Faculty of Engineering and Applied Science

ENGR 4940U Capstone Systems Design for ECSE I

Remote and Mobile Healthcare System for Home Care

R2: Concept Generation, Conceptual Design, Prototype Report

Team Members:

Student Name	Student ID
Mamun Hossain	100553073
Gobikah Balaruban	100742539
Manreet Kaur	100766207
Aaditya Rajput	100622434

Faculty Advisor: Dr. Mikael Eklund

Capstone Coordinator (Fall 2022): Dr. Vijay Sood

Table of Contents

1. Concept Generation and Analysis **3**
1.1 Understanding the Problem 3
1.2 Existing Solutions 4
1.2.1 Chronisense Medical 4
1.2.2 Telehealth 5
1.2.3 Wearable Devices (eg. Apple watch, Fitbit devices, Samsung watches) 5
1.2.4 Remote Mobile Healthcare Concept Analysis 5
Table 1-1: Component Design 7
1.3 Stakeholder Analysis 9
Table 1-2: Stakeholder Analysis 10
2. Conceptual System Design **11**
Figure 2-1: Overall System Design 11
2.1 Component Level Design 11
Figure 2-2: Component Diagram of the Overall System 12
Table 2-1: Modules 12
2.2 Overall System Design 14
Figure 2-3: Block Diagram of the System 14
2.3 State Diagram 15
Figure 2-4: State Diagram for Heart Rate Sensor 15
2.4 Activity Diagram 16
Figure 2-5: Activity Diagram for Overall System 17
2.5 Engineering Requirements 17
Table 2-3: Engineering Requirements 17
3. Definition of Integration Tests **19**
Table 3-1: Definition of Integration Tests 19
4. Estimated Cost of the Project **20**
Table 4-1: Estimated Cost of the Project 21
5. Updated Project Plan **22**
Figure 5-1: Updated Project Plan 23
6. Contribution Matrix **23**
Table 6-1: Contribution Matrix 23
References **24**

1. Concept Generation and Analysis

1.1 Understanding the Problem

In our Capstone Project Report 1, the problem to be solved was identified as: The development of a hardware and software system that allowed for a remote patient monitoring service that will be created to aid chronically ill patients to have the ability to have their health monitored to decrease health risks by monitoring different health conditions (eg. heart beats per minute, blood volume changes, blood oxygen levels, blood pressure waveform, etc).

To further clarify this problem, due to the previous burden on healthcare facilities globally, which as a result ended up in population anxiety and isolation due to the COVID-19 pandemic, many dietary habits and physical habits were altered. Many people were unable to make visits to their doctor in person due to the pandemic and were unable to monitor their health status as a result. In order to prevent this lack of health monitoring data in the future, we have conducted the required research into different existing solutions to understand and verify the requirements on how to approach the project requirements definition and the development of a well-formed prototype.

To develop a hardware system that interacts with the software based system, we have identified the important requirements to allow for the development of our mobile health care device. Certain requirements will be implemented (i.e., storing the different data from the sensors appropriately), and the calculation of the data (i.e., calculating waveforms, beats per minute, blood oxygen levels) and the potential real-time monitoring of these events.

The main intent of this project is to allow the user to use various health monitoring sensors, such as ECG sensor, a heart rate sensor, a blood oxygen level sensor, and a way to collect the data from those sensors and display them in either a numerical or graphical format. The information would then be stored in a server that is passed through a cloud server.

This section will define and analyze the various components that would be required for a Remote and Mobile Healthcare System for Home Care on a conceptual basis and how they come together to form a cohesive system.

1.2 Existing Solutions

Remote and Mobile Healthcare Systems have been deployed by many industries amongst both the commercial industry and the healthcare industry for several different purposes. From a basic level purpose to an enterprise level requirement, Remote and Mobile Healthcare systems have been continuously evolving. Industries such as the telecommunications industry and the healthcare industry are using remote and mobile healthcare systems as a way to monitor both lifestyle habits as well as the different health risks that can be imposed on an individual. A typical remote healthcare system will consist of several different electrical components, which in turn will connect to the backend server via the cloud network that will communicate with the system as well. The typical aspects of a system of this size would involve the sensor actuation reading the inputs, sending that data to the system and then the distribution of that data to the server. Listed below are some alternative solutions to the aforementioned product at a systems level.

1.2.1 Chronisense Medical

A medical company in charge of the development of medical-grade devices that allow the monitoring of vital signs and other health factors which allow the patients and healthcare professionals to manage chronic disease without interfering with the patient’s everyday activity in or outside their home [1].

The way this works is with the usage of their toolset that measures medical data, they are able to take in different output values, such as ECG and blood oxygen levels, and it can communicate with the system in real time. They use a specific sensor called the “Radial Artery Sensor” which gives live time accurate vital sign data in the format they find is needed for the reading, be it waveform or in numerical form. It is a patient friendly device that is wearable on the wrist, like a watch that presses up against the radial artery to retrieve the patient data and send it over the server.

1.2.2 Telehealth

With the requirement of digital information and communication technology, Telehealth has created a way to allow the access of healthcare services and being able to remotely manage the patient's healthcare from a remote distance. The way this works is that a provider will issue the patient with devices that wirelessly send data, such as blood pressure, blood sugar, and oxygen levels. They are able to do this by using web-based or mobile-based applications that can either update the data from the specific monitor in real-time or have the patient manually input the data, such as dietary logs throughout the day. They also have devices that can record data such as heart rate or even data that detects activities such as falls for dementia patients [2].

Unlike Chronisense however, the application is not as small as a wearable watch-like device, but a larger physical device that will act on behalf of the provider's instructions, such as inputting blood pressure readings into the app manually, as Telehealth is catered to those with limited access to resources.

