Faculty of Engineering and Applied Science

ENGR 4941U Capstone Systems Design for ECSE II

Remote and Mobile Healthcare System for Home Care

R3: Detailed Design and Integration Test Report

Team Members:
Mamun Hossain, 100553073
Gobikah Balaruban, 100742539
Manreet Kaur, 100766207
Aaditya Rajput, 100622434

Faculty Advisor: Dr. Mikael Eklund
Capstone Coordinator (Fall 2022 and Winter 2023): Dr. Vijay Sood

Abstract

Our project's goal is to create a remote mobile healthcare system whilst incorporating different IoT aspects to further benefit the world of remote mobile healthcare. However, in order to do this there are many limitations that have been imposed, such as technological limitations, server limitations, and security protocols that may be in place. In order to combat these challenges, our group will be creating a mobile healthcare unit using an Arduino Mega 2560 that will be connecting to a cloud hosting service which is Digital Ocean, whilst it containing Apache scripts for web pages, PHP scripts to enable the web pages to be usable for the mobile application, and MySQL acting as a database to store all the information. We will also be using a SEN-11574 for PPG ratings, a MAX30102 biosensor for blood oxygen levels, and a 3 lead system being controlled by an AD8232 to monitor ECG signals.

This report focuses on our project design and the changes we have made from the first semester leading into this second semester and the necessary system developments alongside testing that we had to make along the way. The product initially had used an Arduino UNO to control all the hardware inputs that was connected in tandem with a Raspberry Pi Model 3 to enable a web server within the cloud limitation of a Raspberry Pi, however due to hardware limitations we had come to a conclusion that it would not be an efficient solution to the cloud server as to what we had initially thought.

To ensure the success of our project, alongside with the tests in this report, we will be conducting several different types of tests, including unit tests, integration tests and UI tests, alongside having a detailed design indicating the process our system will undergo. This will allow us to be able to identify and fix any major issues that will occur before delivering the final prototype.

Acknowledgements

We would like to express our sincere gratitude to our Capstone Coordinator and Capstone Professor for their unwavering support, guidance, and encouragement throughout our project. Their expertise and feedback have been invaluable to our success.

Our Capstone Professor, Dr. Mikael Eklund, provided us with insightful advice, and expert knowledge, and helped us to remain on track with our objectives. His mentorship and regular feedback have been invaluable to the success of our project.

We also extend our heartfelt appreciation to our Capstone Coordinator, Dr. Vijay Sood, for his unwavering support and guidance throughout the project. He is always available to offer assistance and support in all areas of the project, and we are grateful for his dedication and commitment to our success.

We would also like to extend our thanks to our colleagues and fellow students who provided valuable feedback and support during the development of our project. Their contributions and insights helped us to refine our ideas and develop a more robust and effective system.

Finally, we would like to thank our families and loved ones for their unwavering support and encouragement throughout our academic journey. Their love and support have been a constant source of strength and inspiration, and we are deeply grateful for everything that they have done for us.

Table of Contents

List of Figures

List of Tables

List of Acronyms used

PPG: Photoplethysmograph

ECG:Electrocardiogram

PHP : Hypertext Preprocessor

UI: User Interface

REST : Representational State Transfer

API: Application Programming Interface

MySQL: My Structured Query Language

HTTP: Hypertext Transfer Protocol

1 Introduction

Healthcare is a crucial part of our everyday lives, and with the increasing use of technology, healthcare services are becoming more and more important as each day passes, especially through the usage of mobile applications. Because of the rapid advancements in technology, it has become possible to connect software components to medical grade hardware components and develop different types of healthcare systems. In this regard, the aim of this project will be to create a reliable and efficient healthcare system that is user friendly and highly available. This report will contain the system design and unit testing process that will allow us to further our understanding in this regard.

1.2 Problem Statement

The problem that we have is that we are trying to create a mobile healthcare system with the integration of our Arduino Mega system, where we will be using DigitalOcean as the cloud hosting platform, Apache to run the webpages, PHP to communicate with the webpages and MySQL to hold receive the data from the Arduino and redistribute that data to the mobile application. The goal is to develop a highly efficient and robust system that will allow users to access healthcare services on their mobile devices. This report should provide comprehensive details on the system design and the implementation, including system architecture and unit testing results.

1.3 Overview of the Report

The remainder of this report is organized as follows. Section 2 presents the detailed design portion, Section 3 presents the unit and integration testing, Section 4 showcases the updated project plan, and finally conclusions are presented in Section 5, and contributions in Section 6.

