Climber's Ligament Injury Mitigation Band

Alexander Nguyen Author, Jubahed Qayum Author, and Joshua Ramos Author

Department of Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*—**In this paper, CLIMB, a system capable of monitoring, analyzing, and visualizing forces applied by the hand over time, is proposed. In the form of a glove-like wearable device, CLIMB aims to improve the state of training and rehabilitation in rock-climbing by providing climbers with an unobtrusive, injury-preventative training device. As a result, climbers using CLIMB will avoid overloading their pulley-joints during their workouts, and will identify areas of improvement in their rehabilitation/training via the CLIMB–mobile app.

***Index Terms*—**
B: Sensor baseline reading
CLIMB: Climber's Ligament Injury Mitigation band
Γ: Force threshold value
Hang-board: a training tool that helps improve hand and finger strength for climbing and bouldering
Pulley-injury: a tear or rupture of the ligamentous bands that hold the tendons in place in the fingers.
Pulley-joint: a ring-shaped ligament in the finger that holds the flexor tendons in place against the finger bones.
Pulley-tendon: a fibrous band of connective tissue that stabilizes the tendons in the fingers.

I. INTRODUCTION

THE INTERNATIONAL Federation of Sport Climbing (IFSC) estimates that as of 2019, there are approximately 44.5 million climbers around the world. Climbing (in all of its forms) is a sport that comes with an inherent risk of physical injury, and studies have found that as much as 81% [1] of climbers have experienced at least one injury over the course of a single year. Of these, finger (pulley) injuries were by far the most common, constituting approximately 61% of all injuries.

A pulley injury occurs when one or multiple pulleys in the fingers, categorized from A1 to A5, ruptures. The pulley ligaments are responsible for holding the finger tendons tight to the bones, so when they are overloaded with weight, they tear, unable to withstand the force of the constricting muscles attached to the tendons. [2] These injuries are painful in both the short term and during the long term recovery, and the most severe pulley injuries (Grade IV) require surgery and months of rehabilitation to recover. Thus, it is crucial for climbers to be able to prevent these types of injuries as best they can.

There are several ways for climbers to mitigate the risk of pulley injuries; the most prominent of these solutions is the PulleyPal, a finger splint which aims to support the pulley by squeezing it against the finger bone. Additionally, climbers can train their fingers to increase pulley strength and increase the maximum load of each finger. However, while careful training and finger splints aim to assist the pulley ligaments through physical support and strengthening, there is currently no solution on the market that can accurately predict and prevent these injuries before they occur in real time. This project aims to fill the gap in pulley injury prevention for climbers.

The primary goal of CLIMB is to mitigate the risk of pulley injuries occurring during a climbing session by equipping climbers with a wearable device that can monitor force readings from the fingers, determine the moment at which a pulley injury is likely to occur, and immediately alert the user in time for them to return to a safe position. The secondary goal is to further assist climbers in their training and preparation by providing the wearer with weight distribution metrics and data analysis after each climbing session through a mobile application.

II. USE-CASE REQUIREMENTS

A. Ergonomics: Comfortable and Unobtrusive

The device will be comfortable and unobtrusive to the climber. It is imperative that the structure of the wearable covers no more than 30% of a climber's finger surface area and is flexible to avoid applying restraint. This is also so that climbers can preserve their grip (surface area contact with the wall).

B. Durability

The device will be robust and ready to tackle any climbing situation. Therefore, the glove and sensors will support 90 degree bends while accurately measuring up to 120 pounds of direct, perpendicular force per joint over numerous uses.

C. Safety: Injury Prevention

The device will actively work to prevent pulley injuries. Thus, it will maintain an on-device haptic alarm system that fires within 200ms whenever a user applies force on a pulley-joint that encroaches the safety boundary. Further, the obtained sensor readings must be accurate and fine-grained with a maximum error of ±2.5%,

D. Economic: Mitigate Healthcare Costs

The device will eliminate the need for surgical intervention due to pulley-injury. By preventing pulley-injuries, this device will relieve climber's of the potential $9,532 price-tag [4] for tendon repair.

E. Social: Feedback-driven

Lastly, the visualization component of the system, which provides user's analytical feedback, will be seamless, user-friendly, and suggestive toward improvement. To achieve this, the force readings from any given workout will be communicated to the mobile application within 10s of workout completion. Further, user's will be able to compare their training progress with others.

III. ARCHITECTURE AND/OR PRINCIPLE OF OPERATION

The architecture of the system is composed of two central modules: hardware and software. Together, these components operate to obtain applied-force readings, raise haptic alarms, process the readings into suggestive metrics, and visualize these metrics as analytics on a mobile application.

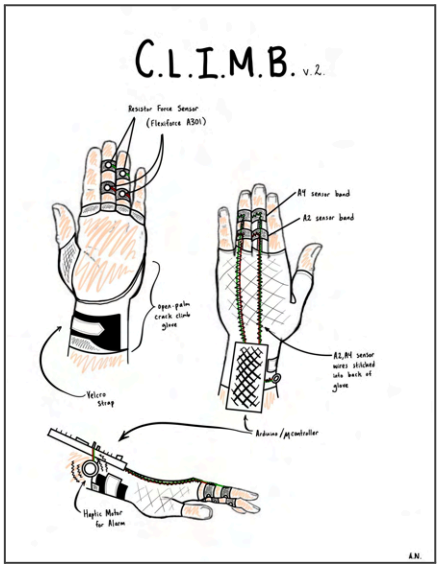


Fig 1. Wearable device

In Figure 1, the placement of supporting hardware (ie. PCB, Sensors) and the structure of the fabric encompassing the hand is emphasized to depict the flexibility and unobtrusiveness of the device. This component acts mainly as the data-collection layer of the design by procuring force readings over an interval of time and packaging/shipping them to the processing layer. The secondary operation of this component is to drive a haptic alarm upon the encroachment of *dangerous* force-reading levels.

