EXTENDING THE SANA MOBILE HEALTHCARE PLATFORM WITH FEATURES PROVIDING ECG ANALYSIS

by

KATERINA TSAMPI

Bachelor Degree in Computer Science and Information Technology of the Technological Educational Institute of West Macedonia in Kastoria, 2012

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

DEPARTMENT OF INFORMATICS ENGINEERING

SCHOOL OF APPLIED TECHNOLOGY

TECHNOLOGICAL EDUCATIONAL INSTITUTE OF CRETE

2017

Approved by:

Assistant Professor
Spyros Panagiotakis

Abstract

The great development of technology recently provides innovations that improve everyday life. The major benefit is that the branch of medicine is also affected, so better healthcare can be provided. In that context, it can be critical for patients who suffer from chronic heart diseases to have in their availability a system that can monitor and analyze their electrocardiogram (ECG) displaying either normal or abnormal findings. The current thesis implements such a system that uploads, stores, processes and displays an ECG, calculating certain ECG findings necessary for doctors to make a diagnosis. To this end, the SANA mobile healthcare platform, with its OpenMRS open source enterprise electronic medical record system, has been chosen and extended for storing, processing and displaying the ECG data. OpenMRS provides a user-friendly interface and a database for collecting medical big data. Analysis of ECG signals is leveraged by the Physionet toolkit. Physionet contains many ECG databases and the WFDB software for processing ECG signals. According to the scenario we have processed, an ECG is uploaded onto OpenMRS platform using a mobile device or any other Internet-enabled device and is stored in the database that OpenMRS uses. Then, ECG signal is filtered using a finite impulse response (FIR) filter to remove noise and using WFDB functions it is processed so certain intervals are determined. Finally, with the appropriate algorithms specific ECG findings are calculated. When the procedure completes, the results are stored into the database using SQL Queries. Using an HTML Form results and graphs are integrated into the OpenMRS website highlighting abnormal values with red color. Authorized users can have access to this information through any web browser. The implementation has been evaluated with various demo ECG signals verifying its correct behavior.

Keywords: Electrocardiogram, OpenMRS platform, WFDB software, ECG processing

Table of Contents

List of Figures

List of Tables

Acknowledgements

First of all, I would like to thank my supervisor Prof. Spyros Panagiotakis for showing me his trust and providing advice and continuous support.

I would also like to thank all my colleagues at the Multimedia Laboratory and especially Georgia and Malvina for the helpful discussions and their useful advice.

Last, but not least, I would like to thank all my family for their support, belief and encouragement during these years.

Chapter 1 - Introduction

The development of technology provides us innovations to improve our everyday life. Information Technology (IT) describes the procedure of creation, manipulation, storage and secure exchange of different types of electronic data [1]. The major benefit is that the area of medicine has been affected as well in order to provide better healthcare. Although the initial target of mobile healthcare systems was to help patients in developing countries, they are regularly used in worldwide scale. Such systems are mainly destined to support health in village areas. The use of such a unified system increases the quality of health care. Additionally, the exponential progress of recent mobile devices with remarkable computational re-sources (CPU, RAM, etc.) combined with internet connectivity that Internet Service Providers guarantee allow quick and instant access to medical data using a simple device (smartphone or tablet).

Health Information Technology (HIT) is a sub-category of information technology that refers to healthcare data storage and processing [2]. It enables health data management and secure medical information exchanges. The Electronic Medical Record (EMR) or Electronic Health Record (EHR) is the major component of HIT. The use of EMR or EHR has significantly improved the healthcare quality.

The EMR systems describe systems that store medical information in digital format providing the potential to read clinical notes using a device [3]. Medical records may include a variety of data, including demographics, medical history, medication, vital signs and personal information such as weight and age. EMR systems use also data that express crucial fundamental health indicators such as blood pressure, respiratory rate, oxygen saturation, temperature etc. These indicators react as warning for instant doctor treatment in case of abnormal operations. One major form of such health data is electrocardiogram (ECG).

An electrocardiogram (ECG) signal describes the electrical activity of the heart using electrodes on the body surface over a time space usually ten seconds. The electrical activity describes the amount of impulses in a heartbeat and provides information about the heart rate, rhythm, and morphology [4]. Heart produces electrical current that we can record. Connecting electrodes between the body and the electrocardiograph creates an electrical circuit. Each pair of attachments is called “lead”.

According to cardiologists, in order to evaluate an electrocardiogram, they have to extract empirically the required metrics. The ECG signal is represented on a graph paper, which makes easier the calculation of those metrics by the doctor. Our system provides the appropriate processing and produces the ECG findings, helping doctors to make a critical decision more quickly. Those ECG findings indicate heart rate, amplitude of the ECG waves such as R-wave amplitude in lead I, duration of the ECG waves e.g. R-R interval, heart axes such as QRS axis and ratio between the waves amplitudes like V1 ratio.

