***Abstract-*** **PerFectIT is a real time fitness monitor that will allow people to observe and correct their fitness progress. This system will provide feedback to the user, allowing them to correct their form in real time, helping decrease injuries caused by repetitive incorrect motion. People ranging from beginners to gym experts can benefit from this device because it provides a quantitative form of feedback that is unlike fitness methods that exist today. PerFectIT incorporates 3 groups of sensors, these sensors will then transmit the data real time to an Android device, which allows for full body analysis of any exercise.**

I. INTRODUCTION

Injuries in sports and exercise are the worst nightmare for any athlete. Injuries can set back athletes both mentally and physically, costing them valuable time when they could be active or improving.

Causes of injuries can occur in many different forms. In the gym, poor form, overtraining, and not warming up are three leading causes of injuries (“Poor technique - Causes of Sports Injuries.”). Even though some injuries may not occur right away, or even over a short period of time, incorrect training can prove to be disastrous to tendons and joints.

Current prevention methods consist of personal trainers, online tutorials, and group classes. Out of these three options, personal trainers are the safest. These people are certified and educated to provide feedback to athletes while they work out. However, personal trainers can be tricky to afford and schedule. As reported per a study conducted by the National Strength and Conditioning Association, the average personal trainer charges $50 per hour (Griffin, R. Morgan). This charge on top of gym membership fees, can be hard to budget for, and in many cases is not affordable. Even if someone could afford a personal trainer, feedback is limited to only qualitative movement corrections.

One way to save on the cost of personal trainers while maintaining the ability of correction is group classes. Group classes are still expensive, but more affordable. One downside that comes with group fitness classes is that the athlete is not being monitored for the complete workout. Instead the professional walks around a class, providing feedback to the class all at once. This results in less attention and feedback with a professional that could potentially correct bad form that will lead to injuries. These classes still prove to be costly, and only provide feedback at short segments throughout the duration of the workout.

Another path that an athlete to take is to watch online tutorials and provide self-correction. While this option proves to be the most cost efficient, it lacks any time of real time correction. Just because someone watches an online tutorial does not necessarily mean that they will be able to replicate this tutorial precisely. This may lead to practiced bad form, which in turn can cause injuries and progress plateaus.

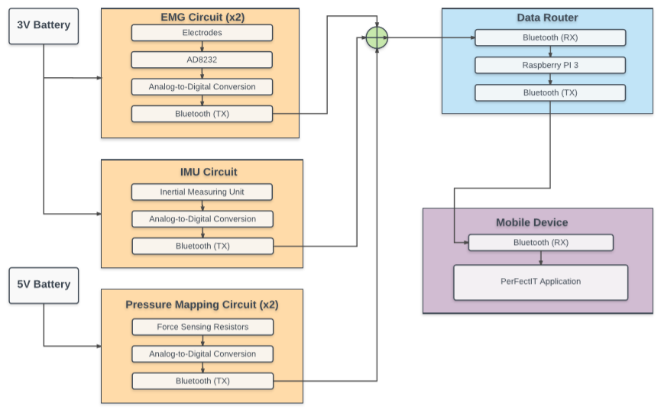
II. DESIGN

*A. Overview*

To address the problem of providing a cost-effective way to prevent injuries in the gym, we have designed PerFectIT. To implement this system, three separate sensing modules are utilized to monitor the users form.

The first module utilizes force sensing resistors (FSR) implanted into the insoles of the user’s shoes to detect and map the weight distribution across the user’s feet. FSRs were chosen because of their accuracy, durability, and comfort necessary for this application.

The second module takes advantage of a electrodiagnostic technique known as electromyography (EMG) to detect and





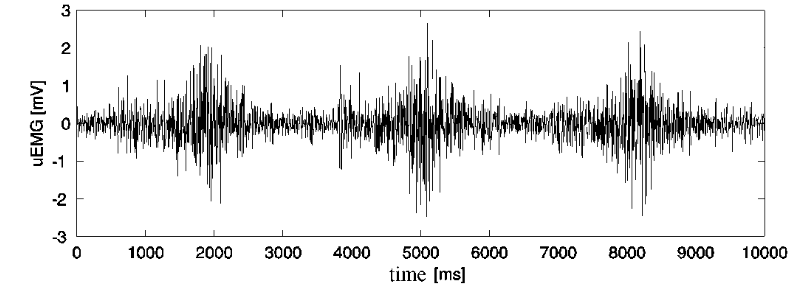
record the electrical signals produced by the skeletal muscles during exercise. The EMG circuit will be built into two wearable knee braces for monitoring muscle activity around each knee during exercise.

