Heart Health App

Group 7

Nicholas Landi

Scott Hinden

Daisy McCoy

Johnson Odibi

Ivette Perez

Abstract

ArduniO2 is a Heart Health mobile application which incorporates both a pulse ox and an android phone. Measuring the partial arterial blood oxygen level (SpO2) and the pulse of the current user. It is used for a variety of people including those suffering congestive heart failure. The Arduino proceeds to transmit the pulse and oxidation results to the mobile platform being the Android device for further interpretation by a health care provider. Congestive Heart Failure (CHF) is one of the most significant chronic diseases affecting about 5 million people in the U.S and it is the leading cause of hospital admissions, especially in elderly patients. The rate of readmissions have increased, 45% of patients will be readmitted for worsening symptoms within 6 months. It has been recognized that daily monitoring of some vital signs can ensure early detection of signs of declination, allowing for timely interventions, leading to a reduction of hospital admission and readmission.

Background

ArduinO2 is a prototype medical application that measures the partial arterial blood oxygen level (SpO2) and pulse rate respectively, by interfacing with an e-Health Sensor Shield and an Arduino device. The Arduino now transmits the results via bluetooth to a mobile platform being the Android device. People suffering with congestive heart failure will find the ArduniO2 Heart Health app the most effective. Heart failure is when the heart is weak and cannot pump blood effectively. Heart failure is a condition in which the heart does not pump well. This causes the heart to lag behind in its job of moving blood throughout the body. As a result, fluid backs up in the body, and the organs in the body do not get as much blood as they need, thus do not get enough oxygen as they need. If your heart does not pump well, it can cause tiredness or weakness, trouble breathing, a racing heartbeat, even while resting, swelling in your feet, ankles, and belly; and these symptoms may worsen or occur more often if not managed well. The development of an app as presented can help control, and follow trends (such as heart rate, oxygenation, weight) to better manage the disease and avoid exacerbation and subsequently hospitalization (Colucci, et al., 2015).

The components that were incorporated include a pulse Oximeter, bluetooth device and an e-Health Sensor Shield which is placed on top of the Arduino device for body monitoring medical application and biometrics. Its nine different sensors include: pulse, oxygen in blood (SPO2), airflow (breathing), body temperature, electrocardiogram (ECG), glucometer, galvanic skin response (sweating), blood pressure (sphygmomanometer), and patient position (accelerometer). For the ArduiO2 the sensors used were specifically for pulse and oxygen in blood (SPO2). Real time monitoring is used to gather information on the patient and this biometric information is sent wirelessly using Bluetooth. Cooking Hacks (developer of the E-Health Shield) provided an open source medical monitoring platform which was used to make ArduniO2 a reality.

The pulse Oximeter measures the arterial oxygen saturation which is the amount of oxygen dissolved in the blood by detecting the Hemoglobin and Deoxyhemoglobin levels. In order to measure the Hemoglobin and Deoxyhemoglobin levels, two different light wavelengths are used to see the difference in the absorption spectra which affects the bloodstream. The absorption coefficients are measured using two wavelengths 660 nm (red light spectra) and 940 nm (infrared light spectra). To calculate the arterial oxygen saturation a photo-detector perceives the non-absorbed light from the LEDs. The pulse oximeter sensor is recommended for patients whose oxygenation is unstable which includes intensive care, operating, recovery, emergency and hospital ward settings, and pilots in unpressurized aircrafts. The accepted ranges for patients are from 95 to 99 percent except those with a hypoxic drive problem is between 88 to 94 percent. The pulse Oximeter is connected directly to the Arduino and does not use external/internal battery. The monitoring of oxygenation (SpO2) is one biological parameter that is essential in detecting exacerbation. This android app can enable the CHF patient to collect the SpO2 through a wireless pulse oximeter sensor and transmits the trends to a physician who can monitor the patient from home and intervene if necessary. Early home interventions may reduce hospitalizations (Fanucci, et al., 2013).

Methods

Below are basic method definitions for our application. Certain methods are derived from super classes and add additional functionality, such as onCreate, onStart, onSaveInstanceState, and a few others. The Reading class has two private classes which are represented by indenting further than previous classes to differentiate between the public classes and Reading’s private classes.