In Figure 2, the mobile application describes the obtained force readings analytically. Prior to displaying these analytics, the readings are processed and stored on the mobile device. Additionally, the haptic-alarm threshold can be set via the mobile application.

In Figure 3, the system and its subsystems are depicted to describe the flow of data. The implementation details of this system will be further discussed in section VI.

Overall, the hardware is maintained by a lightweight, flexible fabric, and the software requires a smart-device to access the mobile application.

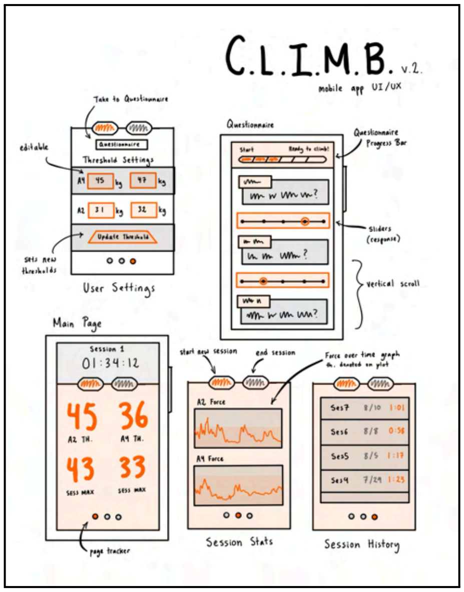


Fig 2. Mobile application

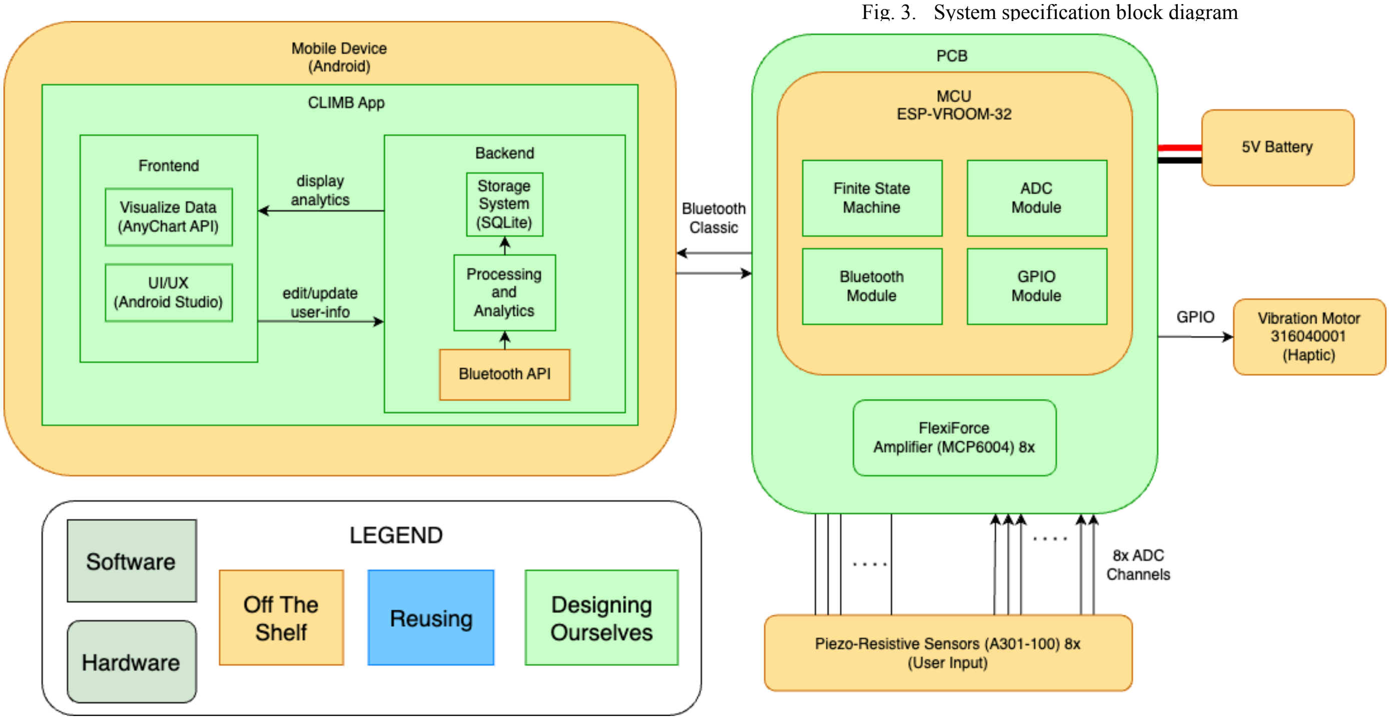


Fig. 3. System specification block diagram

IV. DESIGN REQUIREMENTS

A. *Ergonomic Requirements*

The hardware component will be supported by a standard fabric crack climbing glove, which is known for its durability, comfort, and minimal invasiveness. Furthermore, thin and flexible sensors will be used for data-collection. Since the average adult finger is characterized with a length and width of 80mm x 20mm [3], then to achieve a maximum 30% area reduction with 2 sensors per finger:

$$80\,mm \times 20\,mm = 1600\,mm^2$$

$$(1600\,mm^2 \times 0.3)\,/\,2 = 240\,mm^2$$

$$\sqrt{240\,mm^2} = 15.49\,mm$$

The largest size sensor the design will accommodate is 15.49 mm x 15.49 mm.

B. *Durability Requirements*

The sensor must be capable of bending 90 degrees while undergoing 534N (54 kg) of constant, direct force without tearing. Over the course of 100 uses, the sensor must not lose more than 2.5% of sensitivity and the wearable fabric must not tear or lose formation.

C. *Safety Requirements*

The sensors will be calibrated upon initialization, and the force readings will be measured accurately with fine-grain by amplifying the sensor reading range to support up to 54 kg. Further, using on-device peripherals, changes of at least 5mV/kg will be identified to ensure sensor precision. Moreover, readings within 20% of the safety threshold will signal the alarm subsystem within 100ms to drive the first haptic. Upon threshold excession, a second signal will be fired within 100ms to drive the first haptic again and begin driving a second haptic.

