Boston University

Electrical & Computer Engineering

EC463 Capstone Senior Design Project

Prototype Test Report

Personal Alert Device

November 22, 2024

by

Team 19

PAD Group

Richard Yang richy@bu.edu

Tanveer Dhilon tdhilon@bu.edu

Logan Lechuga llechuga@bu.edu

Renad Alanazi reenad@bu.edu

1 Required Materials Summary

For the hardware portion of the Personal Alert Device, the required equipment included a Raspberry Pi Pico, MAX30100 Heart-Rate Sensor, MLX90614 Contactless Infrared Temperature Sensor, ADXL345 3-Axis Accelerometer, GT-U7 GPS Module, housing unit, jumper cables, and micro USB cable to power the microcontroller. The goal with these components was to demonstrate necessary sensor functionalities within the wearable. While considering the cost-effectiveness of our components, we were able to validate core functionalities such as heart rate sensing, fall detection, temperature sensing, and GPS tracking. This approach ensured we could test the feasibility of the system without exceeding budget constraints, while still providing the foundation for future upgrades with more advanced and reliable components.

The equipment necessary for the software portion of the prototype was a laptop with internet access. Using this laptop, we were able to access Android Studio which housed an emulator for our native application. Additionally, we were able to access Google Firebase where we saw authenticated users and both user and sensor data stored in the Firestore database. Additionally, the laptop provided power to the Raspberry Pi Pico microcontroller via USB to USB-C.

Overall, our equipment was consistent with our prototype's test plan. All sensors were powered successfully, and recorded data. Additionally, the emulation of the native application was run successfully.

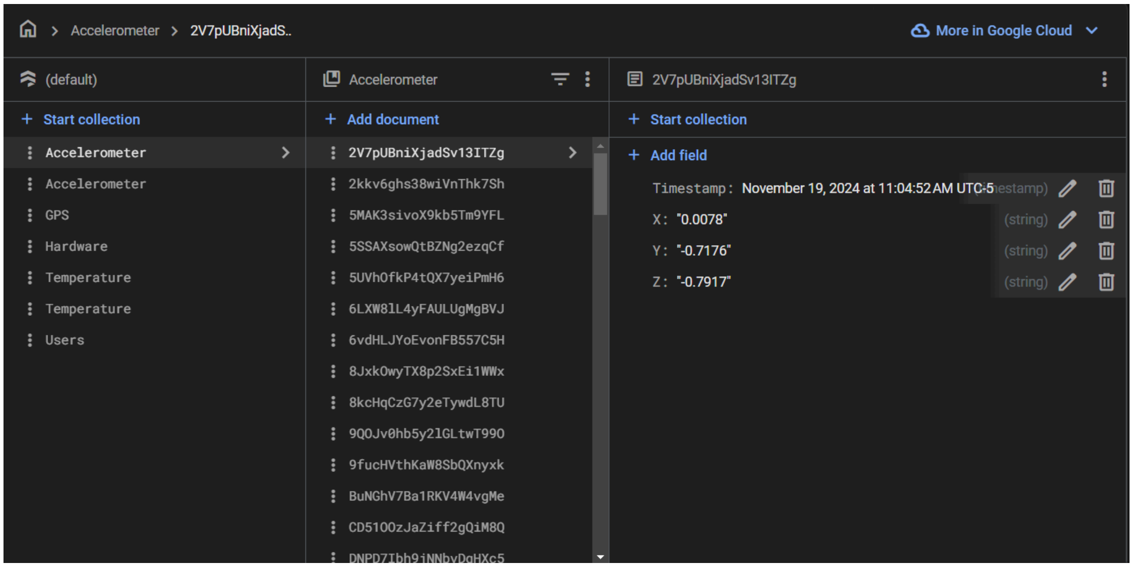
2 Setup Summary

The setup was divided between the hardware and software aspects of the Personal Alert Device. The hardware portion of our setup required the functionality of various sensors, which aided us in determining emergencies related to the user’s health. The software portion, an Android native application, serves as an interface between the user and the device, gathering important information dynamically. For our actual setup, we only included data from our heart rate sensor, the accelerometer, the temperature sensor, and the GPS module. We ran the sensor test code via Thonny, which displayed our recordings for each sensor. Once the sensors recorded data, the data was uploaded to a Firestore collection, which provided our database with observable real-time updates. On the project's software side, we ran the app on Android Studio and then signed into the app using various Google accounts. Once signed in, we then navigated through the multiple pages within the app, such as the profile and contacts page, to show the individual page’s purpose and functionality and smooth navigation throughout the app. In the contacts page, we could add emergency contacts from the user’s contacts list, which would then be uploaded to Firebase, which we could observe in real time. In the profile page, we added the user's age, height, weight, and gender which were also visible in Firebase. Finally, we simply showed that the data stored in Firebase was not deleted if a user signed out of their account and was not overwritten when we signed into the same account again.

