Privacy-Aware Remote Mobile Health and Fitness Monitoring

Caitlin Gruis
Kaidi Du
Yu Xiao

May 5, 2017

Acknowledgement

Professor Ruzena Bajcsy, University of California, Berkeley
Professor Ali Javey, University of California, Berkeley
Daniel Aranki, University of California, Berkeley

BERKELEY
TELE-MONITORING

Privacy-Aware Remote Mobile Health and Fitness Monitoring
Master of Engineering Capstone Report

Caitlin Gruis
with Kaidi Du and Yu Xiao

Friday, April 14th, 2017

Acknowledgements

Professor Ruzena Bajcsy, University of California, Berkeley

Professor Ali Javey, University of California, Berkeley

Daniel Aranki, University of California, Berkeley

Table of Contents

Abstract

Chronic health conditions such as heart disease, asthma, cancer, and diabetes account for more than $1.5 trillion of the $2 trillion the United States spends each year on health care. Berkeley Telemonitoring is an ongoing project that seeks to combat the outstanding costs of patients with these diagnoses by reducing hospital readmission rates and unnecessary doctor visits. This is accomplished by developing a set of open-source Android libraries that aim to monitor a patient's health at a distance. Information such as blood pressure, heart rate, and weight are collected by BLE devices and a smartphone and sent to a server to be analyzed. Feedback can then be sent to both doctors and patients about the health data. Our team improved the current Berkeley Telemonitoring framework by expanding its BLE capabilities, adding privacy features, and building up the server library.

Chapter One: Technical Contributions

Introduction and Motivation for Berkeley Telemonitoring

It comes as little surprise to the average United States resident that the cost of health care is expensive - the country spends more than $2 trillion annually on it (Partnership to Fight Chronic Diseases, 2009). However, what may come as a surprise is that the majority of health care money is being spent on chronic health conditions. Chronic health conditions such as heart disease, asthma, cancer, and diabetes account for more than $1.5 trillion of the $2 trillion spent each year on health care. Over 47% of Americans have one or more of these chronic conditions, which corresponds to more than 164 million people. (Partnership to Fight Chronic Diseases, 2009). These numbers are startling, and Berkeley Telemonitoring[1] believes we can help lower them.

Before we can talk about what Berkeley Telemonitoring seeks to achieve, it is important to understand what telemonitoring is. Telemonitoring is defined by the National Center for Biotechnology Information as "the use of information technology to monitor patients at a distance" (Meystre, 2005). Most often, it is used by doctors to keep a watchful eye on patients who have just had surgery or have chronic medical conditions. One example of telemonitoring would be as follows. A Wi-Fi enabled mobile device is kept close to the patient to monitor information such as their breathing and heart rate. External devices such as heart rate monitors, blood pressure monitors, or glucose meters may be connected to the mobile device via Bluetooth to assist in collecting data. A server analyzes this data coming in from the patient's mobile device, and a notification can be sent to both the doctor and the patient if the data is abnormal. The doctor can intervene to help the patient if necessary. Figure 1 gives a visual representation

[1] https://telemonitoring.berkeley.edu

of the telemonitoring process just described. This monitoring of patients at a distance would reduce readmission rates as well as unnecessary “false alarm” trips to the doctor’s office and emergency room, thus reducing the overall costs of health care.

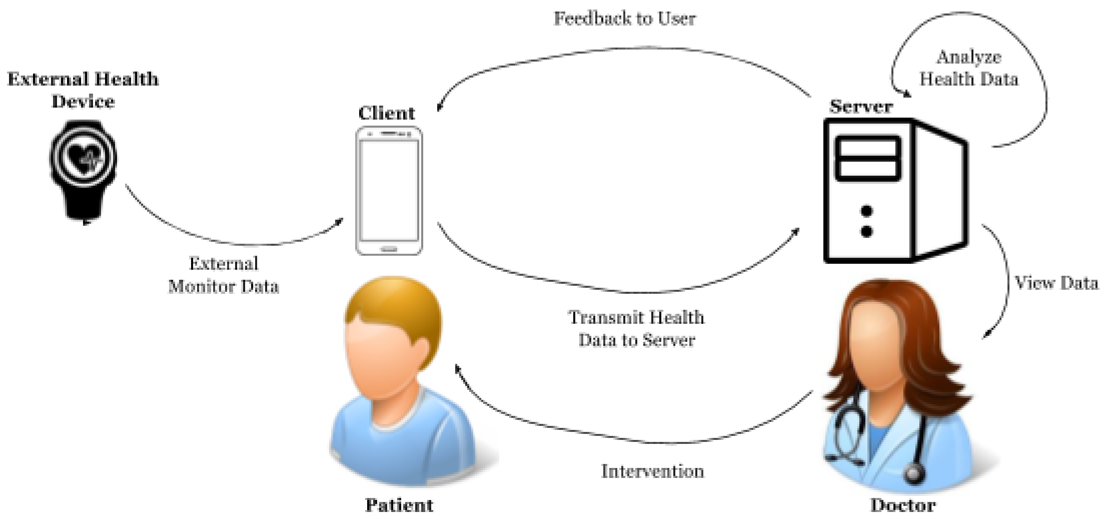


Figure 1. The Telemonitoring Cycle

The Berkeley Telemonitoring project began in 2013, and the next paragraphs aim to inform the reader about previous project developments and project status when this year’s M.Eng. team took over.

The purpose of Berkeley Telemonitoring is to develop a collection of open-source libraries written for Android phones that aim to make the telemonitoring process easier. Many doctors and others in the medical field lack the time and software engineering expertise to implement their own telemonitoring services from scratch. Therefore, this project does all of the behind-the-scenes work of implementing complex health analysis algorithms, handling Wi-Fi and Bluetooth protocols, and storing and securing the data. At the same time, the Android framework is flexible enough to allow doctors and others to use these libraries for their own unique applications (Mani et al., 2015) .

Since the library code itself is a framework and not executable on its own, the Berkeley Telemonitoring team also needed to implement an Android application in order to test the telemonitoring framework and ensure its functionality (Aranki et. al., 2017). To do this, last year's team decided to take a slight pivot and look at what applications telemonitoring could have outside of the medical field. In addition to medical use, it was discovered that telemonitoring also has interesting applications in the areas of coaching and fitness. Telemonitoring has the ability to provide instant feedback to an athlete about their workout and health data, allowing them to adjust their training appropriately without the need of an external human coach. The ability to analyze health data in real time in athletic situations has been proven to reduce costs, save time, and make users more health conscious (Rooney, 2016).

