Tactile Time Logging System

Authors: Alexander B Decker, Ali Mohamed Alhabshi, Hudson Mazza, Ryan Jeffrey

Date Created: 10/22/2021
Last Updated: 05/06/2021

Table of Contents

1. Overview

1.1. Executive Summary

Our group is going to create a running band similar to a Fitbit and an Apple Watch that can track various statistics while the user (everyday runners) run. The difference between our tracking band compared to the competition is that ours will have an application that will allow users to connect with one another. Furthermore, It will have a companion app that will display those stats in various ways (graphs, tables, ect.). The system will use a WIFI/Bluetooth Module microcontroller to transmit data wirelessly to the companion app.

The team is working collectively to find applicable ideas on what components are going to be used for the project. The running band will include a heart rate sensor that monitors the user's heart rate and sends it to the user's connected device. A good sensor that is compatible with our microcontroller is going to be considered. We will also use an accelerometer and the user's height to calculate how many steps and how far they have traveled in a given time frame. The app will display the user's information such as their heart rate and accelerometer data. The user will also be able to connect to others through the app and see their data as well. The device will output its battery charge level to a small LED light.

1.2. Team Communication Protocols and Standards

1.2.1 Team Members and their contacts

Team Members	**Contact Information**
Alexander Decker - Hardware	deckeral@oregonstate.edu
Ali Mohamed Alkabshi - Hardware	alhabsha@oregonstate.edu
Hudson Mazza - Software	mazzah@oregonstate.edu
Ryan Jeffrey - Software	jeffrrya@oregonstate.edu

Table 1: Members' Contacts

1.2.2 Communication With Project Partner

Our project is 'Project Partnerless'. Due to this, our mock project partner is Don Heer. Through consultation with him and him reaching out to the team, it has been decided that the primary form of communication between Don and the team will be via email. However, it has also been expressed that if needed we can meet in person and he is willing to clarify questions or concerns in his office hours as well.

1.2.3 Standards and Protocols

Team Standards	**Communication Standards**
• Team members are required to be up to date with what is going on within the project. • Team members are to respect each other's opinions including cultural beliefs. • All team members are expected to communicate during the process of building the project. • Team members are encouraged to ask for help whenever it is needed. • Be accommodating for group members' lives outside of the project's.	• One Team member is assigned to communicate with the project partner via email every week. • Team members are to use Discord Platform to communicate regularly. • The Team is set to meet every Monday at 2:30 pm. • Update project partner with what has been done and is going to be done every week. • At least 2 meetings are to be held every week to have updates on the project's process and also to complete tasks assigned. • Use the Click-Up website to keep track of tasks.

Table 2: Standards and Protocols

1.3. Gap Analysis

Our device will differ from competitors like fitbit in that it can amalgamate the data of multiple devices that connect to a single phone. Unlike other devices like Apple watches that send their data back to the manufacturer (like Apple), our device sends data only to where the user wants it to go so the user can use it for their own purposes. Our approach has advantages: for example, a coach could see how every member of the team is performing simultaneously. However, our device also allows for players to see their own data on the fly such as their heart rate.

Our device would be useful to anyone wanting to go on a fitness journey or trying to lose weight in a way that is scientifically augmented. A scientific approach to losing weight is an especially effective one since all you would need to lose weight is to lose more calories than you gain. However, our device is most useful when used to analyze data from a group of people like a team, or even a couple of people in competition.

The ability to analyze a team's performance has become a major issue in the last couple of decades as big-budget sports teams aim to outcompete their competition. This scientific analysis of a team is called performance analysis and it would be greatly aided by our device which would give coaches a lot of on the fly information that would give them immediate results. They could instantly tell if a team member is underperforming for example.

1.4. Timeline/Proposed Timeline

1.4.1.1. Fall Term:

- Hardware and Software frameworks are chosen.
- First prototype (non-PCB)

1.4.1.2. Winter Term:

- Flutter Experimenting
- Sensors design testing
- All hardware testing
- Blocks Validation
- Second Prototype

1.4.1.3. Spring Term:

- Hardware Finalization
- Software Finalization
- Building final product
- Final system verification
- Showcase

PROJECT	ASSIGNED TO	PROGRESS	START	DAYS	END	BUDGET
Tactile Time logging						
Electrical Development						
Researching compents	Alex	100%	11/2/2021		12/18/2021	
First prototype(non Final PCB)	Alex and Ali	100%	11/8/2021		12/19/2021	
Prelimenary sensor testing	Ali	100%	1/4/2022		2/10/2022	
Mechanical Development						
Enclosure design research	Ali	100%	11/11/2021		1/30/2022	
Software Development						
Flutter Experimenting	Hudson and Ryan	100%	11/3/2021		1/3/2022	
App	Ryan	100%	1/13/2022		4/25/2022	
Bluetooth data receiving	Ryan	100%	1/13/2022		2/28/2022	
User Interface	Hudson	100%	1/14/2022		4/30/2022	
Software Finalization	Hudson and Ryan	100%	3/1/2022		4/25/2022	
Researching Software Framework	Hudson and Ryan	100%	10/24/2021		12/5/2021	

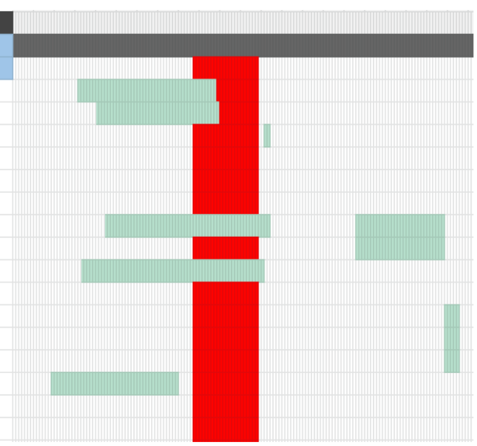
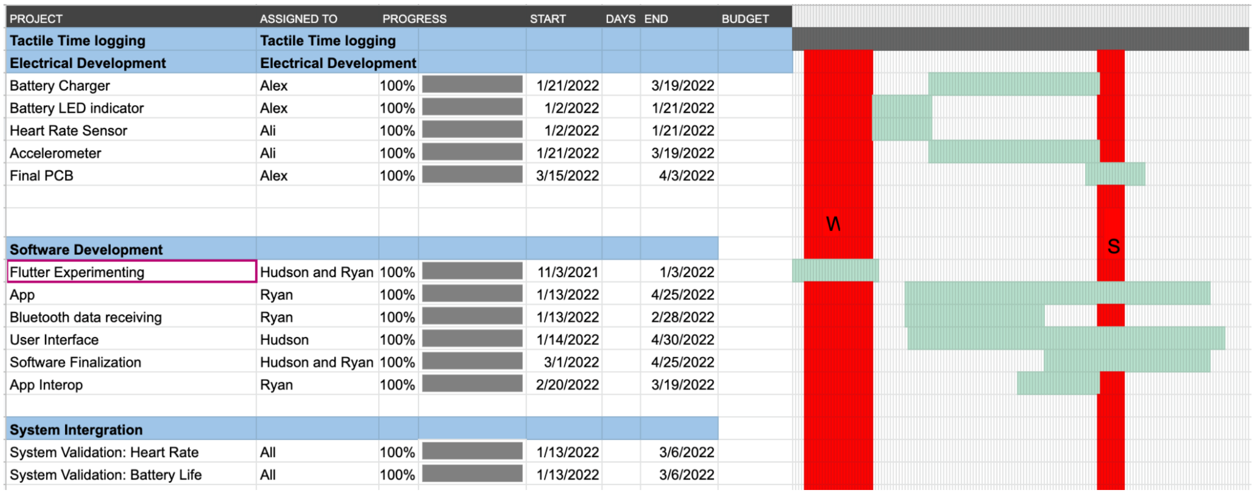
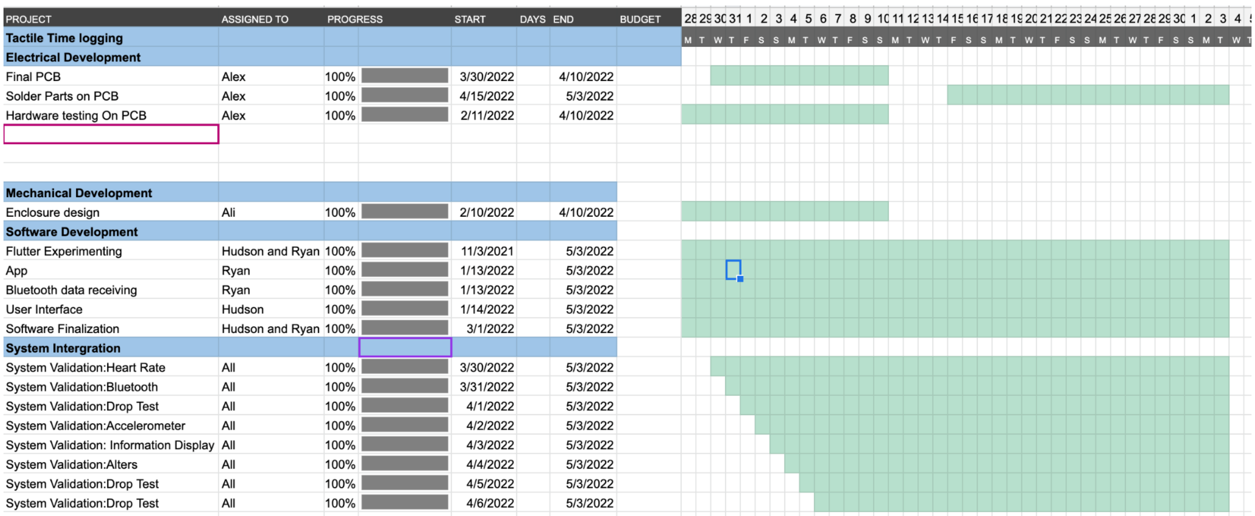


Figure 1: Fall timeline



PROJECT	ASSIGNED TO	PROGRESS	START	DAYS	END	BUDGET
Tactile Time logging	**Tactile Time logging**					
Electrical Development	**Electrical Development**					
Battery Charger	Alex	100%	1/21/2022		3/19/2022	
Battery LED indicator	Alex	100%	1/2/2022		1/21/2022	
Heart Rate Sensor	Ali	100%	1/2/2022		1/21/2022	
Accelerometer	Ali	100%	1/21/2022		3/19/2022	
Final PCB	Alex	100%	3/15/2022		4/3/2022	
Software Development						
Flutter Experimenting	Hudson and Ryan	100%	11/3/2021		1/3/2022	
App	Ryan	100%	1/13/2022		4/25/2022	
Bluetooth data receiving	Ryan	100%	1/13/2022		2/28/2022	
User Interface	Hudson	100%	1/14/2022		4/30/2022	
Software Finalization	Hudson and Ryan	100%	3/1/2022		4/25/2022	
App Interop	Ryan	100%	2/20/2022		3/19/2022	
System Intergration						
System Validation: Heart Rate	All	100%	1/13/2022		3/6/2022	
System Validation: Battery Life	All	100%	1/13/2022		3/6/2022	

Figure 1: Winter timeline



PROJECT	ASSIGNED TO	PROGRESS	START	DAYS	END	BUDGET
Tactile Time logging						
Electrical Development						
Final PCB	Alex	100%	3/30/2022		4/10/2022	
Solder Parts on PCB	Alex	100%	4/15/2022		5/3/2022	
Hardware testing On PCB	Alex	100%	2/11/2022		4/10/2022	
Mechanical Development						
Enclosure design	Ali	100%	2/10/2022		4/10/2022	
Software Development						
Flutter Experimenting	Hudson and Ryan	100%	11/3/2021		5/3/2022	
App	Ryan	100%	1/13/2022		5/3/2022	
Bluetooth data receiving	Ryan	100%	1/13/2022		5/3/2022	
User Interface	Hudson	100%	1/14/2022		5/3/2022	
Software Finalization	Hudson and Ryan	100%	3/1/2022		5/3/2022	
System Intergration						
System Validation:Heart Rate	All	100%	3/30/2022		5/3/2022	
System Validation:Bluetooth	All	100%	3/31/2022		5/3/2022	
System Validation:Drop Test	All	100%	4/1/2022		5/3/2022	
System Validation:Accelerometer	All	100%	4/2/2022		5/3/2022	
System Validation: Information Display	All	100%	4/3/2022		5/3/2022	
System Validation:Alters	All	100%	4/4/2022		5/3/2022	
System Validation:Drop Test	All	100%	4/5/2022		5/3/2022	
System Validation:Drop Test	All	100%	4/6/2022		5/3/2022	

Figure 1: Spring timeline

1.5. References and File Links

1.5.1.1. The current WiFi Module that we are looking at is https://components101.com/sites/default/files/2021-09/ESP12E-Datasheet.pdf

1.5.1.2. https://eecs.oregonstate.edu/capstone/submission/pages/viewSingleProject.php?id=6aXqqcknsUNrayHH

1.5.1.3. Potential heart rate monitor MAX30102 Datasheet | Maxim Integrated

1.5.1.4. Gantt Chart Link

https://docs.google.com/spreadsheets/d/1PXux20Vjh_WzalE1d6OQr0vCXa5mjiwbCaJiESVoWdM/edit?usp=sharing

1.6. Revision Table

For our revision table, we plan on having a table like the one shown below to with a list of the weeks and then A list of the bullets points of what has changed since the last week

