the students to hook up the sensors and see the electronics that make up the system. Our goal was to design a bioinstrumentation platform for students to use in BME 3500. We decided the best solution was to design a PCB board with an accompanying GUI for the students to utilize when gathering sensor data.

Our main constraint in achieving our goal was time, because we had to get it done by the spring semester. This resulted in us having a very simple design in order for us to get it done on time. In a future implementation, we would want to investigate a more advanced design.

8.2 VALUE PROVIDED

Our design provides a hands-on, integrated learning tool for students working with biomedical signals. Unlike isolated hardware or software solutions, CyVital demonstrates the full pipeline from physical signal acquisition to processed data visualization. The system addresses key user needs like Clarity: real-time graphs and metrics help users understand physiological signals; Interactivity: users can select sensors, start/stop data collection, and export results; Practical exposure: students gain experience with both hardware (PCB, sensors) and software (data processing, GUI). Evidence of value includes successful live signal visualization, consistent sensor readings during testing, and a functional interface that supports real-time feedback and data export. The modular design also allows future teams to extend functionality without redesigning the entire system.

8.3 NEXT STEPS

Future work will focus on improving student experience with the system along with updating prototypes according to our clients wishes. The CyVital application will also be developed on a Mac device in order to expand our student audience. These steps are important to transition the system from a functional prototype to a more stable and scalable educational tool, while also enabling future teams to build on a strong technical foundation.

9. References

- [1] BIOPAC Systems Inc., "BIOPAC Student Lab," [Online]. Available: <https://www.biopac.com/product/bsl-pro/>.
- [2] OpenBCI, "OpenBCI: Biosensing for Everyone," [Online]. Available: <https://www.openbci.com/>.
- [3] H. da Silva et al., "BITalino: A novel hardware framework for physiological computing," Proc. ACM Int. Joint Conf. Pervasive and Ubiquitous Computing Adjunct Publication, 2014, pp. 479–482. DOI: 10.1145/2638728.2641693.

[4] Texas Instruments, "Reliability Calculators," [Online]. Available: <https://www.ti.com/support-quality/reliability/calculators.html>

[5] S. Haykin and B. Van Veen, *Signals and Systems*, 2nd ed. New York, NY, USA: Wiley, 2002.

[6] IEEE, "IEEE Code of Ethics," IEEE. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8>

[7] BIOPAC Systems, Inc., *MP36-MP45 MP Acquisition Units Hardware Guide*. [Online]. Available: <https://www.biopac.com/wp-content/uploads/MP36-MP45.pdf>

10. Appendices

10.1 APPENDIX 1 - OPERATION MANUALS

10.1.1 User Manuals

<https://github.com/CyVital/CyVital/wiki>

10.1.2 Hardware Manuals

Analog Devices, "AD8221 Low Power, Precision Instrumentation Amplifier," Datasheet, Rev. F, 2011. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/ad8221.pdf>

Analog Devices, "MAX32664 Biometric Sensor Hub," Datasheet, Rev. A, Jun. 2019. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/MAX32664.pdf>

Analog Devices, "MAX30101 High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health," Datasheet, Rev. C, Nov. 2020. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/MAX30101.pdf>

Analog Devices, "AD8232 Heart Rate Monitor Front End," Datasheet, Rev. C, Oct. 2011. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/ad8232.pdf>

Digilent, "Analog Discovery 3: USB Oscilloscope, Logic Analyzer, and Multi-Function Instrument," Datasheet, 2023. [Online]. Available: <https://files.digilent.com/datasheets/Analog-Discovery-3-Datasheet.pdf>

R. Arora, "I2C Bus Pullup Resistor Calculation," Application Report SLVA689, Texas Instruments, Feb. 2015. [Online]. Available: <https://www.ti.com/lit/an/slva689/slva689.pdf>

Texas Instruments, "CD74HC4067 16-Channel Analog Multiplexer/Demultiplexer," Datasheet, Rev. SCLS189D, Apr. 2015. [Online]. Available: <https://www.ti.com/lit/ds/symlink/cd74hc4067.pdf>

Texas Instruments, "TLO82H JFET-Input Operational Amplifier," Datasheet, Rev. SBOS735A, Apr. 2018. [Online]. Available: <https://www.ti.com/lit/ds/symlink/tlo82h.pdf>

10.1.3 Software References

Digilent, "DWF Python API Documentation: Introduction," [Online]. Available: <https://dwfpy.readthedocs.io/en/latest/introduction.html>.

