

System Design for Mechatronics Engineering

Team #1, Back End Developers

Jessica Bae

Oliver Foote

Jonathan Hai

Anish Rangarajan

Nish Shah

Labeeb Zaker

January 18, 2023

1 Revision History

Date	Version	Notes
18 January 2023	1.0	Revision 0 for System Design

2 Reference Material

This section records information for easy reference.

2.1 Abbreviations and Acronyms

Refer to [SRS](#) for a comprehensive list of abbreviations and acronyms.

symbol	description
Lipo battery	Lithium polymer battery
PCB	Printed circuit board
TFT LCD Display	Thin film transistor liquid crystal display
FSM	Finite state machine
UI	User interface
CAD	Computer aided design for 3D models
SPI	Serial peripheral interface

Contents

1 Revision History	i
2 Reference Material	ii
2.1 Abbreviations and Acronyms	ii
3 Introduction	1
4 Purpose	1
5 Scope	2
6 Project Overview	3
6.1 Normal Behaviour	3
6.2 Undesired Event Handling	3
6.3 Component Diagram	3
6.4 Connection Between Requirements and Design	3
7 System Variables	4
7.1 Monitored Variables	4
7.2 Controlled Variables	4
7.3 Constants Variables	4
8 User Interfaces	4
8.1 Hardware User Interface	4
8.2 Software User Interface	7
9 Design of Hardware	10
10 Design of Electrical Components	13
11 Design of Communication Protocols	15
12 Timeline	16
A Software Interface	18
B Mechanical Hardware	22
C Electrical Components	26
D Communication Protocols	27
E Reflection	27

List of Tables

1	Components of Hardware UI	4
2	Components of Software UI	7
3	Timeline of project	17
4	Components of Hardware Design	24
5	Electrical Components of modules from PCB schematic	26

List of Figures

1	System Context	3
2	FSM for Hardware UI	5
3	Display on activity tracker at startup	6
4	Display of Date/time on activity tracker.	6
5	Main Window for Software UI	8
6	FSM for Software UI	9
7	TFT Display with custom 3D printed case	10
8	Circuit of custom built reactive touch sensor	11
9	CAD Assembly of Touch Bezel, Display case and TFT Display	11
10	Top view of Custom PCB	12
11	Bottom view of Custom PCB (Lipo battery holder and SD card slot)	12
12	Schematic for PCB	14
13	Layout of PCB design	15
14	Configuration Window	18
15	Create Record Window	19
16	Record Window	20
17	Login Dialogue box	20
18	Data View Window	21
19	Generated Graph for Heart Rate vs Time	22
20	CAD model of touch sensor (bezel)	22
21	CAD model of case for TFT Display	23
22	CAD model of watch assembled	23
23	Example of Someone Experiencing a Full Body Suit	30

3 Introduction

This document provides a detailed description of the system design for the EMAnator; the system currently under development by the Back End Developers which aims to assist researchers in performing Ecological Momentary Assessment for older adults. The goal of this design is to construct a system which fulfills all the requirements specified in the [System Requirements Specification](#), and that meets the needs of EMA researchers.

This document provides the big-picture goals of the system, an overview of the project, a comprehensive list of system variables, and details regarding the user interfaces of the system. It also lists the various hardware and electrical components involved in the design of the system. Finally, it includes a high-level timeline for the development of the system.

The design presented in this document is the result of collaboration between the Dr. Luciana Macedo of the School of Rehabilitative Sciences and the Back End Developers development team. We have discussed the project requirements and identified the best approach to meet them. This is covered in detail in the document titled [Problem Statement and Goals](#).

The Back End Developers hope this document serves as a useful guide for anyone involved in the development or deployment of the system.

4 Purpose

In general, engineering design documentation is a set of documents that outline the detailed specifications for an engineering project. It describes the design of the project, including its requirements, the materials to be used, the processes involved, the safety and environmental considerations, and the estimated costs. The purpose of engineering design documentation is to provide a comprehensive record of the project that can be consulted by engineers, oversight bodies, and other stakeholders throughout the project's life cycle.

This documentation includes technical drawings, process diagrams, system and component specifications, and relevant schematics and images. It also provides a basis for quality control and assurance, as well as a way to track progress and identify potential areas of improvement. It is an essential part of any engineering project, as it ensures that all stakeholders have a clear understanding of the project and the necessary steps for its successful completion.

This project currently has the current pieces of design documentation available:

- [Problem Statement and Goals](#)

- Development Plan
- System Requirements Specification
- Hazard Analysis
- VnV Plan
- Module Guide
- Module Interface Specification

5 Scope

This system in theory is very simple. An array of sensors grabs info regarding the position, speed, orientation, etc. and uses that information to understand the current state of the user. This data is then used to generate prompts that the user will answer and all the collected data is compiled, processed and stored within the system. Finally, researchers will analyze the collected data and generate observations.

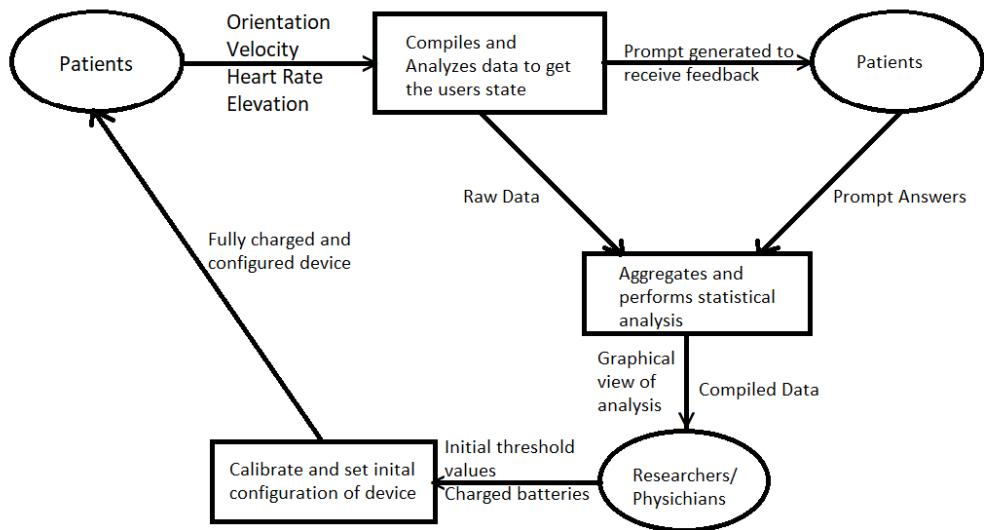


Figure 1: System Context

[Include a figure that show the System Context (showing the boundary between your system and the environment around it.) —SS]

6 Project Overview

6.1 Normal Behaviour

Refer to [SRS](#) Section 12.