D. *Economic Requirements*

The economic requirements are directly dependent on the safety requirements. By preventing injuries, worst-case healthcare costs are mitigated.

E. *Social / Feedback Requirements*

To ensure data transport within 10s, there must be a sufficient communication speed. Given a workout of 10 minutes, at a 100 Hz sampling rate, with 16-bit force reading values:

$$10\,min \times 60\,sec/min \times 100/sec \times 16\,bits = 9.6\,Mb$$

$$9.6\,Mbits\,/\,10\,sec = 96000\,bps \Rightarrow 96\,Kbps$$

The system needs to maintain a minimum data transfer speed of 96 Kbps.

V. DESIGN TRADE STUDIES

A. *MCU Platform*

Four criteria for selecting an MCU were considered: computational capability, power consumption, size, and connectivity. As described in Table I, the Raspberry Pi is certainly the most computationally capable of the controllers, but it consumes significantly more power, is much larger, and supports more connection methods. While it does contain Bluetooth capabilities, it lacks built-in peripherals ADC, so its use in this system would require additional equipment, which would add more size. The STM32F4E platform has the lowest power consumption of the three, but it would require additional modules to support Bluetooth, increasing its overall size factor and power consumption.

TABLE I. COMPARISON OF MICROCONTROLLER PLATFORMS

Attributes	**MCU Platform**		
	STM32F4E	***ESP-WROOM 32***	***RPi***
Clock Speed	160 MHz	240 MHz	Up to 1.8 GHz
Connectivity	ADC, GPIO, UART, I2C, (Bluetooth with HC-05)	ADC, GPIO, UART, I2C, Bluetooth, WiFI	GPIO, WiFi, Bluetooth, HDMI, USB, Ethernet, PCIe
Power Consumption	0.3 W	0.6 W	<= 6W
Built-in Exposed ADCs	9	15	0
Max Storage	4 GB	4 GB	8 GB
RAM	256 KB	320 KB	1 GB
Size	68 mm x 53 mm x 13 mm	25 mm x 18 mm x 3 mm	85 mm x 56 mm x 17 mm
Familiarity	Very	None	None

While the ESP32 consumes slightly more power than the STM32, the miniature size, built-in Bluetooth capabilities, peripherals providing extensibility, and more capable CPU contribute to its selection. Furthermore, although the system developers lack familiarity with the ESP32, it is supported by numerous Arduino libraries and documentation, allowing for quick understanding, iteration, and development.

B. *Force Sensors*

To support this system, durable, thin, and flexible force-sensitive resistors were investigated to determine a suitable sensor for obtaining accurate force readings. Beyond the sensors considered in this trade-study, there are a number of force-sensitive devices for obtaining force readings, however, to abide by the system's design requirements, piezo-resistive sensors were selected due to their lightweight and simple nature. Of the contenders, the Tekscan A301-100 was selected.

Table II. Comparison of Sensors

Attributes	**Force Sensitive Resistors (FSRs)**		
	Interlink FSR 402	***Tekscan A301-100***	***ALPHA MF01A-N-221-A06***
Diameter	18.28mm	14.00 mm	14.7 mm
Thickness	0.2 – 1.25mm	0.203 mm	0.205 mm
Flexibility	yes	yes	yes
Repeatability Accuracy	±2%	±2.5%	±2%
Durability (# actuations)	≥ 10 million	≥ 3 million	> 10 million
Range	0.1N – 100N	0N – 4,448N	0.3N – 9.8N
Response Time	< 3µs	< 5µs	< 5 ms

In comparison, the Tekscan has comparable thickness, flexibility, and response time. In contrast, the Tekscan is less durable and slightly less accurate, however, it provides a significantly larger force-range and smaller diameter.

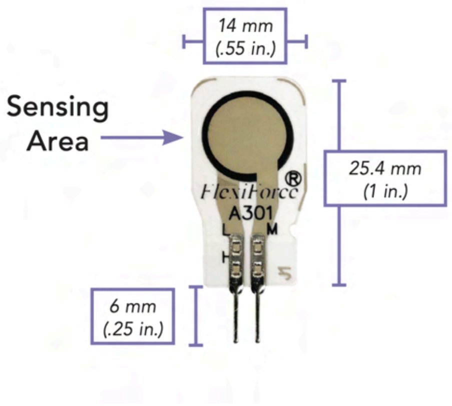


Fig 4. Tekscan FlexiForce A301-100 Sensor [7]

Overall, the deciding factors were the A301's range and diameter. To satisfy the design requirements, the sensor must read up to 54 kg (534N) and obtrude up to 30% of the finger. The Interlink option fails to support the area requirements, while both fail to satisfy the force-reading requirements; thus, the Tekscan is the sensor for this application. [5] [6]

C. Communication Paradigm

The process by which force-readings will be transported to the processing layer will follow one of two schemes: (1) send the force readings in real-time in parallel with measurement, or (2) collect the force readings, and send them all at the end of a session.

Option (1) has the benefit of instant, real-time analytics. A user can see the data propagating throughout the system during their climb. However, this approach implies that a climber needs to keep their phone nearby to sustain connectivity while climbing, potentially increasing the risk of damage or injury if their phone is hanging around in their pocket or if the climber falls on it.

Furthermore, option (1) introduces more overhead, whereas option (2) relieves the system of real-time complexities such as scheduling and interrupt-handling, providing a more simplistic approach. Moreover, because climber's are focused on their climbing, it is unnecessary to provide real-time analytics as it provides no value to the climber, and fails to address any design requirements.

While it may require larger memory utilization on-device, the risk of injury or disconnection is mitigated, and overhead and complexity is minimized with option (2).

D. Android App Development Platform

When choosing an android app development platform, both Android Studio/Kotlin and the Flutter/Dart framework were considered. Developing apps in Flutter has several benefits, including cross platform capabilities, a diverse set of widgets, and the ability to use a single codebase for multiple platforms.

