5-6-2022

Remote Heart Diagnosis

Emma Miller
Trinity University

Edward Armitage
Trinity University

Gerald Daniels
Trinity University

Michael Marquez
Trinity University

MEMORANDUM

To:	Prof. Mehran Aminian, Project Sponsor/Team Advisor
From:	Remote Heart Diagnosis Team: Emma Miller, Edward Armitage, Gerald Daniels, Michael Marquez
Subject:	Final Project Report
Date:	May 6, 2022
CC:	Dr. Keith Bartels, Technical Advisor Dr. Darin George, Senior Design Administrator

For those in impoverished communities or remote regions, obtaining adequate healthcare can be a burden. Furthermore, limited access to specialists like cardiologists can make curable conditions a death sentence by leading them to be identified too late. An essential factor in the identification of heart conditions is the use of an electrocardiograph to measure the signal of the heart. In this project, the Remote Heart Diagnosis Team endeavored to design and build a prototype capable of remotely collecting and analyzing an electrocardiogram and displaying the results to a cardiologist in any location for review.

The prototype is composed of five main components. The first component is a printed circuit board designed to record the electrocardiogram. The second is a Raspberry Pi and touchscreen programmed to guide the user through the collection process, compile patient data and read the output of the circuit, run an artificial intelligence algorithm, and store all the information in the third component, a remotely deployed database, using a wireless connection. The fourth component is a website that accesses the database and allows doctors to view and interact with the device data. The final component is a three dimensional printed casing that houses the circuit, microcomputer, and touchscreen.

In early stages of testing, the team identified the need to transfer the circuit from a breadboard to a printed circuit board as the circuit often failed after being moved due to loosened wires. The team also discovered that the noise in the circuit was dependent on the wall outlet being used, leading to the addition of a filter in the circuit. As shown in the success of all but one final test, the prototype meets all expected qualifications, allowing for the changes in the potential diagnoses with the approval of the project sponsor. The only design requirement not achieved was in regards to the ability of the website to replicate a commercial electrocardiogram in form with 90% accuracy; however, as there were limitations with the commercial electrocardiogram used in terms of details in the data, measurement methods, and accuracy, the visuals were deemed reasonable due to their similarity to a traditional electrocardiogram.

Overall, the prototype is a fully functional proof of concept, as it is able to measure a clean electrocardiogram signal from a patient and collect their biographical data, analyze the electrocardiogram using a deployed artificial intelligence network, and store the results in a manner that can be remotely accessed on a website. This prototype is only a proof of concept, however, as, while the developed artificial intelligence network proved that the deployment and use of this type of network is possible, the network is unable to achieve functional accuracy due to a limited dataset. Before commercialization of the prototype, the neural network would need to be retrained using a large dataset of electrocardiograms collected using the device and labeled by a trained cardiologist. Furthermore, the neural network architecture and its ramifications in terms of classification should be reviewed with a professional cardiologist to ensure that image classification has the potential for functional accuracy. In addition, while outside the scope of this project, security code would need to be added to protect the website and transmissions before the prototype could be commercialized to comply with patient privacy laws and medical information regulations. Therefore, while the prototype was successful in achieving the desired functionality and in meeting the requirements, additional improvements will need to be made prior to the expansion of use.

I. Introduction

Despite extensive modern advancements in healthcare, heart conditions continue to plague our world, with heart disease being the primary cause of death in the United States for most demographic groups [1]. Furthermore, it is estimated by the Centers for Disease Control and Prevention that 12.1 million people will have atrial fibrillation in the United States by 2030 [2]. Many heart conditions or risk factors can be alleviated if identified in the early stages, preventing life-threatening consequences. However, due to cost, availability, and distance constraints on healthcare access for many individuals, these conditions go unnoticed until they become fatal. For example, between 2016 and 2017, the United States lost approximately $363 billion dollars per year towards heart disease in the form of medical services, medication, and lost labor hours [1]. While a significant strain on the US, impoverished nations and people without healthcare or stable employment cannot afford these costs, leading to a lack of proper medical care for these life-threatening conditions. By reducing the travel and time expenses for cardiologists and patients through semi-automated remote care, this device seeks to increase the accessibility of specialized cardiac healthcare for hard to access and impoverished populations.

The objective of this project was to develop a proof of concept prototype to demonstrate that this problem could be mitigated through the use of artificial intelligence and networking technology. The prototype was expected to remotely collect and analyze an electrocardiogram (ECG) and then transfer the collected data to a cardiologist in a different location for final confirmation of diagnosis. To demonstrate that the prototype's artificial intelligence could reliably inform a cardiologist's diagnosis, the algorithm was required to be able to diagnose the patient for normal sinus rhythm, atrial flutter, atrial fibrillation, myocardial infarction, and congestive heart failure with at least a 90% accuracy rate in validation testing. As a part of this requirement, the device must also record an ECG signal with minimal noise and at least 5 full cardiac propagations to avoid undermining the accuracy of the artificial intelligence algorithms. Furthermore, the networking for the device was required to operate between the device, database, and website with loading times under 5 minutes, a 90% accuracy rate in connection, and no loss of data. To allow cardiologists to confirm the diagnoses made by the prototype, the website was also expected to replicate signals of commercial ECGs in form with a 90% accuracy rate. Finally, to allow the device to function in remote regions, the device must be both portable and durable for transport, fitting within a carry-on suitcase matching Transportation Security Administration dimensions and surviving a drop test of 4 feet.

The primary constraints on the prototype were that it must follow Institutional Review Board policies to allow for human testing, and that the project must adhere to the two-semester timeframe and remain within the $1200 budget. Since the intended use of the prototype is medical, a variety of relevant codes and standards apply, including Food and Drug Administration regulation 21 CFR 820. Another key standard of note was that airlines provide limitations with regards to the maximum size of a carry-on, which is on average 22 inches by 14 inches by 9 inches. This standard is relevant because the device must remain in a carry-on to follow US Transportation and Security Administration requirements relating to the transport of batteries and electronic devices. In addition, codes including IEC 60601-2-25:2011, ISO 13485:2016, and IEEE 11073-10102-2012, provided guidance on key standardization and safety requirements for electrocardiograph devices.

To achieve the project objectives and requirements, the prototype was developed with five main subsystems, namely data collection, signal analysis, information transfer, display, and casing. The use of a convolutional neural network for electrocardiogram image classification, described in Section II.B., was chosen to achieve the requirements for results and accuracy of the analytic algorithm. To record the ECG signal and reduce noise, a circuit consisting of a signal monitor and power filter, along with an analog to digital converter to read the signal was employed. A relational cloud database accessed over Wi-Fi was selected to ensure short loading times, data preservation, and constantly accessible connections. Finally, the dimensions of the casing and the use of 3D materials were determined to ensure the portability and durability of the device. The full design will be described in the next section.

II. Overview of the Final Design

The final prototype is composed of five subsystems, namely data collection, signal analysis, information transfer, display, and casing. By combining these subsystems, the prototype directs the nurse to enter the patient's information such as birth date and phone number, attach the three leads to the patient, and then trigger an electrocardiogram reading. The electrocardiogram reading is then displayed on the device to allow the nurse to either save the reading or retake it if an error occurred. If saved, the electrocardiogram is analyzed using a deep learning algorithm to predict possible heart conditions, and then the signal and predictions are stored in a remote database using Wi-Fi. A doctor, once registered by the website administrator, can then search for patients by name and view their electrocardiogram signal and either confirm or dispute the diagnosis.

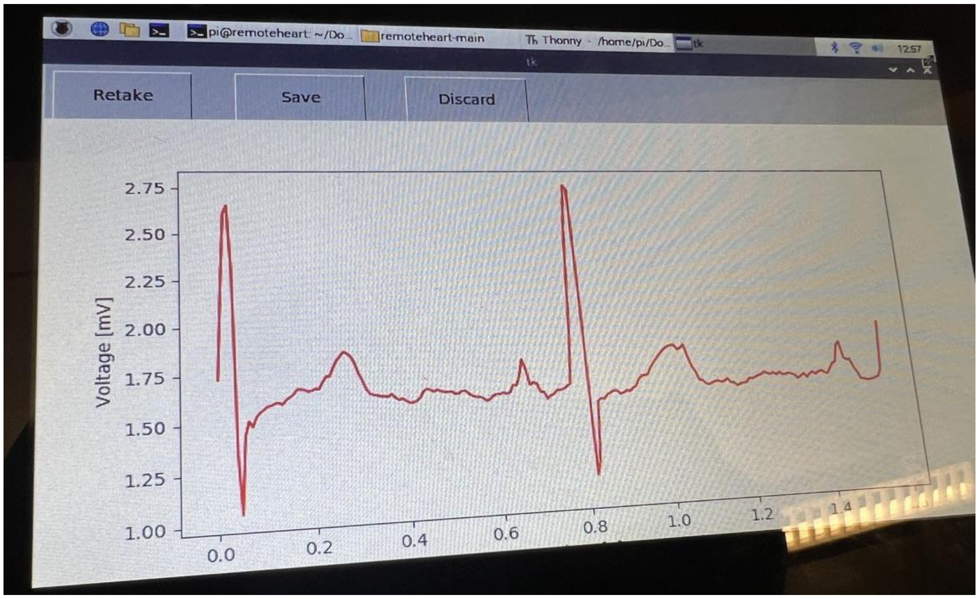


Figure 1. Photograph of electrocardiogram reading taken and displayed on prototype

II.A. Data Collection

To achieve its intended purpose, the prototype must collect two types of data, namely the patient's electrocardiogram signal and their biographical data to relate the signal back to them once it has been diagnosed. To facilitate this process, a touchscreen interface was designed to step the user through each aspect of the data collection procedure. The user interface was written in Python using the Tkinter graphical user interface library on a Raspberry Pi with a Raspberry Pi 7 inch touchscreen attached. A Raspberry Pi was selected as the device microcomputer due to its large quantities of open-source code and its versatility, and it serves as the brain of the device, reading in the electrocardiogram, analyzing it, and storing it in the database. A simplified flow representing this process is shown in Fig. 2, with a detailed software flow including data acquisition and analysis and database interactions included in Appendix Section V.B.

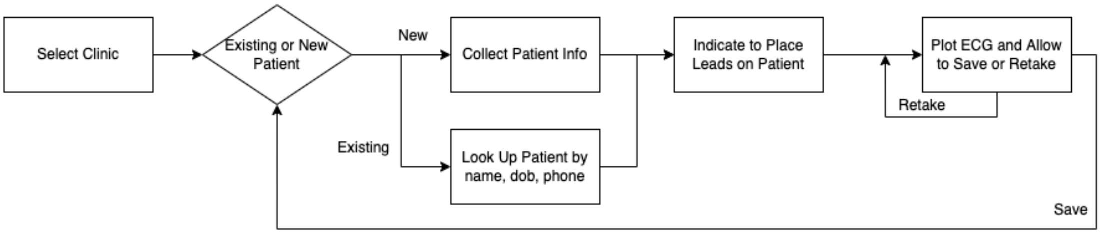

Select Clinic
Existing or New Patient
New
Existing
Collect Patient Info
Look Up Patient by name, dob, phone
Indicate to Place Leads on Patient
Plot ECG and Allow to Save or Retake
Retake
Save

Figure 2. User Interface Software and Process Flow Diagram

The graphical user interface collects the clinic's assigned identification number, and then allows them to either look up an existing patient using name, date of birth, and phone number, or record a new patient with name, gender, date of birth, phone number, and address. This data allows the doctors to reach out to the patients with any diagnosis after the analysis and maintains a record of previous electrocardiogram results for comparison. A key change from the Preliminary Design Report is the inclusion of an existing patient look-up feature, to allow for multiple electrocardiograms to be stored and compared for each patient, and to simplify the collection procedure. The interface then triggers the collection of an electrocardiogram, and plots the electrocardiogram for the nurse's approval. The software then cycles to prepare for another patient.

The electrocardiogram is collected via three leads connected to the patient in a standard triangular configuration, specifically right chest, left chest, and right abdomen. The signal is measured by attaching the three leads to a heart rate monitor integrated circuit (AD8232) and is then converted into a digital signal using an analog to digital converter (MCP3008), which is then passed into the Raspberry Pi, as the Raspberry Pi cannot accept analog inputs. The schematic for this circuit is included as Fig. 3.

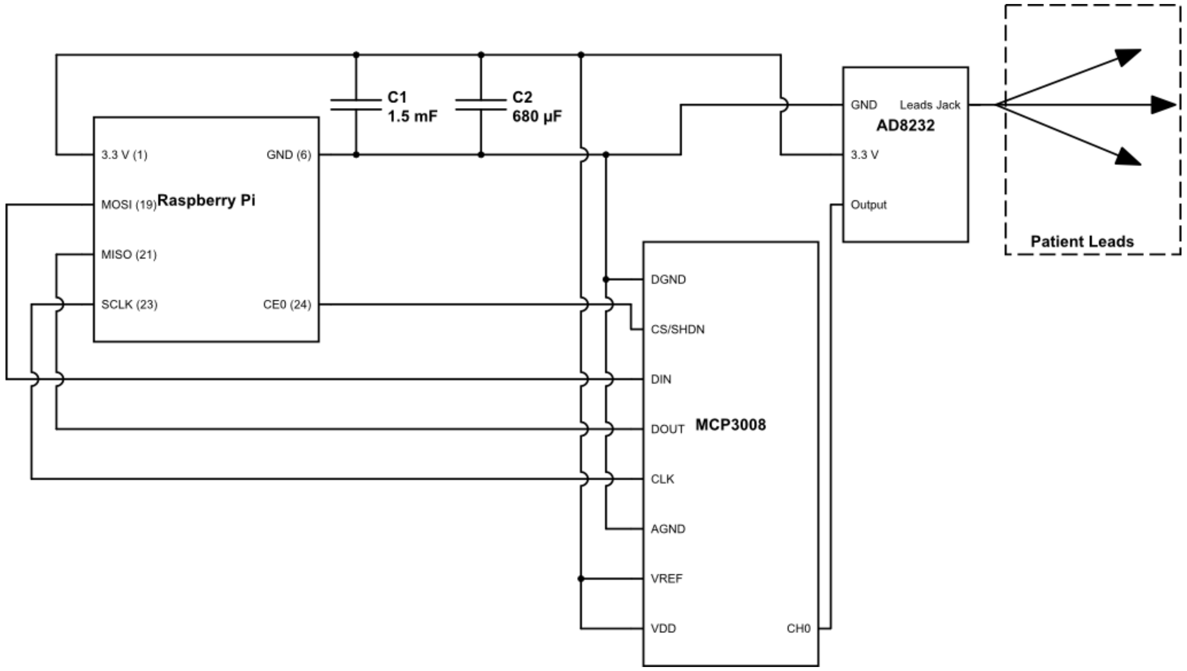


Figure 3. Schematic of the electrocardiogram circuit, simplified to display only used pins

A new change to this circuit is the inclusion of a filter in the form of two grounded capacitors between the 3.3 V output of the Raspberry Pi and the power inputs for the rest of the circuit, as it was discovered in testing that noise in the wall outlet used as a power source could transfer through the circuit and lead to significant noise in the recorded electrocardiogram signal. The capacitors were selected using the equation for the impedance of the capacitor, shown as Eq. 1, along with testing.

$$X_c = \frac{1}{j\omega C} \tag{1}$$

Another key change is the use of the Raspberry Pi 3.3 V output to power the other two components, as it was identified to be a poor user experience to have to plug the device into the wall and change a battery. Furthermore, the circuit was transitioned from a portable breadboard into a printed circuit board (PCB), as testing indicated that the readings produced by the prototype became highly irregular after being moved due to wires becoming loose.