In light of these findings, a long distance running coach application was developed to test the telemonitoring framework and explore the use of telemonitoring in fitness. This application helps to train marathon runners by giving them real-time feedback about their heart rate, cadence, and running pace and encourages them to meet their goals. Figure 2 below shows the Running Coach screen mid-run, giving information to the user about their heart rate, current run time, and cadence. (Sarver et al., 2016). Note that the screen is green because the user's target cadence is being met - when it drops below the target, the screen turns red. This is one of multiple ways that telemonitoring and real time feedback are incorporated into the application.

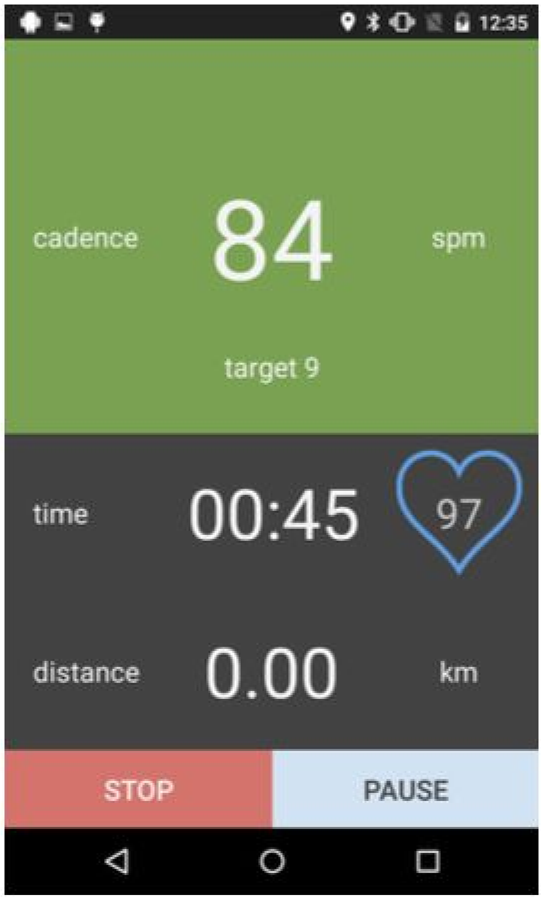


Figure 2. The Running Coach telemonitoring application shown while a user is mid-run.

This was the status of the project at the time my teammates and I began our work in the fall. Much of the Android telemonitoring framework had been created, and the running coach application had been written. Our job was twofold: we needed to continue to expand the telemonitoring framework, as well as test and improve the running coach application. The division of labor among my teammates and the importance of this work is discussed in the following section.

Division of Project Work

During the fall semester, my teammates and I worked together with Ph.D. advisor Daniel Aranki to gain background on the project and learn Java programming. This was essential as the team had no prior experience in Java, and Android applications and libraries are written in this language. We worked through the code to understand the contributions of our Berkeley Telemonitoring team predecessors. Although this step yielded no original work, it was critical to our project because in order to contribute new additions we needed to thoroughly understand the framework that was already in place.

During the spring semester my teammates and I performed some tasks together, but mainly took a divide-and-conquer approach to the project. Our first goal was to expand the existing tele-monitoring framework. There were three distinct areas in which the framework needed to be improved:

1. Bluetooth Low-Energy (BLE) capabilities, specifically in the area of increasing the number of types of external devices compatible with the telemonitoring framework.
2. Sanitization of health-sensitive data to make it private and secure when being stored and transmitted between the client and server.
3. Building up the server and creating a generic database to store and analyze user data.

I contributed mainly to the expansion of the Bluetooth capabilities. My teammate, Kaidi (Kate) Du, added to the privacy and sanitization aspect of the project, while Yu (Sean) Xiao worked on building up the server. Please refer to their papers for additional information on these areas.

Our second goal was to run a pilot study on the Running Coach application that was implemented using the telemonitoring framework. This was a task that all three members of the team worked on together. Before running pilot tests, we needed to complete the server side of the Running Coach that would store the user's running data. This involved creating a database using MySQL and updating the Java code on the server side to store the data in the database. I will talk about my implementation of the MySQL database, while Kate and Sean's papers contain information about the updated Java backend. Finally, the three of us along with our Ph.D. advisor and the rest of the Berkeley Telemonitoring team are in the process of running pilot tests for the Running Coach, so I will provide insights on this testing as well. Figure 3 below gives a summary of our work flow throughout the year. The remainder of the paper will reflect my contributions both in the areas of expanding BLE capabilities and finishing and testing the Running Coach application.

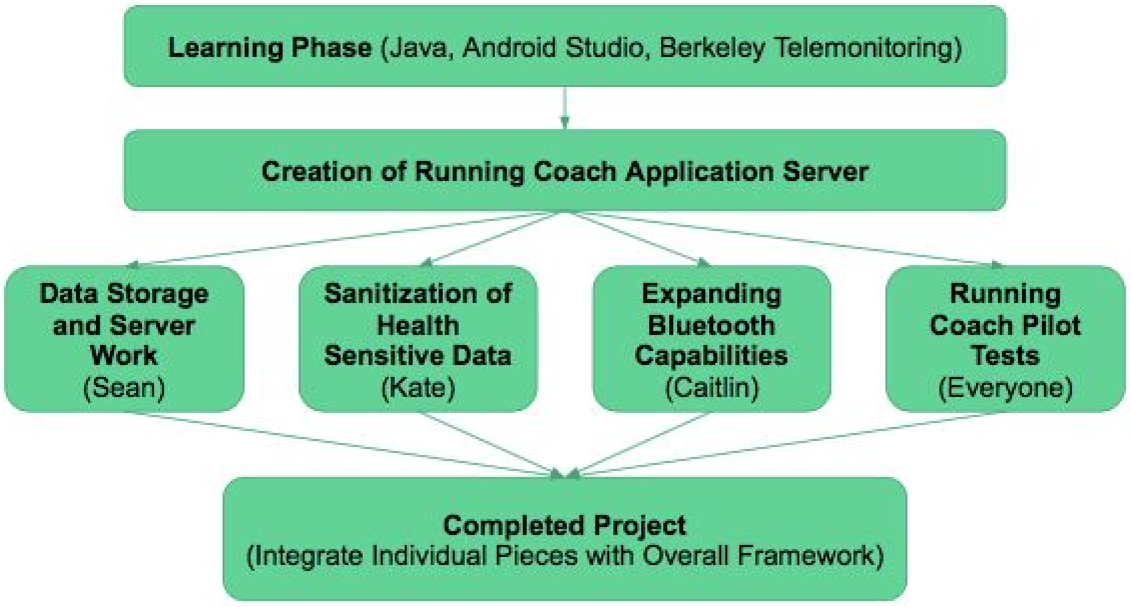

Learning Phase (Java, Android Studio, Berkeley Telemonitoring)
Creation of Running Coach Application Server
Data Storage and Server Work (Sean)
Sanitization of Health Sensitive Data (Kate)
Expanding Bluetooth Capabilities (Caitlin)
Running Coach Pilot Tests (Everyone)
Completed Project (Integrate Individual Pieces with Overall Framework)