6.2 Undesired Event Handling

Refer to [SRS](#) Section 13.

6.3 Component Diagram

6.4 Connection Between Requirements and Design

[The intention of this section is to document decisions that are made “between” the requirements and the design. To satisfy some requirements, design decisions need to be made. Rather than make these decisions implicit, they are explicitly recorded here. For instance, if

a program has security requirements, a specific design decision may be made to satisfy those requirements with a password. —SS]

7 System Variables

7.1 Monitored Variables

Refer to [SRS](#) Section 5.1.2.

7.2 Controlled Variables

Refer to [SRS](#) Section 5.1.3.

7.3 Constants Variables

Refer to [SRS](#) Section 1.4.

8 User Interfaces

8.1 Hardware User Interface

The device is worn by a participant on the wrist for measuring activity and generating prompts. The following items will be shown on the display of the activity tracker:

Description	Behaviour of TFT Display
Power up of activity tracker.	Displays Back End Developers on startup.
Default behaviour, no activity tracked.	Displays date and time.
Activity tracked.	Prompt generated on screen, for example: Are you in pain?
Answering prompts using touch sensor (bezel).	Toggle between different options on screen. For example: (Yes/No).

Table 1: Components of Hardware UI

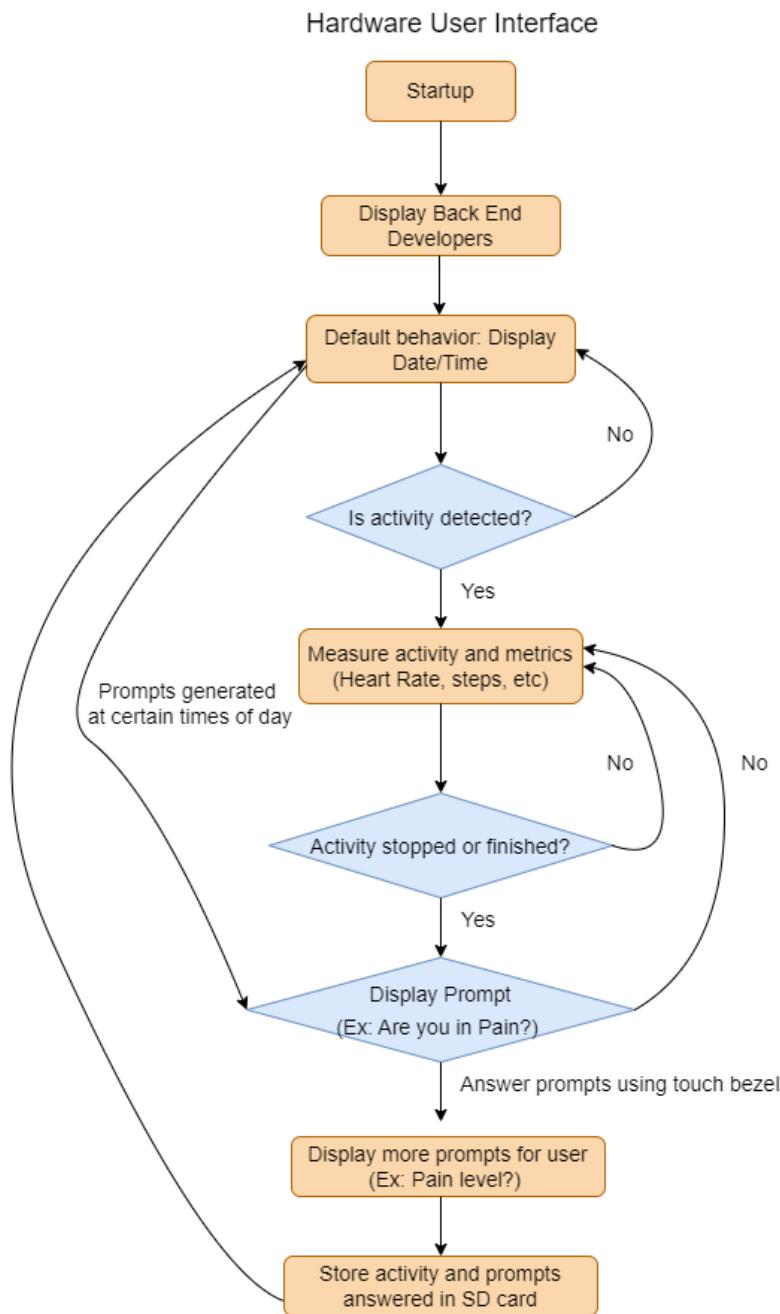


Figure 2: FSM for Hardware UI



Figure 3: Display on activity tracker at startup



Figure 4: Display of Date/time on activity tracker.

8.2 Software User Interface

The Software User Interface will be used by the Researcher for configuring the activity tracker. The interface will be on the Host Computer and will be able to store participant data, create new data and view records using encryption. The interface will also have authentication, and only the Researcher will be able to log in. The following features are available on the Software User Interface:

Options on UI	Description
Main window	Main menu that leads to different windows when clicked.
Connect to tracker	Connects to SD card for device and shows status of connection.
Create Records Window	Creates new record for participant and stores it in a database. A record can only be created if the correct username and password is provided.
Records Window	Participant records can be viewed in a tabular format and can be searched/filtered.
Data View Window	Data stored on SD card can be viewed and filtered. Data can also be plotted using Graph button. For example: Heart Rate vs Time.

Table 2: Components of Software UI

Below is an example of the Software User Interface for the Main window.

