Mobile Application-IoT Based EKG Monitoring System

Poornima Eshwara*, Hamid Shahnasser

Software Developer at Libman Co., 201 E Madison St, APT2, Arcola, IL, San Francisco State University, 1600 Holloway Ave, San Francisco, CA, USA.

* Corresponding author. Tel.: 415-966-6029; email: poornimaeshwar10@gmail.com
Manuscript submitted October 25, 2018; accepted January 22, 2019.
doi: 10.17706/ijcce.2019.8.2.39-49

Abstract: ECG is the most commonly performed cardiology test that provides vital information to understand the state of person's heart condition. It essentially trances the electrical activity of the heart as it pumps blood to rest of the body and is very useful for determining the state of the heart and any symptoms. Conventionally patient visits the clinic for the ECG to be taken and is given report in a day or two. The whole process is both time consuming and tedious for the patient. With the latest development in Internet of Things, Cloud technology and reliable and faster data transmission, this process can be made lot more convenient. This paper proposes a cost-effective, remote system to read such ECG data of a patient via a handy sensor-Shimmer sensing device and send it to an android device via Bluetooth. The android device in turn sends this data to the cloud for storage and analysis and is then transmitted to a doctor's android device for observation. Unique feature of the proposed unit lies in the fact that it provides a singular platform wherein the patient is directly connected to his/her healthcare provider for transmitting the ECG data with little or no delay. Apart from ensuring secure transmission of ECG data from patient to doctor, this channel between the patient and the doctor lets them communicate with each other. It is used to receive any valuable feedback or guidance from the doctor and gives opportunity to constantly monitor the effects or symptoms and responses to medicines that the patient is undergoing. The unit makes use of several technological advancements in cloud such as data processing, real-time data streaming, security, user account sync while making all these available remotely through the android and the sensor device.

Key words: Android application, body sensors, ECG, EKG, health monitoring system, cloud processing, shimmer, amazon web services, bluetooth, kinesis, real-time data streaming.

1. Introduction

Electorocardiagraphy (ECG or EKG) is the process of recording the electrical activity of the heart over a period of time using electrodes placed on the skin. This electrical signal measured across specific regions when plotted as Voltage v/s time gives a repetitive pattern of graph with peaks and dips. Depending on number of electrodes/wires used, the resulting graph can be either 3-lead, 5-lead or 12-lead ECG. 3-lead gives minimalistic information and is generally used while in transport, 5 lead ECG is generally used in ICU's as it gives essential information needed for constant monitoring. 12 lead gives a very detailed report on each sections of the heart. Irrespective of which lead ECG is used, resulting graph will have same distinctive regions of peaks and dips which are given names as p, q, r, s and t. A heart surgeon can understand the state of the patient by carefully analyzing these peak, dips and the distance between each other. Presently reading such ECG is carried out in the hospitals or clinical labs in general, adding an overhead of visiting them.

This project proposes a way to capture and transmit 5-lead ECG on android platform. Android platform is chosen, as a cellphone is something every person will own these days. This means, the only additional requirement is a device to get the ECG and two android applications, one each for the patient and doctor. In this project, Shimmer sensor with 5 wires is used, that are attached to certain places on patient's skin to capture the electrical changes in heart and sends it to android phone through Bluetooth. Android application is developed to receive the ECG readings on doctor's phone with the following features.

- Display ECG waves on mobile device for observation
- Store ECG over time for future analysis.
- Send a notification to patient asking for his/her ECG measurements.

Data from multiple patients can be handled through an account-based system where each user is given a unique ID, as an additional feature, other details of the patient about his/her health conditions, body temperature from a temperature sensor, previous visits to the doctor, other test results can also be linked to it. Making it a one-point access for a quick review about patient's state.

