March 26th, 2021

Dr. Shervin Jannesar
Dr. Andrew Rawicz
Dr. Craig Scratchley
School of Engineering Science
Simon Fraser University
Burnaby, BC, V5A 1S6

RE: ENSC 405w Design Specification Document

Dear Dr.Jannesar, Dr.Rawicz and Dr.Scratchley,

The document attached to this letter defines the design specifications for Levo Technologies performance tracking barbell attachment, the Levo. The Levo product will allow weight lifters to record data reflecting their lifting performance, which can then be analyzed and used towards improving the athlete's capabilities.

This document puts an emphasis on the proof-of-concept (alpha) prototype. These choices have been split into two primary sections being sensor unit, and mobile app. Design choices and their respective justifications will outline the decision making process and how the design choice fulfills relevant alpha phase requirements for the Levo. Finally, foreseeable design considerations are included for the beta phase prototype and production prototype.

The alpha phase prototype of the Levo aims to demonstrate the application of engineering, physics, and kinesiology principles learned at SFU, and through our own research. The Levo Technologies team looks forward to the presentation of the Levo proof-of-concept prototype, and the continued development of the Levo.

Should you have any questions, or require further clarification on the attached documentation, the Levo, or Levo Technologies, please contact CCO Graham Fader at gfader@sfu.ca at your convenience.

Sincerely,

Natalia Page

Natalia Page
Chief Executive Officer
Levo Technologies

Company 1

Matthew Chute 301281840
Graham Fader 301276131
Antonio Kim 301333584
Chris May 301319469
Natalia Page 301307232

March 26 2021

Abstract

The Levo is a performance tracking barbell attachment that will enhance and aid an athlete's ability to understand their performance in the weight room. By tracking the movement of the barbell, several useful performance measurements can be extracted, analyzed and communicated to the athlete in a meaningful way. This document will state all current design choices and requirements they fulfil, and their justifications in regard to their proof of concept. These design choices are made for the sensor unit and microcontroller or mobile app, which are the modules required for completion of the Levo proof-of-concept prototype. Structural and safety requirements, as well as optimizations for the sensor unit, microcontroller and mobile app are analysed as future considerations for the beta phase prototype and mass production.

Contents

List of Figures

List of Tables

Levo Version History

Version Number	Implemented By	Revision Date	Approved By	Approval Date	Reason
A 1.0	Matthew Chute Graham Fader	03/25/21	Natalia Page	03/26/21	Solidification of proof of concept design specifications already in progress

1 Introduction

The Levo aims to accurately collect, analyze and communicate the data associated with an athlete's performance in the weight room. Strength coaches, teams, and athletes use the data collected to better assess areas for improvement, and how to tailor programs to the athlete. Furthermore, metrics that can analyze each rep can help give clues to the technique of the athlete during exercise and can help alter and improve their technique, while maximizing the effectiveness of the rep towards its targeted muscles. This document provides a background for the Levo's overall functionality for the end product, and provides a general block diagram on what the Levo will do during the alpha phase proof of concept. The rest of the document provides current design choices and justifications for the sensor unit, microcontroller, and mobile app design to meet the alpha phase requirements stated in Levo Technologies "Requirement Specification"[1]. As well as future design considerations and optimizations for the beta phase prototype, and production prototype.

1.1 Background

Implementing a lightweight sensor which can be attached to a barbell to track its movement during an exercise performed by an athlete will allow for the calculation of a variety of metrics which have been indicated as important by polled athletes and coaches. An illustration of the implementation of the Levo can be seen in the below figure:

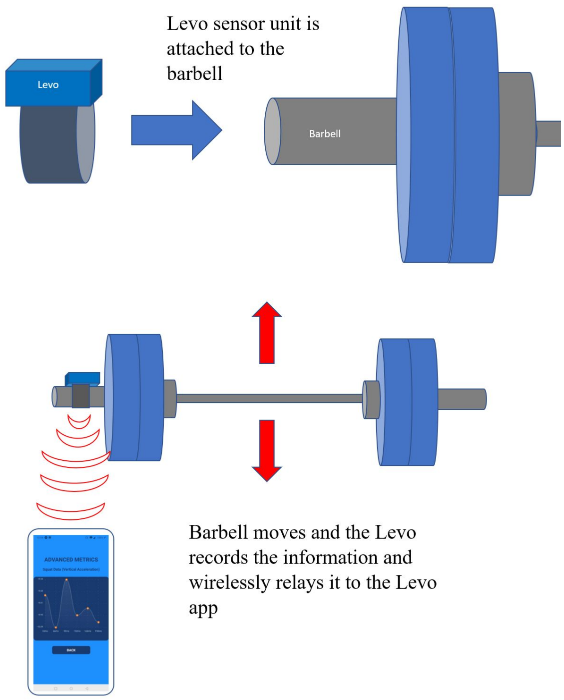

Figure 1: Production Phase Illustration of Levo Usage

1.2 System Overview

The sensor unit will consist of gyroscopes and accelerometers, whose data can be processed to be used in the calculation of targeted metrics. The following block diagram illustrates the data path of the Levo from sensor reading to display.

Figure 2: System Overview Block Diagram

1.3 Design Labelling

This document uses the following labeling system to organize design specification.

Des U.V.W

U: The category of the design, Sensor and microprocessor as 1, mobile app as 2.
V: The design number for that section.
W: Denotes whether the design is needed for the alpha phase proof of concept, beta phase prototype. 'A' will be used for alpha phase design, 'B' for beta phase design.

2 Sensor Unit and Microcontroller Design

The following section will be broken into two sections, the first pertaining to the alpha product and the later for the beta product. The table in section 2.5 describes the design specifications for both alpha and beta stages with explanations describing these decisions and the requirements they meet.

