



Boston University
Electrical & Computer Engineering
EC464 Capstone Senior Design Project

User's Manual

Personal Alert Device



Submitted to

Michael Ruane
151 Broadstreet Hollow Rd
Shandaken, NY 12480-0550
845-688-5357
mfr@bu.edu

by

Team 19
PAD Team

Team Members

Richard Yang richy@bu.edu
Renad Alanazi reenad@bu.edu
Logan Lechuga llechuga@bu.edu
Tanveer Dhillon tdhillon@bu.edu

Submitted: April 18, 2025

Personal Alert Device User's Manual

Table of Contents

Personal Alert Device User's Manual	1
Table of Contents	1
Executive Summary (Team)	2
1 Introduction (Richard Yang)	3
2 System Overview and Installation	4
2.1 Overview Block Diagram (Richard Yang)	4
2.2 User Interface (Richard Yang)	5
2.3 Physical Description (Logan Lechuga)	10
2.4 Installation, Setup, and Support (Logan Lechuga)	14
3 Operation of the Project (Tanveer Dhillon and Richard Yang)	15
3.1 Operating Mode 1: Normal Operation	15
3.2 Operating Mode 2: Abnormal Operations (Richard Yang and Tanveer Dhillon)	17
3.3 Safety Issues (Tanveer Dhillon)	18
4 Technical Background (Richard Yang, Renad Alanazi, and Logan Lechuga)	19
4.1 Hardware Components	19
4.2 Software Components	22
5 Relevant Engineering Standards (Renad Alanzi, Tanveer Dhillon, and Logan Lechuga)	25
6 Cost Breakdown (Richard Yang)	27
7 Appendices	28
7.1 Appendix A - Specifications (Renad Alanazi)	28
7.2 Appendix B – Team Information	29
7.3 Appendix C – Firebase (Authentication and Firestore)	29
7.4 Appendix D – SMS to Designated Contacts	31

Executive Summary (Team)

The population of seniors above 65 is growing rapidly as the Baby Boomers age. Many live alone and struggle with physical or mental disabilities. Home accidents, typically falls, happen even with healthy elders. Current solutions in the industry are expensive and suffer from high false alarm rates, typically triggered by unintended activation of the alert device. A more intelligent, more reliable design is needed.

We propose to deliver a home-based wearable alert actuator that interfaces with the user's mobile device through a native application. By incorporating high-accuracy sensors and utilizing the user's phone as a computational hub, we can better diagnose medical emergencies while providing the user with dynamic health information and records. Through the innovative usage of machine learning models in our system, we can better differentiate and diagnose real emergencies automatically. Through the integration of the device with our native application, a new layer of personalization, usability, interoperability, and adaptive functionality can be achieved, enhancing the overall user experience and effectiveness of our alert device as compared to traditional alternatives.

1 Introduction (Richard Yang)

The concept for the Personal Alert Device is derived from a necessity for a more intelligent and refreshed approach to modern home-based alert hardware. Our system addresses challenges faced by current industry solutions by creating a compact wearable IoT device that offers real-time health monitoring, fall detection, speech recognition, emergency actuation, and overall ease of usage. Complementing the wearable is a mobile app that displays live device information, gathers personal and medical information, and provides immediate emergency notifications to designated emergency contacts.

The Personal Alert Device leverages advanced technologies on both the software and hardware side of the system. Key sensing components chosen for the wearable due to their robustness and popularity include the MAX30102 pulse oximeter and the NTC 10K Ω thermistor. The core component, the XIAO nRF52840 Sense microcontroller, features a ARM Cortex-M4 32-bit processor and 256 KB of memory, which was chosen due to the integrated 6-axis IMU and PDM used for fall detection and speech recognition respectively. Additionally, the microcontroller is compact yet possesses enough memory to support an ML model and live sensor operations. Uniting the microcontroller, sensors, and other components such as LEDs, resistors, and a buzzer is a custom designed PCB which heavily improves the form factor of the device. Finally, a wireless charging system with a separate transmission module was designed for simplified charging of the device's 3.7V 1100mAh LiPo battery with up to 96 hours of continuous usage.

For the software and communications, an Android app was developed as per the project's requirements. Communications via the device and the app are established through BLE (Bluetooth Low Energy), which reduces power consumption while also resolving the issue of a network loss. The communications are facilitated via LightBlue and Adafruit IO, while Google Firebase was chosen for its authentication functionalities, Firestore Database, and connection with existing Google accounts given that the app is Android-based. One special feature of the software integrated in the wearable is the speech recognition model developed in Edge Impulse. The model uses the popular approach of MFCC feature extraction to classify the speech with a 1D convolutional neural network. Finally, for fall detection, a two-interval threshold model was chosen for robustness and to minimize false alarms of misclassified ADLs.

It is important to keep in mind that the Personal Alert Device is still very much a proof of concept and is not to be relied on in a real emergency. Additionally, the wearable is prone to water damage and short-circuiting if water were to get inside the enclosure. Finally, as the device uses a lithium polymer battery, safe handling and storage of the device is important. Nonetheless, personal information is securely stored under unique and dynamic Firestore collections that require Google Authentication to reference from.

The final product is a culmination of a full range of modern technologies ranging from IoT communication, PCB design, machine learning, and cloud integration. The following sections of this manual will provide a comprehensive guide to the Personal Alert Device including its installation, usage, and operations.

2 System Overview and Installation

2.1 Overview Block Diagram (Richard Yang)

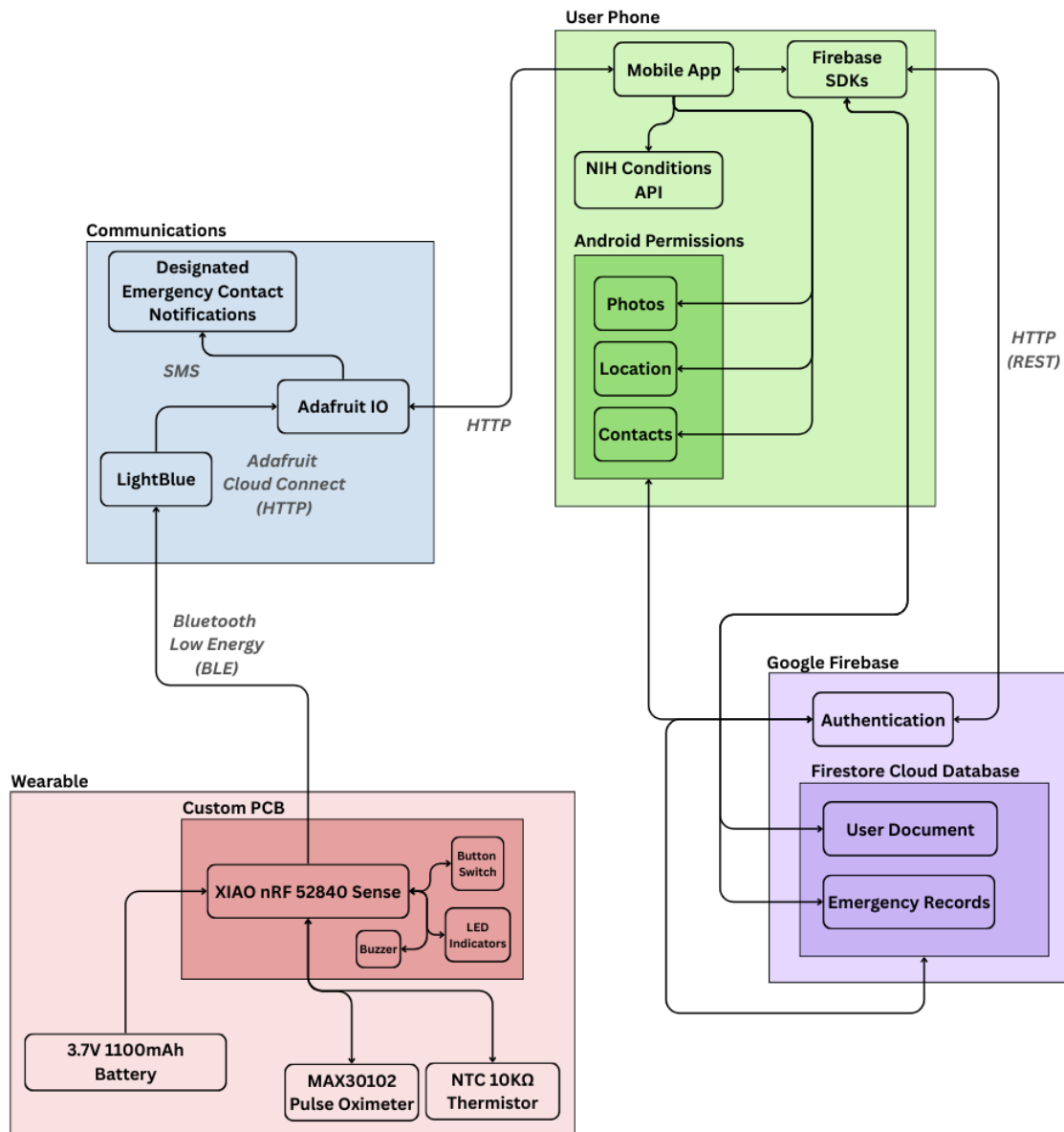


Figure 2.1.1: Overall system block diagram with sub-system block diagrams illustrating communication flow between wearable, app, and cloud. Data collected by the physical device is transmitted via Bluetooth to Adafruit IO. Data in the IO feeds are read from the mobile app and stored in Firestore if the user is authenticated. The mobile app uses the phone's permissions API to gather data automatically. The app can also write to a designated IO feed to send emergency notifications to emergency contacts.

2.2 User Interface (Richard Yang)

The figures shown below serve as a visual demonstration of user interface design, cloud based data storage, emergency response handling, and overall app functionality.