The third module consists of a 3-axis inertial measuring unit (IMU) to monitor the users center of gravity located around their hips. The module is intended to clip onto the users shorts, pants, or weight lifting belt and will keep track of the user’s form.

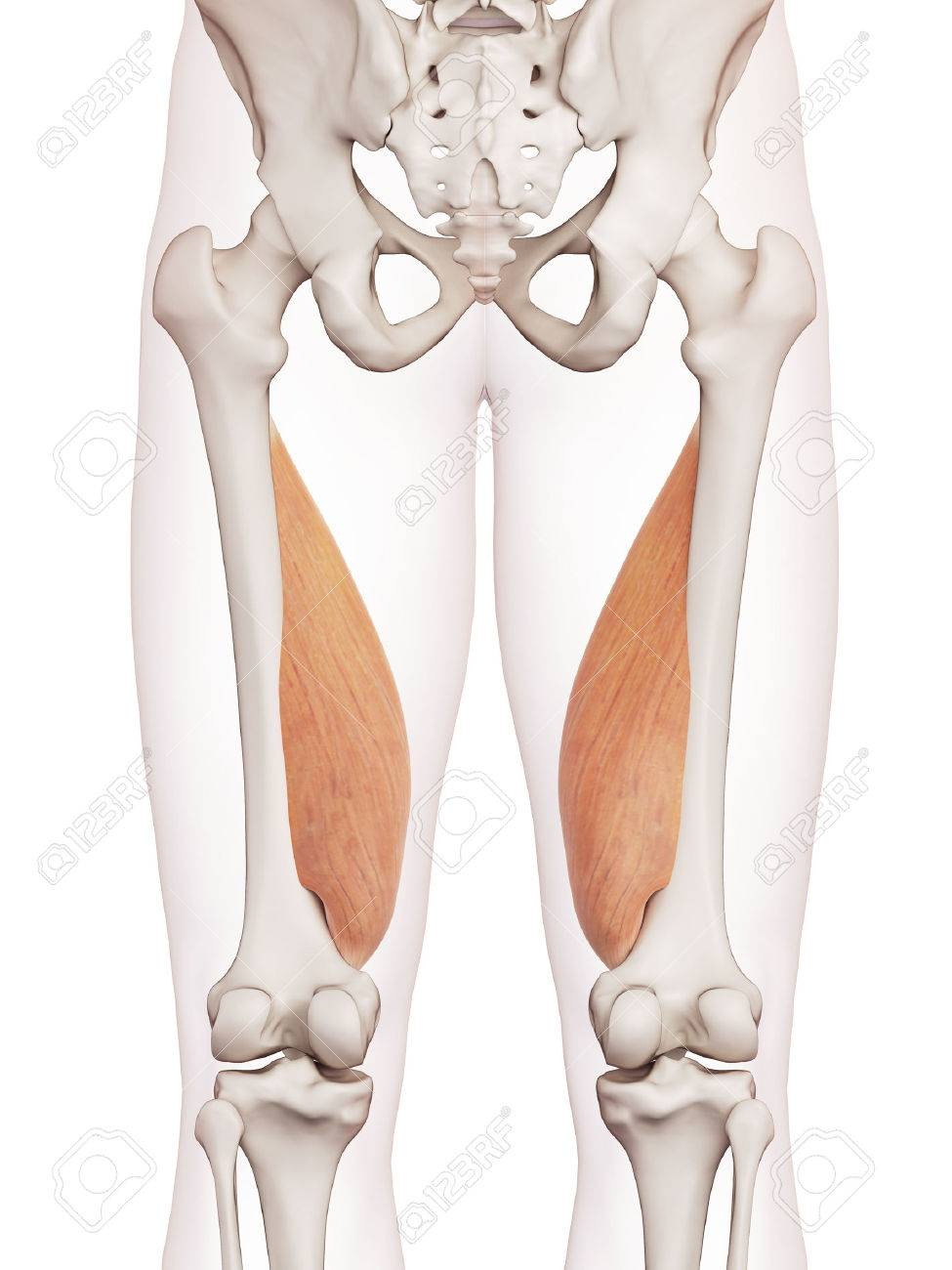
Each module will communicate it’s data via Bluetooth to a Raspberry Pi 3, where it will then be communicated to a mobile application for effective user feedback.

