Healthy-Gamer Design Review

Prepared for:
Professor Nathalia Peixoto

Prepared by:
Saad Abdullah
Moneeb Ahmad
Aayush Aryal
Priyam Das
Ryan Gunawan
Muhammed Jamil Rahman

II. Summary of Motivation, Identification of Need and Requirements definition.

Motivations:

It is easy for someone to get hooked onto a certain activity or hobby that they participate in for entertainment or for stress alleviation. Gaming can be a fun and an addictive hobby, however, it can sometimes be hard for a gamer to know when to quit playing. The amount of time a gamer spends playing video games has steadily increased, from 5.1 h/week in 2011 to 6.5 h/week in 2019 [1]. This extended period of gaming can be detrimental to a gamers psychological well-being as well as causing issues with their health. In the DSM-5, the American Psychiatric Association defined Internet Gaming Disorder with diagnostic criteria closely related to Gambling Disorder [1]. As for physical health, extended gaming can cause major repercussions such as Carpal Tunnel Syndrome (CTS). CTS is an instance of pain, numbness, and/or tingling that occurs as a result of this nerve being pinched, squeezed, or compressed [2]. Most often, CTS will arise due to a series of repetitive movements performed over a long period of time, for example, a gamer typing on a keyboard or button-mashing a controller. As gaming continues to grow in popularity these physical and psychological issues that are associated with long gaming sessions will continue to arise in frequency. In order to mitigate these issues we plan on creating a device called the “Healthy-Gamer” that will be used to monitor a gamers psychological and physical health while they are gaming. If the Healthy-Gamer detects any health issues that are arising in the gamers, then the device will send an alert to the user letting them know to take a break from their gaming session. With the implementation of the Healthy-Gamer, a gamer can continue to enjoy their hobby while also retaining their psychological and physical health.

Identification of Need:

This Seamless Physiological Monitoring product will constantly monitor and measure a user’s mental and physical health while they are gaming for an extended period of time. The Healthy-Gamer will record a user’s stress level, heart rate, body temperature, movement and alert them if their levels are high. The data that is being recorded from a user will also be uploaded to a cloud-based dashboard for off-site monitoring. The data will also be displayed to the user through an LCD screen where they can cycle through their different measurements in real time.

Project Requirements Specifications

Mission Requirements

- The Healthy-Gamer shall measure the physical and mental health of gamers.

Operational Requirements

- Input/output requirements
  - The Healthy-Gamer will have multiple sensors on in order to calculate a user's symptoms and display it onto a simple LCD screen.
  - The Healthy-Gamer shall accept an input from a user through a push button in order to allow users to cycle between different monitoring sections on the LCD screen.
  - Data will be stored onto a cloud system.
- External Interface requirements
  - The deceive will be battery powered.
- Functional Requirements
  - The Healthy-Gamer will use the sensors to collect a user's symptoms every 10 milliseconds.
  - The Healthy-Gamer should detect errors and provide visual notification.
  - The Healthy-Gamer will alert the user if their stress level, heart rate, body temperature, or wrist movement is at a high rate for an extended period of time.
- Technology and System-wide requirements
  - The cloud storage system that will be used for this project is Azure.
  - The program languages that will be used for this project are C and Python.
  - The sensors that will be used in this project are: Reflection-Type Heart Sensor, Accelerator, Temperature Sensor.
  - The Microcontroller that will be used for this project is a Raspberry Pi Zero.An LCD display as well as a wifi card will be used in this project.
  - The buses that will be used in this project will be I2C and UART.
  - The cost for this project will be less than $600
  - The power requirements for this project will be less than 15 volts.

Background:

Stress Level:

- Stress can be calculated by finding the user’s Heart Rate Variability
- HRV can be calculated using the formula:

$$HRV = STD\,RR = \sqrt{1/(N-1)\sum_{j=1}^{N}(RR_j - \overline{RR})^2}$$

- HRV is the standard deviation of the of all of the RR intervals which is the distance between each heartbeat

- The Variables in the formula itself are as follows:
  - N is the number of heartbeats that will be measured
  - RRj is the starting heartbeat the calculation will go from
  - RR (repeating) signifies that the heart will keep beating

Carpal Tunnel Syndrome:

- Correct Wrist Position and angle can help lessen the chance of developing CTS.
  - By using an accelerometer, it is possible to alert the user if their wrist is positioned at an angle that puts strain on the wrist
- To calculate Rotation Angle of the wrist the following equations are used for the x and y axis:

$$\alpha = tan^{-1}\left(\frac{A_y}{A^2{}_x + A^2{}_z}\right)$$
$$\beta = tan^{-1}\left(\frac{A_x}{A^2{}_y + A^2{}_z}\right)$$

- For the z axis the following equation is used:

$$F = \sqrt{X^2 + Y^2 + Z^2} = \sqrt{H^2 + Z^2}$$

III. System Design/ Architecture - System Decomp and Interfaces Functional Decomposition

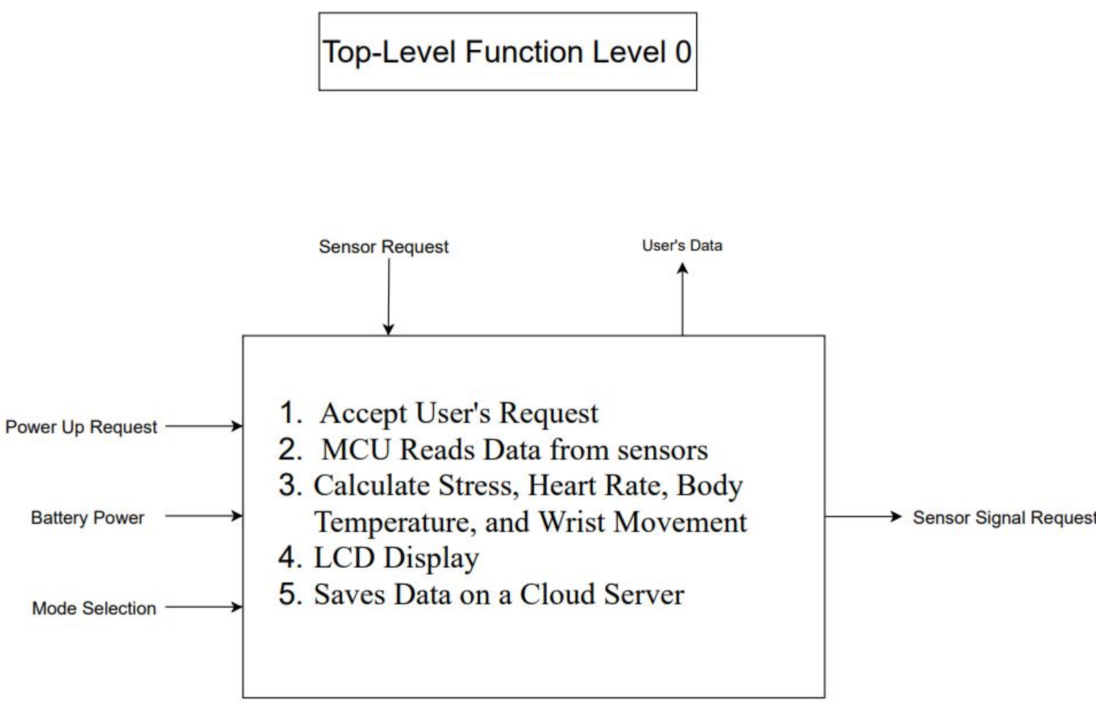

Figure 1: Top Level Function Level 0

This is level 0, a general overview of the system design/architecture of the physiological device. The inputs to the Healthy-Gamer are placed on the left and top side of the block diagram. Above it, we have the user input that is going to interact with the Healthy-Gamer. Users will be able to interact with the Healthy-Gamer through the click of a button. Then, the input will go through the device and be output as sensor signal request which is placed on the right side of the diagram. The main functions of the Healthy-Gamer are as follows: accepts user requests, has data reading capability, calculates stress, calculates heart rate, body temperature using sensors, and wrist movement using accelerometer.

