Final Report

Problems

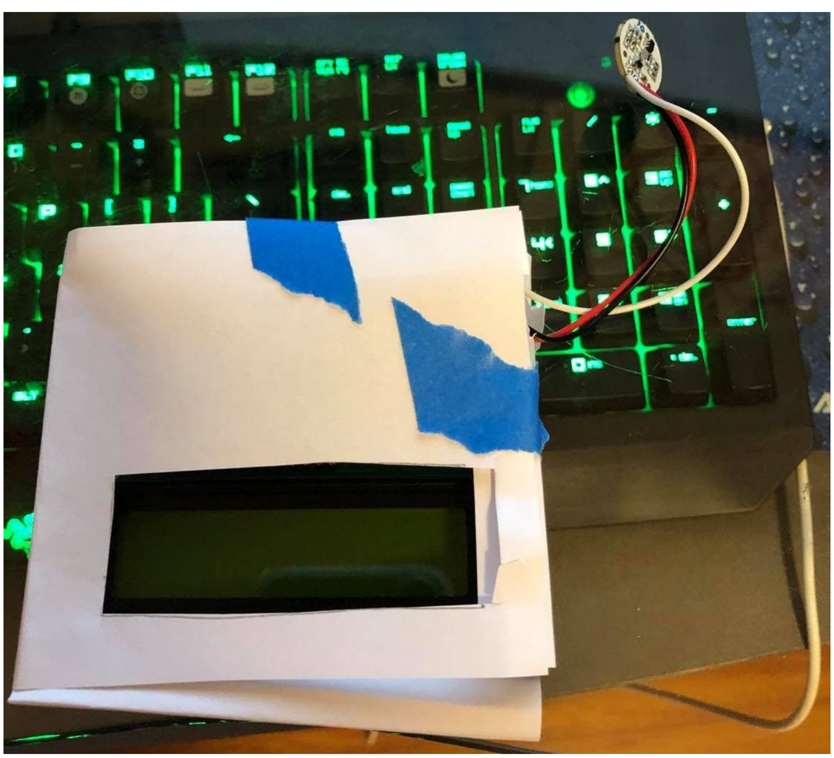
The problem we are trying to solve with our project is assisting people who exercise, especially while running. Our project is a fit-kit which tracks a person's heart rate in beat per minute and the steps taken. Our product is especially designed to track and give useful information for the user while they are exercising, this can help with their progress and body condition. In a society where obesity and diabetes are an issue we hope to encourage exercising by making tracking one's progress simpler.

