**Perfect Posture**

**ECE 445 Design Document**

Apoorva Josyula, Rohan Kanianchalil, Julianna Gecsey

[jgecsey2@illinois.edu](mailto:jgecsey2@illinois.edu), [josyula3@illinois.edu](mailto:josyula3@illinois.edu), [rohansk2@illinois.du](mailto:rohansk2@illinois.du)

Group 38

TA: Daniel Vargas

Spring 2021

**1 Introduction**

* 1. **Objective**

In the time of Covid-19, we are seeing a rise of the “WFH” lifestyle as more and more non-essential workers are having their offices be shut down and their work be moved to a remote setting. This sudden change of workplace is creating a new sort of health crisis to occur. According to an April Facebook survey from the American Chiropractic Association, 92 percent of chiropractors (out of 213 respondents) said that patients report more neck pain, back pain or other musculoskeletal issues since the stay-at-home guidance began [1]. The most common posture related issues that are arising are coming from the upper back to neck area, Scott Bautch, the president of the American Chiropractic Association’s Council on Occupational Health, says that as screen time has exploded, we’re more at risk of “Text Neck” and “Selfie Elbow” [1]. We recognize that this issue will only continue to get worse as more jobs continue to become remote, requiring more time to be spent sitting in front of a desk. Our solution is to introduce our device “Perfect Posture”. Through the means of the resources and time we get in ECE 445, we plan to create a wearable posture device that can be attached to a user’s back. This device will accurately assess a person’s posture and the alert the user on the potential necessary adjustment they will need to keep their posture in check. The overall goal here is to help train a user to recognize back problems and eventually create the habit of working with perfect posture.

* 1. **Background**

The products for creating “perfect posture” devices are endless, from back braces to even single system wearable devices that one can wear right on their back. These products however all lack in holistically assessing a person’s posture. When it comes to physical braces, the external mechanics of a back brace force a person’s body/muscles to rely on the physical weight of the brace to hold up their back posture. This can be harmful as the body will become dependent on that physical brace to help keep them upright and could led to in some cases muscle dystrophia with the lack of use of muscle in the back region [2]. Other wearable smart devices, such as the “Upright Go” are helpful in eliminating the need for a physical back brace however, they typically only focus on one aspect of a person’s spine. As studies have shown, the result of bad back posture is not just isolated to one specific area of a person’s spine but has to deal with the holistic approach of maintaining one’s spinal curvature [3]. These posture devices don’t take into account how a person’s spine can be held “upright” but fail to maintain a healthy curvature of the total spine [3].

This is one of the many reasons why we have created our device to not just be a posture sensor that detects whether a person’s upper spine is being held upright, but to calculate a person’s spinal curvature to see if they are maintaining the healthy amount of tension and bend in their upright position [3].

When we talk about curvature, we are looking at how the spine looks, from a side viewpoint. The whole spine itself, from the top of the cervical spine to the bottom of the lumbar spine, all form an important side view curvature that resembles an S-shape [4]. For the most part, there is a lot of focus on worrying about lower back problems, and thus most products are looking at the lower back angles to fix spinal curvature shape and position. What many products neglect to realize that is there are very few products that focus on the upper spine curvature and shape. We are specifically talking about the shape and position of the thoracic spine, one’s upper-middle spinal column. The thoracic kyphotic curve has the shape of a normal C, which is proportional to all humans. A majority of the modern-day posture issues such as text-neck, hunching, and selfie elbow stem from the lack of maintaining that thoracic kyphotic curve [4].

Diagram

Description automatically generated

Fig 1. Kyphotic and lordotic curves of the vertebral column [5]

The thoracic (rib cage) portion of the spine has a normal forward curvature, which has a normal range (20 to 50 degrees). Any exaggerated rounding of the forward curvature in the upper back is called hyperkyphosis (meaning too much kyphosis), but the term "kyphosis" is commonly used to refer to the clinical condition of excess curvature of the upper back (greater than 50 degrees), leading to a stooped forward posture [5].

Our product will help to maintain a healthy range of curvature and bend in this area by ensuring that a user does not exceed the range of 50 degrees or below the range of 20 degrees.