*B. EMG Circuit*

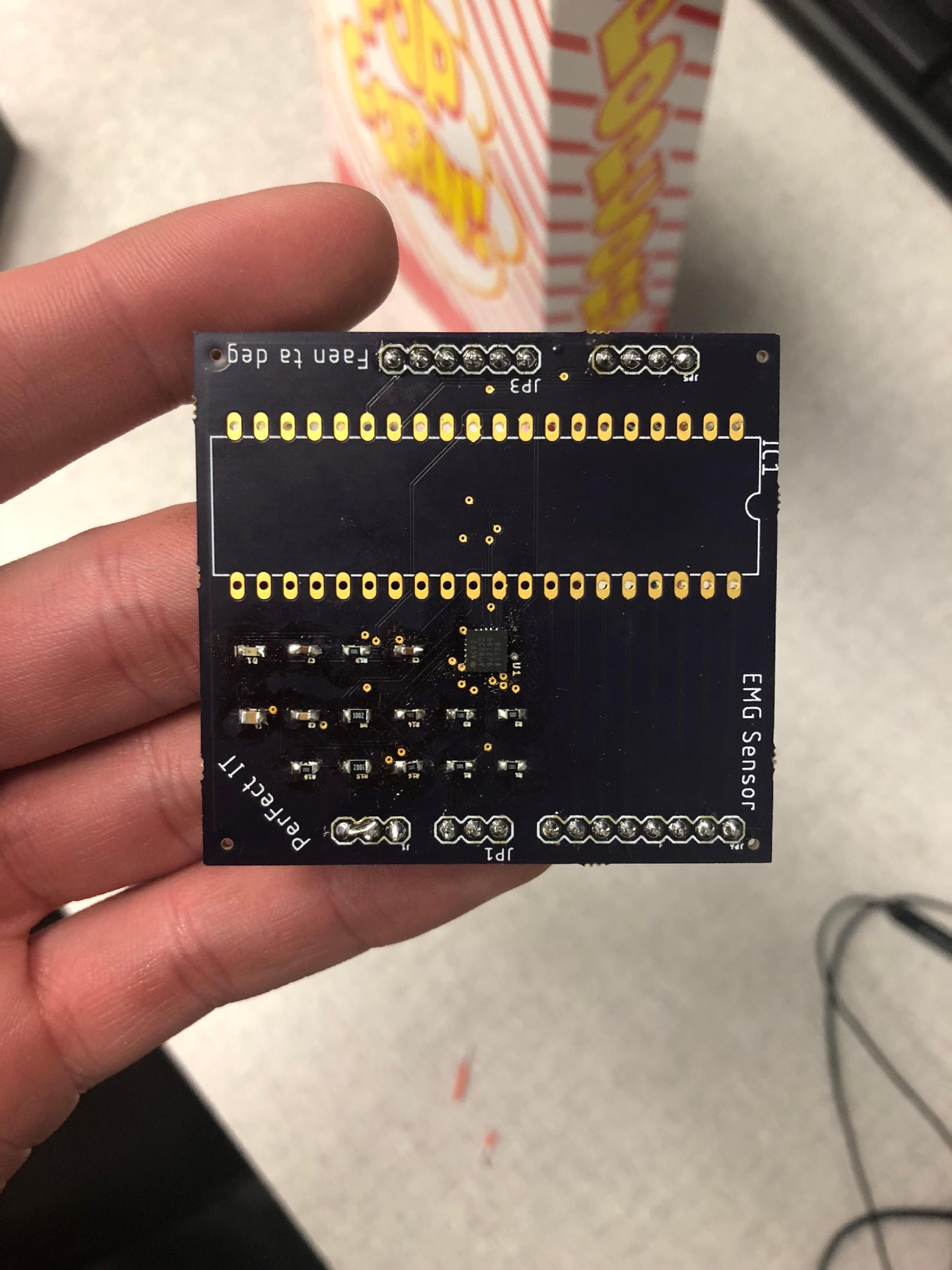
The purpose of this block is detect and measure the muscle activity at each knee during the repetition of a squat. During muscle exhaustion, the central nervous system sends a signal via motor neurons to innervate the muscle fibers. The motor neuron units cause a chemical reaction that generates an electrical potential along the fibers resulting in muscle contraction. The resulting signal is called the electromyogram (EMG) and can be measured by placing electrodes along the fibers of the muscle being activated. Figure X shows what a typical EMG signal looks like