2.1 Raspberry Pi 4 Microcontroller

The alpha stage of the Levo design requires a platform which allows for simple connections to a sensor module via GPIO, and wireless connectivity capabilities allowing communication with the mobile app. The Raspberry Pi Microcontroller(**RPi**) satisfies these requirements and has an active community of users, which offers resources and documentation in regard to connecting and using third party sensors. The alpha prototype is not required to be powered by battery, making a battery powered development board unnecessary. An alternative to using an RPi controller would be an I2C to USB converter connecting a laptop to the sensor. However, the Levo Technologies team had RPi available from previous unrelated project purchases. Arduino is another viable alternative to the RPi, however, Arduino offered no major benefit over an RPi. Levo Technologies team members are familiar with the RPi and how its usage. The above factors indicated that RPi is the best choice for completing the proof of concept prototype. The requirement specifications that this design decision satisfies are as follows: 2.4.A, 2.9.A, 2.11.A

2.2 GY-521 MPU-6050

The sensor chosen for the alpha stage is a development board housing an MPU-6050 sensor chip. The alpha phase requirements for the sensor are: ability to communicate with the Raspberry Pi via serial connection with the RPi GPIO pins, and capable of measuring acceleration and angular velocity. More complicated sensors such as a Eddy current sensor, or Linear-variable differential transformers (LVDT) exist. However, the Eddy current sensor is susceptible to interference from metallic objects which are ubiquitous in workout equipment, and requires careful placement. LVDT sensors have components which are required to be stationary, which does not meet the requirement that the Levo may be attached anywhere on the bar. Another method for obtaining data for the movement the bar is through video tracking points on the equipment and using a software to process the frames as those points moved. This would require more processing power and would not be implementable through a mobile app. Furthermore, this method would require more than one video feed reliably collect data in all directions of movement during barbell exercise, which requires an excessive amount of user set up compared to the accelerometer implementation. The above factors indicated that RPi is the best choice for completing the proof of concept prototype. The requirement specifications that this design decision satisfies are as follows: 2.1.A, 2.2.A, 2.10.A

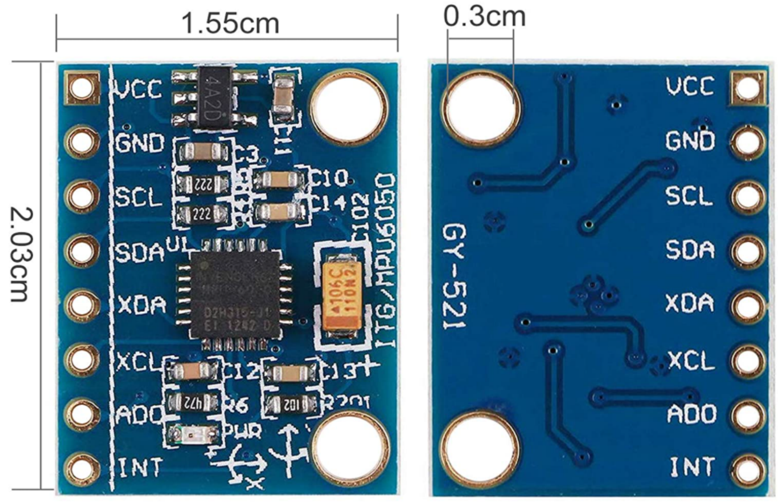

Figure 3: Picture of Development Board Sensor [2]

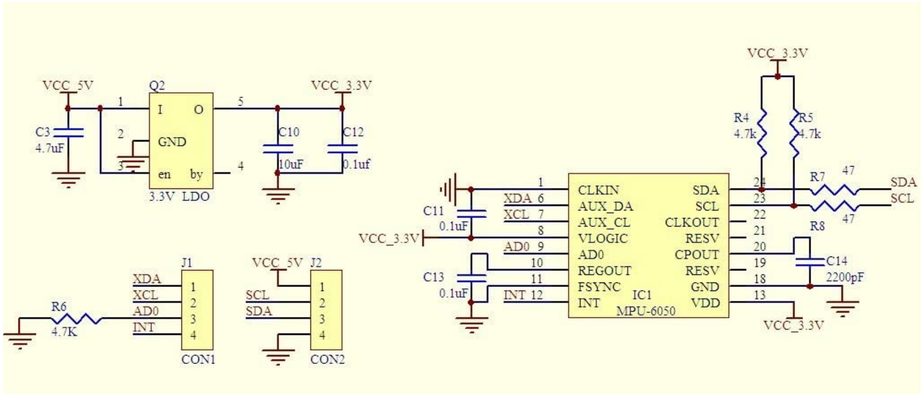

Figure 4: Electrical Wiring Schematic of GY-521 MPU-6050 [2]

2.3 Data processing sensor data

Embedded software running on the RPi is written in python due to its wide array of libraries, and developer familiarity with the language. C, C++, Javascript and MATLAB are appropriate alternative options. Python was chosen because it is free to use, can be written at a higher level than C/C++, and is easy to deploy on the RPi platform. The requirement specifications that this design decision satisfies are as follows: 2.1.A, 2.2.A, 2.3.B, 2.8.B, 2.9.A, 2.10.A, 2.12.A, 2.14.B

To calculate velocity and displacement, the RPi polls the sensor at a frequency of 40Hz to obtain discrete acceleration values. Numerical integration is done on the acceleration using trapezoid rule, which results in velocity. The obtained velocity is integrated using the same method to obtain displacement.The requirement specifications that this design decision satisfies are as follows: 2.1.A, 2.2.A

$$\int_a^b f(x)dx \approx \frac{\Delta x}{2}[f(x_0) + 2f(x_1) + \ ... \ + 2f(x_{n-1}) + f(x_n)] \qquad (1)$$

Trapezoid rule is simple to implement and only requires the current point and the previous data point.

Trapezoid rule does not require all data points of a set, or a function for use. By running the trapezoid rule on the current point and the last point and summing it with all previous values, the result of the integral can be updated in real time. Trapezoid also has a lower error than Riemann sum computations which are taken in a similar method.

Simpsons rule is also usable for numerical integration, and provides lower error than trapezoid rule. Simpsons rule, however, requires three data points, and implementing it to provide real time updates to the integral would require a higher sampling frequency than trapezoid rule in addition to a more complex code.