Final Project Status

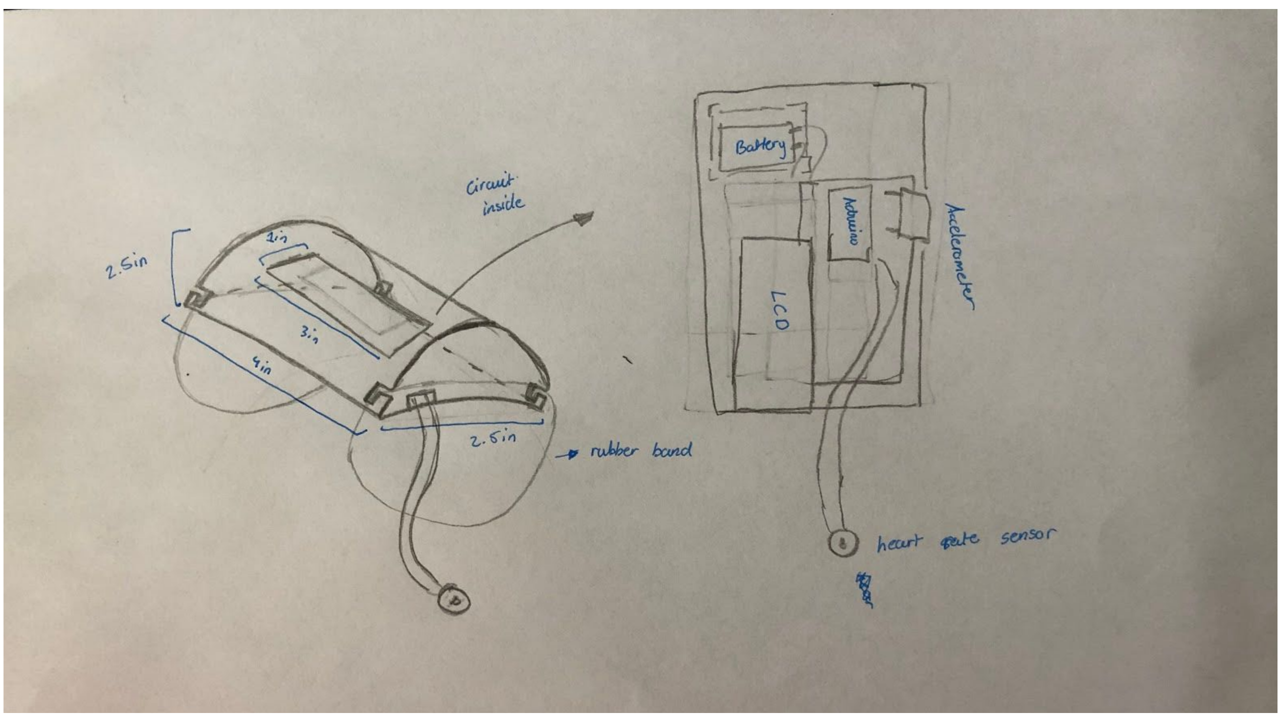
Our final scope of our project can keep track of the user heart rate which updates every 10 seconds and keep track of steps taken live. The accuracy of these measurements is around 5-10% errors. There were not any significant changes to our design since the technical design report. Minor details like adjusting threshold value in the code to yield better accuracy on our measurement as well as integrating casing with electronics were the only changes we made. We finished our casing model which does completely resemble our CAD model because we could not replicate the curve we wanted in our design with the materials we had so we improvised with a trapezoidal shape which mimics our original design.