2 Detailed Design

2.1 Deployment Diagram

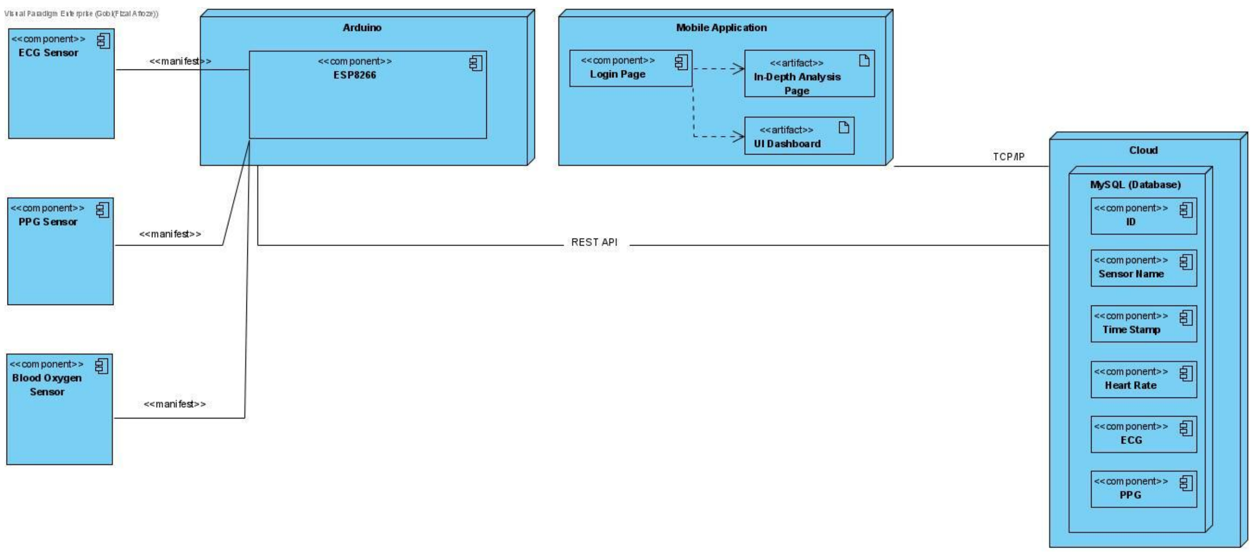


Figure 2.1 - Deployment Diagram

The Arduino is connected to the Cloud Host (DigitalOcean) via the RESTful API (REST API). The data collected from the sensors will be sent to the cloud, which is later stored in the MySQL Database.

DigitalOcean (Cloud)
Digital Ocean is the cloud service provider for the system that is responsible for handling the data being stored and retrieved from the Arduino and the mobile application. The DigitalOcean Cloud droplet has PHP, MySQL, and Apache set up on it. The PHP component receives the data from the Arduino Mega through HTTP requests and passes it on to the MySQL database. MySQL is a relational database management system used to store the data collected from the Arduino. It has various parameters including the User's ID, the measurements of data and the name of the sensor used for that data, and a time stamp for the time the measurement occurred.

Mobile Application

The mobile application is connected to the cloud and requests data from the database. The data is sent from the cloud to the mobile application, where it is able to be viewed by the user in a readable UI. It can be viewed by the patient and/or the doctor.

2.2 Data Flow Diagram

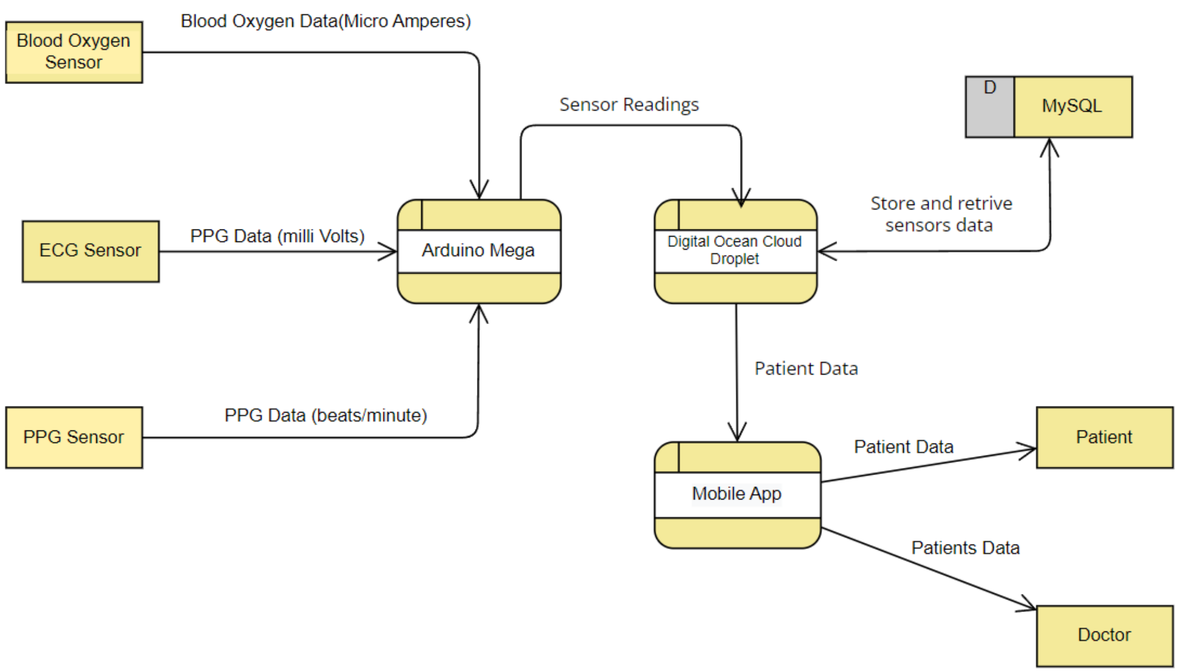


Figure 2.2 - Data Flow Diagram

The above data flow diagram illustrates the flow of information and processes in a our system that collects and monitors health data from sensors using an Arduino Mega 2560 microcontroller, stores the data in a MySQL database on a DigitalOcean server, and displays the data on a mobile application created in Android Studio. The data flow begins with the sensor devices, which collect health data such as PPG data, blood oxygen level, and ECG waveform data. The Arduino Mega 2560 is then used to communicate with the sensors and retrieve this data, which is then stored in the MySQL database on the DigitalOcean server. The mobile application is able to access this data by requesting it from the server, and is able to display the PPG graphs, blood oxygen level, and ECG waveform graphs in a readable format. This data can be accessed by doctors and patients either through cloud or through