Android

**About**

onCreate: Sets up view for activity

**ArticleFragment**

onCreateView: Inflates fragment layout

onStart: Overrides super method and check to see if there are any arguments

updateArticleView: updates current view

onSaveInstanceState: saves information in a bundle if activity is restarted

**Education**

onCreate: checks if fragment container exists and places first fragment

onArticleSelected: handles clicks, exchanges fragments when necessary, also checks if device is multi-paned

**HeadlinesFragment**

onCreate: formats style of list and shows headlines in the placed fragment

onStart: handles differences between one-pane and multi-pane devices

onAttach: Makes sure the container interface has been implemented

onListItemClick: Handles clicks and transitions to selected article

**MainMenu**

onCreate: Sets up activity view

onCreateOptionsMenu: Inflates options menu

about: Transitions to about activity

education: Transitons to education activity

reading: Transitions to reading activity

results: Transitions to results activity

**Reading**

onCreate: Sets up list of possible bluetooth devices to select from

onStart: Turns on Bluetooth for user if off

setup: Gets the array of Bluetooth devices to be displayed

onDestroy: cancels Bluetooth connection

onActivityResult: Finishes activity after data is taken(not used)

startThreadConnected: Starts socket connection

**ThreadConnectBTdevice**

ThreadConnectBTdevice: Constructor, creates connection

run: Runs and maintains Bluetooth connection

cancel: Closes Bluetooth connection

**ThreadConnected**

ThreadConnected: Constructor, sets up input and output streams

run: Collects data from Bluetooth and displays data

write: Sends OK to Bluetooth to begin data transmission

populate: Parses received data and stores data

cancel: closes Bluetooth connection

**Result**

Result: Constructor, formats data string for storage

**Results**

onCreate: sets up activity view and finds ListView for display

displayResults: Sets up ArrayAdapter

readResults: Reads results from local storage and puts into string

parseResults: Places results into ArrayList for adapter to work properly

Arduino

The arduino only has two methods, and one is simply to access the pulse oximeter. The other method is a run method that is looking for input from the Android device that will let the Arduino know when to take data and transmit said data. The necessary functions are provided by the e-Health sensor libraries and can be used as such:

|  |
| --- |
|  |
| eHealth.readPulsioximeter(); |
|  |

Results

The pulse oximeter receives a reading which it then transfers it to the android device in which the data is stored on that device. The stored data is displayed as in figure 3 in the appendix as the final product. Fragment implementation was used to create the displayed results.

Discussion

A few improvements would be to add support for multiple users, cleaner, design for both the app and the Arduino, making the app more efficient, adding support for healthcare providers, and adding pictures as visuals.Challenges that were faced include not getting a database to work, trouble implementing support for tablets, and storing data in a database into the cloud. Other methods include implementing a cloud database.

Conclusions

ArduinO2 gives people the chance to monitor their health and the health of their love ones. The key to having a successful medical app is informing the user on the symptoms and causes about heart health, so they can better their health. By explaining to the user what their results mean in medical terms and having two emergency plans depending on how high or low their BPM and SpO2 deviate from the normal ranges will possibly save lives.

References

Bluetooth Library and Tutorial:

<https://www.cooking-hacks.com/documentation/tutorials/ehealth-v1-biometric-sensor-platform-arduino-raspberry-pi-medical#step6_2>

e-Health Library and Tutorial:

<https://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical#step3>

Developer Sites Used as Reference:

<http://developer.android.com/training/index.html>

<http://developer.android.com/reference/packages.html>

Bluetooth Example:

http://android-er.blogspot.com/2015/07/android-example-to-communicate-with.html

Medical References:

Colucci, W.S., Gottlieb, S.S., Yeon, S.B. (2015). Evaluation of the patient with suspected heart failure. *UpToDate.* Retrieved from http://www.uptodate.com/contents/evaluation-of-the-patient-with-suspected-heart-failure?source=search\_result&search=heart+failure&selectedTitle=8%7E150

Fanucci, L., Saponara, S., Bacchillone, T., Donati, M., Barba, P., Sanchez-Tato, I., & Carmona, C. (2013). Sensing devices and sensor signal processing for remote monitoring of vital signs in CHF patients. *IEEE Transactions on Instrumentation and Measurement, 62*(3), 553-569. doi:10.1109/TIM.2012.2218681