A key requirement governing this subsystem is that the electrocardiogram circuit must record a signal with at least 5 full cardiac propagations and enough detail to be analyzed by artificial intelligence algorithms, which led the software to collect a full three minute signal and encouraged the addition of a filter to ensure that excessive noise would not prevent an accurate diagnosis. The circuit design was also restricted by the constraints on size in terms of the components that could be used, as well as the safety requirements imposed by International Electrotechnical Commission (IEC) 60601-2-25:2011 and by the Trinity University Institutional Review Board for testing on human subjects. Due to the need for the device to be transportable, Transportation Security Administration requirements for carry-on dimensions and battery types were also relevant to the design.

Initially, the circuit was designed on a portable breadboard, and the signals were frequently assessed using an oscilloscope. The PCB was etched using the Bantam Tools Desktop PCB Milling Machine in the Makerspace based on a schematic created in Eagle, and all the relevant components were soldered either directly into the PCB or attached via wires. For more detailed reference, the Eagle schematic for the PCB is in the Appendix Section V.C.I.

II.B. Signal Analysis

II.B.I. Background on Deep Learning and Artificial Intelligence

Before describing the design for the artificial intelligence algorithm, there are a few key concepts that must be addressed. Neural networks are a type of machine learning, and are typically used for deep learning algorithms [3]. Neural networks are designed to allow a computer to function like the brain, with nodes, representing neurons, providing either a value of 1 or 0 to indicate whether the input exceeded the activation threshold or not. These nodes are interconnected to form a series of layers, branching from the input layer to varying amounts of hidden layers to the final output layer [3]. For image classification such as that for electrocardiograms, a specific type of neural network known as a convolutional neural network is typically used, as it allows for the compression of data without the loss of feature information [3].

A convolutional neural network accepts a structured data array, such as an image, as input, performs feature extraction to identify key components of the image, and then classifies that image based on the identified features and related weights and biases [3]. The weights and biases are initially randomized and then developed through training prior to the deployment of the network for classifications. In training, a series of input images are provided to the network, and for each image, the output is predicted and then compared to the expected output from human labeling, and the differences between the results and the intended prediction are then back propagated through the algorithm to update the weights and biases using a user-selected learning rate and a gradient descent function to minimize the loss, which in turn increases the accuracy.

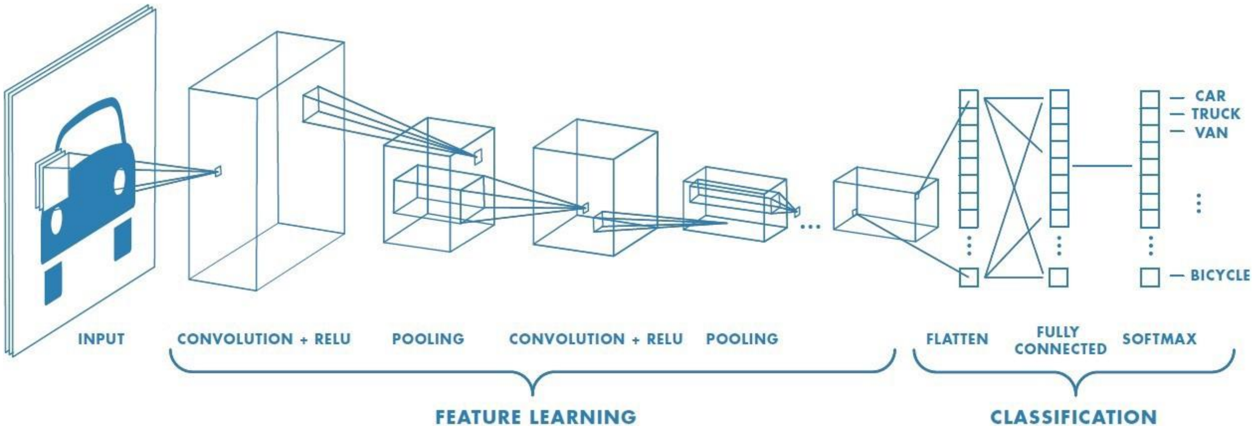


Figure 4. Diagram of a convolutional neural network [4]

As displayed in Fig. 4, a convolutional neural network involves a series of layers in the prediction process to convert a 3-dimensional input, a true color image, into a classification [3]. The first type of layer is the convolutional layer, in which a 2-dimensional matrix, known as a kernel, with values chosen to extract important features such as edges, is stepped across the input and a dot product is performed between the kernel and the overlapped subsection of the input, until this procedure has been performed for the entirety of the input matrix [3]. This process produces a feature map, which is often reduced in size compared to the input matrix. The final step in this layer is the application of a Rectified Linear Unit (ReLU) transformation to the feature map in order to provide non-linearity to the layers [3]. The ReLU function returns 0 if the input is below zero, otherwise it returns the input.

The next key layer in a convolutional neural network is the pooling layer, which reduces the dimensions of the matrix being passed through the network to improve efficiency and reduce computational complexity [4]. To do so, a kernel is traversed through the input to this layer, but instead of performing a dot product with weights in the kernel matrix, an aggregation function is applied to the section of the input covered by the dimensions of the kernel [3]. Typically, the pooling layer either computes the maximum or the average of the values in each region of the input covered by the kernel as it traverses, producing a reduced-size output array [3]. The convolutional and pooling layers can then both be repeated as many times as desired to refine the feature extraction in the network.

Once the convolutional and pooling processes are fully completed, the output of these layers is passed to the fully connected layer to perform the classification [3]. First, the input is flattened to produce a column vector [4]. This vector is then passed to a neural network, which consists of a series of hidden layers of nodes that activate, return 1, or do not activate, return 0, based on the inputs from the previous layer, the trained weights and biases, and an activation function, such as the ReLU function [4]. The neural network concludes with an output layer that functions similarly to the hidden layers, but often employs a different activation function [4]. For convolutional neural networks, the activation function typically used in the output layer is the softmax function, Eq. 1, which converts the numerical inputs into probabilities to represent the likelihood of the image matching each classification category [4]. In Eq. 2, σ represents the softmax function, *z* represents the vector input, *K* is the number of classification options, the *i* subscript denotes input, and the *j* subscript indicates output.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad (2)$$

The output of the layer with the softmax activation function is the final result of the convolutional neural network, and includes a probability for each potential classification that the image matches that category.

A drawback of convolutional neural networks is that they require significant quantities of labeled data in order to develop the weights and biases for the neural network layers in training. To overcome this issue, a strategy known as transfer learning can be applied, in which a pre-trained neural network for a similar task is retrained by changing only a few layers, such as the output layer, to adjust for the new potential classification options. The model can then be supplied with a smaller set of labeled training data to readjust the weights, but less adjustment of the weights is necessary as they have been partially pretrained. However, a risk in this process is that if too small a dataset is used, the model can become overfit, meaning that it is highly accurate for the data it was trained on, but significantly less accurate in real world applications.

For this project, transfer learning was utilized to retrain GoogLeNet to classify electrocardiograms for the selected five conditions. GoogLeNet is a convolutional neural network designed for image classification, with a base architecture similar to that described above, but with some additional features to improve accuracy and reduce the likelihood of overfitting [5]. For more information related to convolutional neural networks, please see [3] and [4], and for more information related to GoogLeNet, please see [5].

II.B.II. Design

When the user elects to save the electrocardiogram, the prototype performs signal analysis, and then saves the results along with the full electrocardiogram in the database. The signal analysis is performed on the prototype, and takes in the collected array of electrocardiogram data, plots the data, converts the plot into a true color image, and passes this image into the convolutional neural network for classification. The network then returns an array containing the probability that the signal displays any of the five possible conditions.

Since the Preliminary Design Report, the potential classifications for the model were changed from arrhythmia, myocardial infarction, atrial fibrillation, cardiomegaly, and normal sinus rhythm to atrial flutter, myocardial infarction, atrial fibrillation, congestive heart failure, and normal sinus rhythm. Arrhythmia was replaced with atrial flutter as atrial fibrillation was found to be a subset of arrhythmia in further research, and this overlap led to inaccurate classifications. In addition, cardiomegaly was replaced with congestive heart failure due to the availability of labeled data. The on-device signal analysis is performed using a C++ script and executable, as the optimal training program for the neural network was MATLAB which cannot be run directly on a Raspberry Pi.

Before the network could be used for classification, the convolutional neural network had to be trained. The inspiration and base code for the training program was derived from the MATLAB example, “Classify Time Series Using Wavelet Analysis and Deep Learning” [6]. As the use case in this situation is image classification, a convolutional neural network was selected. Furthermore, due to the limited labeled data available for training, the neural network was created using transfer learning, by retraining the GoogLeNet convolutional neural network to classify true color images of plotted electrocardiograms by changing the output layer along with key training parameters. Initially, another image classification convolutional neural network, SqueezeNet, was also retrained, but GoogLeNet was selected as it produced a higher validation accuracy in all testing.

The design requirement impacting the signal analysis component was the requirement that the algorithm be able to diagnose the patient for normal sinus rhythm, atrial flutter, atrial fibrillation, myocardial infarction, and congestive heart failure with at least a 90% accuracy rate in validation testing. This requirement was primarily relevant in the development of the network architecture, the selection of potential classifications, and the selection of training parameters such as learning rate and batch size, which impact the speed and accuracy of training.

All labeled electrocardiograms were downloaded from Physionet, an online repository of free medical data, and the list of all datasets used can be found in the Appendix Section V.G. These datasets were imported into MATLAB, and the program was trained using the aforementioned MATLAB program run on the Trinity University High Performance Computing System (HPC), known as Leviosa. The network was then extracted into a C++ executable file and deployed directly on the Raspberry Pi using MATLAB toolboxes, where collected data could be passed into the trained network for classification.

II.C. Information Transfer

To transfer the information from the prototype to the website, a database accessed over Wi-Fi was selected. The database is a PostreSQL database, which is an open source relational database, and is hosted on Heroku, a cloud platform as a service company that allows for limited free deployment of databases and websites. By deploying on Heroku, both the device and website are able to easily connect to and edit the database, and the use of Heroku and PostgreSQL was selected due to the team's familiarity with the set-up and the free access capabilities. Both the device and the website access the database through a wireless connection and Structured Query Language (SQL) commands. The database accesses on the device are performed using the "psycopg2" library for PostgreSQL in Python, and the database is accessed on the website using GET and POST requests.

The database stores a variety of relevant fields to record electrocardiograms and patient data, as well as key verification and account information. The structure of the deployed database can be viewed in Fig. 5 below, and the full relational schema is in the Appendix Section V.D.

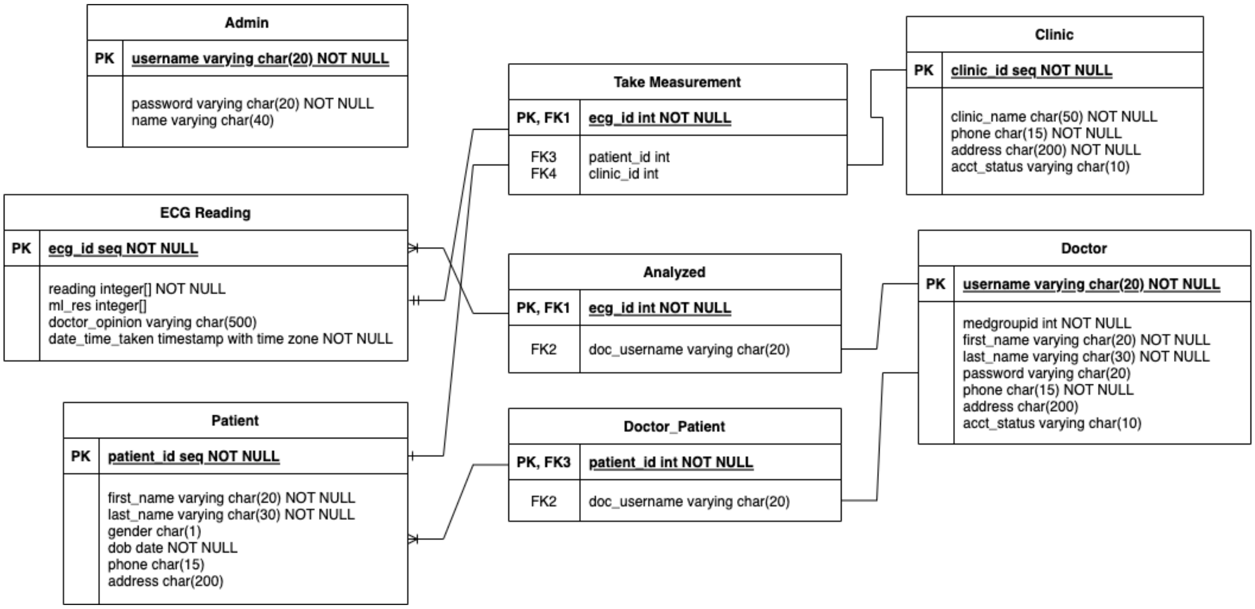

Admin
PK
username varying char(20) NOT NULL
password varying char(20) NOT NULL
name varying char(40)
Take Measurement
PK, FK1
ecg_id int NOT NULL
FK3
FK4
patient_id int
clinic_id int
Clinic
PK
clinic_id seq NOT NULL
clinic_name char(50) NOT NULL
phone char(15) NOT NULL
address char(200) NOT NULL
acct_status varying char(10)
ECG Reading
PK
ecg_id seq NOT NULL
reading integer[] NOT NULL
ml_res integer[]
doctor_opinion varying char(500)
date_time_taken timestamp with time zone NOT NULL
Analyzed
PK, FK1
ecg_id int NOT NULL
FK2
doc_username varying char(20)
Doctor
PK
username varying char(20) NOT NULL
medgroupid int NOT NULL
first_name varying char(20) NOT NULL
last_name varying char(30) NOT NULL
password varying char(20)
phone char(15) NOT NULL
address char(200)
acct_status varying char(10)
Patient
PK
patient_id seq NOT NULL
first_name varying char(20) NOT NULL
last_name varying char(30) NOT NULL
gender char(1)
dob date NOT NULL
phone char(15)
address char(200)
Doctor_Patient
PK, FK3
patient_id int NOT NULL
FK2
doc_username varying char(20)

Figure 5. Diagram of the database structure

There have been a few changes to the relational schema since the Preliminary Design Report to accommodate newly identified needs during the website development, including the incorporation of an admin table to store administrator credentials to allow for the approval of new doctors and clinics, and the addition of an account status (acct_status) field for doctors and clinics in their respective tables in order to allow doctors to apply and then be approved or deactivated for additional security.

