



## Mobile Electrocardiogram Monitoring System with Cloud-Based Approach

Goh Voon Hueh<sup>1</sup>, Muhammad Akmal Ayob<sup>2</sup>, Nurul Izzati Darul Zaman<sup>1</sup>, Yuan Wen Hau<sup>1,2\*</sup>

<sup>1</sup>School of Biomedical Engineering and Health Sciences, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

<sup>2</sup>IJN-UTM Cardiovascular Engineering Centre, Institute of Human Centered Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

\*Corresponding Author [hauyuanwen@biomedical.utm.my](mailto:hauyuanwen@biomedical.utm.my)

Received 04 July 2022; Accepted 04 August 2022; Available online 6 August 2022  
<https://doi.org/10.11113/humentech.v1n2.30>

### Abstract:

Cardiovascular disease (CVD) is a heart related disease and is the top silent killer in worldwide. Frequent electrocardiogram (ECG) monitoring for patients with CVD is vital to check if arrhythmia occurs. Though there are lots of commercialized portable ECG monitoring device available, many of them are dedicated to professional clinical staff with complex user interface, or only targeted to specific arrhythmia for home monitoring with lack of data management system. In this study, an in-house developed ECG monitoring device was integrated with an Android-based mobile application through Bluetooth wireless communication and implements Google Cloud Technology. The Android-based mobile app supports main features of real-time ECG graph display, automated location detector, data management system and simple arrhythmia detection. The ECG graph displays acquired ECG signal in real-time by in-house ECG acquisition unit through Bluetooth wireless communication and stored as text files in phone's local memory. Firebase Authentication and Firebase Storage based on Google Cloud technology are implemented for data management system development. This allows users and administrators to upload or access data securely through online Google Cloud Platform. Three types of heart rhythm, namely normal sinus rhythm, bradycardia and tachycardia can be classified based on heart rate analysis. Smartphone's location service is enabled to allow user shares their location with caretaker in emergency case. The final solution utilizes the Internet-of-Things (IoT) technology to facilitate heart disease management towards telemedicine applications.

Keywords: Arrhythmia; Cardiovascular disease; Electrocardiogram; Google technology; Mobile application

### 1. Introduction

According to World Health Organization (WHO), cardiovascular diseases (CVD) such as coronary heart disease, strokes, transient ischemic attack, peripheral arterial disease, and aortic disease cause 17.9 million people death globally in 2019 [1]. Meanwhile, the Department of Statistics Malaysia (DOSM) reported that ischemic heart diseases remained the principal cause of death, with 17.0% of the 109,155 medically certified deaths in 2020 [2]. CVDs can be caused by factors like family inheritance, unhealthy lifestyle such as tobacco and high cholesterol or oily food intake, inactive physical activity, frequent alcohol consumption, etc. [3].

An arrhythmia occurs when the heart rhythm is abnormal, whether the heart pumps too fast, too slow, or irregularly. There are a few arrhythmias: atrial fibrillation, bradycardia, tachycardia, ventricular fibrillation, and many more [4]. Certain arrhythmias are harmless, but specific arrhythmia is life-threatening if the condition prolongs to a certain extent. The causes of arrhythmia are numerous, from caffeine consumption, long-term drug and alcohol

consumption, and electrolyte imbalances to health problems like coronary artery disease, high blood pressure, diabetes, and so forth [5]. Arrhythmia detection is vital as a strong indicator of potential sudden heart attack or other dangerous heart diseases.

Electrocardiogram (ECG) is a non-invasive medical device that measures the heart’s electrical activity and displays the heart rhythm in the waveform. This heart screening process is called electrocardiography, which involves placing electrodes on human skin to measure electrical voltages. Along with the technological advancement and significant market demand, many portable and light-weight commercial ECG monitoring devices are available compared to the bulky but standard hospital-grade devices. However, these commercialized ECG devices are primarily targeted for ECG acquisition and recording only with a complex graphical user interface (GUI) and without self-interpretation for professional clinical staff, or only targeted for simple or specific arrhythmia detection to facilitate home monitoring.

On the other hand, many research studies have proposed designing a low-cost ECG monitoring system based on mobile technology. For example, a low-cost ECG monitoring system based on Arduino technology with Bluetooth wireless data transmission is proposed by [6]. An Android-based mobile application that allows real-time ECG monitoring and automated arrhythmia detection by analyzing ECG parameters has been proposed [7]. These related works generally share limitations in not supporting a data management system and online cloud storage for user access. The same limitation also applies to an in-house designed home-based heart rhythm monitoring device, which its mobile app only supports Windows-based mobile phones and does not support real-time ECG graph plotting [8]. It can only display offline ECG records acquired by the acquisition unit and perform data interpretation based on offline stored data. In short, these related works do not support and utilize Internet-of-Thing (IoT) technology in their solution for telemedicine applications. IoT technologies guarantee that an affordable and reliable system can be embedded for patients’ seamless remote health monitoring and wireless communication with physicians for medical advice [9]. Thus, managing the ECG data recorded by the patient through IoT technology such as cloud storage help physician monitor the ECG of the patient anywhere and anytime.

This article presents an enhancement of the existing in-house design reported in [8] by integrating a mobile application that targets Android-based mobile phones. The Android-based mobile application allows IoT technology to be fully utilized since users can upload their ECG data to the caretaker. A home-based heart screening device with Bluetooth wireless communication to facilitate real-time ECG monitoring is available for users. Users can also view their ECG results using the Android mobile application and share stored ECG raw data with their physicians for further clinical consultation. By enabling wireless transmission of ECG raw data to the cloud storage, users and caretakers can access the data easily and do not require manual record downloading from the device. With self-detection of arrhythmia available, user can detect abnormal heart rhythm for early heart disease detection and prevention by seeking professional clinical consultation before the serious heart disease onset.