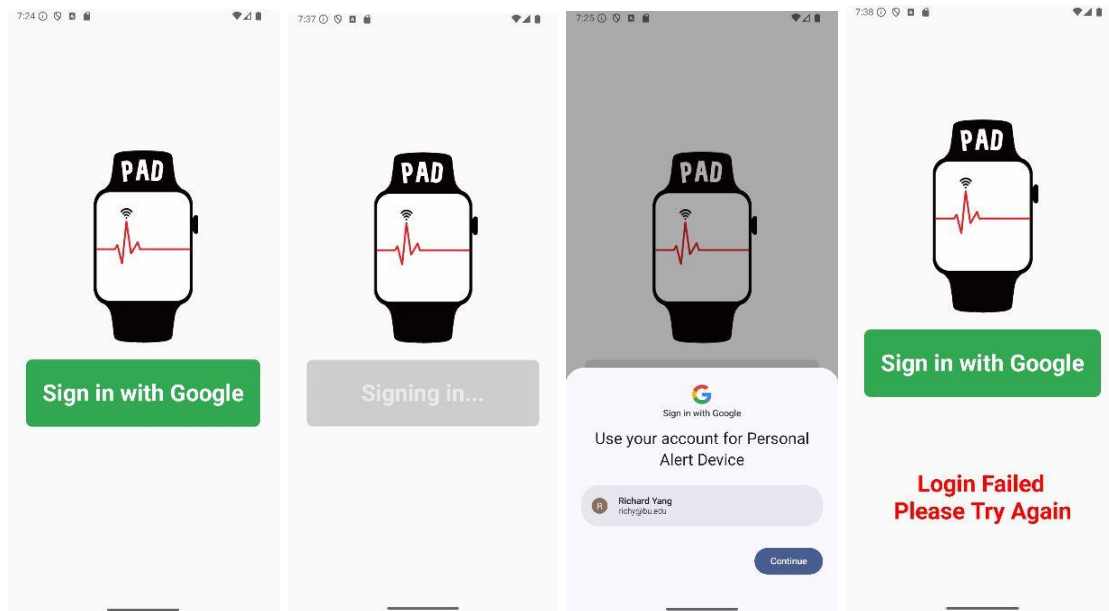


Figure 2.2.1: Four different “states” of the sign-in screen using Google Single sign-on from Firebase Authentication. Single sign-on automatically fetches Google accounts associated with the user’s phone, offering the simplest mode of authentication.

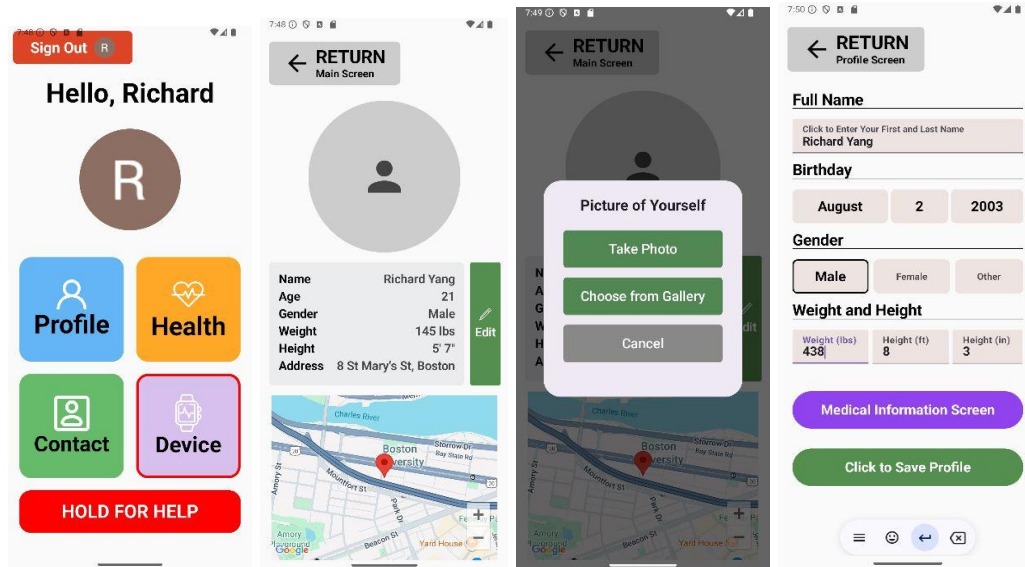


Figure 2.2.2 (left): Main screen: Users can navigate to all other screens, initiate a manual emergency response, see their connection status, and sign out.

Figure 2.2.3: Profile screen: Users can upload a new profile picture, see their personal information, edit their personal information, and see the automatically detected location.

Figure 2.2.4: Upon clicking the profile icon, users can select from multiple options to set a new profile picture.

Figure 2.2.5 (right): Edit Profile Screen: Upon clicking the edit icon in the profile screen, users can fill in their personal information fields. Dropdowns and buttons are used for ease of usability.

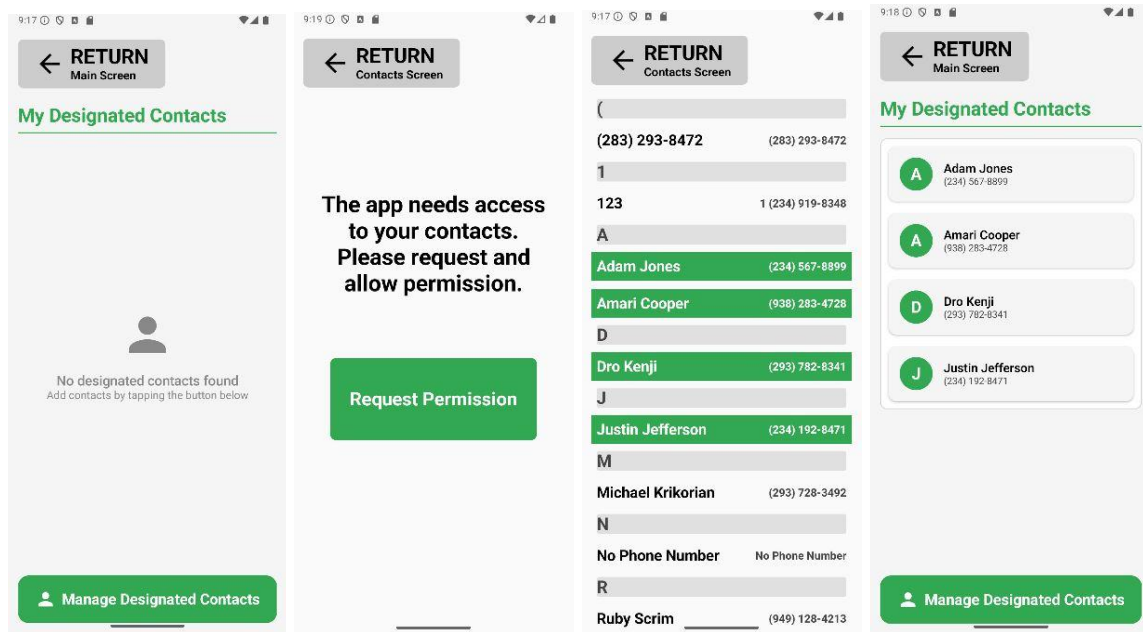


Figure 2.2.6: The Contacts screen shows a list of all the designated emergency contacts. The user can manage these contacts by pressing on the “Manage Designated Contacts” button and allowing access to the contacts permission. When allowed, the app will automatically fetch all contacts in the user’s phone where the user can simply press on a name to designate it. Once designated, the contacts will appear on the Contacts screen.

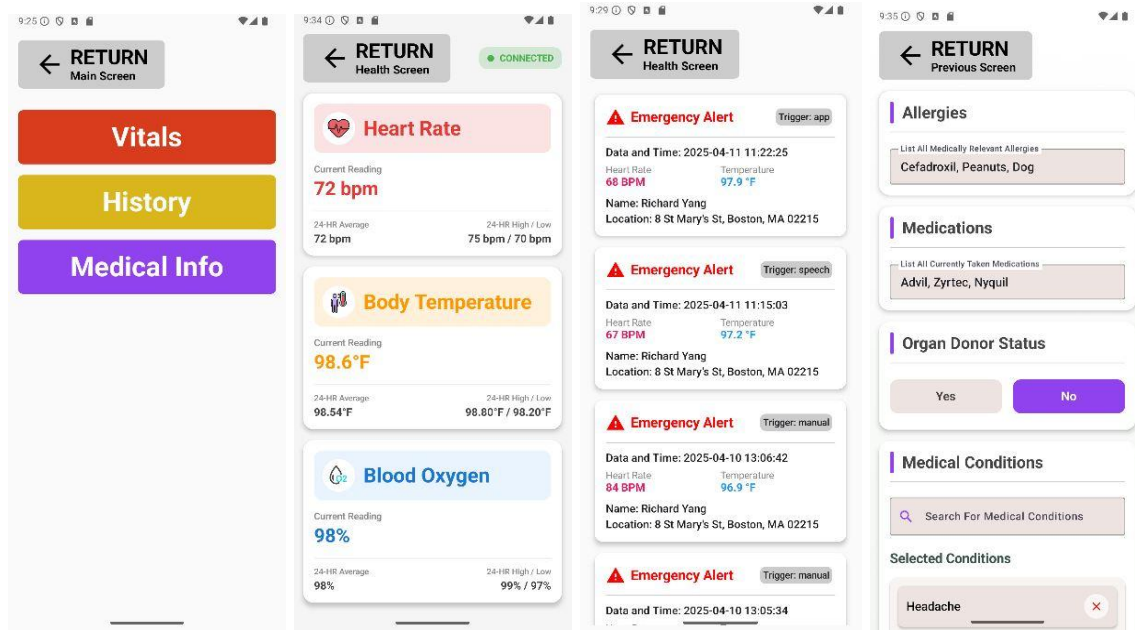


Figure 2.2.7 (left): Health screen: Users can navigate to other screens pertaining to health and emergency information.

Figure 2.2.8: Vitals screen: Displayed is a live reading and 24-hr average/high-low of the user's vitals gathered by the sensors on the wearable. Also shown is the connection status. If the device is disconnected from the phone, the values will show "Invalid".

Figure 2.2.9: History screen: Users will see a scrollable feed of their emergency response history with the associated information at the specific point of trigger. The trigger method is also shown.

Figure 2.2.10 (right): Medical Info screen: Users enter in any relevant medical information used to assist first responders in their emergency response. The search for medical conditions creates an autocomplete dropdown from the NIH's API for medical conditions. Users will see a list of all selected conditions in which they can remove them by pressing the "X" icon.