3 Measurements Taken

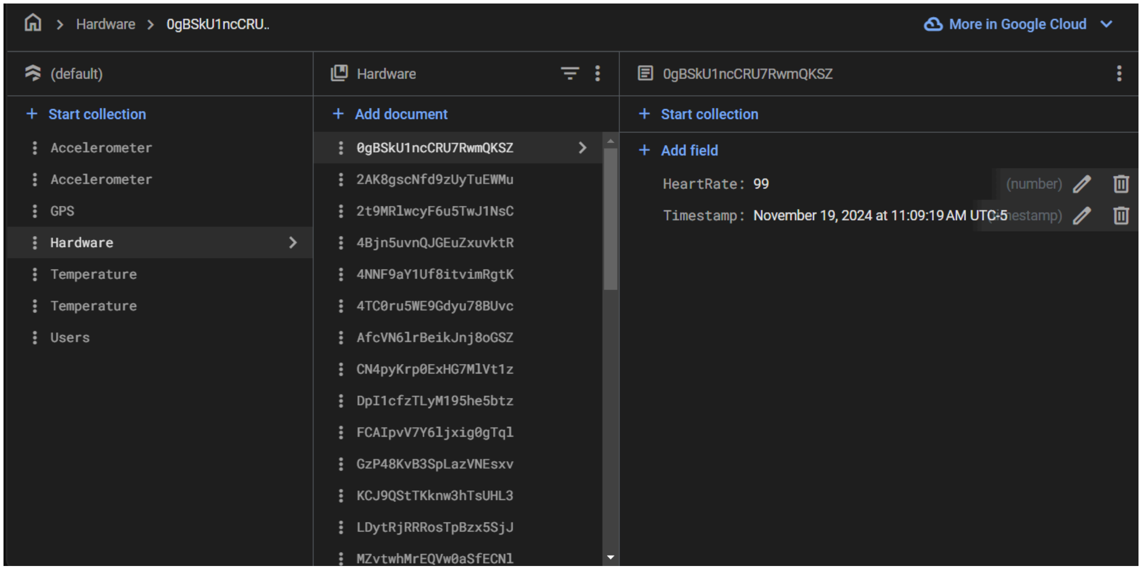
The measurements taken for the prototype test include acceleration data from the ADXL345 in the x, y, and z directions, heart rate data from the MAX30100 heart rate sensor in BPM, body temperature data from the MLX90614 contactless infrared temperature sensor in degrees Celsius, and location data from the GT-U7 GPS module. The data collected from these sensors was then stored in Firestore. Although data was obtained from all sensors, the GPS module was unable to obtain location indoors resulting in empty Firestore entries. In addition to this, the temperature sensor was not able to obtain accurate body temperature, but rather read values closer to room temperature.

For the mobile application, we measured the success of various application functionalities. Such functionalities included: successful Google Single Sign-On, user data stored in unique Firestore collection, user added to Firebase Authentication, screen navigation using buttons, user name and profile picture displayed on the home page, request permission to access contacts, adding designated emergency contacts, contacts stored in respective Firestore collection, logging out, and having user data persist throughout application refresh, close, or sign out. All these tasks were successfully performed during testing.



```
Accelerometer      2V7pUBniXjadSv13ITZg     Timestamp: November 19, 2024 at 11:04:52 AM UTC-5 (timestamp)
Accelerometer      2kkv6ghs38wiVnThk7Sh     X: "0.0078"  (string)
GPS                5MAK3sivoX9kb5Tm9YFL     Y: "-0.7176" (string)
Hardware           5SSAXsowQtBZNg2ezqCf     Z: "-0.7917" (string)
Temperature        5UVhOfkP4tQX7yeiPmH6
Temperature        6LXW8lL4yFAULUgMgBVJ
Users              6vdHLJYoEvonFB557C5H
                   8JxkOwyTX8p2SxEi1WWx
                   8kcHqCzG7y2eTywdL8TU
                   9QOJv0hb5y21GLtwT990
                   9fucHVthKaW8SbQXnyxk
                   BuNGhV7Ba1RKV4W4vgMe
                   CD51OOzJaZiff2gQiM8Q
                   DNPD7Ibh9iNNbyDqHXc5
```

