**Abstract- In today’s society, personal health is a prominent public issue. The number of people with poor health increases everyday. There are not enough measures to catch several conditions early on. Often, people do not visit the doctor until the illness is causing serious issues in their day to day life. Elevated vitals are ignored by the patient and usually mediated by over the counter medicine. Doctors are not able to access a patient’s vitals without physically seeing the patient. Medical records are updated per appointment rather than ongoing time based records.**

**This report will focus on the use of mobile devices for health monitoring for medical records as an efficient, low cost and convenience measure. The paper will also include our approach to a solution to health monitoring via mobile devices.**

**Keywords- Health monitoring, mobile devices, healthcare, patient care**

I. INTRODUCTION

The quality of healthcare is depended upon the resources used. The current resources have evolved to from the past but still not efficient enough for healthcare practices today. There is an ongoing pattern of wide variation in health care practice, including regional variations and small-area variations. This is a clear indicator that health care practice has not kept pace with the evolving science of health care to ensure evidence-based practice in the United States. Millions of people do not receive necessary care and suffer needless complications that add to costs and reduce productivity. Each year, an estimated 18,000 people die because they do not receive effective interventions [1].

Therefore, the best avenue for improvement is remote health monitoring. The health monitoring is the continued oversight of the progression of a patient. It is limited to the a clinical study within a hospital and regular doctor visits. Remote health monitoring provides a convenience factor to health monitoring. There are various instruments available in market to keep track on internal body changes. But there are many limitations regarding their maintenance due their heavy cost, size of instruments, and mobility of patients[6]. Through mobile health applications, sensors, medical devices, and remote patient monitoring products, there are avenues through which health care delivery can be improved. These technologies can help lower costs by facilitating the delivery of care, and connecting people to their health care providers[2]. The health professionals can only monitor patients with the current state of technology. As technology and the medicine industry evolves more measures will implemented that can accommodate innovative health monitoring.

II. BACKGROUND

Mobile health mechanisms help frontline health workers and health care providers extend their reach and interactions – enabling them to be more efficient and effective in their provision of medical assistance[5]. Mobile health is a prominent tool in the medicine industry. Many patients use wearable devices to track their daily vitals or metrics.

A Wireless Body Area Network (WBAN) allows the integration of intelligent, miniaturized, low-power sensor nodes in, on, or around a human body to monitor body functions and the surrounding environment. It has great potential to revolutionize the future of healthcare technology and has attracted a number of researchers both from the academia and industry in the past few years.The consideration of WBANs for medical and non-medical applications must satisfy stringent security and privacy requirements. These requirements are based on different applications ranging from medical (heart monitoring) to non-medical (listening to MP4) applications. In case of medical applications, the security threats may lead a patient to a dangerous condition, and sometimes to death[7]. It is important that these wireless health devices are secure with their network and server. The data shared with the network is not accessible by any outside network. Using a mobile based remote health monitoring allows the patient to control the use of their personal data.