1.2.3 Wearable Devices (eg. Apple watch, Fitbit devices, Samsung watches)

With wearable devices taking the world by storm in the past decade, the applications of wearable technology have significantly evolved as well. From the initial phases of simply taking calls to being able to monitor blood oxygen levels, taking ECGs, as well as fall detection, wearable tech has become a viable form of a mobile healthcare system. With several sensors in place that sends data from the end user, these can be blood pressure monitors that send numerical data and waveform data, ECG data that sends waveform data, or different types of biosensors, which collect data that will be sent as user data over to the mobile device that is connected to the cloud for easy and in depth viewing of the data. This provides both a hardware and software interface for the required needs and monitoring purposes [3].

1.2.4 Remote Mobile Healthcare Concept Analysis

The remote mobile healthcare system will have two major components that will need to follow suit to be on par with the aforementioned components. In contrast to the other components, where the end user will have constant access to their own data, we will have major data be sent via a cloud server that can be accessed by the user to reduce anxiety for the patient, and they will be able to view the data as much as they deem is necessary.

Comparisons:

We had discussed several different systems, beginning with using a system that incorporated an Arduino system that can interact with a portal based device with an IoT aspect. The way that this would work is that an android application would be created that would have an implementation of a WiFi module that will connect to an ECG module that will be available to access wirelessly to the user using the Arduino cloud program within the IDE. With this system in place, the user will be able to access their information on a waveform every time that it is updated to the application. It will use the analog pins on the Arduino while having the ECG waveforms be sent over via Bluetooth using a serial interface. The data that is accessed here will be able to be viewed by anyone using a phone or web based application with the URL provided from the system.

A similar application that we had conducted research upon was to create a system using the Raspberry pi module and create a web based application that will allow the user to send the required input such as ECG signals whilst using a 3-lead ECG sensor. This in turn will allow for an ECG monitoring system that will be able to convert an analog signal that will be converted to a digital signal that can be sent to the embedded Raspberry Pi. The way this will work is that using an online plotting tool (ie. Plot.ly), the signal can be viewed if anyone has the programmers username and password for the mentioned online tool. The real time data will be monitored and the user will be able to see if there is any abnormality reached in the ECG signals.

A third system that we had researched about would be the incorporation of both the Arduino module as well as the Raspberry pi module. The way that this would work is that beginning with the Raspberry Pi, there will be an interface with the 3-lead ECG sensor which will be able to sample the data at the given rate we desire in our web based application. From there, the analog to digital conversions will occur for the sample rate in Hz. After that, we will be able to view the peak detection of the ECG signal in our web based application that will display the information and it will upload the analysis to the cloud. Now, to display the actual peak detection, this is where we will be incorporating the Arduino that will be able to chart the single data and give us the waveform chart, and with the waveform chart, we will also be able to create a square waveform chart as well, allowing us to have multiple readings of different requirements.

The application will be linked from the sensor devices to the IoT devices and will then relay the detailed information with the appropriate units of measurement that will wirelessly send the data over a cloud server and store the information there. In addition to this, the user will have access to some minimal data, such as beats per minute. That will give some sense of stability to both the user to ensure that the device is functioning and to the server that is collecting the data to confirm that the product is being used correctly.

With the applications in mind and the systems to narrow down, we will be using the third system as we feel that when needing to interact with the cloud server, it is ideal to use the Linux based kernel to allow access to that server seamlessly, and allow us to connect different sensors directly to the board, without needing to have any extra modules such as a Bluetooth or WiFi module as they are already integrated into the board, whilst the Arduino also will be able to assist with plotting the data into our own web server rather than needing to use a general web based application such as plotGraph.oy for example so that way the information that we record will remain confidential to the user that is using and there is no risk of any third party application taking that information.

The overall intent when creating the concept generation of the device with the cloud server is to prioritize the user friendliness and make the device with as many sensors as possible to relay as much data as possible to monitor the patients' health and ensure that they are living a healthy lifestyle, whilst monitoring the progression of their health statistics as it is being stored to the cloud server statically. This will be done through the implementation of the server that will be created through the IoT devices, such as the Raspberry Pi and Arduino Mega unit that will connect to many different sensors via the GPIO pins on the circuit board. The more sensors that the system will contain, the more types of data the system will be able to display and store. The application will consist of different screens displaying different data sources and the static data accumulated over a period of time. It should also be noted that all the connections will be done wirelessly over an unspecified cloud server. The table below displays a comprehensive list of the various sensors and devices that the application will contain. The server will be using a Raspberry Pi and Arduino system to control the data and send it back to the server via a currently undefined cloud platform. However, we have gone down the route of integrating both the Arduino unit and the Raspberry Pi because with the Arduino unit that will be connecting the sensors, and the Raspberry Pi that will be hosting the cloud server, it will leave our system to be very scalable in its design whilst having a reliable cloud server. This can be compared to an existing solution such as they have their reliable cloud servers that will store their data and be able to look at the information very clearly and not have any discrepancies in the data collection, however not only will be collecting the main data, but unlike the other solutions we will be continuously updating the data the second the user wears it rather than having to enable a certain setting to use it (eg. Apple watch “Start workout” method). Doing this will ensure that we have an accurate real time response to the data.