2.3 Sequence Diagram

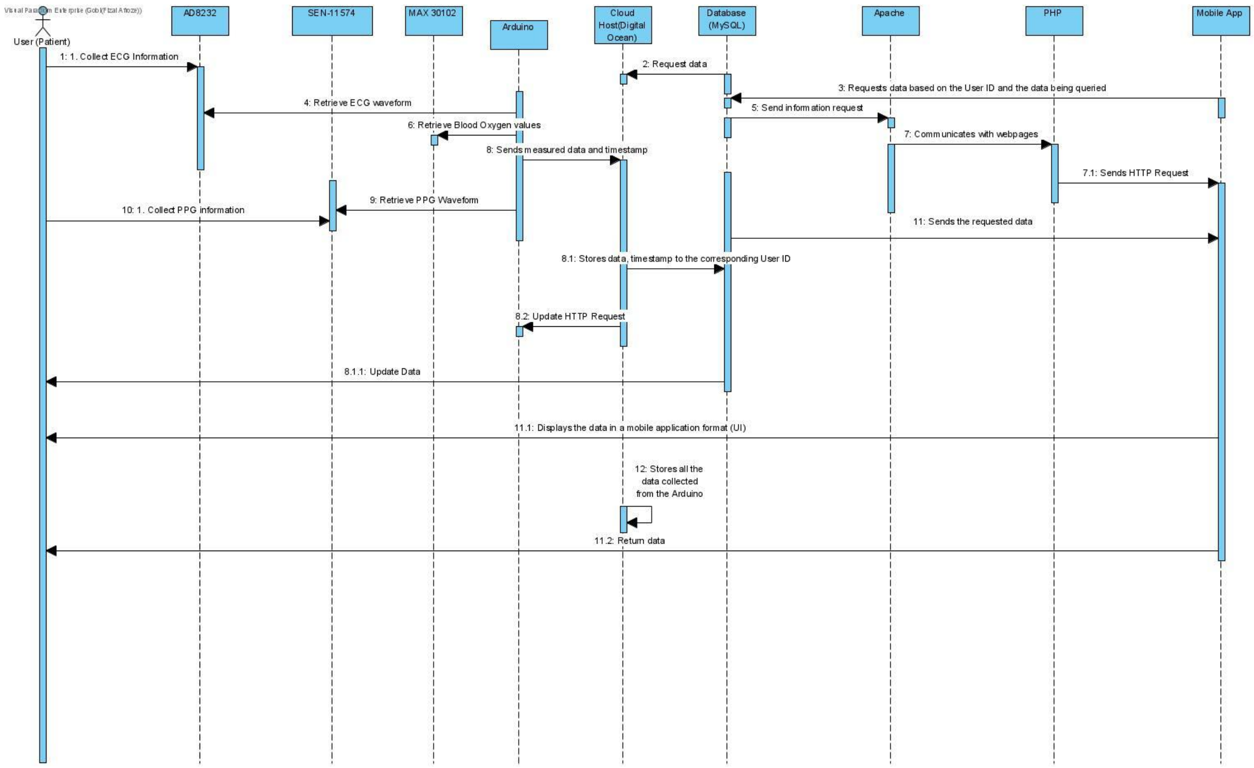
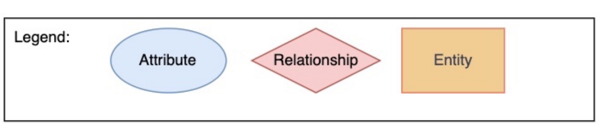


Figure 2.3 - Sequence Diagram

The main functions in this sequence diagram is:

1. The ability to turn the device on
2. The ability to send real time data to the cloud and then to the mobile application
3. The ability to store offline data and then upload to the cloud when possible
4. To view data as deemed necessary
5. View the timestamps of the data