2.4 Future Sensor Unit and Microcontroller Considerations

2.4.1 FSP200 - IMU Processor

A component chosen for the Beta stage of the Levo is an IMU processor which will process and translate serial data between the sensors and the microprocessor. When selecting a IMU serial output is a requirement.The FSP200 can communicate via UART and receive I2C data. Another benefit to using the FSP200 is that the documentation describes how to connect the sensors and also gives some compatible for sensors for the FSP200.The requirement specifications that this design decision satisfies are as follows: 2.3.B, 2.10.A, 2.13.A [1]

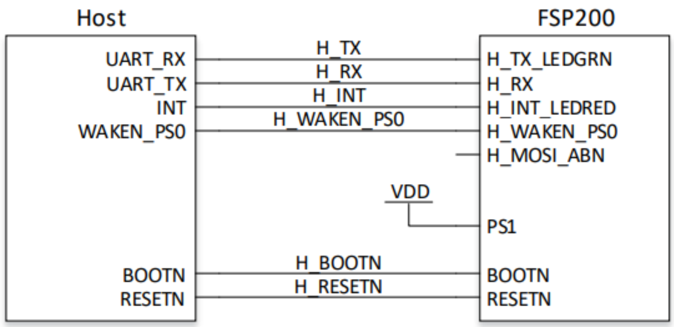

Figure 5: UART Connection Example to host Microprocessor [4]

2.4.2 Battery

The lithium-ion 2600 mAh battery is chosen because of its broad availability and usability. Most of these units come already wired with leads for the positive, negative, and ground terminals making it simple to connect to a PCB. The capacity can also be increased by having two of these batteries in parallel which will meet battery life requirements for the Levo. The requirement specifications that this design decision satisfies are as follows: 2.4.A, 2.5.B, 2.6.B, 2.7.B [1]

Figure 6: Battery to be used in Beta product [3]

2.5 Table of Sensor Unit and Microcontroller Designs

Des 2.1.A	The Levo will use a Raspberry pi 4 to communicate between the sensor and the mobile app and process data.
Des 2.2.A	A GY-521/MPU-6050 3 Axis Gyroscope and 3 Axis Accelerometer 6DOF Sensor Module will be used to collect acceleration data which is connected to the GPIO of the Raspberry Pi.
Des 2.3.A	Raspberry Pi will communicate with the mobile device via wifi by uploading the data to an web server.
Des 2.4.A	Raspberry Pi collects data and converts it to a JSON file before transmition.
Des 2.5.B	IMU processor will collect data from sensors via I2C.
Des 2.6.B	IMU processor will send data to microprocessor VIA UART
Des 2.7.B	Microprocessor will convert received data to a JSON file before transmission.
Des 2.8.B	Microprocessor will send data via Bluetooth 5.0 via its integrated BLE module to the mobile device.
Des 2.9.B	The Microprocessor, IMU, and sensors will be placed on a printed circuit board.
Des 2.10.B	Will use a 2600 mAh battery.

Table 1: List of Sensor Unit and Microcontroller Designs

3 Mobile App Design

The purpose of the mobile application is to communicate the data obtained from the Levo sensor unit and microcontroller in a meaningful way. The user must be able to view their immediate results, as well as track their performance over time. The app will save user's workout data, display data, and compare data from previous workouts. The mobile app will follow the Model-View-Controller model, where Model will store the workout data, View will display the workout data, and Controller will retrieve workout data and send it to the app. The mobile app will contain both the Controller and the View.

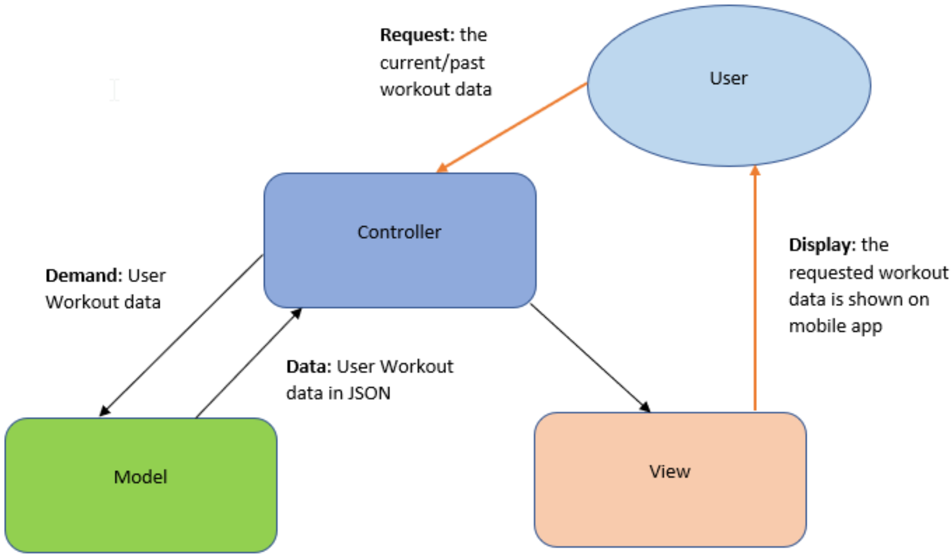

Figure 7: MVC Model of the mobile app

The current states of the Levo consist of sensor, Raspberry Pi, and the mobile app. The sensor will detect acceleration, and transmit the data to the Raspberry Pi. The Raspberry Pi will act as a signal processor and the web server where the mobile app can retrieve data. The requirements specification that this satisfies is Req 3.1.A [1]. The mobile app will act as a controller that would retrieve data from the server and display the data onto the mobile device.