2. Current Technologies and Challenges

There is a lot of research going on in improving existing technology for developing an efficient Health Monitoring System. There are several commercial health tracking devices like fitbit, Jawbone, Misfit but their primary focus is on general features like tracking sleep cycle, physical activity, Calorie consumption and are not so accurate for applications like ECG. There are also several devices online which make use of latest developments in IoT and offer full-fledged system like Mysignal [1] with API support, Cloud storage and development platform. Each of these systems come with its advantages and disadvantages. [2] provides an excellent reference of the challenges faced and the areas that need improvement. [3]-[5] shows various aspects to be considered for a remote ECG systems like Bluetooth based ECG system and proposal of a GPRS- based tracking system for future applications [3], data compression for fasted transmission [4], while [5] offers the same with Body Area Network and shows that we need at least 100 seconds of data for obtaining required accuracy for bio signal processing and [6] uses a battery less microsystem implanted in mice for real-time monitoring. [7], [8] compares different sensors used in taking measurements and their influence over accuracy. [9] and [10] present research and observations of using Shimmer Sensing device for ECG reading. While [9] provides a non-real-time based data transmission, [10] presents a way of processing and streaming raw and real-time data on Contiki OS using Cooja Simulator, MSPSim, Matlab and Labview. The proposed Mobile-based ECG monitoring system aims at offering a platform that will consolidate information of other healthcare details with ECG data, with latest features in Cloud computing and Android development like security, user authentication, data validation and processing for understanding ECG signals. As we can see, there are number of devices that are mobile and provide real-time generic healthcare information regarding various aspects of one's bodily functions like sleep cycle, calorie consumption, calories burnt. There are also various units for monitoring the ECG of a person. This project combines both the features and offers a platform for monitoring ECG like other biomedical signals while still making it a remotely available device. It also gives a comprehensive detail of all the AWS and Shimmer components used in the process.

3. Project Overview

A typical IoT based health monitoring system will essentially have three steps in common: data acquisition, data transmission, and processing and a data visualization unit for end user observation. Time-intensive operations like data analytics on streaming data might not be supported by smaller devices like Cellphone so cloud is a good place to carry such operations without impacting the latency of

transmission. Fig. 1 below shows a general overview of one such patient monitoring system hosted on a cloud.

As discussed earlier, the project consists of 3 major parts.

- Android application at Patient's end, also referred as User app in the paper, for data acquisition from the sensors.
- Account and data management in Cloud using AWS for data transmission from user to cloud.
- Finally, the Android application at Doctor's end for receiving information from cloud and data visualization.

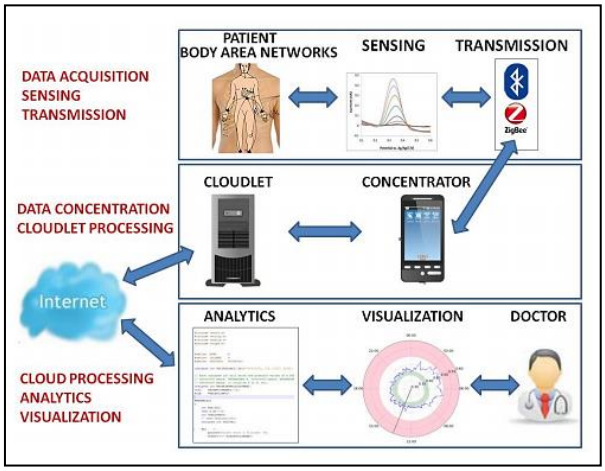

Fig. 1. A typical cloud-based mobile application enabled ECG system to transmit patient's data to doctor/ clinical expert.

3.1. User/Patient Android Application

This is app mainly consists of two parts: Interaction with the Shimmer device over Bluetooth using Shimmer SDK and secondly, data logging and data transmission to cloud using AWS Android SDK.

The user can also view the ECG data on graph before initializing the data transmission. And since the project is all about the promoting the welfare for the user, as an added feature it also displays a randomly picked quote for the day.