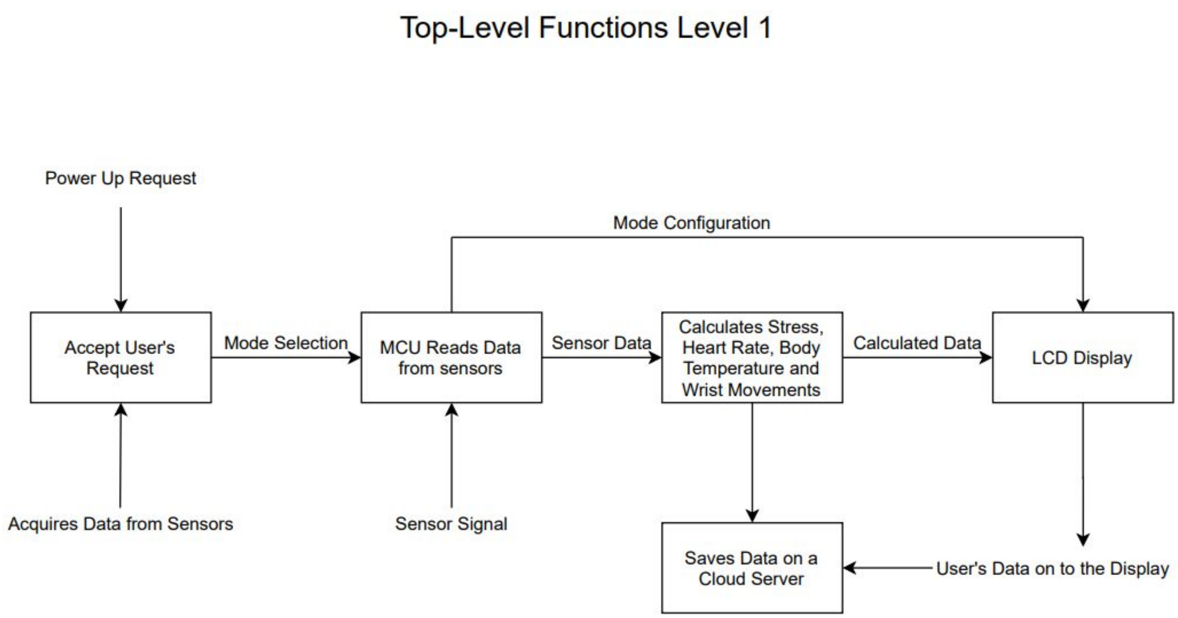

Figure 2: Top Level Function Level 1

A one step down from level 0 functional architecture is level 1 functional architecture. It goes into more detail regarding the top level functions and how they are integrated with each other. The above diagram illustrates how each element communicates with each other and shows the device's operation as it goes through its functions. On the top left corner, we have the device accepting the users' requests. Then, it sends that to the MCU which reads data from various sensors. The output from that then goes through the CPU of the device and various values are measured based on the inputs. After that, the calculated data goes to the LCD display for the user to see and take all necessary actions based on that. Along with sending the data to the LCD display, we also send the data to a cloud server where we will save it and the user can go through his/her data accumulated over the time.

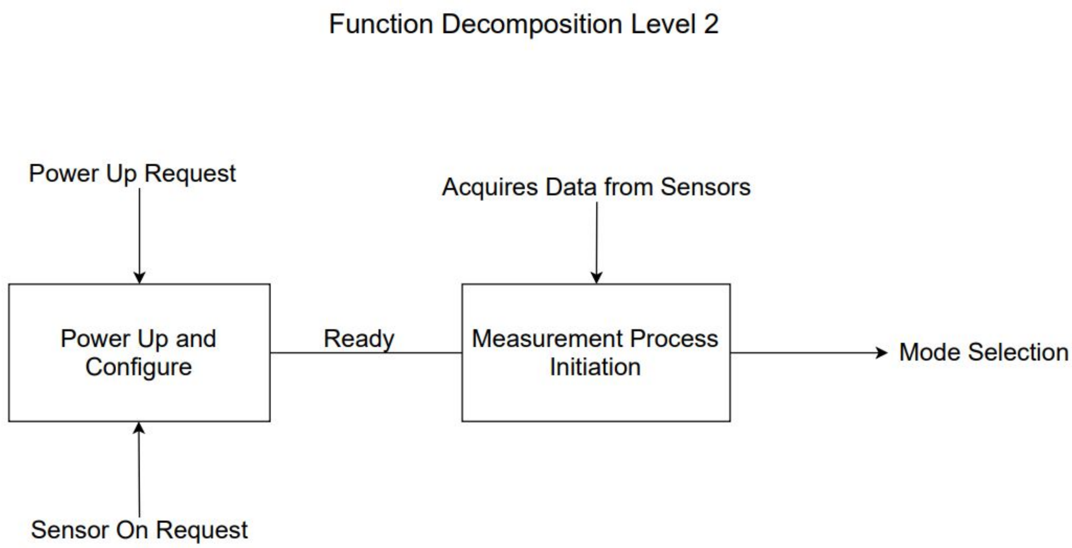

Figure 3: Function Decomposition Level 2

The user will need to power on the Healthy-Gamer in order for the device to start reading their vitals. Once the user turns on the Healthy-Gamer the sensors will begin to read the user's health information. The user will also be able to use the mode selection feature in order to cycle through their different health readings.

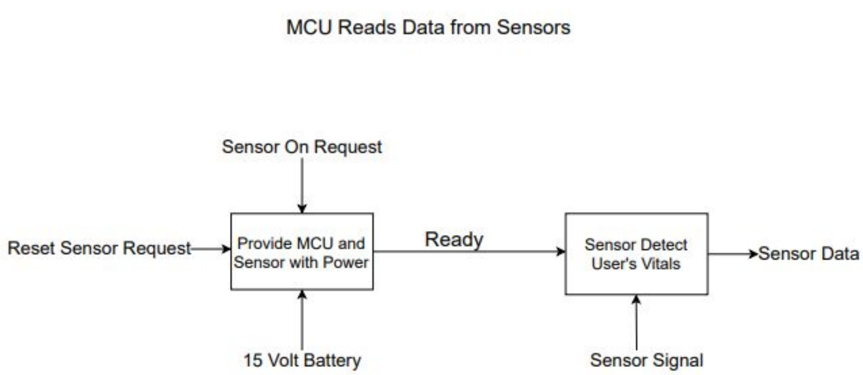

Figure 4: Reading Data from Sensors

The sensor within the Healthy-Gamer will send the user’s data over to the MCU. The MCU will then begin to process the user’s health information.