In contrast, app development in Kotlin has the built in advantage of native performance, which comes as a result of Kotlin compiling to native code (which typically results in smoother execution when compared to Flutter's rendering engine). Additionally, when combined with Android Studio, developing in Kotlin gives the developer access to the full breadth of available Android APIs making it easy to integrate with native features, services, and libraries. Android Studio also provides a rich array of tools for android development, including an Android phone emulator which is incredibly useful for debugging and testing.

Each system has many unique benefits, however in the end Android Studio/Kotlin was selected for frontend and backend development for its advantages specifically in native Android development.

VI. System Implementation

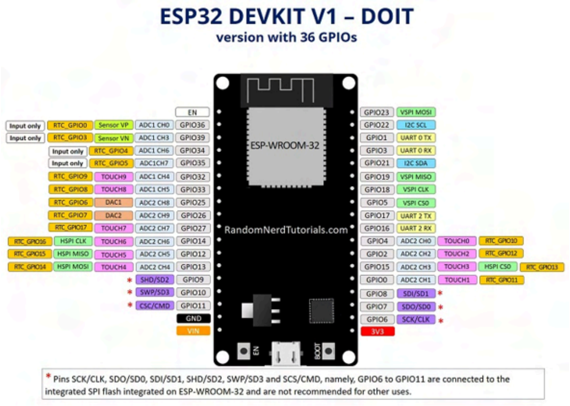
The following implementation describes the architecture for a 1 handed device. To support two hands, simply double the system architecture with two PCBs, two MCU, etc. Though there will be slight modifications in the reception/transmission of user-data. This will be described in the *Dual-Wielding* section.

A. Printed Circuit Board

At the base of this system is a printed circuit board maintaining all of the electrical components. In particular, it will allow for the mounting of an MCU (Fig. 5) via female header pins to allow for flexible device testing. It will maintain a power-jack to provide 3.3V from an external power-supply or standalone battery to allow for wired or on-the-go use. It will route the power-jack to the MCU and the operational amplifiers. It will route the MCU's ADC units to the amplified sensor reading outputs for data collection. It will route the MCU GPIO pins to two haptic vibration motors (see Fig. 7) which comprises the alarm system. It will route the amplifier circuit depicted in Fig. 6. Lastly, it will serve as a strong, durable base for the device. This PCB will be designed using the Fusion CAD software and manufactured locally.

B. *Processing Unit: ESP-WROOM-32*

Aboard the ESP-WROOM-32, the Arduino ESP32 libraries will be used to program the controller's peripherals to interface with the surrounding electrical components. Furthermore, a Bluetooth (Classic) procedure will be implemented to allow for serial communication with the mobile device. Upon initialization, the MCU will wait for a start signal from the mobile device via Bluetooth. This signal will be followed by a user-set force threshold value that will be converted to an ADC threshold value.



* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.

Fig 5. ESP32 Layout

The ADC units will be programmed to read from 4 channels at a 100 Hz frequency with a 10-bit resolution such that miniscule changes in force readings can be detected. And the GPIO units will be programmed to interface the system haptics when the ADC readings exceed the user-set force threshold. After receiving the start signal, the MCU will begin processing the readings, constantly monitoring the threshold and recording the readings in a buffer. Upon receiving a stop signal from the mobile device, the MCU will stop collection and begin data transmission to the mobile device. It will then clear the buffer, and idle until the next session starts.

C. *Sensing and Amplification*

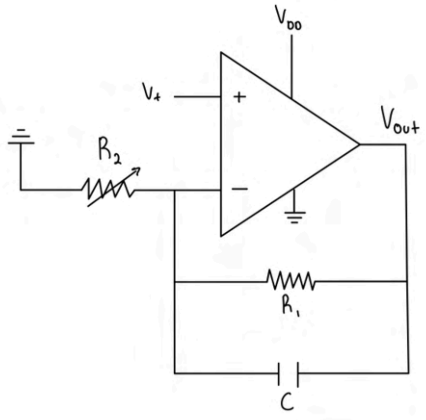


Fig 6. Sensor Amplifier Circuit

The A301-100 sensors (Fig. 4) will be wired as described in Fig. 6 to a non-inverting operational amplifier, resulting in the following gain representation and output:

$$V_{out} = V_{+}(1 + \frac{R1}{R2})$$

$$R_1 = 1\,M\Omega,\ R2 = [500K\Omega,\ 5M\Omega] + V_{DD} = 3.3V$$

$$V_{out} \in [1.2V,\ 3V]$$

This model provides a 1.8V range for force-detection. Additionally, this falls within the ESP32's ADC voltage range of 0V to 3.3V, utilizing ~55% of it. Further, the sensors will be stitched onto the wearable fabric while having their prongs directed towards the back of the hand, where they will be wired to the MCU's ADCs via an unobtrusive wire.

D. *Peripheral Engagement*

The on-device haptics will consist of two vibration motors shown in Fig. 7 [8]. These motors will be driven whenever they receive a current from their respective GPIO pin.

Fig 7. Haptic Vibration Motor [9]

Their GPIO pins will receive a signal to begin driving the motors from the MCU when the ADC reading values encroach on the safety threshold. The device will also have a button on it to turn off this alarm.

E. *Android Application Implementation*

The user can access all of the weight distribution data collected during their session through the mobile app on their Android device. This app will be built in Android Studio and both the frontend and backend will be written using Kotlin. There are three four main screens that the user can interact with: the landing screen, the session statistics screen, the session history screen, and the user settings screen.

Upon opening the app, the user will see the landing screen, where key metrics and data points from the last recorded session will be displayed. The session statistics screen will display graphs of raw force data over time for each pulley that has been monitored. The session history screen will display a list of all past climbing sessions and will allow the user to view a summary of that session's statistics. Finally, the user settings screen will give the user the ability to manually change the alarm thresholds for each individual pulley sensor on the wearable device. In order to update these thresholds in the wearable device, the device must be paired and connected via Bluetooth with the user's mobile device. Once the device is connected the user can then press the "Update Thresholds" button in the user settings screen and the alarm thresholds will be updated in the wearable device for the next session. For the benefit of the user, a questionnaire will also be included which, when completed, will provide recommended thresholds for the user to consider sending to the wearable device.