Date	Who	Section	Changes
10/22/2021	Ali	1.4	Made Timeline
10/22/2021	Alexander	1.2	Wrote Team Expectations
10/22/2021	Alexander	1.5	Added References
10/22/2021	Hudson	1.3	Wrote Gap Analysis
10/22/2021	Ryan	1.1	Wrote Summary
10/29/2021	Ali	1.4	Improved Timeline Section and added more detail
11/12/2021	Ali	1.1	Rewrote Executive Summary
11/12/2021	Ali	1.2	Added More Detail to Communication Standards
11/12/2021	Ali	1.4	Replaced Timeline Figure
11/12/2021	Ryan	1.3	Rewrote Gap Analisis
11/19/2021	Hudson	1.2	Added Communication with Project Partner Section
12/2/2021	Ali	1.2	Updated format of team contacts table
12/2/2021	Hudson	1.6	Updated Revision Table Format
12/2/2021	Ryan	1.1	Update Executive Summary
12/2/2021	Alex	1.4	Replaced Timeline Figure
12/3/2021	All	All	Reformatted and Cleaned Up Entire Section
12/3/2021	Alex	1.4	Changed programs for Gantt Chart to google sheets.
12/3/2021	Ali	1.4	Reformatted timeline info

Table 3: Revision table section 1

2. Requirements Impacts and Risks

2.1. Requirements

This section includes the requirements that have been chosen. This team has no project partner that expects a specific outcome from the project, so the team has collaboratively chosen these requirements. The team expects to meet all requirements in order to have completed the project successfully. Below are the requirements for the tactile time logging system project:

#	Requirement	Progress	Software or hardware
R1	User information is displayed in the App	Finished	Software
R2	The system will report user steps and distance traveled in meters.	finished	Software
R3	The system will operate normally for 2 hours on a full battery.	Finished	Hardware
R4	The system must be able to transmit data collected wirelessly to a phone in order to receive and organize data to be viewed by the user	Finished	Software
R5	The system will collect the user's heart rate.	Finished	Hardware
R6	The user is alerted for events they set.	Finished	Software
R7	User teams where users can view others' data share.	Finished	Software
R8	The system will be durable to Halt and Pass Product drop test standards	finished	Hardware

Table 4: Requirements

2.1.1. Verification Method

2.1.1.1. User information is displayed in the App:

2.1.2. Open app.

2.1.3. Log in to pre-existing account.

2.1.4. Show that user metadata and run data is displayed.

2.1.5. Show that the account's data is unique by showing off multiple accounts.

2.1.5.1. The system will report user steps and distance traveled meters:

2.1.6. Turn system on.

2.1.7. Take two steps.

2.1.8. Check the app to see the system transmit two steps.
2.1.9. Stop walking.
2.1.10. Show that the system has stopped sending steps to the app.

2.1.10.1. The system will operate normally for 2 hours on a full battery:
2.1.11. Remove the batteries from the system.
2.1.12. Attach a DC power supply to the system.
2.1.13. Set the DC power supply to 3.7V
2.1.14. Observe the draw in Amperes of the system.
2.1.15. Calculate the total draw by using $\frac{Ah}{A}$ To find the total amount of hours the battery will last. To find the exact number of minutes multiply by 60.

2.1.15.1. The system must be able to transmit data collected wirelessly to a phone in order to receive and organize data to be viewed by the user:
2.1.16. Power system with batteries
2.1.17. Open phone application
2.1.18. Connect to bt05 through bluetooth
2.1.19. Observe data being received in application
2.1.20. Compare to Serial monitor to verify that

2.1.20.1. The system will report the user's heart rate:
2.1.21. Turn system on
2.1.22. Make sure the sensor is working properly and illuminating light
2.1.23. Place heart rate monitor on your wrist
2.1.24. Observe the readings in the serial monitor or on the phone application
2.1.25. Compare readings to average resting heart rate of humans and then measure the pulse for 30 seconds to find heart rate. Compare

2.1.25.1. The user is alerted for events they set:
2.1.26. Create an event that triggers once per second.
2.1.27. Show that the event triggers once per second.
2.1.28. Show that the user can toggle between events and no events.

2.1.28.1. User teams where users can view others' data share:
2.1.29. Create two accounts that are part of a team.
2.1.30. Show that the app displays the information of these two accounts in real time.
2.1.31. Create a third account.
2.1.32. Show that the system does not show information about the new account since it is not part of the group.

2.1.32.1. Create a durable shell:
2.1.33. Drop the system from a height of three feet.
2.1.34. Show that the system still transmits data to the app.

2.2. Design Impact Statement

1.0 Introduction:

Our group's initial project was to create a time logging system that had a physical component to initialize time instances or to gather information. After coming up with many ideas, the team decided to create a wearable exercise tracker. This product will track the user's heart rate, distance, and steps as they run. Alongside this, it will wirelessly communicate with the user's phone and log this information in a server connected with a companion app. With this comes plenty of positive impacts, like promoting better health and easier tracking of exercise. However, devices like the ones our project is trying to replicate can lead to possible issues as well. Some of these issues include populations getting addicted to running. This societal shift could lead to more running related injuries. Furthermore, wearable tech has been shown to lead to more e-waste and be harder to recycle when broken or replaced. Lastly, the typical high cost of products like these and low repairability can affect lower income areas that are statistically more affected by weight related issues that this product is trying to help fight against.

2.0 Cultural and Social Impacts:

For the cultural and social impacts, our exercise band could bring people into new social circles that they had no idea about. For some people, exercise introduces them to new people and they gain a long lasting friendship because of the bond that is formed working together. Currently, in the United States, there seems to be a revival of bodybuilding and exercise with social media like TikTok and Instagram [9]. This type of culture is in great contrast to what is currently in younger generations because most young people are very entitled and fragile. By changing the culture you change society. Another impact would be the benefits of exercise and the chemicals it releases. Exercise releases endorphins that interact with receptors in your brain causing a sense of joy. This is very useful against depression or other mental health problems[10].

However, an issue that could arise from this wristband that tracks your physical activity could be that people can get addicted to exercise, or further other severe issues like eating disorders. As a society, we are already obsessed with our physique. Adding an exercise tracer like this, no matter how beneficial it could be, could feed into those controlling vices. According to the National Federation of Professional Trainers, over fifty percent (50%) of people who have anorexia use a fitness tracker application to count calories. Furthermore, forty eight percent (48%) have ‘cross addictions’ between eating disorders and exercise addiction. [1]

This correlation of nearly 50% shows that wearable technology could lead to serious issues. Not only could lead to issues, it could be existing issues in users. To curb this issue, our team will establish a platform that encourages healthy workout habits. Furthermore we will not include any sort of calorie counting in our program. Including these features is the exact reason how people abuse apps for negative reasons. Our team wants our app location and wristband to be used for good, healthy habits. Including these features could encourage people to use these for negative habits. Our team wants our application and wristband to be used for good, healthy habits. Including features like calorie counting could encourage people to use these for negative habits.

3.0 Public Health, Safety, and Welfare Impacts:

One positive impact of this product is that people could be encouraged to run, or be more active in general, more. Some studies show that using a device like a pedometer increases people's motivation to work out and be more active [2]. Furthermore, it enabled these people to quantify progress, further motivating them. Lastly, it allowed people an avenue to internalize these motivations, removing the need for a wearable device to motivate the user to be more active. This would be a positive impact our wristband could have. It would enable people to get active and lose weight when otherwise that might not have been inclined to do so is the main goal of this project. Meeting this goal would be huge for the team.

However, with every positive impact, there is a negative one as well. In the case of meeting our goal to increase the users running frequency, this opens the door to more possible injuries. lead to more injuries. Studies show that sudden and dramatic changes in running patterns lead to a dramatic increase in injuries [5]. In a case where our product could lead to running addiction, people could gain an obsession with running. In the aforementioned study, it shows that people who are obsessive runners are thirty six percent (36%) to receive a running related injury. These injuries can range from something minor like blisters or light cases of shin splints to something more severe like chronic stress fractures.

This issue brings into question how to maximize motivation while mitigating the desire to run more. This can be done in a few different ways. One way is to incentivize users to take rest days. Resting is a key component in any running regiment and it is not something that the team has seen in any other similar products. Doing this would build a healthy relationship with taking rest that could minimize obsessive running. This could be done by giving in app rewards of some kind. While this uses positive reinforcement, negative reinforcement could be done too. While there is only so much that can be done, the team could implement a system where users are notified of too frequent runs. Lastly, a more neutral approach could be to provide a native running program that has suggested distances and rest periods for given goals. Doing any of these could prove to be effective to maximize running motivation and minimize over exertion and obsessive running.

4.0 Environmental Impacts:

For the environmental impacts, there are more negative impacts compared to positive impacts. However, this does not mean that there are no positive impacts. One major positive impact on the environment would be that people would be able to change their health and adjust their habits in life. Our product is designed to encourage people to get up and be active. If people were to embrace the fitness industry we are trying to cultivate, it could lead to them to do activities such as biking to work or running to work. This would lower the carbon footprint of individuals. This mindset applied to a wide audience could greatly impact the environment for the better

Now, with that one positive impact stated, it is time to look at the negatives. As with any consumer product there will inevitably be some environmental impact since the parts to make this product will have to be created, shipped, and then assembled. Another environmental impact is electronic waste (e-waste). According to a 2007 paper by Balakrishnan Babu [6] IT equipment produced 357 tonnes of e-waste in the year 2000, along with “Electric and electronic tools" producing another 28 tonnes. As our device will inevitably break, some of our devices will become part of this statistic. To help reduce the e-waste problem the team decided to use Lithium Ion (Li-Ion) batteries since they are more efficient than regular batteries and reduce battery e-waste by being reusable [7]. The team has also decided to use generic parts which removes the need for very specialized tooling and makes the device more modular.

It is also important to consider the effects of Electromagnetic Field (EMF) Radiation. Though a paper published by the NCBI stated that the effects of EMF radiation on humans and the environment are unclear [8] that does not mean we should write this problem off completely. Though the group has decided to use relatively low power and low radiation solutions like Bluetooth the device still needs to connect with the user's phone or computer, and if the user is part of a group the app will be continuously sending data to its peers. Creating an efficient data transfer method will be crucial to ensuring this device does not produce too much EMF radiation.

One last thing to consider is the plastic enclosure of the Tactile Timing system. The enclosure will be 3D printed and will be made of plastic. The first problem with a 3D printer is the energy consumption of the printer. 3D printers consume 50 to 100 times more electrical energy than an injection molding of plastic. On top of that 3D printers also produce unhealthy emissions. PLA filament emitted about 20 billion ultrafine particles per minute and ABS can emit up to 200 billion particles per minute[11].

5.0 Economic Factors:

One issue the team foresees economically is the cost. We want these products to be affordable. The issue in this lies that the industry standard for these types of products tend to be over 60 dollars. For example, the Fitbit Flex, a fairly standard model for Fitbit users, costs just shy of one hundred US dollars. This then begs the question of how much the production per unit costs. After disassembly and looking at all the parts, the evaluated cost of the wristband is just over seventeen US dollars [3]. With this major cushion between retail cost and cost of production, the team could theoretically make just as good a quality of a product at a lower cost and take lower net profits.

However, unless we lower the retail price of the product dramatically, cost could still be an issue. The main goal of our product is to help the population be more active and lose weight. However, if the product is too expensive, we could miss the demographic that is disproportionately affected by obesity rates. In the US (our main target demographic at the moment) low income families face a dramatically higher rate of obesity when compared to people with more wealth. Furthermore, poverty is shown to have a correlation with activity levels, with more wealthy individuals on average living less sedentary lifestyles [4]. If we do not ensure that this demographic can afford our device, then not only are we missing a huge target demographic, but we are compromising the goal of our product.

The solution to this problem is to try to cut costs where possible while maintaining the integrity of the product. Making an affordable product is attainable, but could be done by going towards cheaper parts. These cheaper parts could lead to the product breaking earlier and then having to be replaced. Not only would this further the issue stated before in the Environmental Impacts section, but it would also affect low income families due to the repetitive buying of the product. This is why we as a product development need to prioritize having quality parts for our product while also maintaining a cost friendly and cost effective product.

6.0 Conclusion:

While this product could do a lot of good, the team has to recognize that it can also do a lot of harm. It could encourage users to inadvertently run themselves to chronic issues while also potentially leading to a societal shift that encourages addiction to exercise. Alongside these issues, it would add to the vast amount of e-waste generated a year to the different wearable technologies and could inadvertently widen the gap for people in low income communities and put them at a disadvantage. While we believe that, like other products similar to ours, the positives far outweigh the negatives, mitigation efforts must be put into place to minimize and eliminate these issues at the source. Our team's mitigation efforts are to exclude public statistic sharing and calorie counting features, provide users with a positive relationship with the rest and recovery, and build our product with cost effective materials to increase the likelihood that highly affected low income communities can buy the products to increase their physical activity. While difficult to do correctly, our group is committed to creating this product correctly in an ethical manner. Well difficult to do correctly, our group is committed to creating this product correctly in an ethical manner.

References:

[1] C. Kronemer, "Fitness trackers and 'techorexia,'" *National Federation of Professional Trainers*, 01-Dec-2020. [Online]. Available: https://www.nfpt.com/blog/fitness-trackers-techorexia . [Accessed: 06-Dec-2021].

[2] C. Donnachie, S. Wyke, N. Mutrie, and K. Hunt, "Utilizing self-determination theory to understand men's experiences of using pedometers to increase physical activity," International Journal of Behavioral Nutrition and Physical Activity, 05-May-2017. [Online]. Available: https://ijbnpa.biomedcentral.com/articles/10.1186/s12966-017-0505-z . [Accessed: 06-Dec-2021].