3.2. Data Handling at AWS

AWS offers a NoSQL database service– DynamodB which is fast and scalable offering consistent, single-digit millisecond latency at any scale. It also integrates with AWS Lambda for creating triggers and a means of fine-grain access control using IAM-Identity Access Management. This becomes especially important since we are dealing with sensitive healthcare information. Another advantage of using cloud service like AWS is that it is easily scalable which is extremely useful on a long-term as user base grows and one the administrators are charged only for the resources used.

3.3. Doctor Android Application

This is relatively simpler of the three applications. This essentially is another Android application that will be used by the Doctor. The app receives ECG data from AWS's DynamoDB and displays the graph for observation. It also helps in keeping the patient and the doctor connected through a messaging achieved by Simple Notification Service offered by AWS. Using this, the doctor gets a notification every time the ECG data is sent and if the patient is expected to send his/her ECG but failed to do so in timely manner, a reminder can also be sent to the patient requesting them to send ECG for observation.

The Fig. 2 below depicts a block diagram of this project on a high level. Each of the above modules are discussed in detail in the following sections.

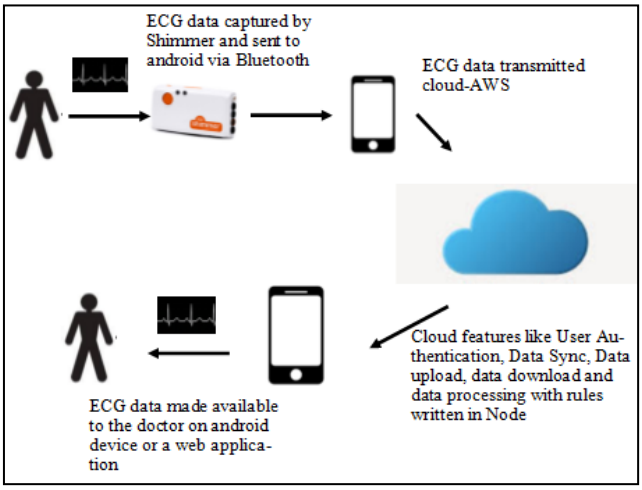

Fig. 2. Flow of ECG data as it is transmitted from patient to the doctor through sensor, android, cloud and android again to be observed by the doctor.

4. Data Acquisition

Shimmer Sensing device for ECG provides a configurable physiological signal through its 5-lead Electrography. This ECG measuring unit is chosen because it provides highly accurate and scientifically reliable raw data to allow complete control over capture and interpretation of sensed data in real-time.

Shimmer sensor has five color coded electrodes/wires namely as shown in the Fig. 3 and listed below:

- White-Right Arm (RA)
- Black- Left Arm (LA)
- Green-Right Leg (RL)
- Red- Left Leg (LL)
- Brown-V1 toV6 (Vx)

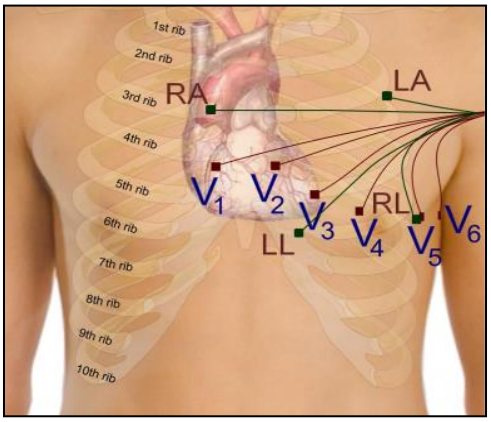

Fig. 3. Electrode placement for ECG measurement.

Above 5 electrodes are connected to user's body in corresponding place to provide a four lead ECG graph. Three bipolar and Unipolar Leads as described below:

Of these 4 leads, Lead I, II, III are bipolar leads with Lead I(LA-RA) measured from Right Arm to Left Arm position, Lead II(LL-RA) measured from Left Arm to Right Arm and Lead III(LL-LA) measured from Left Leg

to Left Arm. The forth Lead- Unipolar Lead can be measured by placing the electrode on one of 6 positions, V1 to V6 as per user requirement. Once the device connections are set up, required data rate can be set ranging from 125Hz to 8000Hz. A data rate of 125-200Hz offers a good tradeoff between the data accuracy and speed of transmission. But 512Hz is usually recommended for medical applications.