To start and end a climbing (force tracking) session, there are two buttons (start new session and end session) that can be pressed. These buttons will be available no matter which of the four screens the user is on. Once the session has been completed, data collected by the wearable device will be received by the app via Bluetooth. After this data has been processed and analyzed, each screen will be populated with the most recent session's data accordingly.

F. Sensor Calibration Implementation

The piezoresistive sensors will be attached to the fingers through elastic bands wrapped around both the sensor and the finger at each pulley location. Because of this, there will be some force applied to the sensors by the elastic band which will need to be accounted for during the session. Adjusted alarm thresholds will be calculated by a calibration routine which will run before each session begins.

When the user starts a session, the app will make sure that the wearable device is paired and connected to the mobile device. Once this has been confirmed, the app and wearable device will commence the sensor calibration routine. First, the app will begin the routine by sending a start signal, along with the alarm thresholds for each pulley sensor, to the wearable device. When the wearable device receives this start signal, it will take sample force readings over a period of 10 seconds. The user will also need to be wearing the wearable device during calibration or the calibration will not succeed, so if the returned readings are abnormally low (close to zero kgs), the calibration will fail and require the user to start a new session. If the calibration fails, the wearable device will send a NACK signal back to the app, which will let the app know that the calibration has failed. After receiving the NACK signal, the app will terminate and delete the session. As a safeguard, there will be an indicator on the app screen during calibration telling the user not to disconnect or take off the wearable device while calibrating.

Once non-zero (legitimate) force readings have been returned, the MCU will calculate the baseline force measurement for the session for each individual sensor. This will be calculated individually for each sensor by taking the mean of all force readings from each sensor taken over the ten second calibration period using the mean equation:

$$B_i = (\Sigma\, x_1, x_2, x_3, x_4, \ldots, x_n) \,/\, n\,,$$

for $i = 1,2,3,4$ and n = total number of readings

After the baseline force reading for the A2 pulley sensor on the index and middle fingers (B_1, B_2 respectively) and the baseline force reading for the A4 pulley sensor on the index and middle fingers (B_1, B_2 respectively) have been calculated, the adjusted alarm thresholds must be calculated accordingly, before being sent to the wearable device to be updated. In order to do so, the raw thresholds (Γ_{raw}) that were sent from the app must be converted from grams (g) to a converted threshold (Γ_{conv}) in millivolts (mv):

$$\Gamma_{conv,i} = -548 + 217 ln(\Gamma_{raw,i})\text{, for i = 1,2,3,4}$$

where Γ_{conv} is in mv and Γ_{raw} is in g.

Once the raw thresholds from the app have been converted to millivolts, the adjusted alarm threshold (Γ_{adj}) for each pulley sensor can be calculated:

$$\Gamma_{adj,i} = \Gamma_{conv,i} - B_i\text{, for i = 1,2,3,4}$$

where Γ_{conv} = corresponding threshold set in the app

Finally, when the calibration routine has successfully completed (and the updated adjusted thresholds have been sent to the wearable device), the wearable device will send an ACK signal back to the app to indicate successful calibration and the session will start.

G. Data Visualization & Analysis

When the session ends, the app will attempt to pair and reconnect with the wearable via Bluetooth. Once the wearable is connected to the app, it will send a signal to the wearable device indicating the end of the session. Once the wearable receives this signal, it will stop collecting force data and will send all data from the session to the app via Bluetooth. When all the data has been sent to the app, the wearable will send a signal to the app indicating that all data has been transmitted. This data will be stored in a database (further elaborated upon in the next section).

When analyzing the data, the force readings received from the wearable will be converted from millivolts to grams:

$$f_g = e^{((f_{mv}+548)/217)},$$

where f_g is the force reading in grams and

f_{mv} is the force reading in millivolts

Then, key data points (session maximum force, session average force, high risk percentage) will be extracted from the converted force readings from each sensor. Using AnyChart API, the converted session data will be displayed as a line graph with a y-axis of force (in grams) and x-axis of time (in seconds). The preset alarm thresholds of each pulley sensor will also be plotted on the graph to show force readings with respect to the high risk threshold.

H. Database Implementation

The database plays an integral role in the functioning of the app. It is the central, connecting point between the hardware platform and the app frontend. At its core, it is based on SQLite, an embedded open-source database management system. The largest benefit of using SQLite for the database is that it is lightweight and universally compatible: in other words, it allows the app (even if running on a moderately weak mobile device) to store and query data in a completely standalone manner, without needing to interact with external factors. Many more complex database management systems are not designed for this workload and would typically be hosted via cloud services. Because of this, the entire platform can run without the need for an internet connection.

Nonetheless, the database schema comprises of two tables:

workout_data

time	force_reading	workout_id
INT	INT	INT

thresh

threshold
INT

Fig 8. Database Schema

The table 'workout_data' stores force readings received after a workout ends. For each force reading *fr* received, a new row is created on this table, where the time column is occupied by the clock time on the mobile device, the force_reading column is occupied by the value *fr*, and the workout_id column is occupied by a unique, non-null integer generated at the start of the workout. When a user aims to retrieve data from the database via graph visualizations, this workout_id column is what will be queried to differentiate between workouts.

The table 'thresh' will contain one row and one column. This is the threshold value for the haptic alarm that the end user configures through the app frontend.

I. Wearable Device Fabrication

The wearable device housing (glove) will be custom-made from a mix of thick fabric and elastic polyester using a sewing machine. The glove backing and wrist wrap will be constructed of thick fabric to ensure durability and comfort, while the bands holding the sensors to the fingers will be constructed of elastic polyester to minimize invasiveness and allow the bands to snugly fit variously sized fingers. A velcro strap around the wrist will serve to tighten the glove and ensure that it does not move while in use.