Appendix

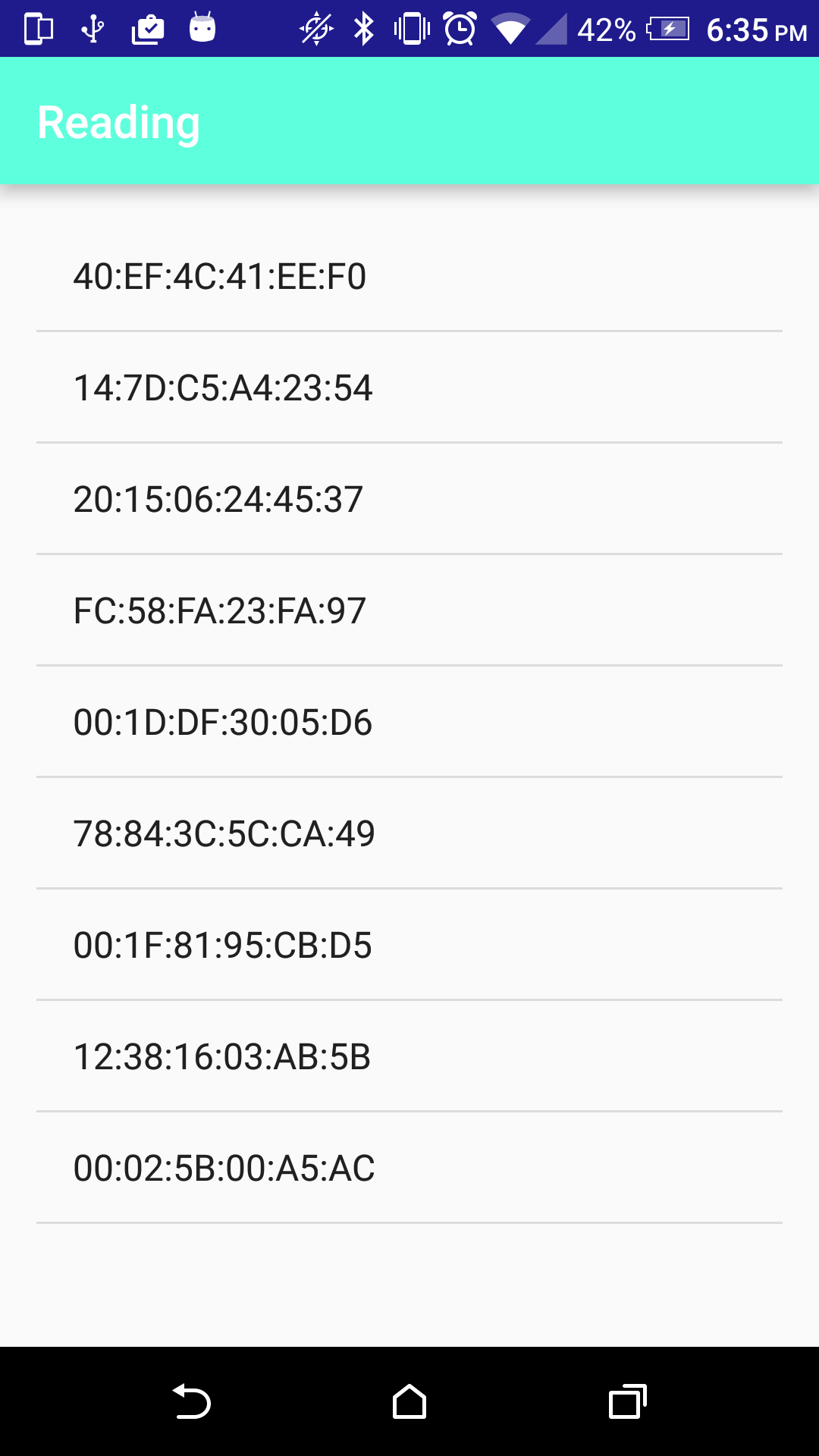