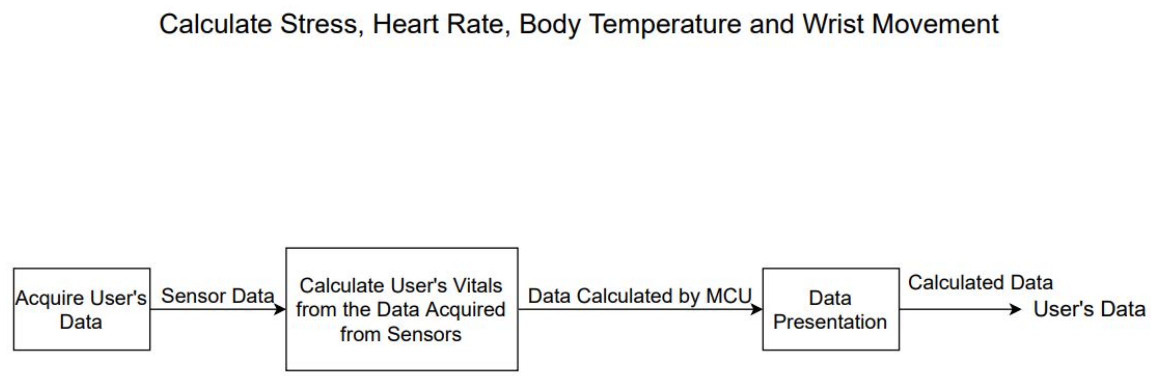

Figure 5: Calculations done by the MCU

The data collected from the user is sent to the Raspberry Pi Zero where data from different sensors such as the Heart Rate-Sensor and Accelerometer calculates the user’s vitals. The Raspberry Pi will now begin to show the user’s data.

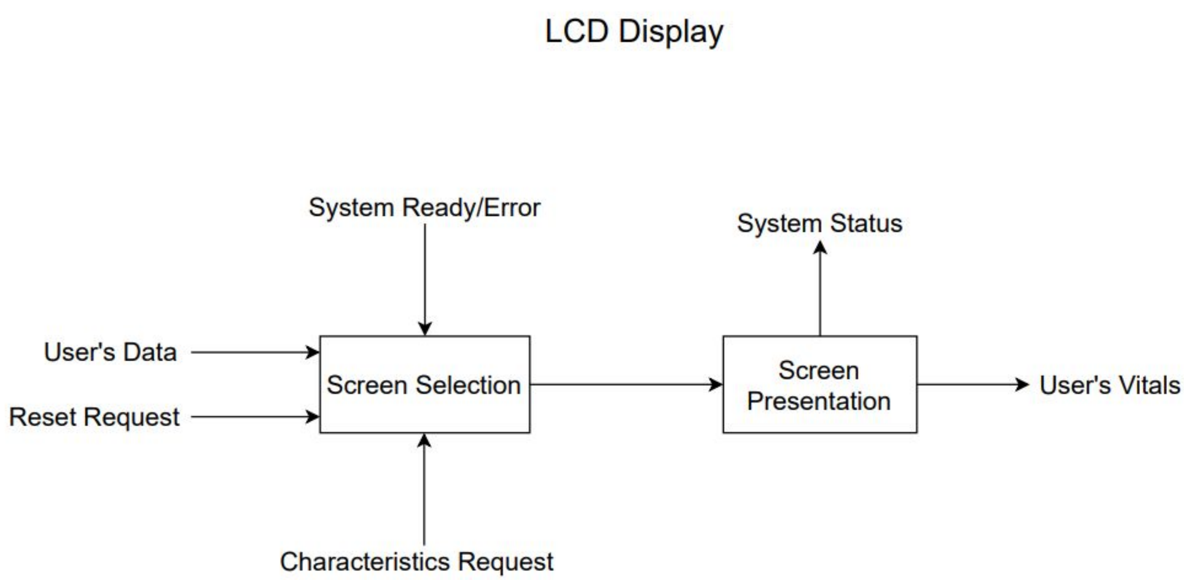

Figure 6: LCD Display

The user's Data is then sent to the LCD Display's screen selection allowing the user to choose what they would like to view. After a selection is made, the data collected from the sensor is shown to the user.

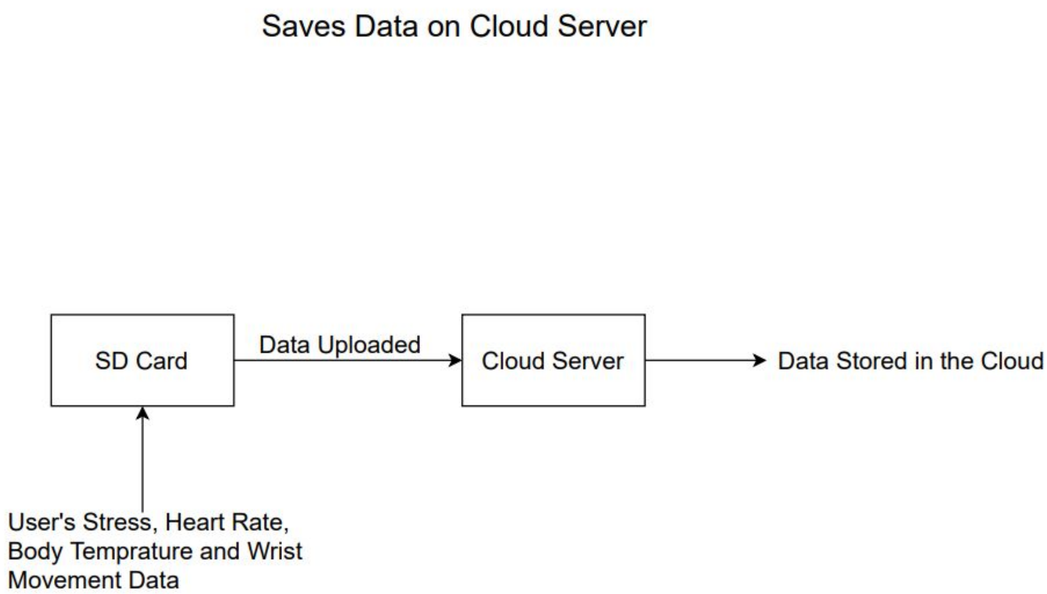

Figure 7: Saving Data on a Cloud Server

A copy of the data collected from the sensor is transported to the SD Card of the Raspberry Pi, it is then uploaded into the ThinkSpeak Cloud Server to be stored.