## 2. Materials and Methods

### 2.1 System Design Overview

Figure 1 shows the top-level system design of the enhanced ECG monitoring system, which consists of an in-house ECG acquisition unit [8], the developed Android-based mobile app and the Google Firebase platform.

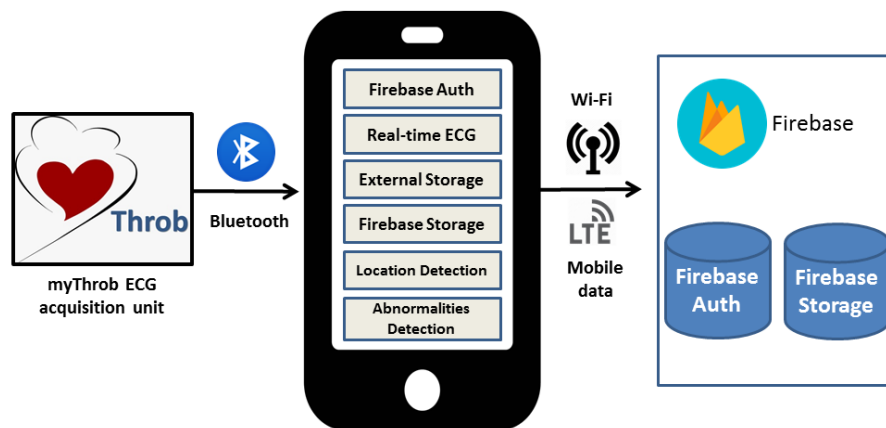


Figure 1. Top-level system overview

The in-house ECG acquisition unit was developed as reported in [8], which consists of an Arduino NANO module, an OLIMEX SHIELD-EKG-EMG module [10], and an HC-05 Bluetooth module to acquire single lead ECG raw data. The Arduino NANO module acts as the master microcontroller to receive an ECG signal obtained by OLIMEX SHIELD-EKG-EMG and wirelessly transmit the ECG raw data to the developed Android mobile application via the HC-05 Bluetooth module after the signal amplification, filtration, and quantization. During wireless data transmission, the operation mode of the HC-05 Bluetooth module is configured to Attention (AT) mode.

The mobile app allows users receive and display the ECG signals in real-time. After the ECG monitoring, all the recordings are automatically saved as a text file in offline storage. Using simple calculation based on heart rate, the user would identify if the heart rhythm were in the normal or abnormal range. Through Firebase Cloud technology, the system allows users to perform authentication activities like sign up, log in and sign out using Firebase Authentication, which the developers can monitor all the user activities. ECG data management system is controlled by implementing Firebase Storage. By enabling a Wi-Fi or 4G data connection of the smartphone, a user can choose to upload ECG raw data to Firebase Storage for physicians’ further post-analysis. The location detection feature of the mobile device is also enabled the user can select and copy the location text if they wish to share their location with their caretaker or family member. In this manuscript, the mobile app development and the data management based on Google Firebase technology will be clearly described, whereas the reader could refer to [8] for detailed ECG acquisition unit design.

**2.2 Android-based mobile application**

The Android-based mobile app is developed to perform certain functionalities, which include wireless data receiving through Bluetooth, local data management system, real-time ECG graph display, location detection, and simple arrhythmia detection. The application is developed using Java programming language in Android Studio and its Software Development Kit (SDK) tools, such as platform-tools, build tools, SDK-tools, Android Debug Bridge (ADB) and Android Emulator for developers [11].

The mobile app consists of five main Java classes, namely *Splash.Java*, *ExpandableListAdapter.Java*, *ExpandedMenuModel.Java*, *MainActivity.Java* and *SignUpActivity.Java*, with each of them serving different purposes, as shown in Figure 2. *Splash.java* is an activity which displays the icon image and welcoming screen for the user. *ExpandableListAdapter.java* and *ExpandedMenuModel.java* are the Java classes for designing activities with navigation drawer properties. The *MainActivity.java* is the activity classes that contain all the aforementioned main functions. *SignUpActivity.java* is an activity class designed for user’s sign up activities who do not have a registered account for this application.

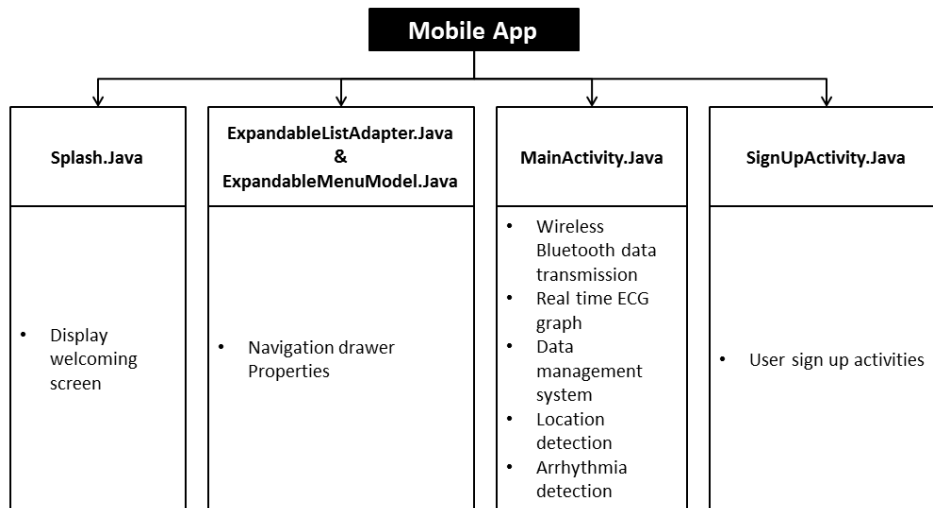


Figure 2. Summary of activities in Java classes

**2.2.1 Welcoming screen and navigation drawer**

*Splash.java* is created to display a welcoming screen for the user, with an animated icon appearing for 2 seconds. The intent is then designed to automatically direct the user to *MainActivity.java* after the animation has finished.