J. Dual Wielding

To support dual-wielded CLIMB, the only change occurs on the mobile-device-end in regard to Bluetooth communication. The mobile device's Bluetooth controller will be modified to pair with both CLIMB devices, receive serial data from both, and transmit start/stop signals to both. On the hardware side, the system's natural modularity allows this change on the mobile-end to have no effect on the hardware end.

VII. TEST, VERIFICATION AND VALIDATION

A. Ergonomic Validation

To validate the system's ergonomic requirements, a survey will be conducted on 50 randomly selected participants. Each participant will provide the shape characteristics of their fingers. With this information, the system sensor size requirements will be tested to only impose up to 30% surface area coverage for at least 90% of participants' fingers.

This meets the system's ergonomic use-case requirements by providing a comfortable and unobtrusive wearable device that covers no more than 30% of a user's finger.

B. Durability Verification and Validation

To verify the system's durability requirements, the device will undergo 50 uses, each for 5 minutes with a constant 10kg applied at a 90 degree angle. This will be repeated for 50kg as well. After each, the sensitivity loss of the device will be verified to be within 2.5% and the wearable fabric will be examined for any tears or disfigurations.

To validate durability, the device will undergo dynamic weight applied by a user at various angles for 1-hour, 10 times. After each, the sensitivity loss of the device will be validated to be within 2.5% and the wearable fabric will be examined for any tears or disfigurations.

This meets the system's durability use-case requirements by tackling even the most extreme cases (54kg/sensor) and verifying that the system remains robust.

C. Safety Verification

To verify that the system initializes properly, the system will be booted up and calibrated 50 times. The system shall successfully calibrate 95% of the time.

To verify sensor precision, incremental weights will be placed atop the sensor (1kg, 2kg, 3kg, etc.) and a voltmeter will be used to monitor the voltage changes per kg. The system shall reflect a 5mV/kg voltage change up to 54 kg.

To verify that an alarm signal is triggered within 100ms when the force reading is within 20% of the safety threshold, an instant force exceeding the threshold will be applied after initialization, and an external timer will record the delay. The system shall fire its alarm within 100ms at least 98% of the time.

To verify the second haptic's functionality, the first haptic will be disconnected. Immediate force will be applied until the threshold is reached. The system shall fire its second alarm within 200ms of threshold excession with a 98% success rate.

This meets the system's safety use-case requirements by demonstrating that the system will actively work to prevent pulley injuries by remaining with a maximum error of ±2.5 and firing alarms within 200ms.

D. Economic Verification

The economic use-case requirements are directly dependent on the safety use-case requirements. By preventing injuries, worst-case healthcare costs are mitigated. And in the previous section, injury prevention is verified.

E. Social / Feedback Verification

To verify the feedback requirements of the system, tests will be performed using mocked force readings to verify the minimum data transfer. To do this, a mocked buffer input of 96 Kb will be generated and communicated to the mobile device. This test will be timed, recorded, and repeated 50 times to verify that the shipment completes in 10s with a 95% success rate.

To verify the user-experience related requirements, 96Kb of data will be mocked on the mobile device and passed to the database handler to ensure that information is stored properly and can be pulled to display on the application charts. This test will be repeated 50 times to verify that the flow of data on the mobile device can reach the user's screen with a 95% success rate.

These tests meet the use-case requirements by verifying that the system is providing users with a seamless experience in accessing analytical feedback that is suggestive towards improvement and can be used to compare progress with others.

F. End-to-end Verification and Validation

To verify end-to-end integration, the system will undergo 100 simulations where mocked force readings will be generated and communicated to the mobile device, testing to determine if the data is successfully displayed analytically 95% of the time.

To validate the system, a test using a hang-board with force-sensors mounted will be used to perform dynamic tests. The force on the hand-sensors will be cross-referenced with the force-sensors on the board. This test will be conducted 20 times for 10 minutes each. The system will be tested for the average error to be below 2.5%

VIII. PROJECT MANAGEMENT

A. Schedule

The project schedule breaks tasks down into five main categories: Project Deadlines, Wearable Device Design, Embedded Implementation, Mobile App Design and Implementation, and Integration and Unit Testing. Each task has a team member who is primarily responsible for its completion, although there is overlap and collaboration on every task is expected. The GANTT chart used to visualize the schedule can be found at the end of the document as Table I.

B. Team Member Responsibilities

Alexander is primarily responsible for developing the Android mobile application, including the front end UX/UI, overall app architecture, and functionalities where the app interacts with the wearable device (calibration, bluetooth pairing, etc.) and database (data visualization/analysis). He is also responsible for design and construction of the wearable device housing (glove).

Jubahed is primarily responsible for end-to-end Bluetooth connectivity setup, as well as the application backend. This includes processing incoming serial data, keeping relevant information through filtering and processing, and inputting into the SQLite database. He is also responsible for allowing frontend elements to interface with the database safely and securely.

Joshua is primarily responsible for the development of the hardware platform. This includes designing the PCB, programming the MCU to interface peripherals and communicate via Bluetooth, testing and integrating sensors and haptics, and designing circuit schematics.

C. Bill of Materials and Budget

Our current projected total cost for our project is $174.45. This will likely increase as additional materials are acquired for testing and replacement parts. The full bill of materials and budget can be found at the end of the document as Table IV.

D. Risk Mitigation Plans

There are only two risky scenarios identified that can possibly prevent the system from delivering on the use-case requirements.

(1) The haptic alarm does not fire or it is not alarming enough. In this scenario, the user has already encroached on their force threshold. In the worst case, the user exceeds the threshold and tears their pulley-tendon. At best, the user will continue to use the product and eventually discover that it is not alarming, or tear a pulley-tendon. To mitigate this case, a second haptic is introduced and fired within 100ms of threshold excession. This introduces a second layer of safety in case of system failure or user-error.