Generic Physical Architecture

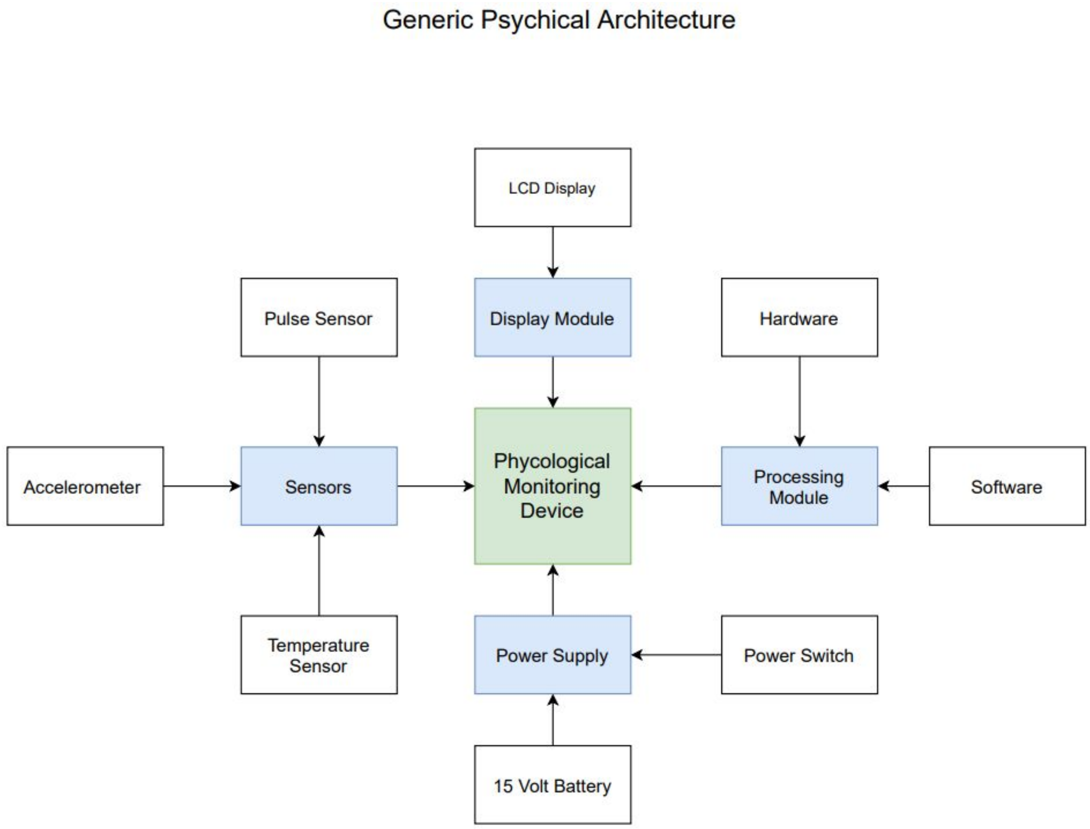

Figure 8: Generic Physical Architecture

The figure above represents the overall physical architecture for our health monitoring device. In the Healthy-Gamer the sensors that we will be using are going to be: an accelerator, heart pulse rate sensor, and thermal resistor sensor. The device will contain a 15 volt battery for its power supply. This battery will be able to provide power to our: sensors, LCD screen, and the MCU. The MCU will be responsible for interpreting the user's health information as well as uploading their information to the cloud.

System Architecture

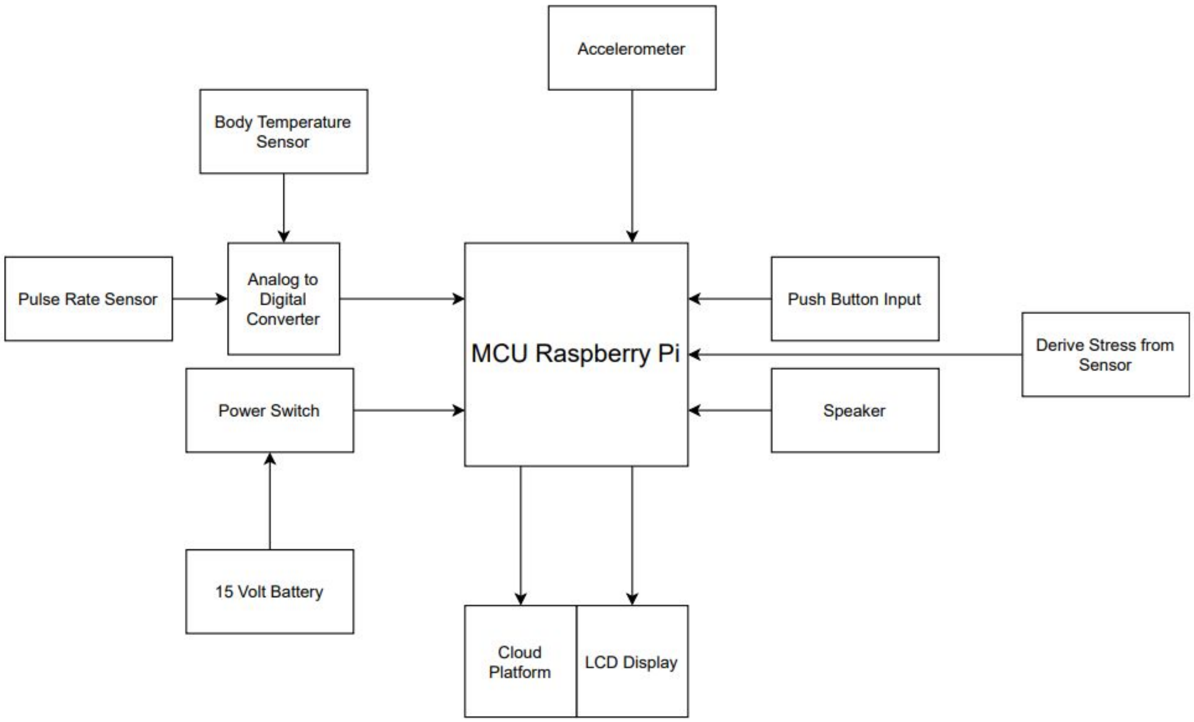

Figure 9: System Architecture

The figure above is a visual representation of all components that are either directly connected or is accessed by the Raspberry Pi. The Raspberry Pi will connect to all the sensors through the serial interface and receive data from the Analog-to-Digital Converter for both the Temperature and Heart Rate sensor. The LCD Display is connected to the Raspberry Pi directly through Serial Peripheral Interface (SPI). The Display will allow the user to view vital information about their health. Also, the Raspberry Pi will store all its data within an SD card - the data is then sent to the ThinkSpeak servers for secure storage.