We recognize that it is humanly impossible to never move from your given posture, there will be times that will require a person to slouch for a bit. Our device shouldn’t be going off every second if a person were to just bend down to pick up a pencil. Thus, there is leeway in how long a person can be in a certain position without it being considered “detrimental” to that person’s given posture.

Another factor to consider is how long a user should sit at their desk with a device attached to them. What most researchers have found is that a low-risk time spent sitting at your desk is typically around 30 minutes with a max interval of 4 hours all together [5]. In order for our device to encourage holistic posture health, we will allow for our device to remain operational for a max of 4 hours.

With the knowledge of how we need to be maintaining these angles and for how long can time elapse before a person engages in bad posture, we as ECE students will create a device that will accurately be able to detect deviances from these important angles and create a system that will help alert the user on when they have been continuously in a bad posture.

**1.3 High Level Requirements**

* Sensor system is placed on a user’s thoracic spine to detect if a user’s degree of spinal curvature exceeds 50 degrees, with an error margin of +/- 2 degrees.
* Device relays data taken from sensors to control unit to calculate if the feedback system needs to be alerted with a repeated alert interval of 2 minutes, information transmitted has a max delay of 500ms.
* For device to not interfere with a user’s natural spinal position, weight of device should not exceed more than 150 grams.
* Users are able to read their data through a computer app that will visually show their posture readings in a given sitting with a max usage range of 4 hours.

**2. Design**

**2.1 Block Diagram**

Our wearable device will require the following four subsystems to be considered successful: A sensor/feedback system, a control unit, a power system, and an app interface. The layout is shown below in Fig. 2. The power system will ensure that the system can be powered for a healthy 4-hour period of time the user might be spent working with the proper 3.3V output. Our sensor system will help to read a person’s spinal position and our feedback system will help to alert the user when that spinal position is off and will repeat that alert for in a steady 2-minute interval. The control unit will consist of our microcontroller that will parse and process the sensor data and send bad posture alert to the feedback system as well as sending processed data via a Bluetooth module to the user app interface. Our app interface will have the users posture data in an interactive way for the user to see the trends in their posture.

Diagram

Description automatically generated

Fig 2. Block Diagram of Perfect Posture

**2.2 Physical Design**

Diagram

Description automatically generated

Fig 3. Physical Design

As we mentioned before, our goal is to ensure upper back posture, so our placement will be along the cervical/thoracic spinal column. The curve of the thoracic kyphotic curve can be measured anywhere from the range of the C7 spinal disk to the T10 spinal disk, so for our sakes we will be focusing on placing the sensors from the expanse of the C7 to the T8. That roughly translates to placing our sensors at bottom nape of the neck, the middle of the shoulder blades, and at the middle of the back.

An important thing to note about our physical design is that we want to keep this relatively small. The individual sensor components and vibrational motor will be no large than a 4x4cm design. Each of our components will be encased in a 3-D printed plastic case. The control unit and battery pack will have the leeway to be a bit larger (around 11x8cm design). The weight in order to not affect the individual posture will be no larger than 150 grams. We will use a soft textile to run our wires through. That will connect the different components without irritating the users back. The attachment to the actual person’s back will consist of a skin safe adhesive a user can either attach directly to their back or on top of their clothes.

**2.3 Block Descriptions and Requirements and Verifications**

2.3.1 Senor/Feedback System

For our sensor subsystem, we decided that the best way we could measure angular data was with the combination of a gyroscope and accelerometer. The combined data of the raw acceleration values from the accelerometer and the angular velocity values from the gyroscope can be used to figure out angular position at a particular place.

Three sensor units will be placed along the spine to ensure we can read an accurate picture of how a person’s thoracic kyphosis curve looks like on their upper back. The sensors will relay their positional data to the microcontroller, where it will store the changes of values when there is significant movement.

Our choice design for the sensor system was to use the “Triple Axis Accelerometer and Gyroscope Breakout from Sparkfun” (MPU-6050).