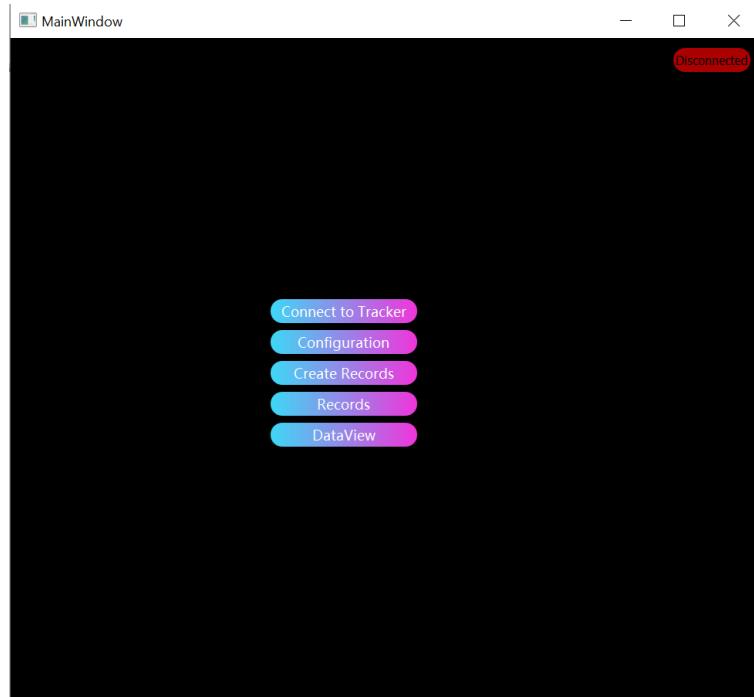


Figure 5: Main Window for Software UI

For more examples of the Software User Interface, refer to Appendix [A](#).

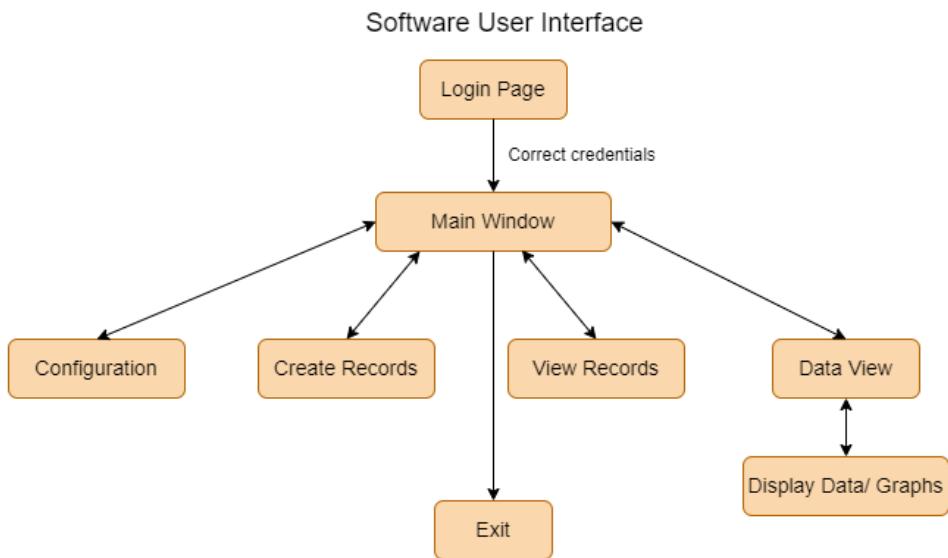


Figure 6: FSM for Software UI

9 Design of Hardware



Figure 7: TFT Display with custom 3D printed case

The touch sensor shown below will be used as a bezel to navigate options on the activity tracker. The wires connected to each segment send a corresponding signal which can be used to navigate options and select a response.