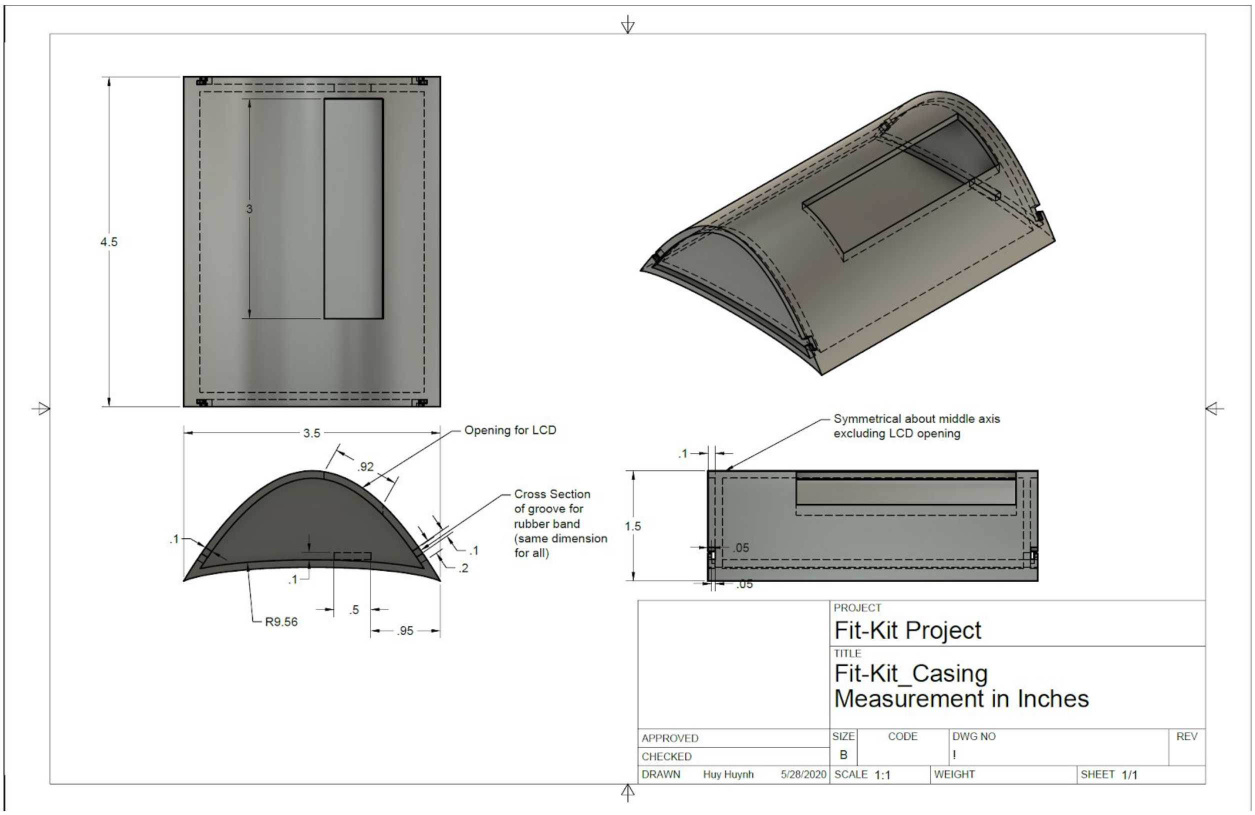
Build Process and Prototyping

At first we thought of a simple box like design for the casing which will store all the electronics parts. A wire extension with a heart rate sensor that extends out from the casing and is attached to the user index. The casing is meant to be rested on the user arm but we didn't work out how that was going to work.