[3] I. H. S. E. & Media, "Teardown: Fitbit Flex," *Electronics360*, 17-Oct-2013. [Online]. Available: https://electronics360.globalspec.com/article/3128/teardown-fitbit-flex . [Accessed: 06-Dec-2021].

[4] J. A. Levine, "Poverty and obesity in the U.S.," *Diabetes*, 01-Nov-2011. [Online]. Available: https://diabetes.diabetesjournals.org/content/60/11/2667 . [Accessed: 06-Dec-2021].

[5] J. de Jonge, Y. A. Balk, and T. W. Taris, "Mental recovery and running-related injuries in recreational runners: The moderating role of passion for running," *MDPI*, 06-Feb-2020. [Online]. Available: https://www.mdpi.com/1660-4601/17/3/1044/htm . [Accessed: 05-Dec-2021].

[6] B. R. Babu, A. K. Parande, and C. A. Basha, "Electrical and electronic waste: A global environmental problem," International Solid Waste Association, Aug. 1, 2007, p. 308. [Online]. Available: https://krc.cecri.res.in/ro_2007/141-2007.pdf [Accessed: 12/04/2021].

[7] "Used lithium-ion batteries," Environmental Protection Agency. (Aug. 11, 2021), [Online]. Available: https://www.epa.gov/recycle/used-lithium-ion-batteries [Accessed: 10/29/2021].

[8] G. Redlarski, B. Lewczuk, A. Żak, A. Koncicki, M. Krawczuk, J. Piechocki, K. Jakubiuk, P. Tojza, J. Jaworski, D. Ambroziak, Ł. Skarbek, and D. Gradolewski, "The influence of electromagnetic pollution on living organisms: Historical trends and forecasting changes," BioMed Research International, vol. 2015, Feb. 5, 2015. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4355556/ [Accessed: 10/29/2021].

[9]Robertson, Lacey. "The rise of the Fitness culture." The Seraphim, Ndpnews, 14 May 2019, ndpnews.org/focus/2019/05/14/the-rise-of-the-fitness-culture/ [Accessed: 11/20/2021]

[10]Fulghum, Debra. "Exercise and Depression." Guide, Web MD, 18 Feb. 2020, www.webmd.com/depression/guide/exercise-depression [Accessed: 11/20/2021]

[11]Gilpin, Lyndsey. "The dark side of 3D printing: 10 things to watch." TechRepublix, TechRepublix, 5 Mar. 2014, www.techrepublic.com/article/the-dark-side-of-3d-printing-10-things-to-watch/. [Accessed: 11/13/2021]

2.3. Risks

Risk ID	Potential Risks	Risk Category	Risk Proba-bility	Risk Impact	Performance Indicator	Responsible Party	Action Plan
R1	Chip Shortage/Ship-ping delays	Timeline	H	H	Parts not in ontime	Group	Avoid: Order Chips Sooner
R2	Waterlogged Components	Hardware	M	M	Components waterlogged from rain/mud	Ali A. and Alex D.	Reduce: Test the watch casing before inserting the components
R3	Issues with learning new software	Timeline	M	L	Software is taking longer than expected to develop	Hudson M. and Ryan J.	Reduce: Use Familiar Software frameworks where possible
R4	Long Term Durability of Project	Hardware	M	L	Project deteriorates over development	Ali A. and Alex D.	Avoid: Use Quality Parts on initial build
R5	User data is stolen/leaked	Security	M	H	Security Information on users is being published online or taken advantage of my malicious hackers	Hudson M. and Ryan J.	Avoid: Practice Good Security Coding

Table 5: Risks

2.4. References and Files Links

2.4.1.1. Potential Heart Rate monitor https://discordapp.com/channels/895732673841094656/901235305372319826/903460460198781000

2.4.1.2. Potential Gyroscope https://discordapp.com/channels/895732673841094656/901235305372319826/903458482655739914

2.4.1.3. Potential software https://flutter.dev/

2.4.1.4. Potential PCB design software https://www.autodesk.com/products/eagle/free-download

2.4.1.5. Potential PCB design software https://www.kicad.org/

2.4.1.6. Potential Wifi Module for wireless data transfer https://www.amazon.com/HiLetgo-Internet-Development-Wireless-Micropython/dp/B010O1G1ES?th=1

2.4.1.7. Potential PCB manufacture https://jlcpcb.com/DEA

2.4.1.8. Helpful Software for uploading Gerber Files https://oshpark.com/

2.4.1.9. Parts distributor https://www.digikey.com/

2.4.1.10. ISTA drop test standards https://www.safeloadtesting.com/what-is-ista-drop-test-and-how-to-comply-it/

2.4.1.11. Halt and Hass Product Drop testing https://www.halthass.co.nz/reliability-services/environmental-testing-category/product-drop-testing/

2.5. Revision Table

Date	Who	Section	Changes
10/22/2021	Ali	2.1	Created the Requirements
10/22/2021	Alexander	2.2	Filled in references table
10/22/2021	Ryan	2.5	Filled in Risked
10/22/2021	Hudson	2.4	Filled in risks
11/12/2021	Alexander	2.1	Converted requirements to a table
11/12/2021	Hudson	2.2	Reformatted risks
11/19/2021	Hudson	2.3	Added column of risk probability and risk impact.
12/2/2021	Alexander	2.1	Added verification methods
12/2/2021	Hudson	2.6	Updated Revision Table Format
12/2/2021	Ali	2.1	Updated requirements

Table 6: Revision table section 2

3. Top-Level Architecture

3.1. Block Diagram

3.1.1. Black Box Diagram

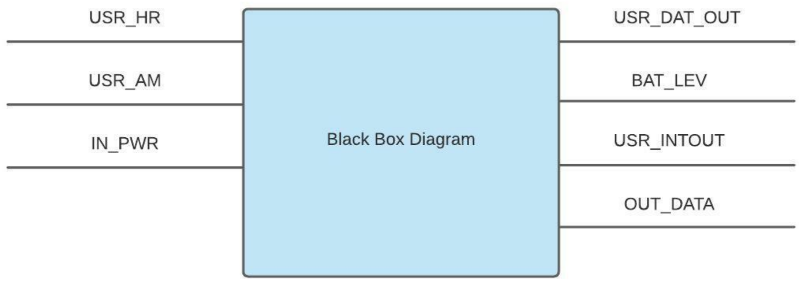


Figure 3: Black Box Diagram

3.1.2. Top-Level Block Diagram

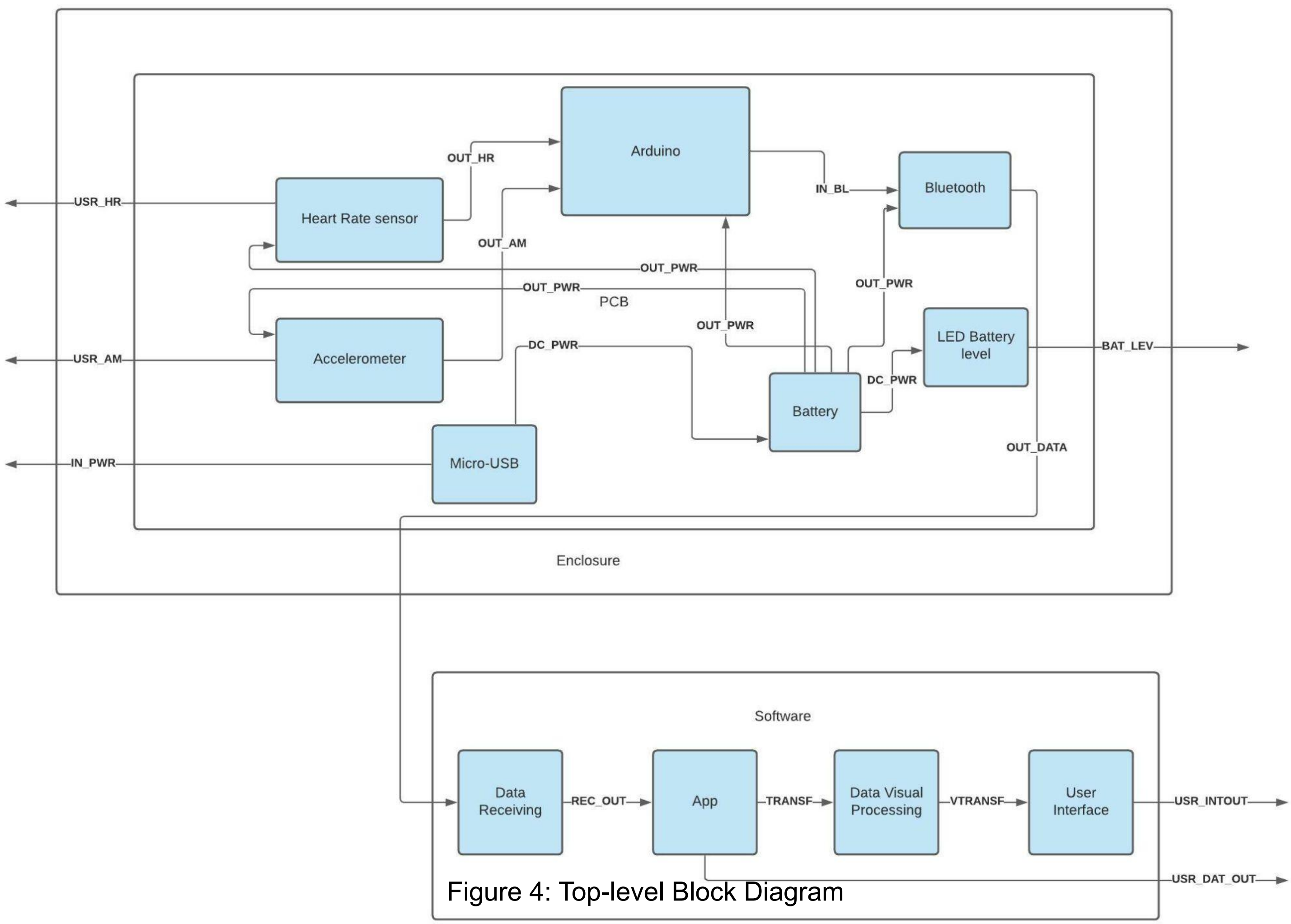


Figure 4: Top-level Block Diagram

3.2. Block Descriptions

3.2.1. App

Champion: Ryan Jeffery

This block receives data from the Receiver block and performs statistical analyses such as detecting changes in the user's heart rate, detecting changes in the user's running speed, etc..

3.2.2. User Interface

Champion: Hudson Mazza

The User Interface block converts the data given to it from the App block and converts it to a format the human user can understand. The final format will be charts, simple information like BPM and steps taken, and an interface for viewing the data from multiple devices at once.

3.2.3. Data Visual Processing

Champion: Hudson Mazza

Once the app has analyzed the data collected from the user this block will transform the analysis into a visual format (such as a graph or a table) that the user can then read on their phone.

3.2.4. Data Receiving

Champion: Ryan Jeffrey

Once the user's information has been collected from the device it will be wirelessly transmitted to the user's phone, this block receives this data for use in the app.

3.2.5. Heart Rate Sensor

Champion: Ali Alhabshi

The Heart rate sensor will collect data from the user's wrist using an LED that senses when blood is flowing through a vein. This data will be sent to the Arduino PCB to find the user's beats per minute.

3.2.6. Accelerometer

Champion: Ali Alhabshi

The accelerometer will collect data generally from the user's performance to measure the acceleration or the motion of the user. Data collected from the accelerometer will be sent to the Arduino PCB to track the user's steps.

3.2.7. Battery

Champion: Alex Decker

The battery will supply power to all the components inside the enclosure like Arduino PCB. It will receive charging power from the Micro-USB on the PCB.

3.2.8. LED Battery Level

Champion: Alex Decker

An LED on the enclosure will change color depending on the battery level. For example, the LED will be green from 100% battery - 60% battery, yellow from 59% battery - 26% battery, and red from 25% battery - 0% battery.

3.2.9. Micro USB

Champion: Alex Decker

The Micro USB will be used to power the battery and receiver power from outside the enclosure.

3.2.10. PCB

Champion: Alex Decker

The PCB will be used to connect all the components on a small PCB that will be optimized for size. It will also allow for a good connection between components

3.2.11. Enclosure

Champion: Ali Alhabshi

The enclosure will be the protective shell for the PCB and all of its components. The enclosure will be specially designed for the PCB to allow for the heart rate to be detected on the user.

3.2.12. Bluetooth

Champion: Ali Alhabshi

The Bluetooth module will take the data that was collected and be sent to the App that will then be organized into and formatted for the user to easily view. The Bluetooth module will be powered by a battery.

3.2.13. Arduino

Champion: Alexander Decker

The Arduino will be the device that connects all of the components inside the PCB and transfers the data. It will also do all of the arithmetic logic for organized data.