Diagram, schematic

Description automatically generated

Fig 4. Circuit Schematic for Triple-Axis Accelerometer/Gyroscope (MPU-6050) [7.1]

There were many benefits to using this specific breakout. For physical design we want to keep the sensors small, and the dimensions for this board is 25.5 x 15.2 x 2.48 mm, keeping true to the size aspect we need to reach. The embedded algorithms included help for compass calibration so there isn’t any user intervention required to calibrate initial direction [6].

The I2C digital output pins are able to transfer data from a rotation matrix, quaternion, Euler angle, or raw data format, so when connected to our microcontroller the data transfer from the I2C bus can be specified to our needs for how we will utilize the data to read angular position. For our purposes we are going to use raw data format to feed into our microcontroller algorithm [6].

Since a user’s typical posture moves at a significant slow rate, our sensors can run at a frequency at a moderate rate to detect slow gradual movements. The sensor we will be using will the MPU-6050 which samples at a rate of 100 Hz in low noise mode, which will be more than enough for us to determine an appropriate angle measurement of a user in sedentary state [6].

The feedback system will be our vibrational motor. The vibrational motor will be in charge of alerting the user whenever their posture is off. It will receive either a High signal (indicating the posture is off so turn on motor) or Low signal (posture is fine for now so stop motor). The vibrational motor will be set off again whenever the time spent in the bad posture continues to increase. As the vibrational motor will be connected to our microcontroller, it will be in charge of timing the HIGH signal to maintain voltage input for about 10 seconds.

Diagram

Description automatically generated

Fig 5. Circuit Schematic for Vibrational Motor (ROB-08449)

|  |  |
| --- | --- |
| Requirement | Verification |
| 1. Accelerometer/Gyroscope must be able to send raw x,y,z and g force data to the microcontroller via I2C. 2. MPU-6050 must be able to sample data at a rate of at least 100Hz. | 1. See if microcontroller receives packets of sensor data from the MPU-6050. 2. Test accelerometer by first powering the device to a 5.0V power in the VCC pin, ground to GND pin, and the data pins onto our ATMega328p chip. 3. Connect Microcontroller to computer and let it be connected to an Arduino Serial Monitor. Check to see if connection has been made between the two parts. 4. Write program that will print out the raw g values and degree values from sensors onto the monitor. 5. Lay down accelerometer without moving and check to see if reading is saying 0.5g in the X, Y axis and 1g in the Z axis (with variation from 0.2g). 6. Rotating about the x-axis, rotate accelerometer 90 degrees, and check to see if the reading is showing change of Y axis to be 1g and see if the value is kept at 1g for 2 seconds. 7. Rotate to 180 degrees now, and check if the values now read, 0.5g, 0.5g, 0.0g in the x,y,z plane and are steady at that reading for more than 2 seconds. 8. Repeat process to see if values for angle reading from gyroscope read the angle values of its turn in the right orientation. 9. Check 100 samples per second on average is being collected from MPU-6050. 10. Load program for reading timestamps onto ATMega328p 11. Connect MPU-6050 to microcontroller. 12. Assure power connections. 13. Start recording data for 20 seconds. 14. Check data collected on microcontroller. If the number of samples that was collected was greater than 2000 samples than data was successfully recorded. |

*Table 1: R&V table for MCU-6050 sensor*

|  |  |
| --- | --- |
| Requirement | Verification |
| 1. When microcontroller sends HIGH signal to motor, the motor will vibrate for 10 seconds to indicate bad posture. | 1. Connect vibrational motor to a 5V   power source and connect to our microcontroller chip as well.   1. With our microcontroller connected from USB to our computer, we will write an Arduino program that is going to turn on the digital pin to HIGH for 10 seconds and then LOW for 10 seconds. 2. Hold vibrational motor feel for vibrations when switched onto high. 3. Record the time the vibrational motor is held on HIGH to see if it stays on for 10 seconds. |

*Table 2: R&V table for Vibrational Motor*

2.3.2 Power System

For our power system we had to consider three main things: the charge time of continuous use of our device. How much operational voltage and current each component drew from the power supply, and the weight of the overall battery back.