10.2 APPENDIX 2 - CODE

<https://github.com/CyVital/CyVital>

10.3 APPENDIX 3 - TEAM CONTRACT

Team Procedures

- Day, time, and location (face-to-face or virtual) for regular team meetings:
 - We will meet weekly on Tuesdays from 4-5pm (after Senior Design) with the client. We will use any extra time to discuss progress and action times for members to complete.
 - The office of our client (Dr. Lu) will be used for client meetings. For additional meetings, we will meet virtually.
 - We expect members to reserve our meeting time on their calendar except in emergencies (such as sickness and family emergencies).
- Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):
 - Our primary method of communication within the team will be text messages. We will communicate with our client through Slack.
 - We expect members to check the text message group chat and Slack channel every day during the school week.
- Decision-making policy (e.g., consensus, majority vote):
 - For overall decisions, we will decide based on a vote with the winning option needing three-fourths of the team in support. We will also allow the exception of allowing for expertise according to one's major. The electrical engineers will have authority when dealing with hardware issues, while the software engineers will have authority when dealing with software issues.
 - Those in the same major should work to reach a consensus before making major-specific decisions.
- Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):
 - We will create a Google Document, shared among our group members in our shared Drive, that contains meeting notes. We will designate a record keeper as part of our team responsibilities listed below.
 - We will be creating a ledger in Google Sheets to document any materials we wish to be purchased. This ledger will contain cost, manufacturer, shipping, and date of purchase. This will ensure we are under budget and complying with the client's standards.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:

- 1.1. We expect our team members to show up on time to all meetings that concern them.
- 1.2. We expect team members who are unable to attend will alert the team at least 2 hours prior to their absence.
- 1.3. We expect students who will be late to alert the team as soon as possible with the estimated amount of delay.
2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:
 - 2.1. We expect team members to fulfill their responsibilities before the deadline.
 - 2.2. We expect team members to communicate if the deadline is too tight, and if they are struggling with fulfilling certain responsibilities.
 - 2.3. We expect team members to manage their time responsibly to prevent the team from following behind the project schedule.
 - 2.4. We expect team members to put in their best effort when fulfilling their responsibilities and spend adequate time to complete any assigned tasks.
3. Expected level of communication with other team members:
 - 3.1. We expect team members to communicate with each other, whether through text or in-person, with any issues or additional project information.
 - 3.2. We expect team members to keep a written record of important information about the project which can be added to the meeting minutes.
4. Expected level of commitment to team decisions and tasks:
 - 4.1. We expect all team members to participate in team decisions and tasks, and inform others if they are having trouble or are in disagreement.
 - 4.2. We expect all team members to respect another member when speaking through active listening and asking clarifying questions in a neutral tone.
 - 4.3. We expect all team members to stay focused on Senior Design material when meeting in order to respect everyone's time.
 - 4.4. We expect all team members to contribute to the assigned Weekly Reports.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):
 - 1.1. Claire will be in charge of team organization and integrative design. She will help resolve any conflicts that come up between team members. She will ensure that each of the software requirements are met in the hardware design. Claire will also be in charge of ordering all necessary materials for the design.
 - 1.2. Reza will be in charge of client interaction. This entails scheduling meetings and taking the lead in communicating with him.
 - 1.3. Kate will be in charge of taking notes during meetings and organizing team documents. She will also be in charge of software design.
 - 1.4. Max will be in charge of testing and component design. This entails designing unit tests for hardware. He will focus on making recommendations when the testing is not successful. He will work on understanding different modules and installing these sensors on the hardware.
2. Strategies for supporting and guiding the work of all team members:

- 2.1. Team members within the same discipline will work on scheduling one-on-ones to ensure that everyone is on the same page and see if the other member needs assistance.
- 2.2. The team will keep track of their progress through the Google Drive and Spring board. If someone is facing a tight deadline, members who are not facing a pressing deadline should offer assistance to ensure that the team stays on track. The “Phone a Friend” strategy should not be used as a substitute for poor time management but may need to be utilized when there are roadblocks in certain aspects of the design process.
3. Strategies for recognizing the contributions of all team members:
 - 3.1. Contributions will be assessed by other team members in their area of responsibility (e. g., the team members on the software side will assess the amount and quality of work done by other software focused team members).
 - 3.2. At the end of major contributions, the team will convene for a design review. This can be a team opportunity to praise and constructively give feedback on ways to improve the design.
 - 3.3. Each team member will have ownership of their own contributions in public-facing documents such as the Senior Design website, BME3500 material, and Lightning Talks.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.
 - 1.1. We have SE and EE students on the team, bringing a variety of majors to the project. Our SE students – Kate and Reza – have expertise in programming Python applications, like the one we will create. Kate also is pursuing a minor in cyber-physical systems like the one we’re creating. EE students, Claire and Max, have backgrounds in signal processing and circuit design. They will be in charge of managing the hardware aspects of the Cy-Vital. Each EE member has experience with PCB design and data management.
2. Strategies for encouraging and supporting contributions and ideas from all team members:
 - 2.1. Our team will remain inclusive and respectful of every team member’s ideas and contributions. If an idea needs to be revised, we will provide constructive criticism to reach a general consensus.
 - 2.2. Our team will work to accommodate everyone’s schedules when possible to ensure that no one is behind on important information.
 - 2.3. Our team will use a Sprint board similar to Jira where each person will be assigned a task and can propose other ideas to the team. In these schemes, members are allowed to award points based on the feasibility and value of the tasks. We can review the task board during our weekly meetings.
3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)
 - 3.1. If somebody feels they are not being included or experiencing conflict, they should speak to the team leader. The team leader should do their best to meet individually with the person(s) of concern and identify the root cause of the situation. The

team leader should bring up a course of action to make the environment more inclusive and collaborative.

- 3.2. If somebody feels they are unable to speak in the team environment, they should seek advice from the graduate TAs as to how to best address their concerns. The graduate TAs will have discretion to notify the professor, but it is hoped that the team can reach a peaceful resolution.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:
 - a. Since BME 3500 will be taught starting next semester, our main goal is to have a fully working product by the end of the semester.
 - b. Hardware goals are to add respiratory and blood oxygen sensors. If there's time, we will potentially also add a blood pressure sensor. We will also create a modular PCB for student interaction.
 - c. The software goals are to expand the existing GUI to incorporate the new sensors, as well adding ways to perform data analysis.
2. Strategies for planning and assigning individual and team work:
 1. We will plan to assign work to team members based on their major/skillset so that individual work will be assigned to the person most qualified.
 2. We will create a tasklist or Sprint board using software like Monday. This will help ensure that each of the members are assigned tasks of relatively equal responsibility.
 3. We will ask team members what individual work they may be interested in. This will help keep everyone engaged in the project.
3. Strategies for keeping on task:
 1. We will set time and real goals to work outside of class, and keep up with deadlines.
 2. Individual group members will try to be assigned to work on aspects they are interested in to keep their attention.
 3. We will use resources from past students whenever stuck.
 4. We will talk to Prof Meng (PhD in EE) whenever stuck, especially with sensor connectivity and circuitry.

Consequences for Not Adhering to the Team Contract

1. How will you handle infractions of any of the obligations of this team contract?
 - 1.1. We will ask the team leader to have a one on one conversation with the person committing the infraction.
 - 1.2. If the team leader is committing the infraction, the other team members will convene and remind the team leader of their obligations.
2. What will your team do if the infractions continue?
 - 2.1. There will need to be at least three discussions with the non-adhering members. If it is clear the member will not adhere to the contract, we will contact the graduate TA for the best course of action.
 - 2.2. We expect the graduate TA to escalate any team problems to the professors as necessary if the course of action did not result in a solution.

- a) *I participated in formulating the standards, roles, and procedures as stated in this contract.*
- b) *I understand that I am obligated to abide by these terms and conditions.*
- c) *I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.*

- 1) _____ Claire Haas _____ DATE 4/3/2026 _____
- 2) _____ Reza Choudhury _____ DATE 4/3/2026 _____
- 3) _____ Kate Endersby _____ DATE 4/3/2026 _____
- 4) _____ Max Tanruther _____ DATE 4/3/2026 _____

10.4 APPENDIX 4 - DESIGN ANALYSIS FROM FALL 2025

On the hardware side, the team has been able to design the PCB and order it. While waiting for the PCBs to arrive, the hardware team plans to develop testing methods and manufacturing plans. This will minimize the downtime the team has and make sure the team is ready once the parts arrive.

On the software side, the team has built the base GUI and created windows for over half of the sensors. The proposed design from 4.3 has worked so far. One thing that did not function as expected was the push button response. There was an electrical short that can be attributed to the prior breadboard system. From this experience, the team plans to work on documenting the software and providing instructions to students. Depending on the progress made, the software team hopes to add the blood pressure cuff and respiratory sensor to the GUI. The team continues to make progress and adjust deadlines as necessary.