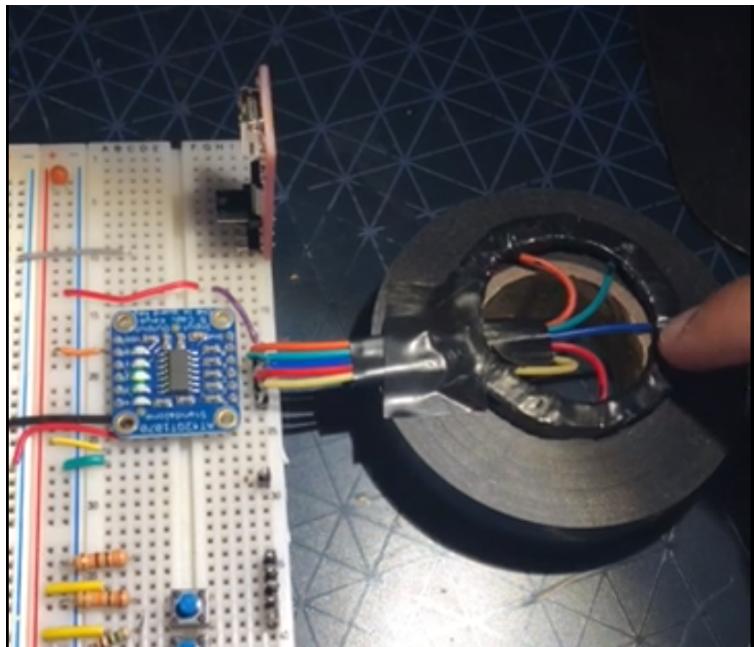


Figure 8: Circuit of custom built reactive touch sensor

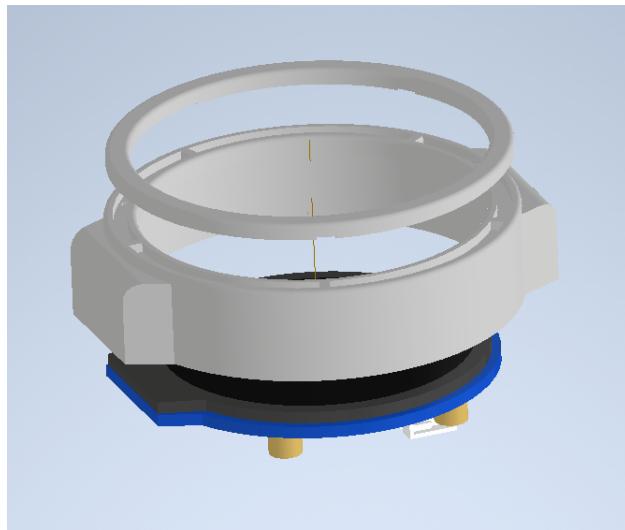


Figure 9: CAD Assembly of Touch Bezel, Display case and TFT Display

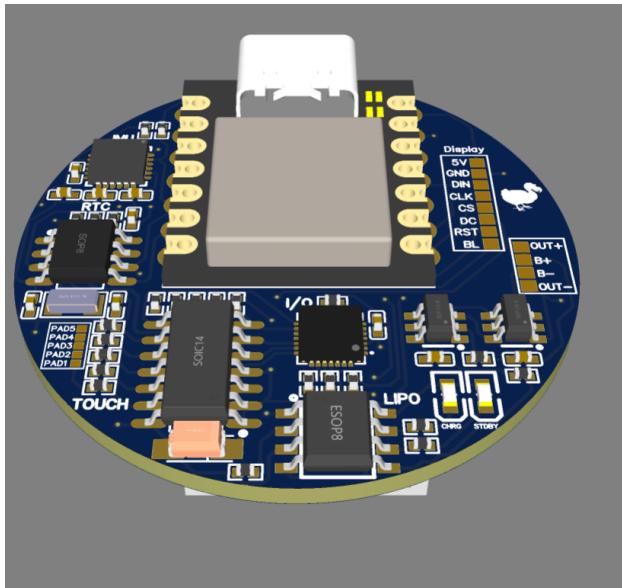


Figure 10: Top view of Custom PCB

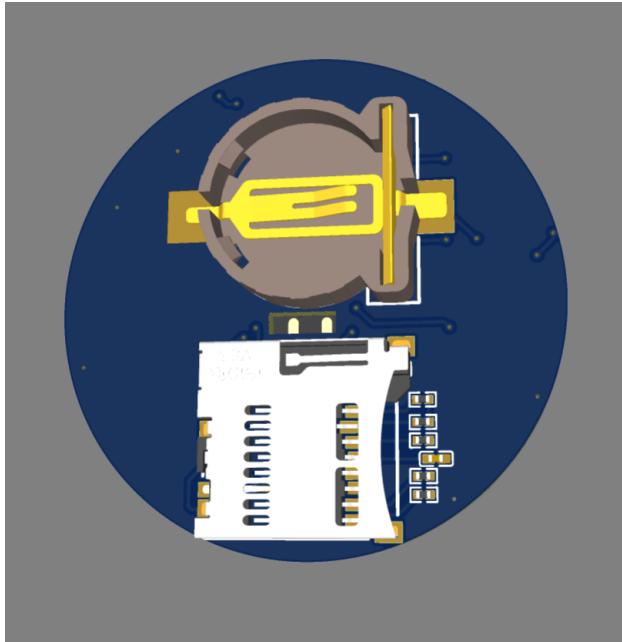


Figure 11: Bottom view of Custom PCB (Lipo battery holder and SD card slot)

Refer to Appendix [B](#) for individual CAD designs.

The following list of items will be custom designed and fabricated:

- Custom PCB designed using [Easy EDA](#) and fabricated using [JLCPCB](#)
- 3D printed casing for TFT Display/activity tracker.
- Touch sensor (bezel) to navigate activity tracker.

Refer to table [4](#) in Appendix for detailed information on high-level hardware components used.

10 Design of Electrical Components

The schematic/circuit diagrams shown below are used to generate the PCB layout. It consists of the following modules:

- Seeduino Micro-controller
- Touch sensor module
- IO Expander
- RTC
- MPU 6050
- SD card module
- Lipo module