The key design requirements for the network are that it should be able to access and transfer patient information to and from the device, website portal, and a complementary database with a loading time under 5 minutes and no loss of data and reliably transmit patient data wirelessly over a distance of at least 8 feet with a 90% accuracy rate in connection. By performing all accesses over wireless internet connections and maintaining a structure that does not require the transmission of large data sets, this design ensures that these requirements are met.

II.D. Display

The website was constructed using two separate languages and a Heroku database, described in the previous section. The front-end of the application was built in javascript using the React.js markup language. For the coding done in the frontend, new techniques called Hooks were used which were introduced in version16.8 of React. Hooks allow for the programmer to have a more functionally oriented code, instead of the traditional, heavily object-oriented method of programming a website in React. Using the basic react libraries, a framework for a single web application was created. However, at this stage the website could not transfer data to/from a database. In order to transfer data between the database, an Application Programming Interface (API) was implemented using Microsoft's .NET framework, written in C# (C sharp). The API is able to make HTTP requests, meaning requests that take place over a wireless connection to the database, which is deployed on heroku. In order to pull data in a meaningful fashion to the front-end of the application, libraries were used. The most important of these libraries was the react-chartjs-2 library. This allows the application to plot data collected as an array from the database. The website was styled using CSS (Cascading Style Sheets).

The reason React.js was chosen as a front-end language was because it allows for an easy transfer between single-page web applications and mobile application languages through the use of a related language called react native. If the designers wanted to deploy the application on the app store it would take less effort than if the program was written in a different language. The reason C# was used with a .NET framework for the API is because the group member who was working on the development of the website had the most familiarity with the language, and could implement authentication with Microsoft's services (Microsoft also developed the C# language).

The requirements of this design indicated that doctors using this software needed to be able to see and diagnose ECG signals presented on the website, and the data needed to be presented to them in a timely fashion, over varying WiFi connections.

In order to build this device, a windows machine was used in tandem with two separate folders developed in the Visual Studio Code IDE (integrated development environment). These folders were then run using commands corresponding to each language. However, since the development was entirely on the computer, the construction was entirely based on typing code into the IDE.

II.E. Casing

The casing for the device was designed in Autodesk Fusion360 and printed on a 3D printer, due to its adaptability, cost, and ease of use. In addition, 3D printed materials are relatively durable, and the flexibility of 3D printed design ensures that the circuit components are securely housed. The design of the case had to be built large enough to not only firmly house the screen, but also let the attached circuit comfortably rest beneath. The case was modeled and printed as two different pieces: the case and the lid.

The case houses the screen and the circuit. The length of the screen is 192.96mm. Thus, to provide easy insertion and removal of the screen, the L-shaped mounts were modeled to be an extra 2.14mm long. There are only two mounts within the case, preventing the screen from falling. There are no mounts above the screen, however, if the entire case were to be flipped upside down, the lid would keep the device in place. There are also no mounts holding the screen in place on the top and bottom, only just the two sides.

Once the screen rests on the mounts, the circuit dangles below. The case needed to be deep enough to let the circuit rest comfortably at the point, but not squish the circuit to the bottom of the screen. Thus the depth of the case was designed to be 75.419 mm. Because the circuit is relatively small, and to avoid unnecessary printing, the case was designed at an angle below where the mounts are. The sloped design of the base also allows the case to rest at an angle, allowing for easy viewing for the patient. Two port-holes on one side of the case (the sloped portion) were designed to allow the patient to plug in both the power and ECG pads.

The lid and case attach to each other by a snap-fit design. The length and width of the lid is the same as the case, 230mm and 120mm, respectively. The lid was designed so that only the touch portion of the screen was visible (the screen has a black border that does not need to be visible). A slight downward curve along the inside borders helps the user reach the corners of the touch screen, as well as provide a more aesthetically pleasing look. The model for the casing is displayed in Fig. 6.

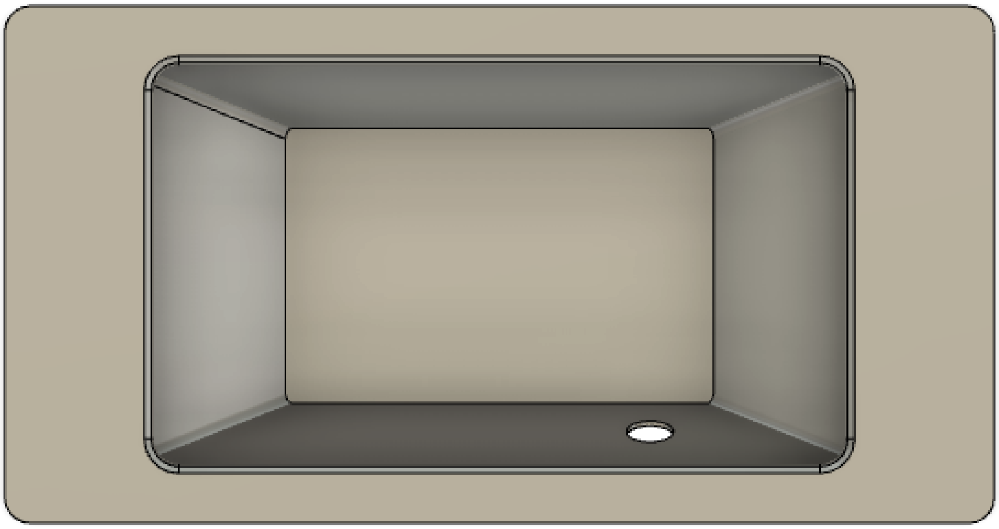


Figure 6. Three-dimensional computer-aided design model of casing

The material used for the casing is required to be relatively durable, as the project assumes the patient would be traveling with the device. The device and casing were required to be small enough to fit comfortably within a TSA compliant carry-on bag or backpack, and the dimensions selected in the design will ensure that this is possible.

The bottom portion of the case was printed using a 3D-printer, specifically, the Original Prusa i3 MK3. The material used was 1.75mm thick PLA. The lid on the other hand, was printed using 1.75mm ABS Nylon Black on an Ultimaker S5. Both the lid and case were modeled using Autodesk Fusion360. The lid was sliced with the Ultimaker's respective software, while the case was sliced with Prusa Slicer.

III. Design Evaluation

The following section delineates the tests performed by the team to evaluate each requirement or related grouping of requirements, as well as some testable constraints, and the results of that testing as it relates to the success of the prototype for each requirement.

III.A. Budget

III.A.I. Constraint Description

The cost of the project is limited to a $1200 budget provided by Trinity University. This constraint limited the scope, and restricted the processing power available for machine learning, as well as the options of ECG sensors for integration due to limits on purchasing.

III.A.II. Evaluation

The project remained well under budget, spending only $385.64 out of the $1200 allotted by the department. As such, this constraint was successfully fulfilled.

III.B. Wireless Transmission

III.B.I. Requirement Descriptions

The device must reliably transmit patient data wirelessly over a distance of at least 8 feet with a 90% accuracy rate in connection. The network should be able to access and transfer patient information to and from the device, website portal, and a complementary database with a loading time under 5 minutes and no loss of data. Furthermore, doctors should be able to interact with the machine and database with fewer than 5% errors in operation and loading times below 5 minutes.

III.B.II. Associated Tests

These requirements were assessed in the Database Storage and Retrieval Test, which tested the ability of the patient device, website, and server-based database to exchange and store information. This test was intended to evaluate the following requirements: the network should be able to access and transfer patient information to and from the device, website portal, and a complementary database with a loading time under 5 minutes and no loss of data, and the device must reliably transmit patient data wirelessly over a distance of at least 8 feet with a 90% accuracy rate in connection. The Database Storage and Retrieval Test assessed the functionality of the information transfer feature of the prototype, testing the ability of the device to transfer data between the physical device and the endpoint website and store the data over time.

This test gauged the prototype and the website's ability to access the database both when they are on the same wireless network and when they are operating on two different wireless connections, in essence testing the ability of the device to transfer data over a distance, especially distances exceeding 8 feet, where separate wireless connections would become necessary. Due to limits on available WiFi networks, both connections were of high strength, and so this test did not examine the impacts of a poor connection.

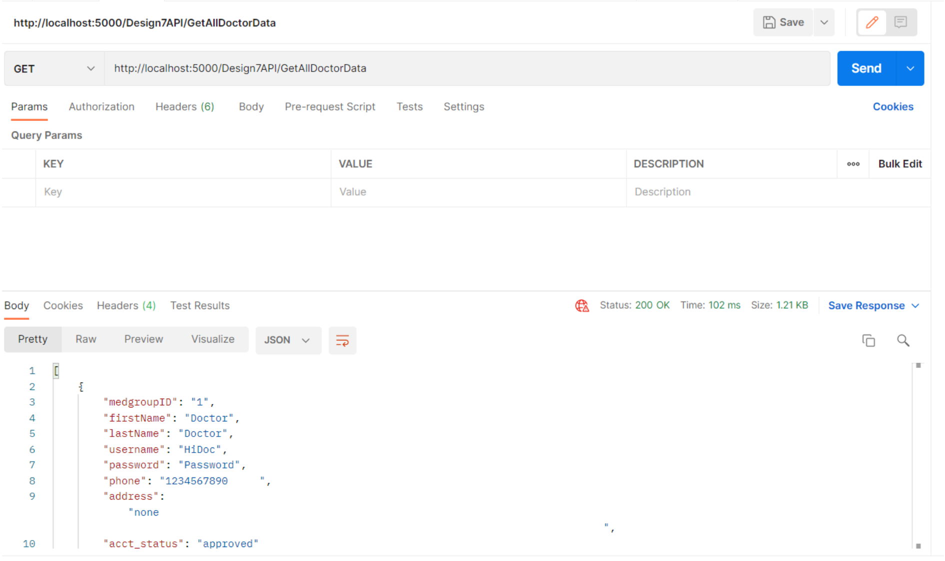
Testing the database required materials that are readily available for the team members, including the Raspberry PI, a computer, access to the Heroku database via an access string, and a software that can read database responses. From the website, the team executed a network request that connected to the database and returned a value. For the Raspberry PI, the database was accessed using the Python library "psycopg2" and MySQL commands, or directly using the Heroku Command Line Interface (CLI). In this test, a series of database additions were performed from the Raspberry PI, and then confirmed by direct database access, as well as by attempting to retrieve the data on the website. Then, a series of database additions were performed on the website, and then confirmed by direct database access, as well as by attempting to retrieve the data to the Raspberry PI.

The assumption in this test was that the data the team sent to and received from the database could be converted into a readable format, which was addressed in other tests. The team also assumed that the testing data utilized was reasonably representative of the data that will be entered during actual use of the device. The information collected from this test included the test database entries composed of the following: patient data (name, date of birth, etc.), electrocardiogram data, clinic data, and physician data. Each category contained multiple different forms of computer data (strings, ints, etc.), and each component of the category was collected for each set of test data inputted into the database and retrieved. For each iteration of the testing, the team recorded the value added to the database and the values returned by each retrieval method to ensure all returned the same result. This test was considered successful if all data entered could be retrieved by the website or patient device as well as through direct access to the database. In addition, all data was expected to be retrievable from the database within 5 minutes after the transfer process was initiated, and no data retrieval process should have exceeded 5 minutes. In all attempts to connect to the database, over 90% of these attempts should have been a success regardless of the relative positioning of the device and website host.

III.B.III. Test Results

All data transfer attempts were successful in both transmitting and receiving data of the expected types with no lost data or delays in service. The distance in time and space between the submission and retrieval of data proved to have no effect on the access capabilities. All data transfer operations proceeded within milliseconds, with the longest database access for the website completing in 733.65 ms. Furthermore, data submitted via the Raspberry Pi was accessible in well under a minute on the website, and vice versa. Furthermore, all attempts to access the database and the wireless connection were successful on both ends, indicating an 100% accuracy rate in connection on a stable wireless signal.

The following figure represents how the data was accessed without the use of the front-end Graphical User Interface (GUI) using the Postman software.



http://localhost:5000/Design7API/GetAllDoctorData

GET http://localhost:5000/Design7API/GetAllDoctorData

Status: 200 OK Time: 102 ms Size: 1.21 KB

```
[
    {
        "medgroupID": "1",
        "firstName": "Doctor",
        "lastName": "Doctor",
        "username": "HiDoc",
        "password": "Password",
        "phone": "1234567890     ",
        "address":
            "none
                                                                          ",
        "acct_status": "approved"
```

Figure 9. API request titled "GetAllDoctorData" being called in the Postman application.

In the above figure, the values returned from the API call are listed in the bottom half of the image. The time to complete this HTTP request is shown on the middle right, which also indicates the size of the request and the status of the request. All of these values fall within the acceptable ranges provided by the design team for successful tests. Similar results with varying time and size components exist for each function. However, all of the API requests that are used for this project are called in under 5 minutes, per the third design requirement assessed in this section.

III.B.IV. Evaluation

In all testing performed, all data entered, even incomplete data, was stored in the database in the same form as entered, and all accesses to the database to extract, add, or update data were performed in under a minute regardless of distance or medium. As the prototype exceeded the accuracy, time, and distance expectations set forth in this requirement, the design was considered successful both in this test and in the fulfillment of the wireless transmission requirements overall. This indicates that the wireless transmission process is fully capable of handling the data transfer operations necessary to achieve the primary objective, a device capable of assisting with remote automated healthcare for cardiology.