Figure 3. Project Work Flow and Division of Labor Among Teammates

Expanding Bluetooth Low-Energy (BLE) Capabilities

The purpose of Bluetooth Low-Energy (BLE) in the Berkeley Telemonitoring framework is to enable external health devices such as heart rate monitors, blood pressure monitors, or glucose meters to pair with an Android mobile device to help track the user's health. This is significant because without these external devices, the phone would be severely limited in the type of data it could collect. Previously in the telemonitoring framework, a BLE stack had been developed and a protocol had been written to make three different types of device families - heart rate monitors, blood pressure monitors, and thermometers - compatible with our framework (Mani and Azar et al., 2015). My work is a direct expansion of this and aims to add to the list of compatible device families.

In addition to the protocol currently implemented for heart rate monitors, blood pressure monitors, and thermometers, I wanted to implement protocols for two additional device families - glucose monitors and weighing scales. My reasoning for choosing these device families was because they are essential in monitoring multiple types of chronic diseases.

A glucose monitor is used by diabetes patients to measure the sugar level of their blood and ensure it is normal. Since 21.9 million people (9.3% of the population) in the United States have diabetes, adding compatibility for these devices has the potential to help a large number of people monitor their condition in a more cost-effective manner (Centers for Disease Control and Prevention, 2015).

The reasoning behind choosing a weighing scale was because of its similar importance in chronic diseases. Weight fluctuations are common in cancer patients, and significant gains or losses can be indicative of a problem (American Cancer Society, 2017). Additionally, patients with cardiovascular health conditions are often instructed to record their weight daily for the same reason (Nicholls & Richards, 2007). Since heart disease and cancer account for approximately 48% of deaths in the United States annually (Centers for Disease Control and Prevention, 2016), telemonitoring framework with compatibility for weighing scales could also make a large impact.

In order to test that these new protocols were functional, it was necessary to obtain devices to test with. These devices needed to be BLE-enabled glucose monitors and weighing scales and also needed to abide by the ISO/IEEE 11073 standard that the telemonitoring framework uses. This standard is a method of communication between medical devices and external computer systems (IEEE Standards Association, 2014). The following devices pictured in Figure 4, the A&D Medical UC-352 BLE Weighing Scale and the Accu-Chek Guide Glucose Monitor, were chosen after researching their compatibility with this standard. The following three subsections will describe how BLE works, and then the design of the two new protocols.

(a) Accu-Chek Guide Glucose Monitor (b) A&D Medical UC-352 BLE Weighing Scale

Figure 4. Chosen BLE devices to test the glucose and weighing scale framework.

1. Transmitting Data Through BLE

Adding a new compatible device family involved a deep dive into understanding how Bluetooth Low-Energy functions. BLE is unlike traditional Bluetooth where an external device must pair with the host device to establish communication. In BLE, the remote device broadcasts data in the form of an Advertising Payload and Scanning Payload.

The Advertising Payload is the mandatory data that the remote device transmits at discrete intervals to alert host devices (in our case, the Android smartphone) that it exists. The host device can then request more information from the remote device through a Scan Response Request. The remote device then supplies this information using the Scanning Payload (Townsend, 2017). The diagram in Figure 5 below illustrates this communication. In our case, the central device is the Android smartphone, and the peripheral device is the new glucose meter that we are adding compatibility with.

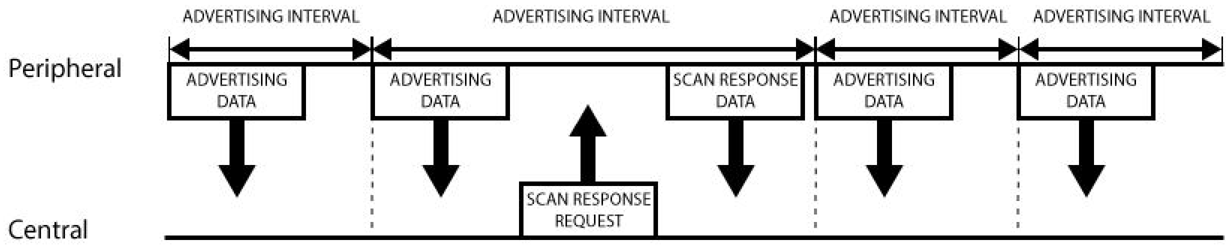


Figure 5. Bluetooth Low-Energy Process

At this point, direct communication between the external device and the central device is established, and the external device will only transmit information to the connected host device. This direct communication involves the exchange of what is known at Generic Attribute (GATT) Profiles.

GATT transactions are high-level, nested objects that consist of a Profile, Services, and Characteristics as shown in Figure 6 below.

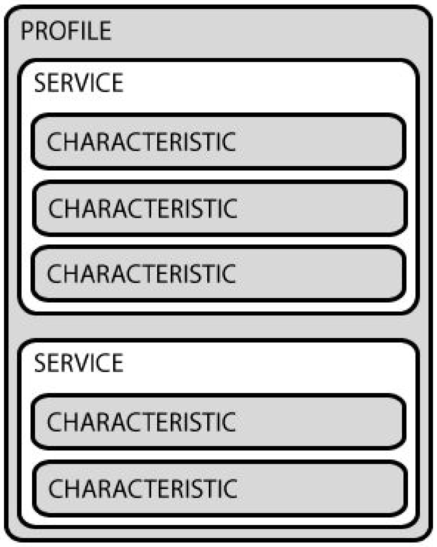


Figure 6. GATT Architecture

At the lowest level of the GATT object are characteristics. A characteristics contains a single datapoint. For example, in the case of the glucose monitor being implemented, there is a Glucose Measurement Characteristic that contains the glucose concentration of a sample as well as a timestamp of when the datapoint was taken.

At the intermediate level of the GATT object are services. A service contains multiple characteristics wrapped together. For example, the Glucose Service contains a Glucose Measurement Characteristic as well as a Glucose Feature Characteristic that describes any additional information about the sensor (such as if a low battery was detected or if an error occurred).

At the highest level of the GATT object is a profile. A profile contains multiple services wrapped together. For example, the Glucose Profile contains the Glucose Service as well as the

Device Information Profile, which contains information about the glucose device (such as the manufacturer's name and model number) (Bluetooth, 2017).