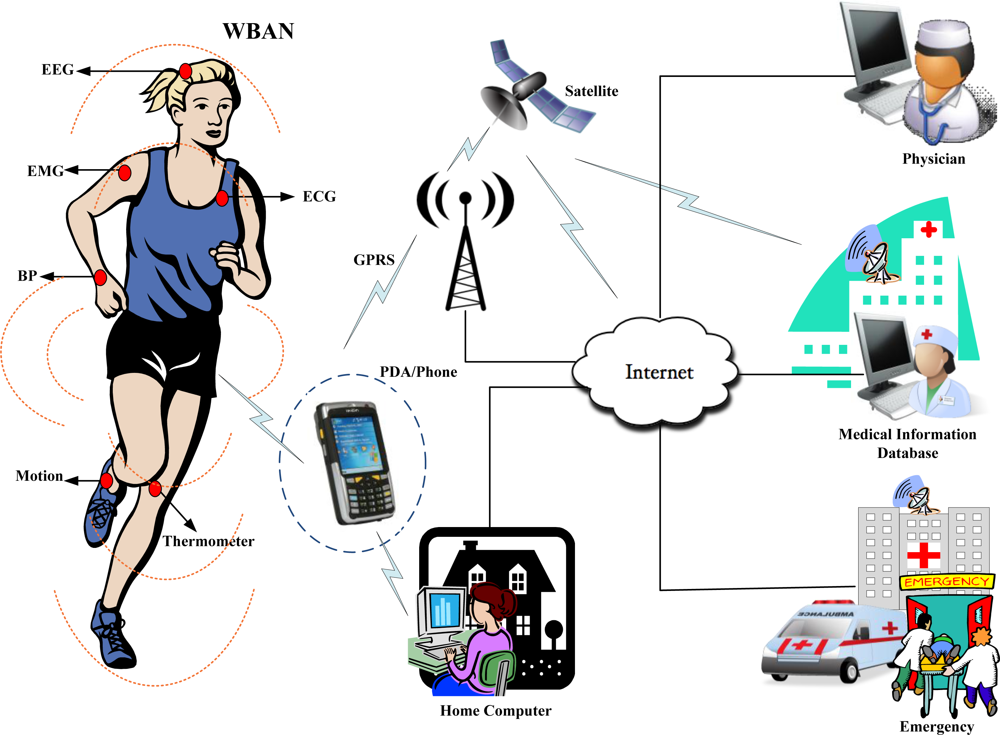


Figure 1: WBAN architecture for medical apps

Mobile Web services provide a natural opportunity for enabling instant access to real-time data. Vital signs can be collected by medical sensors and transmitted via a short-range link like Bluetooth to the a patient’s mobile device[2]. Bluetooth connection capability allows constant data retrieval as long as the device is connected.Wearable devices with Bluetooth capability maintain a secure network to the server. Today, more health care devices are using the Bluetooth connection as an avenue for mobile servicing.

When creating a new resolution for mobile healthcare monitoring it is important to understand the basic attributes of the medical devices and its network.

III. APPROACH

Our approach to the issue of remote health monitoring is to create a Android mobile application that collects vital signs of a patient (through a medical device or wrist band) and sends it to a backend server that maintains a database for EHR (Electronic Health Records). The database will be accessed by the healthcare professional via web page.

There will be a database that collects the patient's vital for their records. The vitals we end up measuring will rely mostly on this accurate reading and the devices available to us. The backend database could be implemented using MySQL, SQL server, or preferably influxDB. A remote health monitoring facility can access these data at the backend, request a fresh copy of current vital signs directly from the user's mobile device, or change the sampling rate on which data are collected. The patient's mobile sends alerts to the healthcare facility if the patient's vital signs exceed a specific predefined threshold. A physician or an authorized care facility practitioner can set/change this threshold remotely on the patient's mobile device.

Our application’s most important features of the user’s vitals will rely more on the biometric will rely on a consistent connection to the backend database and a consistent background process running on the mobile device itself. Accurate reading device we eventually use. Any effort to maintain transmission required on the part of the user will be considered insufficient beyond initial setup and intermittent testing. Our ultimate goal is for the app to transmit data without interrupting the user’s mobile environment. Finally, it is important that those monitoring the database have the ability to push information back to the mobile device, should the user display any vital signs worthy of concern.

IV. SYSTEM DESIGN

Device to Android Phone

The wearable device that is used to collect a patient’s heart rate is the Polar OH1. The Polar OH1 is an optical heart rate sensor worn on the arm. It can accurately and consistently capture heart rates [3]. The Polar OH1 sends raw heart rate data (“attributes”) to an Android mobile application. The team has built an application that supports Bluetooth Low Energy (BLE). BLE allows us to transfer small amounts of data between nearby devices, interact with sensors and communicate with BLE devices that have stricter power requirements since BLE is designed to provide significantly lower power consumption.

The Generic Attribute Profile (GATT) is used for sending and receiving short pieces of data known as “attributes” over a BLE link. GATT is built on top of the Attribute Protocol (ATT) which uses GATT data to define the way that two Bluetooth Low Energy devices send and receive standard messages [4]. ATT is optimized to run on BLE devices and it uses as few bytes as possible. Each attribute is uniquely identified by a Universally Unique Identifier (UUID), which is a standardized 128-bit format for a string ID used to uniquely identify information. The attributes transported by ATT are formatted as characteristics and services.