Fig 1. Accelerometer data stored in Firestore Database



(default)

- Accelerometer
- Accelerometer
- GPS
- Hardware
- Temperature
- Temperature
- Users

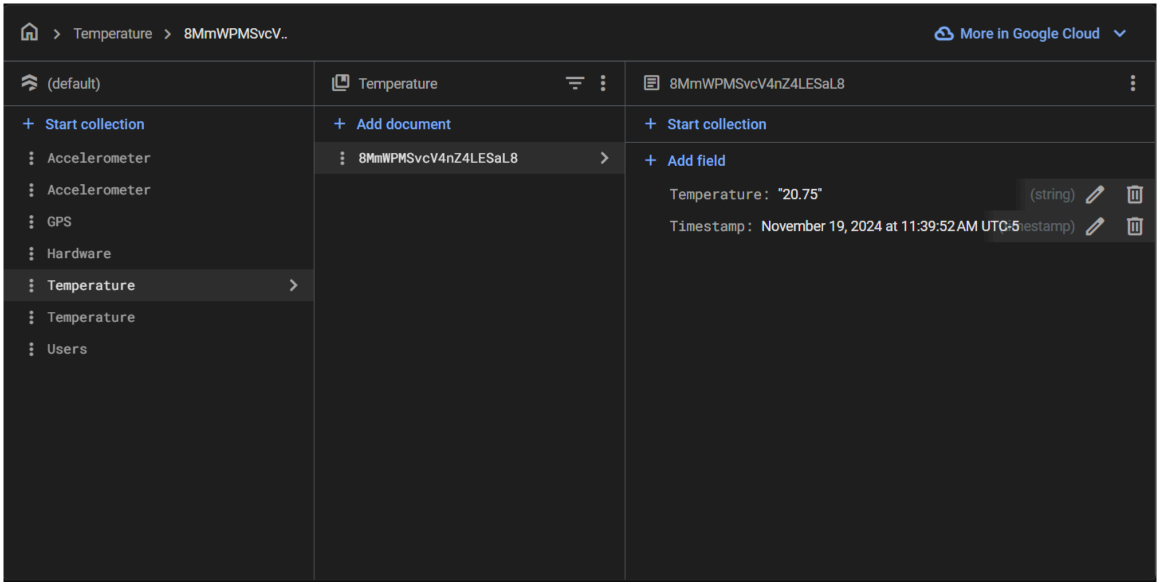
Hardware

- 0gBSkU1ncCRU7RwmQKSZ
- 2AK8gscNfd9zUyTuEWMu
- 2t9MRlwcyF6u5TwJ1NsC
- 4Bjn5uvnQJGEuZxuvktR
- 4NNF9aY1Uf8itvimRgtK
- 4TC0ru5WE9Gdyu78BUvc
- AfcVN6lrBeikJnj8oGSZ
- CN4pyKrp0ExHG7MlVt1z
- DpI1cfzTLyM195he5btz
- FCAIpvV7Y6ljxig0gTql
- GzP48KvB3SpLazVNEsxv
- KCJ9QStTKknw3hTsUHL3
- LDytRjRRRosTpBzx5SjJ
- MZvtwhMrEQVw0aSfECNl

0gBSkU1ncCRU7RwmQKSZ

```
HeartRate: 99                                            (number)
Timestamp: November 19, 2024 at 11:09:19 AM UTC-5        estamp)
```

Fig 2. Heart rate data stored in Firestore Database



(default)

- Accelerometer
- Accelerometer
- GPS
- Hardware
- Temperature
- Temperature
- Users