Table 1-1: Component Design

Component Name	Description	Justification
Server Unit (Raspberry Pi Model B)	The Raspberry Pi Model B will be the Server computer that will be the brain of the system. It will be connected to the entire system via the provided GPIO pins. It will be accessible as a User Interface (UI) server system that will have sensors connected to it via the GPIO pins and using a MQTT process will provide information receiving the patient’s electrocardiogram signals and the end user’s heartbeat. It is also responsible for storing the patient’s information that can be later accessed by the administrator and the health care professionals who can override the administrator commands to view the stored data.	Will be able to send the information via the cloud server accessing the Raspberry Pi by enabling SSH features and connecting to IoT hub with Node.js and using the GPIO pins in unison with the various sensors
Sensor Unit (Arduino Mega)	The Arduino will act as a high powered computer that will be able to connect to the server via SSH protocol and send the data retrieved from the device to the cloud server.	Will be connecting all the sensors to the Arduino Mega unit to seamlessly power the sensors and retrieve the user data that sends it back to the

	It will be accessible via a User Interface (UI) server system that will have the cloud script be running within the Arduino IDE to send information from the device to the allocated cloud server receiving the users ECG reading, PPG, BPM, and blood oxygen readings. It will also be responsible for withholding the user information that can later be accessed by a healthcare professional that will have administrator rights to the system to view the stored data.	Server Unit.
Display Unit	The display unit is responsible for showing that the device is functioning. The time and battery life will be displayed to the end user.	Screen will serve as a main menu to display basic statistics and ensure functionality.
ECG Unit (Pulse sensor (Heart rate sensor))	The heart rate monitor will act as a way to monitor and receive the information from the end user, by accessing the Raspberry Pi's GPIO pins. It will keep track of the heart rate fluctuation.	The sensor is able to send the beats per minute data information to show the user on the display unit and to store in the server.
End User Unit (BPM Reader)	The user will be able to access their BPM.	Connects to the display unit to ease anxiety.
Monitoring System Unit	The Monitoring System Unit will be where all the information and the server will have all its main access to. It will be connected to every component in the system. It will be responsible to store all the end user information as required by the individual.	Will allow for ease of access monitoring of the system data to ensure things are progressing and processing smoothly in a timely manner.
PPG Unit (Pulse Sensor + MCP3008 ADC IC)	The PPG unit will retrieve the data from the pulse sensor and pass it along to the MCP3008 via an analog signal and then be able to transfer that information of the	The sensor sends pulse data to the cloud server and not shared to the user, displaying the data via a waveform chart.

	PPG to the number of Hz that is being transmitted.	
Oximeter Unit (MAX30100)	The oximeter unit will be able to monitor oxygen level and heart rate that will have its information stored and assessed by the healthcare professional to check for any abnormalities in the reading that gets stored into the back end server.	Sensor to monitor blood oxygen levels to increase functionality of hardware systems.

Using the table above, with detailed planning and taking in account the time and resources we have available to complete the project, we have decided to take select the usage of the Raspberry Pi to be able to connect into the cloud server much more efficiently whilst also use lower power consumption that will in turn increase the battery life of the system and result in longer data retrieval so there is more data that can be analyzed for the lifestyle and health standards of the patient.

1.3 Stakeholder Analysis

For the proposed system design, the various stakeholders and associated requirements must be considered as shown in the table below.

Table 1-2: Stakeholder Analysis

Stakeholder	Stakeholder Requirements (Non-functional and functional requirements)
Project Team Members	- The project must be finished within the specified time constraint from September 2022 to March 2023 - The project must be completed within the allocated budget specified by the coordinator, which has been budgeted to $800 - The project must have a unified approach where all the team members are able to identify the development and be confident in the final integration

	- The project must be completed with a high level of solution for the task at hand with an adequate grasp of the technologies used
Faculty Advisor and Course Coordinator	- The project will provide a prototype during the development period (December 2022) as well as a completed prototype by the end of the development period (March 2023) - The project must be completed within the allocated time and budget as mentioned previously
Client/Patient	- The completed prototype must be flexible to allow the user to be able to comfortably use the system - The completed prototype must be able to consume the allocated real time data that will be displayed in correspondence with the performance requirements, which includes being able to use the display unit to view the beats per minute as it gets updated at the specified time

2. Conceptual System Design

In this section, the conceptual system design covers the several concepts generated in the previous section, concept generation and analysis. The individual components will be further specified, including their responsibilities and interfaces. The behavior, structure, parameters of the components are specified in this section. The operations of the components and how they interact with each other will be illustrated with the diagrams below.

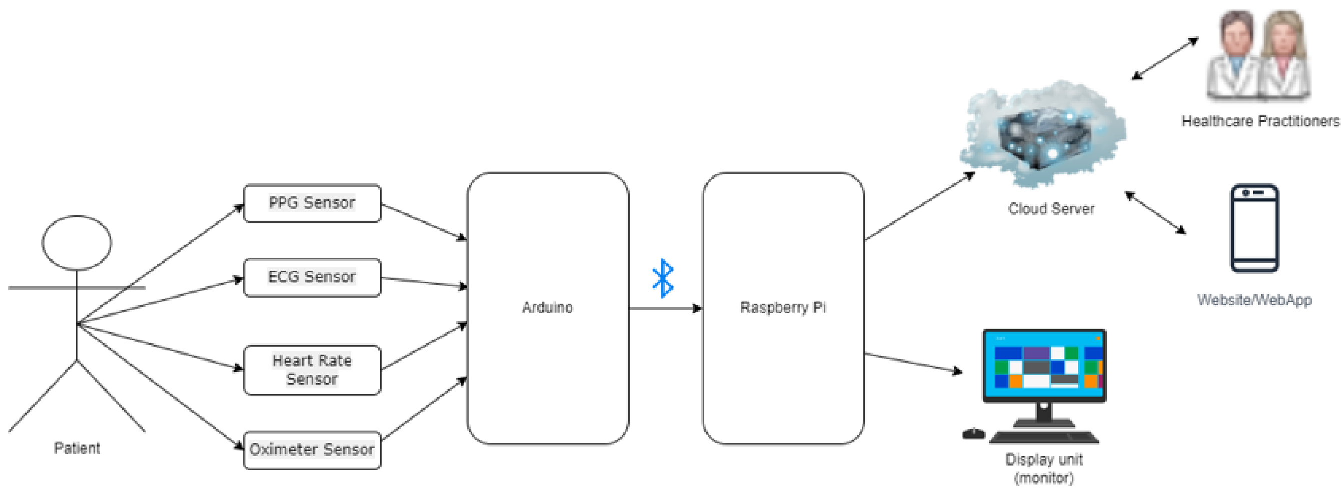

Figure 2-1: Overall System Design

In the above figure, it illustrates the overall system design and how the components interact with each other. Starting from the left, the user has their vitals inputted by the sensors, which is connected to the Arduino. For the Arduino to connect to the Raspberry Pi, bluetooth is used. From there, the display of the diagnostics is shown on the Display Unit, while the data is sent to the cloud server. From the cloud server, the data can be accessed and retrieved by healthcare practitioners and/or by the user accessing it from the website/mobile application.