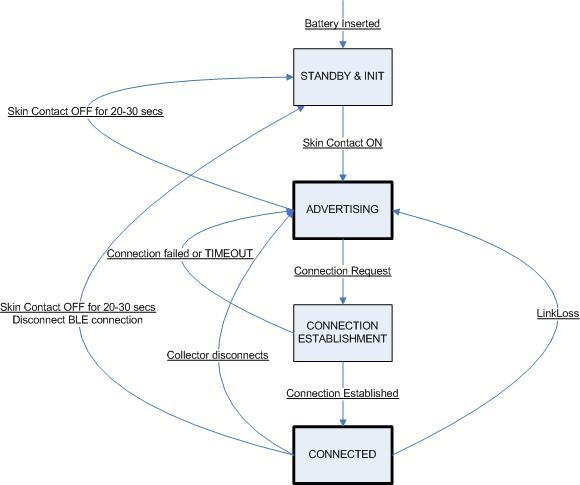
The Android app connects to a discovered device’s GATT server and iterates through the server’s list of available GATT services. Reading what data that it can, specifically name and possible heart rate data, the app converts the data assigned to each UUID to values which are then pushed to to an array. One GATT service may have multiple characteristics assigned to it, but in the case of services detected by the app, namely device information and heart rate, there is only one associated characteristic per service. Callbacks ensure that any changes to the values stored in relevant GATT characteristics will be immediately updated within the app.

Figure 2: Polar Bluetooth State Machine

The Android phone acts as the central role while the Polar OH1 acts as the peripheral role. To establish a BLE connection, we need one device to support the central role and another device to support the peripheral role. Once the phone and heart rate sensor have established a connection, they start transferring GATT metadata to one another. Depending on the kind of data they transfer, one or the other might act as the server. For example, if the heart rate sensor wants to report sensor data to the phone, the heart rate sensor will act as the server. If the activity tracker wants to receive updates from the phone, then it will act as the server.

Android Phone to Server

Once the Android phone receives the heart rate data, it will send the data to a server using the Java Database Connectivity (JDBC). JDBC is an application program interface (API) specification for connecting programs written in Java to the data in databases. We use this to access request statements in Structured Query Language (SQL) that are passed to the program that manages the database. JDBC connections support creating and executing statements such as SQL’s CREATE, SELECT, INSERT, UPDATE and DELETE.

We are using the JDBC API to query and update data in our database. Since we have a heart rate table that contains a time column and a heart rate column, we need to update the table constantly. The application will use the DriverManager.getConnection() method to establish a JDBC connection. The IP of the server will be included in the URL section. The data sent to the server will be a constant value for the heart rate and a timestamp at the time the heart rate is measured. The heart rate attribute is pushed to the server as a background process, set in the app to run every 30 seconds, or upon user command by button press. This same 30 second timer serves as a threshold for alerting the user that their heart rate is above a certain level.