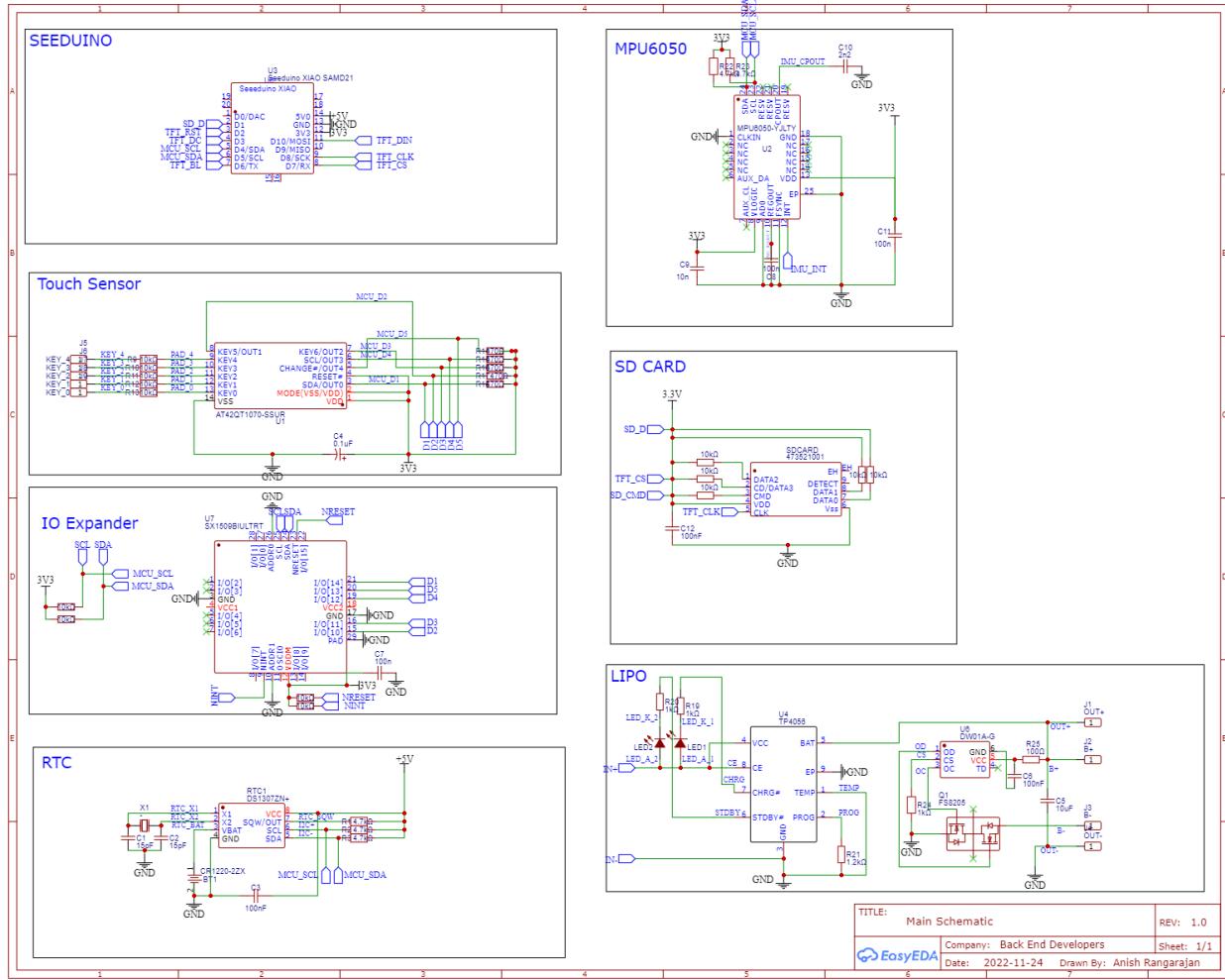


Figure 12: Schematic for PCB

The custom designed PCB is shown below.

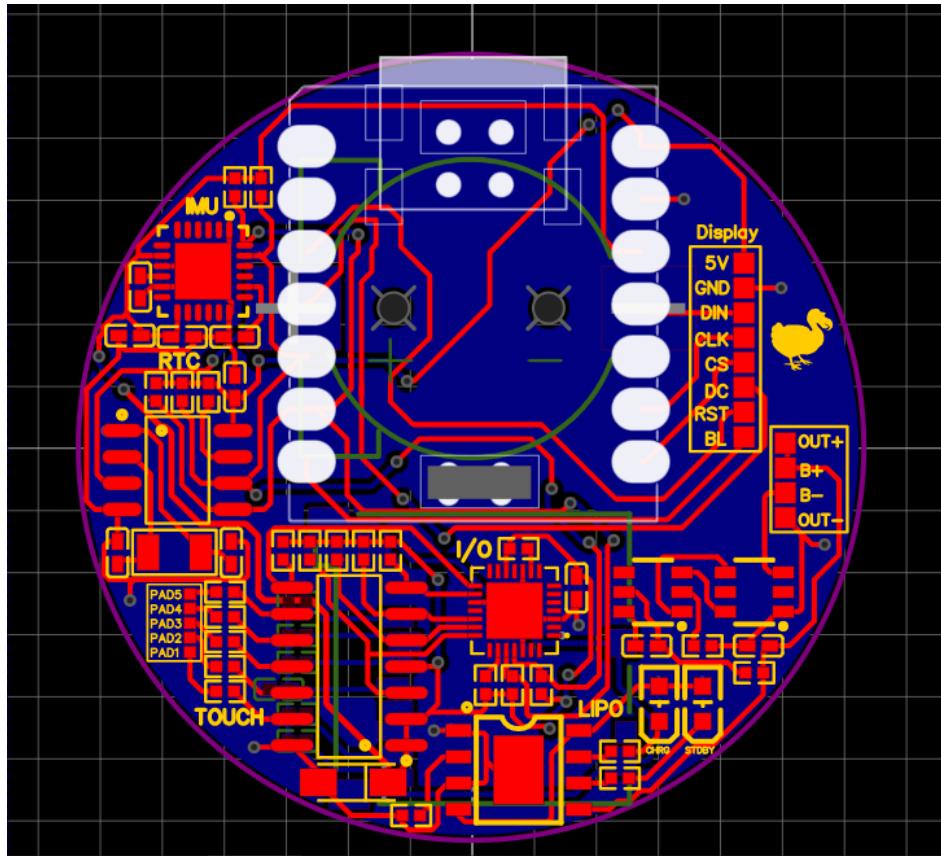


Figure 13: Layout of PCB design

Refer to table 5 in Appendix for detailed information on electrical components (module-wise) used in the PCB design.

11 Design of Communication Protocols

There are 2 standard communication protocols that are used in the project, I2C and SPI. The following modules use I2C communication protocol:

- Code for RTC using I2C communication.
- TFT display driver.
- MPU 6050 driver.

The IO expander module uses SPI communication protocol.

12 Timeline

The project timeline shown below as per important tasks/modules are required for the implementation of the Revision 0 demonstration:

Timeline	Project Task	Team member(s) responsible and duties	Description
Dec 2022 - Jan 2023	Software UI and functionality	Labeeb Zaker (backend), Nish Shah (frontend)	User friendly UI implemented with authentication and encryption working.
Jan 2022	Database Management and writing to SD card	Nish Shah (DBMS), Jessica Bae (Communication protocols)	Implement database with Software UI to read, store and write data. Storing data read from sensor in proper format on SD card.
Dec 2022- Jan 2023	PCB schematic design	Anish Rangarajan (General schematics), Jessica Bae (SDCARD module), Jonathan Hai (Lipo module)	Implement schematics for different modules in PCB.
Dec 2022 - Jan 2023	PCB layout design	Anish Rangarajan	Implement layout of PCB with proper routing.
Nov 2022	Display drivers/CAD models	Labeeb Zaker (Drivers), Anish Rangarajan (CAD)	Display drivers for LCD display and CAD models for device.
Dec 2022 -Feb 2023	Integration of all sensor modules	Jonathan Hai, Oliver Foote	Integrate all sensors and systems and create test benches to check correct functionality.
Jan-March 2023	Fabrication of PCB and Testing	Jessica Bae, Oliver Foote	Fabricating PCB and System-wide testing on PCB based on VnVPlan
Continuous development till March 2023	Documentation	All team members	Various project documentations.