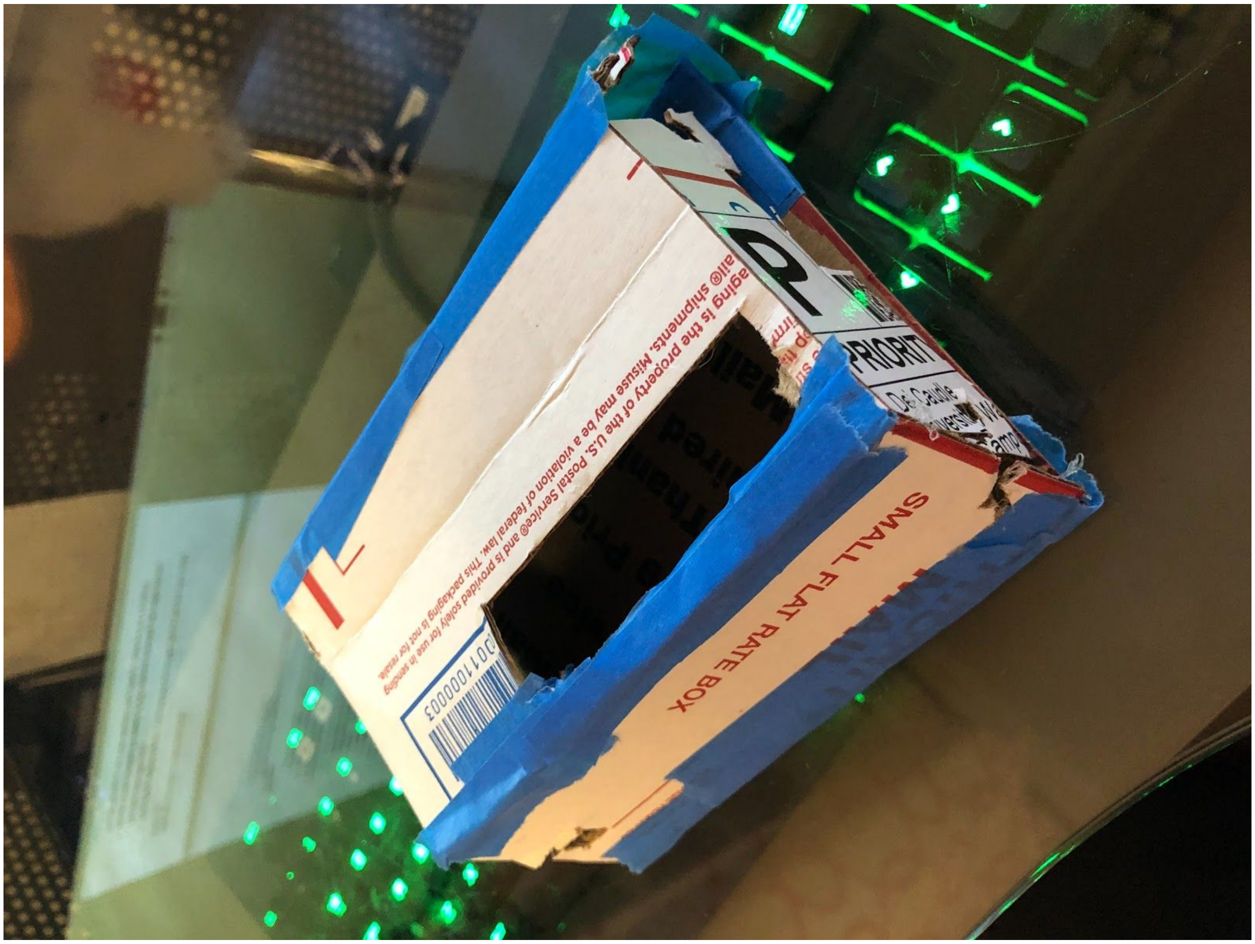


When we went to designing a final concept, we altered the simple box shape to a curved half cylinder shape sharing resemblance to an arm guard. The idea is to make our fit-kit comfortable to wear for the user and more secure to the user. The curved shape of the casing will go along the shape of the user's arm and sit on top nicely. As for attaching the casing itself to the arm, we implemented groves on the front and back wall of the casing which allow rubber bands to be wrapped around and serve as a way to hold down the casing to the user's arm. There might be some discomfort for the user with this approach but consider the materials we have this might be the best solution. Similar in our initial design we will have a cut out rectangle on top to show the LCD and extension wire for the heart rate sensor to be attached to the user's index finger. The overall layout electronics inside is simple and tries to minimize the area it takes. Every electronics part will be held down by frame-like structures. Note all measurements in the diagram are in inches.





Our digital design is not quite the same as our physical one because as mentioned before the curve we wanted couldn't be replicated well with the material we had so instead we chose a trapezoidal shape. The trapezoidal shape accomplishes a similar specification we want which reduces the bulky ness of the fit-kit on the user arms although might be not as well but that is something we have to put up with. Our original design was going to be in 1 or 2 pieces but our physical models turn out to be 5 pieces which we attach together using tape and hot glue. We use fast prototyping to test the dimension and size of our casing. Our initial CAD model's dimensions were rough estimates since we did not know the full size of our circuits going to be. Fast prototyping using paper helped us quickly test if everything was going to fit and make any adjustment needed before creating the cardboard model. Turned out our initial width of our casing was a little bit too short to fit everything in so we had to make it a half inch longer in the end. Also, we did not have frame like structure in our casing from our original design since everything fit snugly in already and movement was not an issue.



Experimental Results

Trails	bpm (fit-kit)	bpm (app)	Trial	Heart Rate Test at Rest Errors	Heart Rate Test While Moving Errors
1	78	74	1	5.41%	2.94%
2	76	75	2	1.33%	2.91%
3	77	73	3	5.48%	3.81%
3	77	76	3	1.32%	2.97%
3	79	75	3	5.33%	4.81%

For our heart rate testing we used an app on the phone to compare the measured value from our fit-kit. I am not sure about the accuracy of the app but since we did not have any better measurement we chose to use it. As you can see we were able to get an overall error percentage of less than 5% for when the user is at rest and while on the move meeting our engineer specification for the heart rate accuracy.