2. Glucose Meter

My role in implementing this new glucose device family was to ensure that any glucose monitor that sends data based on this architecture would successfully be able to transmit GATT packets to our Android application.

The ability to establish connection between any BLE device and the phone was implemented by the 2015 Berkeley Telemonitoring team (Mani et al., 2015). What was lacking was the ability to parse the data of a glucose monitor device. To do this, three major components were needed:

1. The framework needed to be able to recognize that the received GATT packets were being sent from a glucose monitor, and therefore should be listened to.
2. Once the packets were being listened to, there needed to be a way for the framework to match specific bits from the packet to particular pieces of information - i.e. which bits of a glucose characteristic correspond to the concentration measurement, and which bits to the units the concentration was measured in?
3. After the data was parsed, the framework needed a way to store the relevant glucose data.

With regards to the first component, each type of BLE service and characteristic is stamped with its own 16-bit numeric ID called a UUID (Bluetooth, 2017). A class in the telemonitoring client framework called `GattUUIDLookupDirectory.java` contains all of the relevant UUIDs the framework is interested in. By adding the Glucose Service, Glucose Measurement Characteristic, and Glucose Feature Characteristic UUIDs to this class and comparing them to incoming services and characteristics the framework is receiving in a class called `BluetoothHealthDevice.java`, the framework now recognizes when data is sent from a glucose monitor.

With respect to the second component, my design added a method to the `BluetoothHealthDevice.java` class that reads in Glucose Characteristics. Upon reading in a characteristic, it uses the 11073 BLE standard to determine the location of the relevant bits. The method looks for information about the glucose concentration, what the units of the measurement are, and what time the measurement was taken.

Once this data is parsed, it needs to be stored. To accomplish this task, two new classes called `GlucoseDataPoint.java` and `GlucoseContainer.java` were added to the framework. The `GlucoseDataPoint.java` class stores a single glucose data point, which includes the blood sugar concentration measurement and its units (either mol/L or kg/L). The `GlucoseContainer.java` class wraps the glucose data point together with the time of the measurement to complete the data storage.

Now, if a user of our telemonitoring framework would like to take measurements from a glucose meter, they are only required to implement a glucose listener. This listener would extend our `GlucoseListenerInterface.java` interface and must contain one method. This method would be called each time glucose data is received by the framework, and would allow the user to access the glucose data within their application.

3. Weighing Scale

As was the case with the glucose monitor, my role in implementing this new weighing scale device family was to ensure that any weighing scale that sends data based on the GATT architecture and 11073 Health Device standards would successfully be able to transmit GATT packets to our Android application.

The design solution for this device family is very similar to the glucose monitor device family. The same three major components of recognition, matching, and storage needed to be implemented in order for BLE weighing scales to be compatible with our telemonitoring framework.

Similar to the glucose monitor, the Weigh Scale Service, Weigh Scale Measurement Characteristic, and Weigh Scale Features each are marked with their own unique UUID. By adding these UUIDs to the `GattUUIDLookupDirectory.java` class and comparing them to incoming services and characteristics the framework is receiving in `BluetoothHealthDevice.java`, the framework now recognizes when data is sent from a weighing scale.

To match specific bits from the Weigh Scale Characteristics to information about the weight measurement, weight measurement units, and time of the measurement, a method was added to the `BluetoothHealthDevice.java` class that reads in Weigh Scale Characteristics. Upon reading in a characteristic, it uses the 11073 BLE standard to determine the location of the relevant bits.

To accomplish the task of storing the data, two new classes called `WeighScaleDataPoint.java` and `WeighScaleContainer.java` were added to the framework. The `WeighScaleDataPoint.java` class stores a single weigh scale data point, which includes the weight measurement and its units (either pounds or kilograms). The `WeighScaleContainer.java` class wraps the weigh scale data point together with the time of the measurement to complete the data storage

As with the glucose meter, if a user of our telemonitoring framework would like to take measurements from a weigh scale, they are only required to implement a weigh scale listener. This listener would extend our `WeighScaleListenerInterface.java` interface and must contain one method. This method would be called each time weigh scale data is received by the framework, and would allow the user to access these data within their application.

Expansion and Testing of Running Coach Application

As previously mentioned, the second overall goal of our project was to expand and test the Running Coach application, an Android app that implements the Berkeley Telemonitoring framework. This application helps to train marathon runners by giving them real-time feedback about their heart rate, cadence, and running pace and encourages them to meet their goals. The problem with the old implementation that my teammates I encountered was that the Android application was outputting the running and health data to unsecured files instead of a secured server. This was dangerous because not only did it make the data more difficult to process, but it also posed a security risk since this health sensitive data being stored was open for anyone to view. This was viewed as a problem that must be fixed before we began our study with human subjects. In order to change this, my teammates and I needed to alter the Android Java backend to write the output data to an encrypted MySQL database instead of a plain text file, and we also needed to create the MySQL database. I focused on the creation of the database while my teammates altered the backend code.

1. MySQL Database

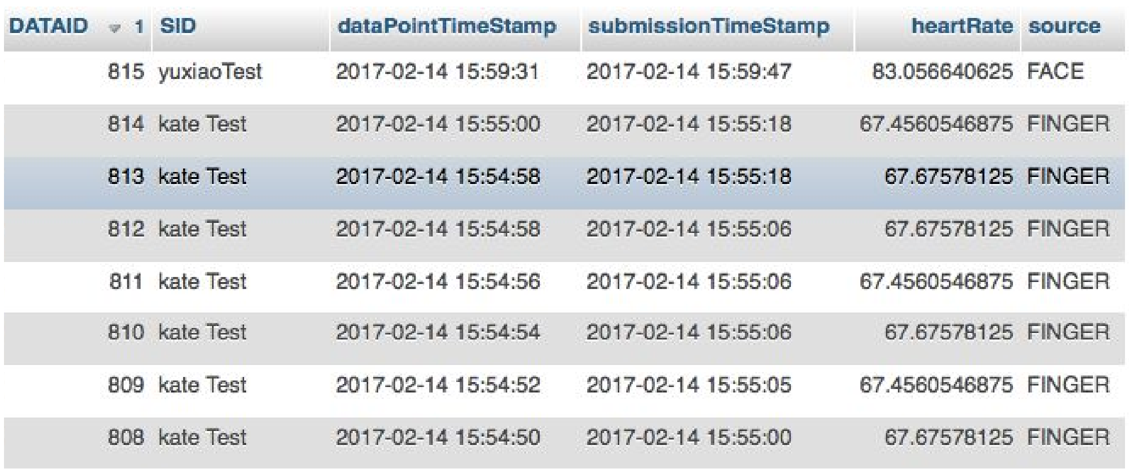
MySQL is an open-source database management system that runs on a server (MySQL, 2017). The way that a database management system operates is by allowing users to organize data efficiently into tables for easy retrieval later. Typically, multiple tables are created with each table representing a certain type of data. At least one of the columns in each table usually has some type of identifying information, such as a user ID. This way, data coming from the same user can be linked across tables.