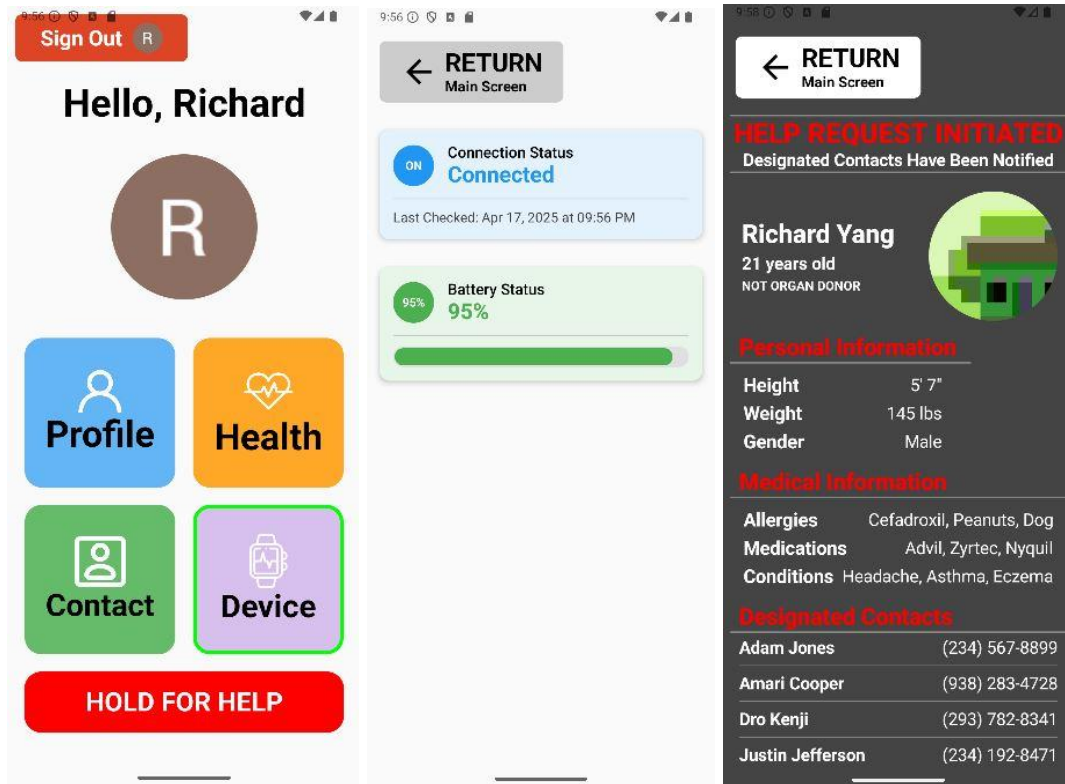


Figure 2.2.11 (left): Main screen with connected status.

Figure 2.2.12: Device screen: Shows the connection status and the timestamp in which it was last checked. Also shows the current battery level of the device.

Figure 2.2.13 (right): Help screen: Medical ID that contains all necessary information about the user for first responders in the case they recover the user's phone. This screen automatically opens when an emergency is requested via any trigger method.

Screenshots of user information and additional data stored in Firebase authentication and their unique Firestore document can be found in Appendix C.

2.3 Physical Description (Logan Lechuga)

There are two main parts for the system’s hardware: the enclosure of the wearable and the wireless charging stand. The wearable enclosure is made up of three separate pieces: the lid, the bottom, and the body. Below are part drawings and photos of the physical 3D-printed designs. The lid will not have its own drawing, as it is simply a flat piece that sits on top of the enclosure.

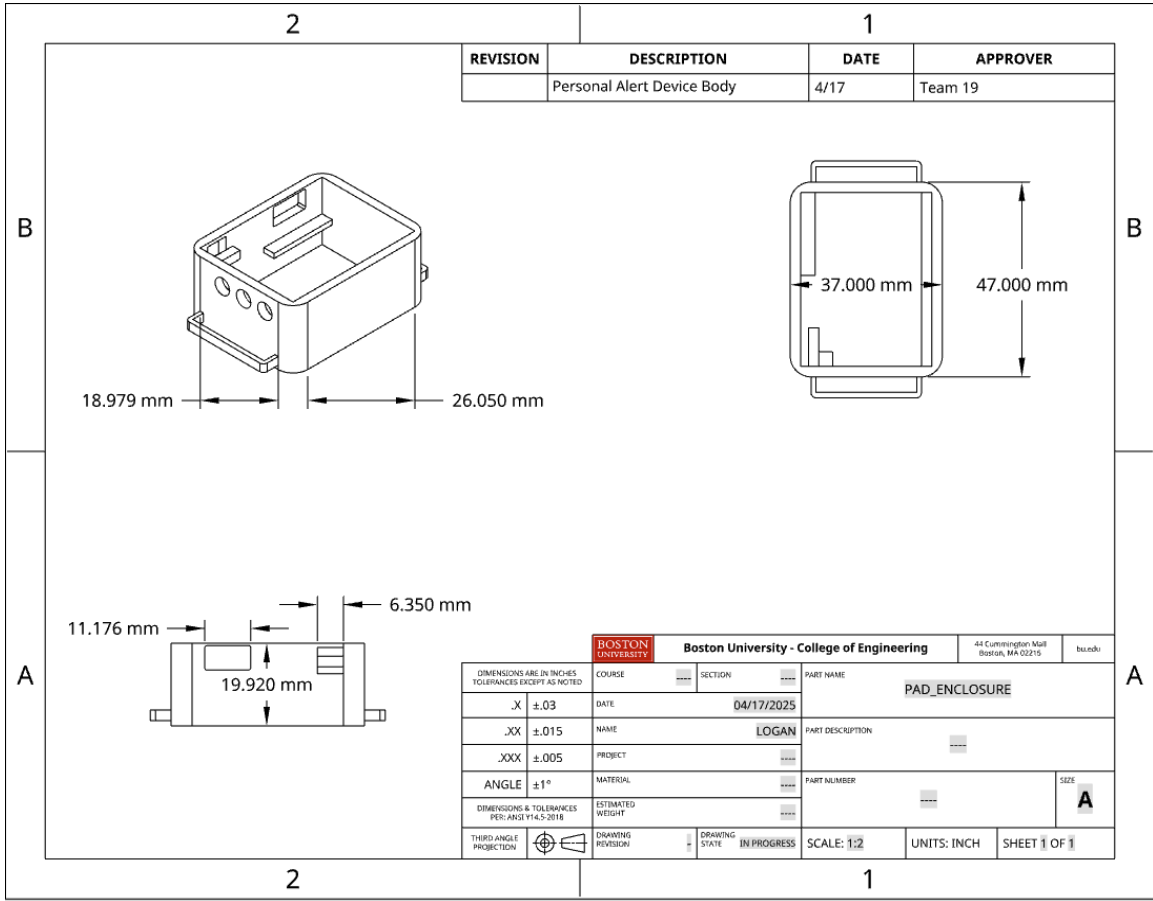


Figure 2.3.1: The image above depicts the body of the enclosure and includes notable measurements for the device in millimeters. This is the central part of the enclosure, as it houses all of the sensors, the PCB, and the device's battery. The holes in the enclosure are for the LEDs, USB-C, and a button.

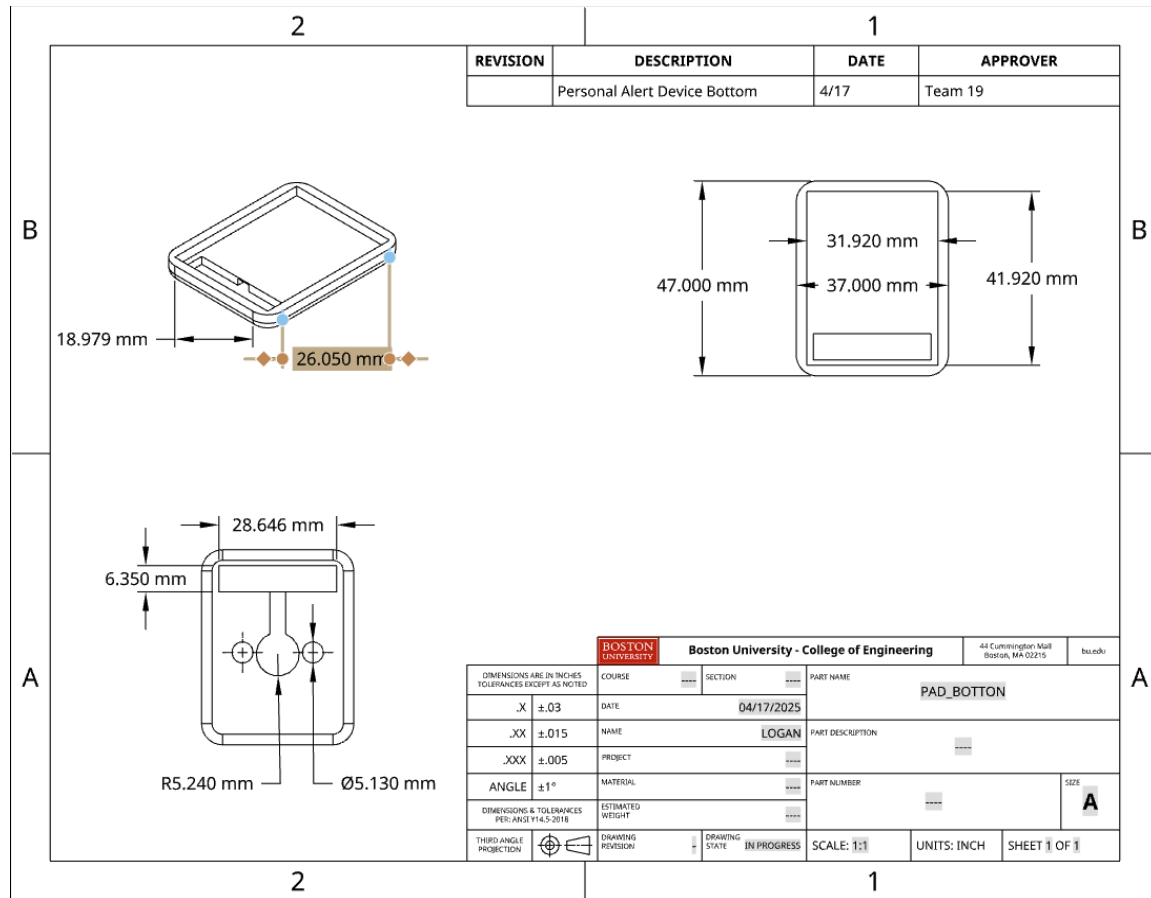


Figure 2.3.2: The image above depicts the bottom of the enclosure. There are holes included on the bottom for magnets and for the wireless charging coil. There is also a window cut out for wires to pass through, including the coil from the battery, the jumper cables for the heart rate sensor, and the thermistor. The length and width of the bottom also match the measurements above for the body.