Table 3: Timeline of project

A Software Interface

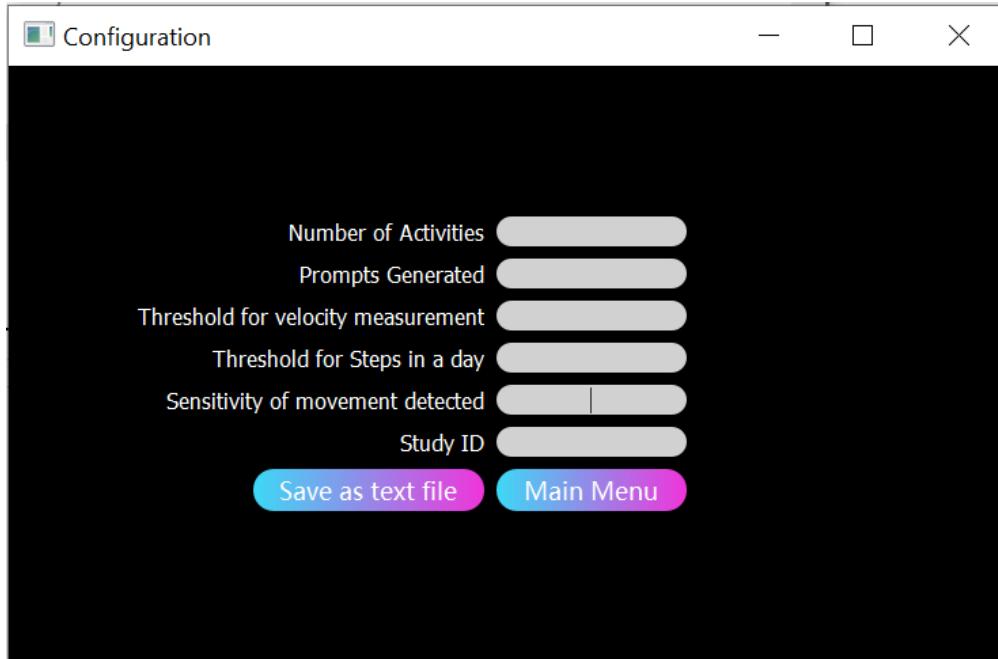


Figure 14: Configuration Window

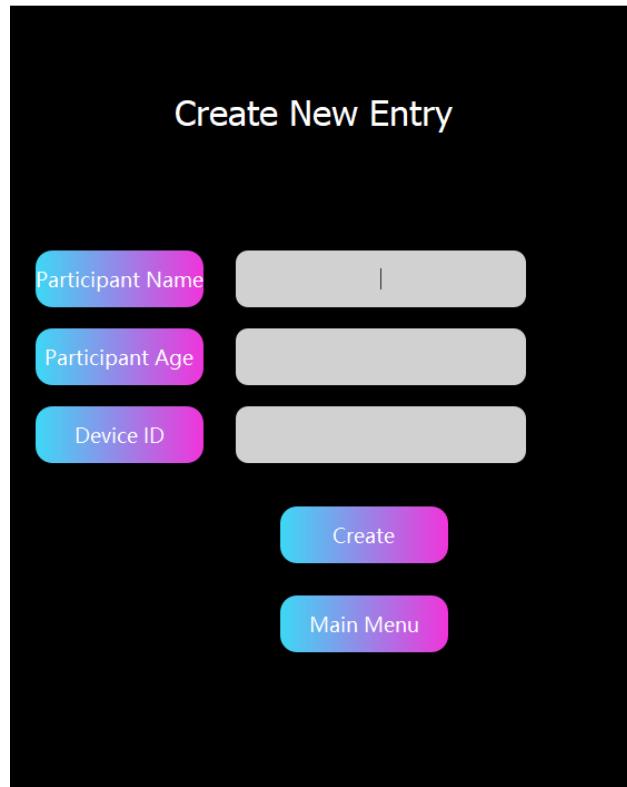


Figure 15: Create Record Window

Record

The Record window contains a top section with input fields for First Name, Last Name, Age, Participant ID, Gender, Weight, Height, Phone Number, Email, Address, Monitoring Period, and Tracker ID. It includes a 'Load CSV' button, a 'Search' button, and a 'Filter' button. Below this is a table with 4 rows of data:

	First Name	Last Name	Age	Participant ID	StudyID	Gender	Weight (kgs)	Height (cm)	Phone number	Email ID	Address	Monitoring Period	Tracker model
1	Jack	Jones	60	1	32	Male	75	180	(338) 437-5840	jack@gmail.com	8710 Hilltop St.	13	1
2	Rose	Lindt	65	2	32	Female	63	150	(864) 315-3964	rose@gmail.com	Mundelein, IL 60060	12	2
3	Ashley	Dunder	63	3	32	Female	57	160	(238) 233-4530	ashley@gmail.com	9022 Jennings Drive	15	1
4	May	Potter	75	4	32	Prefer Not ...	80	172	(990) 200-7813	may@gmail.com	North Miami Beach, FL 33160	15	1