3.3. Interface Definitions

Interfaces	Properties	Definitions
OUT_HR	N/A	Output heart rate sensor data
OUT_DATA	N/A	Raw digital data transfer to the app
USR_HR	N/A	User Heart Rate
USR_AM	N/A	User Accelerometer data
IN_PWR	Vmax:5V Vmin:3.3V Ipeak:20A Inominal:1.5A	Power in for charging
OUT_PWR	Vmax:4.2V	Out power to components

	Vmin:0.3V Ipeak:2.6A Inominal: 0.52A	
DC_PWR	Vmax:4.2V Vmin:3.0V Ipeak:1.3A Inominal:0.52A	Direct current power
OUT_AM	N/A	Output of accelerometer
USR_INTOUT	N/A	Output data for user to access
IN_BL	N/A	Inputs data from Arduino to Bluetooth module
REC_OUT	N/A	Output data from the bluetooth receiver.
TRANSF	N/A	Output transformed and analyzed data from the app.
VTRANSF	N/A	Output visual representation (diagrams, graphs) of transformed data.
USR_DAT_OUT	N/A	Output user's transformed data to other app instances.
BAT_LEV	Vmax:5V Vmin:1.5V Ipeak:25mA Inominal:20mA	Output LED that changes color based on battery level

Table 7: Interface definitions

3.4. References and File Link

3.4.1. Block Diagram on Lucid chart
https://lucid.app/lucidchart/4fb482ff-adab-4dc4-977b-272bc2c2b6b5/edit?viewport_loc=-76%2C-63%2C2263%2C1092%2C0_0&invitationId=inv_e26eaf28-5fa8-4443-a78c-654427dd2da2

3.4.2. Battery datasheet

https://www.ineltro.ch/media/downloads/SAAItem/45/45958/36e3e7f3-2049-4adb-a2a7-79c654d92915.pdf

3.4.3. LED Datasheet https://www.lc-led.com/products/b500rgb4h.html

3.5. Revision Table

Date	Who	Section	Changes
11/17/2021	All	3.1	Made Block Diagram
11/17/2021	All	3.2	Made Block Diagram Descriptions
11/17/2021	All	3.3	Made Interface Definitions
12/2/2021	Hudson	3.5	Updated Revision Table Format
12/2/2021	Ali	3.2	Added champions
12/3/2021	Ali	3.1	Updated black box diagram
12/3/2021	Ryan	3.1	Updated block diagram
12/3/2021	Ryan & Ali	3.3	Update module interface definitions

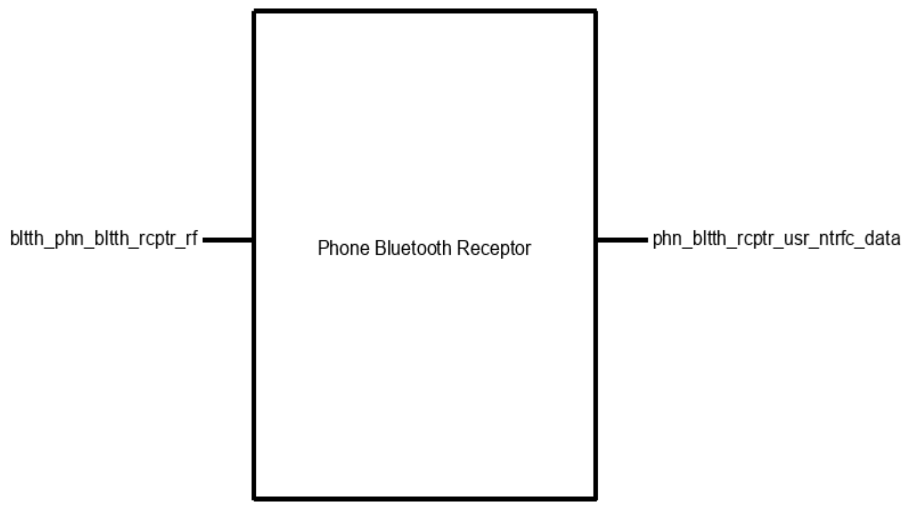
Table 8: Revision table section 3

4. Block Validations

4.1. Phone Bluetooth Receptor (Champion: Ryan Jeffrey)

4.1.1. Description

This block is dedicated to the retrieval of data from the physical device to the user’s phone or computer for use in the companion app. Retrieving the data will be done with the operating system’s native Bluetooth libraries (such as Bluez on Linux, for example). The data will be sent to the companion app as JSON.



4.1.2. Design

Some basic Java/C-like pseudocode:

```
main () {
        initFirebaseConnection () ; // Connect to the database .


        async {
                getBluetooth () ; // Get bluetooth data .
                if( fiveMinutesPassed ) {
                        push_bt_data () ; // Push data gotten from bluetooth .
                        pull_firebase_data () ; // Get data pushed by other apps .
                        display_data () ; // Write the data to the screen .
                }
        }
}
```

4.1.3. General Validation

This block is necessary for the app to be able to permanently save data and to share it with others. Alternatively to Firebase you could have a system where the app has a local sqlite database, and pushes its changes to a remote database at some interval via some networking protocol like XMPP or HTTP. This was our original plan, however this proved to be too complex of a system, Firebase offered much more ease of use. However, using a more complemented setup would have given our users more control over their own data as they would have been able to set up their database rather than rely on its developers to provide one.

Firebase is a mobile and desktop application tool meant for synchronized databases [1]. It works similarly to this block's original design; it was originally a chat messenger system that evolved into a system meant to send application data rather than just text. It has a proprietary API that offers authentication, user sign in, and data management. It is a Google product and is well integrated into the Google ecosystem, providing cloud messaging and analytics [2].

The main cost of this block is the time that it will take from researching Firebase's unique and proprietary API and integrating it into the app.

4.1.4. Interface Validation

Interface Property	**Why is this interface this value?**	**Why do you know that your design details for this block above meet or exceed each property?**

input bltth_phn_rcptr_rf:

Data transfer rate: 1Hz	The app will store unsent data in a buffer and will send in batches. It sends at a low rate to not overwhelm the network.	Data does not need to be sent at a very fast rate as it is not very time sensitive
Info: Queue: Unhandled messages will be queued and resent until handled	Based off of needs of the graphical processing block	Information only
Bidirectional: This interface can take messages from the app	Based off needs of the app to configure this block	Information only

input bltth_phn_rcptr_rf:

Data transfer rate: Once per five minutes.	The app will store unsent data in a buffer and will send in batches. It sends at a low rate to not overwhelm the network.	Data does not need to be sent at a very fast rate as it is not very time sensitive
Bidirectional	Can send and receive data from the server	Information only

Protocol: Firebase	Firebase JSON messages	Information only

4.1.5. Verification Plan

1. Meet with group to verify the interface from device to app, and between different blocks of the app.
2. Write Bluetooth retrieval pseudo code to analyze logic
3. Write Firebase pseudo-code to analyze logic.
4. Integrate into the app.
5. Unit test to discover bugs.
6. Repeat from step 3 onward until all platforms are satisfied.

4.1.6. Revision Table

Date	Description
Jan 7, 2022	Document creation
Jan 8, 2022	Added Revision table, description, JSON, interface, verification plan
Jan 15, 2022	Added more JSON fields and example
Jan 16, 2022	Added pseudocode
Jan 20, 2022	Added Validation tables
Jan 21, 2022	Rewrote general validation, verification plan. Added block diagram
Apr 21, 2022	Rewrote description, psuedo code, validation for new Firebase approach.

4.1.7. References

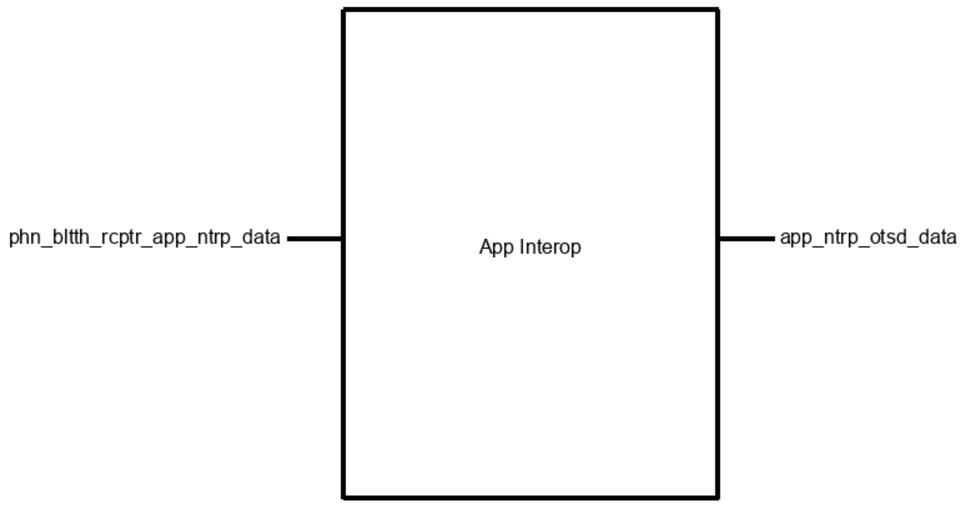
[1] S. Melendez, “Sometimes you’re just one hop from something huge,” Mar. 15, 2020. [Online]. Available: https://www.fastcompany.com/3031109/sometimes -youre -just -one -hop -from - something-huge (visited on 05/27/2014).

[2] J. Tamplin, “Firebase expands to become a unified app platform,” May 18, 2016. [Online]. Available: https://firebase.blog/posts/2016/05/firebase -expands -to -become -unified -app - platform (visited on 04/21/2022).

4.2. App Interop(Champion: Ryan Jeffrey)

4.2.1. Description

This block is dedicated to the publishing and downloading of data from the app to a secure database. The app does not store data itself (except some very recently collected data in memory), it relies on the remote database for permanent storage. The data can be retrieved either from the app that published it to the server or another app that has the permissions to view it. This block will use Firebase, a synchronous database by Google.



4.2.2. Design

Some basic Java/C-like pseudocode.

```
main () {
        initFirebaseConnection () ; // Connect to the database .
        async {
                getBluetooth () ; // Get bluetooth data .
                if( fiveMinutesPassed ) {
                        push_bt_data () ; // Push data gotten from bluetooth .
                        pull_firebase_data () ; // Get data pushed by other apps .
                        display_data () ; // Write the data to the screen .
                }
        }
}
```

4.2.3. General Validation

This block is necessary for the app to be able to permanently save data and to share it with others. Alternatively to Firebase you could have a system where the app has a local sqlite database, and pushes its changes to a remote database at some interval via some networking protocol like XMPP or HTTP. This was our original plan, however this proved to be too complex of a system, Firebase offered much more ease of use. However, using a more complemented setup would have given our users more control over their own data as they would have been able to set up their database rather than rely on its developers to provide one.

Firebase is a mobile and desktop application tool meant for synchronized databases [1]. It works similarly to this block's original design; it was originally a chat messenger system that evolved into a system meant to send application data rather than just text. It has a proprietary API that offers authentication, user sign in, and data management. It is a Google product and is well integrated into the Google ecosystem, providing cloud messaging and analytics [2].

The main cost of this block is the time that it will take from researching Firebase's unique and proprietary API and integrating it into the app.

4.2.4. Interface Validation

input bltth_phn_rcptr_rf:

Interface Property	**Why is this interface this value?**	**Why do you know that your design details for this block above meet or exceed each property?**
Data transfer rate: 1Hz	This value was chosen based on the needs of the system.	This value should be the same as the input as this block simply performs a conversion and should not try to analyze the data at all.
Info: Queue: Unhandled messages will be queued and resent until handled	Based off of needs of the graphical processing block	Information only
Bidirectional: This interface can take messages from the app	Based off needs of the app to configure this block	Information only

input bltth_phn_rcptr_rf:

Interface Property	**Why is this interface this value?**	**Why do you know that your design details for this block above meet or exceed each property?**
Data transfer rate: Once per five minutes.	The app will store unsent data in a buffer and will send in batches. It sends at a low rate to not overwhelm the network.	Data does not need to be sent at a very fast rate as it is not very time sensitive
Bidirectional	Can send and receive data from the server	Information only
Protocol: Firebase	Firebase JSON messages	Information only

4.2.5. Verification Plan

1. Meet with group to verify the interface from device to app, and between different blocks of the app.

2. Write Bluetooth retrieval pseudo code to analyze logic
3. Write Firebase pseudo-code to analyze logic.
4. Integrate into the app.
5. Unit test to discover bugs.
6. Repeat from step 3 onward until all platforms are satisfied.

4.2.6. Revision Table

Date	Description
Jan 7, 2022	Document creation
Jan 8, 2022	Added Revision table, description, JSON, interface, verification plan
Jan 15, 2022	Added more JSON fields and example
Jan 16, 2022	Added pseudocode
Jan 20, 2022	Added Validation tables
Jan 21, 2022	Rewrote general validation, verification plan. Added block diagram
Apr 21, 2022	Rewrote description, psuedo code, validation for new Firebase approach.

4.2.7. References

[1] S. Melendez, “Sometimes you’re just one hop from something huge,” Mar. 15, 2020. [Online]. Available: https://www.fastcompany.com/3031109/sometimes -youre -just-one -hop -from - something-huge (visited on 05/27/2014).

[2] J. Tamplin, “Firebase expands to become a unified app platform,” May 18, 2016. [Online]. Available: https://firebase.blog/posts/2016/05/firebase -expands -to -become -unified -app - platform (visited on 04/21/2022).