The developed EMG system consists of two knee braces. Each knee brace will be a self-contained EMG monitor that sends the data it collects to a mobile application for analyses. Integrated into each knee brace are two cloth electrodes sewn into a position as to effectively acquire the EMG at the target muscle (the Vastus Medialis, see figure 3). The external connections and hardware is removable from the brace via snap connectors – allowing the brace to be washed if necessary. Two wires will snap into place, connecting to the cloth electrodes that interface with the skin above the muscle as to detect the desired EMG signal. The 

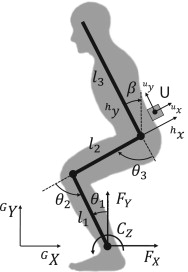


EMG signal can vary from 5 microvolts to 200 microvolts with a frequency band between roughly 5Hz and 800Hz. The small signals and their low frequency band require that the signal be amplified and filtered before sending out for analyses. Analog Devices AD8232 was chosen because of its high input impedance and large gain (~40dB). This device also has a programmable band pass filter (BPF) making it an effective device for an EMG application. The intended applications for the AD8232 are health monitoring systems such as heart rate monitors or EMG sensing circuits. After reviewing the data sheet provided by Analog Devices, it was seen that if the AD8232 is placed in close proximity to the source of the signal, no reference electrode is needed (allowing for just two electrodes) and that the higher frequencies can be ignored. Because the distance from the muscle is small, the signal is strong and there is less muscle artifact interference. As a result the device is able to filter out low frequencies via a high-pass filter with a cutoff frequency at 5 Hz, while providing the necessary amplification of the signal so as to be effectively analyzed by the mobile application. The EMG sensing circuit was designed using PSpice and a PCB layout was created using EAGLE. The resulting EMG signal processing PCB is shown below in Figure 4.



*C. Inertial Measuring Unit (IMU)*

The purpose of this subsystem is to monitor the position and movement around the user’s hips during a squat. During a squat, the user’s center of gravity is located around their hips. It is important that the user maintains the proper form when driving through their squat, and that the user does not arch their back or twist their hips while performing the squat. The IMU is a 3-axis accelerometer, gyrometer, and magnetometer that detects motion along 3 different axis – allowing the system to monitor any incorrect form at the user's hips. The IMU transmits its data real-time via Bluetooth with the mobile application, allowing for useful feedback to be given to its user.





*D. Pressure Mapping Circuit*

The foot pressure mapping measurement system is an implementation of sensors to create a foot map of relative weight. This foot map provides quantitative feedback of how much weight is put on the heel, arc, and ball of the foot, which in turn tells us if the user is performing a squat correctly or not.

To construct this foot map shoe insoles are used. The sensors are butterflied in half, and the sensors rest between the upper and lower halves of the insoles. The wire leads then leave the shoe insoles out the back of the heel of the shoe. Outside of the shoe the wires are then connected to a PCB which provides force sensitive resistor circuitry and a Bluetooth transmitter. This data is then sent to the Raspberry Pi to be conveyed to the android device. We 3D printed enclosures for the PCB’s to go in from the Studio Lab on the third floor of W.E.Bb. Dubois. We used SolidWorks to design these enclosures/

The shoe sensors ended up working well. One problem that was encountered with the sensors was that the PCB couldn’t be printed in house as we had hoped. This resulted in wiring a protoboard so that the device was portable. This worked well until there was a short in the circuit which caused the battery to overheat. We designed a PCB to combat this issue but it did not arrive in time for FPR / Demo Day.