Two java activities, named *ExpandedMenuModel.java* and *ExpandableListAdapter.java*, are created to design a navigation drawer layout, which can expand and retrieve the drawer when it is not needed. The navigation drawer is designed to have group items and child items. This provides options for the user, for instance, choosing an available Bluetooth device to be paired.

*ExpandedMenuModel.java* is a Java class to return the value of a selected item when it is called as a List by *ExpandableListAdapter.java* activity to list the position of a group item and child item. The position will be returned as a value if an item is selected. On the other hand, *ExpandableListAdapter.java* is called *MainActivity.java* as a List Adapter. When selected, the items will perform the features as designed in *MainActivity.java*.

**2.2.2 User sign up activity**

Figure 3 shows the behavioural flowchart of user sign up activity. In the navigation drawer, there is a selectable item named “User”, which a user can click the item to log into the application account. Once a user clicks the menu, a dialog box will appear. If the user has not signed up for an account, they should click the “Sign Up” button, which will lead them to *SignUpActivity.java*. This system applies email address and users’ customized passwords as authentication methods during account registration provided by Firebase Authentication Application Programming Interface (API). Once the users have signed up successfully, their email address and user ID will be available in the Firebase Authentication console. The administrator can manage their application’s users from the console. Only users who have signed up successfully would be able to select and upload the text file in Firebase Storage, as discussed in the subsections later.

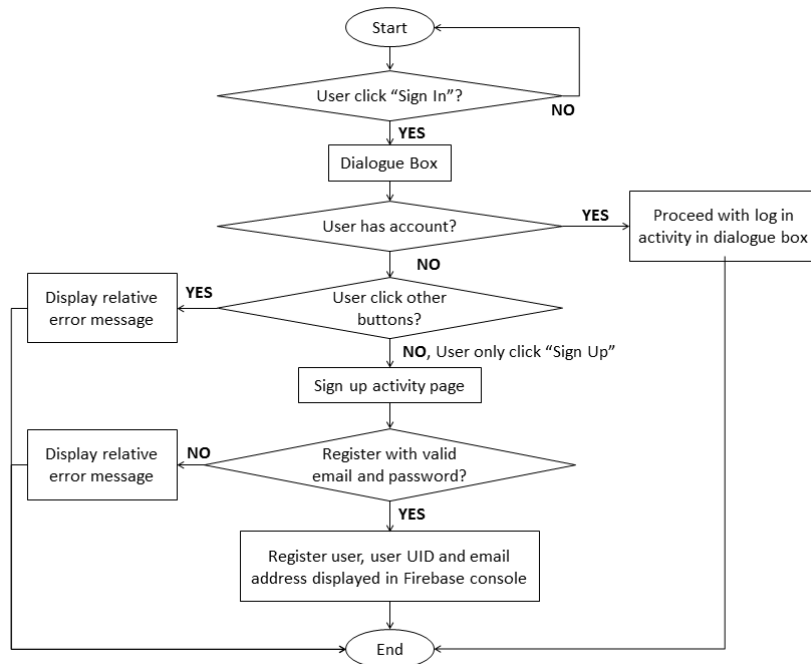


Figure 3. Flowchart of user sign-up activity

**2.2.3 Wireless Bluetooth connection**

Figure 4 shows the behavioural flowchart of wireless Bluetooth pairing activity between mobile apps with the in-house ECG acquisition module for ECG data transmission. In the navigation drawer, there are three items named “Bluetooth”, “Search”, and “Select Device”, which functionality activities are created in *MainActivity, java*, and will be called if a user clicks on them. If the user has not turned on Bluetooth, they will have to click on “Bluetooth” to turn it on. The intent is to check if the mobile device has Bluetooth features, seek the user’s permission, and turn the Bluetooth features of the mobile phone on and off. The broadcast receiver receives the intent and determines the message’s state. After the device’s Bluetooth has been turned on, a Toast message will notify the user. To search for the device, the user must click on the item “Search” to start the device searching process. If an available Bluetooth device can be paired with, the device name and address will be saved in a temporary list. To select the device, the user must click on the item “Select Device”. If there is any available device to be paired with, the child item list will expand and show the

names of devices. When the user clicks on a device, an intent is sent to pair with the device. The broadcast receiver will determine the message and bonding states, hence performing the task that has been requested.