To power most of our components of our circuit they require a typical 3.3V input for operational use, and the maximum current usage of 150mA (mostly for Linear voltage regulator). For that reason, we chose to stick with a Li-Po battery that had a nominal voltage supply of 3.7V with 2500mAh knowing most batteries don’t operate at their ideal battery life.

Our sensors, vibrational motor, Bluetooth module, and microcontroller all need a voltage output of 3.3V, so we chose to include only one linear voltage regulator. The linear voltage regulator is to help ensure all our components are being powered safely without burnout from high voltage inputs.

Diagram

Description automatically generated

Fig. 6 Circuit Schematic for Li-Po Battery (DTP502535) [7]

Diagram, schematic

Description automatically generated

Fig 7. Circuit Schematic for Li-Po charger (MCP73833/4) [8]

Diagram

Description automatically generatedChart

Description automatically generated

Fig 8. Circuit Schematic of Linear Voltage Regulator (LD1117D33TR). [9] Fig 9. Pin Connections [9]

|  |  |
| --- | --- |
| Requirements | Verification |
| * Charger should be outputting 4.2V +/- 0.03V at 400mA +/- 10mA when plugged into USB charging plug. | 1. Plug in charger from USB end to computer to start charger. 2. Connect an oscilloscope to the charger. One terminal is at the negative end of the charger, while the other terminal is at the positive end. This will help us to measure voltage across the charger. 3. Unplug oscilloscope and plug in ammeter in series with the charger. 4. Once in series, measure current through and check to see if there is a steady output current of 400mA +/- 10mA. |
| * Charger should stop outputting voltage supply to Li-Po battery once battery reached max capacity at 3.9V +/- 0.03V | 1. Connect an uncharged Li-Po battery to Li-Po charger, for a full charge of the Li-Po battery at the current rate of 400mA, that should take from anywhere from 50 minutes to 70 minutes. 2. Taking the oscilloscope, probe the charger across the output terminals of the charger. One terminal end is at the positive terminal end of the charge, while the other is at the negative terminal end of the charger. 3. Once battery is fully charged, the output voltage across the charger should be 0V. 4. Remove the oscilloscope, and place on ammeter in series with the charger. This reading of the current, once the battery is fully charged should read 0A. |

*Table 3: R&V table for Li-Po Charger*

|  |  |
| --- | --- |
| Requirement | Verification |
| 1. At full charged capacity, battery should be outputting a voltage charge of 3.7V +/- 0.2 V at 400 mA for maximum current 2. Fully charged, the battery should last around 4 +/- 1 hours with continuous usage. | 1. Use an uncharged Li-Po battery, and plug into a Li-Po charger. The LED indicator on the Li-Po charger will show when the Li-Po battery has been fully charged (around 50 to 70 minutes).    1. Once charged, connect an oscilloscope to the battery (one end to the positive terminal of battery, other end to the negative terminal of battery).    2. Graph the voltage readings to ensure, the output voltage is around 3.7V +/- 0.2V.    3. To measure current output, with the oscilloscope already connected, connect the battery to a 10ohm resistor, and check current output on graph.    4. To verify the usage of the battery, with the full circuit’s components connected, keep oscilloscope plugged into the ends of the battery as described above.    5. Check to see where the oscilloscope current readings and voltage readings are at. 2. Given under the specs of the battery, the battery should run at 400mAh at continuous use, so divide total current use of the circuit by 400mAh to determine, length of time the battery is able to sustain continuous use of the device.    1. To verify the usage of the battery, with the full circuit’s components connected, keep oscilloscope plugged into the ends of the battery as described above.    2. Check to see where the oscilloscope current readings and voltage readings are at.    3. Given under the specs of the battery, the battery should run at 400mAh at continuous use, so divide total current use of the circuit by 400mAh to determine, length of time the battery is able to sustain continuous use of the device. |

*Table 4: R&V Table for Li-Po battery*

|  |  |
| --- | --- |
| Requirements | Verification |
| 1. Given a voltage of 3.7 +/- 0.4V, our linear voltage regulator must output a voltage of 3.3 +/- 0.05V. | 1. Use a function generator to supply the given voltage output. The output from the voltage generator should be 3.7V +/- 0.4V. 2. Attach an oscilloscope where one end is attached to the positive terminal of the voltage regulator, and the other end is attached to the negative terminal of the voltage regulator. 3. Check the graph results from the oscilloscope to see what the output voltage from the linear voltage regulator is showing. The results should be around 3.3V +/- 0.05V |

*Table 5: R&V Table for Linear Voltage Regulator*

2.3.3 Control Unit

The components of our control unit will consist of a microcontroller chip (ATMega328) along with our wireless transmission, which in our case will be our Bluetooth Module.