Trails	Steps(count)	Steps(measured)	Steps errors
1	10	9	11.11%
2	5	6	16.67%
3	15	17	11.76%
4	20	23	13.04%
5	25	29	13.79%

As for your steps counter, we count the steps taken by us and compare that with the measured value of the fit-kit. We were able to overall stay below 15% error which does not meet our engineering specifications for the accuracy of the steps counter. However, steps taken for the measurements were low so testing with more steps can give us an overall accuracy of the fit-kit. Also, a step is subjected since a step could be small stride or big one, we tried to be consistent in the steps we take in the experiment so it would not mess up the data.

Challenges

Some challenges we faced when creating our projects are getting accurate data we want from the raw data from the sensors and size of the overall fit-kit. For the size of the fit-kit, we wanted it to be compact so it does not hinder or discomfort the user wearing it and this factor was mainly determined by the size of the circuit. Getting the circuits to be compact was just tackling the problem of wire management and efficiently placing and connecting the sensor, LCD screen to the arduino. After tinkering around to see the most optimized result then we could build a casing based on the circuit layout. As for getting accurate data like with the heart rate and steps, we overcame this problem by doing research online and seeing how other people have done it and adapting it to our own project. To fine tune as best as we could we adjust values like threshold which determines what reading counts as a heartbeat and not. Another idea we found was useful in getting better results is periodically updating the threshold as reading could change with different conditions like with the heart rate sensor where the surrounding light does it affect its reading. Taking everything into account we were able to get decently accurate results with less than 5% errors overall for heart rate and overall less than 15% for steps count.

For the accuracy of the steps counter we wanted to be less than 10% but we were a little shy of that of what we wanted in the engineering specifications. However, the high error might not be that bad because as mentioned before a step is subjected so steps can range a lot so having a rough estimate of the overall steps might be just as good. Despite the subjectiveness of a step, there is a problem in how we get the steps. The accelerometer measures the acceleration due to gravity which fluctuates greatly as we move the fit-kit around so it is easier for spikes in the axis aligned with gravity to pass or threshold than that of the other axis. A potential way to factor out gravity is to determine which of the axes best aligns with gravity and scale down the reading from that axis by a certain percentage so it does not have a huge impact on the step reading.

Considering the effects of the class being online, some challenges were building the casing for our fit-kit. A 3D printer would be nice to make a viable casing with our curve design and protect the circuit from outside elements like water. The dynamic of our team was good considering everything is online but being able to work in person and having a single project we put efforts in would improve our ending product. The online setting meant we had to build the project separately so our project might not be as polished as it would have been done in person together. I think our communication was good and we got our work done so that was not much of an issue for us.

Next Steps

Some next step we can take is implementing a distance counter using our steps with the calculation of steps times stride length which we learned from researching what FitBit, a popular product similar to our does [1]. Another improvement we can do is with our casing which in its current state is just pieces of cardboard held together by tape and hot glue. We can try to get access to a 3D printer to print our CAD model which will be a huge improvement since it can be taken outside and have the curved shape to better fit on the user's arms. Also, we would like to use a velcro-like arm strap instead of rubber bands to attach the fit-kit to the person's arm as rubber bands can be uncomfortable and not as stable.

Appendix

Can view a demonstration video of our project in action:
https://drive.google.com/file/d/1Cc2pOlzlugWrcJppxCnTW1D6tEX_tK-g/view?usp=sharing