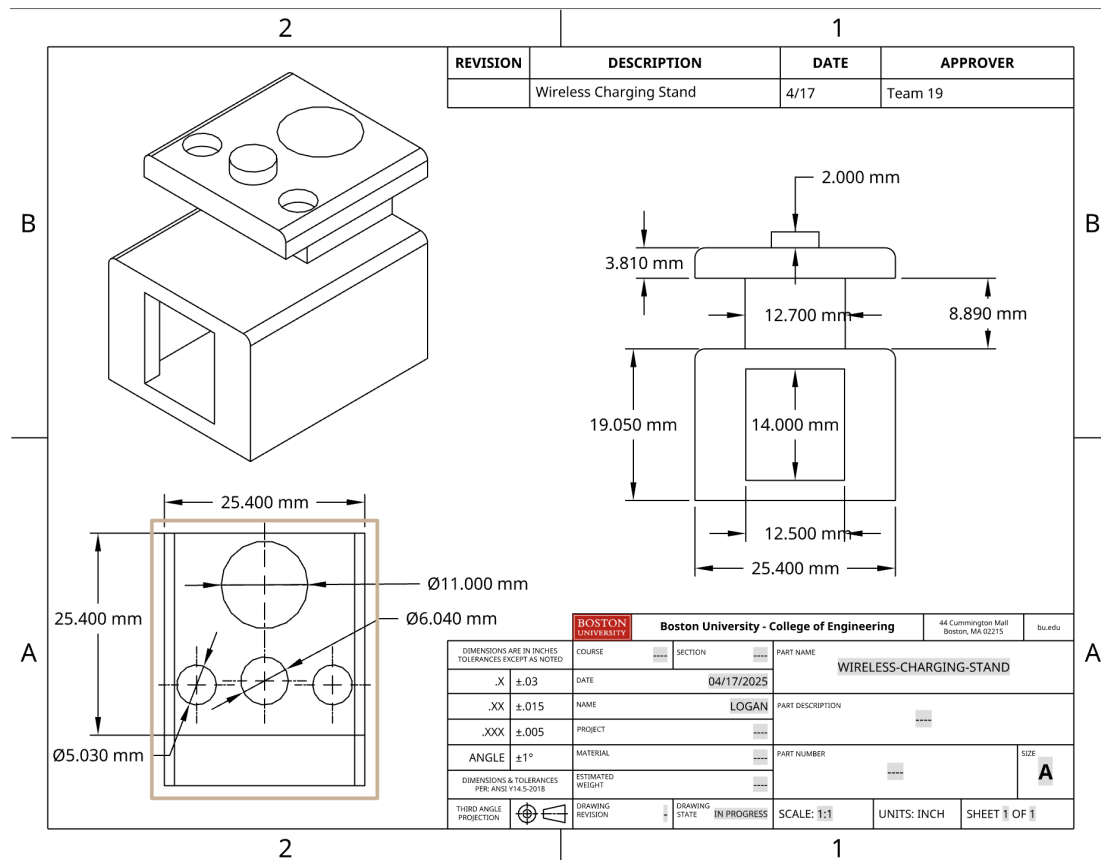


Figure 2.3.3: The image above shows the wireless charging stand used for our device. The bottom of the enclosure sits on top of the stand and uses magnets to ensure proper connection. The coil also sits on top of the charging stand to stay in place. The port connection is what's stored inside of the charging stand and also offers support to the stand.



Figure 2.3.4: Image of the latest model for the Personal Alert Device, which shows the LEDs, button, and USB-C cutouts. The wrist strap also houses the thermistor and pulse oximeter while maintaining a comfortable and secure fit for the user.

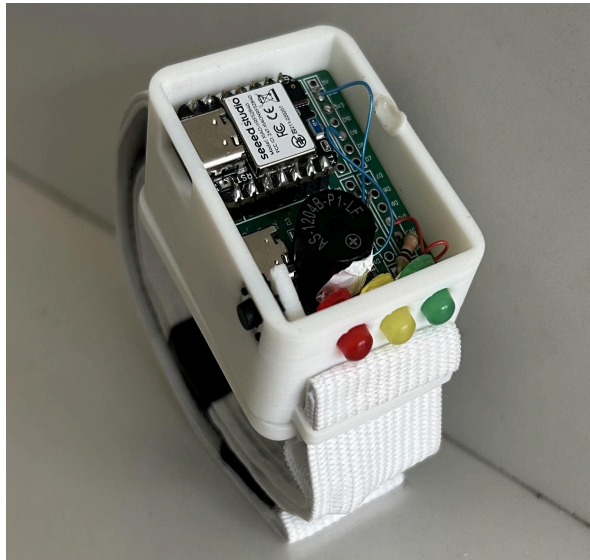


Figure 2.3.5: Another view of the enclosure with the exposed interior. The PCB, buzzer, LEDs, button and microcontroller are all housed within the device. The only components not pictured here are the battery, which sits under the PCB, and the pulse oximeter sensor and thermistor which are stored within the watch band.

2.4 Installation, Setup, and Support (Logan Lechuga)

The Personal Alert Device package will come with three core physical components: the wearable enclosure with a sensor-integrated wrist strap, the wireless charger (transmission module), and a charging cable (120VAC to 5VDC converter).

To begin, the user should place the wearable device on the wrist of their choice. When deciding how to orient the device, it is important to consider the location of the button on the enclosure, as it will be used periodically for various functions. It is recommended that the user position the device so that the button is easily accessible and comfortable to press with ease. Additionally, the wrist strap should be tightened enough so that the device feels stable and the sensors make adequate contact with the wrist.

After firmly securing the device, the user should check the battery level by pressing the button once. One of three LEDs will illuminate to indicate the battery status. If the green LED does not light up (less than 80%), it is recommended to charge the device before further use.

The wireless charging stance uses a USB-A to USB-B 2.0 cable, which must be connected to a suitable power source (wall outlet) before attempting to charge the device. Once connected, the user can simply place the wearable device onto the charging stand. It is important to note that the magnets will only align properly if the device and the charging stand are oriented correctly; the face of the device with the LEDs must align with the face of the charging stand with the USB-B port.

Moving to the software side, the user should download the Personal Alert Device app on a phone supporting Android 14 or above. The phone should have adequate service and Bluetooth capabilities. The user should connect to the device via Bluetooth and log in to the app via their Google account.

Once these steps are completed, the Personal Alert Device will be ready for use.

3 Operation of the Project (Tanveer Dhillon and Richard Yang)

3.1 *Operating Mode 1: Normal Operation*

Wearable Device (Richard Yang)

After the user has securely fastened the wrist strap in the correct orientation and ensuring adequate contact is made with the skin:

1. Pressing the button switch one time will turn on the battery level indicator. The green LED corresponds to a battery level of above 80%. The yellow LED corresponds to a battery level between 20% - 80%. The red LED corresponds to a battery level of under 20%.
2. When the pulse oximeter and thermistor have adequate contact with the user's wrist, the device will collect accurate and consistent heart rate, blood oxygen, and body temperature data.
3. To charge the Personal Alert Device, the base of the device is aligned with the top of the transmission module and snaps into place magnetically. The magnet orientation ensures that the receiving module is positioned correctly. Once the transmission and receiving module are aligned correctly, the device will charge wirelessly, taking from 4.5 to 5 hours to completely charge from empty.

Emergency Response Triggering (Tanveer Dhillon)

There are four trigger methods for an emergency response. Three are located on the wearable and one is located in the app.

1. Holding the button switch down for more than five seconds initiates a warning period in which the red LED blinks for ten seconds and the buzzer is activated. The user can cancel the emergency response within this warning period with a singular button press. If the user does not cancel the emergency response, the emergency response is started.
2. If the user says the keyword "Send Help" into the device twice, the speech recognition machine learning model on-board the wearable will classify the trigger and a warning period in which the red LED blinks for ten seconds and the buzzer is activated is initiated. The user can cancel the emergency response within this warning period with a singular button press. If the user does not cancel the emergency response, the emergency response is started.
3. If the device detects a fall due to acceleration and gyroscope thresholds being met within a five second interval, the device will enter a confirmation period for another five seconds in which additional movement surpassing a revised threshold will cancel the response. However, if the device detects immobility after five seconds, the emergency response is started.
4. If the user holds the "HOLD FOR HELP" button found on the bottom of the Main screen in the app until the indicator bar is completely filled, then the emergency response is triggered.

Emergency Response (Tanveer Dhillon)

1. The user's Medical ID (Help Screen, Fig. 2.2.13) will automatically open on the user's app from any current screen.
2. An entry in the user's emergency record in their Firestore collection will be appended with the means of trigger, timestamp, location, and vitals at the point of emergency trigger.
3. The History screen (Fig. 2.2.9) will be updated with the most recent emergency response with the information at the time of trigger.
4. SMS notifications will be sent to designated emergency contacts containing the user's name, location (and directions), means of trigger, timestamp, and vitals at the point of emergency trigger (Appendix D).