In the case of the Running Coach application, one database was created with 12 tables - one table for each type of data being output from the app. Every table contained a column for the user ID in order to identify which phone client the data was coming from, as well as timestamps both for when the data was stored on the phone and when it was committed to the MySQL server. The 12 tables and a short description of the data in each are as follows:

1. User Parameters - Records information about the user's age, height, weight, gender, leg length, beginning and target running cadence, start and end date of their training program, and the steepness of the training regimen (alpha).
2. Run - Records information about the user's run, including current and target running cadence, average and target speed, average heart rate, and run start and end time.
3. Distance - Captures short distances travelled while the user is running. By adding all of these together, total distance run can be obtained.
4. GPS - Records the latitude and longitude of the user while running.
5. Cadence - Records the cadence of a user while running.
6. Speed - Records the speed of a user while running.

7. Heart Rate - Records the heart rate of a user either before, after, or while running, as well as whether the heart rate was taken by an external heart rate monitor or through the phone application using their face or their finger.
8. Battery - Records the battery percentage of the user's phone and whether or not the phone is currently charging.
9. Screenlight- Records whether the user's screen is turned on or off, and can be used to tell if the user is looking at their screen or not while running.
10. Energy Expenditure - Records how much energy the user is spending while running.
11. Survey - Records information from a post-run survey that all users can take. The survey asks questions about how accurate they thought the application was and how tired they were after the run.
12. Report - Records generic information about the application such as when someone opens or closes the app and when they start or stop a run.

Now, whenever a user runs using our Running Coach application, data from their run is collected in these tables. Figure 8 shows an example of the Heart Rate table collecting data from two of my teammate's phones. Note that the table matches its description in #7 above - each row contains a heart rate measurement and whether or not the measurement was taken from the user's face or finger using the phone application. It also contains a subject ID (SID), timestamp of when the data were taken (dataPointTimeStamp), and time the data were submitted to the server (submissionTimeStamp).



DATAID ▼ 1	SID	dataPointTimeStamp	submissionTimeStamp	heartRate	source
815	yuxiaoTest	2017-02-14 15:59:31	2017-02-14 15:59:47	83.056640625	FACE
814	kate Test	2017-02-14 15:55:00	2017-02-14 15:55:18	67.4560546875	FINGER
813	kate Test	2017-02-14 15:54:58	2017-02-14 15:55:18	67.67578125	FINGER
812	kate Test	2017-02-14 15:54:58	2017-02-14 15:55:06	67.67578125	FINGER
811	kate Test	2017-02-14 15:54:56	2017-02-14 15:55:06	67.4560546875	FINGER
810	kate Test	2017-02-14 15:54:54	2017-02-14 15:55:06	67.67578125	FINGER
809	kate Test	2017-02-14 15:54:52	2017-02-14 15:55:05	67.4560546875	FINGER
808	kate Test	2017-02-14 15:54:50	2017-02-14 15:55:00	67.67578125	FINGER

Figure 8. MySQL Table Collecting Data

Before beginning the study on human subjects, the database was tested for an extended period of time to ensure it was stable. Upon completion of this testing, the study began.

2. Pilot Study and Results

At the time this paper is being written, our group is currently conducting our pilot study. Although no definite conclusions have been drawn at this point, we can explain how the study is being conducted and the results that we have so far (Aranki et. al., 2017).

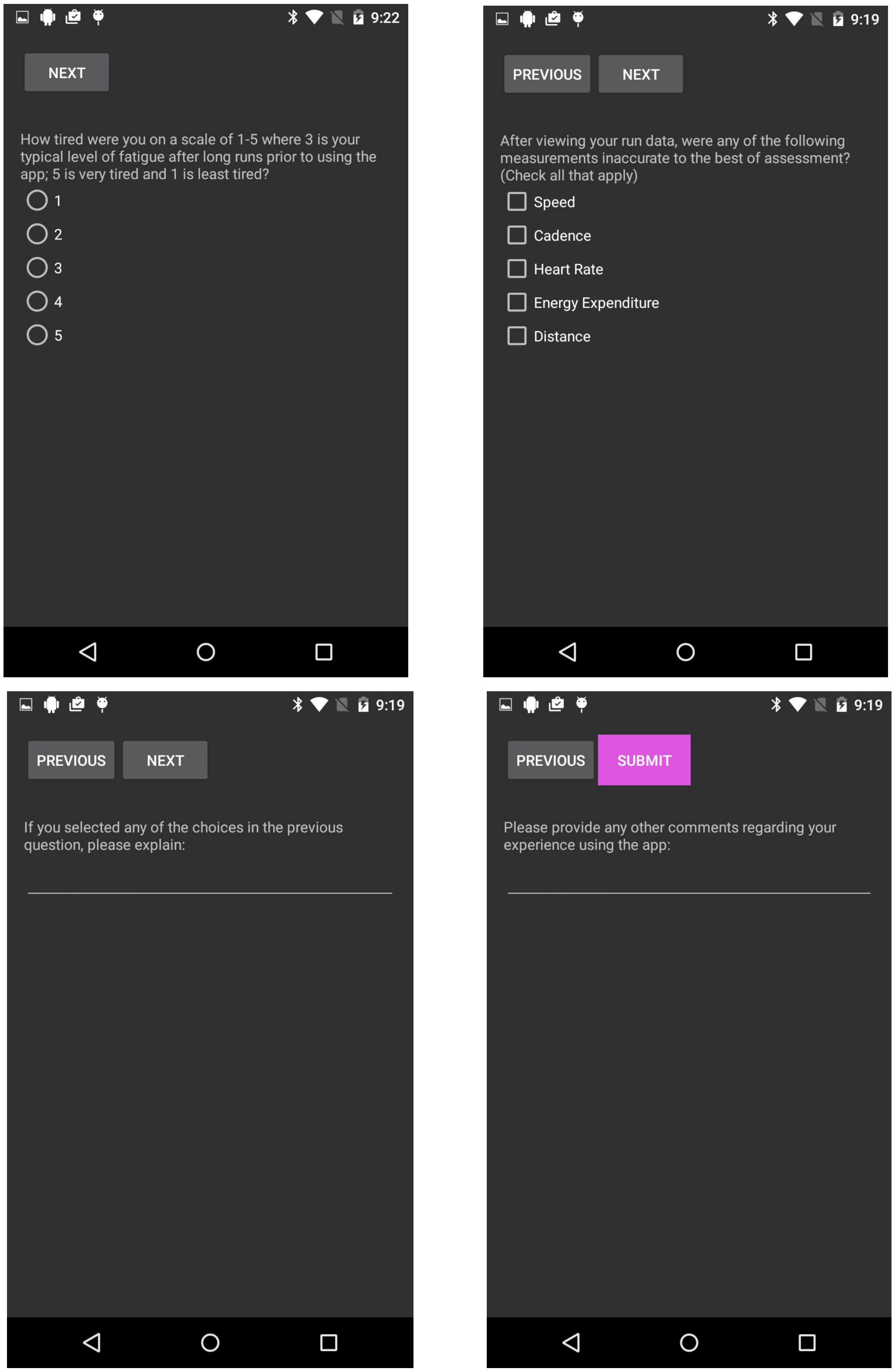
The goal of conducting pilot tests with the Running Coach application was to analyze the effectiveness of our telemonitoring framework. In particular, we wanted to know if it would positively affect a runner's training regimen and if our algorithms for various health measurements would be sufficiently accurate in real time running situations.