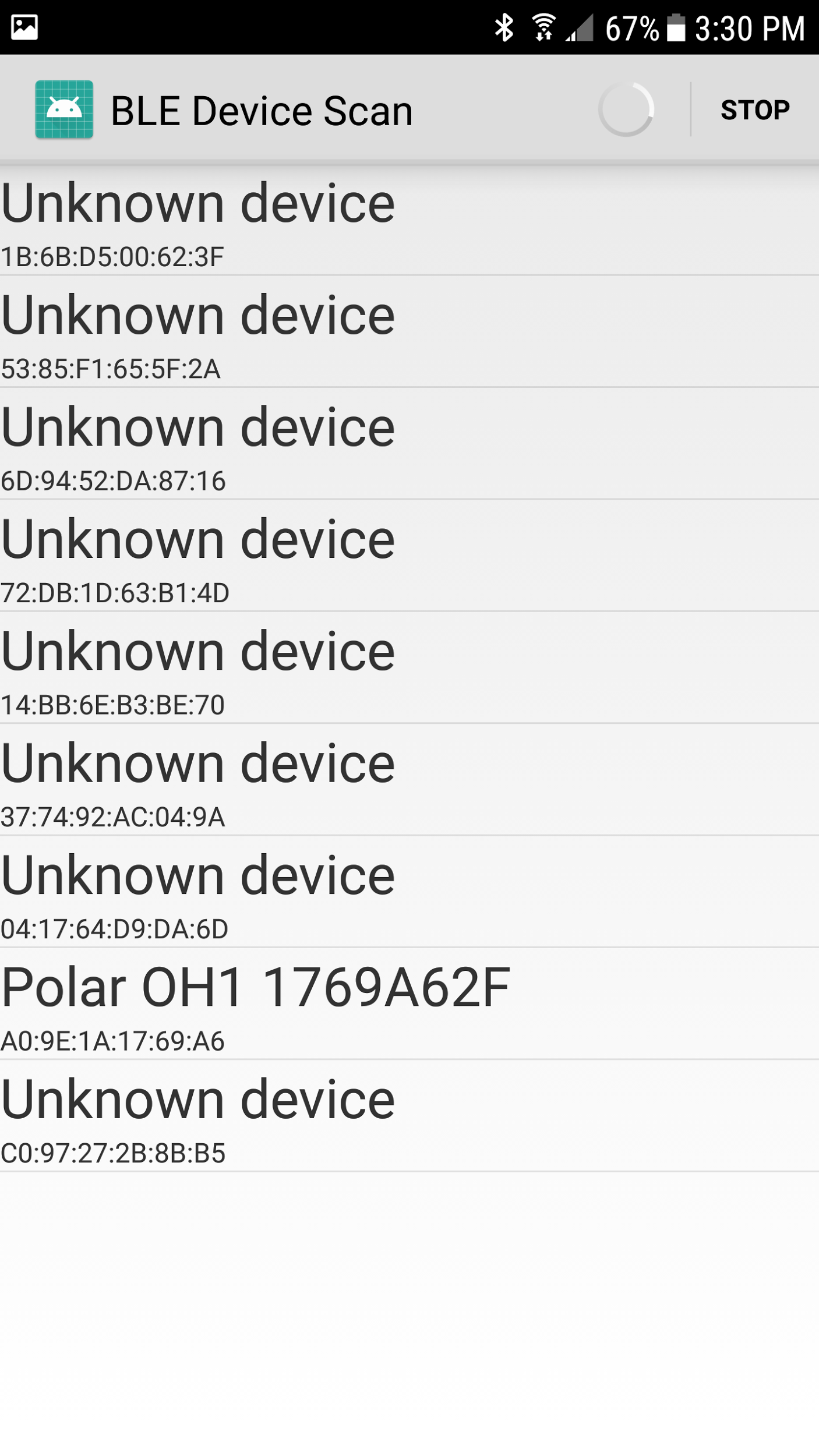
Server/Database/Web Page

We have a WAMP server running on a laptop. The WAMP server consists of the Apache web server, OpenSSL for SSL support, MySQL database and PHP programming language. We set up a MySQL database on the WAMP server. The database consists of 3 tables; Patient, Heart Rate and Threshold. The patient table will contain information about the patients such as their name, patient ID, and age. The Heart Rate table will record the heart rate of patients and the time for each of the heart rate was measured. A threshold table is needed since every patient will have a different heart rate threshold. Next, we have 3 simple PHP web pages. The first page is the login page for the doctor. The doctor will have to enter his or her username and password to get to the next page. The second page is will be a list of all the doctor’s patients. We will use the Patient table in the database to list all the patients on the PHP page. There will be a button for each patient that the doctor can click on to monitor the heart rate of the patient. The buttons for each patient will bring us to the third page. This page will show the Heart Rate table. The doctor can look at all the heart rate of a patient. If the patient’s heart rate exceeds the threshold value, the PHP page will turn red. This serves as an alert to the doctor. The doctor has the ability to turn off the alert and either increase or decrease the threshold heart rate for the patient.

V. PRODUCT

Our current prototype functions properly and fulfilled the proposed methodology. The Polar OH1 is turned on by the button enabling Bluetooth visibility. The Healthy Living application searches for devices upon launch. The Polar OH1 is recognized from the bluetooth device scan in Figure 3.