Our microcontroller will be calibrated to receive and parse the raw data from our sensors and turn them into readable angular position data. It will be programmed to keep track of when data coming in from the sensors are outside the given range of what a healthy curve in the spine is supposed to be (anything exceeding 50 degrees). It then must be in charge of sending a signal to our feedback system for when the posture is out of place in two minutes intervals.

Our choice design for our microcontroller was to use the ATMega328 chip. One reason was because of the range of baud rates it provides that can accommodate the data rate for our application use. The range it provides can go from 2400 bps to 230.4kbps which is well within what we need.

Diagram, schematic

Description automatically generated

Fig 7. Circuit Schematic for AtMega328 Microcontroller

All of our angular data information will also be sent to our Bluetooth module so that when a user wants to pair their smart device, they can visually see the changes being made in their posture and how much they improve each time they use the device.

The Bluetooth module is responsible for receiving digital values that it gets from our microcontroller and sending it to a user’s paired device, through the form of a front-end app. The communication with our microcontroller will be through a Universal Asynchronous Receiver/Transmitter (UART).

From our front-end app, when a user selects to start recording data, it will start up a program that will collect data from the Bluetooth TXD pin. Our app can also make our control unit to stop collecting data once a user is done using the device. The program will send a signal that will cause the microcontroller to stop receiving sensor data through our Bluetooth module via RXD pin.

Diagram, schematic

Description automatically generated

Fig 8. Circuit Schematic for Bluetooth Module (RN-41) [10]

The exact Bluetooth module we will be using will be the RN-41 Bluetooth Transceiver Module. The reasons for us choosing a Bluetooth module over Wi-Fi was mostly for the reason that if a user is sitting at their desk, they are most likely close by their phone/computer so the distance would probably not be farther than 5 meters. The continuous data transmission from our microcontroller is also not larger than 1 Mbps, which the RN-41 can handle, transferring data at speeds of 3Mbps in 10–20-meter range at a low-power consumption. The Bluetooth module saves more energy and space for our application. The main protocol that will be used is the L2CAP protocol.

|  |  |
| --- | --- |
| Requirements | Verification |
| 1. Be able to read raw sensor data and calculate an angle approximation via I2C. | 1. Attach our sensors by the analog pins on the microcontroller. 2. Flash user defined code onto microcontroller that will take inputs from sensors as inputs to angle calculator function. 3. Place sensors at predetermined angles (using protractor for best angle reading). 4. On the Arduino serial monitor, wait to see if the function code will return the predetermined angles. 5. Using an oscilloscope probe the negative and positive output terminals of microcontroller. 6. Determine on graph to see frequency rate. |

*Table 6: R&V Table for Microcontroller (ATMEGA328)*

|  |  |
| --- | --- |
| Requirements | Verification |
| 1. Be able to communicate with microcontroller via UART at a speed of 1.125kBd. | 1. Open up terminal on computer and attach microcontroller to a USB UART bridge 2. Have terminal receive at 1.125kBd. 3. Have set characters to be sent and return back. 4. Check to see if all characters are received from and match the set characters that programmed. |

*Table 7: R&V Table for Bluetooth Module*

**2.4 Software flow**

The following flow chart shows the software flow diagram of how our microcontroller will be receiving and transmitting angular data within the device. We will flash our microcontroller with the code to run the following loop, until user shuts down device.