3.1 Operating System and Language

The Levo app will be built on both iOS and Android devices using React Native and Expo. React Native framework allows developers to build mobile applications using JavaScript, and can be deployed on iOS and Android. The requirements specification that this satisfies is Req 3.4.A [1]. This will save Levo Technologies development resources needed to build on two different platforms. The Expo service allows for display and exporting of the app to both iOS and Android platforms. The app shall be compatible to minimum Android Lollipop 5 or greater in Android OS, and iOS 10 or greater on iPhones, to ensure that the minimum hardware is met to accept and install Expo on their device. The requirement specification that this satisfies is Req 3.5.A [1].

It is indeed possible to write the app on the native languages of the mobile device depending on the platform: Kotlin/Java in Android, and Swift in iOS. The goal is to implement cross-platform use ability, with minimal sacrifices to performance and functionality. Considering these factors, React Native and Expo are the best choices.

Figure 8: Functional Model-View-Controller

3.2 Graphical Display

To display the graphical analysis of a users data, the Levo application will be using the react-chartjs-2 package [7]. This package uses the MIT license, and is able to be changed and edited, provided the MIT license is added to the Levo code base. This package also has a small unpacked size of 262KB which means it will not significantly affect the application size. Additionally, this package is used by a large number of developers who provide documentation and examples using this package. The requirement specifications that this satisfies is Req 3.2.A and Req 3.7.A [1].

3.3 Future Mobile App Considerations

3.3.1 Size limit of the Mobile app

The Levo app will have limited size to be 500 MB. The standard guideline in iOS [5] is to have maximum size of 500 MB for the executable. Android has no specific app size limit, but each APK file must not exceed a maximum of 100 MB [6]. The requirements specification that this satisfies is Req 3.6.B [1].

3.3.2 Database for the Mobile App

The Levo application will use Google's Firebase to store and retrieve data. This is due to two of the Levo developer's familiarity with the service. Firebase is recognized as a NoSQL database. Other storage services include MySQL and PostgreSQL, but the Levo app will be using Firebase due to developer familiarity. The requirements specification that this satisfies is Req 3.9.B [1].

3.4 Table of Mobile App Design

Des 3.1.A	Application shall request data in JSON format from a web server.
Des 3.2.A	Application shall present repetitions, peak and average velocity during specific repetitions, and displacement tracking
Des 3.3.B	Application shall present repetitions, peak and average velocity during specific repetitions, peak and average acceleration during specific repetitions, peak and average force during specific repetitions, depth checking, and displacement tracking
Des 3.4.B	Application shall display users performance metrics using react-chartjs-2 package.
Des 3.5.A	Application shall be written using React Native for deployment natively on iOS and Android.
Des 3.6.B	Application size shall not exceed 500MB.
Des 3.6.B	Application shall run on iOS 14, and Android 11.
Des 3.7.B	The user shall create an account to gain access the application.
Des 3.8.B	Application shall store user data to Firebase upon internet connection.
Des 3.9.B	Application shall work with no internet connection in an offline state.
Des 3.10.B	Application shall be able to identify barbell weight through user input.
Des 3.11.B	Application shall be able to run in the background.
Des 3.12.B	Application shall not share personal data with any other third-party without user's consent.
Des 3.13.B	Application shall recognize when the user has finished a set.
Des 3.14.P	Application shall have user preference setting to set a choice of units.
Des 3.15.P	Application shall not interfere with accessibility features on the device.

Table 2: List of Mobile App Design

4 Conclusion

Levo Technologies is committed to delivering a high quality product to its users. The design choices outlined in this document will allow for the successful development of the Levo proof-of-concept prototype in a timely manner. While future considerations are still subject to change, the design choices in regards to the beta and production phase of the Levo will build upon the solidified design choices of the alpha phase of the Levo.

5 Glossary

API - Application Program Interface
APK - Android Package Kit
CCO - Chief Communications Officer
CSA - Canadian Standards Association
IMU - Inertial Measurement Unit
ISO - International Organization for Standardization
RPi - Raspberry Pi
Levo - Product Name
Levo Technologies - Company Name
OS - Operating System
DOF - Degree of freedom
GPIO - General purpose input output

References

[1] M.Chute, G.Fader, A.Kim, et al, "Requirement Specification," Levo Technologies, Burnaby, British Columbia, Canada, 2021

[2] "WayinTop 3pcs GY-521 MPU-6050 3 Axis Gyroscope and 3 Axis Accelerometer 6DOF Sensor Module 16 Bit AD Converter Data Output IIC I2C for Arduino," amazon.ca, Aug.20, 2020. [Online]. Available: `https://www.amazon.ca/gp/product/B07P5YZ7ZD/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1`. [Accessed Mar.26, 2021]

[3] "Rechargeable lithium ion battery cell 18650 3.7v 2600mah li ion battery with PCM NTC," amazin.ca, Aug.20,2020. [Online]. Availiable: https://www.amazon.ca/Rechargeable-lithium-battery-18650-2600mah/dp/B08D9HMSQQ. [Accessed: Mar. 26, 2021].

[4] CEVA Inc, "FPS200 Datasheet," FPS200 datasheet, 2019.

[5] "Maximum build size," help.apple.com, [Online]. Available: https://help.apple.com/app-store-connect//dev611e0a21f. [Accessed: Mar. 26, 2021].

[6] "Manage APK files," support.google.com, [Online]. Available: https://support.google.com/googleplay/android-developer/answer/9859152?hl=enzippy=%2Cmaximum-size-limit. [Accessed Mar. 26, 2021].

[7] “react-chartjs-2,” npmjs.com, [Online]. Available: https://www.npmjs.com/package/react-chartjs-2. [Accessed: Mar.26, 2021].

[8] D. Norman, *The Design of Everyday Things*. Basic Books (AZ), 2013.

[9] “Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts,” iso.org, 2018. [Online]. Available: https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/35/63500.html [Accessed Mar. 19, 2021].

[10] “Ergonomics of human–system interaction — Part 210: Human-centred design for interactive systems,”richardcornish.s3.amazon.com, 2010. [Online]. Available: https://richardcornish.s3.amazonaws.com/static/pdfs/iso-9241-210.pdf [Accessed Mar. 19, 2021].