2.1 Component Level Design

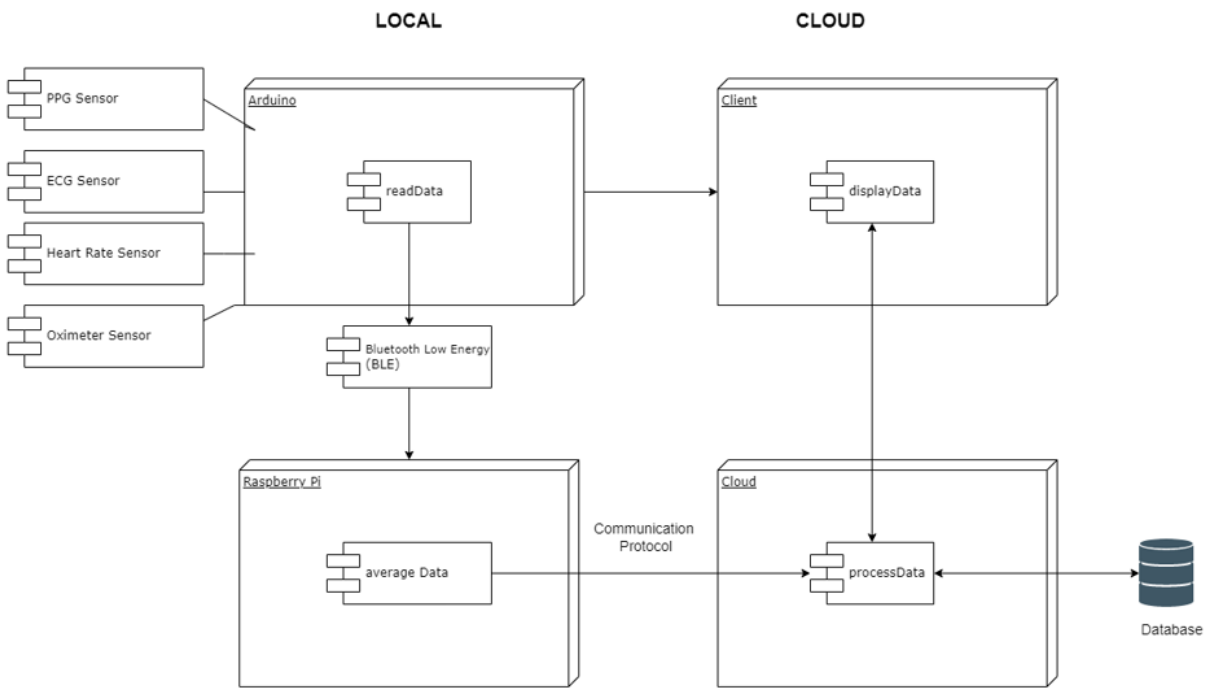

Figure 2-2: Component Diagram of the Overall System

The diagram in Figure 2-2 above continues with the overall system components and displays the interfaces of the components. It also shows how the various components connect with each other too. For instance, in the Arduino block, there is a readData module that reads the data from the sensors that are linked to the block on the diagram. As this diagram shows interactions between components, it also shows how the local components are connected to the cloud components. The Raspberry Pi is able to connect to the cloud via a Communication Protocol. The modules are further explained in the table below:

Table 2-1: Modules

Module	**Description**
readData	Arduino reads the data from the sensors periodically(for example in every 2 mins)

	and then concatenates them into strings . These strings are then encoded into a char array and sent to the Raspberry Pi using a bluetooth module(BLE).
averageData	Raspberry Pi then reads the data from the bluetooth module by decoding the array . The data is then added into an average function which gives the avg data (or average readings of the data). Then this averageData will be transmitted to the cloud server using REST/MQTT communication protocol.
processData	This module will be responsible for storing all the data received from the raspberry pi to the database. This module will then send data to the displayData in every 10s. Apart from this processData will be responsible for sending the requested data to the displayData(for example if the client requested the data for a particular day or time.
displayData	This will be a client facing app (like a webApp) which will display the data to the user. Patients and health practitioners will have their own login id’s . There will be graphs of each sensor readings(for example ECG graph) and the current data readings .

2.2 Overall System Design

The Block Diagram shows how the components work together as a whole unified system. The diagram below illustrates how and where they are connected, starting from the inputs coming in from the left and the outputs on the right. Starting from the left hand size, the Arduino receives the data collected from the various sensors. The Arduino then transfers this data to the Raspberry Pi, which acts as a central hub for the data. The data can be sent to the website, which can be viewed by the user and/or healthcare professionals. It could also be sent to the digital display, for the user to view immediately. The digital display for Raspberry Pi may or may not be implemented into the system and is currently under review.