Mobile App Functionality (Richard Yang)

1. From the sign in screen (Fig. 2.2.1), users can authenticate themselves using Google Single sign on which automatically fetches Google accounts associated with their phone. Relevant information fathored by the authentication process, such as the user's name, profile picture, email, and unique user identifier as entered in their Google account are stored in their unique Firestore document. A new document is created for new users. If the authentication process is unsuccessful, the screen will indicate so.
2. From the Main screen (Fig 2.2.2), users can navigate to the other screens, sign out, or trigger an emergency response via the "HOLD FOR HELP" button. The connection status of the device and phone appears as the border of the "Device" button with red and green indicating disconnected and connected respectively. The user's name and profile picture fetched from their Google account are displayed on the screen.
3. From the Profile screen (2.2.3), users will see their updated personal information as well as their location as fetched from their phone's location permissions. The user can click on the circular profile icon to change their profile picture either from their camera or photo album. If not already granted, a pop-up will appear to allow photo permissions. An uploaded photo will replace the one displayed on the Main screen. The uploaded photo URL will be stored in Firestore.
4. Upon pressing the edit profile button, users will be navigated to the Edit profile screen (Fig. 2.2.4) where they can enter in their personal information and save it. The updated information will reflect on the Profile screen and in Firestore.
5. In the Contacts screen (Fig. 2.2.6), users will see a list of all of their designated emergency contacts. If no contacts are added, the screen will indicate so. Upon pressing the "Manage Designated Contacts" button, a prompt to allow contacts permissions will appear if previously not allowed. When allowed, a list of all contacts fetched from the user's phone will be displayed in which the user can select or remove emergency contacts. Any additions or removes will be reflected in both their Firestore document and on the Contacts screen.

6. Upon clicking on the “Health” button from the Main screen (Fig. 2.2.7), the user will have the option to navigate to the Vital screen, History screen, or Medical Info screen.
7. In the Vitals screen (Fig. 2.2.8), the user will see the live reading and 24-hr average/high and low of their vital signs (heart rate in BPM, blood oxygen in percentage, temperature in Fahrenheit) as collected by the wearable-based sensors. The screen will also indicate the connection status of the device, and indicate vitals as “Invalid” if disconnected. The vitals data is stored in the user’s Firestore collection and are fetched at the time of an emergency.
8. The History screen (2.2.9) displays updated entries of the user’s emergency record in chronological order. Each entry has an associated timestamp, means of trigger, vitals data, and location.
9. The Medical Info screen (2.2.10) allows the user to fill in any medical information relevant in assisting first responders in the event of an emergency. Users can select from an autocomplete drop down list of medical conditions from the NIH’s medical conditions API. The information persists within the fields of the screen and is saved to the user’s Firestore document upon exiting the screen.
10. When the device is connected to the phone, the Device screen (Fig 2.2.12) will indicate the last time the connection status was checked as well as the live battery level. If the device is disconnected, i.e., there has been no update from the device for over five seconds, the connection status will show disconnected and the battery level will read “Invalid”. The connection status and battery level are stored in Firestore.
11. The Help screen (Fig. 2.2.13) shows updated user personal information, medical information, and designated emergency contacts.
12. Updated information from any screen will persist across screen changes, refreshes, app closes, or memory clears.
13. All screens have a “Return” button in order to return to the previous screen.

Because an emergency can occur at any moment, there is currently no way to stop all operations. For all operations to remain functional, it is expected for the user to maintain a charged device, charged phone, and consistent connection between the device and phone.

3.2 Operating Mode 2: Abnormal Operations (Richard Yang and Tanveer Dhillon)

Hardware Errors

1. If the device is off, as indicated by either a dysfunctional LED indicator or IR LED on the pulse oximeter is off, the user should charge the battery for one hour and check again.
2. If charging the battery did not work, the user should plug a USB-C cable into the wearable and unplug. This may have been caused by the microcontroller entering a “sleep mode” only during periods of prolonged disconnection.

Connection Errors

1. If the app is displaying the device to be “Disconnected”, the user should check their Bluetooth and network settings on their phone. This status indicates a loss of communication between the device and the phone.
2. The user should check LightBlue and ensure all characteristics are subscribed to and connected to Adafruit IO.

Inaccurately Detected Emergency

1. If an emergency is falsely detected either by accidental triggering of the speech recognition, accidental holding of the button, or incorrect fall detection, the user should cancel the response within the given warning period.

Note: The sensitivity threshold of the speech classification inference and fall detection threshold can be adjusted as well as the warning period duration.

App-Based Errors

1. If the app is frequently crashing or fails to update correctly despite a “Connected” status, navigate to the app’s settings and clear cache/memory. This could be due to conflict with outdated API responses or Firestore rules.
2. If the app automatically denies permissions such as for contacts and photo access, manually allow permissions via the app’s settings.

3.3 Safety Issues (Tanveer Dhillon)

1. The Personal Alert Device requires connection to the user’s phone via Bluetooth and will not be able to respond to emergencies if the device is disconnected. The user is able to see their connection status in the app and will also be notified if the device is disconnected.
2. The Personal Alert Device is not guaranteed to detect all falls, and is only able to detect falls that exceed the set thresholds. Similarly, the speech recognition will only detect the keyword if the inference value is above the set threshold.
3. The Personal Alert Device uses a lithium polymer battery, so safe-handling and proper storage of the device is important. It must be charged with the charger included with the device. It is best practice to not leave the device sitting on the charger overnight; leaving the device to charge for extended periods of time could result in damage or deterioration to the battery.
4. The Personal Alert Device is not yet waterproof, so users must be sure to shield the device from any liquids it may come into contact with. If the device is exposed to liquids, it may cause a short-circuit and cease proper operations.

4 Technical Background (Richard Yang, Renad Alanazi, and Logan Lechuga)

4.1 *Hardware Components*

Wearable Enclosure

The wearable enclosure was designed with emphasis on a small form-factor, user comfortability, user ease of usage, and compact packaging of all components within one housing unit. A watch-style design was selected to meet these requirements effectively while bringing a sense of familiarity for the user. To create a lightweight and easily adaptable structure, the enclosure was 3D printed which allowed for rapid prototyping and precise adjustments to fit the internal components. 3D printing also provided flexibility in design iterations while optimizing comfort and secure fitment of the device worn on the wrist. The final enclosure measured 26 mm in height, 47 mm in length, and 37 mm in width, achieving a compact footprint while accommodating all other internal components. The 3D printing time for the complete wearable is around 1.5 hours.

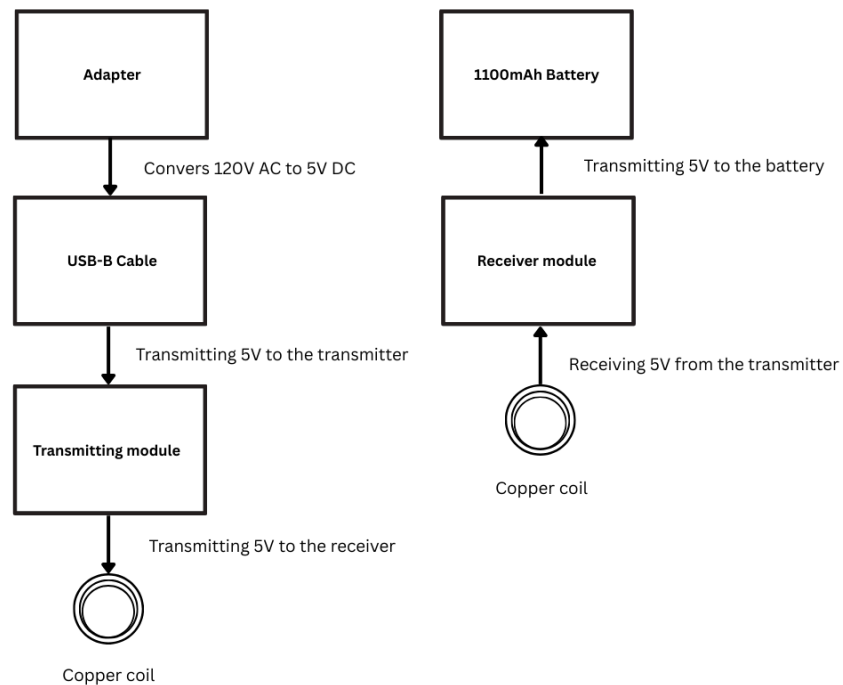
Wristband

One important feature to note is the sensor integrated wristband. The thermistor and pulse oximeter are housed within the watch's wristband. This approach was chosen to ensure that both sensors maintained adequate contact with the user's wrist for accurate vital readings. The wristband is designed to provide the tightest point of contact around the user's wrist at the location of the sensors while also maintaining a high level of comfortability. This approach minimized any external light, gaps, or movement that could affect the sensor readings. Additionally, the users are unable to feel any abnormal extrusions from the wristband caused by the integrated sensors. For the pulse oximeter, this means that a hole for the IR LED to reach the skin had to be cut in the wristband. The material chosen is an elastic nylon polymer which provides comfortability, flexibility, and breathability.

Power System

The wireless charging stand, developed after the wearable unit, is designed to complement the existing features of the wearable enclosure. The design is meant to be simple, portable, and was also 3D printed for the purpose of simplifying design iterations. The wearable unit sits on top of the stand and uses both a mechanical and magnetic fit to ensure the correct positioning of the transmitting and receiving module. The magnet fit also ensures that the two enclosures are held in place firmly throughout the charging process. As a result, the receiving module on the wearable was designed to be located on the base of the device while the transmitting module on the charging stand is on the top. The charging stand's dimensions are as follows: 31.75 mm in height, 25.4 mm in width, and. The total 3D printing time for the charging enclosure was 39 minutes. Overall, wireless charging was chosen to provide the easiest means of charging and intuition for

the user. Also in the power system, a 3.7V 1100mAh LiPo battery was chosen due to its larger capacity and small size.



Figure

4.1.1: Circuit diagram for the power system

Custom PCB

A custom PCB was designed to serve as the central hub for all of the electrical components of the wearable device. The PCB allowed the microcontroller to interface with components such as the LEDs, buzzer, and button without the need for extraneous wiring. By consolidating these components onto a single board, the design minimizes size while also reducing potential points of failure associated with wired connections.