(2) The force-sensors are inaccurate. Under shear force, the sensors may lose sensitivity since they will be absorbing force from indirect angles. To mitigate the scenario where too high or too low readings are recorded, the system will cross-reference readings across neighboring sensors. This introduces a layer of real-time validation before the readings are compared to the threshold.

IX. RELATED WORK

Currently, there are no products like CLIMB. However, there are products with a similar goal to CLIMB. For instance the NiceClimbs PulleyPal [10], a pulley-injury rehabilitation device that operates purely mechanically by clamping down torn pulley-tendons to the bone. This is very similar to the pulley-splint [11], a common miniature mechanical contraption that compresses the area around a climber's finger to keep the tendon squeezed to the bone. And besides physical products, there are rehabilitation/training programs climbers can purchase if they need guidance when navigating pulley injury rehabilitation. For instance, the Rock Rehab Protocol claims to "take you step-by-step through the entire rehabilitation process." [12] And there are many more products and guides like these out on the market. However, all of them fail to solve the central issue: pulley-injury prevention.

X. SUMMARY

CLIMB aims to greatly improve safety in the sport of rock climbing by equipping climbers with a wearable device that can monitor the weight distribution across their fingers and alert them via a haptic motor when the risk of pulley injuries is dangerously high. There currently are no real-time injury prevention devices available on the market for climbers to use. Our product provides a solution to this problem, not only by providing near-instantaneous injury prevention while climbing via a wearable device, but through data analytics and visualization on the CLIMB mobile app.

Pulley injuries are common amongst climbers, and the pain of the injury and long recovery process are dreaded. With CLIMB, pulley injuries will no longer be a source of concern, and climbers will climb confidently, knowing that they will not be hurt by the sport they enjoy.

Although CLIMB aims to fully prevent pulley injuries from happening, there are several challenges that will need to be addressed during development. The reliability and durability of the piezoresistive sensors will be tested thoroughly, since climbing is a rigorous activity which will strain the construction of the device and its sensors, and it would endanger the climber to climb while the device is not collecting accurate readings. It is also key to ensure the reliability of the alarm system, for the same reasons. Overcoming these challenges will be vital to achieving our goals and delivering a wearable device that will protect climbers for years to come.

GLOSSARY OF ACRONYMS

ADC – Analog to Digital Converter
ACK – Acknowledge
ESP-VROOM-32 – Espressif Systems Microcontroller Board
GPIO – General Purpose Input-Output
NACK – Not Acknowledged
MCU – Microcontroller Unit
PCB – Printed Circuit Board
RPi – Raspberry Pi

REFERENCES

[1] Josephsen G, Shinneman S, Tamayo-Sarver J, et al. “Injuries in Bouldering: A Prospective Study.” *Wilderness & Environmental Medicine.* 2007;18(4):271-280. doi:10.1580/06-WEME-OR-071R1.1

[2] Zafonte B, Rendulic D, Szabo RM. “Flexor pulley system: anatomy, injury, and management.” *J Hand Surg Am*. 2014 Dec;39(12):2525-32; quiz 2533. doi: 10.1016/j.jhsa.2014.06.005

[3] J. M. Blackstone, “Children and gender-differences in exposure and how anthropometric differences can be incorporated into the design of computer input devices,” *Scandinavian Journal of Work*, Environment & Health, 26 (2007).

[4] MDSave, Hand/Finger Tendon Repair. 2024 https://www.mdsave.com/procedures/hand-finger-tendon-repair/d78bfecc

[5] Mouser Electronics. (n.d.). MF01A__A01 Datasheet [PDF]. Retrieved from https://www.mouser.com/datasheet/2/13/MF01A__c3_a2_c2_96_c2_a1_A01-1915118.pdf

[6] Interlink Electronics. (2010). FSR 402 Datasheet [PDF]. Retrieved from https://cdn.sparkfun.com/assets/8/a/1/2/0/2010-10-26-DataSheet-FSR402-Layout2.pdf

[7] Tekscan, Inc. (n.d.). A301 sensor datasheet [PDF]. Retrieved from https://www.tekscan.com/sites/default/files/FLX-Datasheet-A301-RevI.pdf