IV. Detail Design - (a) Circuit Schematic level (b) flow diagrams with identifications of subroutines:

A.) Circuit Schematic Level

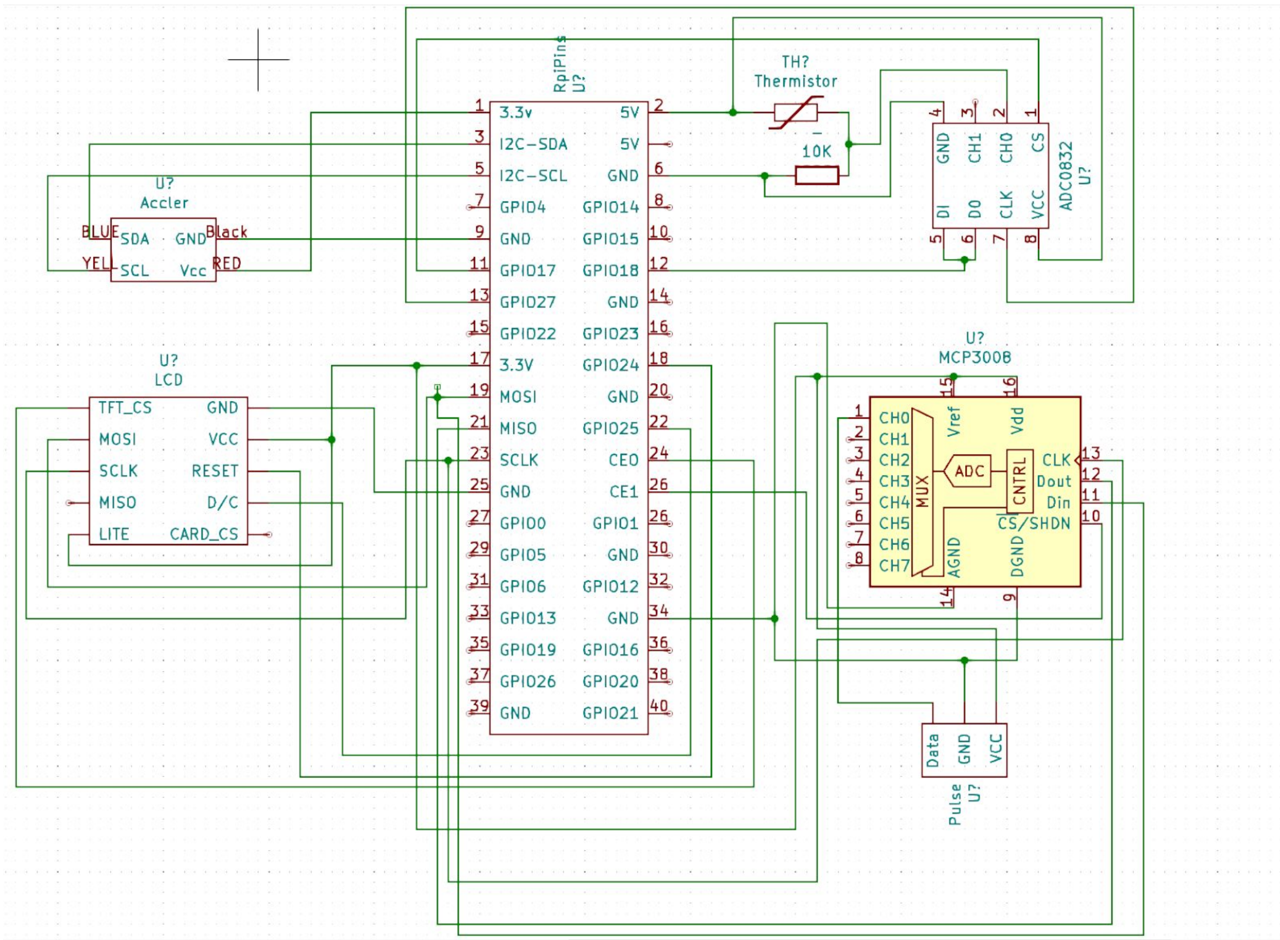

Figure 10: Circuit Schematic

The accelerometer is connected to the Raspberry Pi's SDA and SCL ports. This allows for the I2C protocol to be properly established. This allows synchronous communication between the Pi and the accelerometer . The thermistor is hooked up to an ADC and the ADC data is read from the Pi. The pulse sensor is also hooked up to an ADC which can allow the pulse sensor to speak to the SPI bus on the Pi. The LCD is also hooked up to the SPI bus.

B.) State Diagram

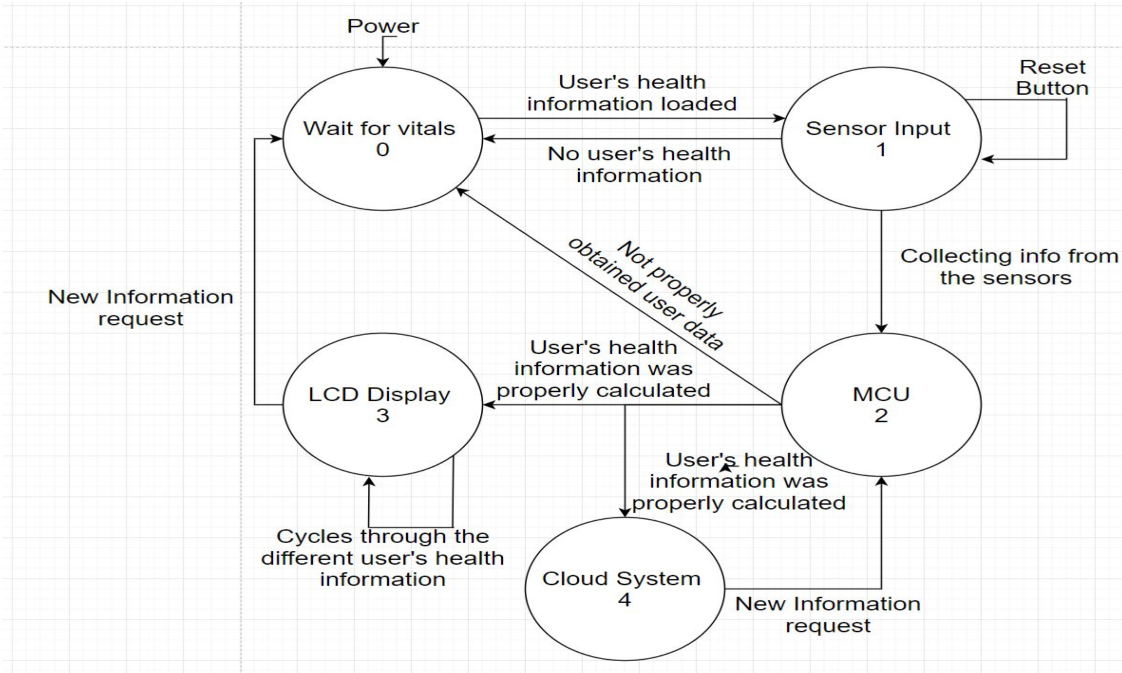

Figure 11: State Diagram

The figure above represents the state diagram for the Health Monitoring device. Once the Healthy-Gamer powers on it the software will request the sensors to read a user's health information. Once the sensors in the Healthy-Gamer collect the user's vitals it will send it over to the Raspberry PI where it will then begin to convert the data from the sensor into readable information. Once the Raspberry PI finishes converting the information received from the sensors, it will then send it to the LCD screen where it will display the user's health information. The Raspberry PI will also be responsible for uploading a user's health data into the cloud. The LCD will then alert the user if their health is in a critical state and advise them to take a break from their gaming section.