Diagram

Description automatically generated

Figure 9: Software Flow Diagram

**2.5 Tolerance Analysis**

One key piece of information that we need to keep track of throughout our design process is the actual angle of the user's spine and how far deviant it is from the normal spine angle. As we talk about the appropriate angle to keep the thoracic kyphotic curve, it’s important to note that the angle used by medical professionals to determine a healthy curvature is called the Cobb angle. Cobb angle can be used for many different measurements across the spinal column. Since we are focused on the upper back, we are going to be looking at how the Cobb angle be used to detect hyperkyphotic [11].

As previously state, a healthy degree of curvature for this angle is anywhere from a range of 20-45 degrees, with 50 degrees being the exceeding point.

Diagram

Description automatically generated

Fig 10. Calculating Kyphosis angle

With our design configuration of using three sensors placed along the thoracic spine, we can get the kyphosis angle from the top and bottom sensor reading, while our middle sensor will ensure that an appropriate 90-degree angle is being held.

The software module we will be having our microcontroller be in charge of will be responsible for helping take the raw data angles and convert them into the angular data we need. Our accelerometer’s raw g-force values will not be that helpful, but we can use them to determine orientation about the axis with combination of raw x, y, z axis reading we get from the gyroscope.

Text, letter

Description automatically generated

Eq 1: Angles about the x and y axis

These calculations will then be relayed to our microcontroller where the user will be notified if their posture is off and whether their current position may lead to potential back problems. In a medical environment an X-Ray would be necessary to garner the exact severity of one's spine, however, we aren’t necessarily treating spinal injuries, instead we are designing a product that will indicate deviations in an attempt to stop those changes before permanent damage occurs.

**3 Cost and Schedule**

**3.1 Cost of Labor**

When organizing an analysis on the cost of our labor we find that assuming we work approximal 10 hours a week, with a team of 3 students, at a standard rate of $35 an hour, our labor would equate to $16,800 over the course of the 16-week semester.

**3.2 Cost of Parts**

The price of labor is not the only cost in this project, we will also have equipment to be purchased in order to build out our project. A summary of the costs has been calculated below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Part** | **Quantity** | **Cost per Unit** | **Total Cost** |
| Adafruit LSM6DSOX 6 DoF Accelerometer and Gyroscope - STEMMA QT / Qwiic | 2 | $11.95 + shipping | $ 25.90 + shipping |
| DZS Elec 4pcs Mini Vibration Motor with 35mm Length Lead Wire DC 3V 85mA 12000rpm Flat Button-type Motor For Mobile Phone Tablet Bluetooth Mini Fan Electronics Appliances 10x2.7mm | 1 | $5.49 | $5.49 |
| HTRC LiPo Charger 2S-3S Balance Battery Charger 7.4-11.1V RC B3AC Pro Compact Charger(White) | 1 | $13.97 | $13.97 |
| 3.7V 3000mAh 105151 Lipo Battery Rechargeable Lithium Polymer ion Battery Pack with JST Connector | 1 | $12.49 | $12.49 |
| Linear Voltage Regulators 1.2-37V Adj Positive 1.5 Amp Output | 1 | $0.56 + shipping | $0.56 + shipping |
| ATmega328 Microcontroller | 1 | $1.92 + shipping | $1.92+shipping |
| RN-41 Arduino Wireless Bluetooth Receiver RF Transceiver Module Serial Port Transmitter Module | 1 | $10.57 | $15.57 |
| Double Sided Medical Grade Adhesive Tape Roll, 1 Inches x 108 Inches-Clear | 1 | $6.02 | $6.02 |
| Fabric Wire Encasing (cut and sew old shirt) | 2 | $3 | $3 |
| 3D Printed Box for Electronics | 3 | $0 | $0 |
| **Total** | | | **$84.92** |

*Table 8: Cost Table*

Due to the nature of our project, we will not need to have any work from the ECE shop besides the printing of the PCB which will cost around $20 because of the small size. With all of these costs added up the complete total of our project will cost $16,904.92.