The implementation describes a system that processes ECG signals and produces necessary results using an Electronic Medical Record (EMR) System. The ECG signals come from Physionet Database, which is compatible with the Standard formats such as EDF, BDF and SCP-ECG. The PTB Physionet Database contains 15-lead ECGs. Twelve of them are the standard leads and the remaining three are the Frank XYZ leads. The database contains clinical information for each record. Physionet provides the PhysioToolkit, which contains free applications and functions for ECG signal processing. The Physionet and its software are analyzed in ***Chapter 3 - Physionet.***

The EMR system that is used for the current implementation is the OpenMRS platform. OpenMRS contains a database for big data storage and a user-friendly interface [5]. OpenMRS provides expandability and free access. Moreover, OpenMRS is compatible with mobile devices and provides connection with Sana. Sana Technology Platform is a generic mobile Health system that integrates OpenMRS. The functionality of OpenMRS and Sana is described in ***Chapter 4 - OpenMRS – Sana.***

In previous implementations, OpenMRS is used as a database that accepts and stores data in image format. Additionally, it provides visualization capability that is used by the doctor for empirical diagnosis. In the current implemented system, we try to extend OpenMRS capabilities. Apart from the previously described operation, OpenMRS is assigned to analyse the input signals (ECG waveforms) that are uploaded by the medical staff. Additionally, our system allows direct connectivity with the health monitoring devices without requiring any medical staff. Also, the wearable health monitoring devices can transmit real time measurements using TCP/IP protocol. Finally, the system uses arithmetic data for the analysis, diagnosis and visualization process on contrast with previous implementations that display only uploaded data images.

In this research work, we implemented a system in which users, who have authorized access, upload an ECG signal file through their mobile phone, or a web browser, to the EMR system. After uploading the file, ECG signal processing takes place. During the processing stage, we initially remove noise by using several filters (such as FIR). After this step, our system evaluates the amplitude and duration of the signal waveforms and this information is stored into the database. Our application also illustrates patient's heart performance into graphs providing representative visualization to the medical staff. Authorized users can access electrocardiograms via this medical platform and detect possible abnormal operation. Abnormal findings are highlighted with red color to alarm the user. More details about the implementation of the system are described in ***Chapter 5 -Technical Implementation .***

For further evaluation, experiments using abnormal ECG signals have been done. The abnormal signals are signals with Myocardial Infarction, Myocardial Hypertrophy and Cardiomyopathy Heart Failure. The results from the experiments are produced by comparing normal and abnormal ECG signals. The experiments are described in ***Chapter 6 - Experimental Evaluation.***

Chapter 2 - Related Work on Health Care Applications

Overview

Health Information Technology (HIT) improves the health of individuals and the performance of providers. It also strengthens the quality and effectiveness of healthcare, making it efficient and productive. A major factor is that HIT reduces the medical expense. Exchanging health information between doctors shortens the diagnosis-waiting period and prevents from medical errors. HIT enables big data manipulation and sharing of healthcare information in worldwide scale using Internet in order to make effective decisions and help medical research area.

Healthcare applications contain a variety of technological achievements. Many researchers work hard in order to improve healthcare conditions. Recent innovative approaches often contribute towards this direction. For this reason, healthcare systems can be approached from many different perspectives. In the following sections, we analyze generally healthcare applications, the role of Big Data in this area and underline additionally the use of Internet of Things (IoT) technology and mobile devices for high quality healthcare services. Based on the above, we focus on Electronic Medical Record (EMR) systems by introducing similar projects and we conclude by describing the background of EMR systems that analyze electrocardiograms (ECG), also providing the required information for this topic, which is the main objective of the current research work.

Big Data and HIT

Big Data is a term referred to a large amount of data that expand rapidly. Big Data in health includes data originating from different types of sources related to health condition such as sensors and mobile devices. A subcategory of Big Data is the Medical Body Area Networks (MBAN), which provide continuous monitoring of patient's health by transmitting measurements from heart rate, blood pressure, respiratory rate, body temperature, and electrocardiogram (ECG). The role of the Big Data is crucial for clinical decision and health information systems [6].

The main features of Big Data in health are the volume, variety, velocity, veracity, validity and volatility. These characteristics refer to the amount of data, the sources that data stems from, the continuous data collection in real time and the availability over time [7]. The usage of Big Data is critical for taking instantly important decisions about patient's treatment. Big Data provides a proper and complete overview of a patient's condition, offering solutions against chronic diseases and tries to eliminate time that patients spend in hospitals. [8].

In developing countries, quality of health service has to be high, so the applications of data analysis can contribute towards this direction. A platform for analyzing Health Data is OpenMRS, which enables graph creation and allows quick access to medical records. It is proved to be useful in over 220 clinics worldwide [9]. Big Data are personal data and should be guaranteed and protected from harmful users. Finally, the Big Data analysis helps the improvement of healthcare services in developing countries [10], especially in cases of dangerous diseases for human life.

IoT-based Healthcare Applications

Remarkable technology development during last decades has improved the health information technology. Different types of health care services are provided, preventing patients from staying in hospitals, improving their quality of life. The Internet of Things (IoT) area offers new technologies using sensor networks, providing a system that offers doctors the capability of remote diagnosis. In industry, there are sensors for movement detection such as gyroscope and some others for vital signs monitoring such as temperature [11].

IoT development in recent decades results in innovative applications that are useful in biomedical and healthcare research. Due to easy use and little size, portable devices are widely used in IoT-based applications. Vital signs are monitored via these devices and are sent to the doctors via IoT and Android Application Framework collects the required data in order to have an accurate diagnosis.

The Body Sensor Network (BSN) system consists of various types of wireless wearable sensors that collect measurements from patient's body and transmit them to a data collector unit (e.g. a database). Based on the metrics that they monitor, sensors are separated into two

categories. The first category includes sensors that should monitor continuously and collect a large amount of data such as electromyogram (EMG) sensors and electrocardiogram (ECG) sensors. The second category includes sensors that collect smaller amount of data such as temperature sensors and blood pressure sensors. BSNs are also divided into categories based on the way of data transmission. Wireless sensors use wireless communication technologies such as Zigbee and Bluetooth, Radio Frequency Identification Devices (RFID) and Ultra-Wide Band (UWB) for communication with other sensors or devices [12].