III.C. Transportation and Durability

III.C.I. Requirement Description

The device must be portable and durable for transport through a carry-on in an airport setting following Transportation Security Administration standards on dimensions and survive a drop test of 4 feet while enclosed within a carry-on suitcase. This requirement also overlaps with the constraint that the patient device must be small and durable enough to be transportable in remote conditions.

III.C.II. Associated Tests

The Size and Durability Test aimed to evaluate the suitability of the prototype for travel and usage in remote conditions. This test assessed if the prototype met the design requirements that the device be portable and durable for transport inside a carry-on in an airport setting following Transportation Security Administration standards on dimensions, as well as that the device must be able to function following a drop test of 4 feet while enclosed within a carry-on suitcase. It was designed to evaluate the portability and durability features of the prototype and its casing.

The Size and Durability Test was intended to ensure that the device met the durability and portability requirements stated in the introduction. To validate that the prototype would survive traditional forms of travel by air and land, the device was placed inside of a TSA compliant carry-on suitcase along with some padding in the form of clothing. The suitcase was then dropped from various heights reaching a maximum of 4 feet, kicked, and knocked over to cause rotation, for a duration of two minutes. The device was then inspected and used in its intended manner to evaluate the damage and assess the transport durability. This test was considered a success if the device could fit within a standard carry-on suitcase and was still functional without any substantial damage following the drop test.

III.C.III. Test Results

The prototype was easily able to fit within the carry-on bag, with room for additional padding and other storage in the suitcase. After the two minutes of continuous jostling and impacts on the device, no damage was identified in the casing, touchscreen, or the internal circuit, and the device demonstrated no change in recording capabilities in subsequent testing of the electrocardiogram reading procedures.

III.C.IV. Evaluation

As the prototype fits in a carry-on suitcase and can survive a period of impacts similar to those expected in a travel situation, it achieves the acceptance criteria for the testing. Furthermore, the prototype meets the requirement and constraint on the size and durability of the device, as the team can reasonably expect the device to survive traditional travel conditions.

III.D. Data Collection

III.D.I. Requirement Description

The device must record an ECG signal with minimal noise and at least 5 full cardiac signal propagations so that less than 10% of the recorded signals return “noisy recording” from the artificial intelligence algorithms.

III.D.II. Associated Tests

The primary test of this requirement was the Data Acquisition Accuracy Test. The Data Acquisition Accuracy Test assessed the ability of the prototype to measure an electrocardiogram and transfer that data onto the microcomputer, as compared to a similar electrocardiography device. This test was designed to confirm that the device meets the requirement that it record an ECG signal with minimal noise and at least 5 full cardiac signal propagations. In addition, this test ensured that the primary device functionality of measuring an electrocardiogram in a manner that can be wirelessly transferred was achieved. Furthermore, the team intended to use this test to confirm that the electrocardiograms measured by the device matched a commercial electrocardiogram with relative accuracy, to ensure that the irregularities noticed in the November testing have been resolved. This test evaluated the data collection process, including the capabilities of the hardware circuit to measure an electrocardiogram from a human patient using electrodes and the usability of the touchscreen interface for patient information acquisition.

The test used 10 test subjects under desirable conditions, with all data entered correctly and a seated and still patient. In most cases, the electrocardiogram was administered in the proper locations; however, due to the lack of a medical professional present, in some cases the electrode placement impacted the signal output. Three of the previous subjects were brought back to assess the existing patient data entry format. The Data Acquisition Accuracy Test utilized an oscilloscope attached using a probe between the signal monitor, AD8232, and the analog to digital converter, MCP3008, in the hardware circuit, three electrodes attached to a human subject, a monitor attached to the Raspberry PI unit to read any collected data, and an Apple Watch to use as a sample commercial electrocardiogram attached to the same patient. Each test subject was attached to the electrodes of the prototype, asked to complete the data entry on the touchscreen using provided fake patient information, and then asked to remain still for 3 minutes while an electrocardiogram was recorded. An Apple Watch, selected due to the ease of access, was then attached to their wrist, and they followed the instructions on the screen to measure their electrocardiogram in that manner. To conduct this test, the team had to use various tools and settings on the oscilloscope as well as the principles of electronics in relation to the comparison of the three different signals recorded.

For the scope of this test, the team assumed that the use of the oscilloscope did not alter the signal read by the prototype, and that the data recorded on the device would be the same data transferred into the database. In addition, the team assumed that the electrocardiogram measured by the Apple Watch was accurate and a reasonable indicator of similar commercially available products. This test collected two sets of electrocardiograms from each of 10 test subjects, with one being measured using the prototype and the other being measured using the Apple Watch. The prototype electrocardiogram readings were stored in two formats, namely the oscilloscope reading and the Raspberry PI data array. In addition, each test subject entered fake patient data provided by the team to test the data entry system, and this data was collected by the Raspberry PI and confirmed against the team's dataset of fake information. The use of fake patient information allowed the team to validate the full data collection process without violating the Institutional Review Board expectations that the data not be stored in a manner where it can be traced back to the test subject.

The test was considered successful if the noise visible in the electrocardiograms measured under reasonable conditions on the device microcomputer matched within a 10% margin of error the electrocardiogram produced by the commercial device and included 5 full cardiac propagations. In addition, no data from the patient data entry testing should have been lost.

While the Data Acquisition and Accuracy Test focused on the data collection capabilities of the prototype, this requirement was also reviewed in the Full Process Test in terms of the functionality of the circuit in the entire intended cycle of use. The Full Process Test evaluated the prototype for the full cycle of functionality expected, beginning with the collection of the patient electrocardiogram and information and completing with the presentation of the data and diagnosis to the doctor in a different location. This test intended to confirm that all of the subsystems of the prototype can successfully function in tandem to produce the desired process and results requested by the sponsor in the definition of the project, and it evaluated all the key features in terms of their ability to produce the output necessary to continue towards the end goal, including data collection, signal analysis, information transfer, and display.

As edge cases have been evaluated for each subsystem individually, the Full Process Test included 5 iterations of the full process from data entry to confirmation of final results on the website with 5 different test subjects. This test required a computer to run the website, a WiFi connection, an oscilloscope to measure the electrocardiogram signal from the patients, and fake patient data to be entered into the system. The oscilloscope was attached between the heart rate monitoring board and the analog to digital converter, in order to allow the team to view the input signal for comparison with the output. The fake patient data was used similarly, to ensure that the information inputted matches the output and that no test subject personal information can be linked to their biomedical data. Each test subject was attached to the electrodes and asked to enter the data they have been provided. After the electrocardiogram had been recorded and stored, the website was accessed from another room and the results visible on the website were compared to the oscilloscope reading and patient dataset. The Full Process Test required signal comparison knowledge in terms of key aspects of the signal such as peak-to-peak voltage to ensure that the presented and recorded signals match.

For the scope of this test, the team assumed that any error handling present in the individual subsystem testing also functioned for the full system, with minimal full scale confirmation of these capabilities. In addition, the team again operated under the assumption that the integration of the oscilloscope with the hardware circuit did not influence the signal recorded by the device. In order to use the oscilloscope, the device was also not fully enclosed in its casing, and it was assumed that the closing of the case would not impede the device's functionality. In this test, the initial signal for each patient was recorded as was the results produced by the website. By comparing the final website output and the expected signal, the team was able to assess the base functionality of the device towards the primary goal of the project, the remote collection and analysis of a patient electrocardiogram. This test was considered successful if the website displayed an accurate rendition of the recorded electrocardiogram and a reasonable diagnosis for all trials.

III.D.III. Test Results

In November, the team assessed the basic hardware circuit composed of the leads to connect to the patient and a circuit board designed to extract the signal, along with all needed power subcircuits. This test indicated a need for cleaner and increasingly reproducible signals for the hardware circuit. In addition, the circuit design needed to be changed in order to account for power coming from a different device than the protoboards provided at Trinity University.

During testing, the prototype recorded reasonable electrocardiogram signals for each of the 10 test subjects in the Data Acquisition Test based on visual analysis, with limited noise. The oscilloscope reading and the recording on the Raspberry Pi also matched closely throughout the process. Furthermore, each three minute signal included significantly more than 5 full cardiac propagations. However, the team discovered that the circuit was highly susceptible to motion, and when the device was transported it often had to be recalibrated to obtain a proper signal.

To assess the accuracy of the circuit relative to current available options, the team compared the signal recordings to those performed by the Apple Watch. While the noise appeared to decrease as compared to the results from testing last semester, a more detailed comparison than visual estimation, the technique used in November, was performed to ensure accuracy. Four key peaks relevant to electrocardiogram analysis were examined, namely the P, Q, R, and S peaks. An average value for each peak from throughout both recordings was estimated, and the two values were compared to compute a voltage correction coefficient and a percent difference, the values for which are included in Table 1.

Table 1. Approximate Percent Difference Between the Adjusted Peak Voltages Measured By the Apple Watch and the Prototype

P (% diff.)	T (% diff.)	R (% diff.)	S (% diff.)
10.6	6.7	29.3	5.9
13.6	14.6	43.9	1.2
46.3	23.3	39.3	18.6
5.3	16.4	12.6	12.2
15.5	14.8	35.7	14.7
22.9	52.8	12.0	7.2
0.3	23.7	40.5	11.1
10.7	5.9	11.3	1.1
18.6	24.7	1.5	8.3
22.1	31.0	19.7	3.9
10.2	21.3	9.0	18.5

On average, there was a 17.5% difference between the peak readings, with the S peaks displaying the most similarity with an average difference of 9.3%, followed by the P peaks with 16.0%, the T peaks with 21.4%, and finally the R peaks with 23.2%. Despite some identified improvements for user experience in the graphical user interface, the patient data entry functioned as designed, with all fake data collected and stored in the database accurately with no losses. This indicates that the procedure for the collection of patient profile data achieves all objectives.

In the Full Process Test, the majority of electrocardiograms were collected as intended; however, it was discovered that the noise in the signal was heavily dependent on the power coming from the wall outlet, as the signal changed depending on the outlet used. This indicated that some additional measures needed to be added to reduce noise. During user experience testing, it was also discovered, along with a few minor glitches in the graphical interface functionality that were resolved, that a waiting screen is needed while the electrocardiogram is being recorded to indicate that the device is functioning.

III.D.IV. Evaluation

While the differences in key metrics between the Apple Watch and prototype recordings were higher for many signals than the 10% metric set in the acceptance criteria, the error is reasonable considering that 5 out of 10 signals were analyzed by the Apple Watch to be "Inconclusive" rather than "Sinus Rhythm", indicating that the signals may not have been cleanly collected and invalidating the assumption that the Apple Watch represents an accurate electrocardiogram. In addition, each recording is highly dependent on motion, and as they were taken at different times, the effects of motion may not be equally displayed on both recordings. There is also no clear pattern in terms of which peaks differed the most, indicating that differences between the two readings were irregular in terms of interval and placement, and all peak voltages are estimates due to the limitations of reading from the Apple Watch PDF results and due to the use of an analog-to-digital converter, so that the prototype readings are sampled at 128 Hz. As such, considering that visual analysis of each pair of recordings displayed distinct similarities, the team concluded that the prototype passed this test and resolved the issues identified in November.

Furthermore, while initial testing indicated that the circuit was easily disrupted when moved and power outlets could insert significant noise into the signal, these issues were mitigated through the conversion of the circuit to a PCB and the addition of a filter element. The identified issues in user experience testing were also resolved in the software. As such, the prototype met this requirement, producing signals with more than 5 full cardiac propagations and limited noise in the final testing, and producing no discernable errors in the readability for the artificial intelligence, as the recorded signals matched commercially recorded ones in form.

III.E. Diagnostic Accuracy

III.E.I. Requirement Description

The AI must be able to diagnose the patient from the ECG for atrial flutter, atrial fibrillation, myocardial infarction, and congestive heart failure with a 90% accuracy.

III.E.II. Associated Tests

The diagnostic accuracy of the artificial intelligence was addressed in the Proper Diagnosis Test. The Proper Diagnosis Test employed a wide variety of sample electrocardiogram signals that have been diagnosed by a cardiologist to evaluate the accuracy of the artificial intelligence algorithm. This test was designed to evaluate the previous requirement that the artificial intelligence be able to diagnose the patient using an electrocardiogram for arrhythmia, atrial fibrillation, myocardial infarction, and congestive heart failure with a 90% accuracy rate on all accounts, but also covered the accuracy rate with atrial flutter instead of arrhythmia. The Proper Diagnosis Test assessed the signal analysis feature of the prototype, determining the accuracy of the diagnoses produced by the device.

The tests included over 100 sample electrocardiograms diagnosed by a cardiologist to have normal sinus rhythm, arrhythmia, atrial flutter, congestive heart failure, myocardial infarction, or atrial fibrillation, along with some samples with combinations of these conditions. The accuracy of the artificial intelligence was measured primarily using validation testing during the training process, which employed sample validated and labeled electrocardiograms. As the team was limited in the test subjects that would be available for these testing procedures, and so many of the heart problems would not be present in the dataset, for the scope of this project, it was assumed that the device would be able to diagnose similarly between the provided samples and those collected from physical test subjects. In addition, while measures were taken to ensure the validity of the data, the accuracy of the cardiologist's diagnoses of the samples was assumed. Due to the limited availability of training and testing samples, it was also assumed that the accuracy of the model in validation and testing would be similar. Other reputable sources such as Mathworks have used a similar assumption in situations with limited data, indicating that this is a valid assumption for the scope of this project. The test was considered successful if each model diagnosed at a minimum 90% of the inputted signals with the correct heart condition.

However, artificial intelligence results were also collected during the Full Process Test, which evaluated the prototype for the full cycle of functionality expected, beginning with the collection of the patient electrocardiogram and information and completing with the presentation of the data and diagnosis to the doctor in a different location. As previously described, the Full Process Test involved repeatedly stepping through the entire use case for the prototype, beginning with data collection and completing with the display of the electrocardiogram signal and artificial intelligence results on the website. For the artificial intelligence aspect, the network was deployed onto the Raspberry Pi, and a prediction was generated for each signal prior to it being stored in the database. Each signal in this process is assumed to be normal sinus rhythm, as all test subjects were healthy college students in their twenties with no known conditions. In terms of the artificial intelligence, the test was considered successful if the artificial intelligence was able to perform a classification of the electrocardiogram signal while deployed on the device, and have a reasonable basis for each diagnosis.

III.E.III. Test Results

Initial testing led to the realization that there is overlap between the conditions selected for diagnosis, making a singular classification algorithm for these conditions impossible. Furthermore, since the data can overlap, the accuracy of sequential algorithms cannot be reasonably extrapolated. For example, atrial fibrillation is a type of arrhythmia, and so confusion between the two can be expected when running a direct classification procedure. As such, a new classification model was developed through extensive research on electrocardiogram analysis and by testing a variety of conditions to identify those that are clearly distinguishable from one another, as concurrent or related conditions will exceed the scope of this project without the assistance of a cardiologist. In this process, the team determined that arrhythmia should be replaced with atrial flutter as a diagnosable condition, as mentioned in the design portion of the report.