4.1. Interaction with Shimmer Unit

Shimmer Sensor offers Java/Android API that includes Shimmer Driver and Shimmer Android Driver Library that provides rich set of features and methods and exposes various functionalities of the device to allow developers to build applications faster. Here is brief overview of components of the Shimmer Android API that are primarily used in the development.

Shimmer Object class is the main class where the device's core functionalities are defined. A class called Shimmer extends Shimmer Object class and offers first hand interface between android application and the device. A block diagram of all the main components in the code communicating with each other is shown in the Fig. 4.

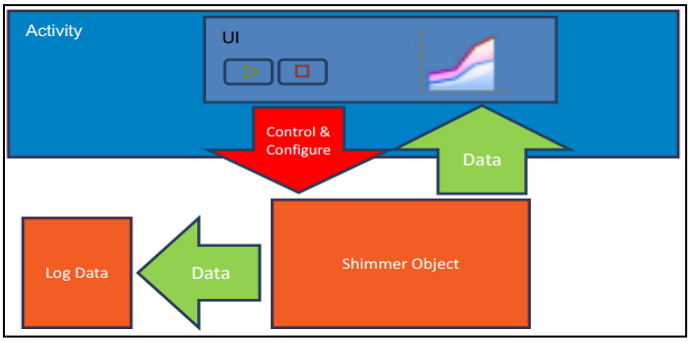

Fig. 4. Individual modules in the patient app for capturing the ECG data from shimmer.

Sensor data from the device is available in the form of a custom-made data structure called Object Clusters which is built on Multimaps. Multimaps are the data structures like Maps in Java with key-value pair except that each key can have multiple values in multimaps. So Object Clusters are multimaps with PropertyClusters as keys with Formatcluster as corresponding values. Format Cluster is a collection of data format (which can be either calibrated or raw), data unit and the values. In the ECG application, calibrated formats are to be used for logging and plotting the graph, unit is mVolts and measure values are stored in double datatype. PropertyClusters basically holds different lead names that are measured by Shimmer, namely: LA-RA, LL-RA, Vx-RL depending on which of the positions are used from V1 to v6. Timestamp is also saved under one of the PropertyClusters for plotting the values on graph.

As for the graph, the values are plotted using an external library called AndroidPlot. This helps the patient to view the ECG graph first hand on real-time. Using this graph, the user can observe if the graph is stabilized by staying still and relaxed. Once a stabilized graph starts appearing on the android screen, the user can click on the send button on the screen to start the transmission and stop it after few seconds. Typically, waveform captured for a duration of 10-20 seconds is printed on a sheet of paper for observation, same can be followed, or one can send it for a longer duration too.

Android application involves lot of time-consuming operations like receiving stream of data coming from Shimmer, sending it to cloud, logging it in the local drive. Since they need little or no interaction with the user, these are run in the background using features like Handlers, AsyncTask, and services available in Java/Android.

A new thread is created when a shimmer unit is connected to android so that all the tasks related to

shimmer can be carried out in the Shimmer thread without blocking the UI tread. Handlers are used for communication between both the threads. An intent service is used for logging and streaming data out to AWS.

4.2. Interaction with AWS

To facilitate seamless interaction with AWS service with Android applications, AWS Mobile SDK is used. It is an open-source software development kit providing several libraries, samples and documentation for using their service. In this project, Cognito, Lambda, and DynamoDB are used.

User login, registration for new users and account management is done by Cognito Service. It offers a feature called User pool in which user's details are saved. This user pool has Pool ID and Application ID associated with it which is used for authentication. Now once the user is logged into the account, more attributes like pre-existing medical conditions, nature of treatment, list of associated healthcare providers can be added. All these details are automatically in sync with AWS and gets reflected instantaneously. Since the data from android app must be stored in DynamoBD-the noSQL data base, the user must be granted permission to the service and a write access to the table. This can be done using federated User Identities.

