Running Safety Test Plan

Team 22:

Thomas Cimino

Zexing Gao

Cong Han

Payton Hauck

Yajing Lai

Customer Requirements:

Professor Pisano is our current customer, and at the beginning of the year, he required a few specific characteristics for our device. They were:

* Automatically contacts help (Does not need to be accompanied by a mobile phone or activated by the user’s touch)
* Comfortably concealed and not visible to the public eye
* Lasting battery life (enough to be able to complete a full running session without needing a charge)
* Part of the device in the shoe to track running force
* Durable (able to survive all physical blows it may have to take)

Our team has done a significant job in meeting all the requirements that our customer gave to us when we began this project. The device has successfully implemented all of the requirements listed above. It was significant that we met all of these requirements, in order to have a competitive advantage against the few current safety devices that exist. Ours is far better than the existing devices in that it does not require user activation, is hidden from the public eye, and is comfortable for the user (being that they have to wear it while running).

Hardware Setup:

Our hardware setup will consist of the two main device components, each showing distinct functionalities of our product. There are two isolated Beetle BLEs connected to each other via bluetooth. The shoe component consists of the slave Beetle, which will be connected to a force-sensitive resistor. The chest strap component consists of the master Beetle which is connected to our GPS/GSM module, pulse monitor, vibration motor, and a push-button.

What to perform for hardware tests:

1. The test will show the autonomy of our design. We will turn on the two sub-devices and they will connect automatically. From there we will run several trials. The first will be repeated, frequent pressure events on the force-sensitive resistor to show that the device remains in standby mode, simulating a jogger’s repeated pressure on his or her feet. The second trial will show the system automatically responding when there is no pressure on the force sensor. After ten seconds of no pressure, the shoe insert will automatically send a signal to the main chest strap module with the vibration motor. The vibration motor will turn on for another ten seconds. After that, the chest strap module will send a signal back to the shoe insert module, which will put that shoe insert the module into permanent standby. The chest strap module (connected to the GPS/GSM board) will enter distress mode, sending the user’s Google Maps location via Twilio to the user’s emergency contact that is in the device database. The third trial will show the functionality of the false-positive cancel button. Again, we will let the shoe insert timeout, sending the alert signal to the chest strap which turns on the vibration motor. This time, however, we will press the cancel button on the chest strap, which will turn the vibration motor off, and reset both devices to normal. We will go on in this trial to let the device automatically respond, to show that they are constantly monitoring their environment even after a false-positive cancellation.

Hardware Measurable Criteria:

1. No false detections under normal operation on pressure sensor (no warning signs)
2. Total system timeout and emergency text sent after 20 consecutive inactive seconds (vibration motor on)
3. Warning of system timeout after 10 consecutive inactive seconds (vibration motor on)
4. Return to normal operation after warning cancellation (no vibration warning)
5. Total system timeout and emergency text sent after 20 consecutive inactive seconds **after** warning cancellation

Software Setup:

1. We need an internet connected laptop with Windows OS.
2. We also the functional mobile phone that receives the “emergency text”
3. We do not need to pre-install any node.js packages, because we have already launched the website on AWS.

Software Tests:

1. User interface
2. Login page: completely implemented our registration & authentication function into the interface Users log in with an email and password on the Login page. If they do not have an existing account they can click on “New Member” and sign up for our app.
3. Home page: a map with user location shows the current location of the device.
4. The user enters the device number and emergency contact on this page, which all will be saved into the cloud database.
5. After user click sign out button, the system would sign out and quit to the initial login page.
6. Users can now access and manual the same webpage on their phones.

Software Measurable criteria:

1. Web page (interface) can be displayed.
2. Users can log in or sign up successfully. User authentication information can be saved correctly in Firebase system.
3. Current user location can be found on the map, and shown properly to the user.
4. The user will register the device and their account by entering the device number and emergency contact that they would like stored.
5. All of the user’s data is stored in the cloud database and now is connected to the hardware system, allowing them to use the device whenever they turn it on.

Improvements considered for version 2.0:

1. The styling (CSS style sheet) of the Web app.
2. Add new contacts buttons, which would allow adding multiple contacts.
3. Launch the web into Amazon Web Server.
4. A new “sign out” button.