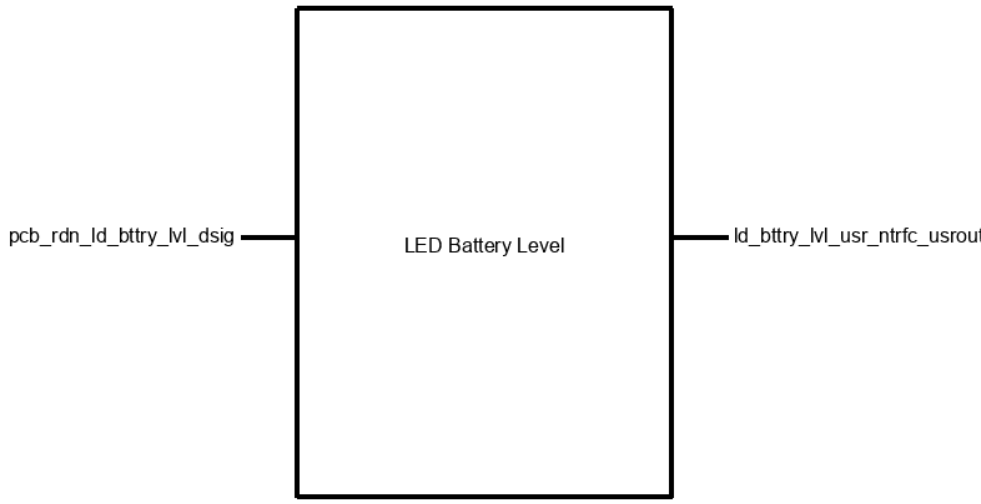
4.3. LED Battery Level (Champion:)

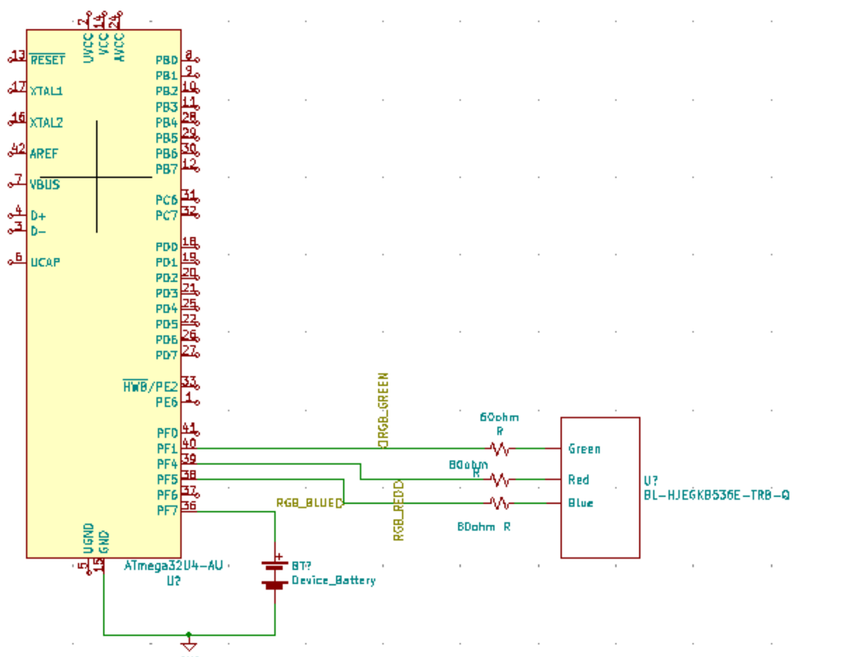
4.3.1. Description

This block will act as a monitor for the Battery to indicate to the user what the battery level is currently at. There will be an RGB light that will represent the current battery level by displaying different colors. There will be three stages for the battery. The Figure below shows the corresponding color given the Battery level.

The microcontroller that will change the RGB light BL-HJEGKB536-TRB-Q[1] will be the Atmega32u4-XUMU[2]. The Atmega32u4-XUMU will be operating as the Arduino Leonardo. There will also be three resistors to protect the BL-HJEGKB536-TRB-Q. The BL-HJEGKB536-TRB-Q will illuminate at least around 20 lumens to the user. The power to the system will be a 3.7V 1000mAh battery.

4.3.2. Design





4.3.3. General Validation

The BL-HJEGKB536-TRB-Q is the ideal light for our Tactical time Logging system. The reason being there are currently 25,177 in stock and it only requires 20mA to drive each leg of the LED. This would come to about a total draw of ao60mA. Another bonus of the BL-HJEGKB536-TRB-Q is that the price of the LED is only $0.48. This will allow the team to purchase a few of them to test the constraints of the BL-HJEGKB536-TRB-Q and not worry about destroying one.

The 3.7V 1000mAh battery was selected because it would be able to power the Arduino Leonardo when connected a boost converter. These batteries are very common and can be ordered on amazon.

4.3.4. Interface Validation

Interface Property	Why is this interface this value?	Why do you know that your design details **for this block** above meet or exceed each property?

pcb_rdn_ld_bttry_lvl_dsig: input

I Forward: 40mA	This forward current will be driven by the Arduino Leonardo off of an I/O pin	For the BL-HJEGKB536-TRB-Q from the Datasheet. • The Voltage range for 0-3.2 for the Super Blue and Green. • The Voltage range for orange/red is 0-2.4 • Peak Wavelength will be around 630nm for orange/red • Peak Wavelength will be around 530nm for orange/red • Peak Wavelength will be around 470nm for orange/red • Peak Forward current 100mA
Vmax: 3V	This will be the maximum voltage for the Green and super blue connections.	For the BL-HJEGKB536-TRB-Q from the Datasheet. • The Voltage range for 0-3.2 for the Super Blue and Green. • The Voltage range for orange/red is 0-2.4 • Peak Wavelength will be around 630nm for orange/red • Peak Wavelength will be around 530nm for orange/red • Peak Wavelength will be around 470nm for orange/red • Peak Forward current 100mA

Vmin:0.5	Since it is an LED the minimum Voltage will only determine how many lumens it outputs. Therefore 0.5 volts will be the bottom so the LED still illuminates some light.	For the BL-HJEGKB536-TRB-Q from the Datasheet. • The Voltage range for 0-3.2 for the Super Blue and Green. • The Voltage range for orange/red is 0-2.4 • Peak Wavelength will be around 630nm for orange/red • Peak Wavelength will be around 530nm for orange/red • Peak Wavelength will be around 470nm for orange/red • Peak Forward current 100mA
Vtypical: 2-2.8V	This will be the typical range of the voltage applied to the connectors based on the Datasheet.	For the BL-HJEGKB536-TRB-Q from the Datasheet. • The Voltage range for 0-3.2 for the Super Blue and Green. • The Voltage range for orange/red is 0-2.4 • Peak Wavelength will be around 630nm for orange/red • Peak Wavelength will be around 530nm for orange/red • Peak Wavelength will be around 470nm for orange/red • Peak Forward current 100mA

Output: ld_bttry_lvl_otsd_envout

Luminous Intensity Max: 317 mcd	The typical luminous intensity for green is 317 mcd so it won't be able to go higher than that.	For the BL-HJEGKB536-TRB-Q from the Datasheet. • The Luminous intensity range for green is 140-317mcd • The Luminous intensity range for orange/red and Blue is 18.5-35mcd
Luminous Intensity min: 0 mcd	This value is for when the battery is totally dead and there is no longer any power to the LED making it have a 0 mcd	For the BL-HJEGKB536-TRB-Q from the Datasheet. • The Luminous intensity range for green is 140-317mcd • The Luminous intensity range for orange/red and Blue is 18.5-35mcd

4.3.5. Verification Plan

1. Test Green
    a. Set DC power supply to 4.2V and then check the voltage before and after both 80Ω. Then check the voltage on both sides of the 60Ω.
    b. Calculate the current flowing to the BL-HJEGKB536-TRB-Q.
2. Test Yellow
    a. Set DC power supply to 3.8V and then check the voltage before and after both 80Ω. Then check the voltage on both sides of the 60Ω.
    b. Calculate the current flowing to the BL-HJEGKB536-TRB-Q.
3. Test Red
    a. Set DC power supply to 3.6V and then check the voltage before and after both 80Ω. Then check the voltage on both sides of the 60Ω.
    b. Calculate the current flowing to the BL-HJEGKB536-TRB-Q.
    c. References

4.3.6. Revision Table

4.3.7. References

[1]"BL-HJEGKB536E-TRB-Q." https://www.americanbrightled.com/pdffiles/led-components/smt/BL-HJEGKB536E-TRB-Q.pdf (accessed Jan. 12, 2022).

[2]"Features • High Performance, Low Power AVR ® 8-Bit Microcontroller • Advanced RISC Architecture -135 Powerful Instructions -Most Single Clock Cycle Execution -32 x 8 General Purpose Working Registers -Fully Static Operation -Up to 16 MIPS Throughput at 16MHz -On-Chip 2-cycle Multiplier • Non-volatile Program and Data Memories -16/32KB of In-System

4.4. Accelerometer (Champion: Ali Alhabshi)

4.4.1. Description

This document is going to describe and give an overview of the accelerometer sensor block for the tactile time logging system project. The tactile time logging system is similar to a running watch or a fitbit that is going to track various data. One of the requirements of this project is to have the system report the user's physical steps. Therefore, this block is going to be dedicated for building the accelerometer block that will be connected to the main project's system. The accelerometer will act as a Pedometer sensor that counts steps of the user and will be sent as data to the website application for the users.

4.4.2. Design

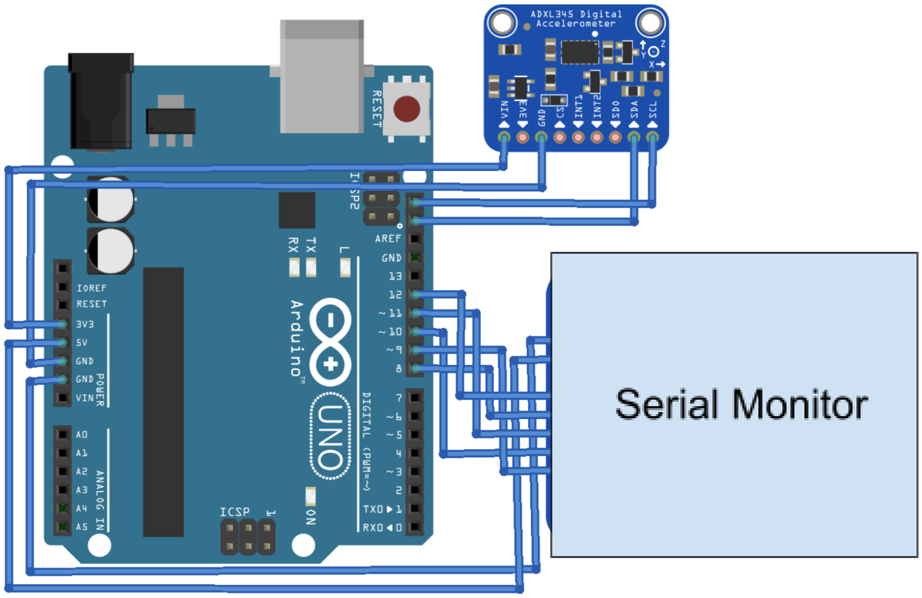


Figure 3: Schematic wiring and Circuit diagram pinout example

4.4.3. General Validation

This block is one of the most crucial parts of the tactile time logging system project as it serves one of the main purposes of the system. The running band will include an accelerometer/pedometer sensor that monitors the user's steps. A system that monitors the user's steps is going to be built. The MPU-6050 sensor was first considered; however, due to its complexity, the ADXL345 sensor is chosen to be used as the accelerometer sensor for this block. The main purpose of this block is to make a pedometer which counts the steps of the user. This sensor greatly meets and fulfills the requirements. Once it's programmed by the arduino, the block will produce the desired values.

To validate that this block is working, the serial monitor tool in the ARDUINO IDE software will be used to observe the performance of the sensor. The sensor is expected to give user steps data and display it on the serial monitor. The sensor will be swinged back and forth during the test and check if its given accurate results of the "steps". An alternate solution for the design is to change the sensor module if the results were not accurate. Possible failures might happen due to an error from the code; therefore, further steps will be taken such as proofreading the code from other team members.

4.4.4. Interphase Validation

Input: **ardn_hrt_rt_snsr_dcpwr**

Interface Property	Justification	Evidence
Vmax: 3.6 V	The sensor is expected to operate in a maximum rating of 3.6V. This is the maximum and the safest value for the system to operate.	From datasheet (pg 1) : The ADXL345 operates from 1.8V and 3.6V power supplies and can be powered down through software. [2]
Vmin: 1.8-2.0 V	The sensor is expected to operate in the range of 1.8-3.6V. A range of the minimum value that could be operated is 1.8-2.0V	From datasheet (pg 3): Operating Voltage Range minimum is 2.0 V [2].
Vnominal: 3.3V	The sensor is expected to operate in the range of 1.8-3.6V. Also, power from the arduino is enough to turn on the sensor from the 3.3V pin.	From datasheet (pg 1) : The ADXL345 operates from 1.8V and 3.6V power supplies and can be powered down through software. Nominal or typical operating voltage is 3.3V as we can provide from the arduino[2]

Input: **otsd_acclrmtr_dcpwr**

Interface Property	Justification	Evidence
Ipeak: 12mA	This value is the expected peak current of the block. The peak shouldn't be reached as it's safe to deal with only the nominal current which is between 8-9mA. 6-12mA is the full range.	The current flowing through pins in the arduino is between 6-12mA. DC current of I/O pins max is 40mA; therefore it's in the correct and safe range. [4]
Inominal: 8-9mA	The sensor supply current is between these values when connected with the arduino.	From datasheet : The current flowing through pins in the arduino is between 6-12mA. DC current of I/O pins max is 40mA; therefore it's in the correct and safe range. [4]
Other: powered by 5V	In order for the arduino to operate, it requires a voltage of 5V from the usb source.	From datasheet [4]: Operating Voltage (logic level) 5 V.