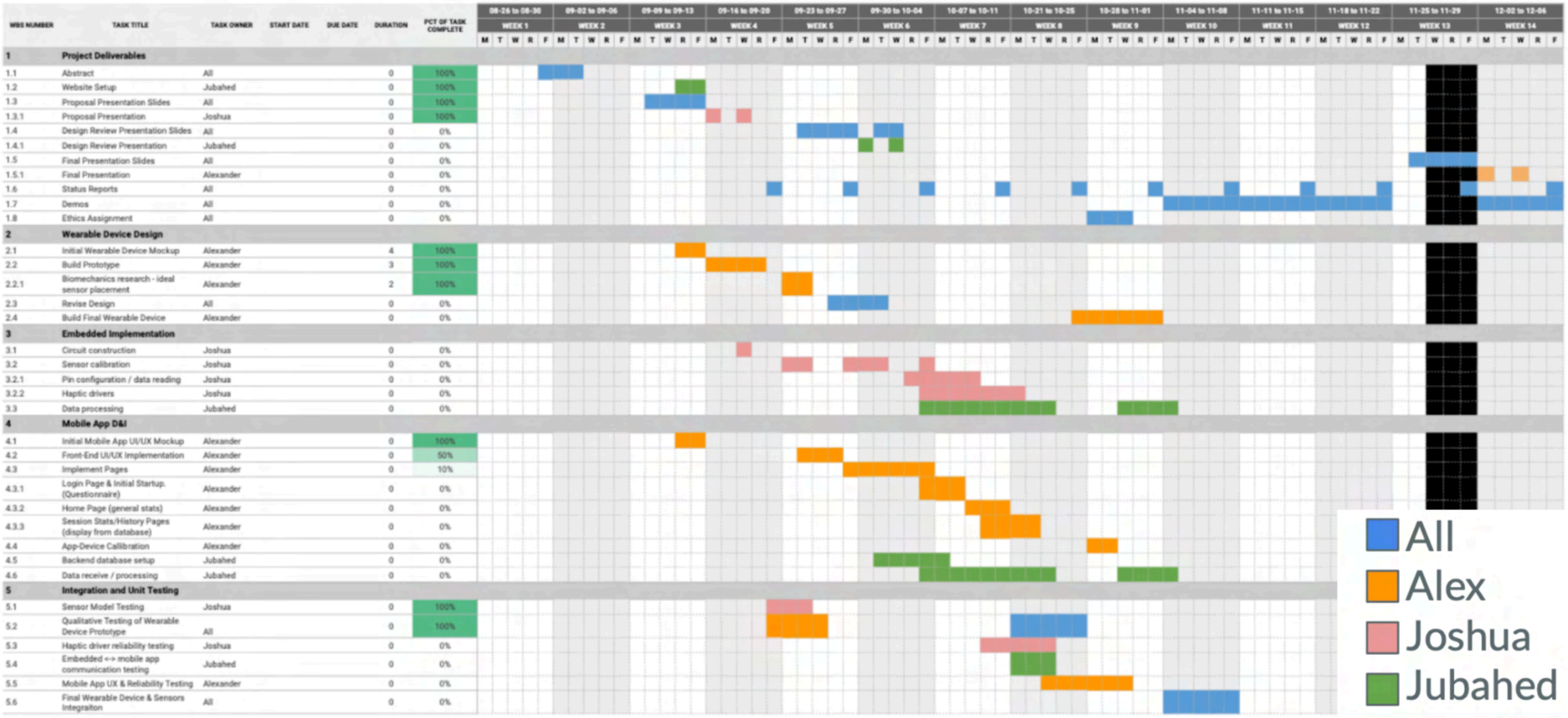
[8] Amphenol ICC. (n.d.). 1020-15-003-001 sensor specification sheet [PDF]. Retrieved from https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/2234/1020-15-003-001_Spec.pdf

[9] Espressif Systems. (n.d.). *ESP32-WROOM-32 datasheet* [PDF]. Retrieved from https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf

[10] NiceClimbs. (n.d.). NiceClimbs Pulley Pal. NiceClimbs. Retrieved [10/12/2024], from https://niceclimbs.com/product/niceclimbs-pulley-pal/

[11] The Climbing Doctor. (n.d.). Pulley protection. The Climbing Doctor. Retrieved [10/12/2024], from https://theclimbingdoctor.com/pulleyprotection/

[12] The Climbing Doctor. (n.d.). Rock rehab protocol: Pulley sprain. The Climbing Doctor. Retrieved [10/12/2024], from https://theclimbingdoctor.com/product/rock-rehab-protocol-pulley-sprain-3/



WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	**Project Deliverables**					
1.1	Abstract	All			0	100%
1.2	Website Setup	Jubahed			0	100%
1.3	Proposal Presentation Slides	All			0	100%
1.3.1	Proposal Presentation	Joshua			0	100%
1.4	Design Review Presentation Slides	All			0	0%
1.4.1	Design Review Presentation	Jubahed			0	0%
1.5	Final Presentation Slides	All			0	0%
1.5.1	Final Presentation	Alexander			0	0%
1.6	Status Reports	All			0	0%
1.7	Demos	All			0	0%
1.8	Ethics Assignment	All			0	0%
2	**Wearable Device Design**					
2.1	Initial Wearable Device Mockup	Alexander			4	100%
2.2	Build Prototype	Alexander			3	100%
2.2.1	Biomechanics research - ideal sensor placement	Alexander			2	100%
2.3	Revise Design	All			0	0%
2.4	Build Final Wearable Device	Alexander			0	0%
3	**Embedded Implementation**					
3.1	Circuit construction	Joshua			0	0%
3.2	Sensor calibration	Joshua			0	0%
3.2.1	Pin configuration / data reading	Joshua			0	0%
3.2.2	Haptic drivers	Joshua			0	0%
3.3	Data processing	Jubahed			0	0%
4	**Mobile App D&I**					
4.1	Initial Mobile App UI/UX Mockup	Alexander			0	100%
4.2	Front-End UI/UX Implementation	Alexander			0	50%
4.3	Implement Pages	Alexander			0	10%
4.3.1	Login Page & Initial Startup. (Questionnaire)	Alexander			0	0%
4.3.2	Home Page (general stats)	Alexander			0	0%
4.3.3	Session Stats/History Pages (display from database)	Alexander			0	0%
4.4	App-Device Callibration	Alexander			0	0%
4.5	Backend database setup	Jubahed			0	0%
4.6	Data receive / processing	Jubahed			0	0%
5	**Integration and Unit Testing**					
5.1	Sensor Model Testing	Joshua			0	100%
5.2	Qualitative Testing of Wearable Device Prototype	All			0	100%
5.3	Haptic driver reliability testing	Joshua			0	0%
5.4	Embedded <-> mobile app communication testing	Jubahed			0	0%
5.5	Mobile App UX & Reliability Testing	Alexander			0	0%
5.6	Final Wearable Device & Sensors Integraiton	All			0	0%

Table III. GANTT Chart (Project Schedule)

Model	Bill of Parts and Budget			
	Quantity	**Cost (total)**	**Manufacturer**	**Description**
FlexiForce A301 Sensor	10	$108.93	Tekscan	Piezoresistor force sensor
DC Motor Vibration, ERM 3VDC [316040001]	2	$2.40	Seeed Technology Co.	Haptic motor (used for alarm)
ESP-WROOM 32	3	$6.67	HiLetGo	Microcontroller
MCP6004-I/ST	4	$2.16	Microchip Technology	Operational Amplifier
PCB Fabrication	1	$50	Undetermined	Printed Circuit Board
3.3V Coin Battery	1	$3.50	Panasonic	Power source
Power-jack	1	$0.89	Digikey	Power connector
Total	22	$174.45		

Table IV. Bill of Parts and Budget