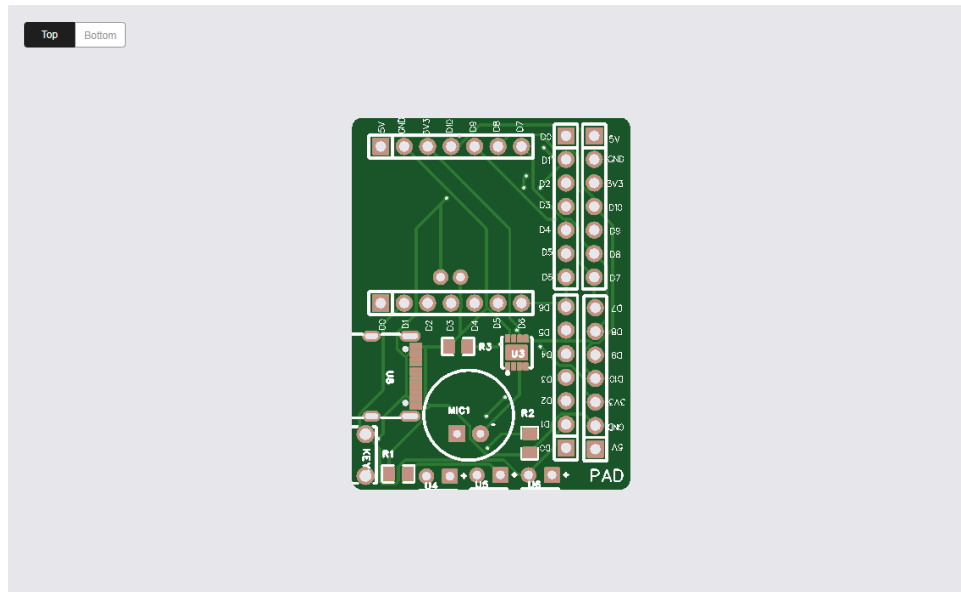


Figure 4.1.2: Top view of PCB received by JLCPCB

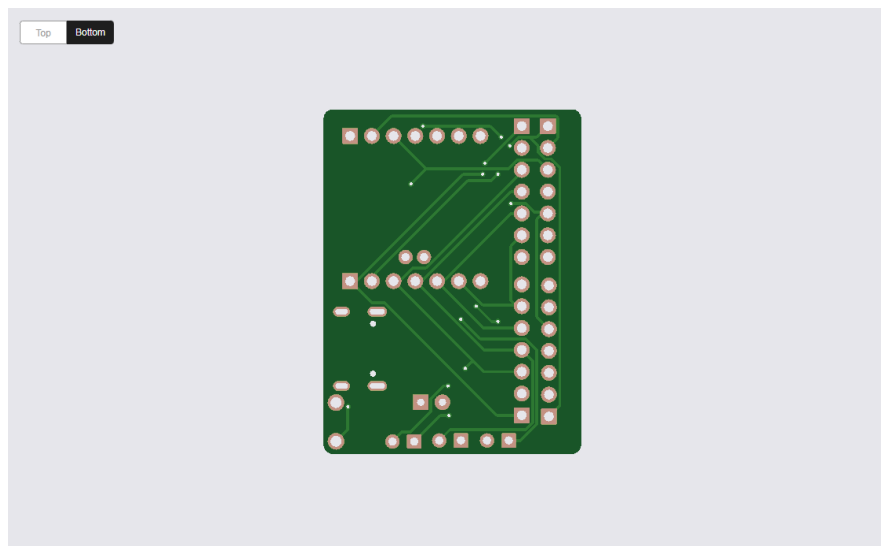


Figure 4.1.3: Bottom view of PCB received by JLCPCB

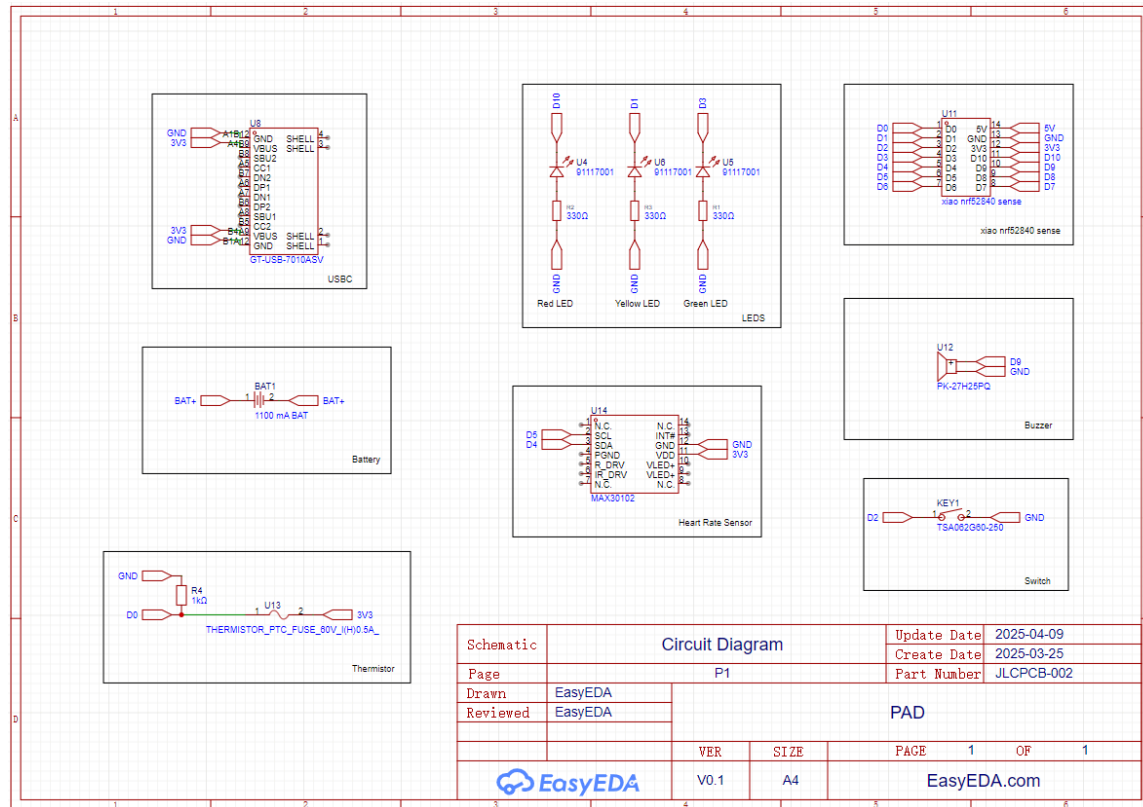


Figure 4.1.4: Complete electric schematic of the Personal Alert Device, designed using EasyEDA

4.2 Software Components

Communications

The wearable uses Bluetooth Low Energy to establish a wireless connection with the mobile application for real-time data transmission. BLE was selected for its low power consumption and suitability for short to mid range connectivity. Additionally, maximizing battery life of the wearable device is critical. The device advertises sensor and emergency characteristics over Bluetooth which are fed to corresponding Adafruit IO feeds. Adafruit IO was chosen due to its integration with LightBlue, the BLE scanner, and its prominence as an IoT platform designed for real-time data visualization and remote monitoring. Other communications, such as from the user's phone to the cloud are done through Firebase SDKs, which abstracts the use of REST APIs, and provide reliable user authentication, database interactions, and access to other Google APIs.

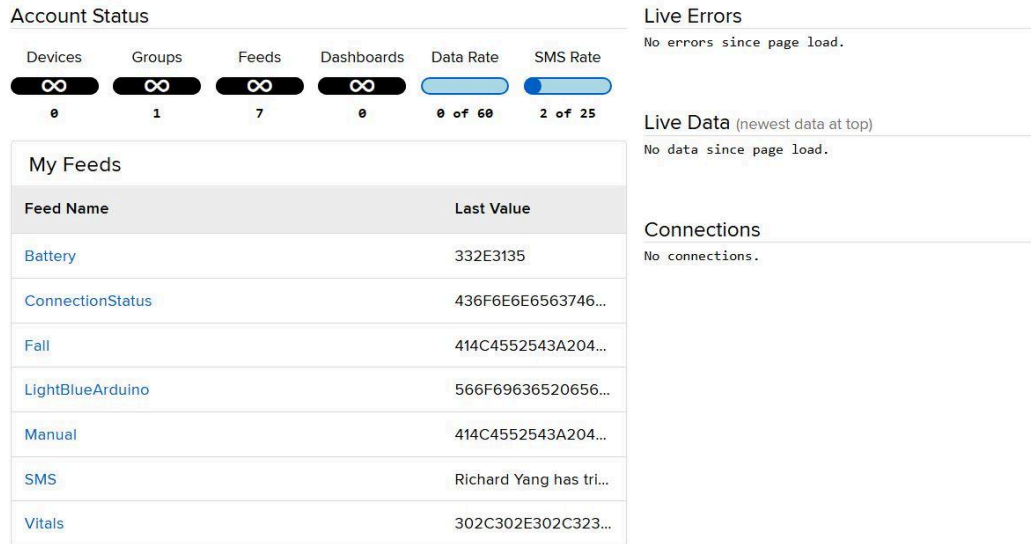


Figure 4.2.1: Overview screen as seen in Adafruit IO displaying all feeds associated with the BLE characteristics.

```

13:55:26.251 - Connected to nearby peripheral: XIAO nrf52840 Sense
13:55:28.650 - 47 peripherals discovered
13:55:52.667 - 48 peripherals discovered
13:56:06.118 - Stopping search for nearby peripherals
13:56:15.506 - Characteristic [00000000-0000-0000-0000-00000000E0C] notified: <414c4552 543a2048 454c5020 52455155 45535420 44455445 43544544 21>
13:56:18.461 - Characteristic [00000000-0000-0000-0000-00000000E0C] notified: <414c4552 543a2048 454c5020 52455155 45535420 44455445 43544544 21>
13:56:22.810 - Characteristic [00000000-0000-0000-0000-00000000E0C] notified: <414c4552 543a2048 454c5020 52455155 45535420 44455445 43544544 21>
13:56:49.572 - Characteristic [00000000-0000-0000-0000-00000000E0C] notified: <414c4552 543a2048 454c5020 52455155 45535420 44455445 43544544 21>
13:56:52.632 - Characteristic [00000000-0000-0000-0000-00000000E0C] notified: <414c4552 543a2048 454c5020 52455155 45535420 44455445 43544544 21>
13:57:07.961 - Starting search for nearby peripherals
13:57:07.961 - Scanning on Main Thread
13:57:08.734 - Stopping search for nearby peripherals
13:57:15.395 - Starting search for nearby peripherals
13:57:15.395 - Scanning on Main Thread

```

Figure 4.2.2: Sample subscribed characteristic notifications seen via LightBlue.