Output: **acclrmtr_ardn_usrout**

Interface Property	Justification	Evidence
Other: Receive data of X and Y axis or vectors	To get the value of steps, an algorithm is set to calculate the acceleration vector from the starting vector point and further compare the value with the threshold value	From datasheet [2]: Several special sensing functions are provided. Activity and inactivity sensing detect the presence or lack of motion and if the acceleration on any axis exceeds a user-set level. Also, SENSOR INPUT Measurement Range: User Selectable for each axis. (page 3).
Type: Step counts (Numerical value)	The system is set to monitor and record the user's steps. Number will be displayed as an output.	This property is set by the owner of the system so they can control from the arduino code the type of output to be displayed.
Usability: Yes	The results of the sensor are usable for user data access.	Serial monitor data is used and multiple users can use this function to get "step count".

4.4.5. Verification Plan

1. **Otsd_acclrmtr_dcpwr:** Powering on the system with its specified power by using a usb cable connected from the laptop to the arduino. This would suggest that the arduino nano is operating with a VIN value of 5V.
2. Calculating properties of **otsd_acclrmtr_dcpwr** and **aardn_hrt_rt_snsr_dcpwr** with a multimeter. Will test that pin 3.3V on arduino is outputting 3.3V. Other min and max values will be tested by changing operating values. Will also test the Inominal and Ipeak by creating an open circuit between Vin and ground.

3. The rest of the **acclrmtr_ardn_usrout** properties such as type and usability will be tested by the user data being displayed on the serial monitor. Step counts will be compared perhaps with a different pedometer to see/check accuracy. The X and Y axis will be tested by using a tool to test accuracy.

4.4.6. Revision Table

2/2/2022	Created the document and added sections
2/3/2022	Created blocks
2/3/2022	Added interfaces
2/3/2022	Proofread the document
2/4/2022	Finalized Draft
2/18/2022	Added a schematic design
2/18/2022	Proofread the document
2/18/2022	Finalized Draft

4.4.7. References

[1] DIY ARDUINO PEDOMETER
https://circuitdigest.com/microcontroller-projects/diy-arduino-pedometer-counting-steps-using-arduino-and-accelerometer

[2] ADXL345 DATASHEET
https://pdf1.alldatasheet.com/datasheet-pdf/view/254714/AD/ADXL345.html

[3] Arduino-based walking steps and distance calculator August 10, 2021 By Nikhil Agnihotri.
https://www.engineersgarage.com/arduino-based-walking-steps-distance-calculator-adxl345/

[4] Arduino nano datasheet by farnell
http://www.farnell.com/datasheets/1682238.pdf

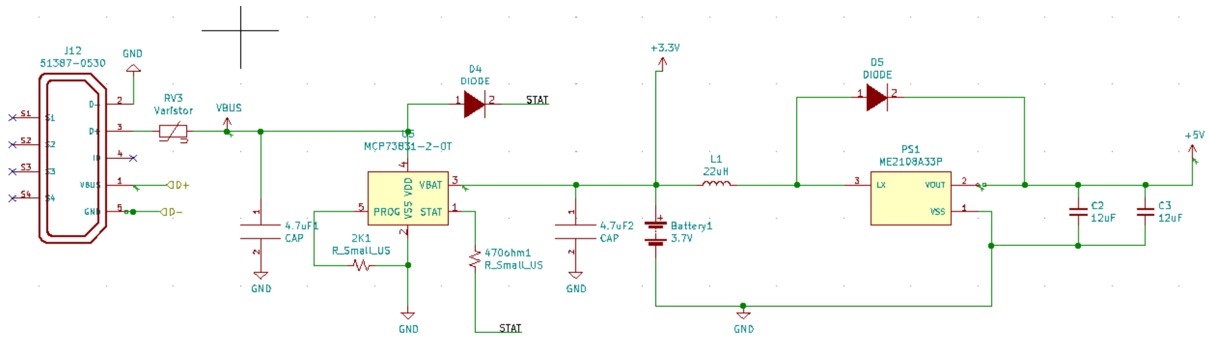
4.5. Battery Charger (Champion:)

4.5.1. Description

This block will be that charger for the two 3.7V, 1000mAh batteries. The power will be supplied through a mini USB supply DC. This block will allow the AC current to be converted into DC current to charge the batteries. The block will also act as the power supply to

Atmega32u4-XUMU. To reach the 5V needed for the Atmega32u4-XUMU a boost converter is used to increase the 3.7V to 5V. Using rechargeable batteries will help Tactile link logging systems to have increased battery life and make it easier for users to maintain the device. Having an enclosed system will also help with the durability and resistance of water.

4.5.2. Design



4.5.3. General Validation

This block will accomplish the task of powering the rest of the system with two 3.7V batteries. The 3.7V 1000mAh battery was selected because it would be able to power the Arduino Leonardo when connected to a boost converter. It could also be stepped down to the 3.3V if needed but all components are rated for the 3.7V. These batteries are very common and can be ordered on amazon.

The block will receive power from a Mini USB that will be connected to the MCP73831-2-OT. The was MCP73831-2-OT selected because of the cheap cost and the familiarity of them from last term with the 18650 battery charger. They are also a common part that hardly ever out of stock. The MCP73831-2-OT will convert the AC current to DC current for the 3.7V batteries allowing them to charge.

Finally the 3.7V battery will be hooked to the ME2108A33P which will act as a boost converter for the Atmega32u4-XUMU. The ME2108A33P will take the 3.7V and step it up to 5V with the help of a few other components like capacitors and an inductor. This was chosen because of the availability of parts.

The alternative for the block would be to just to bypass the ME2108A33P the batteries and the MCP73831-2-OT have have the mini USB supply power to the system. This would be a last case scenario because a filter would need to make the Atmega32u4-XUMU function properly.

4.5.4. Interphase Validation

Interface Property	Why is this interface this value?	Why do you know that your design details **for this block** above meet or exceed each property?

otsd_mn_sb_acpwr: input

Vmax: 6V	This is the max voltage is from the MCP73831-2-OT datasheet	From the MCP73831-2-OT datasheet ● Vmax: 6V ● Vmin: 3.75V ● Inorma:l 510uA
Vmin: 3.75V	This is the min voltage from the MCP73831-2-OT and the battery is rated for 3.7V	From the MCP73831-2-OT datasheet ● Vmax: 6V ● Vmin: 3.75V ● Inormal: 510uA ● Imax: 1500uA
Inormal:510uA	This is the Inormal current from the MCP73831-2-OT datasheet	From the MCP73831-2-OTdatasheet ● Vmax: 6V ● Vmin: 3.75V ● Inormal:l 510uA ● Imax: 1500uA

Imax:1500uA	This is the max current from the MCP73831-2-OT datasheet	From the MCP73831-2-OTdatasheet • Vmax: 6V • Vmin: 3.75V • Inormal: 510uA • Imax: 1500uA

Output: otsd_mn_sb_acpwr

Vmax: 7.0V	This is the max voltage from the ME2108A33P datasheet	From the ME2108A33P datasheet • Vmax: 7.0V • Vmin: 2.0V • Inormal:400mA • Imax: 1mA
Vmin: 1.0V	This is the min voltage from the ME2108A33P datasheet	From the ME2108A33P datasheet • Vmax: 7.0V • Vmin: 2.0V • Inormal:400mA • Imax: 1mA

Inormal:400 mA	This is the Inormal in the voltage range from 2.0V to 5.0V from the ME2108A33P datasheet	From the ME2108A33P datasheet ● Vmax: 7.0V ● Vmin: 2.0V ● Inormal:400mA ● Imax: 1mA
Imin:1mA	This is the min current from the ME2108A33P datasheet	From the ME2108A33P datasheet ● Vmax: 7.0V ● Vmin: 2.0V ● Inormal:400mA ● Imax: 1mA

4.5.5. Verification Plan

otsd_mn_sb_acpwr: input

1. Measure the battery voltage at the start of the demonstration.
2. Supply a varying load to the Mini USB (3.75V-6V).
3. Check the voltage and current going into the batte. The range should be 3.3V to 4.2V
4. Remove the battery and measure again to verify that the voltage has increased.

Output: otsd_mn_sb_acpwr

1. Measure the battery voltage at the start of the demonstration.
2. Measure the voltage and current flowing out the ME2108A33P.
3. Remove the battery and measure again to verify that the voltage has decreased showing the ME2108A33P was getting power from the battery.

4.5.6. Revision Table

Date	Work done
2/4	Created document and filled in all sections
2/16	Updated Description
2/17	Complete Interface Validation
2/18	Revised Referencers, Verification plan and General Validation.

4.5.7. References

ME2108A33P
https://datasheet.lcsc.com/lcsc/1811061121_MICRONE-Nanjing-Micro-One-Elec-ME2108A33PG_C236803.pdf

MCP73831-2-OT http://ww1.microchip.com/downloads/en/DeviceDoc/20001984g.pdf

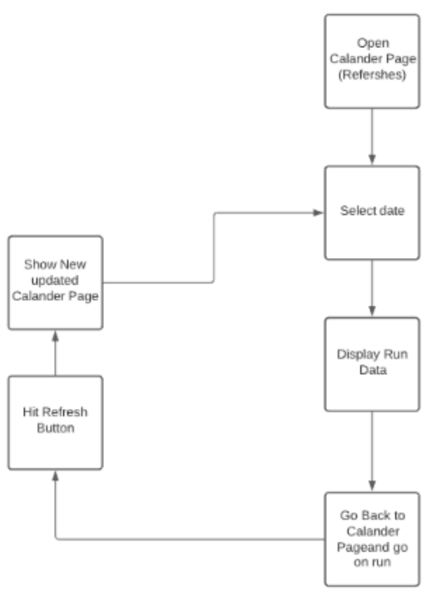
4.6. Display Data (Champion: Hudson Mazza)

Note: This Block Description is out of date. While implementing the system we migrated to a Firebase database structure and a different way to display data. As our project changed different requirements were interfering with each other and changing this block fixed the issues.

4.6.1. Description

This block takes information gathered from the database and displays the information of runs that have been tracked with the Run Buddy(our name for the capstone). It will be showing a calendar like a page that will have an icon showing what day a run occurred, and then when that date is selected display the data related to the run. This Monday will be coupled from a mySQL database.

4.6.2. Design



4.6.3. General Validation

This block will use what was verified in my UI block (Which verified color choices, page layout, and editing of text sizes) and build an actual app page out of it. It will have 2 inputting interphases, one being the data and the other being the user input. The needed resources for this block is the FLUTTER software (to build the app), a mySQL server to store data, and me to code the block. The reason I chose FLUTTER is because that is what me and my coding partner are familiar with. We could've used the android developers studio, but FLUTTER was where you're both most comfortable with. Furthermore, while there are plenty of different servers to hold data, I am not familiar with SQL, which is why I chose that one. With experience I know that SQL will be a good database for the candidate in the store. I also know that SQL communicates with flutter, so it's a good fit.

4.6.4. Interphase Validation

data-in

Interface Property	**Why is this interface this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Messages: Heartrate Data(BPM), Run Data(Speed[Varies], Distance[Varies])	These are the type of data that will be going into the app and that will be tracked	My SQL is able to hold and store many data types

user-in

Interface Property	**Why is this interface this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Other: App will update shown data whenever the calendar page is loaded	This is when the app updates the user data.	Pacifically coding a page to update data when that page is loaded isn't difficult.
Other: calendar page will have a running shoe icon on days that a run had occurred	this is this value because it shows quickly and easily visually where data is present on what days	Sigma has some built-in widgets that allow you to import calendar templates. Furthermore I could add a function that would add an

		icon to show and data is present on a certain date.
Other: Refresh Button	This is a button that would refresh the calendar page with more ease, as opposed to backing out of the page and then re-entering it.	I can just code this button to reload the current page. It would essentially be a button that leads back to the current page.

4.6.5. Verification Plan

1. Show a calendar like page with markers showing the days where runs have occurred
2. When a day is selected, show the run information of that day
3. Back out and manually put in new run information
4. Go back into run calendar page and show new run data that was received (verifies that
backing out of the Calendar page and re-entering updates the data)
5. Without backing out add new data
6. Hit refresh page (verifies refresh button)
7. Display new run data.

4.6.6. Revision Table

Date	Work done
2/18/2022	Created document and filled in all sections
2/16/2022	Updated Description
2/4/2022	Complete Interface Validation
2/4/2022	Revised Referencers, Verification plan and General Validation.

4.6.7. References

"Flutter - implementing pull to refresh," GeeksforGeeks, 01-Jun-2021. [Online]. Available: https://www.geeksforgeeks.org/flutter-implementing-pull-to-refresh/. [Accessed:19-Feb-2022].
S. Enoque, "How to connect flutter app to mysql web server and phpmyadmin," Medium, 24-Feb-2019. [Online]. Available:

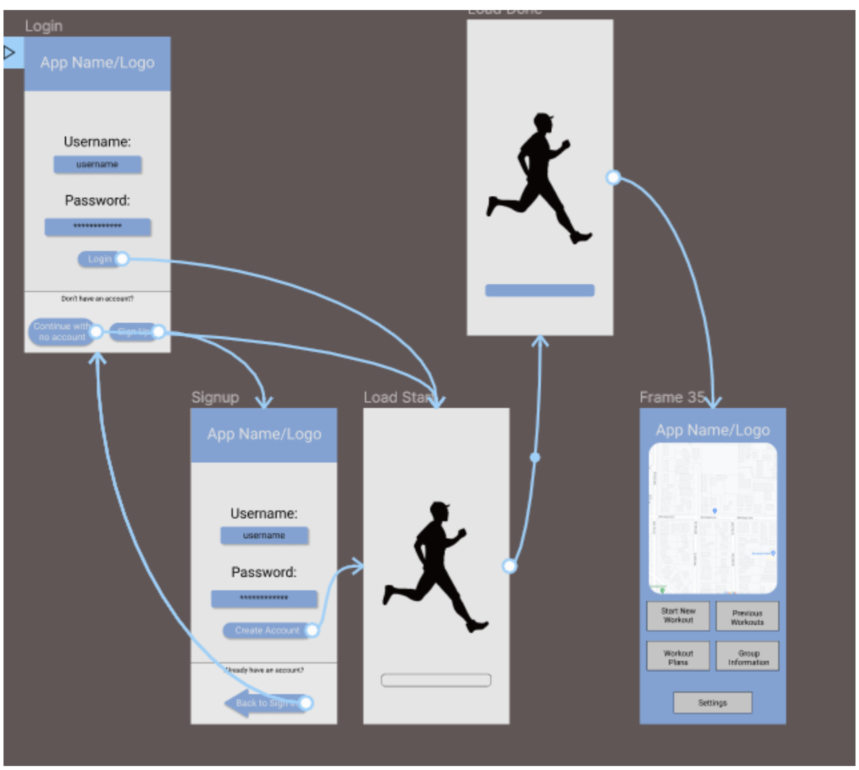
https://santosenoque.medium.com/how-to-connect-flutter-app-to-mysql-web-server-and-phpmyadmin-e100f47bfb82. [Accessed: 19-Feb-2022].
S. khan, “Display dynamic events at calendar in flutter,” Medium, 15-Mar-2021. [Online]. Available:
https://medium.flutterdevs.com/display-dynamic-events-at-calendar-in-flutter-22b69b29daf6 [Accessed: 19-Feb-2022].