The highest accuracy model produced had an overall accuracy in validation testing of 96.9% for the conditions atrial fibrillation, atrial flutter, congestive heart failure, normal sinus rhythm, and myocardial infarction with 128 diagnosed electrocardiogram signals used as validation data. The confusion chart for this model is included in Fig. 7, and displays that each condition was properly identified in over 90% of validation attempts.

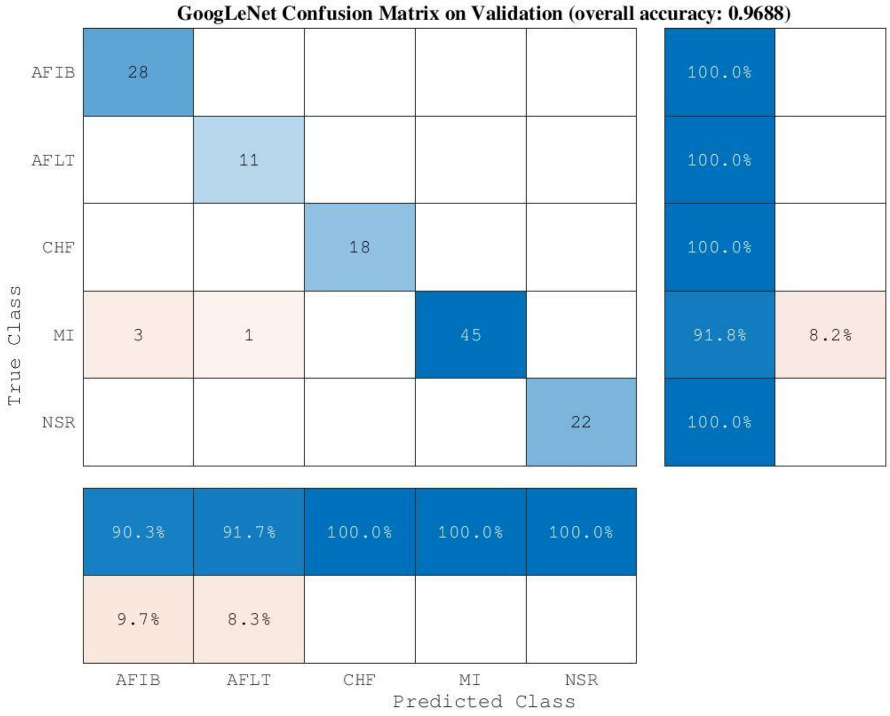


Figure 7. Confusion matrix of validation testing for artificial intelligence heart diagnostic algorithm

The assumption that the accuracy of the model based on validation procedures is equivalent to that for a separate testing dataset likely led to an overestimate of the accuracy. However, this assumption cannot be overcome without a significant quantity of electrocardiograms recorded by the prototype and classified by an experienced cardiologist, and thus exceeds the scope of this proof-of-concept prototype.

To further test the ability of this network to classify signals not included in the training, an electrocardiogram signal diagnosed with arrhythmia was sent to the deployment of the model on the Raspberry Pi. The model diagnosed the signal as having a 61% chance of being atrial fibrillation and 38% chance of myocardial infarction, with all other categories reporting negligible values. As atrial fibrillation is a subcategory of arrhythmia and myocardial infarction is known to be commonly concurrent with arrhythmias, this estimation is logical, indicating that the artificial intelligence handles unknown cases with reasonable accuracy.

For the Full Process Test, the team identified that, despite a validation accuracy of near 97%, the artificial intelligence diagnosed poorly all test subjects based on the collected data. As shown in the results denoted in the Appendix Section V.F.I, the algorithm returned atrial fibrillation for 11 subjects and myocardial infarction for 6 subjects, despite all 15 subjects, 10 from the Proper Diagnosis Test and 5 from this test, being expected to display a normal sinus rhythm. These results, along with the team's exposure to electrocardiogram signals during testing, highlighted the variation in normal sinus rhythm signals, as each collected signal displayed significant visual differences from the others collected, both in terms of the prototype and commercial electrocardiogram recordings. Therefore, as previously denoted, the creation of a functionally accurate artificial intelligence for heart diagnosis requires a significant increase in the quantity and variety of training data as compared to what the team currently has access to, as well as the assistance of a cardiologist to better understand electrocardiogram elements and variations.

Furthermore, the accuracy of the artificial intelligence was potentially limited due to the decision to employ an image classification algorithm and the use of only ten second signals for analysis due to the maximum length of available data. While image classification was used in many algorithms used for inspiration, such as [6], the conditions selected for analysis may not present in a manner that can be evaluated using feature extraction. This concern was highlighted in the comparison of the recorded signals to samples of the training data for the model. For example, a signal that should be normal sinus rhythm was classified as having a 99% chance of being positive for myocardial infarction; however, as shown in Fig. 8, this visually appears to be a logical result based on the matching T peaks, boxed in the diagrams, that do not appear in any of the other condition samples. While the plot only includes one sample of each training set, the small size of the training set enhances the effect of each sample, indicating that this could represent the issue in the functional accuracy of the model.

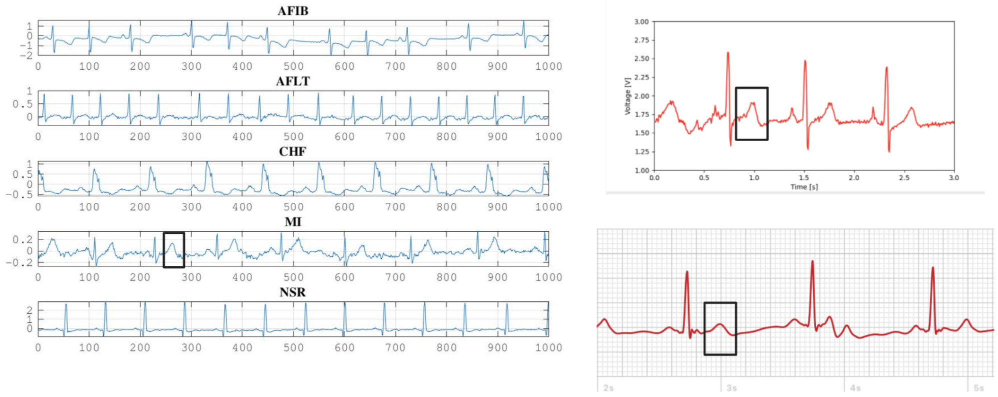


Figure 8. Comparison of training sample electrocardiogram data for each condition (left) to the recorded electrocardiogram from the prototype (top right) and Apple Watch (bottom right), with key attributes boxed in black

This phenomenon indicates that a more accurate model would require further input from an experienced cardiologist in order to refine the feature extraction to ensure that the relevant aspects in diagnosing a condition are used. Furthermore, a much larger training dataset is required to overcome the problems caused by variations.

III.E.IV. Evaluation

The neural network meets the design requirement acceptance criteria for both tests, as it achieved over 90% accuracy in validation testing for all five desired conditions and was successfully deployed on the Raspberry Pi, achieving the intent of the proof-of-concept project. Prior to commercialization, the functional accuracy of the model needs to be improved through the retraining of the model with a significant amount of training data collected on the prototype and labeled by a professional cardiologist. In addition, the choice of network, potential conditions for diagnosis, and the feature extraction methods should be reviewed by a cardiologist to ensure that the model would be capable of identifying the characteristics of each diagnosis in a real world scenario.

III.F. Data Presentation

III.F.I. Requirement Description

The website should replicate signals of commercial ECGs in form based on the machine readings with 90% accuracy.

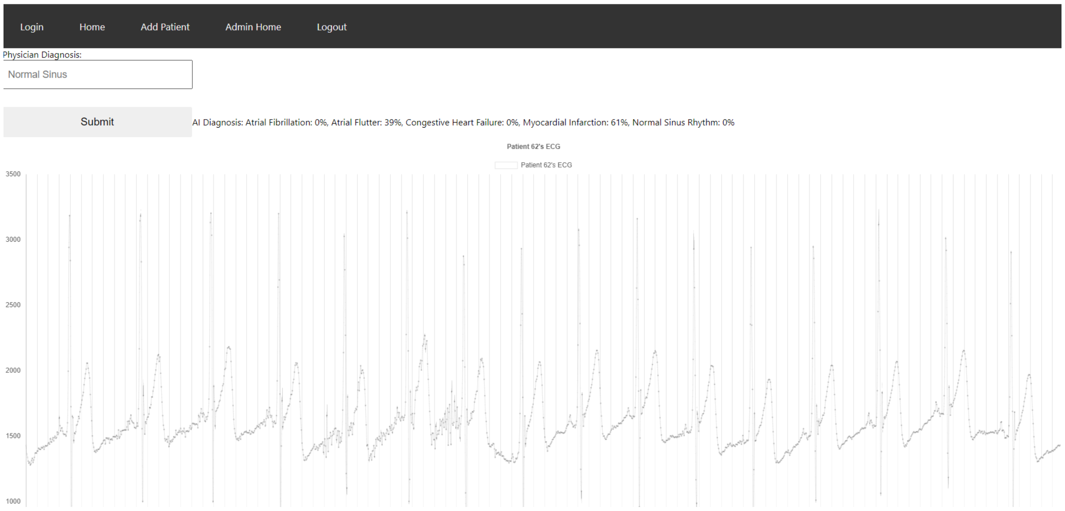
III.F.II. Associated Tests

In order to examine the ability for the website to display ECG data, the visual stimuli presented on the screen were inspected. This test was designed to determine if the application programming interface (API) calls return the correct ECG data. Furthermore, this test assessed if the single-page-application rendered the correct objects in the window of the website. The features evaluated in this test were the website functionality including data posting and retrieval and the rendering of components such as an electrocardiogram plot. For this test, the website had to successfully execute at least one Get and one Post request to the database to acquire data necessary for the cardiologist to review the patient's electrocardiogram. The ability of the website to produce graphical representations of the ECG was also examined by pulling at least one array from the database and presenting it on the screen.

Since this section of the project is entirely based in software development, the team only required a Windows machine and a stable internet connection to run the website application, along with the Postman application, a popular graphical HTTP client for API testing. First, the team used Postman to assess the effectiveness of a basic API call, namely retrieving a patient's information based on their name and phone number. For the non-networked portion of the testing, the team executed queries through the website process as a user would, ensuring that all key components such as login, patient data retrieval, and ECG plot rendered properly and that all interactive elements were functional. The test will be considered successful if the website fulfills all needed functionality during a simulated user experience and if the database accesses all occur with no data loss or delays while rendering the proper components.

III.F.III. Test Results

The website met all expectations for functionality, and the API calls were successful, indicating that the test met the acceptance criteria. Bugs that were discussed in the prototype test plan regarding authentication have since been resolved. Additionally, all components that needed to be rendered were successfully created. While the ECG data was identical on the website to the reading taken by the Raspberry Pi, the ECG displayed, on average, an 18.8% difference between the readings displayed on the website created by the design team and the ECG generated by the Apple Watch. An example ECG plot from a patient who used the device at the Trinity University Campus is shown below.



Physician Diagnosis: Normal Sinus

AI Diagnosis: Atrial Fibrillation: 0%, Atrial Flutter: 39%, Congestive Heart Failure: 0%, Myocardial Infarction: 61%, Normal Sinus Rhythm: 0%

Figure 10. Screen capture of the website created by the design team recording ECG data, which was collected from a student at Trinity University

III.F.IV. Evaluation

All of the tests behaved as expected for rendering the visual components on the website. The ECG data, when compared to the data collected by the Apple Watch has an 18.8% error, which is higher than the design criteria states. However, the Apple Watch does not provide significant detail in its readings, and as such may not be a valid calibration tool. Furthermore, the measurement methods differed, and could lead to differences in the recorded signal, with the prototype employing three leads in a triangle on the abdomen and the Apple Watch using the wrist. When compared to the traditional ECG waveform, shown below, the team's data, Fig. 10, more clearly represents true ECG waveforms than the ECG data from the Apple Watch, in Fig. 12. Therefore, the percent difference between the two readings is sufficient for the prototype since there are reasonable homologous regions between the two sets of data, and the differences are not expected to impede the use of the electrocardiogram for diagnosis.

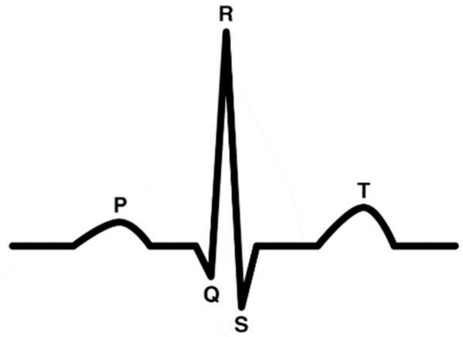


Figure 11. The characteristic, ideal waveform of each heartbeat presented on an ECG

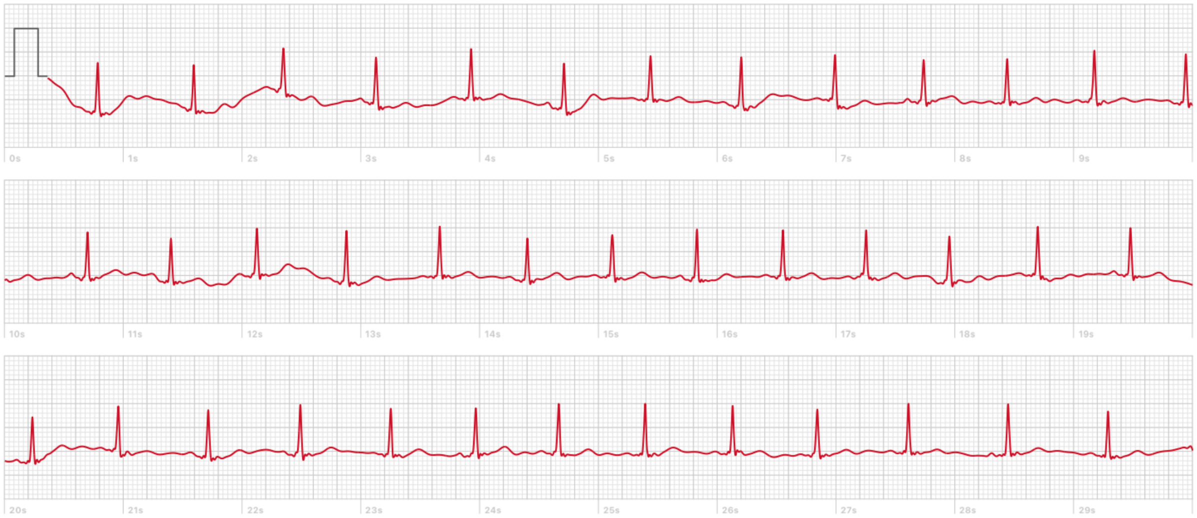


Figure 12. ECG data collected from an Apple Watch

IV. Conclusions

Overall, the prototype meets all qualifications stated in the most recent Project Plan, outside of changes made to the conditions diagnosed with the input and approval of the project sponsor. Furthermore, all but one of the project objectives and design requirements were achieved, with the prototype collecting all necessary data and performing in data transfer times well under the required maximum durations. The prototype was unable to reproduce a commercial electrocardiogram with 90% accuracy; however, due to the limitations on the details within the commercial electrocardiogram data and the difference in data collection methods, wrist versus three leads, the accuracy of the collected data was evaluated to be reasonably accurate and admissible.