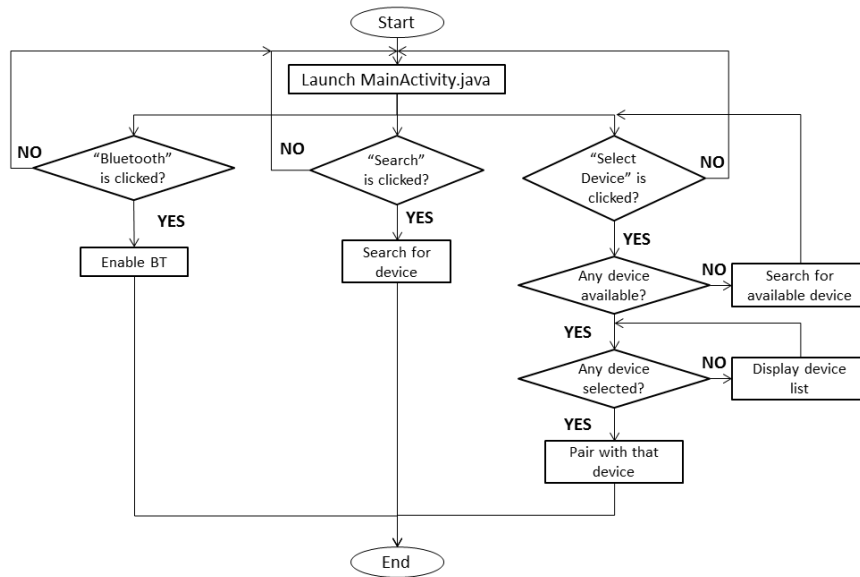


Figure 4. Flowchart of Bluetooth pairing activity

If the user has finished pairing their phone with the in-house ECG acquisition unit, they should click on the “Connect” menu in Main Activity to create an RFCOMM socket which initiates the connection and starts receiving the ECG input stream. When the device is connected, the button “Connect” will change to “Disconnect”. The application reads the incoming raw data sent from the in-house ECG acquisition unit as ASCII characters and stores the data in a temporary buffer with a maximum size of 1024 bytes. Each data is then parsed into an integer data type and stored in an integer array list. If the user wants to terminate data recording, they could click on the “Disconnect” menu to stop the input streaming, and the socket will be closed.

**2.2.4 Real-time ECG graph plotting**

Figure 5 shows the behavioural flowchart of real-time ECG graph plotting using a free and open-source library, named *MPAndroidChart*, developed by Philipp Jahoda [12]. The Main Activity page has a menu named “Plot” to allow users to display the real-time ECG graph based on the received ECG data via Bluetooth. The graph display is initialized by a function named *graphInit()*, while *addEntry()* function call help to plot the ECG graph when the mobile app receives an input ECG stream from Bluetooth. The function *addEntry()* will keep running until the input streaming is finished.

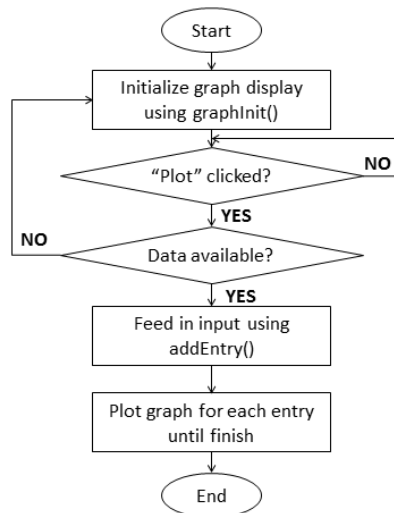


Figure 5. Flowchart of ECG real-time graph plotting

**2.2.5 Data management system - Offline storage**

In the *MainActivity.java* class file, when the user clicks the “Disconnect” menu to stop the input ECG data streaming, all the received data previously saved in the array list will be converted to a string. The file will then be kept permanently in local phone storage in a text file, with the naming system as “username\_dd/mm/yyyy\_hh/mm/ss” to prevent confusion of records. The allowable maximum file size is determined by the available free capacity of Random Access Memory (RAM) which can store the temporary data during input ECG data streaming.

**2.2.6 Location detection**

A function named *getLocation()* is created in *MainActivity.java* class to obtain the device’s location in latitude and longitude form, as shown in Figure 6. This function will run automatically after the user logged in. When this application is launched, the user will be asked for permission to obtain location information for the first time. The location is shown in TextView, where its text selection allows users to copy and paste the location information in case, they wish to share it with others.

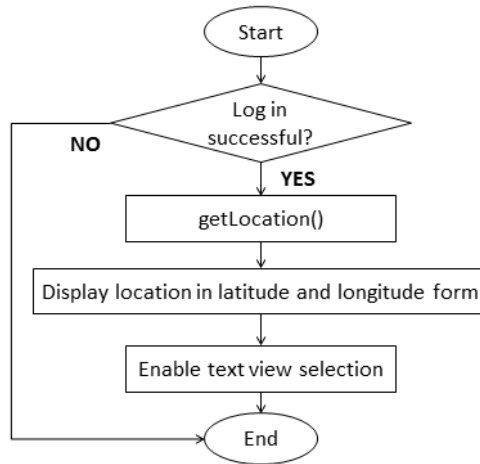


Figure 6. Flowchart of location detection activity

**2.2.7 Arrhythmia detection**

In this ECG monitoring system, the arrhythmia conditions that can be classified are sinus bradycardia and sinus tachycardia, of which the former has an average heart rate of less than 60 bpm, while the latter has a heart rate of more than 100 bpm. Normal sinus rhythm is classified with condition heart rate is between 60 to 100 bpm. One of the methods to identify arrhythmia is by calculating heart rate, as shown in the formula below.

$$\text{Heart rate} = \frac{60}{\text{time interval between two R peaks}} \tag{1}$$

The R peaks detection determines the maximum value in a chunk of values. This method is divided into several steps, as shown in Figure 7. First, inputs that are more than 500 are saved in a temporary array named *temphighArray*, to store amplitude values. Another temporary array list named *temphighIndex* is created to store the values of amplitudes’ index. The sequence of inputs received must be determined to calculate the time interval between two R peaks. The function *findPeak()* then finds the maximum value in *temphighArray*, which means determining the peak and moment where the peak happens. After several R peaks have been determined and stored in a temporary array, the heart rate for each RR peak interval is then calculated. OLIMEX Shield EKG-EMG and Arduino NANO’s sampling frequency was set to 360 Hz. As a result, the time interval between two R peaks is determined as follows:

$$\text{RR interval} = \frac{\text{index of B peak} - \text{index of A peak}}{\frac{1}{360}} \tag{2}$$