4.7. User Interface(Champion: Hudson Mazza)

4.7.1. Description

This block is mostly dedicated to how our companion application looks and how users interact with it. This encopases general application design, user testing, and useability settings (text size, color blindness settings, ect.) and how the user interacts with those settings.

4.7.2. Design



4.7.3. General Validation

This block is fairly vague. It has a lot of ways to make the final product meet or exceed expectations, which are to make the app look appealing and have a clear and straightforward design. This block includes some user testing to observe and research anything from color selection to missing accessibility settings and how users interact with them.

The interphase phn_bltth_rcptr_applctn_sr_ntrfc_data is the data coming in from the database to the application. Otsd_applctn_sr_ntrfc_usrin is the user input that affects the interphase of the application.

The only resources needed for this block is the coder and the software FIGMA, both which require no cost and are accessible at all times. FIGMA is a design tool for creating a Wireframe Mockup of an applications interphase to test and walk through different pages of the application. While this is not a fully formed application, it helps with the validation of this block because it enables people to perform usertesting in a quicker manner than fully developing an app. This will be useful to test the flow and layout of the application. The biggest cost would be time, both in coding and in doing the user research. Furthermore, the design and refining process of that design will take time as well.

4.7.4. Interphase Validation

phn_bltth_rcptr_applctn_sr_ntrfc_data

Interface Property	**Why is this interface this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Datarate: 1 Mbps	Bluetooth 4 runs at a rate of 1 Mbps	Not much data will be transferred so this is more than enough datarate for the information we have.
Messages: Heartrate Data(BPM), Run Data(Speed[Varies], Distance[Varies])	These are the type of data that will be going into the app and that will be tracked	I will make users in the user tests go through different ‘goals’ to see how intuitive the application is.

otsd_applctn_sr_ntrfc_usrin

Interface Property	**Why is this interface this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Useability: Adaptable Text Size	This is this value to enable people with visual impairments to change the size of the text to enable them to read it better	This will be fairly easy to meet as long as I can make it so the text size globally can be changed
Useability:	This is this value to enable	This will be fairly easy to

Color Blind Settings	people with visual impairments to change the colors of the application to preset colorbild settings to enable colorblind users to view the app more easly.	meet as long as I can make it so the Color Schemes globally can be changed in one setting change.

4.7.5. Verification Plan

1. Design rough first draft of application
    a. Alongside with Next steps
        i. Refine general page according to user tests
        ii. Reach out to target audience and do user testing/research
        iii. Evaluate results with group and change what is needed
2. Research how to globally change text size
3. Research Colorblind friendly color schemes/how to implement color blind settings
4. Implement Changing Text Size Globally
5. Implement Colorblind friendly color schemes/Colorblind settings

4.7.6. Revision Table

Name	Date	Revision
Hudson M	1/21/2022	Made Final Submission Changes
Hudson M	1/7/2022	Made document
Hudson M	1/7/2022	Wrote in Document

4.7.7. References

H. Mazza, “FIGMA Wireframe Mockup,” Figma, 07-Jan-2022. [Online]. Available: https://www.figma.com/proto/bj9CyWZ5wiy5Xd3D1MpvOG/CapStone?node-id=4%3A4amp;starting-point-node-id=4%3A4. [Accessed: 22-Jan-2022].

4.8. Heart Rate Sensor (Champion: Ali Alhabshi)

4.8.1. Description

This document is going to describe and give an overview of the heart rate sensor block for the tactile time logging system project. One of the requirements of this project is to have the system report the user's heart rate. Therefore, this block is going to be dedicated for building the heart sensor block that will be connected to the main project's system. The block is currently set to involve a heart rate sensor, arduino nano and a breadboard.

4.8.2. Design

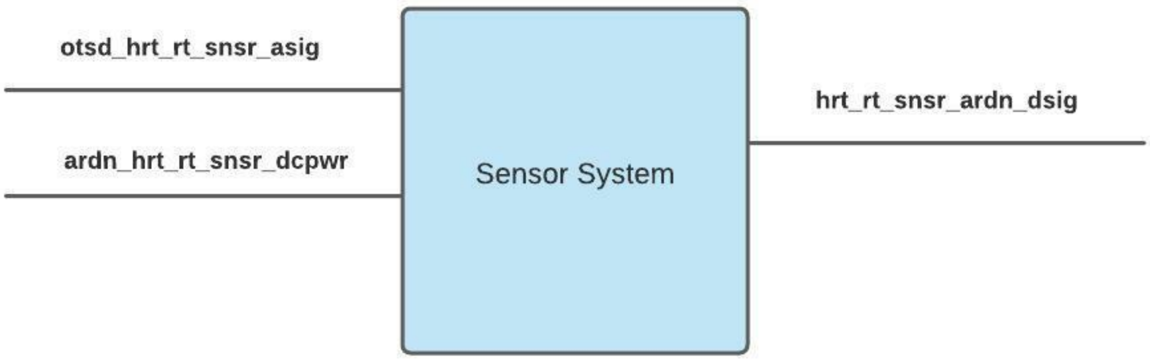


Figure 2: Sensor system black box diagram

4.8.3. General Validation

This block is one of the most crucial parts of the tactile time logging system project as it serves one of the main purposes of the system. The running band will include a heart rate sensor that monitors the user's heart rate and sends it to the user's connected device. In this block, a heart rate sensor and an arduino nano will be used in order to meet the requirements or scope of the project.

A system that monitors the heart rate of a user is going to be built. The pulse oximeter sensor Max30100 is going to bse used. This sensor combines two LEDs, a photodetector, optimized optics, and low-noise analog signal processing to detect pulse oximetry and heart-rate signals.[4] It will be connected to an arduino nano to program the sensor and direct it to specific actions. When implemented in the main system of the project, the microcontroller is just going to send and transfer the user's heart rate data to the application/website. The heart pulse is going to be monitored every 1 second.

To validate that this block is working, the serial monitor tool in the ARDUINO IDE software will be used to observe the performance of the sensor. The sensor is expected to give pulse rates in BPM every 1 second when in contact with the user's skin.

4.8.4. Interphase Validation

Input: **otsd_hrt_rt_snsr_asig**

Interface Property	Justification	Evidence
Vrange: 5 V	The heart sensor is expected to operate in this range. Also, power from the arduino is enough to turn on the sensor.	From datasheet (pg 1) : The MAX30100 operates from 1.8V and 3.3V power supplies and can be powered down through software. [1]
Inominal: 6-12mA	The sensor supply current is between these values when connected with the arduino.	From datasheet (pg 1) : The MAX30100 operates from 1.8V and 3.3V power supplies and can be powered down through software with **negligible standby current**, permitting the power supply to remain connected at all times. [1]
Time delay: 1s	This property was chosen based on the design we planned. The system is set to monitor and record the heart pulse every 1 seconds.	This property is set by the owner of the system so they can control from the arduino code how long the delay is.

Input: **ardn_hrt_rt_snsr_dcpwr**

Interface Property	Justification	Evidence
Vrange: 5 V	The heart sensor is expected to operate in this range. Also, power from the arduino is enough to turn on the sensor. Therefore, arduino pin **3.3V** is going to be connected to the sensor to turn it on.	From datasheet (pg 1) : The MAX30100 operates from 1.8V and 3.3V power supplies and can be powered down through software. [1]
Vin: 7-12V	In order to provide the system with power, the system needs the arduino power. Arduino Nano will operate at a range of 6-20V.	From datasheet : Input Voltage (recommended) 7-12 V. [3] Input Voltage (limits) 6-20 V.
Nominal current: 50mA	As the heart sensor operates in a very small current, the dc current of i/o pins should be fine.	From datasheet (pg 3): DC Current per I/O Pin 40 mA. [3]

		From datasheet (pg 1) : The MAX30100 operates from 1.8V and 3.3V power supplies and can be powered down through software with **negligible standby current**, permitting the power supply to remain connected at all times. [1]

Output: **hrt_rt_snsr_ardn_dsig**

Interface Property	Justification	Evidence
Type: BPM	Type of data to be monitored is BPM.	From article: Heart Beat/Pulse Rate is measured in BPM. [4]
Time: 1s	The system is set to monitor and record the heart pulse every 1 seconds	This property is set by the owner of the system so they can control from the arduino code how long the delay is.
Usability: Yes	The results of the heart pulse are usable for user data access.	Serial monitor data is used.

4.8.5. Verification Plan

1. Powering on the system with its specified power by using a usb cable connected from the laptop to the arduino. This would suggest that the arduino nano is operating with a VIN value between 7-12V.

2. Calculating properties of **otsd_hrt_rt_snsr_asig** and **ardn_hrt_rt_snsr_dcpwr** with a multimeter. Will test that pin 3.3V on arduino is outputting 3.3V. Moreover, due to the negligible value of current operating on heart sensor it will be neglected.

3. **hrt_rt_snsr_ardn_dsig** will be tested by the heart rate pulse being displayed on the serial monitor and checking time with every pulse. BPM value is going to be verified by a method which is either checking actual BPM by a smartwatch or performing a calculation.

4.8.6. Revision Table

[1] Pulse Oximeter and Heart-Rate Sensor IC datasheet for Wearable Health
https://datasheets.maximintegrated.com/en/ds/MAX30100.pdf

[2] “Max30100 pulse Oximeter Arduino Code, circuit, and Programming,” Electronic Clinic, Feb. 23, 2020. https://www.electroniclinic.com/max30100-pulse-oximeter-arduino-code-circuit-and-programming/ (accessed Jan. 22, 2022).

[3] Arduino nano datasheet by farnell
http://www.farnell.com/datasheets/1682238.pdf

[4] Admin, H., (2020, May 10). *Blood oxygen & BPM monitor with MAX30100 Pulse Oximeter & Arduino*. How To Electronics. Retrieved January 21, 2022, from https://how2electronics.com/blood-oxygen-heart-rate-monitor-max30100-arduino/

4.8.7. References

1/4/2022	Created the document and added sections
1/5/2022	Created blocks and flowcharts
1/7/2022	Finalized Draft
1/7/2022	Proofread the document
1/20/2022	Updated flowchart
1/20/2022	Updated block diagram
1/20/2022	Added a schematic of the design
1/20/2022	Updated interfaces properties by showing more explanation.
1/21/2022	Added new references
1/21/2022	Updated testing process as advised by instructor.
1/21/2022	Proofread the document and clarified some wording as advised by the instructor.

5. System Verification Evidence

5.1. Universal Constraints

5.1.1. The “system may not include a breadboard” constraint will be met because the final system will have a custom PCB with soldered-on parts, not a breadboard.

5.1.2. The “final system must contain both of the following: a student-designed PCB and a custom Android/PC/Cloud application” constraint will be met by our custom PCB designed by Ali and Alex and our android application designed by Hudson and Ryan.

5.1.3. The “if an enclosure is present, the contents must be ruggedly enclosed/mounted as evaluated by the course instructor” constraint will be met by having all parts properly mounted and tested to the Halt and Hass Product drop test standards, as listed in Section 2.

5.1.4. The “If present, all wire connections to PCBs and going through an enclosure (entering or leaving) must use connectors” constraint will be met because all but 1 wire is self-contained within the enclosure. The one wire out will be a port that will be used to charge the battery.

5.1.5. The power supply will be a 3.7V battery that will be stepped up to 5V with a boost converter. We will use the ME2108A33PG for the DC to DC converter. The ME2108A33PG has a typical efficiency of 85%, which will satisfy the goal of our system is 65% efficiency.

5.1.6. The “system may be no more than 50% built from purchased modules” constraint will be met because all parts will be bought individually and connected to the custom PCB. Any modules will be used for testing and will be stripped and used for parts.

5.2. Battery Life

5.2.1. The system will operate normally for 2 hours on a full battery.

5.2.2. Verification process

- 5.2.2.1.1. Remove the batteries from the system.
- 5.2.2.1.2. Attach a DC power supply to the system.
- 5.2.2.1.3. Set the DC power supply to 3.7V
- 5.2.2.1.4. Observe the draw in Amperes of the system.
- 5.2.2.1.5. Calculate the total draw by using $\frac{Ah}{A}$ To find the total amount of hours the battery will last. To find the exact number of minutes multiply by 60.

