Boston University

Electrical & Computer Engineering

EC464 Capstone Senior Design Project

Final Test Report

Personal Alert Device

April 13, 2025

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 components included a custom 3D-printed enclosure with an integrated wrist strap. Housed inside the enclosure was a custom PCB with the Seeed Studio XIAO nRF52840 Sense microcontroller. The board also included red, yellow, and green LEDs, three 330Ω resistors (for each LED), a push-button switch, a buzzer, a MAX30102 pulse oximeter, and a B57703M0103A018 thermistor for temperature monitoring. The microcontroller's built-in 6-axis IMU and PDM microphone were also utilized as part of the system. A 3.7V lithium battery was connected to the microcontroller and charged via a wireless charging receiver using copper coils. A housing for the transmission module was also made and 3-D printed. On the transmission side, a USB Type-C cable connected to a 120 VAC to 5 VDC adapter and transmitter module, which connected to the transmitting copper coil. The transmission module is attached to the receiver on the wearable device via mechanical fitment and magnets. It is used to deliver 5V wirelessly to the receiver in order to charge the battery.

The goal of the hardware setup was to demonstrate core sensing functionalities, speech recognition via machine learning via the on-board PDM microphone, and the overall wearable form factor. We successfully validated the accuracy and consistency of the pulse oximeter and temperature sensors, which displayed correct results on the mobile app. The button-controlled emergency and battery status features were also successfully demonstrated. When the button was pressed for less than three seconds, the green LED turned on to indicate battery status and turned off upon a second press. When held for more than five seconds, a warning period is initiated as the red LED blinks for ten seconds and the buzzer is activated. If the emergency is not canceled within the warning period, the emergency response is started (sends SMS, saves emergency record, opens Medical ID on app). However, if the button was pressed again before the warning period ended, the system correctly canceled the alert and did not send any emergency responses.

Additionally, we implemented and validated speech recognition functionality. When the phrase "send help" was detected by the on-board PDM microphone, the system correctly classified it, distinct from background or unknown audio, triggering the red LED to blink and activating the buzzer, simulating an emergency scenario. We also confirmed that the device operated reliably while powered solely by the battery. Finally, the completed enclosure with the wrist strap successfully illustrated the intended user interface and wearable design of the final product.

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 user's collection in Firestore. Furthermore, the laptop established a means of communication between all the components of our project via Bluetooth Low Energy (BLE) as the phone was emulated on the laptop.

Overall, our equipment was consistent with our test plan. All sensors were powered successfully and recorded accurate data. Additionally, all communication between components functioned as expected. Finally, the power system and mobile app worked as expected.

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, including the MAX30102 Pulse Oximeter Sensor, B57703M0103A018 thermistor, a button switch, our speech recognition model, the fall detection threshold model, and the power system. These components allowed us to determine emergencies related to the user's health or surroundings. We also wanted to ensure we received auditory and visual feedback when an emergency was detected by utilizing our buzzer and LEDs. These sensors and devices were stored in the 3D-printed housing unit and wristband. The thermistor was placed in contact with the wrist to ensure accurate skin temperature readings. The heart rate sensor was similarly placed along the user's wrist to prevent interference from external light. The microcontroller also contained the microphone unit, accelerometer, and gyroscope unit. Before showing each sensor's functionality, we uploaded all the code onto the microcontroller and disconnected the device from external power to demonstrate that the power system functioned as expected. Once the microcontroller had all the sensor code, the sensors would actively record data and then upload the data to a Firestore collection, which provided our database with observable real-time updates. We tested speech recognition by saying "send help" into the PDM microphone and having the deployed model classify the speech accordingly. This triggered the red LED to turn on and the buzzer to start sounding off. We also showed how the button on the housing unit could be used to cancel an emergency or issue an emergency by holding it down for 5 seconds, resulting in the same outcome as saying "send help." For the power system, we charged the device before the presentation to show that we could track the battery using the button and receive feedback from the LEDs (in the demo, this was done by clicking the button once, resulting in the green LED lighting up to signify the battery was above 80%). We also brought a separate microcontroller and enclosure to highlight the fall detection, which we modeled by simulating a fall and observing the feedback from the terminal.

The software portion, an Android native application, serves as an interface between the user and the device, gathering vital information dynamically. We ran the application on the emulator using Android Studio to view various pages, such as the profile, contacts, device status, and health page. Walking through the app, we were able to first highlight the easy sign-in mechanism which utilizes Google's single sign-on API. We then showed how users could upload health information specific to their health records through APIs used by the NIH and make edits to personal information, like weight or height. We also showed how to add emergency contacts and how to read current vital information like heart rate and temperature in the health page. We also showed the database we're using on the backend and how user information is stored in unique containers, and also containers that kept track of the user's emergencies in chronological order. Finally, we triggered different emergency responses to show how alerts were sent to designated contacts via SMS as well as to the user's emergency history, which both contained the method of emergency trigger, live vitals data, and location information.