2.4 Entity Relationship Model



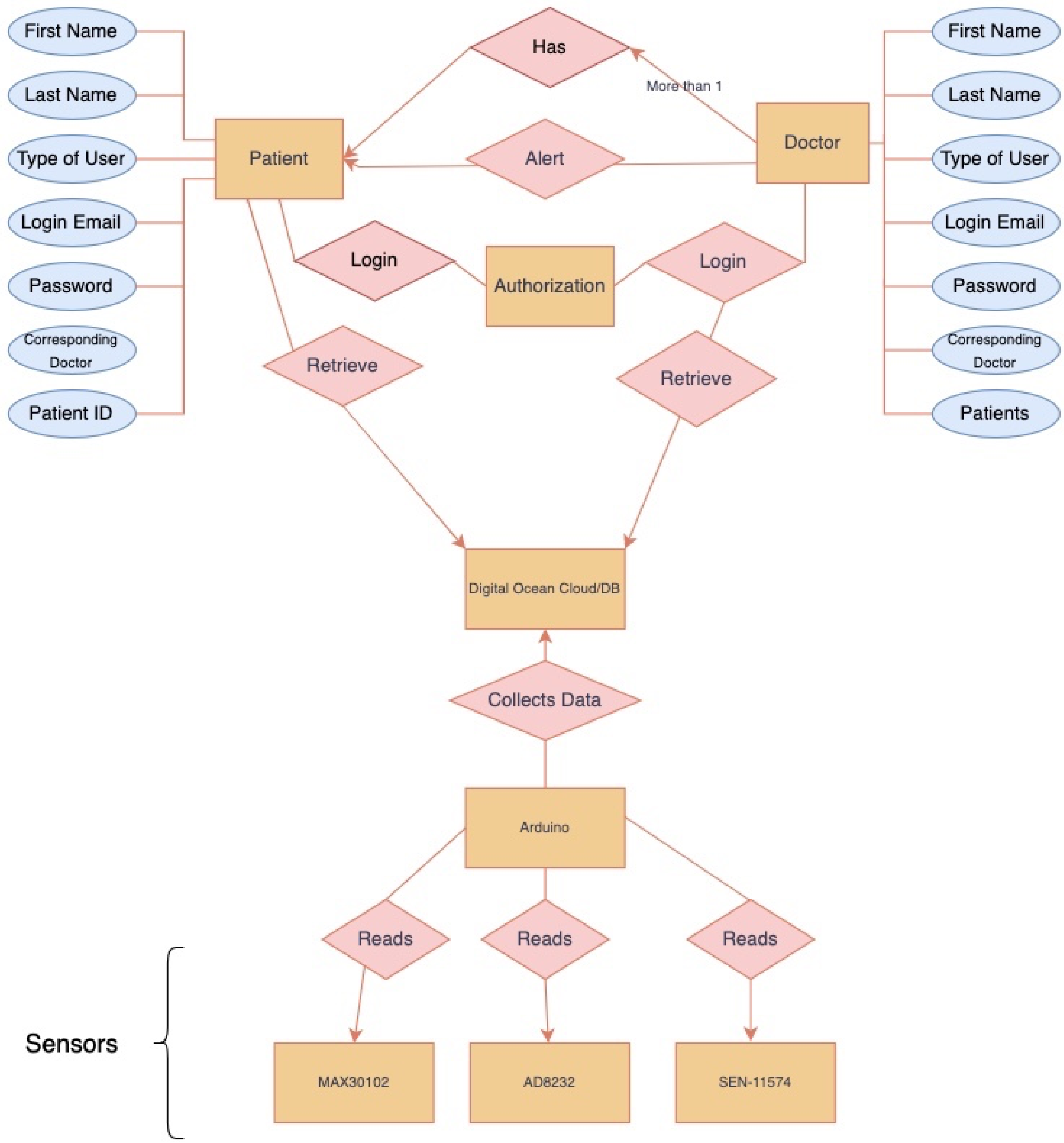


Figure 2.4 - Entity Relationship model

As depicted above, the Patient and the Doctor have many different attributes and interactions with each other and the other entities in the Mobile Application. The main players in this application are in fact, the Patient and Doctor, and what information they can access from their Dashboards/UI. Authentication is a big part of the security and confidentiality of Health Monitoring systems so in order to access any information on the application, both the user and doctor must be logged in. Doctor and Patient can access the Digital Ocean Database via a connection port on the application that will allow Android Studio to read from our MySQL database. This Database will hold information such as personal identification information as well as sensor readings from the Arduino device.