After all the heart rates have been calculated, they are saved in a temporary array. The average heart rate is then calculated using the formula below.

$$\text{Average heart rate} = \frac{\text{Sum of total obtained heart rates}}{\text{size of temporary array storing all heart rates}} \quad (3)$$

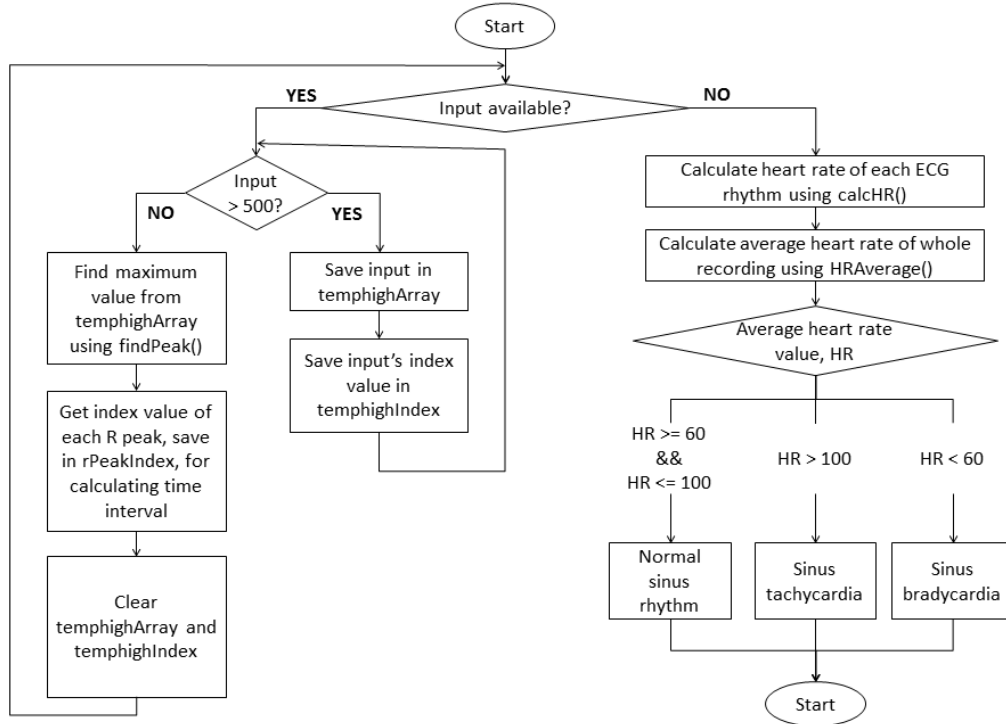


Figure 7. Flowchart of arrhythmia detection activity

The final step is to determine the condition by matching the average heart rate value obtained with the range shown in Table 1.

Table 1. Types of heart conditions

Heart Condition	Range
Normal Sinus Rhythm	60-100 bpm
Sinus Tachycardia	> 100bpm
Sinus Bradycardia	< 60bpm

### 2.3 Firebase authentication

Three features are designed using Firebase Auth API: sign up, login and sign out activities. These APIs and functions are declared in the *MainActivity.java* class file. When the user clicks an item named “User”, a dialogue box will pop up and requires the user to key in information like username, email address and password, as well as buttons like “Log In”, “Log Out”, and “Sign Up”. If the user has not signed up for an account or key in an invalid email address and password, an error message will pop out to inform the user. If a user wants to sign up for an account, click the “Sign Up” button and direct them to another activity which is *SignUpActivity.java*. If the email address and passwords are valid, their successful registered account and ID will be updated in the Firebase Authentication console, and the administrator can monitor the user activity. The administrator also has the authority to delete an account if necessary.

### 2.4 Firebase storage

Figure 8 shows the behavioural flowchart of data uploading to cloud storage activity. In the Main Activity layout, there is a button named “Store”, which will only be enabled after the user has successfully signed up to allow the user to upload any text file to Firebase Storage. This will direct the user to an activity displaying all the files stored in internal phone storage. After selecting a text file, a progress dialog will appear until the uploading process has finished. All the uploaded data are classified accordingly to the user’s email address during account login to avoid any potential data confusion. These functions are called in the *MainActivity.java* class file.

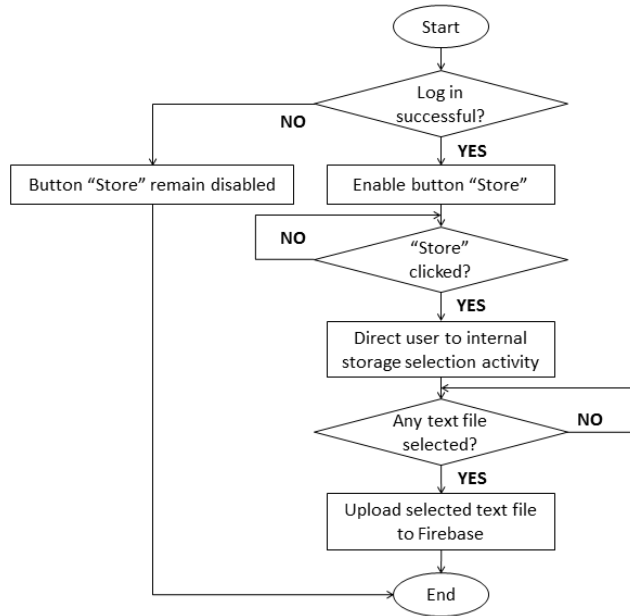


Figure 8. Flowchart of storing files into Firebase Storage

### 3. Result and Discussion

This section presents the developed GUI of the mobile app based on the aforementioned main functionalities in terms of user authentication, Bluetooth device pairing, data management system, real-time ECG graph display, and simple arrhythmia detection.