[11] “Ergonomics of human-system interaction — Part 112: Principles for the presentation of information,” iso.org, 2017. [Online]. Available: https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/48/64840.html [Accessed Mar. 19, 2021].

[12] “Ergonomics of human-system interaction — Part 171: Guidance on software accessibility,” iso.org, 2018. [Online]. Available: https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/03/90/39080.html [Accessed Mar. 19, 2021].

[13] "CSA C22.2 No. 0.23-15 (R2020),"scc.ca, 2020. [Online]. Available: https://www.scc.ca/en/standardsdb/standards/28121. [Accessed: Mar. 21, 2021].

[14] "Ergonomics of human-system interaction — Part 110: Dialogue principles," iso.org, 2020. [Online]. Available: https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/03/80/38009.html [Accessed Mar. 19, 2021].

[15] "Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems," iso.org, 2019. [Online]. Available: https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/05/20/52075.html [Accessed Mar. 19, 2021].

A User Interface and Appearance Design

A.1 Introduction

The Levo is a data collection system to be implemented for usage by an athlete during exercise with barbells. This document and the facts and statements herein are to establish how design choices for the app UI and physical unit mechanisms were informed and met the relevant requirement specifications of the Levo. This document will also outline relevant engineering standards, the projected user base of the Levo, their expected level of expertise, and interactions with other mechanisms or similar apps to the Levo to help inform the Levo's final design specifications. Additionally, considerations for how to test both empirical and analytical methods will be considered as well as a technical analysis of all hardware and software components of the Levo that meet the seven elements of UI interaction from Don Norman's book Design of everyday things [8]. Finally, concept artwork for UI and sketches for the physical units mechanisms will be provided.

A.2 User Analysis

Among athletes there are many levels of competence and expertise in regards to both weight lifting and data analysis. Complex mathematical knowledge cannot be assumed on the user. Many athletes start weight training around grades 8 or 9 often in the pursuit of improvement in one or many sports. Athletes of this age will have very limited knowledge in math or physics– although concepts such as force, velocity and displacement will not be completely foreign to them. Unfortunately, not all lifters capabilities in math or physics will substantially improve past this level. While including some visual features to illustrate more complex concepts can be implemented, the Levo's UI design should not be built specifically around individuals outright lacking complex mathematical skills. By the nature of the Levo being a data analytics system, a certain level of familiarity with physics and math can be assumed for most users. A buyer or user of an analytic tool like the Levo can be assumed to have some understanding of how force, work, and other relevant metrics are present in weight lifting. Furthermore, it is not unlikely that a user of a Levo would have experience with other applications and devices for tracking athletic performance. This would include apps such as the "Nike Run Club," or Apple's "Fitness" application. The aforementioned apps provide statistics such as the user's average pace, and times for each kilometer run without the app needing to breakdown what the statistics mean for the user to understand. More complex data including heart rates and heart rate recoveries are also present for the user to read and interpret. The metrics tracked by the Levo should be considered sufficiently accessible to be presented in this way (Apple's Fitness app) to the most users. One final consideration that must be taken for the app UI of the Levo is its usability when dealing with a fatigued user. If buttons or components of the UI become too difficult for a fatigued user to navigate or interact with, athletes will not use the app. Fatigue is inevitable during exercise, so the users of the Levo can be assumed to be in a varying level of fatigue during large portions of its usage.

For structural design and the attachment mechanism of the Levo to the bar, it can be assumed that a Levo user is familiar with a variety of collar mechanisms for weightlifting. The typical Levo user can also be assumed to understand how to put on a watch/ Fitbit or other pieces of athletic data collecting gear. Furthermore, a Levo user should be assumed to understand that equipment like the Levo is susceptible to crushing and falling hazards in a weight room. Because the Levo is lightweight, no considerations must be made to the user's ability to lift a certain weight in order to obtain data from the Levo.

Because athletes are a competitive bunch, and a user of the Levo is most likely using the Levo to improve their performance in the weight room (and by extension the field), they may take the data obtained by the Levo to indicate what weight they can safely lift, or where they need to push themselves more. Because of this, we cannot assume that any users will be able to accurately predict their maximum capabilities in weight training based off the data the Levo acquires and presents. As such, the Levo must give clear and obvious warnings to the athlete about attempting too much weight, or behaving in an otherwise unsafe manner.

A.3 Technical Analysis

To conduct a technical analysis for the Levo, seven principles taken from Don Norman's book "The Design of Everyday Things" will act as criteria. Principles include discoverability, feedback, conceptual models, affordances, signifers, mappings, and constraints [8].

A.3.1 Discoverability

Discoverability is how easily the user can locate elements when interacting with the Levo for the first time. To limit the complexity of the Levo, it will have one switch and one button. The switch will be to power on and off the Levo, and the button will be for creating a connection over bluetooth. Both the power switch and bluetooth pairing button will be located on the front of the Levo to ensure easy and immediate access. In the Levo application, the landing page will immediately greet the user with four buttons which represent the four main components of the application. Each of these buttons will link to screens which will further guide the user towards the functionality they wish to perform. An example workflow which the Levo application could guide the user through is Home Page: (Lift) -> Lift Page: (Add exercise) -> Add Exercise[Input: movement, number of sets](GO) -> In Exercise Page: (Start Set). Another key component for discoverability within the application, is emphasizing buttons which users will be interacting with most often.

A.3.2 Feedback

The purpose of feedback is to indicate to the user if a certain function is working. It should also indicate the status of the particular function which is being performed. There are three feedback indicators that the Levo will be using. The first form of feedback will be regarding the power status of the Levo. This will be indicated to the user through the use of a multicolor LED where green represents 50 % charge, yellow between 50 and 25 % charge, and red below 25 % charge. The second form of feedback will be regarding whether the Levo is powered on or off. This will be indicated through a single colour LED which is powered on when the Levo is on, and off when the Levo is off. The last form of feedback will be regarding the bluetooth connection status of the Levo. This will be indicated through a single colour LED which will be powered on when the Levo is in discoverable mode. The Levo application will provide feedback to the user through visual prompts whenever needed. For example, indicating a workout has been started through a visual notification displayed to the screen "Set In Progress". Another example will include a visual notification when data has been successfully saved remotely to the database "Data successfully backed up."