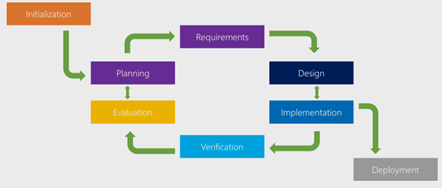
III. Software Specifications

It is essential for our hardware system to be integrated to a device which analyzes and displays the collected data. An ideal device would be mobile and powerful enough to do the required computation for our project. According to the popular technology blog *Verge*, 99.6% of the new smartphones run Android or iOS. They also share 96% of the mobile market as of 2017. Therefore, the two mobile platforms to develop our application are Android and iOS.

1. *Model*

The software team decided to develop the application on the Android platform. There are few reasons why we choose Android over iOS, the team has plenty of experience in Java compared to Swift. Since this project is a time sensitive project, we do not have the time to explore a new language and implement it effectively. In addition, developing for iOS requires an apple product or a MAC Laptop; the team mostly has windows and it is easier to develop using Android Studio then to use a VM and develop for iOS.

We will be using the Iterative Design for our software model approach where we focus on initial, simplified implementation, which eventually rolls over into the final system. The requirements we will be focusing on throughout this scope of this project as presented in our PDR are: Flexible User Interface, PerFectIT Algorithm, Metrics (Power, Reps, Acceleration, and Weight Distribution), User Progress Tracking, Connection to the Database.



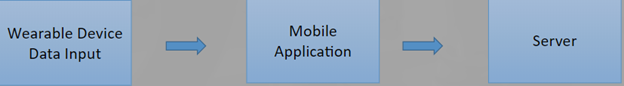
1) Flexible User Interface: The UI is a key component for any app. It must be appealing to the user in terms of visuals and it must be flexible in terms of performance. A user must be able to sign-in/sign-up with ease and quickness. He/She must be able to navigate to all the pages smoothly without any glitches.

2) PerFectIT Algorithm: This is the key component of the application. The data received from the hardware is “raw” and not suitable for the user directly. The purpose of the algorithm is to process the data and measure the deviancy from the actual correct data and present it to the user for feedback. The user can then use this to correct their form.

3) Metrics: This option is useful for users with who require more advanced metrics. These metrics could include the power, the number of reps performed, the total weight lifted, the weight distribution between the two legs and so forth.

4) Progress Tracking: The purpose of any activity is to measure the performance over a set period of time or measure the progress. This allows the user to change their activity to hit certain goals or to set goals to hit by a certain timeframe. Therefore, this is an important component for a fitness application

5) Database Access: There are two different types of storages in most mobile applications. These are either internal and external. Internal storage means all the data is stored in the phone itself. It is useful for application that do not require connection to the internet. External storage is for applications where it needs internet connection.



*C. Database Specification*

As previously mentioned there are two main storage options available for mobile applications. We have the internal and external storage options; internal storage is when the memory resides in the phone itself. The external storage is when the data is stored in a cloud based platform. This is useful when we want the data to be available instantaneously on other platforms. Android Studio has a lightweight internal storage option in SQLite. It is a good option as it is relational database and quite easy to store and retrieve data. Firebase and AWS are two external storages and they are stored in cloud. They offer a variety of services in addition to data storage; these include User Authentication, Data Analytics, and App Crashes.

*E. Implementation of the Application*

We used Android Studio to design and build our application. When testing the app we used Android Studio Emulator which let us choose which version of the android we would like to run our application.

A major component of the user interface is to allow the user to sign into the app using his or her credentials. This feature safeguards the user’s information from potential intruders who may seek to access personal data. This also allows the user to log into his or her portal from more than one device. In order to accomplish this, we cannot use internal storage because all the data is just inside the phone. The most efficient way to accomplish this is to send the credentials to a server which verifies if the credentials are correct and retrieves the user’s data.