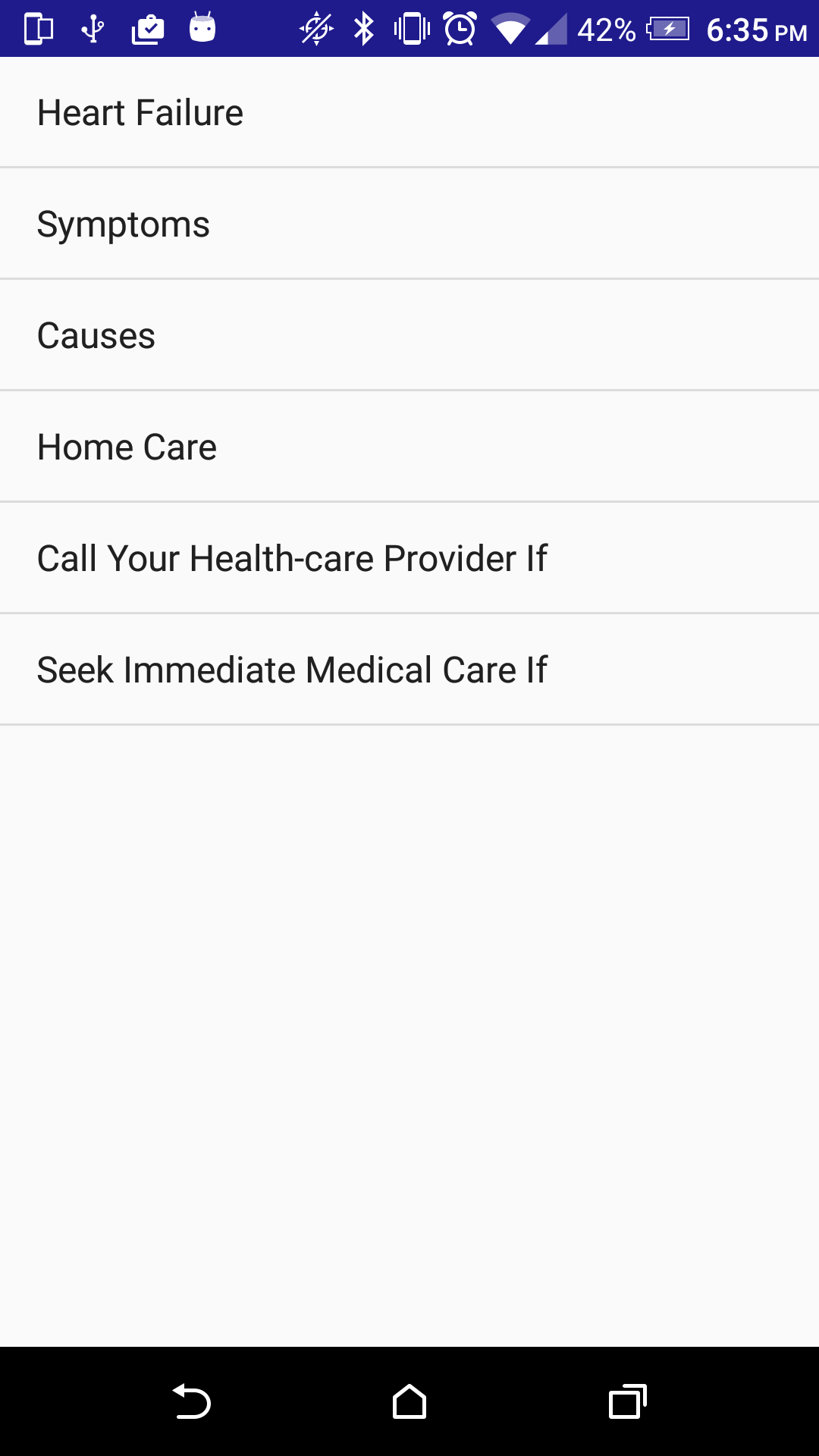