C.) Software Design

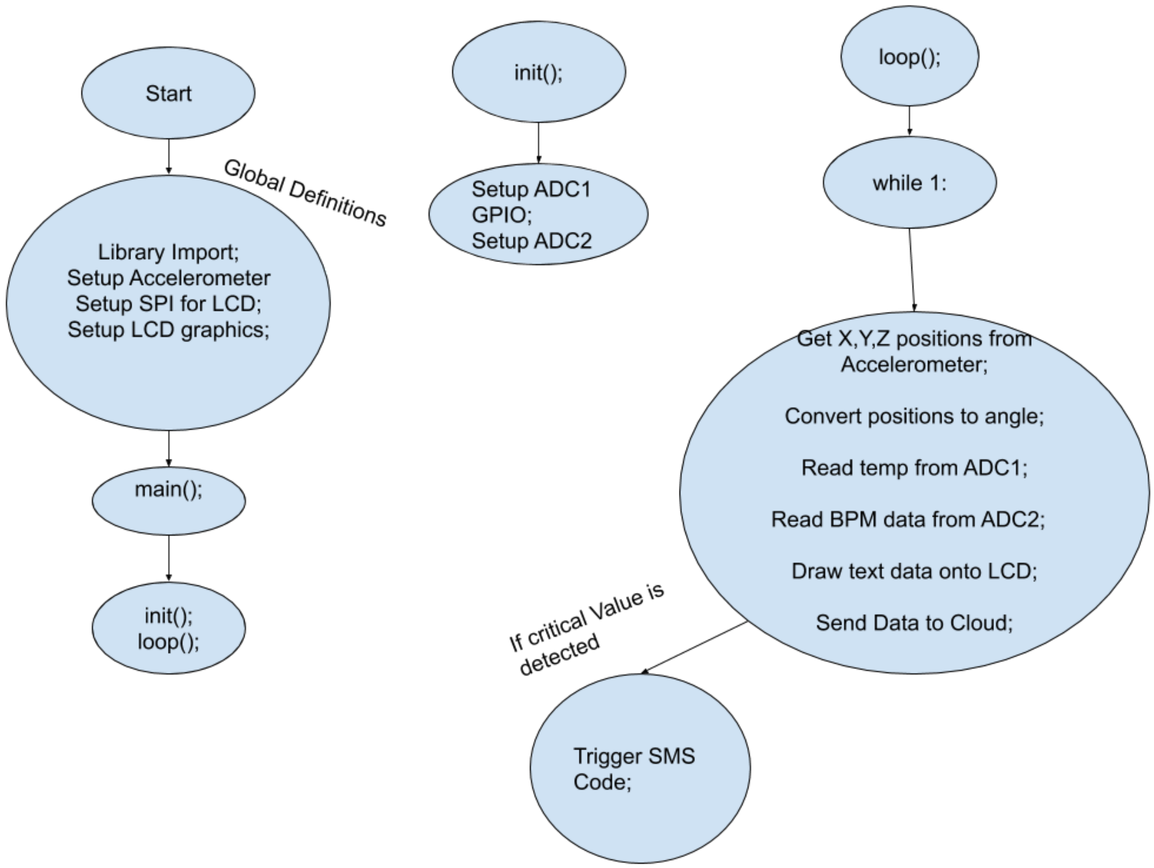

Figure 12: Software Design

The algorithm begins by setting up I2C for the accelerometer and SPI for the LCD. The LCD also needs some graphical data to set up ahead time so everything can be properly drawn on the display. Then the algorithm setups both ADCs to properly convert the thermistor data into a temperature value and to convert the pulse rate sensor into BPM. After that it enters the main loop where everything is done. First it obtains the x,y and z positions from the accelerometer and then converts it into a wrist angle. Then it reads the temp from one of the ADCs, and gets the BPM from the other ADC. Then this data is sent to the LCD and the Cloud. If any critical values are detected then this will trigger the SMS feature.

V. Prototyping progress report - what was built / experimented with, etc.

Prototyping is an important part of this project as it ensures that we have properly designed our portable health monitoring device to work as intended. Although having a slow start we have made a significant amount of progress in creating working portions in our health monitoring device. One of the requirements for this project that we have implemented so far is analyzing a user's body temperature. In order to analyze their temperature we used a thermal resistor which was paired up with an ADC chip which would then send the data of the sensor to the Raspberry Pi. Once the Raspberry Pi received the data we wrote a program that would analyze the data and convert it into a temperature reading. If the sensor detects that a user's body temperature is above 100°F the Health-Gamer will alert the user of their high body temperature and advise them to take a break from their gaming sesion. See figure 13 for a sample screenshot.

Another part of the device that we have successfully implemented is tracking a user's wrist movement using an accelerometer. In order to test our accelerometer we connected it to a breadboard which would then send the information to the Raspberry Pi via the I^2C protocol. The purpose of the accelerometer is to keep track of the user's wrist position and alert them if it is at a strenuous position for a certain period of time. As stated earlier the reason why we decided to focus on keeping track of the user's wrist position is to assure that they don't obtain CTS from their gaming session. See figure 13 for a sample screenshot.

A portion that we have successfully completed is configuring the LCD to display a user's health information. The screen was able to show both the user's wrist angle, heart rate and body temperature in real time. The screen will also alert the user's if their health is in a critical state and will advise them to take a break from gaming. See figure 14 for a sample screenshot.

We have also implemented the pulse rate sensor into our design so it can read a user's heart rate. We were able to design the code which enables the raspberry pi to interpret the information that is produced from the pulse rate sensor. As of right now the pulse rate sensor is able to read a user's heart rate at an accurate but inconsistent level. We believe that the issue arises from the positioning of the sensor. After doing more research we came to the conclusion that the heart rate sensor needs to be elevated slightly above the user's wrist so that way it can read the user's heart rate at a constant value. We plan on adding a thin layer of glass into our Healthy-Gamer device so that the pulse rate sensor can be elevated enough to read the user's heart rate at a constant value. See figure 15 for a sample screenshot.

We have also begun prototyping the different ways that we can offload the user's information to the cloud. We have decided to use ThingSpeak in order to upload all of the user's health information into the cloud. We have decided to go with this cloud platform because it doesn't require much processing power for the raspberry pi to offload the user's information into that platform. We also plan on integrating different charts into ThingSpeak which will display a user's health information over time. The reason we plan on doing this is because it will group the user's health information in a neat and organized manner.

Another prototyping section that we have begun but have not yet completed is the pcb design and the 3D printed design for our project. We have created a schematic for the Healthy-Gamer device on KiCad which can be found in section IV figure 10. Once we do some more testing with our current prototype we will then be able to figure out how we can design our pcb so it can compactly fit all the electronic components into a small enclosed factor that can fit on a user's wrist. See figure 18 for a screenshot of the model.

The lastest section that we implemented is an SMS feature to alert the user of any critical values. This feature was successfully able to send a text message to a phone whenever the Node.js code was called. See section VI. Test Performed for a sample video this code being run.

VI. Tests Performed.

After finalizing all the designs, we moved on to perform some tests to verify that everything is working as expected. All the members were assigned to work on a particular part of the device and they performed their individual tests solo. After verifying that the individual parts were working, we then moved on to integrate all the parts and checked for the complete functionality of the device.

The first test that we performed measured values obtained by the accelerator from wrist movements, and by the temperature sensor. Then we sent those values to the monitor. The picture below demonstrates the output we obtained after moving our wrists haphazardly. The temp value shows the temperature in the room where we were performing the tests. Theta shows the angle of the wrist. As the device was just laying flat on the table, it was giving us a value close to neutral or 0. As we moved the device around, theta values kept changing however, the temp value stayed
the same.

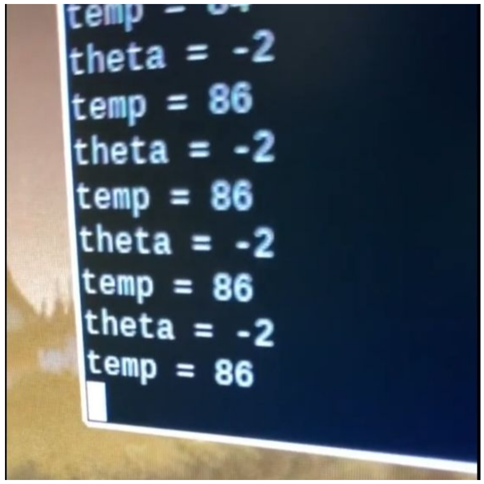

Figure 13: Temperature and Angle Readings

In the second test we performed, we wanted to see whether we could output the value derived from all the sensors to the LCD display. In the first test, we had confirmed that the sensors were working as expected. Now, we wanted to make sure that the LCD display was working in integration with other parts of the device. The figure below was taken during the 2nd test and the video is included in the prototyping video.

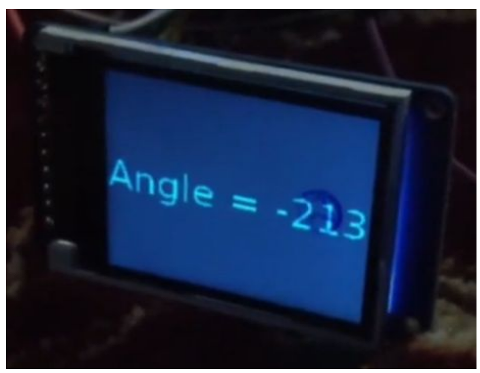

Fig. 14. Screenshot the LCD working and displaying a wrist Angle.