To conduct the test, six subjects were recruited from the Berkeley area, three males and three females. The subjects had to be long-distance runners, which we defined to be people who ran at least 5 kilometers per week, or ran at least once per week for over an hour. The subjects also had to be Android phone owners. They downloaded our application and wore a BLE heart rate monitor while they ran.

We have two ways of analyzing information from our subjects. First, after each run, we ask that the runner fill out a survey on the mobile application. The survey asks a series of questions pertaining to our application:

1. How tired were you on a scale from 1-5, where 3 is your typical level of fatigue after long runs prior to using the app, 5 is very tired and 1 is least tired?
2. After viewing your run data, were any of the following measurements inacurate to the best of your assessment? (Speed, Cadence, Heart Rate, Energy Expenditure, Distance)
3. If you selected any of the choices in the previous question, please explain.
4. Please provide any other comments regarding your experience using the app.

Asking if the user is tired will give us an indication as to if the application is challenging the user and giving them a difficult workout that encourages them to improve their runs. Asking if any of the measurements seemed inaccurate allows us to gauge how well our health measurement algorithms are working. Finally, asking for additional feedback may give us information about other ways in which to improve the telemonitoring framework or application. Figure 9 shows pictures of the survey screens on the Running Coach application.



How tired were you on a scale of 1-5 where 3 is your typical level of fatigue after long runs prior to using the app; 5 is very tired and 1 is least tired?

- 1
- 2
- 3
- 4
- 5

After viewing your run data, were any of the following measurements inaccurate to the best of assessment? (Check all that apply)

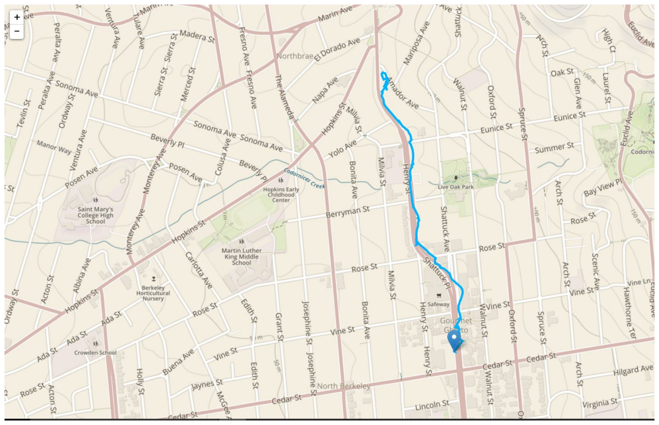
- Speed
- Cadence
- Heart Rate
- Energy Expenditure
- Distance

If you selected any of the choices in the previous question, please explain:

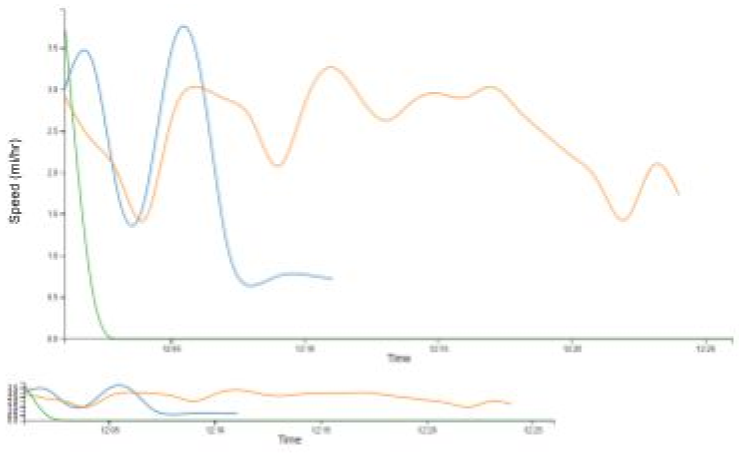
Please provide any other comments regarding your experience using the app:

Figure 9. Running Coach Post-Run Survey

The second way we plan to analyze information from the runners is by creating visualizations of the running data coming into our MySQL server. An undergraduate in our lab, Gao Xian Peh, has created an online dashboard to accomplish this task. Through this dashboard, we are able to select from our test subjects and view the various types of data we have collected from them. For example, we have created a map that shows where our test subjects have run and graphs that show information about their heart rate or cadence during the run. Figure 10 below shows two different ways our data is visualized, with test data taken by our team.



(a) A map that shows the path of the user's last run



(b) A graph that shows multiple user's speeds during their past run. Each line represents a different user. Various time windows can be selected from the smaller graph underneath.

Figure 10. Visualizations of Subject's Data

Visualizing the test subject's data allows us to see if their workouts improved over time and if they are achieving the goals our framework has set for them, and consequently if our telemonitoring framework is successful. It also allows us to see if there is any unreasonable data being received and to fix any potential bugs in the framework.

So far, our pilot study is showing early signs of success. Our MySQL database and dashboard website indicate that subject data is being received successfully. There have been no reports of app crashes by the subjects, and no signs of data loss on the server end. A brief glance at some of the comments users are making after their run indicate that some have had problems with the accuracy of our heart rate algorithms before and after their run, so this is one potential area of improvement.

Upon conclusion of the pilot study, test subjects will fill out a usability and acceptability survey that will give our team more insight into how they viewed our Running Coach application. We hope to use these comments as well as the data we gathered to further improve upon our application and telemonitoring framework.