Healthcare applications provide high quality services using sensors that measure biometric information. A critical decision that should be taken for reliable health monitoring describes the symptoms of the disease that should be examined. For instance, some symptoms for bulimia appear when a person eats in secret or if food disappears in a short period. Furthermore, patients with bulimia have more complex health problems such as hypertension and fever. According to the symptoms described above, applications for bulimia detection should include sensors for movement detection in the kitchen area and sensors for measuring vital signs such as body temperature and blood pressure. Another example is the Alzheimer's disease that is a serious illness than can kill a patient if he is lost. Memory loss and depression are some fundamental symptoms. In this case, geolocation sensors are required because memory loss causes difficulties of orientation. Depression can be detected using a sensor that examines the rhythm of heart rate. Moreover, the intersection with use of blood pressure sensors will provide more information about patient's health that can lead in proactive diagnosis and successful treatment [13].

Biometric sensors can be used to detect stress or aggression in patient's behavior informing people who visit a patient in purpose to avoid violent attacks. IoT can also inform medical staff in cases when equipment of the hospital needs refill such as medication or oxygen tanks. RFIDs scan barcodes of items and warn the staff in case of trackable items is finished such as dressings. RFIDs can also be used in patients' home in order to inform him if a drug needs replacement [14]. It is obvious that IoT technology can provide significant help in doctoral and patient's daily life, leading to a high-quality healthcare system.

IoT evolution has offered great progress in health domain. Healthcare improves the quality of hospital services and dramatically decreases the required time for disease detection. Many IoT applications are used to contribute in healthcare such as blood glucose monitor and

electrocardiogram (ECG) monitor. Via smartphones with Internet connection, heart condition is feasible independently space and time. Moreover, warnings inform doctors about emergency cases. iCarMa is a heart monitoring system that takes as input a photoplethysmogram (PPG) signal from sensors and detects heart diseases such as tachycardia and bradycardia. PPG signal uses infrared light at different body parts and detects changes in light absorption [15].

Cardiovascular diseases (CVDs) are diseases that involve heart such as myocardial infarction. CVDs may cause sudden deaths for example heart attack if there is no early diagnosis. Using some parameters such as age, cholesterol and blood pressure can prevent from CVDs. A mobile system can use these parameters and export results related to the estimation of cardiovascular risk preventing patients from heart failure. An ECG monitor is used for collecting ECG signal and the evaluation starts. When processing is terminated, ECG signal is displayed using a mobile platform [16].

The above analysis shows that EMR have critical role is medical area due to the capabilities they offer related to portability, early detection and accuracy. These factors are crucial in many dangerous deceases that require constant monitoring and instant treatment. Heart diseases have also similar requirements and for this reason, EMR systems that use ECG are well known in medical industry.

Mobile Health Care Applications

The development of the mobile devices has affected the area of medicine significantly. The medical software applications for mobile devices (m-Health systems) have become very important today. Clinicians have access to medical information through tablets and smartphones. Mobile devices have to satisfy some basic requirements in order to be useful in health area. Such operations are easy constant Internet connectivity, quick access in electronic medical records, capability of providing information about treatment or a disease and also can support interactive teleconference conversations between doctors located in different places for exchanging opinions for emergency cases.

The benefits of using mobile devices in health are many, including the improvement of the knowledge level and instant access to medical data. Rapid decisions are made with reducing

risk and the quality of data accessibility and management is increased [17]. Patients can have direct communication better doctors during treatment and doctors can help instantly in cases that patients are located far away from hospital via a simple mobile device.

Mobile devices in combination with body sensor networks provide effective disease prediction in real time and prevent patients from spending money and time travelling or waiting for treatment in hospitals. In developing countries, many people that live in inaccessible regions can be treated by doctors easily which is valuable. According to the above, the health care system should incorporate mobile applications, which are used for storing medical data and accessing medical data via Internet. The patient can use the application to store his medical information including symptoms and the doctor can use this application to view the medical history and write a diagnosis and medication. The patient is informed about the recording of the diagnosis by the application [18].

The architecture of the m-Health systems includes three layers, which are the data collection, data storage, and data processing. The data collection layer describes the procedure of collecting data via mobile device. The data storage layer is responsible for storing it in Big-data form and the data processing layer analyses the data and display the report or the diagnosis [19].

Many mobile healthcare platforms have been created to provide help and better patient care. A well-known mobile healthcare platform is eMOCHA (electronic Mobile Open-source Comprehensive Health Application), which is a free open-source application, developed by the Johns Hopkins Center for Clinical Global Health Education and was designed to be applied in developing countries. Moreover, the National Library of Medicine (NLM) Mobile Resources contains a collection of mobile-friendly websites and applications. Magpi (formerly EpiSurveyor) is a free mobile phone and Web-based data collection system for global health. FrontlineSMS is open-source software that allows laptop and mobile phone to communicate each other, exchanging messages. Additionally, RapidSMS is free and open-source framework for data collection and communication by sending and receiving messages [20].

One additional example of m-Health systems is called Sana Technology Platform, which was developed by a team at MIT and simplifies the procedure of data collection. Via a mobile device, a user can send medical data that the doctors can view, make a diagnosis and add a medication. The results from the doctors are accessible also via a mobile device. Sana can be integrated into OpenMRS [21] or other medical records systems.