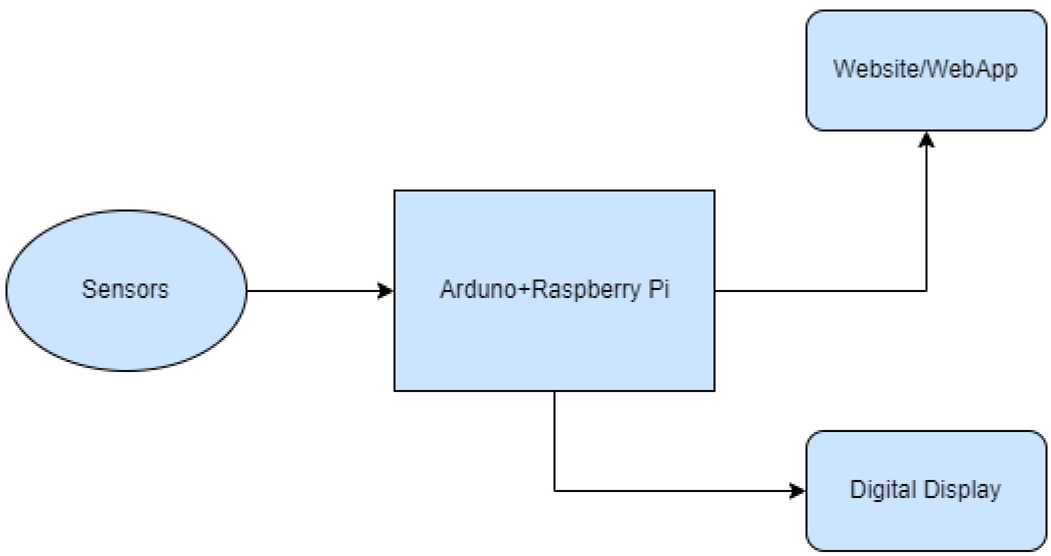

Figure 2-2: Block Diagram of the System

2.3 Activity Diagram

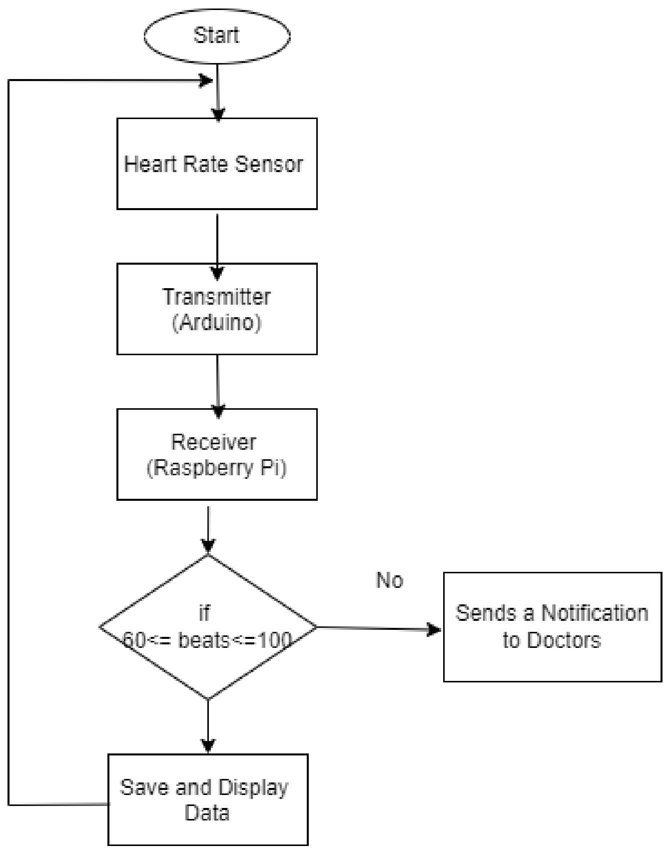

Figure 2-3: Activity Diagram for Heart Rate Sensor

The activity diagram above showcases one of the sensors in the monitoring system: the heart rate sensor. It begins with the user requesting a heart rate diagnostic and the diagram shows the steps that proceed with the request. One of the steps includes a condition placed with a range of an adult's normal heart rate. If the condition is not satisfied, a notification/alert is sent to doctors/users. If the heart rate goes beyond a certain limit, such as surpassing 120 BPM, a more serious alert is sent to the user and to healthcare authorities. The same case applies for an unusually low heart rate, which is abnormal.

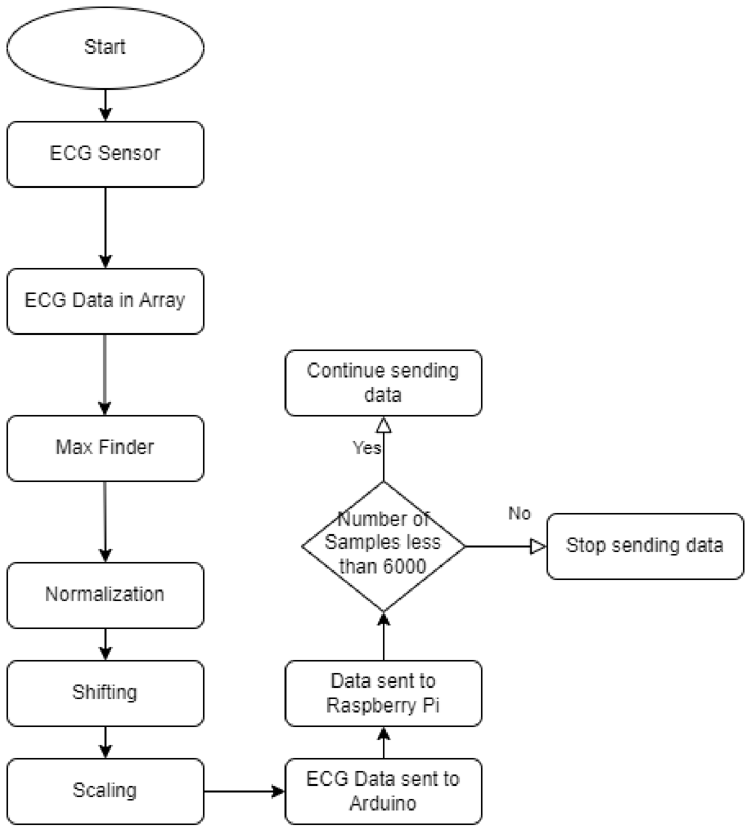

Figure 2-4: Activity Diagram for Electrocardiogram (ECG) Sensor

The activity diagram above shows the activity for the Electrocardiogram (ECG) Sensor. It starts off with the user starting the ECG diagnostic and the sensor collects the data. The collected data gets put into an array. Then, more data processing occurs with max finder, normalization, shifting, and scaling. The signal data and patterns are processed and the data is sent to the Arduino. It is then sent to the Raspberry Pi. If there are less than 6000 samples, the process will continue until it satisfies the condition.