2.5 Mobile Application Prototype and Design

Figure 2.5.1 - Home Page

Figure 2.5.2 - Login Page

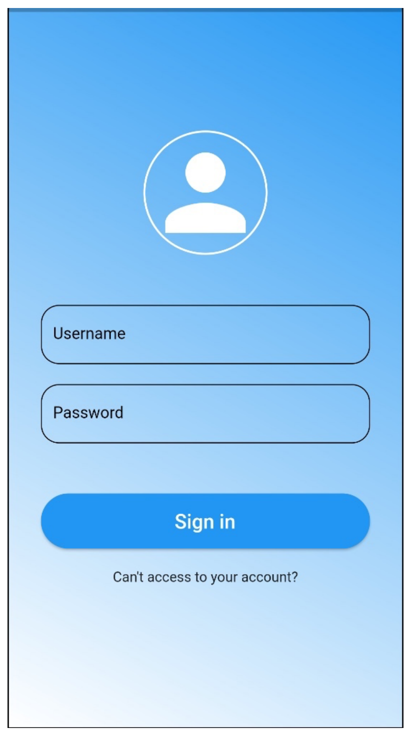
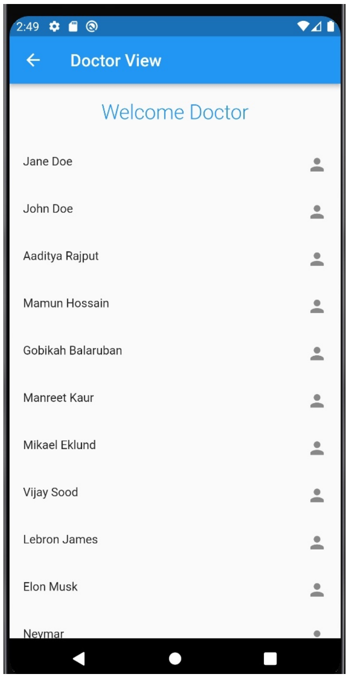
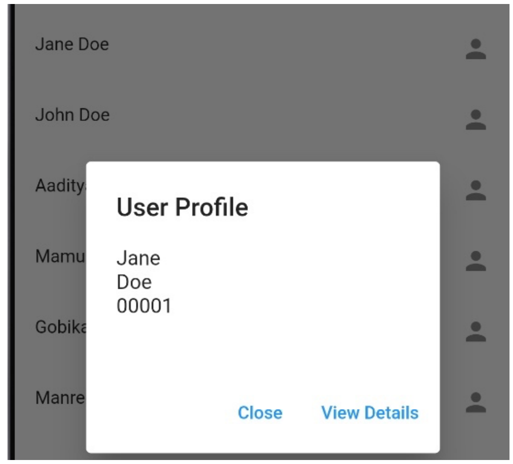


Figure 2.5.3 - Doctor's Dashboard

Figure 2.5.4 - User Details: The doctor is able to see the user details upon clicking on a profile of their patient:





The UI displayed from Figure 2.5.3 to 2.5.4 is showing the current state of the application and how the home screen should look like. This is currently a prototype and a work in progress there are many improvements that need to be added, however we have accomplished a user interface goal by keeping it clean and easy to read. The user will be able to log in and from there be able to view the data for the specified person, however the specific data has yet to be implemented. Some design considerations that need to be made is how to keep the UI consistent with the theme and having the details of the specific sensor that wants to be seen rather than having a “View Details” option to show all of them.