5.2.3. Was checked off during the winter term using a placeholder system. The system needed to operate for 2 hours and we wear using the two 1000mAh batteries in series with one another. Making for a 2000mAh battery. The system pulled 0.06 mA making the system be able to run for well over 2 hours.

$\frac{2000mAh}{0.06mA} = 333.33\ hours$. *{Verified May 4th, 2022}.*
Video proof: https://media.oregonstate.edu/media/t/1_gezx3jtj

5.3. Heart Rate

5.3.1. The system will collect the user's heart rate.
5.3.2. Verification process
- 5.3.2.1.1. Turn system on
- 5.3.2.1.2. Make sure the sensor is working properly and illuminating light
- 5.3.2.1.3. Place heart rate monitor on your wrist
- 5.3.2.1.4. Observe the readings in the serial monitor or on the phone application
- 5.3.2.1.5. Compare readings to average resting heart rate of humans and then measure the pulse for 30 seconds to find heart rate. Compare

5.3.3. *{Verified May 4th, 2022}* The following video shows proof that the system is able to collect fluctuating heart rate data from the sensor. Video: https://media.oregonstate.edu/media/t/1_7n4rk0uk

5.4. Hardware Sends Data To Phone

5.4.1. The system must be able to transmit data collected wirelessly to a phone in order to receive and organize data to be viewed by the user
5.4.2. Verification process
- 5.4.2.1.1. Power system with batteries
- 5.4.2.1.2. Open phone application
- 5.4.2.1.3. Connect to bt05 through bluetooth
- 5.4.2.1.4. Observe data being received in application
- 5.4.2.1.5. Compare to Serial monitor to verify that

5.4.3. *{Verified May 4th, 2022}.* This Video Shows the Accelerometer working in and transmitting the data to the cell phone application.

5.5. Team View

5.5.1. User teams where users can view others' data share.
5.5.2.
- 5.5.2.1.1. Create two accounts that are part of a team.
- 5.5.2.1.2. Show that the app displays the information of these two accounts in real time.
- 5.5.2.1.3. Create a third account.
- 5.5.2.1.4. Show that the system does not show information about the new account since it is not part of the group.

5.5.3. *{Verified May 4th, 2022}.* This video shows a working Team View system where a pre-existing account and a new account join a team and are able to get live updates to changes on the other user's data. This video shows that other users are not able to see data from other users unless they are in that group.

5.6. Information Display

5.6.1. User information is displayed in the App.

5.6.2.

5.6.2.1.1. Open app.
5.6.2.1.2. Log in to pre-existing account.
5.6.2.1.3. Show that user metadata and run data is displayed.
5.6.2.1.4. Show that the account's data is unique by showing off multiple accounts.

5.6.3. *{Verified May 4th, 2022}.* The team view also shows proof of this requirement as well. Once the user is logged in they can see their distance traveled in meters, their current heart rate in beats per minute, their stride that they can set in centimeters, the last time any of this information was updated, and the names of everyone else on their team.

5.7. Durability Testing

5.7.1. The system will be durable to Halt and Pass Product drop test standards

5.7.2.

5.7.2.1.1. Drop the system from a height of three feet.
5.7.2.1.2. Show that the system still transmits data to the app.

5.7.3. *{Verified May 6th, 2022}* This video shows the system being dropped from above 3 feet and then transmitting to the cell phone application via bluetooth.

5.8. Alerts

5.8.1. The user is alerted for special events they set.

5.8.2.

5.8.2.1.1. Create an event that triggers once per second.
5.8.2.1.2. Show that the event triggers once per second.
5.8.2.1.3. Show that the user can toggle between events and no events.

5.8.3. *{Verified May 4th, 2022}.* This video shows that when the slider is on the notifications show up once every second, but when the slider is turned off the notifications stop.

5.9. Tracking Steps

5.9.1. The system will report user steps and distance traveled in meters.

5.9.2.

5.9.2.1.1. Turn the system on.
5.9.2.1.2. Take two steps.
5.9.2.1.3. Check the app to see the system transmit two steps.
5.9.2.1.4. Stop walking.
5.9.2.1.5. Show that the system has stopped sending steps to the app.

5.9.3. *{Verified May 6th 2022}* This Video Shows the Accelerometer working in and transmitting the data to the cell phone application.

5.10. References

5.10.1. "DC/DC step-up converter ME2108 series ... - datasheet.lcsc.com." [Online]. Available: https://datasheet.lcsc.com/lcsc/1811061121_MICRONE-Nanjing-Micro-One-Elec-ME2108A33PG_C236803.pdf. [Accessed: 07-Mar-2022].

5.10.2. "Product drop testing," *HALT & HASS*. [Online]. Available: https://www.halthass.co.nz/reliability-services/environmental-testing-category/product-drop-testing/. [Accessed: 06-Mar-2022].

5.10.3. https://learnmetrics.com/battery-life-calculator-how-long-does-a-battery-last/

5.11. Revision Table

Date	Who	Section	Changes
3/5/2022	Hudson	5.1	Added 5.1.1-5.1.6
3/5/2022	Ali	-	Updated Section 2
3/6/2022	Alex	5.4	Added References
3/6/2022	Ryan	5.3	Added Unique User Data Requirement
3/6/2022	Alex	5.2	Added Battery Life Requirement
3/14/2022	Ali	5.3	Updated section 5.3

Table 21: Revision Table Section 5

6. Project Closing

6.1. Future Recommendations

6.1.1. Technical Recommendations

1. **Minimize Size**: System design could be made smaller in size for a better market product. Minimizing PCB design would make the final product look smaller in size with a more convenient fit on everyday runners' wrists. Also, changing batteries used to a smaller size would definitely minimize the overall size as well. The reason behind this recommendation is that as it is a running band competing with other bands on the market, a size consideration is important for a comfortable wearable device for users.
2. **Accelerometer**: Our accelerometer sensor wasn't the best in terms of accuracy. The team recommends looking into better solutions concerning this aspect of the system. Furthermore, a better accelerometer candidate would include an accurate sensor in terms of X and Y values and an adaptable version to a no I2C connection to an arduino. I2C bus connectivity to a microcontroller is bad because as one of the wires disconnects the whole system shuts off. The team thinks these aspects make the version of the accelerometer not very reliable to use.
3. **Heart rate sensor**: The team struggled to find an accurate heart rate sensor to be placed on the wrist. The current sensor gives accurate readings when placed on fingers unlike a wrist. Wrist-based sensors are hard to find; therefore, the team recommends putting more effort into perhaps creating a sensor from scratch to meet the system's requirement.
4. **Software**: Making an Android app was difficult due to the asynchronous nature of app development being very different from what most computer science students are used to. The team recommends making a website instead of an app. Website development is easier than app development and the tools are probably more familiar to the average programmer.

6.1.2. Global Impact Recommendations

1. **Minimizing Cost**: The team didn't really have a solid plan of budget. A bill of materials is recommended for keeping track of components bought and their costs. This would help to do an analysis in minimizing the cost of the project and keeping track of components. Consumers would be more likely to use/buy a product that is low cost and efficient.
2. **Reducing Garbage**: This also falls into the bill of materials umbrella. The team has ordered plenty of components in which some failed to meet our expectations so they were excluded and thrown away. A bill of materials would show for future references what components are used, what not and their quantity.

6.1.3. Teamwork Recommendations

1. **Communication**: A team should set at least two fixed weekly meetings to discuss anything concerning the project. The team should maximize communication between each other throughout the year to have a more collaborative working environment. Team members might lose track of the project's progress and have no decent updates on it.
2. **In-person meetings:** The team lacked in-person meetings as we just felt we didn't need them. Most of the team meetings were held in discord except for the last month to work on the physical project. We believe that meeting in-person could enhance better technical and communication skills. Meeting in-person would also help in providing a future look for a plan for the team to settle on.

3. **Connect with mentor**: We had a point where some team members struggled in their roles as we had a period of time where we lost the working hands of one team member. Future recommendations on this would be to contact your mentor or project partner right away if you ever experienced a problem and you needed help with. Don't hesitate to tell them what you actually need as they might fully provide you with the ideal solution. Also, if you also needed some technical recommendations, they'd be the first to give you once you meet with them. None of the team members went to meet with a mentor and that clearly reflected on our progress.
4. **Identify problems**: When a team member faces an issue with their responsible task, they should be identifying the problem first. Also, members should work collaboratively and quickly to identify any problem that affects the project progress. This would help fastening the process of solving the issue. Furthermore, if the team needs help from the project partner, they should be very clear about it and ask for the help immediately without any hesitation.

6.2. Project Artifact summary with links

6.2.1.1. App Code

The code for the app is contained in the lib/ directory on the Runbuddy github repository. Most of the code is in main.dart, which defines the basic layout for Flutter as well as implements app interop and displaying user data. The bluetooth code is in lib/blue.dart, which uses the FlutterBlue library for interacting with the Operating system's Bluetooth capabilities. The other files are small auxiliary files used mostly for UI widgets.

https://github.com/Runbuddy-Capstone/runbuddy

6.2.1.2. Arduino Code

The code for the arduino includes the utilization of the sensors in the system. Both the accelerometer and heart rate sensor are programmed here to perform their designated tasks. The heart rate sensor monitors the heart rate of the user while the accelerometer tracks the distance traveled and steps taken. Data is then transmitted to a bluetooth receiver called the bt05 to display it within the companion app.

https://github.com/Runbuddy-Capstone/Arduino

6.2.1.3. App CodTest Code

Runbuddy's bluetooth testing code used for debugging the app's bluetooth capabilities. Written for an Arduino Uno with a bt05 bluetooth module.

https://github.com/Runbuddy-Capstone/bluetooth_test

6.3 Presentation Material

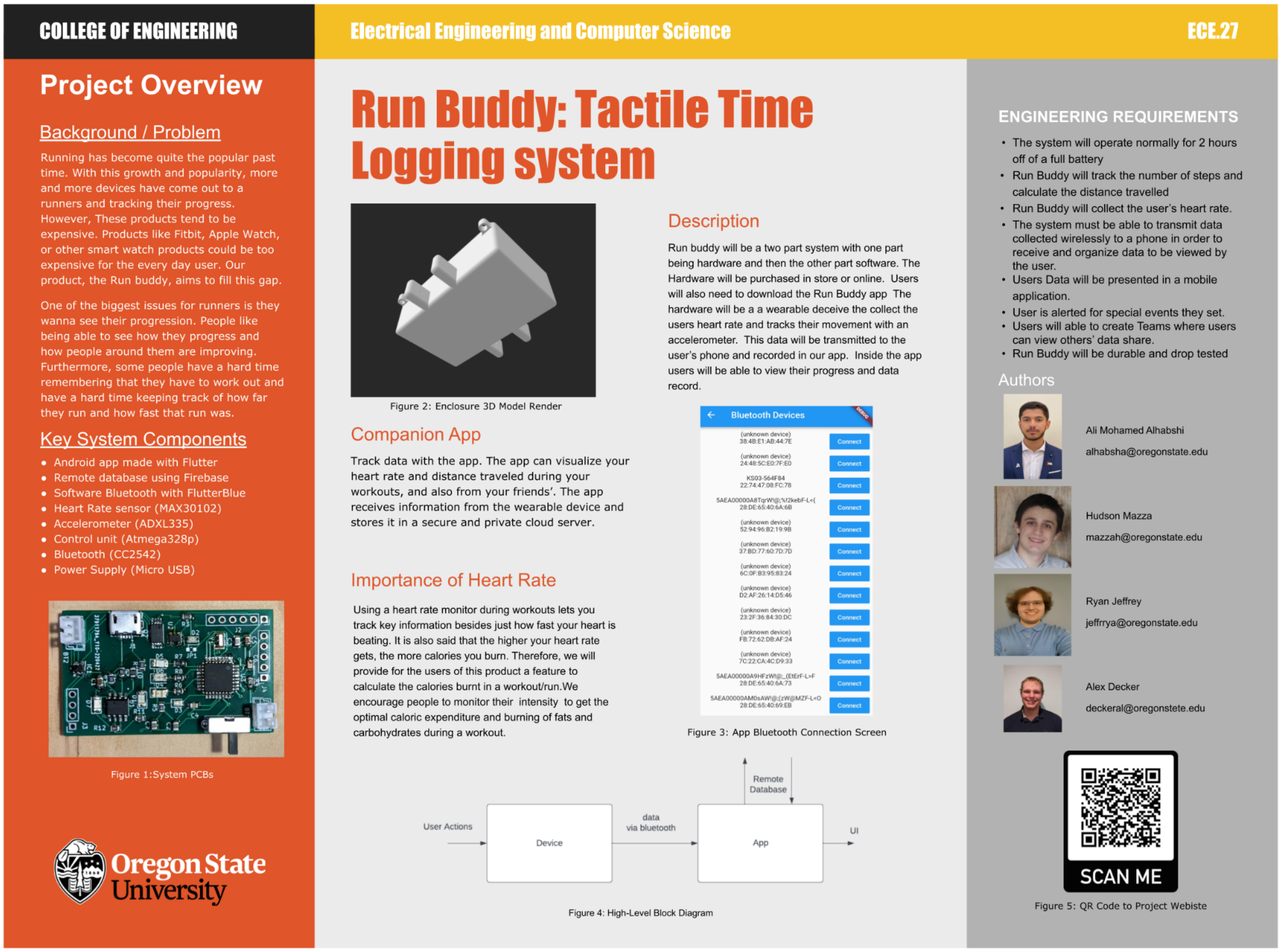


Figure 17: Project poster