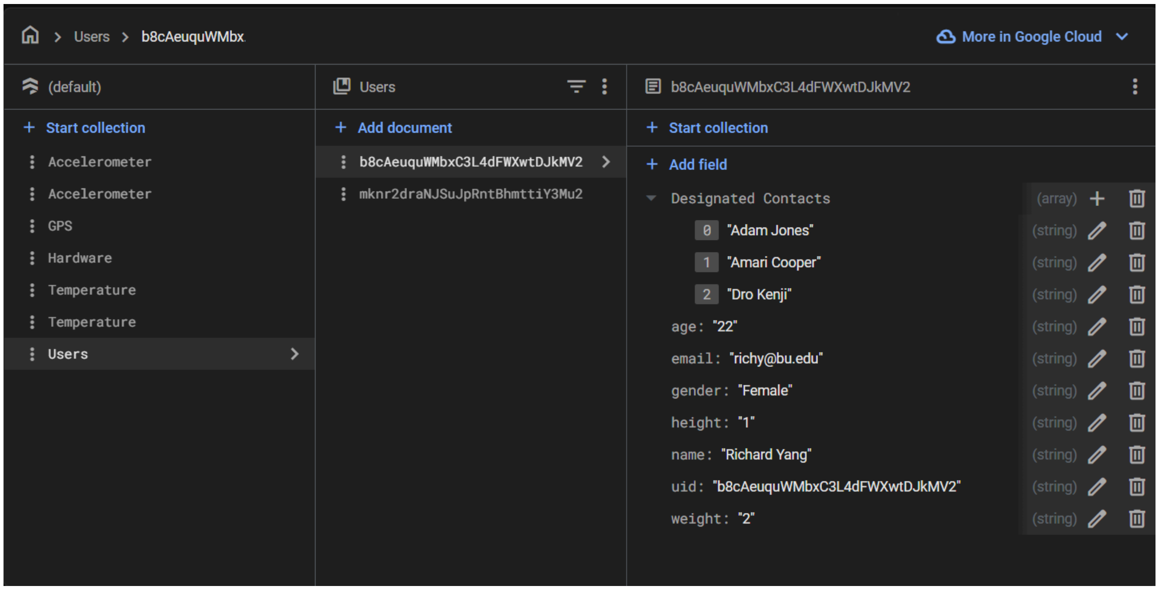
Temperature

- 8MmWPMSvcV4nZ4LESaL8

8MmWPMSvcV4nZ4LESaL8

```
Temperature: "20.75"                                     (string)
Timestamp: November 19, 2024 at 11:39:52 AM UTC-5        estamp)
```

Fig 3. Temperature data stored in Firestore Database



Accelerometer
Accelerometer
GPS
Hardware
Temperature
Temperature
Users

b8cAeuquWMbxC3L4dFWXwtDJkMV2
mknr2draNJSuJpRntBhmttiY3Mu2

```
Designated Contacts                      (array)
    0 "Adam Jones"                       (string)
    1 "Amari Cooper"                     (string)
    2 "Dro Kenji"                        (string)
age: "22"                                (string)
email: "richy@bu.edu"                    (string)
gender: "Female"                         (string)
height: "1"                              (string)
name: "Richard Yang"                     (string)
uid: "b8cAeuquWMbxC3L4dFWXwtDJkMV2"      (string)
weight: "2"                              (string)
```

Fig 4. Users data stored in Firestore Database

4 Conclusions

From our prototype testing, and test data collected, we have distinguished a number of potential issues in hardware to address. Firstly, the temperature sensor, although reading data, was outputting incorrect values which we had recognized prior to the test. We suspect that this is an issue with the sensor itself, and a byproduct of purchasing cheap alternatives. Additionally, we recognize that the GPS sensor failed to register coordinates, but we have predicted this issue beforehand due to low satellite signal. We've also gained valuable insight in the orientation of the accelerometer by recognizing discrepancies in the values for each axis. Nonetheless, the heart-rate sensor performed as expected and gathered accurate data. Furthermore, all sensors were powered successfully and were able to transmit data for cloud storage in Firestore. Moving forward, we will take these considerations into the future purchase of components, potential signal amplification, and enclosure design with respect to sensor orientation.

In parallel, all software functionality and data expectations were met successfully. The native application was able to successfully run and fulfilled all previously outlined requirements. As planned, user data such as their uid, email, name, profile, and designated contacts were all stored in unique collections within Firestore. Additionally, all page navigation and data persistence functioned as expected. In conclusion, we deem that our software has a strong foundation to continue building off of as we look to begin interfacing with the hardware gathered data through Firestore.