2.6 Circuit Diagram(s)

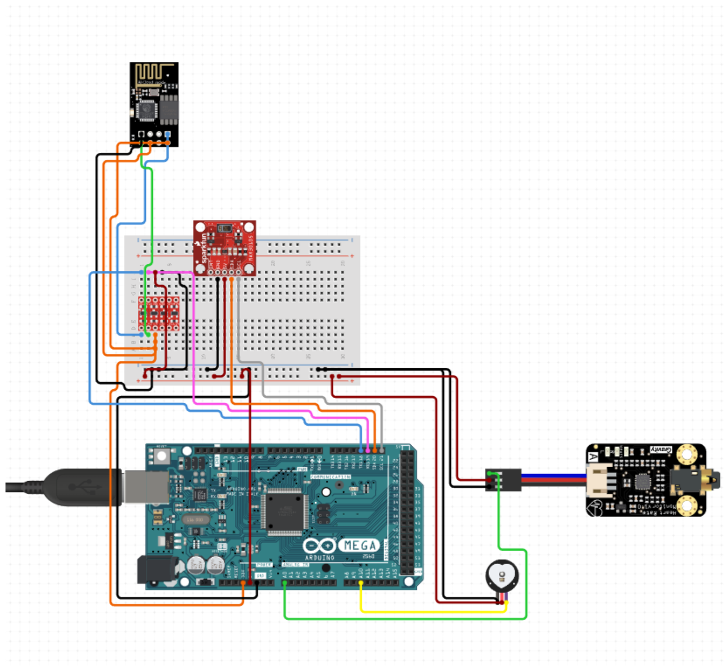


Figure 2.6.1 - Arduino Circuit Diagram of Overall System

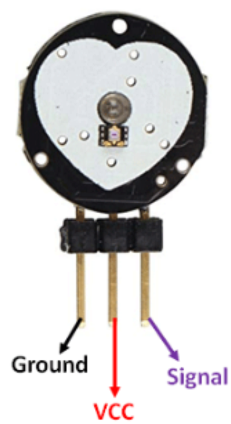


Figure 2.6.2 - SEN-11574

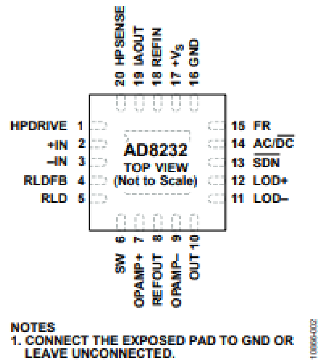


Figure 2.6.3 - AD8232 Pin Layout

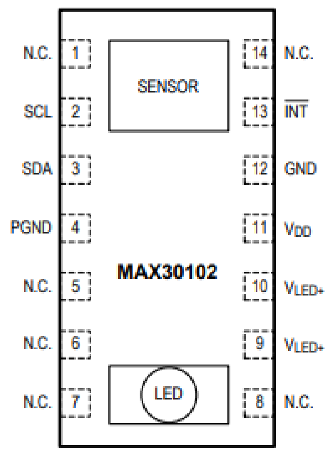


Figure 2.6.4 - MAX 30102 Pin Layout

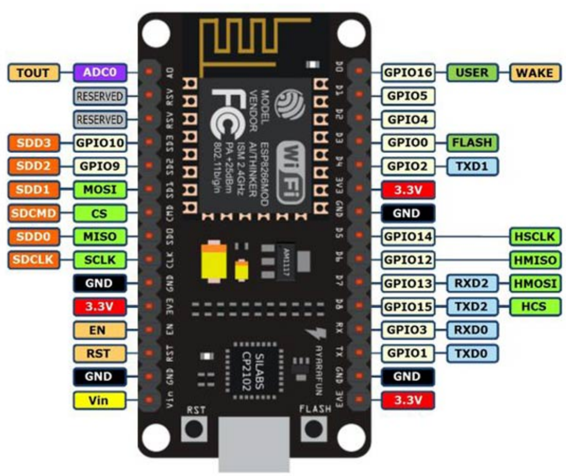


Figure 2.6.5 - ESP 8266 Pinout

The system uses an Arduino Mega 2560 design that will be connected to all the pin layouts as seen in the diagram above. In order to connect to the WiFi, the WiFi Module (ESP 8266) will be connected to the Arduino Mega, not necessarily through a breadboard but can be done if necessary. The other sensors (AD8232, SEN-11574, MAX30102) can also be interchangeable with the UART pins on the Arduino Mega. The WiFi module will allow the sensors to have a reading be sent over to the cloud hosting service and reliably send the data as per necessary.

3. Unit and Integration Testing

Definition of Unit Tests

The unit tests specified are for testing that functions return the correct data and the behavior of the system is as expected. Most of the testing will take place on the Arduino Mega 2560 that will connect to the Digital Ocean server and the mobile application where a large portion of the main functionality will occur.

ID	Test Case Objective	Test Case Description
1	To test the connections between the Arduino Mega 2560 and the sensors are properly established and working as expected.	Test if the Arduino Mega 2560 is able to communicate with the sensors and retrieve accurate data. For the SEN-11574, measuring the PPG data on the Mega and comparing it with the data obtained from the SEN-11574. For the MAX30102 Pulse Oximeter, to measure blood oxygen level from the Mega and compare it with the data obtained from the sensor. For the 3 lead ECG AD8232, measuring the ECG waveform data obtained from the Mega, and then comparing the data that we obtain from the sensor.
2	To test data is properly stored in the SQL database and in the expected format	Test that the data is stored in the SQL database and ensure that it contains all the required information, such as the PPG graphs, the numerical blood oxygen levels, as well as the ECG waveform graphs and ensure that it is in the appropriate format, whilst making sure that the graphs for PPG and ECG are maintaining a series of peak and low points with corresponding timestamps. Ensure that the blood oxygen level is stored as a numerical value.
3	To test the DigitalOcean server is properly receiving and	Test if the data from the Arduino Mega 2560 is being sent to the DigitalOcean server properly and ensure that it is being received and stored. Ensure that the server is

	storing the data	corresponding with the correct HTTP protocol, and that the data is being stored in the server database that has been created with MySQL with the correct required formats and is accessible to the mobile application.
4	To test the mobile application created in Android Studio is able to receive the data from DigitalOcean.	Test that the mobile application is receiving the data properly from the DigitalOcean server that is set up with PHP and MySQL, and making sure that the received data contains all the required information such as the PPG graphs, blood oxygen level, and the ECG waveform graphs in the required format.
5	To test the mobile application is able to display the data	Test that the mobile application created in Android Studio is able to display the PPG graphs, the blood oxygen level in numerical value, and the ECG waveform graphs in a readable format.
6	To test the system being able to handle invalid requests	Test that system will be able to handle incorrect inputs, such as invalid data or data outside the scope of the sensors ie. incorrect timestamps. Ensuring that the system is responding with the error messages. An example of this would be if there is a PPG reading being made outside of the constraints of the sensor, the system should be able to recognize it and allow the user to know that there is an error occurring.
7	To test the system's performance	Test that the system has an appropriate level of performance and reliability by running it over prolonged periods of time and ensure that the system does not have any major issues in doing so, whilst the data is able to store information in real time and have a storage option as well when going offline.

Table 3.1: Definition of Unit Tests

Definition of Integration Tests

Integration testing is necessary to ensure that the required components are not coupled to a point where there will be any issues. We desire our system to be efficient in its performance and as loosely coupled as necessary to ensure that if a component were to fail, the system can still continue to function.