After the Polar OH1 is connected to the phone through the application. The user must select the service of the Polar needed which is the Heart Rate Services as well as selecting the Heart Rate Measurement option. As the Polar OH1 receives data through the optical sensor. The heart rate measurement data is pushed to the application. The application can push the heart rate of the user to the server by the push of the button. When the user’s heart rate is elevated or lowered, the heart rate is displayed on the application screen which illustrates constant retrieval of the heart rate from the device. If the user's heart rate is higher than the threshold set by the healthcare professional, the application beeps as the alert.

Figure 3: Bluetooth Device Scan within application

The heart rate data is pushed to the server and database. The server and database is connected with a front end webpage for healthcare professionals to access. The healthcare professional logs in to the webpage and the selects the patient. The selected patient’s heart rate is displayed in a list form. The list consist of the the time and associated heart rate. The healthcare professional can set the heart rate threshold to pre set values. On the selected patient page, if the patient’s heart rate surpass the set threshold; the healthcare professional is alerted. The web page background changes to red until the heart rate is lowered to a safe heart rate. Our current prototype of Healthy Living satisfies majority of our approach to the task.

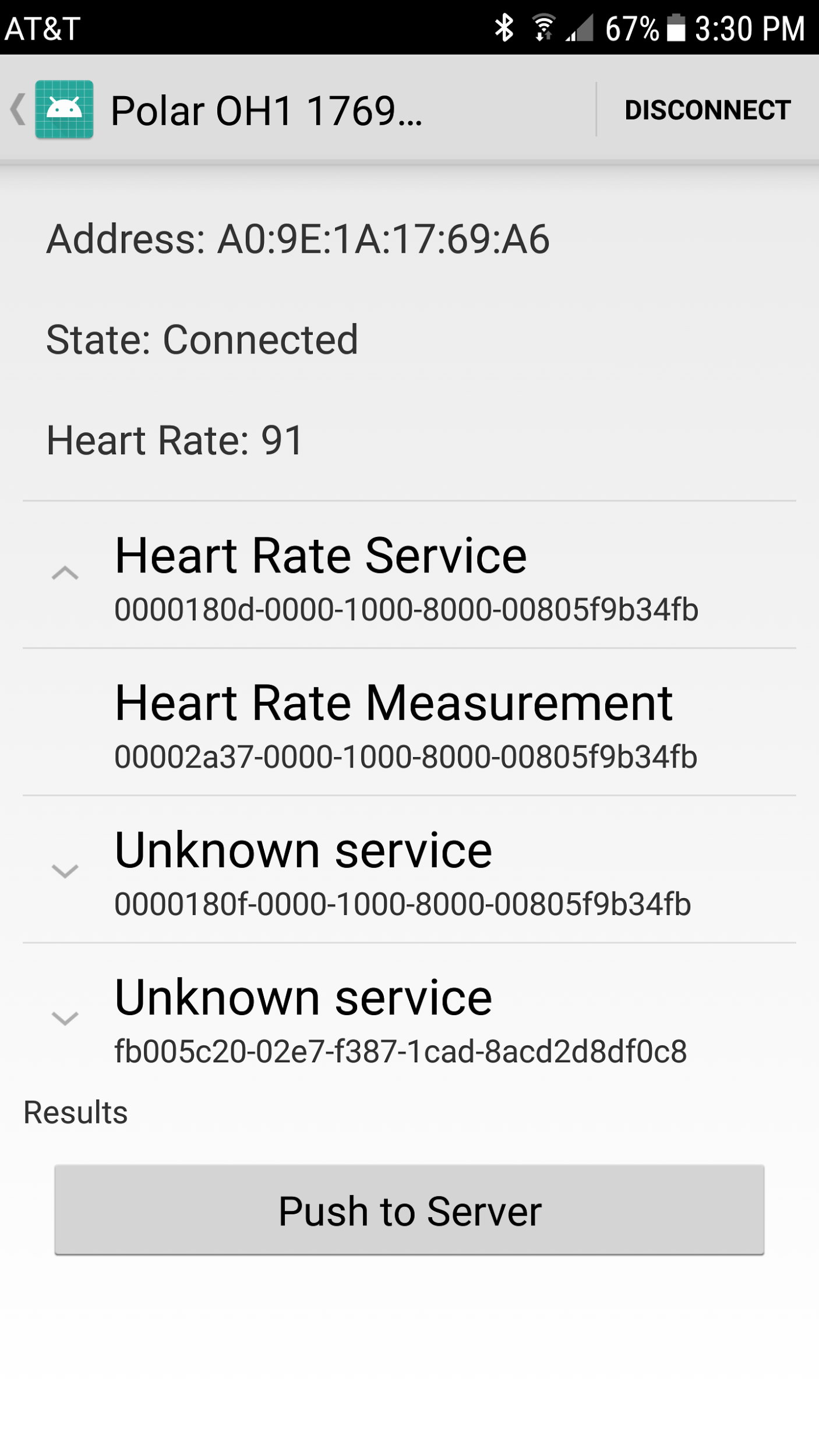


Figure 4: Application Heart Rate Retrieval

VI. IMPLEMENTATION ISSUES

The selection of the device for the implementation was a small hurtle. We needed a simple device that would be functional to the user without complex implementation. The first idea was to use a FitBit. However, the FitBit was a little difficult to work with, it is harder to obtain an api for the FitBit. The next device of choice was Jawbone UP3. The sleek design and specs were ideal for the proposed project. After more research, we realized that the data retrieval is more intricate. In order to retrieve the health data, our team need to contact the developers of the Jawbone database and ask for the data. We later selected to use Polar OH1, which allowed us to collect raw data and manipulate the device to work with our mobile application. The Polar OH1 created a few halts in development after we received the device. There were issue of connectivity based on the lack of detailed instructions provided in the packaging and web page.