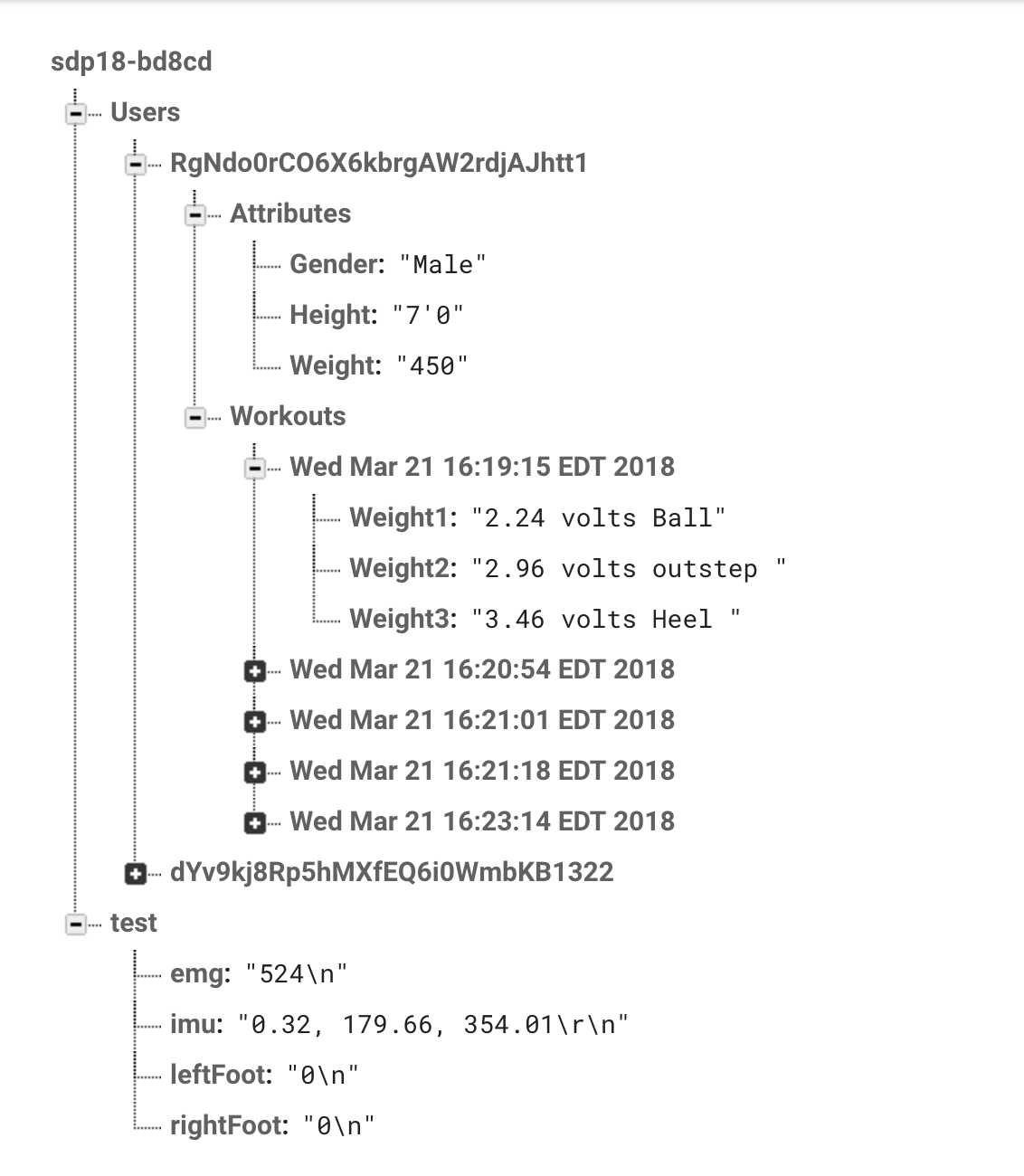
***Firebase*** has a functionality which allows the developers/programmers of the application to authenticate their users. The authentication can be through email, phone, or other social media platforms. We decided to implement this functionality, the user either signs up or he signs in using their email address and their personal password. We have created a profile for each user so they could track their progress and edit any information if they choose to.

Our project consists of three hardware devices as explained in earlier sections. In order to distinguish these components as well as make it visually appealing we created three “views”. Each view shows live time feedback from each of the three hardware system.

We developed graphics to depict the data so that the user isn’t simply seeing numbers on the screen. The three visuals include: a scale, a barometer, and a simple display for the IMU which sends “toasts” or warnings if the user is deviating from the norm. A visual of this on the website which shows how they look.

**Hardware/Software Connection** includes the communication between the Sensors and the App. Specifically the path between the two includes bluetooth communication from sensors to Raspberry pi, wifi communication between Raspberry pi and Firebase database and finally Android queries from the app to the Firebase.