This proof-of-concept prototype is functional, capable of collecting a clean electrocardiogram signal from a patient along with their biographical data, analyzing that electrocardiogram using a deployed artificial intelligence network, and storing that data for remote access via a website. However, as intended, the prototype is only functional as a proof of concept, due to the limited quantity of training data. Prior to commercialization or further deployment of the prototype, hundreds to thousands of electrocardiogram signals will need to be collected using the device and classified by a qualified cardiologist, and the neural network will need to be retrained using this data. In addition, for the scope of the project, it was assumed that the security of the website and transmission could be assumed. However, due to the inclusion of medical data in both aspects, additional security code would be necessary before commercialization to comply with privacy laws. In conclusion, the prototype was successful in its primary purpose, namely developing a method to automate the collection and analysis of an electrocardiogram to allow for remote cardiac care.

V. Appendices

V.A. Prototype Manual

V.A.I. Device Set-up

V.A.I.1. Doctor Access

If this is your first time using the product, please navigate to the following url: https://mydesign8herokuapp.herokuapp.com/. Click on “Apply for Doctor or Clinic Approval”, then click on “Apply as New Doctor” and fill out the form you are directed to. Once you have been approved by the administrator, you will then be able to login and view your patients’ results.

V.A.I.2. Clinic Device Use

If this is your first time using the product, please navigate to the following url: https://mydesign8herokuapp.herokuapp.com/. Click on “Apply for Doctor or Clinic Approval”, then click on “Apply as New Clinic” and fill out the form you are directed to. Once you have been approved by the administrator, you will then receive an identification number, which will allow you to begin taking electrocardiograms.

To set-up the device, plug the end of the black plug into a wall outlet, and place the device on a table, with the wires facing towards the table. Ensure that the three sensor cables are untangled and fully connected to the headphone jack within the device. Wait for the screen to boot, and when it awakes, click on the terminal icon in the upper left of the screen. The application will automatically boot, and you are now ready to take an electrocardiogram.

V.A.II. Device Operating Manual

V.A.I.1. Doctor Access

Upon accessing the website designed for the doctor/administrator you will be directed to a login page where credentials need to be placed into the corresponding boxes for username and password. After this is completed, a function located in the backend, C# application will verify if these credentials match a defined user in the heroku database. If the credentials are a match in the database for a doctor, who has been verified by an administrator, the user will be redirected to doctor homepage, where there are a few options. On this page, denoted “Home” on the navigation bar, there are a few buttons and two boxes which accept string entries. The first button, noted “Find Patient” will return a series of repeating tables which include the information for patients that match the data provided in each of the boxes. The second button titled “Find

Patient ID" will return the patient identification number of a patient whose credentials match those inputs in the form boxes on the page. The third button is used to list all the information of all the patients that are within an individual's perview, when no data has been retrieved from the database that region of the page will read "exception thrown."

Moving to the add patient page, which is another page which a verified doctor is able to access. On this page, there are a number of boxes with the required materials for each box. Once all boxes on the page are filled in with the appropriate patient information the button, located to the left of the patient information, can be pressed. If the information is correctly sent to the database, then an alert dialog window will inform the user that the patient information was added to the database. If the database did not properly add the information to the database, the user will also be alerted that the information was not transferred.

V.A.I.2. Clinic Device Use

First, if you have not already, follow the set-up instructions in Section V.A.I.2 of the manual. Once the application has loaded on screen, follow the instructions provided on the screen to step through the process. You will be asked to click "Begin" and then provided with a text entry box to input a clinic id. Here, please input the identification number you were provided upon the registration of your clinic on the website, and click "Submit". You will then be asked if the patient is a new or existing patient. If the patient has not taken an electrocardiogram with this program before, select "New Patient", fill in all the fields provided on screen with the patient's information, and click "Submit. If the patient has previously used this system, select "Existing Patient", fill in the search fields, and click "Submit". You will then be instructed to attach the electrodes to the patient. If the patient is not already seated, please ask them to sit down. Attach new disposable electrocardiogram electrode pads on the end of each sensor cable, and stick the electrodes on the patient according to the diagram in Fig. 13, taking care to ensure that the color of the cable being attached matches the color of the dot on the diagram at that position.

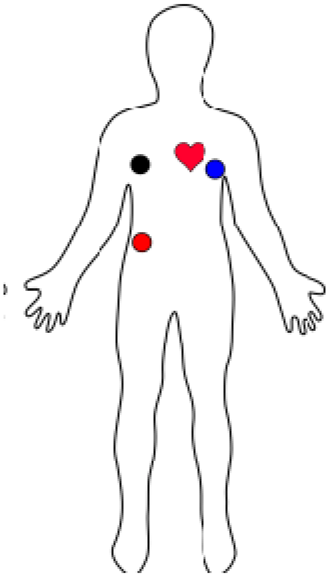


Figure 13. Electrode placement diagram [7]

When the electrodes have been placed, ask the patient to remain still and press “Start ECG”. A plot of the electrocardiogram will appear on the screen when the collection is complete, and you will be able to select either “Retake” if there is an issue in the signal and you would like to try again, “Save” if you would like to store and analyze the signal, or “Discard” if you do not want to save the signal and do not want to try to retake it. Once either “Save” or “Discard” have been selected, you will be returned to the “New or Existing Patient” screen.

If at any time you would like to turn off the device, click “Shutdown” in the menu bar at the top of the screen. Once the screen turns black, unplug the device from the wall. If you would like to start the process over from the beginning or change the clinic id, press “Start Over” in the menu bar at the top of the screen. This will return you to the welcome screen.

V.A.III. Safety Notes

Please be sure to keep the device away from water. The casing is not watertight and the circuitry will likely be damaged, and there is a high risk of electrocution. Avoid touching any wires inside the device while it is plugged in, and avoid contact with the plug when it is entering or exiting an outlet. While the device should not pose any harm to your health, please confirm with a medical professional before use if you have any unusual implants that may interfere with the signal collection.

V.A.IV. Troubleshooting

The signal being recorded is not the proper electrocardiogram

Try disconnecting and reconnecting the sensor probes to the headphone jack to ensure a tight fit. In addition, be sure that the electrodes are connected directly to the patient’s skin in the configuration shown in Fig. 13. If this does not work, try shutting down the device from the application, then unplug the device and plug it back in.

The digital keyboard disappeared for data entry

The keyboard may have become hidden behind the foreground tabs on the screen. Try minimizing the application window and confirming that the keyboard is no longer present. If it is not, close the application using the “x” in the top right corner, and then click on the terminal button again to reopen the application. The keyboard should reappear on the data entry screens.

The application is malfunctioning/frozen

Close the application using the “x” in the top right corner, and then click on the terminal button again to reopen the application. The keyboard should reappear on the data entry screens. If this does not work, try shutting down the device from the application, then unplug the device and plug it back in.

The electrocardiogram is extremely noisy

First, shut down the device from the application, and try plugging it into a different wall outlet. If the noise remains, try disconnecting and reconnecting the sensor probes to the headphone jack to ensure a tight fit. In addition, be sure that the electrodes are connected directly to the patient’s skin in the configuration shown in Fig. 13. If this does not work, try shutting down the device from the application, then unplug the device and plug it back in.

Doctor’s diagnosis is not being recorded

Check the syntax of the input. If there are strange characters such as exclamation points, question marks, or any other characters that would not be accepted in a query, then remove those characters and the diagnosis should be recorded in the database.

Unable to access patient ECG

From the "Home" Menu, remember to click the patient ID column of the table after making your query using the information inserted into the two boxes on this page. If you click on a different region, the page will not progress to the patient's ECG.

Doctor/Clinic was not removed from the to be approved list

Switch to a different page, return to the approval list. If the doctor/clinic is still there. Press the button and repeat one more time. If the doctor/clinic remains after that contact us.

V.A.V. Assembly Instructions

To build another device, first collect all parts described in Appendix Section V.E. Then, remove the microSD card from the prototype, and connect it into your computer. Duplicate the contents of the microSD onto another microSD card, and place this card into the new Raspberry Pi. Follow the instructions that come with the 7" touchscreen to connect the Raspberry Pi and touchscreen using the ribbon cable and two jumper cables attached from the Raspberry Pi pins 3 and 4 to 5V and GND on the display respectively.

Next, use the Eagle files, displayed in Figure 15, to mill a PCB, and solder the AD8232 and MCP3008 into the board. Connect the Raspberry Pi to the PCB according to the diagram in Figure 3 by soldering jumper cables into the PCB. Then, print the device casing using the CAD model, shown in Appendix Section V.C.II. Place the circuit, Raspberry Pi, and touchscreen inside the casing. Feed the power plug for the Raspberry Pi and the sensor cables headphone jack through the holes in the casing, and connect the sensor cables to the mount on the AD8232, and the USB-C power cable into the USB-C port on the Raspberry Pi. Now, close the lid firmly. The device is now fully assembled and ready for use.

V.B. Raspberry Pi Software Flow

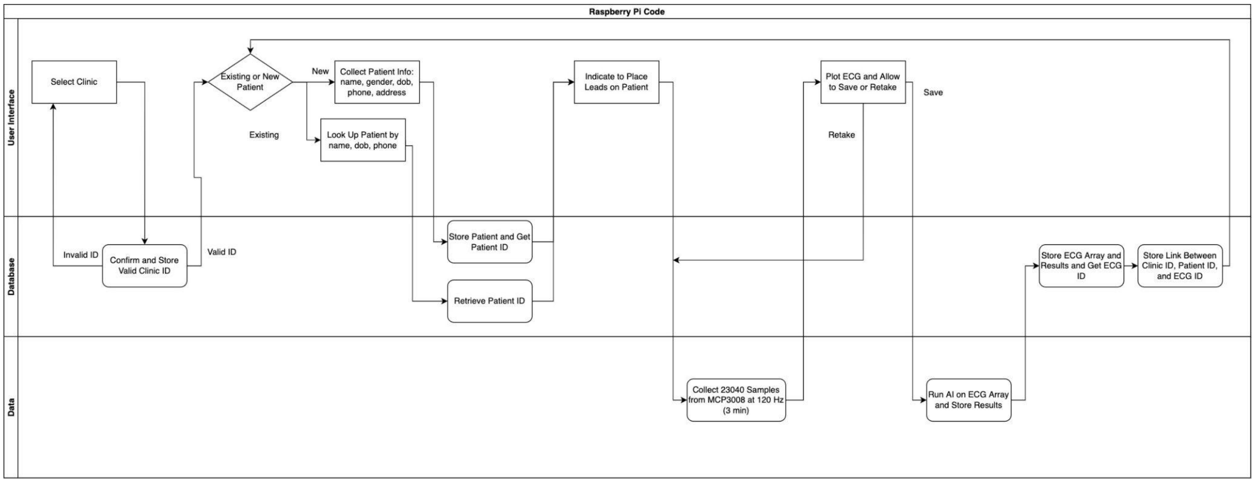

Raspberry Pi Code
User Interface
Database
Data
Select Clinic
Invalid ID
Confirm and Store Valid Clinic ID
Valid ID
Existing or New Patient
New
Existing
Collect Patient Info: name, gender, dob, phone, address
Look Up Patient by name, dob, phone
Store Patient and Get Patient ID
Retrieve Patient ID
Indicate to Place Leads on Patient
Collect 23040 Samples from MCP3008 at 120 Hz (3 min)
Plot ECG and Allow to Save or Retake
Save
Retake
Run AI on ECG Array and Store Results
Store ECG Array and Results and Get ECG ID
Store Link Between Clinic ID, Patient ID, and ECG ID

Figure 14. Flow diagram depicting the software flow on the Raspberry Pi including the graphical user interface, database interactions, and data acquisition and analysis