ID	Test Case Objective	Test Case Description

1	Check if there as been a successful transfer of data of PPG data from SEN-11574	This will check the communication between the Arduino and the SEN-11574s by simulating the PPG data in real time from the SEN-11574 to the server via the Arduino Mega and it is being stored in the MySQL database in the correct format. (This is currently being tested carrying over from the previous report) (The parameters in question will be the PPG reading in beats per minute, and having a general threshold of 500-600 BPM).
2	Check if there has been a successful transfer of data of blood oxygen data from the MAX30102	This will check if the communication from the Arduino and the MAX30102 is working by sending blood oxygen levels in numerical format to the server via the serial connections in the Arduino Mega to the MySQL database in the correct format. (The parameters will be in micro Amperes, and will be typically between 600-1200)
3	Check if there has been a successful transfer of data of ECG data from AD8232 to server	This will check if the communication from the Arduiono and the AD8232 is working by sending the simulated data from the system to the MySQL database and ensuring it is received and stored in the correct format. (The parameters for this will be measured in mV with a range of +/- 1.5mV. The waveform should capture a frequency range of 0.5 Hz to 150 Hz, and have a sampling rate of at least 250 hz to be able to accurately capture the waveform).
4	Check if the retrieval of data is successful of the PPG, blood oxygen, and ECG data by the mobile application	This will check if the HTTP request from the mobile application to the DigitalOcean server is being retrieved to store PPG, blood oxygen, and ECG data and verifying that the data that is being received is being sent to the appropriate display in the mobile application and in the correct format. (This is currently being tested carrying over from the previous report)
5	Check if storage of PPG, blood oxygen, and ECG data is successful in MySQL server	This will check if the data from the sensors from the Arduino Mega 2560 is being sent to the server and being stored in the MySQL database in the correct format.
6	Check if validation of incoming data is successful with PHP scripts	This will check if the simulated data that is sent to the server will verify that the data is being processed in an appropriate manner and being stored in the MySQL database within the correct parameters, whilst also rejecting data that is outside the parameters of the data.
7	Check if integration of	This will check if the data is sent to the server and will

	DigitalOcean, Apache, PHP, and MySQL is successful	verify that the data that is sent over is stored and processed in the database, therefore allowing the system to ensure that all the individual components are working together in cohesion.
8	Check if data is being transferred accurately to the server in real time	This will check that the data that is being sent over will be stored in the MySQL database in real time, to confirm that the system can handle a high volume of incoming data.
9	Check if system can handle multiple data HTTP requests from mobile application	This will check the systems response to sending several HTTP requests within a short time period to retrieve the allocated data (ie. PPG, blood oxygen, ECG) and ensure that the server will return the correct data under the time load.
10	Check if system can handle multiple streams of data from Arduino Mega 2560	This will check the data from the sensors and ensure that the server can receive and process the data in the correct format.
11	Check if the storage is successful and data is transferred in the correct format in MySQL	This will check the data that is sent to the server and ensure that the data that is being stored in the database is in the correct format.
12	Check if the security protocols in place are successful	By sending the data over to the server via an unrecognized source, the system will ensure that the data is rejected and will display an error message.
13	Check if mobile application can handle errors	Forcefully input an error in the mobile application and ensure that the error message is displayed in accordance with the data parameters.
14	Check if server can handle errors	Forcefully input an error in the server and ensure that the error message is displayed in accordance with the data parameters.
15	Check if HTTP protocol is successful	This will check if the data that is being processed and sent over to the MySQL server is using an incorrect HTTP protocol and will reject the data if so and will have an error message.
16	Check if the PHP script data processing can handle errors	By forcing an error in the PHP script for any incoming data outside of the correct parameters, we will verify that it will be able to handle it as so and prompt an error

		message.
17	Check if we have the correct data analytics and visualization tools	Query the MySQL database using the data analytics tool that can be used for test cases to ensure that it meets the needs for the end-users.

Table 3.2: Definition of Integration Tests

Definition of UI Tests

ID	Test Case Objective	Test Case Description
1	Test that the user can connect the Arduino to the Mobile Application	This component has not been tested yet, however this will check if the user can establish a connection between the mobile application and the Arduiono over a wireless connection.
2	Test that the user can view the PPG data	This component has been tested and checks that the pulse sensor does take the readings in real time however it has yet to be connected to the mobile application but we are able to confirm the working accuracy, as well as being able to see the graph waveforms.
3	Test if the user can view ECG data	As per changes, this component is now in the testing phase and this will ensure that the user will be able to see the ECG data in real time, as well as the graph waveforms as provided by the sensor (AD8232).
4	Test if the user can view the Blood Oxygen level data	As per changes, this component is now in the testing phase, and this will ensure that the user will be able to see the Blood Oxygen levels in real time as well as view the numerical values being provided by the sensor (MAX30102)
5	Test if the user can save the data	This component has not been tested yet, but will ensure that the user can save the heart rate, ECG, blood oxygen level, and PPG data to their device for future reference.
6	Test if the user can store data while being offline from the system	This component is currently being tested to see the real time data that is being stored while the Arduino device is separated from the network but still storing any medical information that will get processed to the server when getting reconnected to the system after a period of time.

Table 3.3: Definition of UI Tests

Testing Plans for System

As the cloud server system is still in the process of development and research, as a result our test plan is as well. The plan for testing the system would need to ensure that we are reaching the bare minimum goal of being able to transfer the data from the arduino via an HTTP request to the cloud server which would then get sent to the mobile application with a RESTful API in place. However, this will not solidify the entirety of the testing, but will ensure that we have a guarantee that the system is doing well in this process.

Testing the cloud server will be a major priority at this stage. This means ensuring that the data is being formatted correctly in correspondence with the parameters set for each of the sensors, whether it be a graph based on peaks and lows, or numerical values, as well as storing the information offline. If the model is unable to reach these goals, this will indicate a flaw in our database design or a flaw in the HTTP request (ie. connecting the HTTP request to the wrong port).

Other things that we will need to test in the model is the connection and data transfer of the sensor readings to the mobile application itself and include checking the values for all the sensors and ensuring that it is all within our understanding. Having a cloud based system highly depends on having a highly available and efficient system in place.