A.3.3 Conceptual Models

The design of the Levo should allow the user to intuitively understand how the Levo works. Thus the Levo will place each of the corresponding feedback LEDs directly beside their corresponding button or switch, to ensure no confusion around which LED is providing feedback for what purpose.

A.3.4 Affordances

The purpose of affordances is to inform the user of the connection between an element, and how they interact with it. Due to the Levo's simplicity in design, our main objective is to make sure the users understand the functionality of the power switch and the bluetooth button. For example, indicating which direction to move the switch to power it on or off, and pressing and holding the bluetooth button to place it into discoverable mode.To make sure the user understands how to use each of these functions, the Levo will provide an easy to follow demo sheet with basic instructions involving each of the elements. In the Levo application, to ensure the user understands what function a button performs, the application will either provide a key word, or icon on the button to signify the functionality of the button.

A.3.5 Signifers

The purpose of signifiers is to identify to the user how they should be interacting with a particular element. To ensure there is no confusion around the purpose of the switch and button, we will include symbols above the power switch, bluetooth button to identify their purpose.

A.3.6 Mappings

The purpose of mapping is to assist the user in understanding what certain elements control. To aid in the users understanding, we have made sure to display specific information to the user only on relevant pages. For example, only display "displacement graphs" and "depth checks" on their individual set page.

A.3.7 Constraints

The purpose of constraints are to prevent the user from using the Levo improperly. The Levo application will provide the user with prompts if there are any input errors. For example, displaying a prompt "Incorrect password, please try again." if the user has not entered the correct password. Additionally, improper use of the bluetooth button will be handled by the embedded software of the Levo.

A.4 Graphical Presentation

The following figures show conceptual designs for both the attachable physical unit and app layout options to ensure clear, unambiguous and intuitive navigation for all users. These ideas are subject to change as we move further along in the development stage.

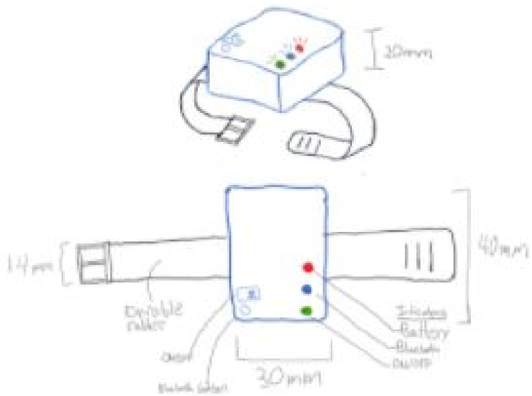

Figure 9: Illustration for the Attachable Physical Unit

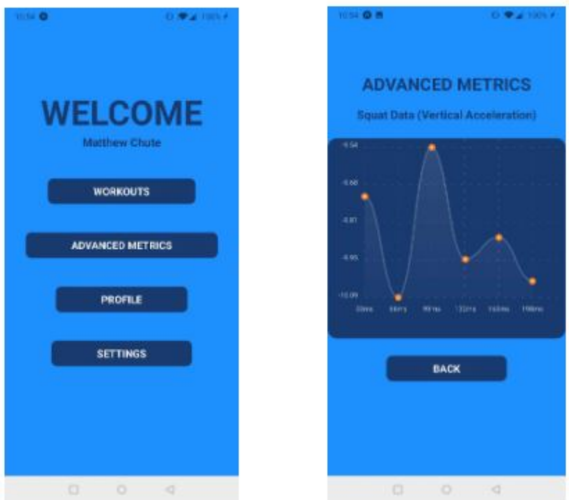

Figure 10: Sample Generated Screens for the Main and Advanced Metrics Pages

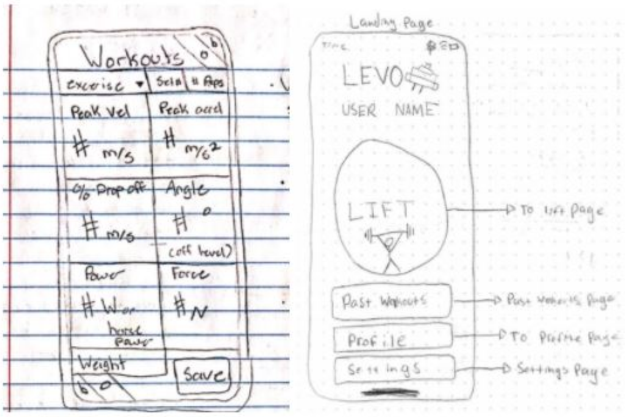

Figure 11: Sketches for Quick Reference Metrics During a Workout and Alternate Main Screen Ideas