**3.3 Schedule**

|  |  |
| --- | --- |
| **Week** | **Plan** |
| 3/1 | Present for design document review, take in feedback and modify the design document to be submitted that week. Begin working on a design review project for next week. |
| 3/8 | Finish the design review and focus on working on initial PCB prototypes to ensure we have a solid design before the deadline. Additionally, we will begin purchasing equipment for testing. |
| 3/15 | Submit PCB design to be made and finish the weekly assignments. If parts have come in, begin to work on testing results and modify the project based on the effectiveness of parts. |
| 3/22 | Correct any changes that need to be made to our PCB design and submit them to ensure that our order will be made in time. |
| 3/29 | Begin to piece together all pieces of equipment for on body testing to perfect the sensors ability to work as we are hoping. |
| 4/5 | Correct all final issues that need to be made to our PCB design for the final PCB order. Continue with collecting and transmitting data cleanly to begin wrapping up our project. |
| 4/12 | Focus on the aesthetic and user interface with the project by printing 3-D casings for sensors and controllers to improve the look. |
| 4/19 | Have the project all wrapped up and finalized/working to present at the mock demo presentations for feedback and last-minute tweaks. |
| 4/26 | Focus on presenting our demonstration. Begin preparing for our presentation as well as begin the final lab paper. |
| 5/3 | Present our presentation and finish up our final paper and submit to finish up the semester. |

*Table 9: Schedule*

Given how short the semester is we are expecting our project to be very all hands-on deck. When deadlines are approaching fast, we will work together to ensure all parts are completed timely and correctly. Beyond that however we will each have our own specialties of course:

* Julianna Gecsey
  + PCB design
  + Ordering parts
* Apoorva Josyula
  + Coding the user interfacing App
  + 3-D printing parts
* Rohan Kanianchalil
  + Initial sensor testing

Through this schedule we will be able to efficiently get our project done.

**4 Safety & Ethics**

When creating our project, we have to consider a few safety issues as well as some ethical as well.

**4.1 Safety**

Starting with the issues that could possibly occur during the development of the project is the usage of 3 accelerometers and gyroscopes. With that many being in use, if we are not careful with how things are wired and powered, it may cause a malfunction in our project. Continuing on, with the usage of a microcontroller and a Bluetooth module, if we miscode the microcontroller, we may cause an adverse effect in our project. We must also consider how the accelerometer and gyroscope may affect the wearer and the rpm of the vibrational motor so that it doesn’t hurt the user. Finally, a factor to consider is the type of adhesive used when placing the device on the user.

Regarding the physical issues with the device in regard to the user, we have to consider many factors. We must consider the voltage of the battery when designing our implementation so that it may not cause harm to the user. With a power supply being physically placed onto the user, we must ensure that the amount of power that is being used will not cause any harm. Although we are only planning on using a 3.7V battery, which is too low for shock (~60V), skin and membrane damage(~600V) or permanent damage(~20,000V), we must consider the current that our design is using. Anything more than 16 mA, will cause discomfort to an individual, so to air on the side of caution, we should not even reach 10mA, to ensure the utmost safety to the user [12]. Even with these precautions, we can ensure the utmost safety to the user by encasing the battery pack as well as the rest of our circuits in non-conducting material, which will be done by 3D printing using material such as carbon fiber.

Another factor that we need to consider is the type of adhesive that will be used on the user’s skin. We must make sure that it’s hypoallergenic, so that there's no adverse reaction to the user and we must make sure that we pick an adhesive that can be reused a multitude of times.

A key ethical issue we must concern ourselves with is that our project is attempting to prevent a medical condition. Therefore, if we are not accurate with our readings and data, we may mislead the user with incorrect information and may actually cause more harm than good to the user. We must account for every possible scenario to ensure that we do not cause a detriment to the health of our user. With this invention being placed on our user, we must also make sure that it is not too heavy (> 150 grams) or else it will cause strain to the user's neck and upper back.

**4.3 Ethics**

There are few ethical codes that we must worry about when we design our project. Because our implementation is physically being put on a person, we have to make sure that we follow code 1.2 of the ACM Code of Ethics: Avoid Harm [12]. We must make sure the adhesive and the components of our invention that are being placed on a person will not harm them in any way and cannot be used in such a way that it negatively impacts the user.

With our project including an app interface, we have to also consider code 1.6 of the ACM Code of Ethics: Respecting Privacy [13]. Our project constantly monitors an individual’s posture as well as many calibration factors, such as weight, height, age, etc., so we must keep the user’s information confidential as those factors are private to oneself.