Any request to authorize a user to access any other service (DyanmoDB and Lambda in this case) is managed by Identity pool. Pool ID and the Application Client ID from User pool should be updated in the Identity pool. Dynamo DB is a non-relational database that stores data in the form of tables. Each table is a collection of number of items. Items can be thought of as a row in a list of data like in excel sheet. Item can comprise of group of attributes; attributes are like column in a list. In our case, items are set of sensor reading taken at one instance. Attributes are the different sensor readings like LA_RA, LL_RA. Each item in a table must be uniquely identifiable which is taken care by Primary Key.

Three main concepts needed for programming DynamoDB are DBClient, Object mapper, and a mapping class. Mapping class is used to define the structure of the table into which the data will be loaded. In our app, a class named DBUpload is written defining each of the attributes, primary key with getter and setter methods. Object Mapper called Dynamo DBMapper class sets up communication between DBUpload and the database. This map expects Amazon Dynamo DBClient as an argument for instantiation. The credential information from Identity pool in Cognito is passed to this DBclient. A part of the code for configuring database connection is shown in Fig. 5. Now using the mapper class all the operations like data insert, delete, update and access can be done from the android application. For our app, we only need it to insert the values in the database which is done in AsyncTask by using *save()* method after the values for all the attributes are updated. A snapshot of the code for invoking the database service from the android app is given below.

```
public static void setDdb(AmazonDynamoDBClient ddb) {
    MultiShimmerTemplateService.ddb = ddb;
}
private static AmazonDynamoDBClient ddb;

public static AmazonDynamoDBClient getDb() {
    return ddb;
}
DynamoDBMapper mapper = new DynamoDBMapper(ddb);
Logging.DBUpload dbUpload = new Logging.DBUpload();
```

Fig. 5. Code snippet for setting up DBClient, DBmapper and mapper class in the app before initiating the data transmission.

4.3. Device Set Up and Connections

This section shows steps involved in setting up the device for transmitting ECG signal. Install and open

user's app. The welcoming screen will be a login page. If the user is already registered, enter username and password and click submit. For new users, the Sign-up option is also available which takes you to the registration page where basic details like names, email, phone number, PatientID are noted down and a verification code is sent to either email or phone number for confirmation. Once confirmed, the user is taken to Patient profile page. By clicking on ECG button user is taken to another activity to connect with Shimmer device. Turn on the Shimmer device that is pre-loaded with LogStream firmware. Here is a picture of Shimmer device with the running android app in Fig. 6 and login page is shown in Fig. 7.

This program is basically for setting up the Bluetooth connectivity for configuring and receiving the sensor data from shimmer to android. Now, on android device click on the button to connect to the shimmer, this needs Bluetooth to be turned-on in android device which is also prompted to user and once the phone and shimmer are paired, connect the electrodes between the patient and the Shimmer. Select 'start streaming' option on android to start receiving ECG data from Shimmer to android and observe the corresponding graph on android screen.

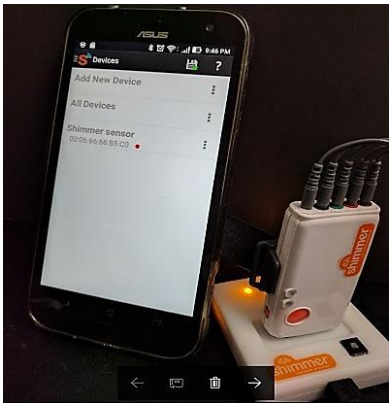

Fig. 6. Shimmer device connected to the android by Bluetooth and ready to be connected to the patient for reading ECG.