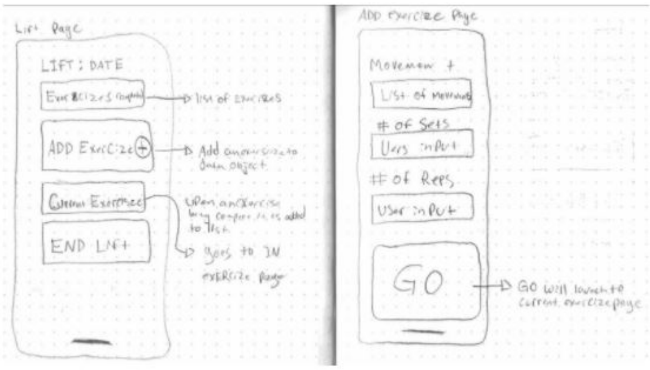

Figure 12: Sketches for Lift and Add Exercise Screen

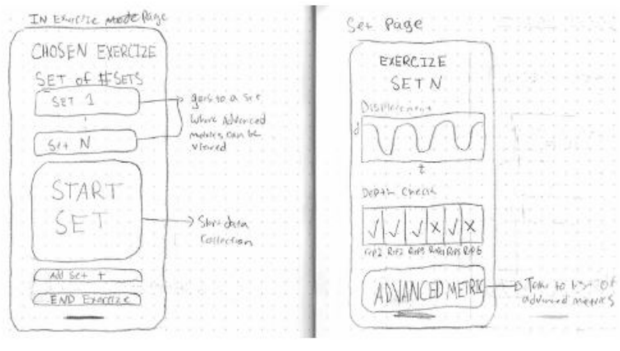

Figure 13: Sketches for Exercise and Set Number Screens

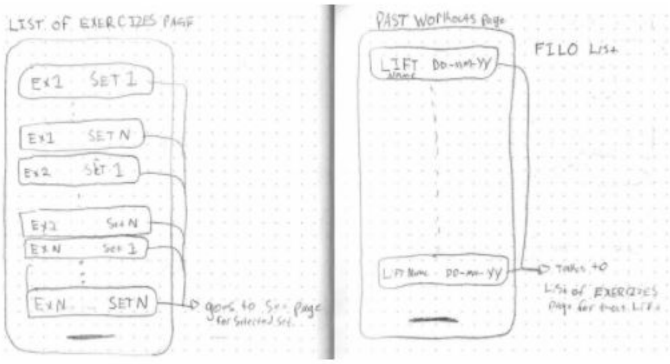

Figure 14: Sketches of Exercise List and Past Workouts Screens

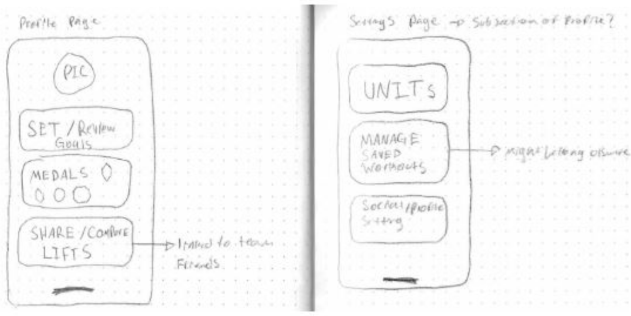

Figure 15: Sketches of Profile and Settings Screens

A.5 Analytical Usability Testing

During the full development cycle, constant feedback from potential users is needed in order to ensure the Levo is safe, reliable, and easy to use. While designing and evaluating the product throughout the proof-of-concept and prototyping phase, it is possible that the Levo is missing critical factors that would hinder an athlete during use of the Levo. To ensure key functionalities are not absent, a group of five to ten individuals will be polled for feedback until the final prototype is built. In the alpha phase, the Levo will be only tasked with accurate collection of data. UI features for the physical unit and the tests pertaining to them will be handled in the beta phase.

A.5.1 Test Procedures

During Alpha Phase

1. Turn on the raspberry pi and power the circuit on the breadboard.
2. Open up the LEVO app on the phone
3. Click on lift icon to start collecting data

4. Starting lifting and lowering the alpha phase unit
5. Go back to the app and press stop to pause collecting data
6. Click on Advanced Metrics button
7. Verify that the data collected is stored and can be displayed on the software

Beta Prototype Phase:

1. Attach the hardware onto a barbell
2. Open up the LEVO app on the phone
3. Click on lift icon to start collecting data
4. Complete a workout set with the barbell and the hardware module attached
5. Go back to the app, and press stop to stop collecting data
6. Click on "Advanced Metrics" icon
7. Verify that the data collected is stored and can be displayed on the software

A.5.2 Heuristic Evaluation

Unlike a formal evaluation with experimenter and the designer, heuristic evaluation allows users to freely discuss and give feedback of the product. Some benchmarks for UI usability during each phase of the design are as follows.

Alpha Phase:

- The software was able to collect and display the data from hardware module
- Navigation between different pages in the app makes sense and intuitive according to the test group
- The measurement collected during the test is sound and reasonable
- User understood the data being displayed

Beta Prototype Phase:

- User was able to properly latch on the module onto a barbell
- The hardware does not interfere with the user during workout session
- Data was transmitted and displayed correctly on the software

A.6 Empirical Usability Testing

Empirical testing methods can be split into physical unit usability and application usability. The physical unit will have usability tests regarding the attachment of the unit to a barbell and the interaction with its switches and buttons. The app usability will be linked to the user's ability to navigate and use the app.

End User Activity 1
Attachment of the Levo to the bar:

1. Were you able to attach the Levo securely to the barbell?
2. Where did you intuitively place the Levo on the bar?
3. Did you assume the Levo was used to stop weights from sliding?
4. Which way did you orient the Levo on the bar?

End User Activity 2
Power on the Levo and connect it via bluetooth to your mobile device:

1. Was the toggle switch obvious as to which was the on state?
2. Were the lights useful to convey the state on the Levo?
3. Was pressing the bluetooth button that turned on a blue light to signify pairing to bluetooth confusing?

End User Activity 3
Using the app and starting an exercise:

1. Was starting an exercise intuitive with the word choice we have given?
2. Did you think there was too much/too little/just the right amount of user input when starting your exercise?
3. Were you satisfied with the amount of metrics we should immediately after an exercise was completed?
4. Do you wish there was a pause button instead of a stop button?

End User Activity 4
Feedback after a strenuous workout:

1. Was it easy to use the app after an exhausting workout?
2. Are the buttons easily and logically placed so that you can quickly save your exercise and not worry if your exercise was properly recorded?
3. Would you wish for an easier way to go back and alter your past exercise?
4. Were the metrics we showed you right after an exercise beneficial to you as you are actively working out or did you not use them?

A.7 Engineering Standards

To ensure that the Levo meets the needs of our users several standards have been taken into consideration from the International Organization for Standardization (**ISO**) and Canadian Standards Association (**CSA**) . Specifically, the **ISO 9241** is a multiple part standard that covers several topics in Ergonomics of Human System Interaction. The following standards are the ones the Levo must meet to ensure a well designed user interface.