One more ethical issue that we should highlight and make sure to follow is the IEEE code of ethics 7.8 subsection 10: to support colleagues and co-workers in following this code of ethics, to strive to ensure the code is upheld, and to not retaliate against individuals reporting a violation. [14]. This is a code that we have to uphold because we are making something from scratch, which entails 100’s of hours of potential work. We must make sure that although we must put that amount of effort into our project, that each individual in our group is making sure that they are both physically and emotionally well and ensure that there is a balance between the work needed for this project as well as a healthy off-work life.

**References**

[1] nytimes.com, “The Pandemic of Work From Home Injuries”, 2020. [Online]. Available: <https://www.nytimes.com/2020/09/04/well/live/ergonomics-work-from-home-injuries.html?fbclid=IwAR3y4TwqmDq2ZLAoYVMlpikjyPZH-7xRMcqfMyD-eQoSS3FaI5UZ14nc5zU>  [Accessed 1-March-2021]

[2] spine-health.com, “How to Measure and Fix Head Posture”, 2018. [Online]. Available: <https://www.spine-health.com/conditions/neck-pain/how-measure-and-fix-forward-head-posture.com> [Accessed 18-Feb-2021]

[3] spine-heath.com, “Forward Head Posture’s Effect on the Cervical Spine”, 2018. [Online]. Available: <https://www.spine-health.com/conditions/neck-pain/forward-head-postures-effect-cervical-spine.com> [Accessed 18-Feb-2021]

[4] Cedars Sinai, “Curves of the Spine”, 2021. [Online]. Available: <https://www.cedars-sinai.org/health-library/diseases-and-conditions/c/curves-of-the-spine.html> [Accessed 4-March-2021]

[5] Spine Health, “Thoracic Kyphosis: Forward Curvature of the Upper Back”, 2010. [Online]. Available: <https://www.spine-health.com/conditions/spinal-deformities/thoracic-kyphosis-forward-curvature-upper-back> [Accessed 4-March-2021]

[6] Sparkfun Electronics, “MPU-6000 and MPU-6050 Product Specification Revision 3.4,” [PDF]. Available: <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf> [Accessed 1-March-2021]

[7] Sparkfun Electronics, “Polymer Li-ion Rechargeable Battery,”[PDF]. Available: <https://cdn.sparkfun.com/datasheets/Prototyping/spe-00-502535-400mah-en-1.0ver.pdf> [Accessed 1-March-2021]

[8] Sparkfun Electronics, “Stand-Alone Linear Li-Ion / Li-Polymer Charge Management Controller” [PDF]. Available: <https://cdn.sparkfun.com/assets/b/a/7/6/8/MCP73833Datasheet.pdf> [Accessed 1-March-2021]

[9] Sparkfun Electronics, “LOW DROP FIXED AND ADJUSTABLE POSITIVE VOLTAGE REGULATORS,” [PDF]. Available: <https://www.sparkfun.com/datasheets/Components/LD1117V33.pdf> [Accessed 1-March-2021]

[10] Sparkfun Electronics, “Class 1 Bluetooth® Module,” [PDF]. Available: <http://cdn.sparkfun.com/datasheets/Wireless/Bluetooth/Bluetooth-RN-41-DS.pdf> [Accessed 1-March-2021]

[11] ncbi.nlm.nih.gov, “Expressing Cobb Angle as a Linear Measurement in Scoliosis”, 2017. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5769934/#:~:text=sin%20(X%2F2)%20> [Accessed 1-March-2021]

[12] ncbi.nlm.nih.gov, "Conduction of Electrical Current to and Through the Human Body", 2009. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2763825/> . [Accessed: 4- March- 2021].

[13] acm.org, "ACM Code of Ethics", 2018. [Online]. Available: <https://www.acm.org/code-of-ethics>. [Accessed: 18- Feb- 2021].

[14] Ieee.org"IEEE Code of Ethics", 2016. [Online]. Available: <http://www.ieee.org/about/corporate/governance/p7-8.html> . [Accessed: 18- Feb- 2021].