Here is a link to our initial prototyping progress video detailing these early efforts https://www.youtube.com/watch?v=kHXUFNpBC6k&ab_channel=MoneebAhmad

The figure below shows the heart rate results that are obtained from the pulse rate sensor. If the sensor detects that a user's BPM is above 100 it will send an alert to the user letting them know to take a break from their gaming section.

Fig. 15. Sample output for pulse sensor.

Link to prototyping video:
https://www.youtube.com/watch?v=vG8EGvoUSFI&ab_channel=MoneebAhmad

In our 4th test we wanted to be sure that our SMS code could properly send a message to one of our phones. A screenshot cannot properly convey how the SMS feature works so the below link is a demo video for it.
https://www.youtube.com/watch?v=qaZ_8nI56wY&ab_channel=MoneebAhmad

VII. Experimentation plan.

This project contains sensors, a power supply, an LCD display and microcontroller. There will be two experiments to test the readiness of the final product: operational requirement evaluation and functional requirement evaluation.

Experiment 1 (Operational Requirement Evaluation):

Goal: To evaluate a user's input and selection using a push button

System Components: LCD, Sensors and Push Button

Testing Process: A user can select through different modes on the Psychological device by clicking on a button in order to see their health data while gaming. The user will be able to click on the button multiple times in order to see the different reading that the device is displaying.

Evaluation: We will verify that once a user pushes the button the LCD on the device it has changed modes in a proper cycle.

Experiment 2 (Functional Requirement Evaluation)

Goal: To evaluate measurements of a user's: stress level, heart rate, body temperature, and wrist movement.
System Components: Sensors, Microcontroller.

Testing Process: We will be using the sensors that are encased in the device in order to measure a users physical and psychological health while they are gaming. The Microcontroller will convert the data that is collected from the sensors into readable data that the user can view.

Evaluation: We will verify that the data that is being collected from the sensor is accurate by using 3rd party health equipment (ie; apple watch) in order to measure a user's health and then compare it to our device reading. This test process will be conducted multiple times in order to assure that the Psychological device that we are creating is consistently outputting the correct information about a user's health.

VIII. List of tasks for 493 and their decompositions into subtasks plus individual responsibility:

- **PCB design:** We have used KiCad to design the schematic we will be implementing on a PCB board for our project. We will be hierarchical schematics so that it will enforce organization to our schematics and help us guide through the designing process. To make it aesthetically pleasing, we are going to make it as neat and clean as possible so that whoever tries to follow the pcb design has a good grip on how to integrate individual devices without confusion. Before placing the sensors in the PCB board, we will verify our circuit board layout by running a design rule check (DRC). Another goal for this part of the project is to make design as small as possible. This will help us fit everything into a compact, light-weight watch which will be feasible to wear anywhere.

- **Cloud server:** In order to store the data we collect from our device, we will use ThingSpeak as a cloud server. ThingSpeak is an open IoT platform for monitoring data online. In ThingSpeak channel we can set the data as private or public according to our choice. ThingSpeak generally takes 15 seconds to update the readings [3]. After creating an account, we will open our own channels where the uploaded data will be visible. Each channel will contain different data collected from the sensors. For example, one channel will contain the body temperature information and variation. Another channel will store the body movement information. Our unique code will calculate the stress level from the sensors and will upload it simultaneously. The users will have access to this information anytime they want to check for it.

- **3d printed case -** after we are done with our exhaustive testing, we will be using a 3D printer to create our own case that will hold our device. We have already designed our case using "Blender" and created the STL file [4]. It will take approximately 3 days to form the 3D printer to build the design.

- **Testing -** We will be conducting extensive testing on our device to make sure the device is performing correctly and efficiently. We will make sure all the sensors are working properly and the connections are in correct order as well as safe. We will monitor how effectively the device is uploading data to the cloud server and if the data sent is accurate. In addition, we will be testing the device on different types of users from different age categories to make sure the device functions properly and is able to differentiate between different users. We will be testing the integrity of the device in various circumstances to see if the device is securely able to stay on the user's body as well as perform the tasks correctly.

Moneeb is in charge of integrating the individual sensors and devices with the Pi Zero. He was in charge of presenting the State Diagram for the video, and then for our second video, the Milestones section too. He is also in charge of taking the ADC value from the thermistor and turning it into a proper temperature. He will be spending his time debugging and implementing codes using the Pi as it has a co-math processor which can handle floating point with square roots and tangents efficiently. He is implementing the code for the circuit and will be working on rigorous testing so achieve maximum efficiency. Moneeb is also building a SMS notification feature for the system which will allow the Raspberry Pi to send a text message alert whenever a critical value is detected on the system which will help users prevent potential dangers.

Saad is in charge of creating the Cloud server and making sure that the data that has been obtained from the sensors will be able to store it in the server. He has also worked on integrating the heart rate sensor into the Healthy-Gamer device. The goal of the heart rate sensor is to keep track of the user's heart rate and alert them if it is at a high level over an extended period of time. He has already created the server and made sure the heart sensor works properly. He will be testing the sensor rigorously and create an algorithm so that it can detect any irregularities in the heart which will alert the user.

Jamil and Priyam were assigned to do the software and hardware design, experimentation plan, and Gantt chart. They worked on integrating the LCD with the device and making sure it is displaying the output of the acquired values from the sensors. Priyam has built a 3D model for the physiological device using Blender. They will make sure the device is light and designed efficiently so that it does not put the user into any discomfort. They will be testing the device thoroughly and make sure the device provides accurate and updated data all the time.

Aayush is in charge of the experimentation plan and to measure stress levels by using his research on Carpal Tunnel Syndrome as well as analyzing Heart Rate Variability. He has also made sure the accelerometer is working accurately and will test the device so that it provides accurate measurement of the position of the user. He is working on trying to make the sensors give real time values on graphs to allow the users understand their data. He will analyze the accelerometer data to determine the range of motion of the user and detect what is a fall or normal user motion. This will minimize the occurrence of any false alerts when the user is active.

Ryan is in charge of prototyping and the prototyping board that will be used to mount the raspberry pi zero. The device will have three layers of PCB. The first layer is the pulse sensor, accelerometer, battery, and raspberry pi mounting hole. The second layer would be the raspberry pi zero. The third layer (upper layer) would be LCD. He will be making sure the device is intact and the connections are well prepared so that any movement of the device and the user does not interfere with the connections and reading of data.

IX. Schedule and Milestones.

Task	Start Date	End Date		Duration(days)
1. System Integration	11/16/2020	12/22/2020		36
1.1 Initialize setup	11/16/2020	11/22/2020		6
1.2 Mode Selection	11/22/2020	11/28/2020		6
1.3 Sensor Functionality	11/28/2020	12/4/2020		6
1.4 Calculation functionality	12/4/2020	12/10/2020		6
1.5 Display Functionality	12/10/2020	12/16/2020		6
1.6 Server Functionality	12/16/2020	12/18/2020		2
2. PCB Design	12/18/2020	12/20/2020		2
3. 3D printed case	12/20/2020	12/24/2020		4
4. Testing	12/24/2020	2/1/2021		39
4.1 Experiment 1	12/25/2020	1/15/2021		21
4.2 Experiemnt 2	1/15/2021	2/1/2021		17
5. Reporting	2/1/2021	3/1/2021		28
5.1 Progress Report	2/1/2021	2/10/2021		9
5.2 In-progress report	2/10/2021	2/20/2021		10
5.3 Final Report	2/20/2021	3/1/2021		9
6. Milestones/Demos	3/1/2021	5/1/2021		61
6.1 Demo#1	3/1/2021	3/20/2021		19
6.2 Demo#2	3/20/2021	4/10/2021		21
6.3 Demo#3	4/10/2021	5/1/2021		21