For emergency notifications sent to designated emergency contacts, SMS was chosen due to it provides a fast, reliable, scalable, and universally supported communication channel. SMS can reach designated contacts instantly on any mobile device, even with limited coverage. While automated calls have not yet been implemented, they would require a third-party call server to place dynamic autonomous calls containing all of the emergency information. Instead, using SMS, emergency notifications can be sent out simultaneously and contain all necessary information for the emergency contact while also providing them with a record of the event. Additionally, SMS operates independently of smartphone platforms, making it a robust solution for emergency notifications.

Mobile Application and Cloud

The mobile application for the Personal Alert Device was developed using Kotlin and Jetpack Compose in Android Studio. Jetpack Compose is Android's toolkit for declarative UI building and allows for highly responsive user interfaces as well as dynamic updating used to display incoming sensor data or update user information fields. The application uses the Android Permissions API to request and manage runtime permissions for features such as location access, contacts access, or photo/camera access.

Communication with the user's Firestore is handled through asynchronous API calls provided by the Firebase SDK, which allows the app to securely read and write user information to a document specific to the user's ID number gathered by Google SSO. Firebase was chosen because it provides a comprehensive suite of cloud services that are both secure and scalable. Additionally, it is highly integrated with Android development and provides the necessary features such as authentication and data storage necessary for the application.

Firestore was chosen as the database solution because of its heavy integration with Android and Firebase, real-time synchronization, scalability, and flexible data structures. From Google SSO, a unique Google ID number associated with the authenticated user is used to label a new document in the Firestore. This provides the user with data security and dynamic app support for multiple users. Firestore also provides offline data persistence which is critical in maintaining data integrity within the user's app.

5 Relevant Engineering Standards (Renad Alanzi, Tanveer Dhillon, and Logan Lechuga)

Software

1. The NIH's API used to generate a drop-down list of medical conditions utilizes REST API standards. All the data fetched from the NIH's API was stored in JSON format and contains proper documentation, so we can correctly categorize the information on the backend. These API calls also used standard HTTP methods, such as GET and POST, for retrieving and storing data in our database.
2. For user authentication, Google SSO was used which uses SAML (Security Assertion Markup Language) 2.0 Post Binding. SAML is an open standard for exchanging authentication and authorization data between a SAML IdP and SAML service providers, in this case Google. When a user signs in, the client SDK handles the authentication handshake and returns ID tokens that contain the SAML attributes. If a user has not been authenticated yet, they will be redirected to a Google sign-in page where they can enter their email. If SSO is enabled, the browser redirects the browser to the IdP with the RelayState and SAMLRequest. The IdP receives the request and authenticates the user. After the user is authenticated, the IdP sends a SAML Response and assertion, and the user is granted access.
3. Firebase SDKs are used to connect the Android application to Firestore over HTTPS. It also manages authentication with Google Sign-In. As noted above, Google Sign-In was used to enable quick and easy registration for the application. Once the user was logged in, this information was stored in our Firebase database. The login information was used as a reference for the rest of the user's personal information and helped in storing real-time user updates, such as emergencies and data from individual sensors.
4. The ISO 8601 is the standard used for representing dates and times being stored in Firestore. Using this standard allows for the data in Firestore to be presented in a standardized format.

Communication

1. Bluetooth Low Energy (BLE) is used to communicate between the device and the Android app and transfer sensor data from the device to the app. BLE is also designed for low power consumption.

Electrical Design

1. The National Electrical Code was followed to ensure that the electrical system of the Personal Alert Device was safely designed. We ensured that all components were supplied the correct voltage and properly grounded.

Governmental

1. As the device stores the user's medical data and is used for detecting health related emergencies, it complies with the Health Insurance Portability and Accountability Act (HIPAA) as the user's data is securely stored in Firestore and only shared with designated contacts.

Hardware

1. Regarding the standards for the hardware, the enclosure was designed using Onshape, following industry-standard design practices. This includes maintaining proper schematics of the enclosure with measurements and version control, as we worked on multiple models of the enclosure and constantly improved on previous designs.
2. The rechargeable lithium battery used in the PAD device aligns with both IEC 62133 and UN 38.3 standards. The 3.7V 1100mAh battery used includes an internal protection circuit to prevent overcharging and discharging and short circuit conditions.
3. The wireless charging system aligns with the Qi wireless charging standard. It operates using resonant inductive coupling with copper coils and delivers a 5V to the receiver module. The system was tested to ensure that the receiver module did not exceed the safe temperature and current thresholds which aligns with IEEE 802.15.1 BLE operation guidelines to minimize interference with power transfer.

6 Cost Breakdown (Richard Yang)

Project Costs for Production of Beta Version (Next Unit after Prototype)				
Item	Quantity	Description	Unit Cost	Extended Cost
1	1	XIAO nRF 52840 Sense Microcontroller	\$22.68	\$22.68
2	1	MAX30102 Pulse Oximeter	\$2.56	\$2.56
3	1	B57703M0103A018 Thermistor	\$4.22	\$4.22
4	1	Custom EasyEDA PCB	\$3.01	\$3.01
5	1	3D Printing Filament	\$15.00	\$15.00
7	1	3.7V 1100 mAh LiPo Battery	\$8.99	\$8.99
8	1	Power System (Receiving/Transmitting Module, Voltage Converter)	\$15.99	\$15.99
9	1	Nylon Polymer Wristband	\$5.00	\$5.00
Beta Version-Total Cost				\$77.45

It is important to note that both the manufacturing costs and software subscription costs of the Personal Alert Device stand to benefit from economies of scale. The software costs are usage-based and thus, not included. The Adafruit IO subscription costs \$10 per month, while the Firestore subscription is a “pay as you go” model, at around \$0.01 for every 300,000 document writes.

To offset these costs, an additional subscription model may be introduced or the base price of the unit can be increased. Nonetheless, our product offers a highly competitive pricing model compared to current industry solutions, while also providing more advanced technologies and reliability.

Furthermore, it can be expected that manufacturing costs will decrease once a fully integrated PCB has been developed. This would significantly offset the cost of the microcontroller by integrating only the processor, PDM, IMU, and other components directly into the PCB.

7 Appendices

7.1 Appendix A - Specifications (Renad Alanazi)

Specification	Measured Value	Notes
Heart Rate Detection Range	50 – 180 bpm	Measured using MAX30102 sensor
Body Temperature Detection Range	35°C – 42°C	Using B57703M0103A018 thermistor
Fall Detection Response Time	10 seconds	Using onboard IMU (6-axis)
Emergency Button Activation Time	5000 ms	Push-button switch
Battery Capacity	1100 mAh	3.7V LiPo rechargeable battery
Battery Life	~72 hours	
Wireless Charging Voltage	5V input, ~4.2V output	Via copper coil and receiver module
Charging Time	~4 hours	Charging current 300 mA
Device Dimensions (L x W x H)	47 mm x 37 mm x 26 mm	3D printed enclosure
Device Weight	58g	Total assembled, including battery
Microcontroller	Seeed Studio XIAO nRF52840 Sense	BLE + PDM + IMU + low-power MCU
Alert Sound Output	~85 dB @ 2.2kHz	Using a buzzer
Wireless Communication	Bluetooth	For app notifications and data upload
Data Upload Platform	Google Firestore	Stores emergency alerts, sensor data, user data
Speech Recognition Machine Learning Model	94.6% accuracy	1000ms window, +500ms sliding window, MFCC, 1D Conv. NN,
Maximum System Latency	Worst case 600 ms	Total communications delay

7.2 Appendix B – Team Information

Richard Yang

- Senior at BU studying Computer Engineering
- Email: richy@bu.edu
- Phone Number: 630-779-8950
- Post Graduation: Incoming Controls Engineer at Haumiller Engineering

Logan Lechuga

- Senior at BU studying Computer Engineering with a concentration in Machine Learning
- Email: llechuga@bu.edu
- Phone Number: 832-693-5306
- Post Graduation: Seeking a full-time role as a software engineer after December 2025.

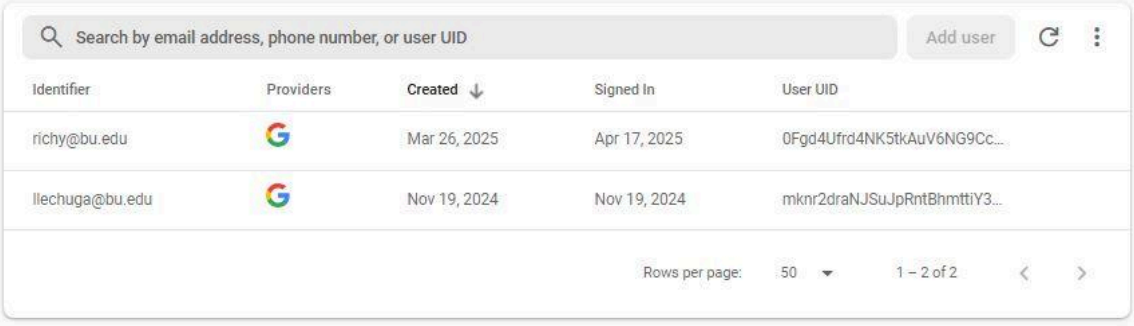
Tanveer Dhilon

- Senior at BU studying Computer Engineering with a concentration in Machine Learning
- Email: tdhilon@bu.edu
- Phone Number: 347-476-5030

Renad Alanazi

- Senior at BU studying Electrical Engineering
- Email: reenad@bu.edu
- Phone Number: 617-899-6452

7.3 Appendix C – Firebase (Authentication and Firestore)



The screenshot shows the Firebase Authentication console interface. At the top, there is a search bar labeled 'Search by email address, phone number, or user UID' and buttons for 'Add user', a refresh icon, and a menu icon. Below the search bar is a table with the following columns: Identifier, Providers, Created, Signed In, and User UID. The table contains two rows of user data. At the bottom right, there is a 'Rows per page' dropdown set to 50 and a pagination indicator '1 - 2 of 2' with navigation arrows.