Sana technology platform is an open source application that provides expandability and variety on storage and processing methods [22]. It is also ease-of-use and comprehensible. It provides a database in which texts, images, videos and whole folders can be stored. It ensures safe data transfer without loss or corruption even in areas without reliable internet connectivity. Using Sana utilities allows users to upload medical data, which then are stored into OpenMRS. Appropriate algorithms process the uploaded data and after this step, the results are sent from OpenMRS back to the Sana application. Data upload process requires WiFi, USB connection, GPRS and SMS. Finally, the Mobile Dispatch Server (MDS) is a program that is responsible for the communication between OpenMRS and Sana, receives and synchronizes the data [23]. Figure 2-1 shows the operation of the Sana technology.

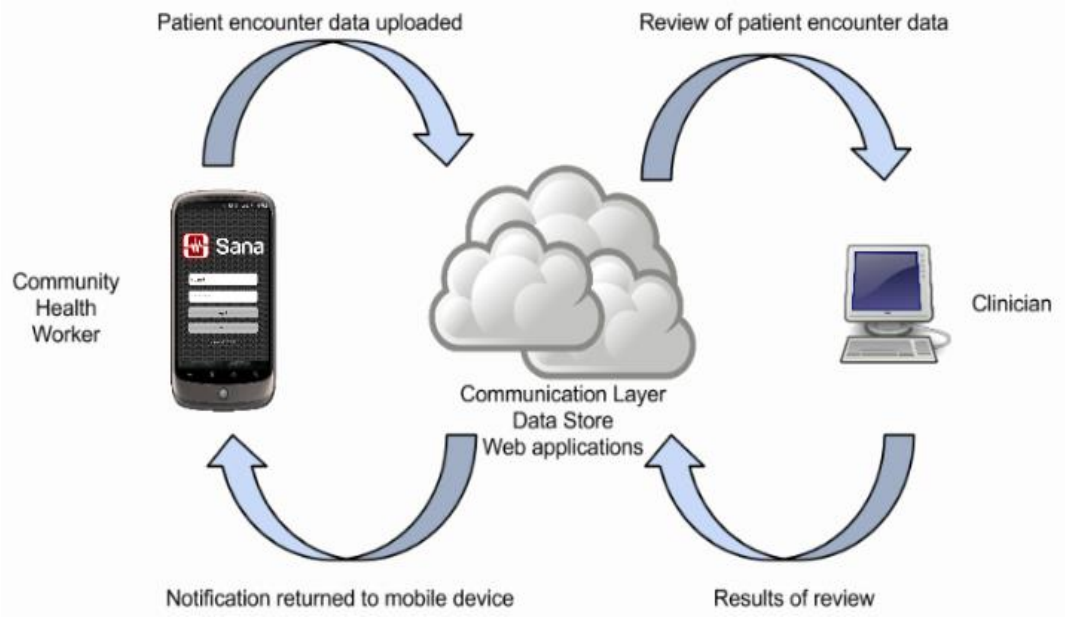


Figure 2-1 Sana Architecture

The development of mHealth systems results in preventing health problems in case of patients who have difficulties in mobility. The way of life is a major factor that the developers took into consideration to build these systems. Mobile applications in combination with sensors have increased the quality of life in people in developing countries [24] and generally contribute in high quality medical services.

Electronic Medical Record Systems

Electronic Medical Record Systems (EMR) target to abolish the established "paper" in hospitals not only in the United States but also in developing countries. The quality of health care has increased using a system that contains all required information. The low cost of the installation and the wide use of those systems are the major benefit.

EMR systems store different types of medical data for many patients. They contain a database whose tables are flexible and in synchronization with each other and are also scalable in case of large-scale systems. They use widely known language and can export data in standard formats. In addition, these systems do not require complex operations and are characterized as secure in terms of privacy [5]. On the network side, EMR systems allow simultaneous data access and insertion from different places using parallelism. Furthermore, data synchronization is a major issue that EMRs provide when there is no internet connectivity. Due to the lack of reliability of the network in developing countries, a local database is used and when the internet recovers, the system synchronizes the medical data [25].

The systems that are preferred in developing countries have not to be expensive for the implementation so the proprietary medical record systems are not recommended. Over time, free proprietary software was developed such as Google Health, Microsoft and Healthvault etc. In order to avoid separated systems satisfying different hospital needs, systems should offer expandability among hospitals. Those systems provide open source software and are available for further processing. The Veterans Health Information Systems and Technology Architecture (VistA) was the first EMR system that supported medical care for the soldiers in U.S. and used a language that was not widely used. Consequently, the use of VistA did not expand. Care2x is another well-known web-based system, which due to disorganization and the lack of structure, was not reliable. OpenMRS (Open Medical Record System) is a descendant of these systems, which needs no programming knowledge and provides adaptability, expandability and free installation. Java is used for OpenMRS programming and runs on the Apache Tomcat web server with a MySQL database. Using the concept dictionary that includes all diagnosis, drugs and all essential information related to medical care, make the analysis easier and accurate. Many developing countries are using OpenMRS such as Kenya, South Africa, Uganda, Tanzania and Zimbabwe [21].

Additional EMR systems are Mosoriot Medical Record System (MMRS), which during data storage process, mistakes happened. In Kenya, Partners In Health (PIH) web system should be reliable to ensure there is no data loss when there was no internet connection. After some years, the problem of the unreliable internet connectivity was solved and the system was working locally. In Uganda, the Careware system is a stand-alone database that uses Microsoft Access. In Malawi, the EMR that is used provides insertion data using a touch screen, which is difficult in case of long data. Similar active and open-source approaches are FreeMED, GNUmed, GNU Health, Hospital OS, HOSxP, OpenEMR, OSCAR, THIRRA, ZEPRS, ClearHealth, and MedinTux [26].