V.C. Building Plans

V.C.I. Circuit Schematic

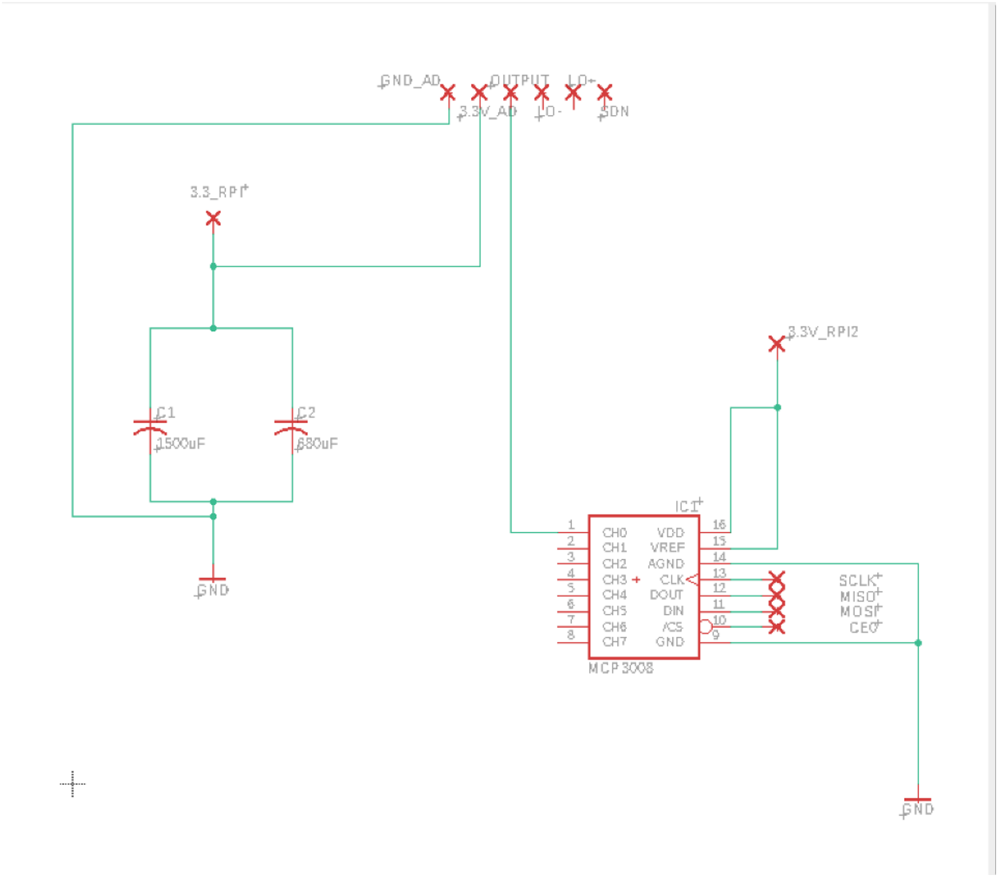
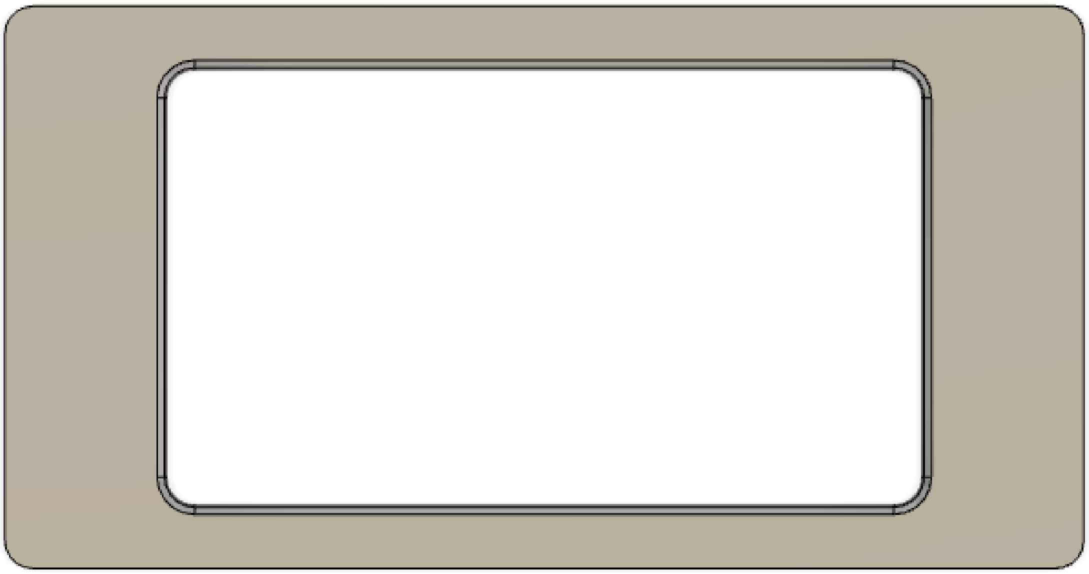


Figure 15. Schematic diagram of the electrocardiogram recording circuit in Eagle that was printed on PCB

V.C.II. Computer-Aided Design (CAD) Casing Models



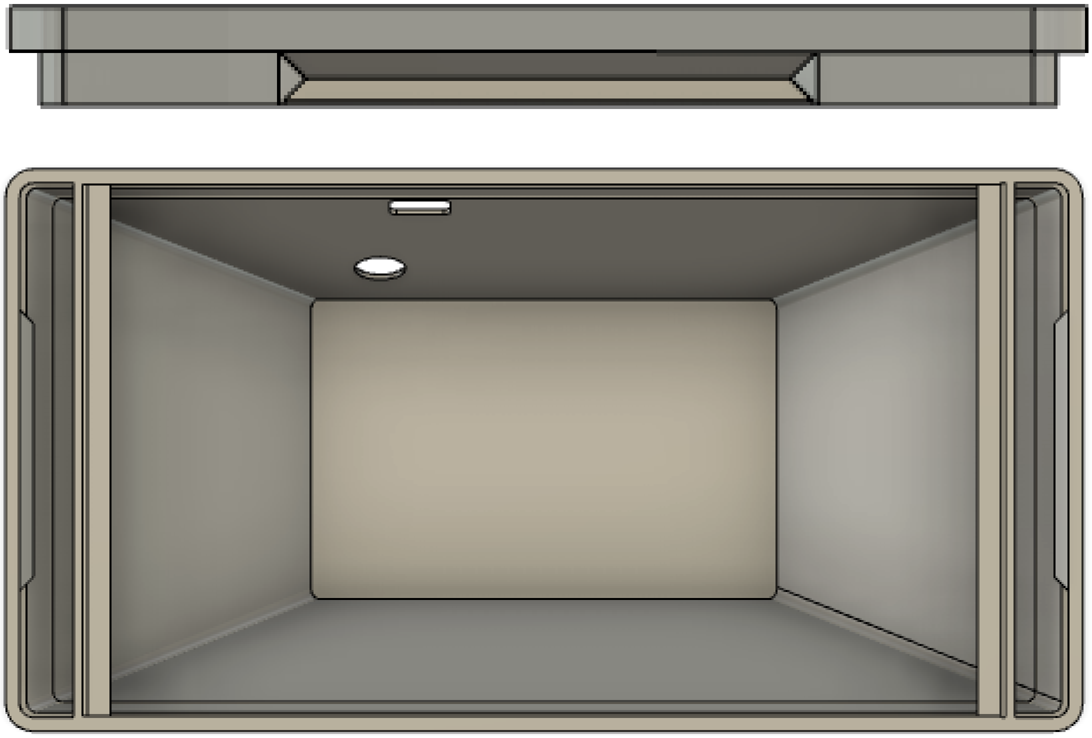


Figure 16. Various views of both the lid and bottom portion of the case

V.C.III. Doctor and Administrator Website

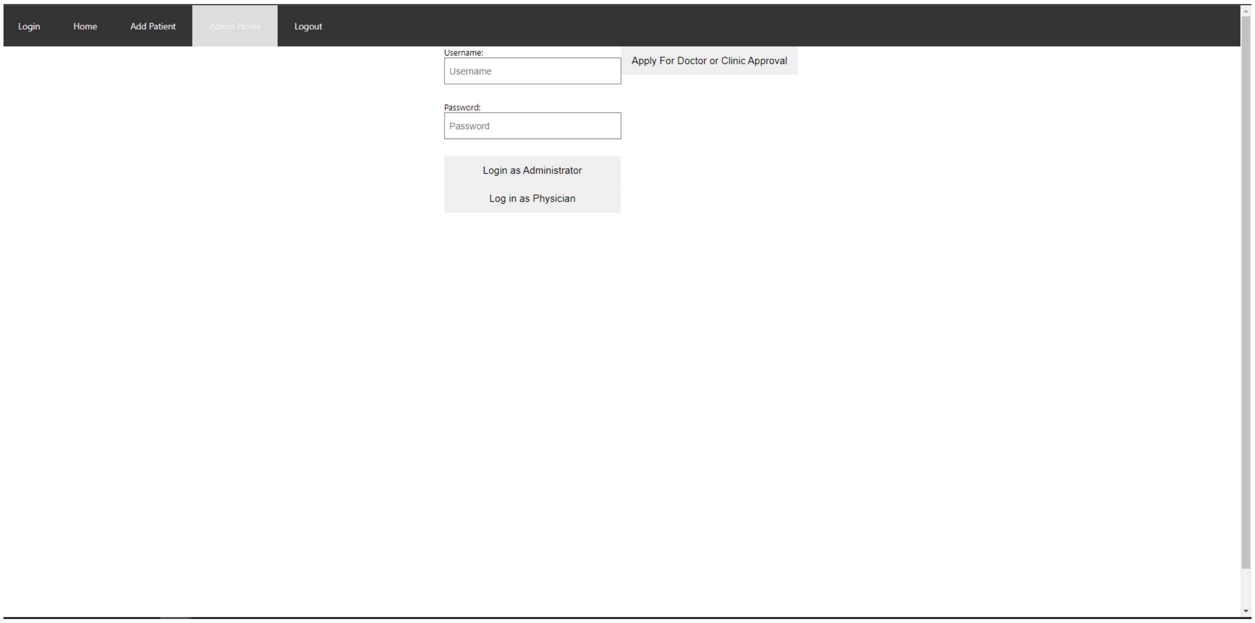
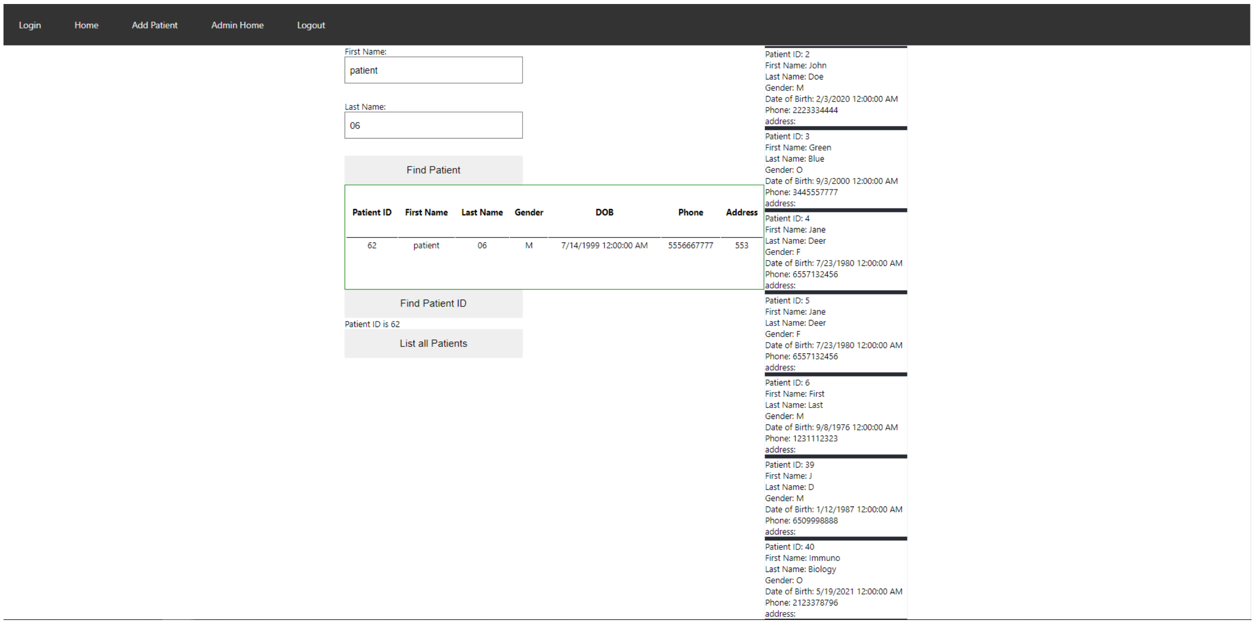


Figure 19. User Login Page as well as Clinic/Doctor Application Page



First Name: patient

Last Name: 06

Patient ID	First Name	Last Name	Gender	DOB	Phone	Address
62	patient	06	M	7/14/1999 12:00:00 AM	5556667777	553

Patient ID is 62

Patient ID: 2
First Name: John
Last Name: Doe
Gender: M
Date of Birth: 2/3/2020 12:00:00 AM
Phone: 2223334444
address:

Patient ID: 3
First Name: Green
Last Name: Blue
Gender: O
Date of Birth: 9/3/2000 12:00:00 AM
Phone: 3445557777
address:

Patient ID: 4
First Name: Jane
Last Name: Deer
Gender: F
Date of Birth: 7/23/1980 12:00:00 AM
Phone: 6557132456
address:

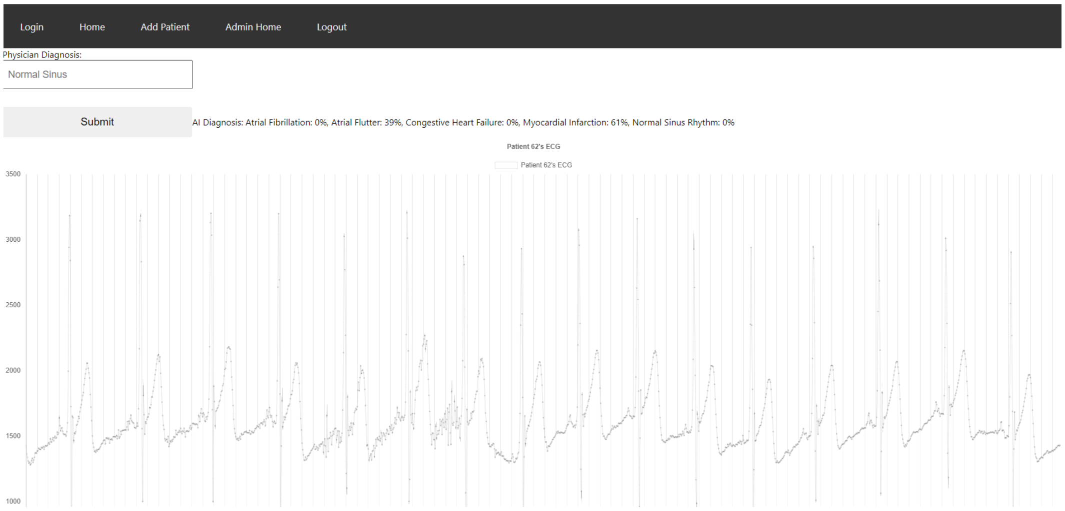
Patient ID: 5
First Name: Jane
Last Name: Deer
Gender: F
Date of Birth: 7/23/1980 12:00:00 AM
Phone: 6557132456
address:

Patient ID: 6
First Name: First
Last Name: Last
Gender: M
Date of Birth: 9/8/1976 12:00:00 AM
Phone: 1231112323
address:

Patient ID: 39
First Name: J
Last Name: D
Gender: M
Date of Birth: 1/12/1987 12:00:00 AM
Phone: 6509998888
address:

Patient ID: 40
First Name: Immuno
Last Name: Biology
Gender: O
Date of Birth: 5/19/2021 12:00:00 AM
Phone: 2123378796
address:

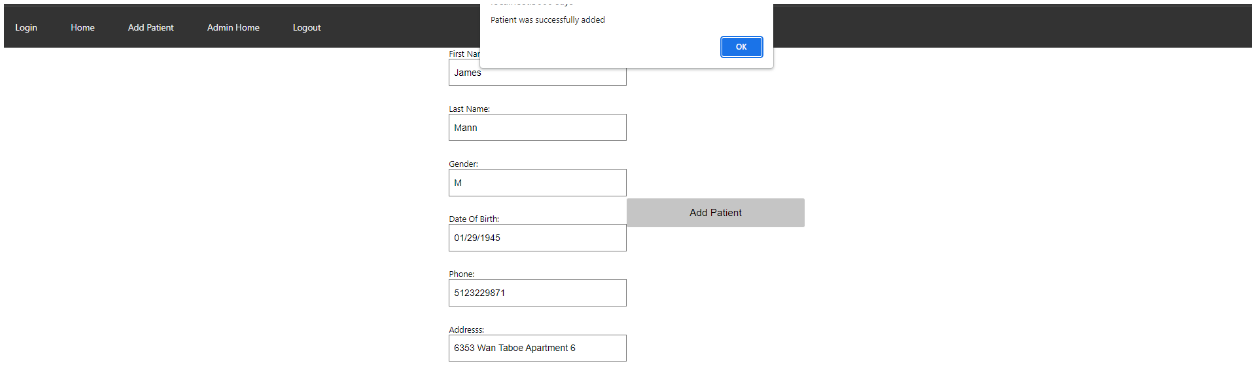
Figure 20. Website Patient Lookup Page



Physician Diagnosis: Normal Sinus

AI Diagnosis: Atrial Fibrillation: 0%, Atrial Flutter: 39%, Congestive Heart Failure: 0%, Myocardial Infarction: 61%, Normal Sinus Rhythm: 0%

Figure 21. Patient ECG viewing, machine learning output, and doctoral diagnosis page



Patient was successfully added

First Nar
James

Last Name:
Mann

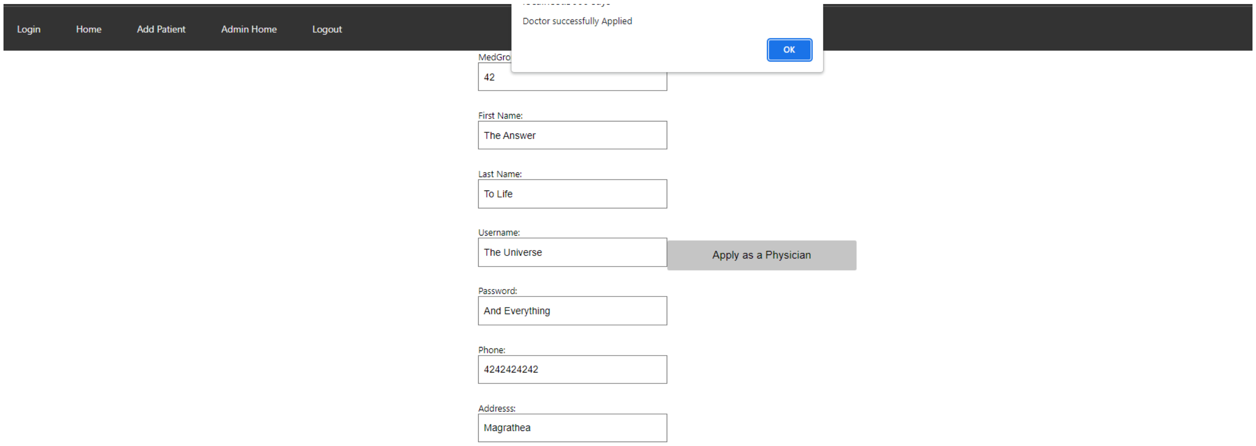
Gender:
M

Date Of Birth:
01/29/1945

Phone:
5123229871

Addresss:
6353 Wan Taboe Apartment 6

Figure 22. Patient addition screen with successful alert



Doctor successfully Applied

MedGro
42

First Name:
The Answer

Last Name:
To Life

Username:
The Universe

Password:
And Everything

Phone:
4242424242

Addresss:
Magrathea

Figure 23. Doctor application screen with successful alert

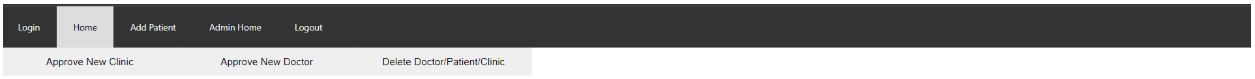
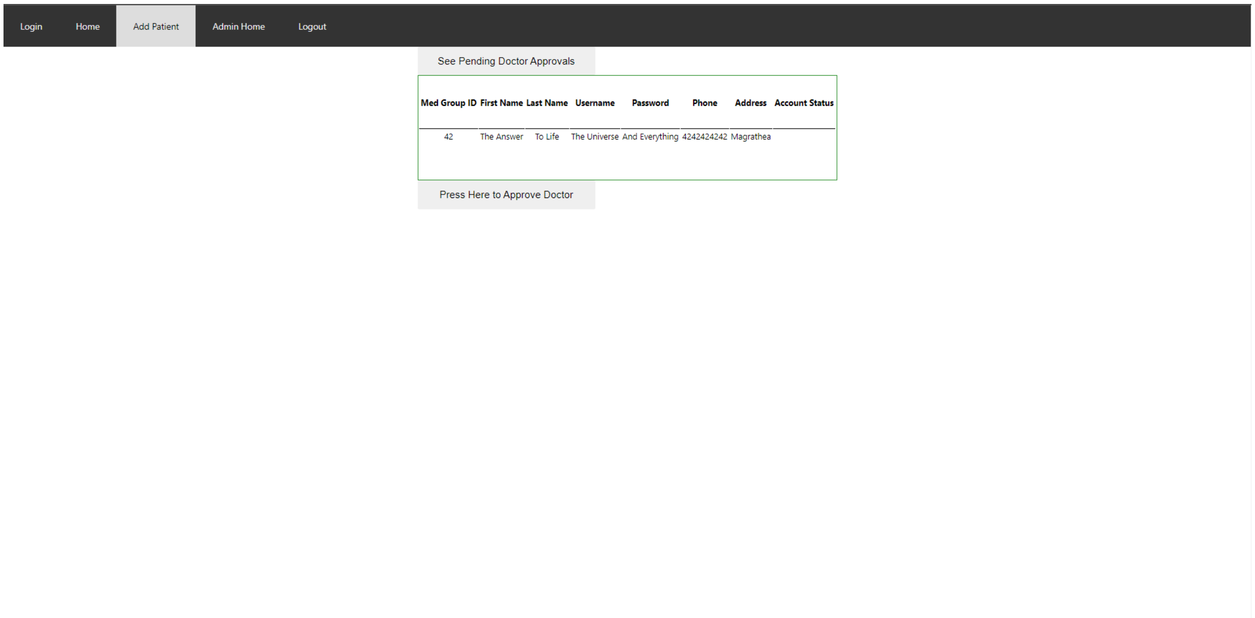


Figure 24. Administrator home



Med Group ID	First Name	Last Name	Username	Password	Phone	Address	Account Status
42	The Answer	To Life	The Universe	And Everything	4242424242	Magrathea	

Figure 25. Doctor approval screen.

V.D. Database Relational Schema

```
CREATE TABLE ECG_Reading(
        ecg_id SERIAL PRIMARY KEY,
        reading integer[] NOT NULL,
        ml_res integer[],
        doctor_opinion VARCHAR(500),
        date_time_taken TIMESTAMP WITH TIME ZONE NOT NULL)
```

```
CREATE TABLE Admin(
        username VARCHAR(20) PRIMARY KEY NOT NULL,
        password VARCHAR(20) NOT NULL,
        name VARCHAR(40))

CREATE TABLE Doctor(
        medgroupid INTEGER NOT NULL,
        first_name VARCHAR(20) NOT NULL,
        last_name VARCHAR(30) NOT NULL,
        username VARCHAR(20),
        password VARCHAR(20),
        phone CHAR(15) NOT NULL,
        address CHAR(200),
        acct_status VARCHAR(10),
        PRIMARY KEY (username))

CREATE TABLE Clinic(
        clinic_id SERIAL PRIMARY KEY,
        clinic_name CHAR(50) NOT NULL,
        phone CHAR(15) NOT NULL,
        address CHAR(200) NOT NULL,
        acct_status VARCHAR(10))

CREATE TABLE Patient(
        patient_id SERIAL PRIMARY KEY,
        first_name VARCHAR(20) NOT NULL,
        last_name VARCHAR(30) NOT NULL,
        gender CHAR(1),
        dob DATE NOT NULL,
        phone CHAR(15),
        address CHAR(200))

CREATE TABLE TakeMeasurement(
        patient_id integer REFERENCES Patient(patient_id) ON DELETE CASCADE,
        ecg_id integer PRIMARY KEY REFERENCES ECG_Reading(ecg_id) ON DELETE CASCADE,
        clinic_id integer REFERENCES Clinic(clinic_id))

CREATE TABLE Analyzed(
        ecg_id integer PRIMARY KEY REFERENCES ECG_Reading(ecg_id) ON DELETE CASCADE,
        doc_username VARCHAR(20) REFERENCES Doctor(username) ON DELETE CASCADE)

CREATE TABLE Doctor_Patient(
        patient_id integer REFERENCES Patient(patient_id) ON DELETE CASCADE,
        doc_username VARCHAR(20) REFERENCES Doctor(username) ON DELETE CASCADE,
        PRIMARY KEY (patient_id))
```

V.E. Bill of Materials

Table 2. Bill of All Purchased Materials Used in Final Prototype (Excludes 3D and PCB Printing Materials Used in the Makerspace)

Part Name	Manufacturer	Source	Part Description
Raspberry Pi 4 Model B	Raspberry Pi	CanaKit	Raspberry Pi Microcomputer with AI Capabilities
AD8232	SparkFun	Digi-Key	Single Lead Heart Rate Monitor
MCP3008	Microchip Technology	Adafruit	8-Channel 10-Bit ADC with SPI
Sensor Cable - Electrode Pads (3)	SparkFun	SparkFun	Tricolor Sensor Cables with Disposable Electrodes
Raspberry Pi 7" Touchscreen LCD	Raspberry Pi	CanaKit	800 x 480 Touchscreen Display with Adapter Board

V.F. Testing Data

V.F.I. Artificial Intelligence Findings for 15 Tested Individuals

Table 3. Percent Likelihoods Exceeding 10% of Heart Conditions as Diagnosed by the Artificial Intelligence for Each Test

Test Subject	AI Result
1	93% Atrial Fibrillation
2	96% Myocardial Infarction
3	100% Atrial Fibrillation
4	99% Atrial Fibrillation
5	35% Atrial Fibrillation 62% Myocardial Infarction
6	39% Atrial Flutter 61% Myocardial Infarction
7	74% Atrial Fibrillation 25% Normal Sinus Rhythm
8	51% Atrial Fibrillation 48% Congestive Heart Failure
9	85% Atrial Fibrillation

	15% Normal Sinus Rhythm
10	93% Atrial Fibrillation
11	91% Myocardial Infarction
12	99% Myocardial Infarction
13	88% Atrial Fibrillation 10% Myocardial Infarction
14	67% Atrial Fibrillation 32% Normal Sinus Rhythm
15	100% Atrial Fibrillation

V.G. Artificial Intelligence Training Datasets

- MIT-BIH Normal Sinus Rhythm Database [8]: Normal Sinus Rhythm
- BIDMC Congestive Heart Failure Database [8][9]: Congestive Heart Failure
- MIT-BIH Atrial Fibrillation Database [8][10]: Atrial Fibrillation
- PTB-XL [8][11][12]: Myocardial Infarction, Atrial Flutter, Atrial Fibrillation, Normal Sinus Rhythm

VI. Bibliography (IEEE)

[1] Centers for Disease Control and Prevention, "Heart Disease Facts," Centers for Disease Control and Prevention, Feb. 7, 2022. [Online]. Available: https://www.cdc.gov/heartdisease/facts.htm. [Accessed: Apr. 23, 2022].

[2] Centers for Disease Control and Prevention, "Atrial Fibrillation," Centers for Disease Control and Prevention, Sept. 27, 2021. [Online]. Available: https://www.cdc.gov/heartdisease/atrial_fibrillation.htm. [Accessed: Apr. 23, 2022]].

[3] IBM, "What are Convolutional Neural Networks?" IBM Cloud Learn Hub, Oct. 20, 2020. [Online] Available: https://www.ibm.com/cloud/learn/convolutional-neural-networks [Accessed: Apr. 30, 2022].

[4] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks - the ELI5 Way," Towards Data Science, Dec. 15, 2018. [Online]. Available: https://towardsdatascience.com/. [Accessed Apr. 30, 2022].

[5] R. Alake, "Deep Learning: GoogLeNet Explained," Towards Data Science, Dec. 22, 2020. [Online]. Available: https://towardsdatascience.com/. [Accessed Apr. 30, 2022].

[6] Mathworks, Classify Times Series Using Wavelet Analysis and Deep Learning. [Online]. Available: https://www.mathworks.com/help/. [Accessed: Jan. 20, 2022].

[7] M. Chhabra and M. Kalsi, "Real Time ECG monitoring system based on Internet of Things (IoT)," vol. 7, pp. 547-550, Aug. 2017. [Figure]. Available: ResearchGate, www.researchgate.net. [Accessed: Apr. 30, 2022].

[8] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P.C. Ivanov, R. Mark, ... and H.E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," Circulation, vol. 101, no. 23, pp. E215–e220, 2000. [Online]
[9] D.S. Baim, W.S. Colucci, E.S. Monrad, H.S. Smith, R.F. Wright, A. Lanoue, D.F. Gauthier, B.J. Ransil, W. Grossman, E. Braunwald, "Survival of patients with severe congestive heart failure treated with oral milrinone," J American College of Cardiology, vol. 7, no. 3, pp. 661-670, Mar. 1986.
[10] G.B. Moody and R.G. Mark, "A new method for detecting atrial fibrillation using R-R intervals," Computers in Cardiology, vol. 10 pp. 227-230, 1983.
[11] P. Wagner, N. Strodthoff, R. Bousseljot, W. Samek, & T. Schaeffter, PTB-XL, a large publicly available electrocardiography dataset, vol. 1.0.1, PhysioNet, 2020. [Online]. Available: https://doi.org/10.13026/x4td-x982.
[12] P. Wagner, N. Strodthoff, R. Bousseljot, D. Kreiseler, F.I. Lunze, W. Samek, T. Schaeffter, PTB-XL: A Large Publicly Available ECG Dataset, Scientific Data, 2020. [Online]. Available: https://doi.org/10.1038/s41597-020-0495-6