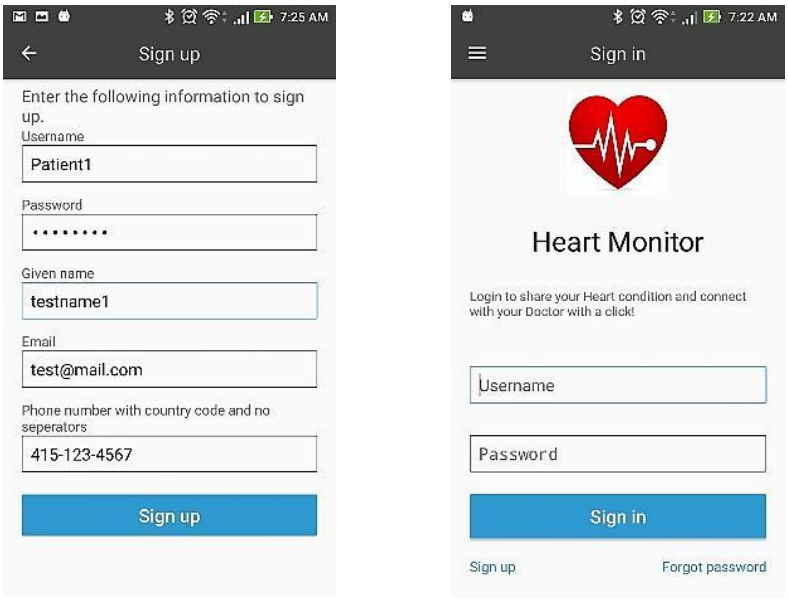

Fig. 7. Screenshot of user login page on the left and the registration page on the right.

Once a satisfactory waveform observed, start transmitting ECG to DynamoDB on AWS. Once it is sent, stop the data stream between shimmer and android and disconnect the device. This whole set up will take

up to 10 minutes. If patients need help with the set up and electrode connections, there is a help button to guide user through the steps involved. If the electrodes are not connected in the right configuration, resulting waveform will be wrong too. To avoid that, few computations are done on the go the just before initiating the transmission. It basically looks for anomalies in the waveforms and asks the user to verify connections and sends the data only after getting the confirmation. Once a satisfactory waveform observed, start transmitting ECG to DynamoDB on AWS. Once it is sent, stop the data stream between shimmer and android and disconnect the device. This whole set up will take up to 10 minutes. If patients need help with the set up and electrode connections, there is a help button to guide user through the steps involved.

If the electrodes are not connected in the right configuration, resulting waveform will be wrong too. Toavoid that, few computations are done on the go the just before initiating the transmission. It basically looks for anomalies in the waveforms and asks the user to verify connections and sends the data only after getting the confirmation.

5. Data Transmission

Cloud platform chosen for the data storages and processing is Amazon Web Service. It is a cloud-based platform supported by Amazon that helps quickly develop high-quality apps and grow user base. AWS has several features like Cognito for User account maintenance, DynamoDB for real-time data streaming. Another service called Kinesis also offers real-time data transmission but has lower latency compared to DynamoDB for read-write operations, hence the DynamoDB database is chosen for the ECG sensor data transmission through the cloud.

Streams of ECG signals read by Shimmer at a given point of time are sent to database and transmitted to the doctor again for inspection. The capacity requirement for the database must be specified during the initial set up. Dyanmo DB throughput capacity is measured in terms of Read and Write Capacity. For this project provision read capacity of 10 units and write capacity of 5 units is used. Each item read/written can be up to 4KB in size and since each sample of ECG is within this limit and read capacity is 5 units, it can read up to 20KB of data in one second. Similarly, it can read up to 40KB of data in a second. While storing the data in the database, timestamp is used as the primary key to make each item uniquely identifiable. So, there will be five attributes for each item, one for timestamp and four for each of the leads taken at the given timestamp.