Electrocardiogram

An electrocardiogram (ECG) signal describes the electrical activity of the heart using electrodes on the body surface over a period usually ten seconds. The electrical activity is referred to impulses that are created in a heartbeat and provides information about the heart rate, rhythm, and morphology.

In 1901, a crucial discovery became when Willem Einthoven invented a device, a string galvanometer, which was the first electrocardiograph. The string galvanometer was very sensitive to changes in intensity of electric activity of the heart and stable in operation. Einthoven assigned the letters P, Q, R, S and T to the waveforms.

The ECG into a normal heartbeat consists of a P-wave, a QRS complex, and a T-wave. P-wave corresponds to depolarization of the right and left atrium. QRS complex reflects the depolarization of the right and left ventricles. T-wave corresponds to repolarization of the ventricles. PR Segment is the delay in atrioventricular node and is the interval between the ends of P-wave to the beginning of the QRS complex. PR Interval begins from the start of P-wave to the beginning of QRS complex. QT Interval contains the duration of the QRS complex and T-wave duration and is the interval from the beginning of the QRS complex to the end of T-wave. ST Segment is a period of inertia and is the interval from the end of QRS complex to the beginning of T-wave. There is also the R–R interval that is the time from one R wave to the next.

R-R is useful because we can measure the heart rate [27]. Figure 2-2 depicts the previously described terms used in ECG bibliography.

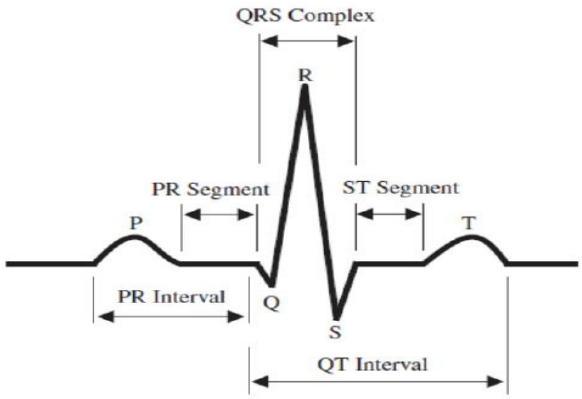


Figure 2-2 An ECG heartbeat signal

ECG Waves

Heart is made of many bundles of muscles that have an electrical charge. In every heartbeat, an electric wave moves quickly through the heart muscle cells. As the wave passes through the heart muscle, provides an electrical current in the chest which flows to the surface and the skin produces electrical voltage differences, which can be measured using electrodes. Electrical activity follows the direction from top right to bottom and left and from back to the chest at the same time and that depends on the position of the heart on the chest [28].

At the beginning of each heartbeat, excitation starts from sinus node (SA node) in the right atrium and passes in as wave through both atria. In the A-V node, a delay is recorded as a straight line (figure 2-3).

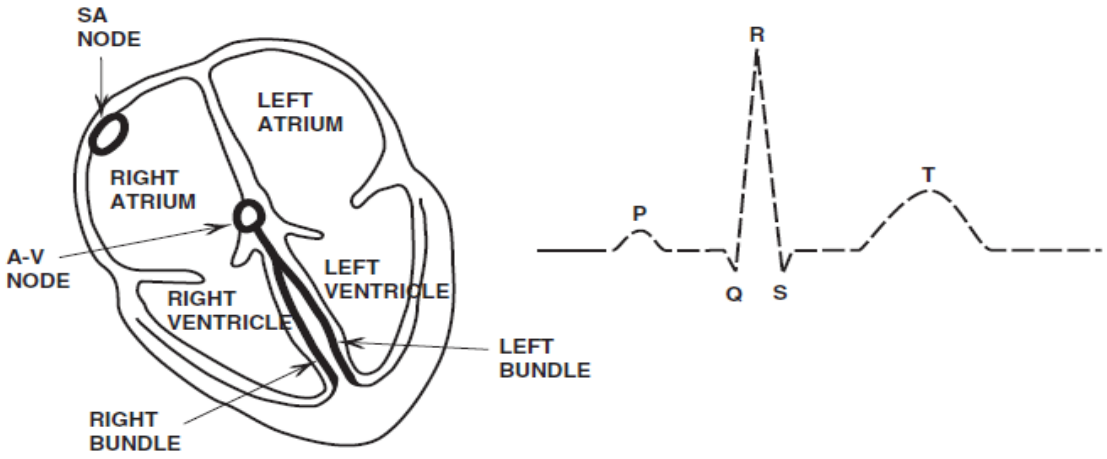


Figure 2-3 Initial state of the heartbeat

As the excitation wave passes through the ventricles, the cardiac muscle contracts. This excitation produces electrical currents that in paper are recorded as a small wave named P-wave (figure 2-4).