Project Conclusions and Future Work

My contribution to the Berkeley Telemonitoring team came in the form of expanding the Bluetooth Low-Energy protocol and making additions to and testing the Running Coach application. The advances I made in these areas give the framework a wider functionality and allow for more future applications for our users. However, for anyone wishing to expand upon this framework there is still much more work that can be done. Particularly in the area of Bluetooth, the protocol could be extended to include additional devices such as fitness trackers or oximeters. The protocol could also be modified to include the IEEE Bluetooth standard in addition to the Bluetooth Low-Energy connection that the framework currently supports. With regards to the Running Coach application, there is more work to be done upon completion of the pilot study. Once more feedback is received from our test subjects, it will need to be incorporated into the framework.

Overall, our team has successfully achieved our goal of expanded the telemonitoring framework to include additional Bluetooth, privacy, and server features. For those in the medical or coaching fields, our framework provides an alternative way to interact with patients and clients. On a large scale, this framework has the potential to reduce costs and improve healthcare for millions of people.

Chapter Two: Engineering Leadership

Engineering Leadership Introduction

We now turn from technical contributions to a new topic of engineering leadership, with the goal of viewing our project from a business perspective. For this portion of the report, we consider bringing the Running Coach application to market. With this application comes several questions that need to be answered regarding where our product fits into industry, how we would market it, and social implications.

Industry Analysis

Today's world is one that is increasingly health conscious. Over 50% of Americans exercise on a regular basis, and this number is on the rise (Mintel, 2014). In addition to this, over 3.3 million fitness bands and exercise trackers were sold between March 2013 and April 2014 (Kovar, 2016). Based on these statistics it is clear that a change in global health patterns is on the horizon, and Berkeley Tele-monitoring believes that it can help by entering into the sports coaching industry.

The sports coaching industry is defined as "consisting of establishments that offer instruction for athletic activities to groups or individuals" (Masterson, 2016, p. 5). This industry tends to target selling towards adolescents and young adults from ages 20-29. Recently, it has been doing quite well, with an average annual revenue increase of over 3.0%. This is believed to be due to an increase in sports participation. (Masterson, 2016).

There are many competitors in the sports coaching industry, and we will take a look at a few of them. The biggest subset consists of human personal trainers and fitness coaches at gyms and athletic facilities. However, a fast-growing portion of this industry is becoming sports coaching software technology, and this is where our product has the most competition. These platforms are known to give teams and individuals that adopt them a competitive advantage in the form of athletic improvement. Major companies in this industry include Coach.me, AthleticLogic, and Coach's Eye (Tiwari, 2015). Upon reviewing these products, they appear to gather a lot of data, whether through filming an athlete doing a sport or collecting sensor data. Many of them involve recording a training session and allowing an athlete to go back and view their workout later. Where these products fall short is that they fail to provide real-time feedback to the athletes, and they use limited sensors to gather their information. Our product seeks to improve upon these downfalls by providing real-time suggestions to athletes during exercising and using a wide variety of sensors such as accelerometers, GPS, and cameras.

Overall, our final product would fit well into the sports coaching industry. Within this industry, there are several major competitors, some of which are well established corporations. However, our team believes that we have a unique value proposition in creating a framework instead of focusing on a physical wearable device, as well as providing real-time feedback to our users for how to improve their workouts. These advantages described above play a large role into how we will choose to market our product.

Marketing Strategy

Our primary clients are long-distance marathon runners. In recent years, marathon events have continuously attracted a lot of professional participants and public attention. According to the Running USA Annual Marathon Report, there were 1,100 U.S. Marathon events and more than 500,000 finishers each year from 2014 to 2015 ("Running USA," 2015). This means that there is a large number of clients in the long-distance running market. A Marathon runner's performance heavily relies on their training, and therefore, a comprehensive training plan plays an important role in his/her good competition outcome. However, current personal one-on-one training programs are expensive and without many flexible schedules. For example, an in-person coach training program in COACHUP is about $100/session ("Running Coaches near San Francisco, CA," 2016). Although these human coaching programs can customize the training plans, they are not able to report real-time feedback regarding performance metrics if the runner does outdoor training. However, our smartphone coaching application can quickly generate a personalized training plan for a runner for a low price. This gives our product a competitive advantage in the market.

As mentioned in the industry analysis, there are many sports coaching competitors in this market. However, we would market our tele-monitoring Android platform framework by showing how it is distinguished from the rest. The IBISWorld Industry Report 61162 showed that there were about 123,094 businesses in the sports coaching field (Masterson, 2016, p. 4). Many of these businesses are one-on-one sports training camps and schools that offer instruction in athletic activities to groups or individuals (Masterson, 2016, p. 2). Additionally, the current health and fitness apps are designed for general sports or fitness purpose instead of specializing in long-distance running training, so the measurement of performance may not be as accurate. For example, the Garmin FR620 was an advanced running GPS watch (Rainmaker, 2013). But, it could not support getting any dynamic running information (Rainmaker, 2013). Our framework design allows implementing user interfaces, data acquisition, data storage, and proper security and privacy mechanisms using APIs (application programming interfaces) through the sensors in the Android based smartphones (Aranki et al., 2016). To be specific, our design will customize the training plan by monitoring the user's heart rate based on the cadence (steps per minute) using the smartphone. This method is specifically designed for long-distance runner training and is easily implemented without compromising human coaches' schedules. It would bring more convenience to the users than traditional human coaching.

In summary, our team's marketing strategy involves analyzing the current holes in the sports coaching industry and identifying how our product can fill those needs. In considering these needs, however, we must also consider their overall social implications.

Social Context

One of the biggest social impacts we see our project having is in the world of health care. While many countries have regulations for the health care of their people, it still remains a growing concern. Particularly in the United States, the increasing cost of health care has lasted for many years. As indicated in the report of the 2016 Milliman Medical Index, "The rate of increase is still well above growth in the consumer price index (CPI) for medical services, and far surpasses the average 2% annual increase in median household income between 2004 and 2014." (Girod et. al, 2016, p. 1). One of the main ideas given to reduce this cost is to reduce the readmission rate for chronic health conditions (ObamaCare Fact, 2016). The high readmission rate means people have to go back to medical facilities to consult with their doctors several times, increasing the cost of their health care.

Since our project provides the possibility that doctors and patients can communicate remotely without going to medical facilities multiple times, it could significantly decrease the readmission rate, which may reduce health care costs. Our project could potentially change the health care structure over time. Instead of going to medical facilities multiple times, doctors could receive health data from their patients via our Android framework, analyze it, and give feedback if necessary. This would save patients unnecessary trips to the doctor's office.