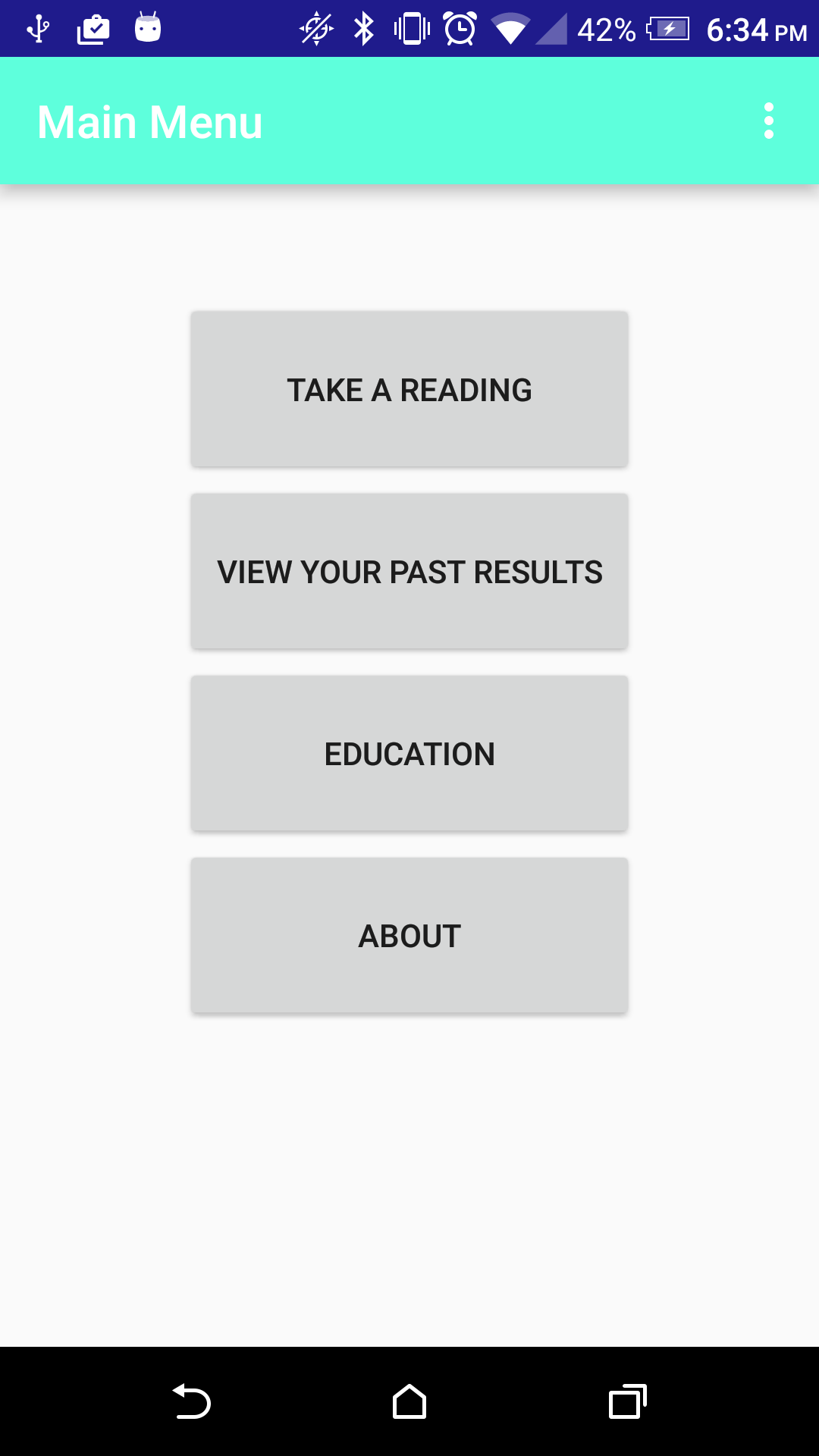