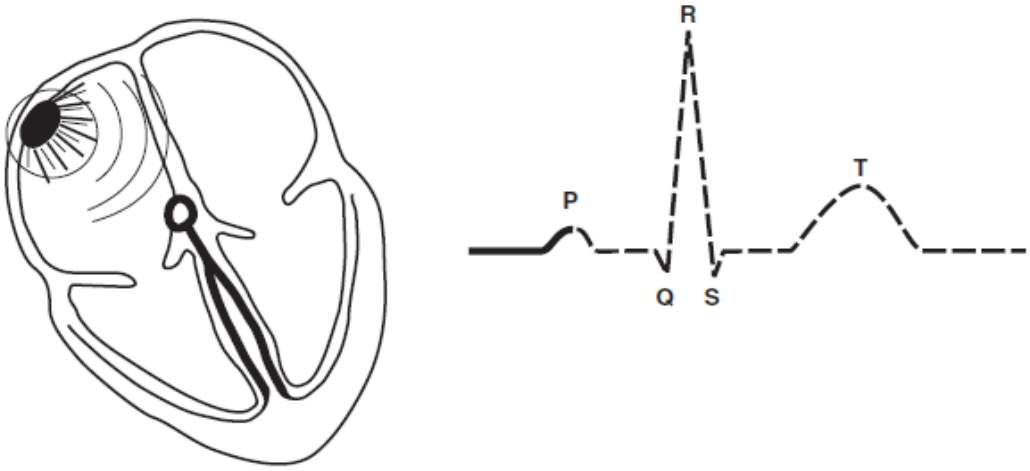


Figure 2-4 P-Wave

The electrical wave creates a delay when approaches the atrioventricular (A-V) node between the atria and ventricles. The P-R interval includes the delay in the A-V node and the duration of the P-wave (figure 2-5).

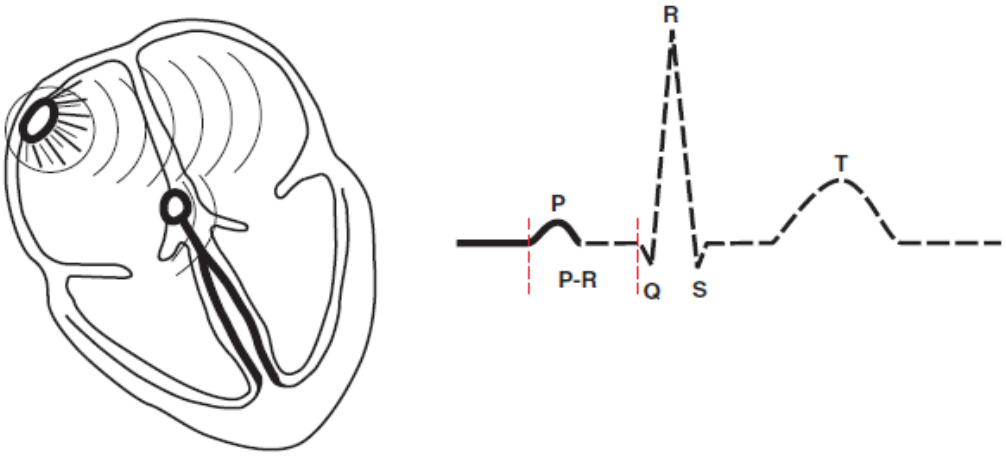


Figure 2-5 P-R Interval

As the electrical wave passes through the muscular septum between the two ventricles, the muscles are activated from left to right. This produces the Q-wave (figure 2-6).

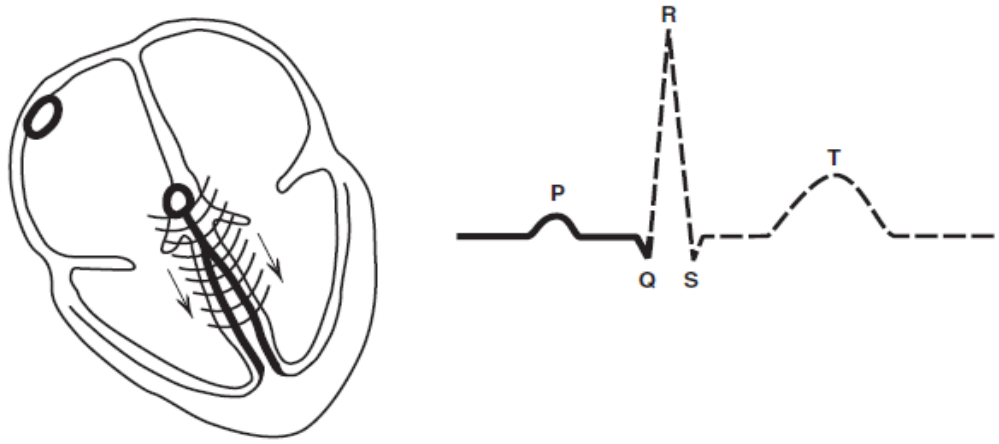


Figure 2-6 Q-Wave

The electrical wave spreads rapidly through the ventricles from right to left direction, activating all the sectors of the ventricles and producing the major ventricular deflection, the R-wave. When the QRS complex is recorded, through the depolarization of the right and left ventricles, pump blood (figure 2-7).