Our code for the arduino can be view here:

```
#include <LiquidCrystal.h>
#include <Wire.h>
#define USE_ARDUINO_INTERRUPTS true

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

const int PulseWire = A1;
int threshold;
float sensorValue = 0;
int count = 0;
unsigned long starttime = 0;
int bpm = 0;
boolean counted = false;
int prevMax = 0;
int prevMin = 1000;
int mini;
int maxi;
int prevBPM;
int steps;
const int MPU=0x68;
const float thresholdA = 6000;
float xavg, yavg, zavg;
boolean flag = true;

void setup() {
	Wire.begin();
	Wire.beginTransmission(MPU);
	Wire.write(0x6B);
	Wire.write(0);
	Wire.endTransmission(true);
	lcd.begin(16, 2);
	Serial.begin(9600);
	analogWrite(6, 100);
	calibrateA();
}
void calibrateA() {
```

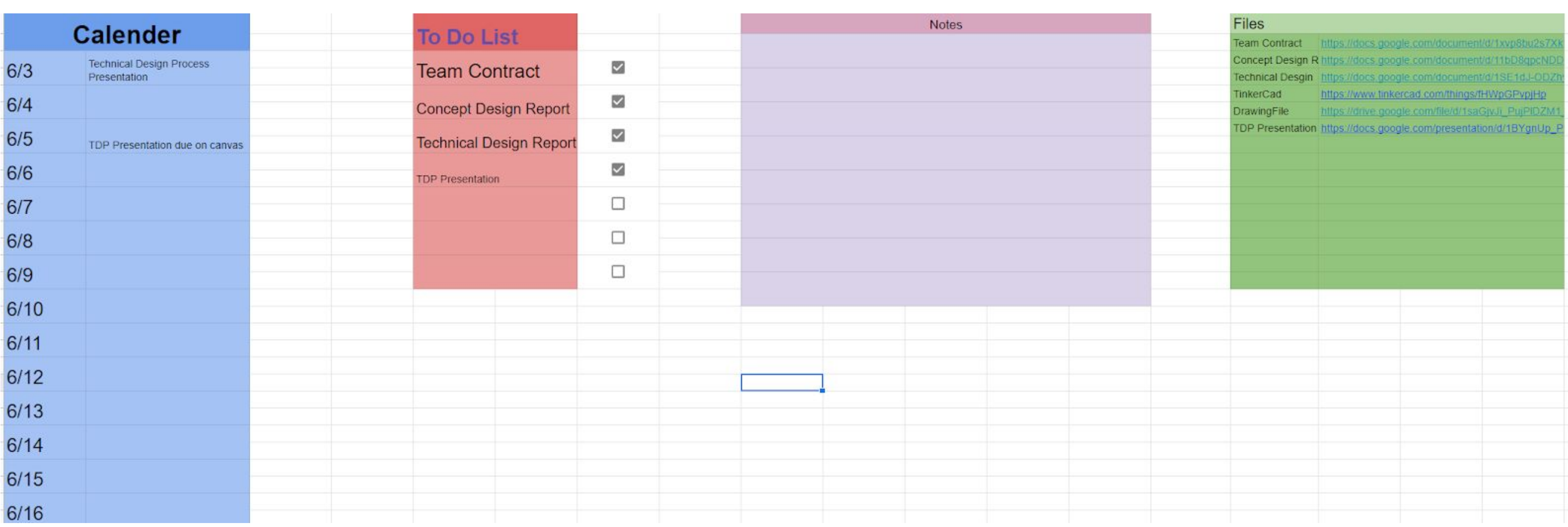
```
	float xval = 0;
	float yval = 0;

	float zval = 0;
	float sumX = 0;
	float sumY = 0;
	float sumZ = 0;
	for (int i = 0; i < 100; i++) {
		Wire.beginTransmission(MPU);
		Wire.write(0x3B);
		Wire.endTransmission(false);
		Wire.requestFrom(MPU, 14 ,true);
		xval = Wire.read()<<8|Wire.read();
		yval = Wire.read()<<8|Wire.read();
		zval = Wire.read()<<8|Wire.read();
		Wire.read()<<8|Wire.read();
		Wire.read()<<8|Wire.read();
		Wire.read()<<8|Wire.read();
		Wire.read()<<8|Wire.read();
		sumX = xval + sumX;
		sumY = yval + sumY;
		sumZ = zval + sumZ;
	}
	delay(100);
	xavg = sumX / 100;
	yavg = sumY / 100;
	zavg = sumZ / 100;
}

void calibrate () {
	lcd.clear();
	lcd.setCursor(0, 0);
	lcd.print("Calibrating");
	starttime = millis();
	while(millis() < starttime + 10000){
	sensorValue = analogRead(PulseWire);
	Serial.println(sensorValue);
		if (sensorValue > 20 && millis() > starttime + 5000) {
			maxi = max(sensorValue, prevMax);
```

```
                mini = min(sensorValue, prevMin);
                prevMax = maxi;
                prevMin = mini;
            }