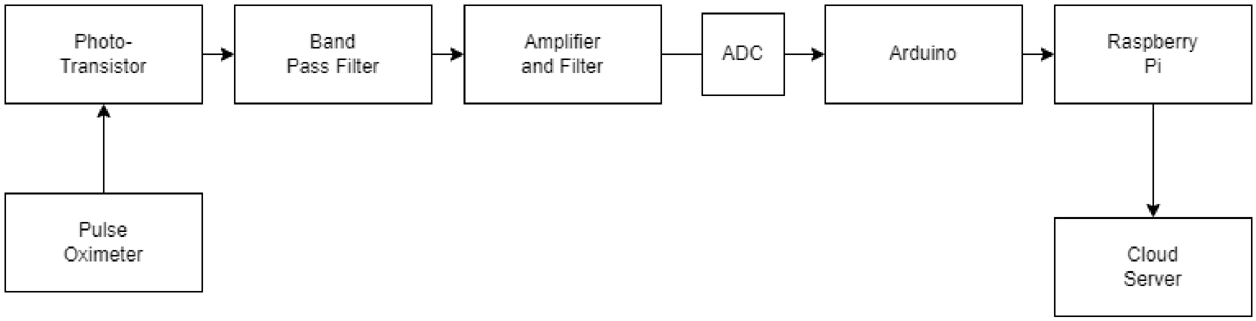

Figure 2-5: Block Diagram for Pulse Oximeter

The block diagram above shows the various components for the pulse oximeter. It consists of the Pulse Oximeter sensor and the various filters that are used. The data is then converted into digital from analog and then sent to the Arduino. The Arduino sends the filtered data to the Raspberry Pi and is finally sent to the Server.

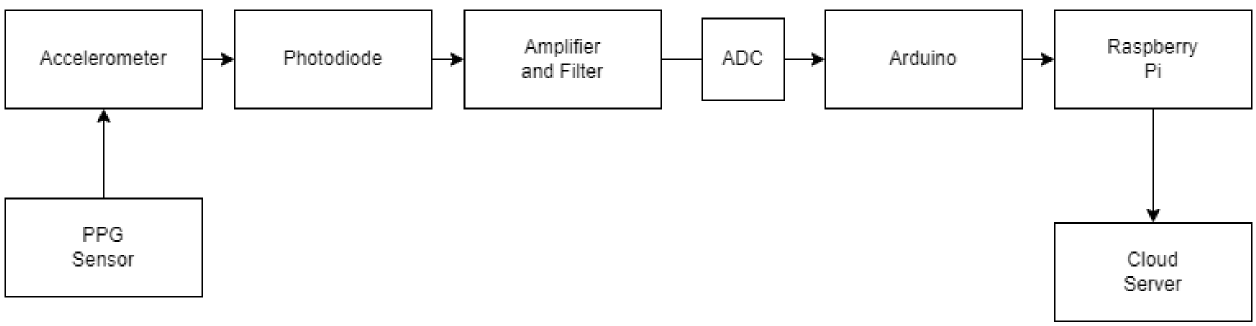

Figure 2-6: Block Diagram for PPG

Similarly to Pulse Oximeter, PPG follows the same process with similar components. It has the sensors, including the accelerometer and photodiode. The signals are then processed and then sent to Arduino, Raspberry Pi, and end up in the Cloud Server.