Figure 1 Main Menu Figure 2 Education Tab Figure 3 Bluetooth Device List

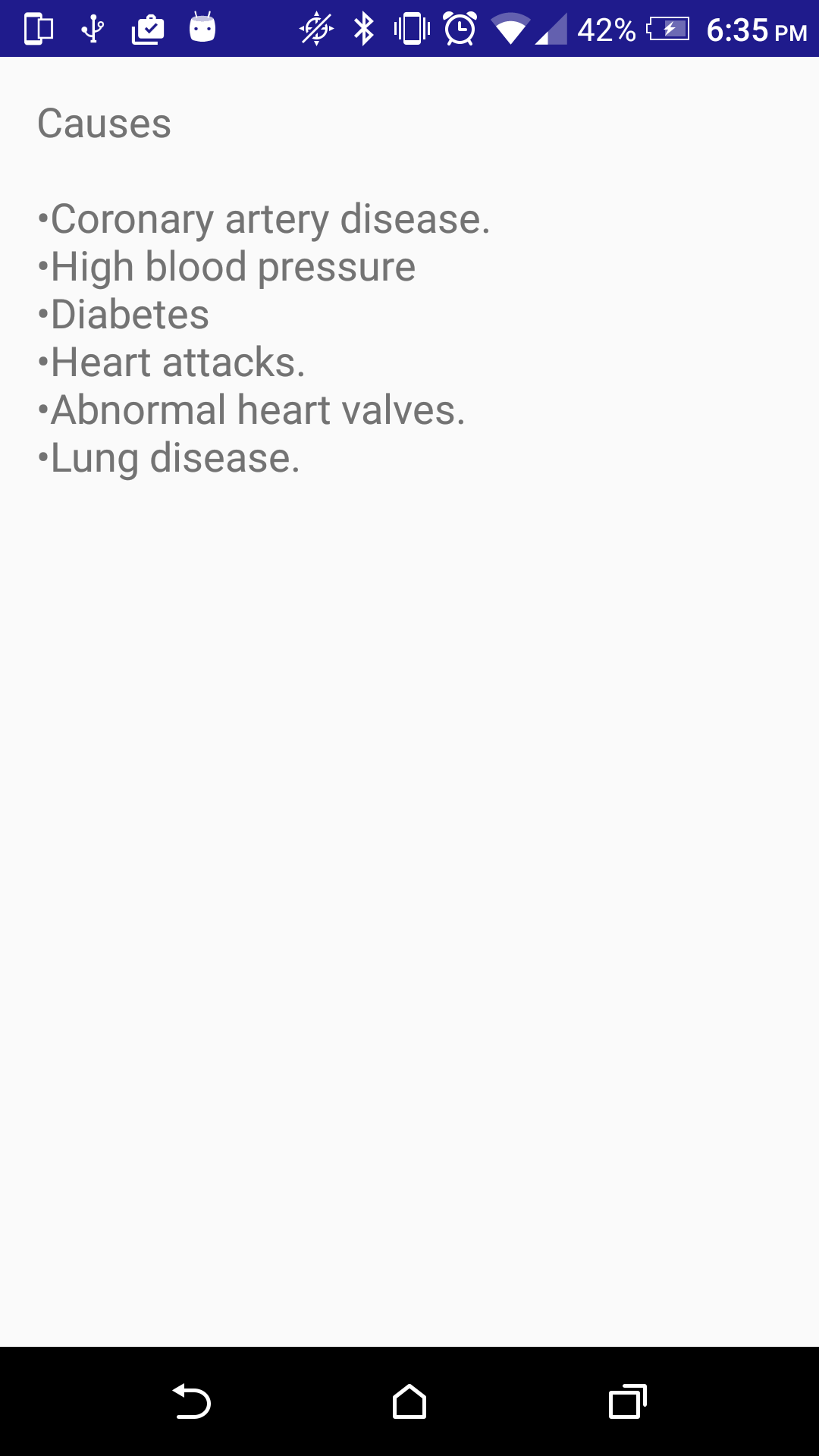
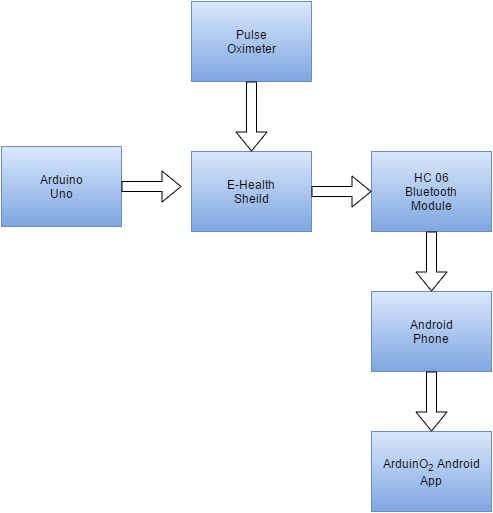
 

Figure 4 Causes of Abnormal Results Figure 5 Block Diagram of Arduino and App

<http://developer.android.com/training/index.html>

<http://developer.android.com/reference/packages.html>