        }
        threshold = (prevMax + prevMin) / 2;

        prevMax = 0;
        prevMin = 1000;
}

int getData(){
        float totVect = 0;
        float aX, aY, aZ;
        float sumVect = 0;
        starttime = millis();
        while (millis() < starttime + 10000) {
            sensorValue = analogRead(PulseWire);
            if (sensorValue < 20 || prevBPM > 200){
                calibrate();
                starttime = millis();
            }

            if (sensorValue > threshold && !counted) {
                count++;
                counted = true;
            } else if (sensorValue < threshold && counted){
                counted = false;
            }
            maxi = max(sensorValue, prevMax);
            mini = min(sensorValue, prevMin);
            prevMax = maxi;
            prevMin = mini;

            Wire.beginTransmission(MPU);
            Wire.write(0x3B);
            Wire.endTransmission(false);
            Wire.requestFrom(MPU, 14 ,true);
            aX = Wire.read()<<8|Wire.read();
```

```
		aY = Wire.read()<<8|Wire.read();
		aZ = Wire.read()<<8|Wire.read();
		Wire.read()<<8|Wire.read();
		Wire.read()<<8|Wire.read();
		Wire.read()<<8|Wire.read();
		Wire.read()<<8|Wire.read();
		totVect = sqrt(((sq(aX - xavg) + sq(aY - yavg) + sq(aZ - zavg))));

		if (sumVect == 0) {
			sumVect = totVect;
		} else {
			sumVect = (totVect + sumVect) / 2;
		}
		if (sumVect > thresholdA && flag) {
			steps++;
			flag = false;
		} else if (sumVect < thresholdA && !flag){
			flag = true;
		}
		if (steps < 0) {
			steps = 0;
		}
		lcd.setCursor(0, 1);
		lcd.print("Steps: ");
		lcd.print(steps);
	}
	calibrateA();
	threshold = (prevMax + prevMin) / 2;
	prevMax = 0;
	prevMin = 1000;
	return count * 6;
}

void loop() {
	lcd.setCursor(0, 0);
	lcd.print("BPM: ");
	if (prevBPM == 0) {
		lcd.print(bpm);
	} else {
```