Outside of healthcare, our Android platform based project also follows a technological trend. According to a research paper, the Android platform market share of the smartphone was 45.4% worldwide in 2015, while secondary platform IOS only had 15.3% of the market share - three times less than Android (Burguera, 2011). With such a huge market share, our Android platform based project could have many more potential customers than any other platform. Also, the market share by Android could increase in the future with the new release of Google Android features later this year (Anirudh, 2016). Furthermore, the trend of increasing use of smartphones has continued for many years. The sales growth rate of the smartphone in China for 2015 is 52% more as compared to 2014, and this rapid growth is projected to continue in following years (Baoling, 2014). This means that not only will the market share of Android increase, but the total number of smartphone users will go up as well.

By following the current social trends, we can conclude that there is potential for wide use of our product. This will help more patients have a better healthcare experience with a lower cost. By taking advantage of Android's large market share of smartphones, our project could potentially change the traditional health care methods for the many people in the future.

Engineering Leadership Conclusion

In brief, our Extended Platform for Android Tele-monitoring product should perform well in the sports coaching industry due to its unique value proposition of providing real-time feedback through an Android framework. Our major buyers are long-distance runners, and marketing patterns show this group increasing every year. As a good competition outcome highly relates to a good training plan, long-distance runners will be willing to spend less money buying this smartphone based application to achieve the same high-level competition results with coaching by human trainers. Finally, in a social context, we believe our product can take advantage of the large Android market share and potentially create a positive impact on healthcare worldwide.

References

American Cancer Society. (2017). Cancer Treatments and Side Effects . Retrieved March 10, 2017, from https://www.cancer.org/treatment/treatments-and-side-effects/physical-side-effects/eating-problems/weight-changes.html

Aranki et. al. (2017). RunningCoach – Cadence Training System for Long-Distance Runners. Paper to be presented at *2017 Health-i-Coach Workshop*. Manuscript in preparation.

Aranki. et. al. (2016, June 27-29). A Telemonitoring Framework for Android Devices. Paper presented at *2016 IEEE First Conference on Connected Health: Applications, Systems and Engineering Technologies*. doi: 10.1109/CHASE.2016.28

Baoling Li, Haiyan Fu (2014, June). Research on the developing trend and strategies for mobile marketing. Retrieved October 16, 2016, from http://ieeexplore.ieee.org/document/6874110/

Bluetooth. (2017). GATT Services. Retrieved March 12, 2017, from https://www.bluetooth.com/specifications/gatt/services

Burguera et al (2011, Oct 17) . Crowdaroid: Behavior-Based Malware Detection System for Android. Retrieved October 16, 2016, from http://delivery.acm.org/10.1145/2050000/2046619p15burguera.pdfip=136.152.208.249&id=2046619&acc=ACTIVE20SERVICE&key=CA367851C7E3CE772E3158474DDFAA3F102E4D4702B0C3E38B352E4D4702B0C3E38B35&CFID=680488789&CFTOKEN=48860795&__acm__=1476255122_c589c038cdedbe7fc0d9761e1015a684

Centers for Disease Control and Prevention. (2015, May 15). National Diabetes Statistics Report. Retrieved March 1, 2017, from https://www.cdc.gov/diabetes/data/statistics/2014statisticsreport.html

Centers for Disease Control and Prevention. (2016, February 23). Chronic Disease Overview. Retrieved March 1, 2017, from https://www.cdc.gov/chronicdisease/overview/

Dhebar, Anirudh (2016, Aug). Bringing new high-technology products to market: Six perils awaiting marketers. Retrieved October 16, 2016, from http://www.sciencedirect.com/science/article/pii/S0007681316300830

F. (n.d.). ObamaCare Facts: Facts on the Affordable Care Act. Retrieved October 16, 2016, from http://obamacarefacts.com/obamacare-facts/

Girod et. al(2106, May 24). 2016 Milliman Medical Index. Retrieved October 16, 2016, from http://www.milliman.com/uploadedFiles/insight/Periodicals/mmi/2016-milliman-medical-index.pdf

IEEE Standards Association. (2014, August 21). IEEE Health informatics: Personal health device communication. Retrieved March 1, 2017, from https://standards.ieee.org/findstds/standard/11073-20601-2014.html

Mani et al. (2015, May). Expanded Tele-heatlh Platform for Android. Retrieved March 1, 2017, from https://www2.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-110.pdf

Masterson, R. (2016, June). IBISworld Industry Report 61162: Sports Coaching in the US. *IBISworld*. Available from http://clients1.ibisworld.com/reports/us/industry/default.aspx?entid=1542

Meystre, S. M. (2005, February 11). The Current State of Telemonitoring. Retrieved November 11, 2016, from https://www.ncbi.nlm.nih.gov/pubmed/15785222

Mintel. (2014, October). Exercise Trends Market Report. Retrieved October 14th, 2016 from Mintel Academic database.

MySQL. (2017). About MySQL. Retrieved March 10, 2017, from https://www.mysql.com/about/

Nicholls, M. G., & Richards, A. M. (2007, April). Disease monitoring of patients with chronic heart failure. Retrieved March 1, 2017, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1861485/

Partnership to Fight Chronic Disease. (2009, August 10). The Growing Crisis of Chronic Disease in the United States. Retrieved February 28, 2017, from http://www.fightchronicdisease.org/sites/default/files/docs/GrowingCrisisofChronicDiseaseintheUSfactsheet_81009.pdf

Rainmaker, D. (2013, November 4). Garmin Forerunner 620 In-Depth Review. *DC RAINMAKER*. Available from http://www.dcrainmaker.com/2013/11/garmin-forerunner-review.html

Rooney, E. (2016). Telemonitoring Benefits. Retrieved November 11 2016, from http://www.telemonitoringni.info/patients-carers/benefits-and-patient-stories/

Running USA. (2015). 2015 Running USA Annual Marathon Report. Available from http://www.runningusa.org/marathon-report-2016?returnTo=main

Sarver, et al. (2016, May). A Tele-monitoring Solution to Long Distance Running Coaching. Retrieved March 1, 2017, from https://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-85.pdf

Tiwari, R. (2015). Sports Coaching Platform Technology Market. Retrieved October 13, 2016. Available from http://www.prnewswire.com/news-releases/sports-coaching-platform-Technology-market-worth-864m-by-2021-561558701.html

Townsend, K. (2017). Introduction to Bluetooth Low Energy. Retrieved March 12, 2017, from https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction

Wearable Technology - US - December 2015. (2015, December). *Mintel*. Available from http://academic.mintel.com/display/757616/?highlight