Main Menu

Figure 16: Record Window

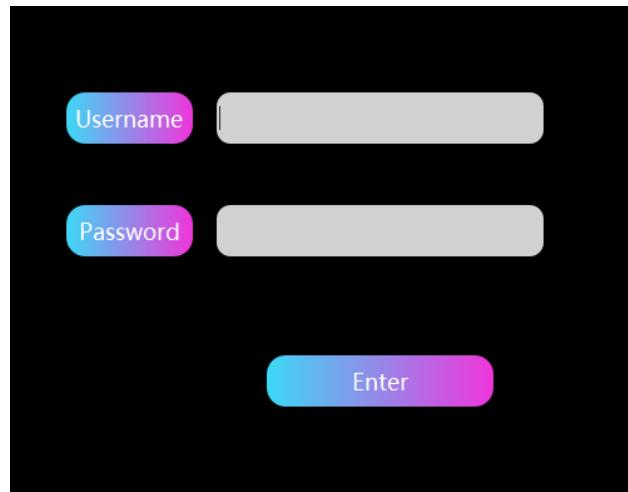


Figure 17: Login Dialogue box



Figure 18: Data View Window

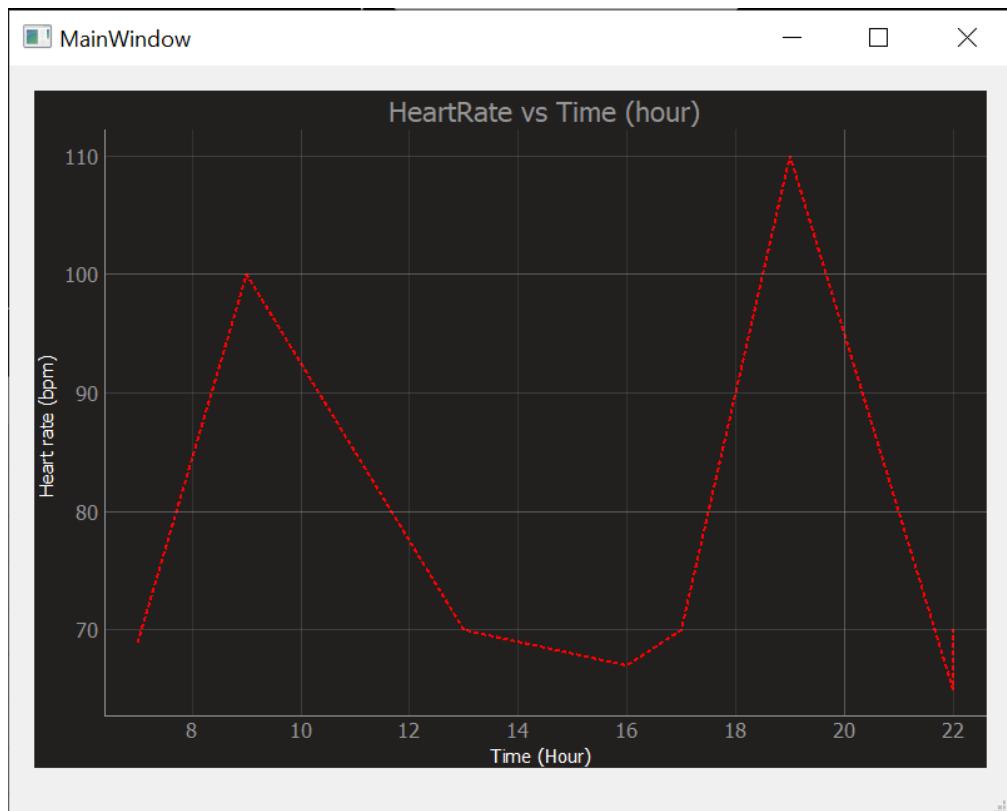


Figure 19: Generated Graph for Heart Rate vs Time

B Mechanical Hardware

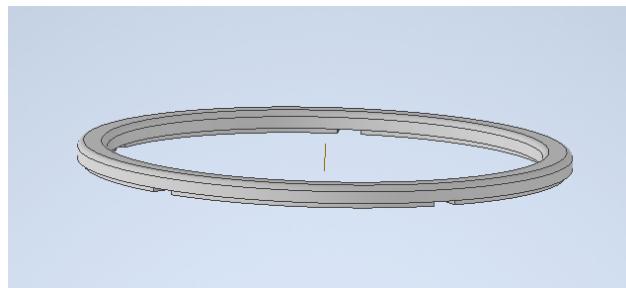


Figure 20: CAD model of touch sensor (bezel)

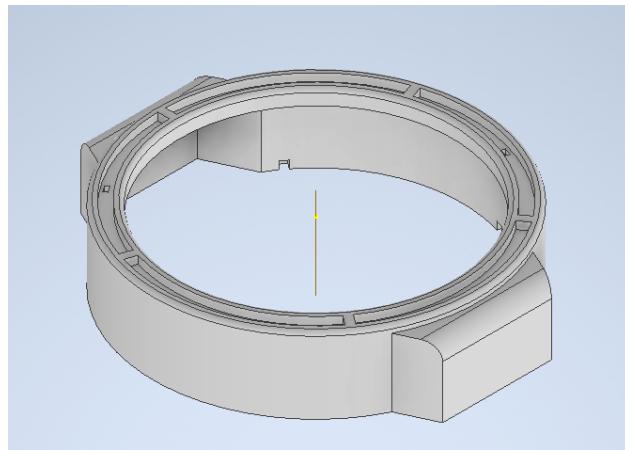


Figure 21: CAD model of case for TFT Display

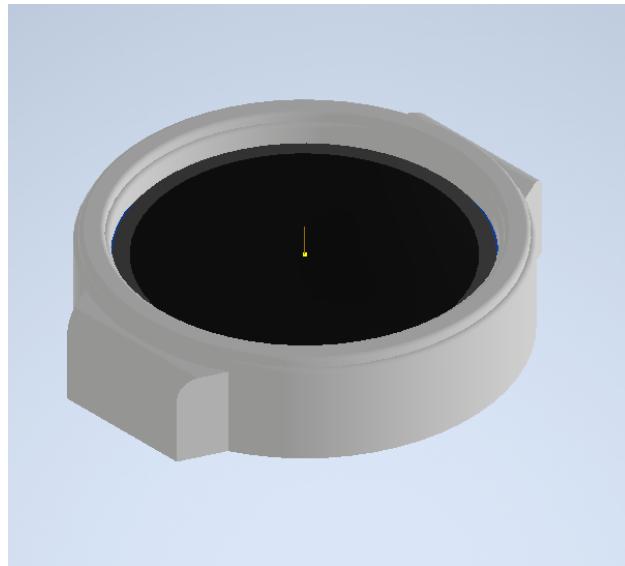


Figure 22: CAD model of watch assembled

Hardware Component	Description
Custom PCB	Custom PCB designed to fit in activity tracker.
Touch sensor module	Custom touch sensor (bezel) used to interact with activity tracker.
Outer casing for TFT Display	Designed using Autodesk Inventor and 3D printed.
AT42QT1070-SSUR	Off-shelf microchip for capacitive touch sensors, used in touch sensor schematic.
MPU 6050	Accelrometer/Gyroscope, off-shelf component.
Seeeduino xiao samd21	off-shelf microcontroller for activity tracker.
DS1307 RTC	Real time clock, off-shelf component.
Waveshare 1.28 LCD-192 Display	Off-shelf TFT display used in activity tracker.
Pulse sensor	Plug and play Heart Rate sensor for Arduino (off-shelf component)
Li-Po Battery	Generic off-shelf lipo battery used for smart watches.
TP4056 Li-Ion BMS	Off-shelf constant-current/constant-voltage linear charge for battery, used in Lipo schematic.
DW01A-G	Off-shelf battery protection IC to protect battery from overcharge, used in Lipo schematic.
USB Type-B charger	Generic off-shelf usb to Type-B charger to charge device.
MicroSD card	Standard off-shelf SD card.
SDCARD connector 473521001	MicroSD connector (off-shelf component)
IO Expander SX1509BIULTRT	off-shelf IO Expander used in PCB.
Watch straps	Generic watch-straps for strapping device onto the wrist.