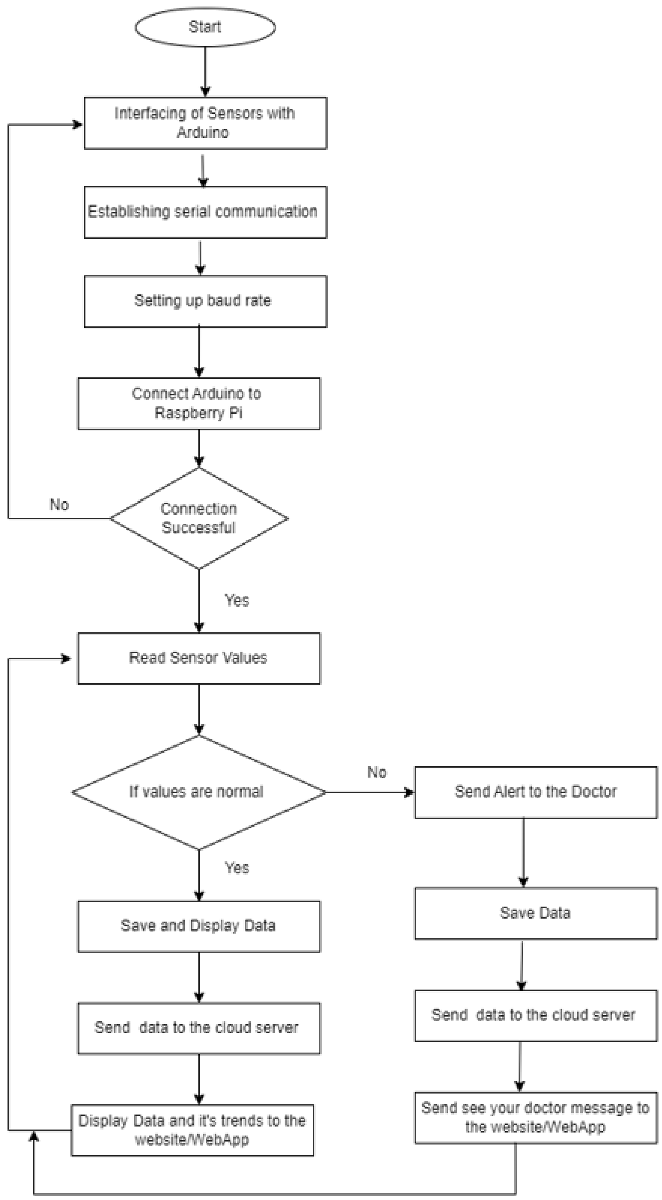

Figure 2-7: Activity Diagram for Overall System

The activity diagram above shows the overall system. It shows the steps and includes all the component units and the various situations that occur too. This includes the case where values are abnormal and the proceeding steps that occur. It also accounts for unsuccessful connections.

2.4 Engineering Requirements and Traceability Matrix

Table 2-5: Engineering Requirements

Req-ID	**Requirement**	**Category**	**Quality Attribute**
REQ-01	The mobile medical monitoring system must be able to perform diagnostics and collect heart rate, blood volume levels, blood oxygen levels, and blood pressure waveforms at 60Hz.	Functional Requirement	
REQ-02	The backend should handle all calculations of data (waveforms, conversions, etc).	Functional Requirement	
REQ-03	The application must be able to store the data from the sensors and provide a real-time analysis of the data accumulated within 3 seconds.	Non-Functional Requirement	Performance, Usability
REQ-04	If the heart rate monitor detects abnormalities ranging 40% higher or lower in heart rate, it should notify the user/doctor via a notification/email immediately and advise them to visit a physician.	Non-functional Requirement	Reliability

REQ-05	The system should run on a server which has a good performing GPU, and at least 16 GB of RAM to allow training and testing of the model done quickly.	Non-Functional Requirement	Efficiency, Stability
REQ-06	The system should handle multiple requests (multiple diagnostics like heart rate, blood volumes) from the user at the same time.	Non-Functional Requirement	Scalability, Efficiency
REQ-07	The monitor system must display the results of the diagnostics in an easy to understand way.	Functional Requirement	
REQ-08	The system should store previous data up on a cloud database server which allows storage capacities up to 10GB.	Non-Functional Requirement	Capacity

2.6 Test Cases

Table 2-6: Test Cases

Test Case ID	**Test Case Description**	**Test Case Steps**
TC1	Run the calculations with an input data set to test	1. Input a test case where the parameters are similar to what the sensors read

	backend calculations and the Display.	and insert them into the backend (Raspberry Pi) of the system. 2. Invoke the methods that run the calculations which measure for average heart rate, blood oxygen, and pulse. 3. Retrieve the output from the system on the Display.
TC2	Verify that sensors transmit to the Raspberry PI Server Unit.	1. User will have their vitals read by the sensors. 2. Data will be sent to Arduino locally and will be read via the readData module. The strings then are encoded into a char array and sent to the Raspberry Pi using bluetooth. 3. Data is then Decoded in the Raspberry PI. 4. Test method to display the decoded raw data that the sensors recorded.
TC3	Verify that the data communication to the cloud server is active.	1. The data that has been resulted from the calculations in the averageData Module is transmitted to the cloud server using REST/MQTT communication protocol. 2. The processData module will store this data to the Database. 3. Retrieve data from the database by querying for the data in the tables and validating the output.
TC4	Verify the stability of the system.	1. Send multiple requests to the system to detect how well it performs under heavy traffic. 2. Overload the database with a large dataset to test for the capacity as well as the reliability of the system when parsing a large database.
TC5	Verify that the System sends an alert if the values are varying by 40%.	1. Input a test data set and have a test database composed of average data in the past. 2. The test dataset will have data that is highly varying to the average data in the database.

		3. The input of the new “abnormal” dataset will validate the functionality of the notifying feature of the system.

Table 2-7: Traceability Matrix

REQ	01	02	03	04	05	06	07	08
TC1	**X**	**X**					**X**	
TC2			**X**		**X**			
TC3	**X**		**X**		**X**			
TC4					**X**	**X**		**X**
TC5				**X**				**X**

3. Definition of Integration Tests

The system has multiple components: Server Unit, Display Unit, ECG Unit, End User Unit, Monitoring System Unit, PPG Unit, Oximeter Unit, Photo Sensor Unit, Pulse Detector Unit and Math Unit.
A set of Integration Tests have been defined below, in Table 3-1, to test the components if they are working properly and if they are working together.

Table 3-1: Definition of Integration Tests

Integration Test Case ID	Integration Test Case Description	Expected Result
[IT-DATA-1]	The Server Unit is running and handling the data that is being stored and retrieved.	Users successfully log their data and are able to retrieve data.
[IT-DISP-1]	After performing diagnostics, the data is displayed to the user with real-time results.	Once the diagnostic is successfully completed, the results should be shown on

		display for the user to view.
[IT-DATA-2]	After a diagnostic has been completed, the data should be logged/stored to the server.	The connection to the server is successful. The data is sent to the server to be stored and can be accessed/retrieved later.
[IT-ALERT-1]	If the data collected from the diagnostic indicates any abnormalities with the user, it should alert the user immediately.	After a performed diagnostic that indicates any irregularities, the system should be able to detect that the data is abnormal and create an alert to be displayed for the user.
[IT-ANLYS-1]	The accumulated data collected can be used to create an analysis to provide the user insight on what the results indicate.	A real-time analysis of the diagnostic should be shown upon request by the user.
[IT-DATA-3]	Some of the data logged should not be accessible by the user, as it is meant for a physician to view.	This data is stored into the server but is hidden, so the user cannot access/retrieve the data.
[IT-DATA-4]	The “bad noise/inaccurate” data is collected by the sensor, providing inaccurate information.	The data is filtered out during or after the diagnostic to prevent inaccurate data from being recorded.
[IT-DISP-1]	The time and battery life is displayed to the user.	The device accurately shows the current time and battery life on the Display Unit.
[IT-CALC-1]	The Math Unit calculates the ECG waveform.	The Math Unit accurately performs a calculation to get the frequency (Hz) of the ECG waveform.