The Bluetooth connection consists of a pico net connection between the sensors and the Raspberry Pi with one bluetooth transmitter dedicated to each foot, each knee and the low back Inertial measuring unit making five transmitters in total.

These signals all transmitted to single Bluetooth receiver connected to the Raspberry Pi which acted as a relay by transmitting these signals to the Firebase through post messages managed via python scripts.

Each sensor wrote to two locations. The first was a real time value location in the Json tree that was used by the real time feedback displays on the app. The second was a set of continuous locations that took a lower sample rate to keep a playback of the sensors to be interpreted and stored on a calendar function to review performance and track progress

These data sets are then pulled down by advance Android Query methods that both check for change in the JSon tree locations and the creation of new subunits of an indicated branch, covering both of our purposes respectively

IV. Project Management

**Hardware Team**

Andrew Sjogren

* Decision between FSR and Piezo Sensors
* Implementation of chosen sensor into constructed foot insoles
* Testing and calibration of foot insoles
* PCB schematic design for foot insoles
* Data collection

Maxwell Gerhardson

* Design and implement AD8232 for EMG sensing into knee brace
* PCB design for EMG sensing circuit
* Implement 9DOF razor IMU

**Software Team**

Sai Yarram

* Server and Backend development and Implementation
* User Authentication, Data Storage/Retrieval Implementation
* UI and Algorithm Development

Nick Raymond

* Hardware and Software Integration
  + Implementation of Bluetooth
* Weight calibration analysis with FSR
* UI and Algorithm Development

CDR Deliverables

Sai Yarram

-Have a fully functioning backend of the application

-Integrate with the cloud and have a database setup

-Authenticate Users, Stability, Analytics

-Work with Nick Raymond on UI and Data Processing/Algorithms

Nick Raymond

-Work on UI, Bluetooth Connection, and Data Processing

-Connect the hardware to the app

-Develop an algorithm with Sai Yaram

Andrew Sjogren

-Complete PCB for foot mapping sensors

-Communicate foot sensor data via bluetooth to Raspberry Pi

-Work with Max to communicate data to Android

-Testing / debugging of integrated subsystem

Max Gerhardson

-IMU

-Complete PCB for emg sensors

-Communicate EMG data via bluetooth to Raspberry Pi

-Work with Andrew to communicate data to Android

-Testing / debugging of integrated subsystem

IV. CONCLUSION

There have been several setbacks throughout the development of this project but we have successfully developed a working mobile application which integrates smoothly with the cloud and the hardware. Unfortunately the system was working for a few days to a week before there were small breaks in the circuit which took a little while to fix. The project was an amazing experience for all the team members. We learned the importance of teamwork and trust, the time and effort needed to correct and debug errors, the willpower to learn new technologies on our own. If we were to have more time on this project, perhaps a week, a month or a year we would go about to improve the efficiency of both our hardware and software modules. In particularly on the hardware side we would begin experimenting on different ways to make the circuits smaller and more precise. On the software side we would clean up the app by adding our graphics to a library we can just call and import rather then built into the app. We would build a more efficient database to store, edit and retrieve data. Finally, we would work on having other people test the system and take feedback so we could make proper adjustments and fine tuning.

ACKNOWLEDGEMENT

Our team would like to thank Professor Yeonsik Noh for taking his time to guide us along the this project. We would not have been able to make the progress we did without his invaluable suggestions. He urged us to meet every week to discuss the progress and the setbacks we encountered individually and as a team. He pushed us to make progress every week and help us set goals so that we could work towards completing the project in a timely fashion.

REFERENCES

Aboukhalil, Robert. “A Tale of Two Clouds: Amazon vs. Google – Robert Aboukhalil – Medium.” *Medium*, Medium, 13 Mar. 2017.

Griffin, R. Morgan. “Finding a Personal Fitness Trainer.” *WebMD*, WebMD, [www.webmd.com/fitness-exercise/features/finding-personal-fitness-trainer](http://www.webmd.com/fitness-exercise/features/finding-personal-fitness-trainer).

“Poor technique - Causes of Sports Injuries.” *Poor technique - Causes of Sports Injuries | Sports Medicine Information*, www.nsmi.org.uk/articles/causes-sports-injuries/poor-technique.html.

Sinicki, Adam. “Developing for Android vs Developing for IOS - in 5 Rounds.” *Android Authority*, 9 June 2016.