The team seeked out different web frameworks such as Django and Laravel. The team worked tirelessly to get Laravel set up working on our own machines. The influxDB also gave the team problems as we attempted to comprehend the documentation and setup. The tool of Grafana is integrated with InfluxDB but was not supported on certain operating systems. The team truly Given the time constraint, we decided to use MySQL with phpMyAdmin since all our team members have had some experience using it.

Trial and error accounted for the process of most of the issues. Most issues that rose were resolved by working through documentation or finding other avenues with functional attributes.

VII. CONCLUSION

Our team has created a base prototype that could be very vital to improvement of the quality of healthcare. It is important to seriously With the time constraints and implementation issues, we were limited to the development of our current prototype. The journey of this project was well worth the end result.

VIII. ACKNOWLEDGEMENT

This research and implementation was supported by Department of Computer Science at the University of Louisiana at Lafayette

IX. REFERENCES

1. Improving Health Care Quality: Fact Sheet. September 2002. Agency for Healthcare Research and Quality, Rockville, MD. <http://archive.ahrq.gov/research/findings/factsheets/errors-safety/improving-quality/improving-health-care-quality.html>
2. K. Elgazzar, M. Aboelfotoh, P. Martin, and H. S. Hassanein, “Ubiquitous Health Monitoring Using Mobile Web Services,” *Procedia Computer Science* , vol. 10, pp. 332–339, 2012.
3. Polar.com, ‘Polar OH1 - Optical Heart Rate Sensor’, 2017. [Online]. Available: https://www.polar.com/us-en/products/accessories/oh1-optical-heart-rate-sensor. [Accessed: 8- Dec- 2017].
4. Developer.android.com, ‘Bluetooth Low Energy’, NA. [Online]. Available: <https://developer.android.com/guide/topics/connectivity/bluetooth-le.html>. [Accessed: 8- Dec- 2017].
5. West, Darrell M. “Improving Health Care through Mobile Medical Devices and Sensors.” *Center for Technology Innovation at Brookings*, Oct. 2013.
6. Shelar, Manisha, et al. “Wireless Patient Health Monitoring System .” *International Journal of Computer Applications* , vol. 62, no. 6, Jan. 2013.
7. Saleem, Shahnaz, et al. “A Study of IEEE 802.15.4 Security Framework for Wireless Body Area Networks.” *Sensors*, vol. 11, no. 12, 2011, pp. 1383–1395., doi:10.3390/s110201383