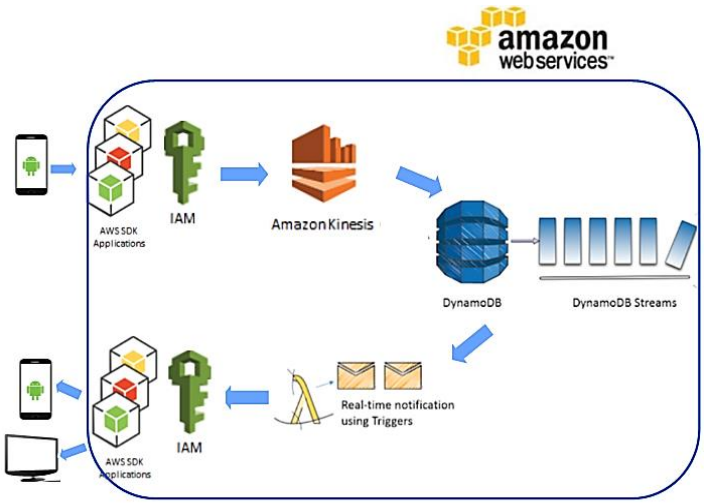

Fig. 8. High level overview to show the data flow and the cloud services used in the process.

Once the patient sends ECG the doctor is notified of the data via SNS notification. This is made possible by

using Lambda functions. Lambda is a server-less compute service. It is a very powerful tool with number of application. It can be used for data processing, real-time file processing, stream processing, extract, transform, and load data. The logic can be written in Java, Node.js, Python or C# and uploaded into the AWS console. Node.js is used in this project to trigger SNS notification whenever the patient sends the ECG data to the doctor. The Fig. 8 below shows the block diagrams of services of AWS that are used and the flow of data as it travels from patient's app to the doctor's app.

6. Data Visualization

Android application that is used by the Healthcare provider will receive a notification whenever there is a new ECG transmitted. Then if the doctor can go the android application and access the graphical display of ECG data associated with each patient as shown in the snapshot below.

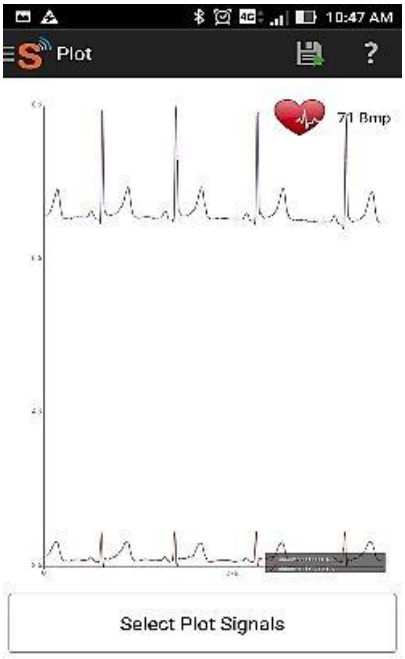

Fig. 9. Screenshot of ECG signals distinctly showing p,q,r,s,t segments for inspection.

Along with the ECG signal other information like (Fig. 9), previous health condition, prevailing disease information, undergoing treatment can also be included with each patient's account that might be helpful in better diagnosis and analysis of ECG. Since this application should receive the data from the cloud, Android AWS SDK is used here as well. For fine-grain control over the data access, this also needs a user login page similar to the patient's application but the user pool under which the login details are different. While the first application login details are registered under Patient user pool with only write access to the database, user registering in this application are registered under another user pool of doctors. These users are granted read only access to the database. When the doctors read the data, the items from the database can either be moved to a file-based storage system like S3 or the be left in the database for future access.

7. Conclusion

Tremendous advancements seen in the field of IoT, sophisticated miniature products, and better processing ability, have opened doors for countless applications to take advantage of it. Medical application is one such field that demanded this level of sophistication in terms of faster data transmission and processing for real-time data transfer and higher security. A unit such as this one can be set up in remote

places of developing countries with limited access to healthcare facility and a medical examiner.