Figure 9 shows the GUI of the extended navigation drawer of the developed mobile app, as well as user sign up and log in activities for user authentication. Figure 10 shows the Firebase Authentication console, which proves that a user is successfully signed up through the developed mobile app and registered to this ECG monitoring application.

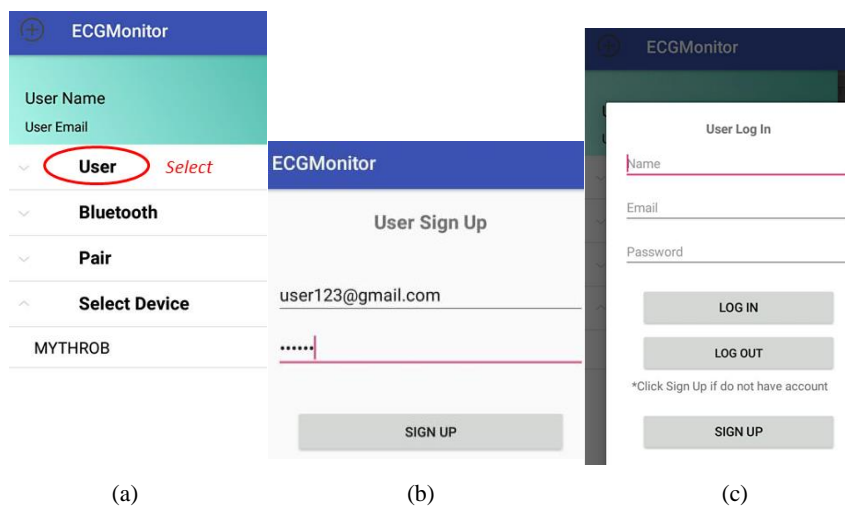


Figure 9. User authentication: (a) Navigation drawer, (b) User Sign Up and (c) User Log In



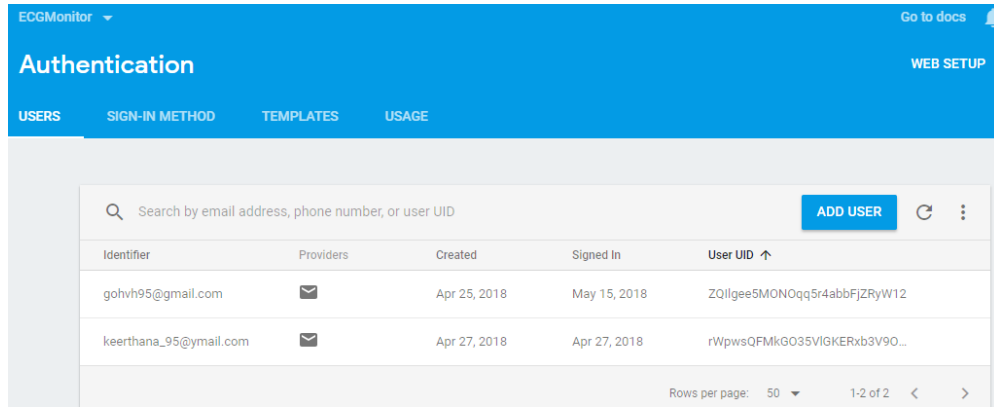


Figure 10. Firebase Authentication console

Figure 11 shows the GUI of Bluetooth device pairing functionality, which proves that the developed mobile app manages to scan the surrounding available Bluetooth devices and pair them with an in-house ECG monitoring device successfully to start input ECG data streaming.

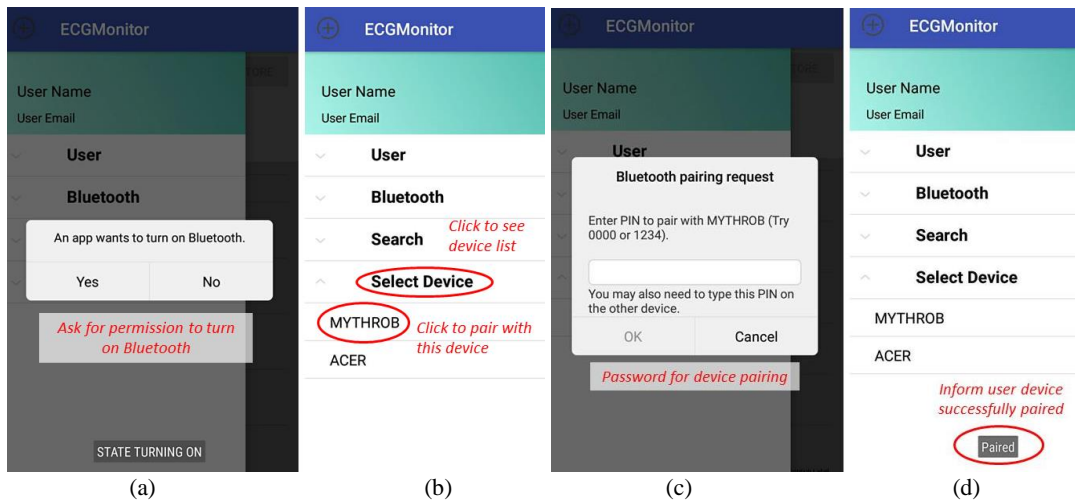


Figure 11. Bluetooth communication: (a) Permission request, (b) device selection, (c) pairing request and (d) paired device

Figure 12 shows the real-time ECG graph plotting with arrhythmia detection in terms of normal sinus rhythm, sinus tachycardia and sinus bradycardia. The condition and average heart rate value will be displayed.