Table 4: Components of Hardware Design

C Electrical Components

Module from schematic	Electrical component
MPU 6050	<ul style="list-style-type: none"> • Capacitors (2nF, 10nF, 100nF) • Resistors (4.7k ohms)
Touch sensor	<ul style="list-style-type: none"> • Capacitors (0.1uF) • Resistors (470 ohms, 10k ohms)
IO Expander	<ul style="list-style-type: none"> • Capacitors (100nF) • Resistors (10k ohms)
RTC	<ul style="list-style-type: none"> • Capacitors (15pF, 100nF) • Resistors (4.7k ohms)
SD CARD Module	<ul style="list-style-type: none"> • Capacitors (100nF) • Resistors (10k ohms)
Lipo Module	<ul style="list-style-type: none"> • Capacitors (100nF, 10uF) • Resistors (100 ohms, 1k ohms, 1.2k ohms) • FS8205 MOSFET
LEDs	Various LEDs in modules used to identify status of operation.

Table 5: Electrical Components of modules from PCB schematic

D Communication Protocols

E Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Problem Analysis and Design. Please answer the following questions:

1. *What are the limitations of your solution? Put another way, given unlimited resources, what could you do to make the project better?* (LO_ProbSolutions)

This is a difficult question to answer. Given unlimited resources, the Back End Developers would have many more options available in fundamental design, and may have pursued a different design altogether — however going forward with this question it is assumed that this question refers to the current design on hand.

There are many forms of "resources" that have to be managed during an engineering project. Cost, time, testing opportunities, and many more constraining factors can be considered resources. Let us consider cost and time, as those have the largest and most obvious impacts on the project.

Given an unlimited budget, the face of the EMAnator likely would have been a touchscreen, as the cost of a touchscreen was prohibitively high and we had to find a workaround using nearby capacitive touch sensors instead. In addition, we would have kept ordering test PCBs of our designs as we iterated through different versions. This would have allowed us to test individual components and wiring routes as we constructed the final design. As cost was limited, we have to finish the whole design first, and proceed to order a very limited number of test PCBs on which we must test all aspects of our design. We will then revise the design to compensate for issues found on the test boards, order the final set, and pray that they work as intended. We would also consider adding an operating system to the device to make it "smart". Currently our device has a very limited number of functions that fit tightly into our requirements. An OS would enable us to expand the functionality of our device beyond its current borders.

If the design could have taken place over an infinite amount of time, there are additional possibilities that we could consider. Primarily, we could investigate using and shrinking down a more powerful microcontroller. Currently, the type of microcontroller that we can use is extremely limited in its capabilities. However if we wish to upgrade to a more powerful one (e.g. a Raspberry Pi), we would have to place a large number of auxiliary components necessary for the functioning of the more powerful (and more needy) processor. This would have taken a large amount of time; as finding reasonable

space on a PCB and routing components like these is notoriously time consuming.

Obviously there are many new and exciting possibilities for this project should certain resources be unlimited. But overall, we are satisfied with the design that we have made given the constraints that have been placed on us. Should we decide to continue developing this device past the capstone course, we may return to these options at a certain point.

2. *Give a brief overview of other design solutions you considered. What are the benefits and tradeoffs of those other designs compared with the chosen design? From all the potential options, why did you select documented design? (LO_Explores)*

During the initial phases of the design process, we encouraged each other to throw around as many design ideas as we could. Here is a short list of a few we considered, along with a few of their pros and cons:

- Anklet Activity Tracker

Pros:

- Discreet and unobstructive to daily activities.
- High-quality data available regarding motion of legs (steps, jumps, falls, etc.).

Cons:

- Difficult to reach for people with difficulty bending over.
- If designed poorly, can potentially look like a prison ankle monitor.
- Difficult for a user to read any data displayed on a screen on the anklet.

- Waist-Mounted Activity Tracker

Pros:

- Being close to the core of the user, the maximum weight weight of the device can be greatly increased.
- Easy access to those with difficulty bending over, or with other movement related issues.

Cons:

- Located in a hard-to-hide location, and may get in the way of reaching below the waist.
- Incompatible with certain clothing.

- Bracelet Activity Tracker

Pros:

- Extremely discreet.

- High-quality data available regarding motion of arms (walker use, exercise, etc.).
- Not obstructive to daily activities.

Cons:

- Interacting with a small-surface area bracelet may be difficult for those with movement issues.
- Small size reduces battery capacity options.

- Full Body Suit Tracker

Pros:

- Highest fidelity data possible, collecting data from every point on the body.
- Difficult to misplace.

Cons:

- Prohibitively expensive.
- Easy to damage.
- Many people would be uncomfortable wearing a full sized-body suit during all their daily activities.

- Mobile App

Pros:

- Inexpensive.
- Flexible; meaning we can modify it according to new requirements whenever necessary.
- Many are already extremely familiar with activity tracking apps; onboarding process will not be difficult.
- Can be linked to off-the-shelf smartwatches with fantastic activity collecting technology.

Cons:

- Not a mechatronics solution.

Why did we pursue the smart-watch solution? For two very important reasons; end-user familiarity and end-user acceptability.

The other solutions that we could have proceeded with may have met all the functional requirements and many of the nonfunctional requirements that we specified, but many of them are unfamiliar and possibly confusing to the end user. For example, having to wear a box around one's waist is likely a novel and strange experience to any user, and could make them uncomfortable with using the device in general. In addition, most people have worn a watch before, and are very familiar with their operations. This makes the process of learning how to use the watch simpler and more intuitive.

A smart-watch was also deemed the most acceptable solution to an end user. It would be difficult to convince anyone to wear a sensor covered spandex suit everywhere they went in their daily life, for a period of up to 14 days. A watch is familiar, socially acceptable, and much less embarrassing than wearing a full bodied suit similar to the figure below.



Figure 23: Example of Someone Experiencing a Full Body Suit