3 Measurements Taken

The final test measurements taken include heart rate data (BPM) and blood oxygen levels (SpO_2), from the MAX30102 sensor, speech inferencing accuracy, and body temperature (°C) from the B57703M0103A018 thermistor. Our live heart rate reads around 70 BPM when the Personal Alert Device is worn on the wrist, which is the expected value. Our body temperature was around 36 °C, which was expected as skin temperature measured on the wrist ranges from 33 to 37 °C. Finally, our Sp02 reading was around 96%, also in the expected value of 95%-100%. Thus, our test measurements taken confirmed the accuracy and functionality of our vital sensors. The values obtained can also be viewed by the user in the mobile application. The functionality of the button, LEDs, and buzzer were also measured. The user is able to press the button for 5 seconds, which should cause the red LED and buzzer to turn on (manual activation trigger). After issuing an emergency, the user is able to cancel it by pressing down on the button again. In addition to this, pressing the button switch one time displays the current battery level on the corresponding LED. Pressing the button switch once again turns off the indicator. These tasks were successfully performed during testing. We also measured the accuracy of our machine learning model for speech recognition. When our keyword, "send help," is said twice toward the Personal Alert Device, it triggers the speech activated emergency response. This response can be canceled by pressing the button once during the warning period. Furthermore, the request was reflected in Firestore under the unique user identifier with an appended time stamp. Throughout the test, there were false positives from background noise or other spoken words due to the accuracy threshold of the speech recognition model being too low. The fall detection algorithm was also tested in which the message "Fall Detected!" will be printed when the thresholds for acceleration magnitude and gyroscope are met. The algorithm then waits another 5 seconds to allow for residual peaks in the fall or the user to get back up after the fall. After this period has passed, the algorithm samples accelerometer and gyroscope data for another 5 seconds. If the user is able to move, the emergency response will be canceled, however if the user does not move during the 5 second period, the emergency response is initiated. Both of these scenarios were successfully demonstrated during testing.

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/removing designated emergency contacts, contacts stored in respective Firestore collection, logging out, editing profile, changing profile picture, editing medical information, seeing proper emergency response when triggered (Help Screen, History/Record, SMS), and having user data persist throughout application refresh, close, or sign out. All these tasks were successfully performed during testing.

4 Conclusions

Working on the Personal Alert Device this semester, our main priority was developing and finalizing the hardware portion of the project. This included a housing for the wearable, power system design, wireless charging setup, and sensor integration within the wrist strap. Through our final testing, we have concluded that our enclosure designs for both the wearable and charging stand are both robust and compact. The form factor of the wearable is minimal in size yet it accommodates a battery life of ~ 4 days. The charging setup is simple yet effective through the implementation of magnets and mechanical fitment. The wrist strap is easy to put on, comfortable, and provides stability. Finally, the integrated sensors are placed correctly, offering accurate and reliable sensor data. Other components of the wearable included the on-board switch, buzzer, and LED indicators. The switch and LEDs are well-placed, offering easy usage and visibility, while the buzzer is reasonably loud. Pressing the switch once successfully turned on/off the battery life indicator. Through the testing and measurements taken, we conclude that all on-board components (switch, LEDs, sensors, buzzer, battery) functioned as expected.

Our implementation of four separate emergency triggers functioned correctly as well. There are three wearable-based emergency triggers: holding down the switch, speech recognition (saying “send help”), and fall threshold. Each of these three triggers performed the correct emergency response by first initiating a warning period in which the user may cancel the response by pressing the switch, which also functioned correctly. However, if the user does not cancel the emergency response, the user’s medical ID opens automatically on the mobile app and an SMS message containing emergency details is sent to designated emergency contacts. Additionally, a record of the emergency and additional information at the time of the response were stored correctly (both in Firestore and History screen). The fourth emergency trigger, an app-based hold button, functioned as expected as well. Each of these four emergency triggers correctly indicated how the response was enabled within the records and SMS.

On the software side, all functionality tests were successful. The wearable successfully connected to the phone via Bluetooth, and the connection status was reflected within the mobile app. Additionally, all other necessary device-based data such as emergency response triggers, vitals, and battery status were reflected accordingly in the app with low latency. All app navigation functioned as correctly, and APIs, HTTP messages, and SDKs such as for medical conditions or user location performed expectedly. All screens reflected correctly updated information both from app-based user input or device-based data, and in parallel, information was correctly stored and retrieved from a user's unique Firestore collection.

One area of improvement that was noticed was the accuracy/sensitivity of the speech recognition. Occasionally, throughout the testing, false alarms for the recognition of "send help" were triggered due to a lowering of the classification threshold from the previous test. Finding a more suitable balance between sensitivity (false positives) and false rejections is significant in improving device's speech recognition accuracy. Additionally, the machine learning model could use more sample data to ensure that all different voices can be recognized with equal accuracy. Finally, a reduction of the classification window or switching over to continuous inference rather than buffer-based standard inference will improve the speech recognition accuracy as well. Nonetheless, the false positives were successfully canceled through pressing the switch within the warning period.

In conclusion, all project requirements for both software and hardware have been met, and through the final testing and measurements taken, can be concluded that they function correctly. All of these functional requirements have been successfully implemented even with the addition of some extra features such as wireless charging and speech recognition. While there still remain some small areas of improvement for accuracy and design, the core goal of improving upon current personal alert industry solutions has been met.