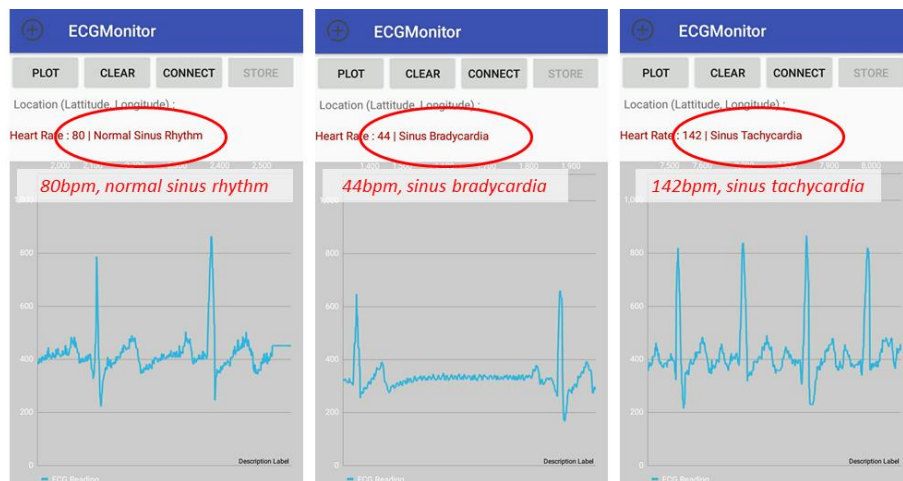


Figure 12. Heart rate and arrhythmia detection

The Fluke patient simulator simulates a few types of ECG signals to compare the heart rate accuracy and plotted ECG graph. The signals plotted in the developed mobile app are visually compared with the original arrhythmia ECG signal generated by the patient simulator at the Lecroy mixed signals oscilloscope, as shown in Figure 13. Results show that the plotted real-time ECG graph of different arrhythmia shows a similar waveform pattern to the initially generated signal displayed at a mixed signal oscilloscope.

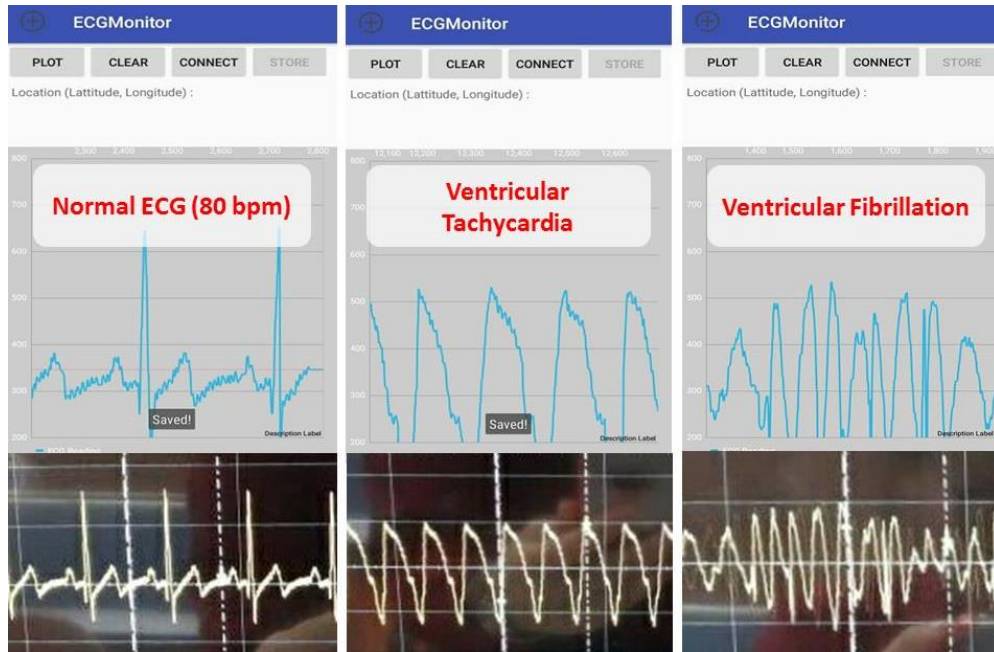


Figure 13. Graph comparison between application and oscilloscope

However, it is noticed that due to the large ECG data stream being received in parallel with the animated real-time graph plotting at the same time, the mobile app somehow lags, and the graph view seems not to react. In this scenario, users must disconnect and reconnect to resume smooth ECG graph plotting. However, the input streaming is not affected. Users can also clear cache files to reduce the chances of application lag during graph plotting. Noise is also observed from the graph when electrodes are placed on the human subject instead of the patient simulator. However, the ECG graph is still evident, and features of PQRST waves can be identified. Other further design details are clarified by Hueh [13].

Figure 14 shows the data management system of the acquired ECG data stored as offline recording in internal local and Google Firebase cloud storage after successful data uploading.