4. Updated Project Plan

Beginning with the first semester, the project plan described the sprints that we had gone through throughout the months. For every interval, there was a deliverable that was required upon us that had exemplified our progress in our system and how to continuously evolve and improve our understanding regarding the project. As seen below, we will be able to see the expectations that were needed by the end of the Fall 2022 term, and the new expectations that we have set in place for the Winter 2023 term. We will notice that we had changed our system due to a hardware constraint, where we as a team had unanimously agreed that having both an Arduino + Raspberry Pi system was a redundant system and was not able to efficiently reach the goals that we had wanted, which in turn has led us to use a singular Arduino Mega 2560 and connect that with a WiFi shield to our desired outputs ie. DigitalOcean server.

Expectations achieved at the end of Fall 2022:

● Establish a connection between the Arduino UNO and Raspberry Pi
● Establish a connection with the Arduino to handle hardware and use the Raspberry Pi as IOT device
● Establish a connection between the server and mobile application
● Have a prototype model
● Create a service that calculates ECG waveforms and PPG Waveforms
● Establish the server as a web server, IoT sensor controller, and ML environment

● Mobile Application Version 1 setup should be complete
● Seamless and Clean GUI on the mobile application
● Create a page on the mobile app that controls user authentication for patient vs doctor
● Information is being relayed between the mobile application and the database.

Expectations by the end of Winter 2023:

● Establish a connection between the Arduino MEGA 2560 to DigitalOcean Droplet
● Establish a connection with the Arduino to handle hardware and use PHP scripts to connect the web server to a mobile application and using Apache to host the scripts
● Establish a connection between the server and mobile application using PHP
● Have a completed refined service that will be tested and ensure a seamless connection to the cloud server
● Finalize the web server for the system
● Ensure we can have offline storage for the sensors
● Work on an administrative side of the system as a potential web application.
● Keep up to date with reports
● Find ways to optimize the system
● Have a poster for the exhibition
● Have a fully functioning prototype

5. Conclusion

In conclusion, our report 3 to create a detailed design report as well as unit testing and integration testing has proved to be successful thus far. We were able to implement a new system and grasp a better understanding of what we were doing in comparison to previous reports. We now have a smaller footprint for our design with a smaller form factor but also having more computational power as well.

Throughout this report we had encountered several challenges, such as understanding how to connect the server to the mobile application, what would be needed for the actual server itself, and understanding how to read different sampling rates and how to incorporate the information we get from datasheets and implement that into our design.

Our project has so far made significant improvements in where we had first started to where we are now, and we are proud of where we have reached at this point. We hope to continue our project with full earnesty and strive to create a system that will prove to be a success.

6. Contribution Matrix

	People			
Task	**Mamun Hossain**	**Gobikah Balaruban**	**Manreet Kaur**	**Aaditya Rajput**
Detail Design	10%	30%	30%	30%
Integration/Unit Testing	40%	15%	15%	30%
Updated Project Plan	25%	25%	25%	25%
Arduino Mega + Sensor testing	70%	10%	10%	10%
Mobile App Design	20%	20%	20%	40%
Cloud Server Connections	10%	20%	60%	10%
MySQL Database	10%	50%	30%	10%
Mobile App Database	30%	20%	30%	20%
Mobile App UI	20%	20%	30%	30%

7. References

[1] “Understating ECG sensors and how to program one to diagnose various medical conditions,” *Understating ECG Sensors and How to Program AD8232 ECG Sensor with Arduino to Diagnose Various Medical Conditions*. [Online]. Available: https://circuitdigest.com/microcontroller-projects/understanding-ecg-sensor-and-program-ad8232-ecg-sensor-with-arduino-to-diagnose-various-medical-conditions. [Accessed: 18-Feb-2023].
[2] “Technical reference - espressif.” [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf. [Accessed: 18-Feb-2023].
[3]“Single-lead, Heart Rate Monitor Front End Data sheet AD8232.” [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD8232.pdf. [Accessed: 18-Feb-2023].
[4]“Sen-11574 Datasheet(PDF) - sparkfun electronics,” *Allicdata electronics*. [Online]. Available: https://www.allicdata.com/pdf/sen-11574-datasheet/2718390.html. [Accessed: 18-Feb-2023].
[5]Last Minute Engineers, “In-depth: Detect, Measure & Plot Heart Rate Using Pulse Sensor & Arduino,” *Last Minute Engineers*, 25-Nov-2022. [Online]. Available: https://lastminuteengineers.com/pulse-sensor-arduino-tutorial/. [Accessed: 18-Feb-2023].
[6]“Errors when uploading a sketch – arduino help center.” [Online]. Available: https://support.arduino.cc/hc/en-us/articles/4403365313810-Errors-when-uploading-a-sketch. [Accessed: 18-Feb-2023].
[7]“Click MAX30102 higsensitivity plse oimeter an heartrate sensor or ...” [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/MAX30102.pdf. [Accessed: 18-Feb-2023].
[8]“Android - PHP/mysql,” *Tutorials Point*. [Online]. Available: https://www.tutorialspoint.com/android/android_php_mysql.htm. [Accessed: 18-Feb-2023].