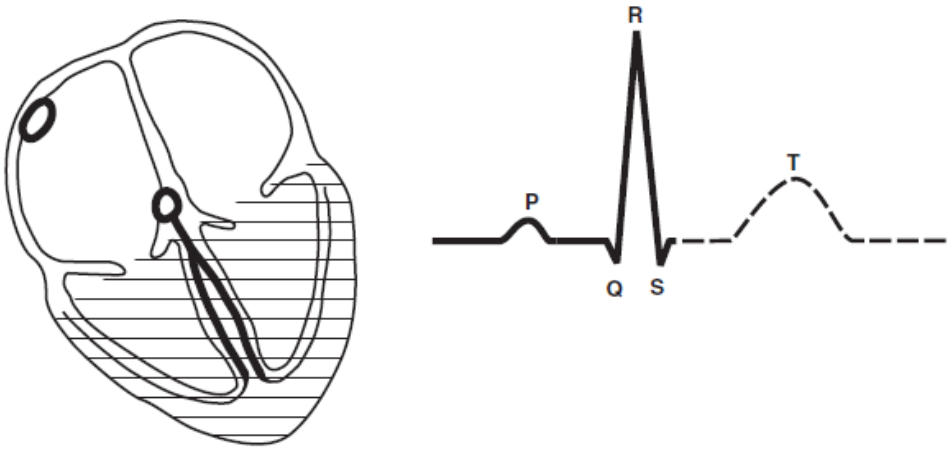


Figure 2-7 QRS-Complex

After QRS complex there is a period of inertia named as ST segment. It follows the QRS complex and represents the time that the ventricles are depolarized (figure 2-8).

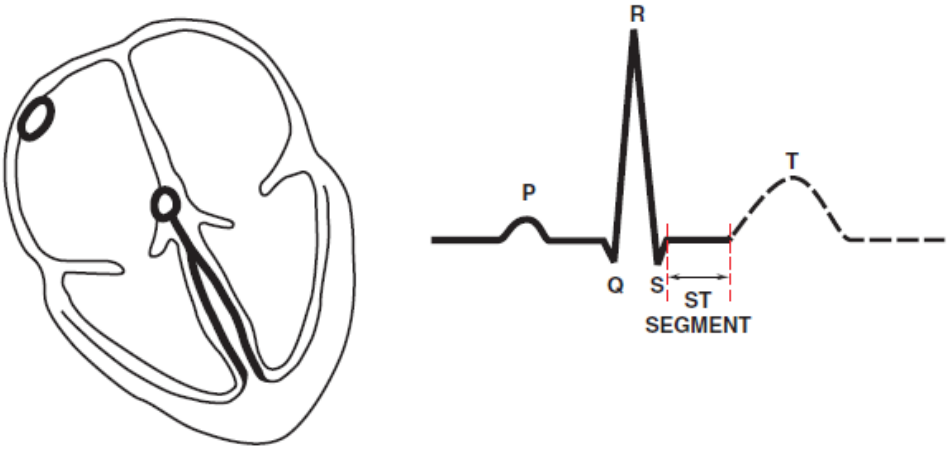


Figure 2-8 ST Segment

Lastly, the wave turns to the reverse of depolarization (figure 2-9), from the epicardium through the ventricular wall, producing a large wave, the T-wave (figure 2-10).