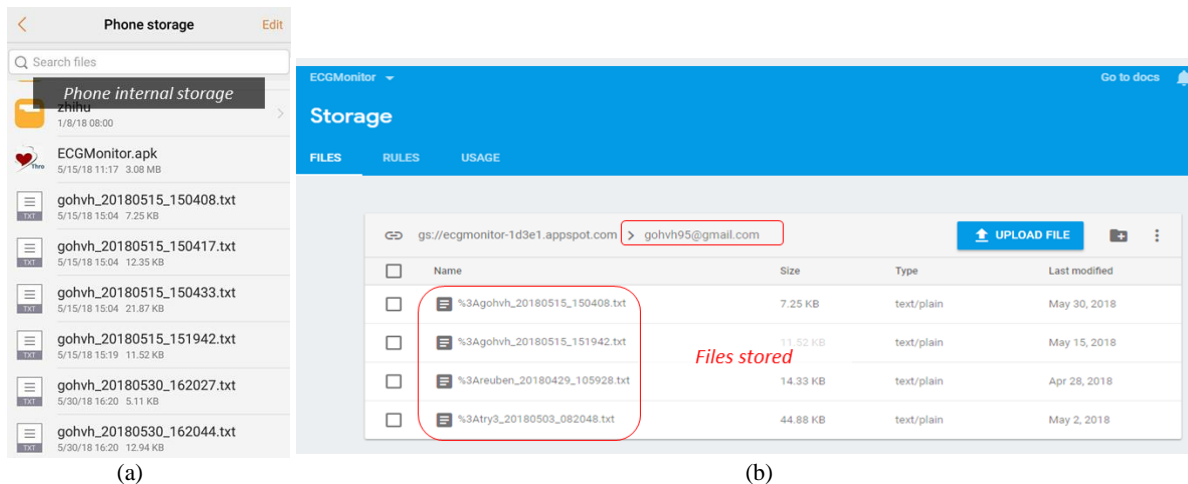


Figure 14. Data management: Offline ECG records in (a) phone internal storage as a text file and (b) Google Firebase console

## 4. Conclusion

This manuscript has presented an enhanced ECG monitoring system based on integrating an in-house ECG acquisition unit, an Android-based mobile application, and Google Firebase technology. This system is enhanced to enable wireless communication via Bluetooth, reducing the troubles of connecting devices to the workstation for manual ECG record extraction. The developed mobile app also provides certain useful functionalities in terms of user sign up or login, real-time ECG graph plotting, self-classification of simple arrhythmia, and data management system in both phone internal local storage and Google cloud storage based on Google Firebase technology. By fully utilizing IoT technology available in the mobile phone as a virtual device of modern life, this enhanced ECG monitoring system will promote home-care service by enabling frequent heart rhythm monitoring for heart disease management. At the same time, it also allows easier access by the clinicians for further clinical consultation and faster patient support. As a future recommendation, the result of the study can be further improved and validated by evaluating functional or non-functional specifications based on user acceptance survey.

## 5. Acknowledgement

This research is supported and funded by the Ministry of Higher Education (MoHE) Trans-Disciplinary Research Grant Scheme (TRGS) with Grant no. TRGS/1/2015/UTM/02/3/3 (UTM vote no. R.J130000.7845.4L842) and Universiti Teknologi Malaysia Research University Grant with Vote No. Q.J130000.2545.16H83.

## References

- [1] Cardiovascular diseases, World Health Organization. [https://www.who.int/health-topics/cardiovascular-diseases#tab=tab\\_1](https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1) (accessed: May 26, 2022)
- [2] M.U. Mahidin, Statistics on causes of death, Malaysia, Department of Statistics Malaysia. [https://www.dosm.gov.my/v1/index.php?r=column/cthemByCat&cat=401&bul\\_id=R3VrRUhwSXZDN2k4SGN6akRhTStwQT09&menu\\_id=L0pheU43NWJwRWVSZklWdzQ4TlhUUT09](https://www.dosm.gov.my/v1/index.php?r=column/cthemByCat&cat=401&bul_id=R3VrRUhwSXZDN2k4SGN6akRhTStwQT09&menu_id=L0pheU43NWJwRWVSZklWdzQ4TlhUUT09) (accessed: May 26, 2022)
- [3] Cardiovascular disease, National Health Service. <https://www.nhs.uk/conditions/cardiovascular-disease/> (accessed: May 26, 2022)
- [4] About arrhythmia, American Heart Association, <https://www.heart.org/en/health-topics/arrhythmia/about-arrhythmia>. (accessed: May 26, 2022)
- [5] Heart arrhythmia - Symptoms and causes - Mayo Clinic, Mayo Clinic Staff. <https://www.mayoclinic.org/diseases-conditions/heart-arrhythmia/symptoms-causes/syc-20350668> (accessed: May 26, 2022)
- [6] M.A. Ahamed, M.K. Hasan and M.S. Alam, Design and implementation of low cost ECG monitoring system for the patient using smartphone, 1<sup>st</sup> International Conference on Electrical & Electronic Engineering (ICEEE), 2015, 261–264. <https://doi.org/10.1109/CEEE.2015.7428272>
- [7] S. Gradl, P. Kugler, C. Lohmüller and B. Eskofier, Real-time ECG monitoring and arrhythmia detection using Android-based mobile devices, 34<sup>th</sup> Annual International Conference of the IEEE EMBS, 2012, 2452–2455. <https://doi.org/10.1109/EMBC.2012.6346460>
- [8] W. Jun Mun, Embedded system design of arrhythmia screening device with wireless communication, Universiti Teknologi Malaysia, 2015.
- [9] M. Sathya, S. Madhan and K. Jayanthi, Internet of things (IoT) based health monitoring system and challenges, International Journal of Engineering & Technology, 2018, 7(1.7):175–178.
- [10] Olimex Ltd, Sheild EKG-EMG bio-feedback shield User's Manual, 2014.
- [11] D. Felker. Chapter 6 Developing with Android studio, Android Developer Tools Essentials by Mike Wolfson. <https://www.oreilly.com/library/view/android-developer-tools/9781449328207/ch06.html> (accessed: May 26, 2022).
- [12] P. Jahoda. MPAndroidChart. <https://github.com/PhilJay/MPAndroidChart> (accessed: May 26, 2022).
- [13] G.V. Hueh, Android-based mobile application for home-based electrocardiogram monitoring device with Google technology and Bluetooth wireless communication, Universiti Teknologi Malaysia, 2018.