ISO 9241-11: Usability: Definitions and Concepts. This standard provides the framework and fundamentals of usability, and the main point that concerns the Levo is "extent to which a system, product or service can be used by specified users to achieve specific goals with effectiveness, efficiency and satisfaction in a specified context of use"[9], [10]. Using the PROFILE button on the Levo's corresponding app allows the user to customize their profile and workout experience through setting personal goals to achieve. By implementing a medal system for reaching certain goals should increase satisfaction for the app and increase motivation to better themselves at the gym.

ISO 9241-112: Principles for the Presentation of Information. This standard focuses on principles for software controlled systems presenting information by user interactive systems to help the user accomplish tasks. This information can be provided through visual, auditory, or tactile/haptic means [11]. The Levo app's main purpose is to display different metrics such as peak acceleration, velocity, depth, and force. These metrics will be displayed via visual means with large fonts on the workouts page for quick, easy reference. Graphs and charts presented within advanced metrics are posted for the sole purpose so that the athlete can analyze their performance and better improve themselves.

ISO 9241-171: Guidance on Software Accessibility. The purpose of this standard is to help guide software design so that it is accessible at work, at home, in education, and in public places. Accounts for a wide range of user abilities and takes concepts described in ISO 9241-11 [12]. Both the Levo and the app must be designed to be accessible at a gym, or at home if the user has a personal gym. Using the communication between devices with the bluetooth will allow for users to access the Levo any gym location.

CSA C22.2 No. 0.23-15 (R2020): General Requirements for Battery-Powered Appliances. Describes standards when using battery powered devices to ensure safety for the user [13].

The following standards are currently withdrawn as of 2019, however for the purposes of learning they still provide important standards to consider when designing user interactive systems.

ISO 9241-110: Dialogue Principles. This standard states several principles when designing a user interactive system to ensure creative processes to achieve a good user experience. Principles include suitability for the task, self-descriptiveness, conformity with user expectations, suitability for learning, controllability, error tolerance, and suitability for individualization [10],[14]. The purpose of the Levo and the app is to encourage growth and motivation towards weight training. All UI displays are easy to read, see and find, as well as are presented in a form that would make the most sense to the users. There is no point in presenting complex data if they can't understand it. The app provides plenty of ways for personalization to pursue goals and be rewarded for making achievements as stated in ISO 9241-11.

ISO 9241-210: Human-Centered Design for Interactive Systems.This standard provides recommendations and requirements for human centered design that involves software and hardware considerations within their life cycle. The purpose of this standard was to improve human interaction with the hardware and software [9],[15]. The Levo must be simple to maximise comfortable use for our users. Clear LED lights will be shown when the system is on/off or taking data, and easy removable batteries to keep the system working while at the gym. App is neatly organized and complements the metrics taken from the hardware for the user.

A.8 Conclusion

User interfaces and considerations are key points when in product development. In the Levo's current state the UI and most of the UI requirements are still subject to change upon receiving feedback from tests. The final specification of the UI will occur late the beta prototype phase. Hardware user interface considerations will not be implemented until the beta prototype phase. Software app UI is subject to change with feedback from the tests mentioned in this document. Several ideas are put in place to provide the simplest app navigation, however very few have been implemented as of the submission of this document. For the alpha phase only a couple basic screens will be shown, which facilitate navigation from the main screen to a data collection screen, and the basic metrics page.

B Hardware Alternatives

B.1 Alpha

B.1.1 LVDT sensor

The LVDT sensor would have been able to measure displacement of a barbell but finding one that was large enough to measure the amount of displacement needed for meaningful data of a barbell exercises for example would be very expensive, and also very cumbersome for the user making it not a viable option.

B.1.2 Eddy-Current sensor

Using an Eddy-Current sensor would have been worked to measure displacement of the metal exercise equipment we are trying to track, but it also would be very prone to the loads of metal equipment that are also in a typical gym environment, and just like an LVDT sensor, would require it to be set up on a stationary spot and meticulously placed such that it can get an accurate reading of the bar, which would have been a big burden to put on the user.

B.1.3 I2C-USB converter

An I2C-USB converter would have been able to send the data from the sensor to a computing device which fulfils the requirements of getting the data to a computing the device, but this would then involve creating a desktop application to instead of a mobile app, which would have set the Beta stage back since a lot of work would have been put into a desktop app, and then come time to design for the Beta stage all that software work would have been scrapped since the goal is to have a mobile app rather than a desktop application.

C Software Alternatives

Kotlin/Java: If React Native is not used, we would have to complete Android Development in Kotlin or Java, which is official language supported by Google.
Swift: Similar to Kotlin/Java, we would use Swift to develop an iOS app and on macOS
C#: C# supports cross-platform development, but Javascript seems to be easier than C, because JavaScript is closer to Python
C++: Possible, but not enough time to develop an app in C++
Objective-C: Old Language that can still be programmed for iOS, but outdated.

D Supporting Test Procedures

Name/Date:			
		Mobile Device App	
Relevant Specification(s)	Procedure	Results	Comments
Des 3.1.A Des 3.2.A	The app can retrieve data from the server and display on the device	Pass [ ] Fail [ ]	
Des 3.2.A	The app can move from main screen to advanced analytic screen and back	Pass [ ] Fail [ ]	
Des 3.5.A	The app both works on iOS and Android phones	Pass [ ] Fail [ ]	
		Sensor Unit and Microcontroller	
Des 2.1.A Des 2.2.A Des 2.3.A	start data collection from the mobile app and perform a movement with the sensor unit. Observe the terminal printouts to see data collection. Open the JSON file generated and saved to the same folder as the script. Observe if the data collected matches the movements of the sensor unit	Data Collected[ ] Data Saved to JSON[ ]	
Des 2.4A	Open the web server and check if the data on the web server matches the JSON file created by the script	Data Matches [ ]	