On the other hand, it can also be used as means of constantly monitoring one's own physical condition without having to visit a doctor every time. After collecting enough data over a long period from variety of subjects, abnormalities can be recognized by using techniques like analysis biomedical signal, machine learning and pattern recognition, and a notification can be sent to the doctor only when one needs medical attention. It should be noted that, with just one glance at ECG, it is not possible to predict if a person is going to have a heart attack. In fact, American Heart Association recommends that one should get ECG checked up only there are other symptoms associated with Heart. This is important because a normal person's ECG can also result in a misleading diagnosis, followed by extensive and often unnecessary treatment. In any case if one shows any symptoms of having cardiac problems ECG is the most important means of understanding the state of the heart and hence remains to be one of the most extensively researched fields of Biomedical world.

Appendix

All the code involved in the developing this project is available in the following GitHub link: https://github.com/peshwara/AWSLoginShimmer_PatientApp

References

[1] MySignals. Mysignals changes the future OS medical and eHelath application. Retrieved October 2016, from http://www.my-signals.com/

[2] Hassanalieragh, M., *et al*. (2015). Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges. *Proceedings of the 2015 IEEE International Conference on Services Computing* (pp. 285-89).

[3] Proulx, J., *et al*. (2006). Development and evaluation of a bluetooth EKG monitoring sensor. *Proceedings of the 19th IEEE International Symposium on Computer-Based Medical Systems*.

[4] Istepanian, R. S., & Petrosian, A. A. (2000, Sept.). Optimal zonal wavelet-based ECG data compression for a mobile telecardiology system. *IEEE Trans. Inf. Technol. Biomed, 4(3)*, 200-211.

[5] Rasyid, M. U. H. A., *et al*. (2016). Portable electrocardiogram sensor monitoring system based on body area network. *Proceedings of 2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*.

[6] Chaimanonart, N., & Young, D. J. (2009). A wireless batteryless in vivo EKG and body temperature sensing microsystem with adaptive RF powering for genetically engineered mice monitoring. *Proceedings of 2009 Transducers, Solid-State Sensors, Actuators and Microsystems Conference*.

[7] Gandhi, N., *et al*. (2011). Properties of dry and non-contact electrodes for wearable physiological sensors. *Proceedings of 2011 International Conference Body Sensor Networks (BSN)*.

[8] Gargiulo, G., *et al*. (2008). Mobile biomedical sensing with dry electrodes. *Proceedings of the International Conference on Intelligent Sensors Sensor Networks and Information Processing 2008*, *ISSNIP 2008* (pp. 261-266).

[9] Richer, R., *et al*. (2014). Real-time ECG and EMG analysis for biking using android-based mobile devices. *Proceedings of 2014 11th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*.

[10] Kerdjidj, O. (2016). Real ECG signal acquisition with shimmer platform and using of compressed sensing techniques in the offline signal reconstruction. *Proceedings of 2016 IEEE International Symposium on Antennas and Propagation (APSURSI)*.

Poornima Eshwara is born in San Francisco, at the date of 03/10/1993. She received her bachelor's degree in electrical engineering from Visveswaraya Institute of Technology, Bangalore, India, in 2011; and the master's degree in embedded electrical and computer systems, from San Francisco State University, San Francisco, CA, USA, in 2018. She is now a graduate teaching assistant at San Francisco State University and a program analyst at Cognizant Technology Solutions. Ms. Eshwara has been a member of the Society of Women Engineers since 2016.

Hamid Shahnasser is born in San Francisco. He got his B.E. degree in electrical engineering from McGill University, Montreal; the MS degree in electrical and computer engineering from Carnegie-Mellon University; and the Ph.D. from Drexel University Pennsylvania.

He is currently a professor of electrical and computer engineering at San Francisco State University and the engineering graduate program coordinator. His areas of interest are communication networks and computer systems.

Dr. Shahnasser has been a research faculty consultant to NASA Ames Research Center projects since 1990 and has collaborated on several research grants with that organization since then. He has received grants from NSA, Department of Education, National Science Foundation and various private companies carrying out research in the areas of his interest.