4. Estimated Cost of the Project

The estimated costs will include the materials needed for the system but will not include labor and build costs at this time. All the prices below are shown in CAD and will correspond in the pricing of the area we will be purchasing from within Canada. The prices listed below are estimates and do not reflect the final price.

Table 4-1: Estimated Cost of the Project

Component	**Component Description**	**Cost per unit ($ - CAD)**	**Physical Quantity**	**Total Cost (cost/unit * quantity) ($ - CAD)**
Raspberry Pi	The Raspberry Pi will be used to run and host a server and power the system and communicate with the display and cloud server for data collection and distribution	240	1	240
Arduino Mega	The Arduino will host all the sensors and connect to the Raspberry Pi and give us accurate readings	68	1	68
Display Unit (240x240, General 1.28inch Round LCD Display Module, 65K RGB)	The LCD Display Module will display the data the end user will have available to them	20.45		20.45

PPG Unit (Pulse Sensor + MCP3008 ADC IC)	The pulse sensor that will send the heartbeat via a waveform pattern and the MCP2008 ADC that will be sending the ECG data in Hz	33 + 6	1 + 1	33 + 6 = 39
Oximeter Unit (MAX30100)	The sensor that will determine the blood oxygen levels and transfer that data to the cloud	18	1	18
Developer man hours	The developer man hours for developing the project	Not added to the project cost	N/A	N/A
Total Estimated Cost of all Components (CAD)				$385

5. Updated Project Plan

An updated Microsoft Project Schedule has been revised for the phases of the project, including the revisited timelines for all the activities. This can be seen below, the schedule will show the entirety of the first semester deadlines and schedule with detailed activities, however for the second semester activities the schedule completion will remain undefined as we are not aware of the complete requirements of the second semester at this time.

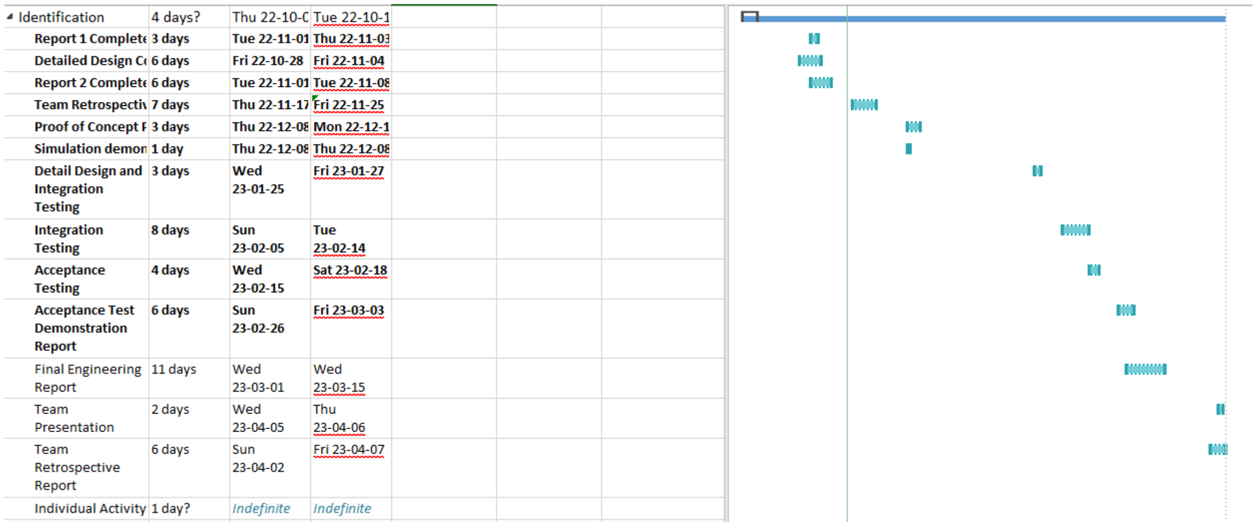

Figure 5-1: Updated Project Plan

6. Contribution Matrix

Table 6-1: Contribution Matrix

	People			
Task	**Mamun Hossain**	**Gobikah Balaruban**	**Manreet Kaur**	**Aaditya Rajput**
Concept Generation and Analysis	70%	10%	10%	10%
Conceptual System Design	10%	40%	40%	10%

Definition of Integration Tests	10%	15%	15%	60%
Estimated Cost	70%	10%	10%	10%
Updated Project Plan	40%	20%	20%	20%
Contribution Matrix	25%	25%	25%	25%

References

[1] “11 Remote Patient Monitoring Companies You should know about,” Plug and Play Tech Center. [Online]. Available: https://www.plugandplaytechcenter.com/resources/10-remote-patient-monitoring-companies-you-should-know-about/. [Accessed: 08-Nov-2022].

[2] “Managing your health in the age of wi-fi,” Mayo Clinic, 18-Jun-2022. [Online]. Available: https://www.mayoclinic.org/healthy-lifestyle/consumer-health/in-depth/telehealth/art-20044878. [Accessed: 08-Nov-2022].

[3] S. M. Lee and D. H. Lee, “Healthcare wearable devices: An analysis of key factors for continuous use intention - service business,” *SpringerLink*, 30-Oct-2020. [Online]. Available: https://link.springer.com/article/10.1007/s11628-020-00428-3#:~:text=Wearable%20devices%20in%20healthcare%20include,and%20biosensors%20(Phaneuf%202020). [Accessed: 08-Nov-2022].