Identifier	Providers	Created	Signed In	User UID
richy@bu.edu		Mar 26, 2025	Apr 17, 2025	0Fgd4Ufrd4NK5tkAuV6NG9Cc...
llechuga@bu.edu		Nov 19, 2024	Nov 19, 2024	mknr2draNJSuJpRntBhmmttY3...

Figure 7.3.1: Authenticated users with their day of creation, last signed in data, and unique user identifier as seen in Firebase authentication.

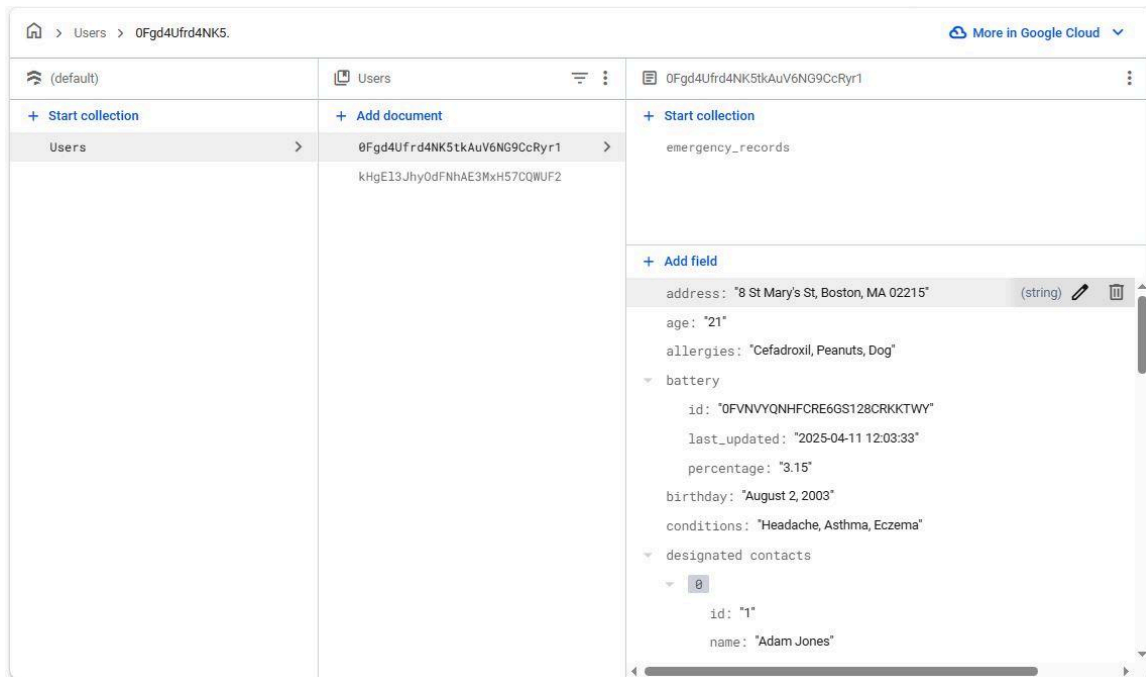


Figure 7.3.2: User's unique Firestore document with associated data fields.

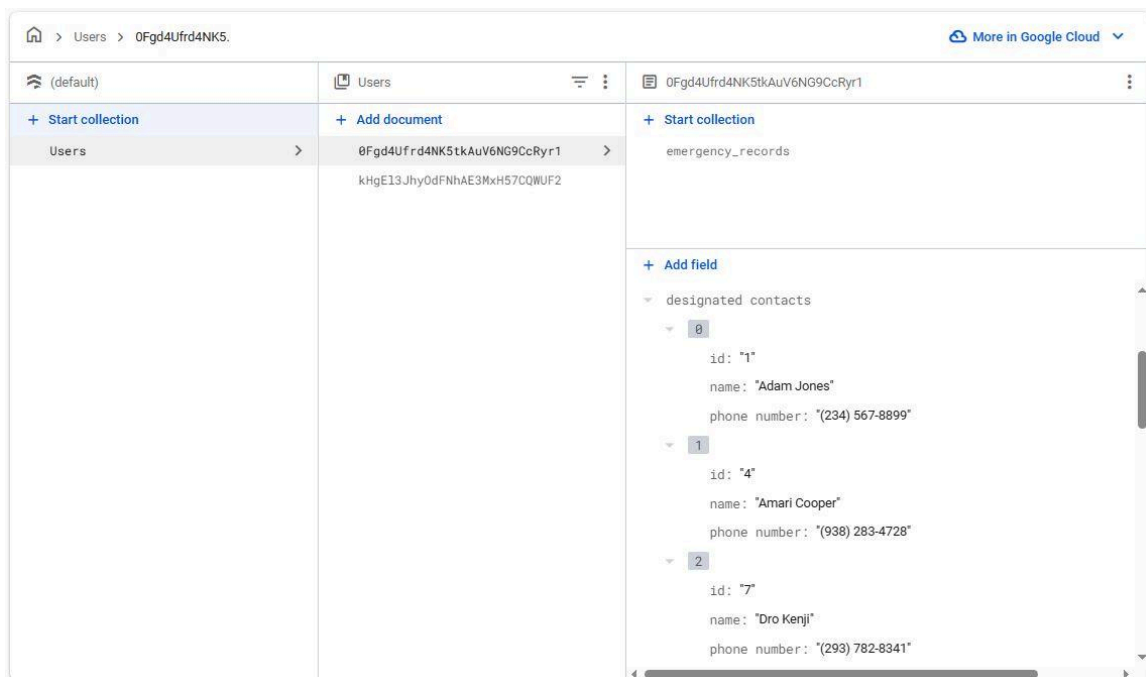


Figure 7.3.3: Emergency designated contacts stored in unique user collection.

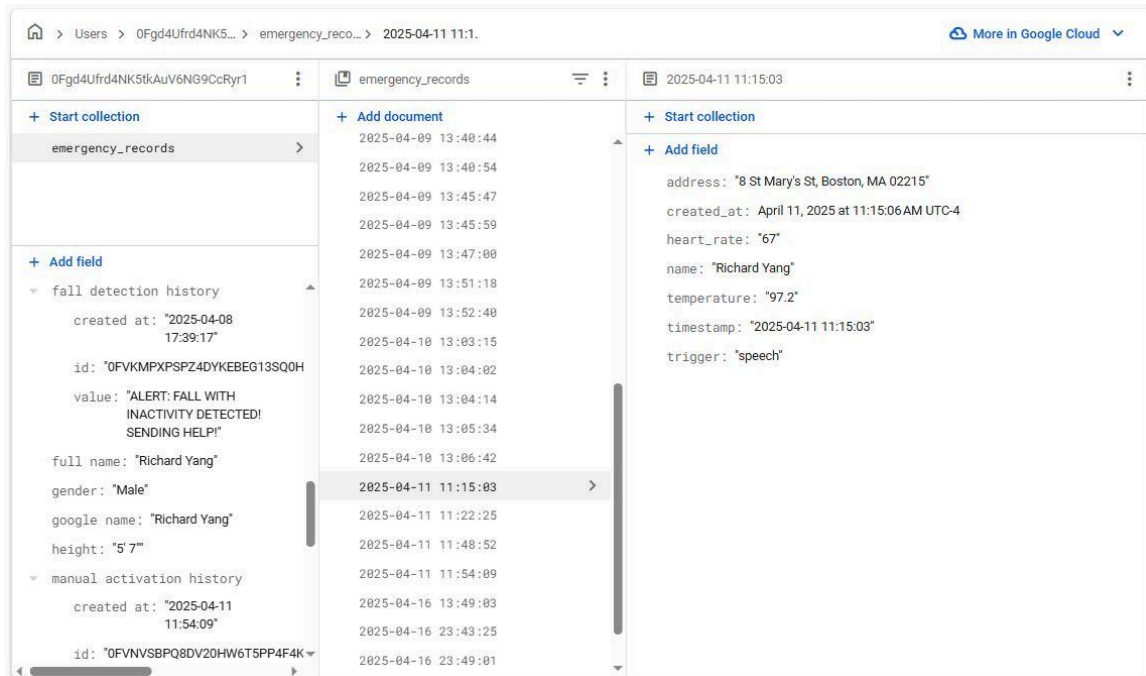


Figure 7.3.4: User's emergency records stored in their unique document.

7.4 Appendix D – SMS to Designated Contacts

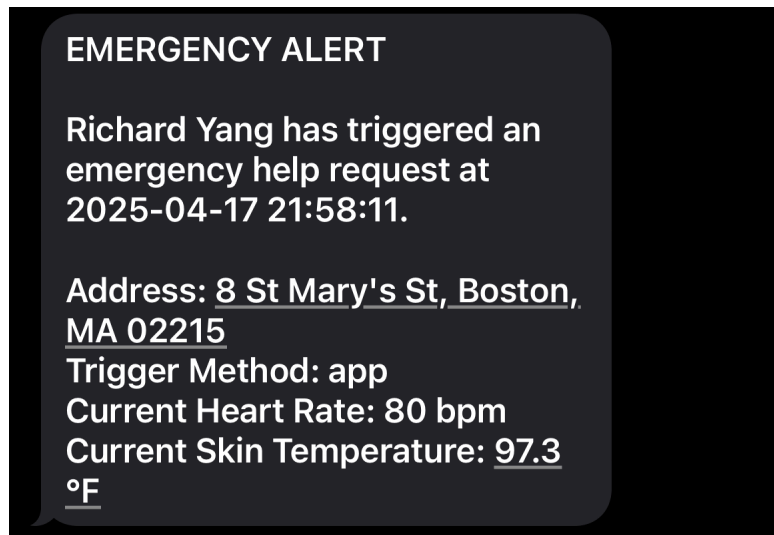


Figure 7.4.1: Emergency SMS notification format with clickable directions.