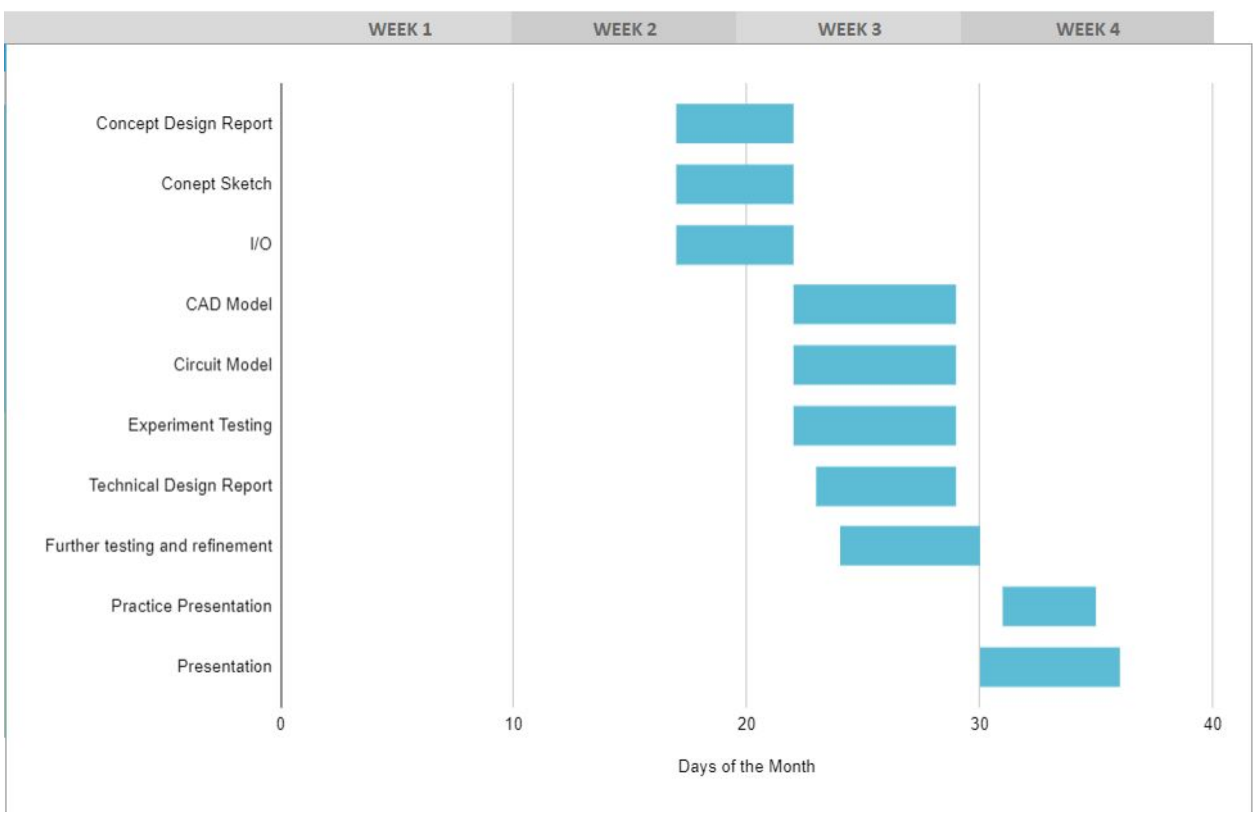
```
            lcd.print(prevBPM);
        }
        prevBPM = (bpm + prevBPM) / 2;
        lcd.setCursor(0, 1);
        lcd.print("Steps: ");
        lcd.print(steps);
        sensorValue = analogRead(PulseWire);


        bpm = getData();
        lcd.setCursor(0, 0);

        lcd.print("BPM: ");
        lcd.print(bpm);
        delay(100);
        lcd.clear();
        count = 0;
}
```

Task chart of our project progress over the quarter:



Google Sheet to keep track of tasks/ files

TASK NAME	START DATE	DAY OF MONTH*	END DATE	DURATION* (WORK DAYS)	DAYS COMPLETE*	DAYS REMAINING*	TEAM MEMBER	PERCENT COMPLETE
Fit-Kit								
Concept Design Report	5/17	17	5/22	5	5	0		100%
Conept Sketch	5/17	17	5/22	5	5	0	Huy	100%
I/O	5/17	17	5/22	5	5	0	Jose	100%
CAD Model	5/22	22	5/29	7	7	0		100%
Circuit Model	5/22	22	5/29	7	7	0		100%
Experiment Testing	5/22	22	5/29	7	7	0		100%
Technical Design Report	5/23	23	5/29	6	6	0		100%
Further testing and refinement	5/24	24	5/30	6	6	0		100%
Practice Presentation	5/31	31	6/4	4	4	0		100%
Presentation	5/30	30	6/5	6	6	0		100%



References:

[1] FitBit Inc., "Fitbit Help," Fitbit Help, 2020. [Online]. Available: https://help.fitbit.com/?l=en_US&c=Topics:Getting_Started. [Accessed: 23-May-2020].