Fig. 16. Schedule of Tasks for ECE 493

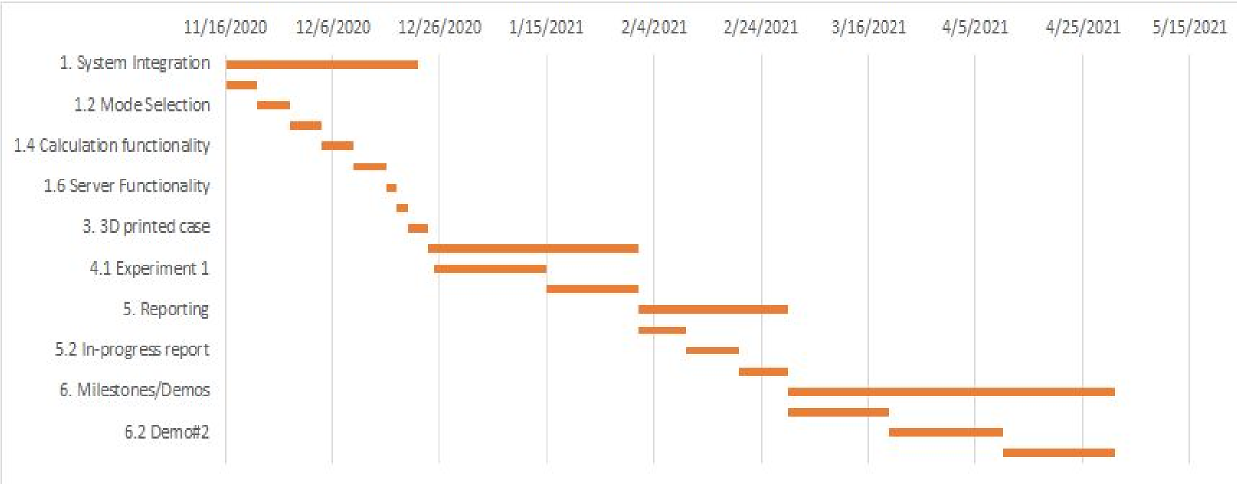

Fig. 17. Gantt Chart for ECE 493.

X. Summary of the project.

For our project we wanted to create a health device that was designed specifically for gamers. The device would be able to measure a user's body temperature, wrist angle, heart rate and stress level. Body temperature would be monitored to ensure that a user is staying well hydrated. Wrist angle check for and to prevent early signs of CTS. Heart rate and stress level would be used to advise the user to take a break from. The device would be able to alert the user of this important vital information through an LCD screen as well through a text message alert. A user would also be able to access their information through an off-site server, which they can review at a later time.

XI. Links to important files:

Design Review Presentation:

https://www.youtube.com/watch?v=ZtVC0dlqUE4&ab_channel=MoneebAhmad

Early Prototyping Video:

https://www.youtube.com/watch?v=kHXUFNpBC6k&ab_channel=MoneebAhmad

Heart Rate Prototyping

https://www.youtube.com/watch?v=vG8EGvoUSFI&ab_channel=MoneebAhmad

SMS Prototyping

https://www.youtube.com/watch?v=qaZ_8nI56wY&ab_channel=MoneebAhmad

3D model:

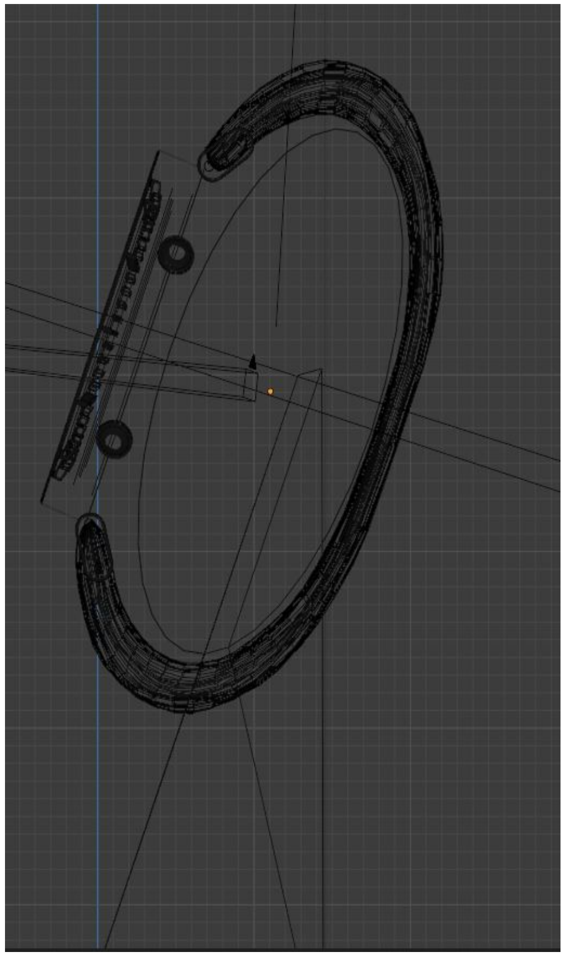

Figure 18: 3D Model of the Device

Link to .stl file:
https://drive.google.com/file/d/1yPVlIbf0Pa-xF3l3GcpZ_00di7R4Yaoi/view?usp=sharing

XII. References:

[1] J. M. von der Heiden, B. Braun, K. W. Müller, and B. Egloff, “The association between video gaming and psychological functioning,” *Frontiers in psychology*. Jul. 2019, DOI: 10.3389/fpsyg.2019.01731.

[2] “Carpal Tunnel Syndrome Fact Sheet,” *National Institute of Neurological Disorders and Stroke*. Apr. 2020, [Online]. Available: https://www.ninds.nih.gov/Disorders/Patient-Caregiver-Education/Fact-Sheets/Carpal-Tunnel-Syndrome-Fact-Sheet

[3] “How to Send Data to ThingSpeak Cloud using Raspberry Pi.” Jan. 2019. https://iotdesignpro.com/projects/how-to-send-data-to-thingspeak-cloud-using-raspberry-pi .

[4] Dibya Chakravorty, “STL File Format (3D Printing) – Simply Explained,” All3DP, Jun. 13, 2017. https://all3dp.com/what-is-stl-file-format-extension-3d-printing/.

[5] L. Haghshenas and M. Pooyan, "Investigating the Effect of Computer Games on the Heart Signal," 2019 27th Iranian Conference on Electrical Engineering (ICEE), Yazd, Iran, 2019, pp. 1775-1778, doi: 10.1109/IranianCEE.2019.8786686.

[6] H.-G. Kim, E.-J. Cheon, D.-S. Bai, Y. H. Lee, and B.-H. Koo, “Stress and Heart Rate Variability: A Meta-Analysis and Review of the Literature,” Psychiatry Investigation, vol. 15, no. 3, pp. 235–245, Mar. 2018, doi: 10.30773/pi.2017.08.17.

[7] Abyarjoo, Fatemeh & Barreto, Armando & Abyarjoo, Somayeh & Ortega, Francisco & Cofino, Jonathan. (2013). Monitoring Human Wrist Rotation in Three Degrees of Freedom. Conference Proceedings - IEEE SOUTHEASTCON. 10.1109/SECON.2013.6567517.

[8] M. Bahrudin, “Carpal Tunnel Syndrome (CTS),” Saintika Medika, vol. 7, no. 1, Oct. 2012, doi: 10.22219/sm.v7i1.1090.