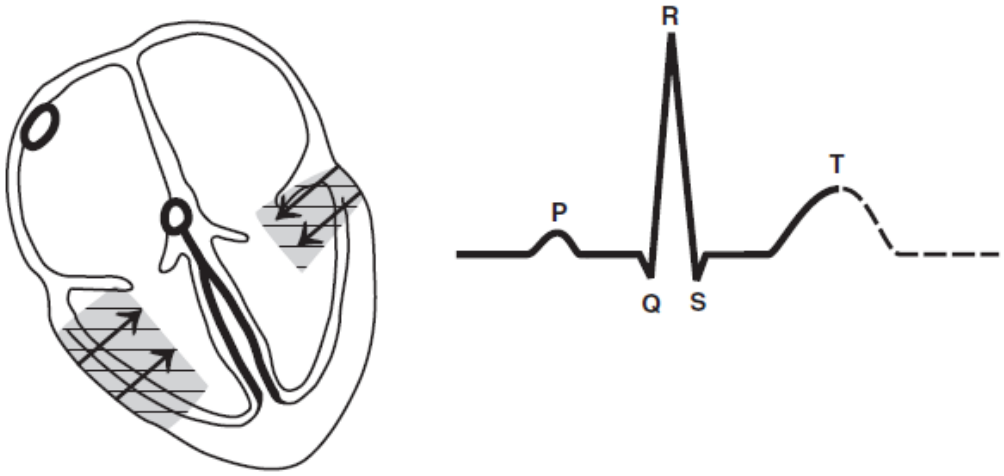


Figure 2-9 Repolarization of the ventricles

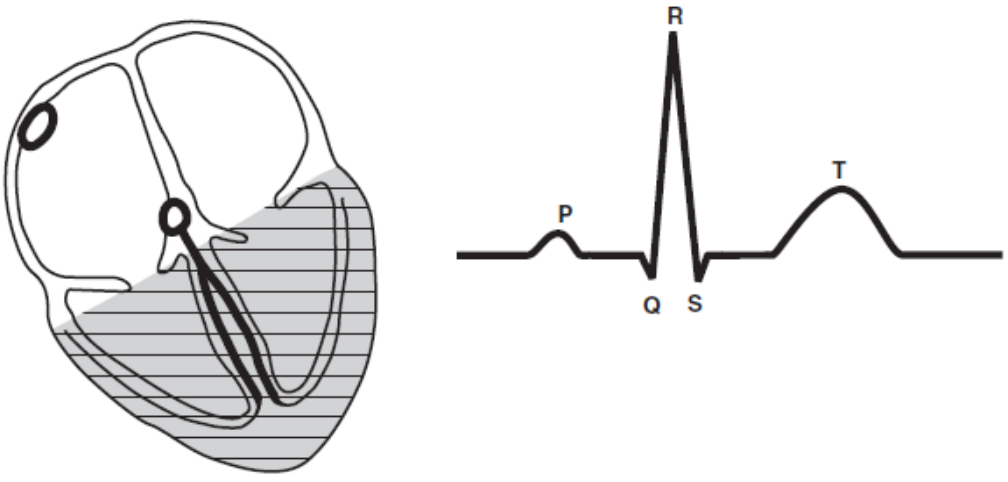


Figure 2-10 T-Wave

ECG Leads

Heart produces electrical current that we can record it. Connecting electrodes between the body and the electrocardiograph, an electrical circuit is made. Each pair of attachments is one "lead". A standard 12-lead ECG needs 10 electrodes. Leads are divided into three sets: Bipolar Limb Leads, Unipolar Limb Leads and Precordial Leads. The 12-lead ECG has a total of three bipolar limb leads, three unipolar limb leads and six Precordial leads. The 10 electrodes in a 12-lead ECG are listed in Table 2-1 [29].

Electrode	Placing Electrodes
RA	On the right arm, avoiding thick muscle.
LA	On the left arm.
RL	On the right leg.
LL	On the left leg.
V1	In the fourth intercostal space (between ribs 4 and 5), to the right of the chest.
V2	In the fourth intercostal space (between ribs 4 and 5), to the left of the chest.
V3	Between leads V2 and V4.
V4	In the fifth intercostal space (between ribs 5 and 6) in the mid-clavicular line.
V5	In the left anterior axillary line.
V6	In the middle axillary line.

Table 2-1 Electrodes in 12-lead ECG

Figure 2-11 depicts the location of the electrodes on human body.

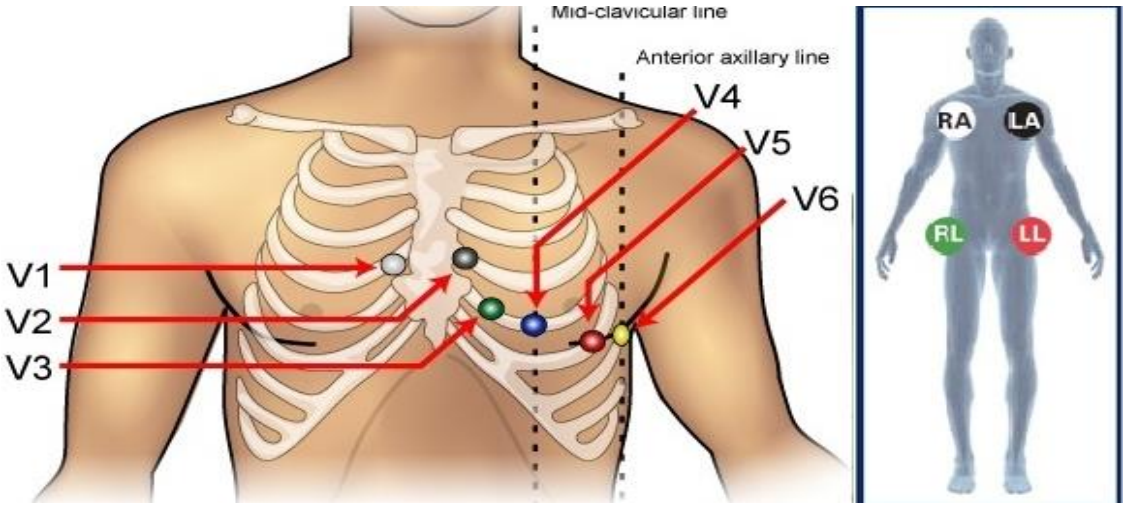


Figure 2-11 Electrodes placement

Bipolar Limb Leads (I, II, III)

Einthoven designed an inverted equilateral triangle (Figure 2-12), whose corners are on the right arm, on the left arm and on the left leg. According to that triangle, the heart is at the center and produces zero potential when the voltages in the three leads are summed. The ECG waves in I, II and III are all positive.

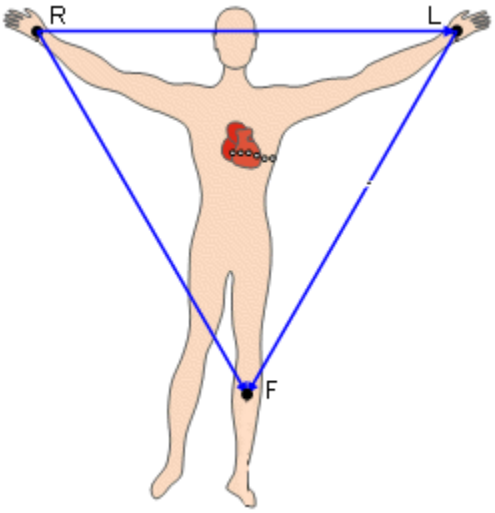


Figure 2-12 Einthoven's Triangle

Lead I is the voltage between the electrode on left arm (LA) and electrode on the right arm (RA):

$$I = LA - RA \quad (1)$$

Lead II is the voltage between the electrode on the left leg (LL) and the electrode on the right arm (RA):

$$II = LL - RA \quad (2)$$

Lead III is the voltage between the electrode that is placed on the left leg (LL) and on the left arm (LA):

$$III = LL - LA \quad (3)$$

Unipolar Limb Leads (aVR, aVL, aVF)

The waves in aVR are generally negative and those in aVL and aVF may be positive but this depends on the anatomic and electrical position of the heart.

Lead augmented vector right (aVR) has the positive electrode on the right arm. The negative pole is a combination of the left arm electrode and the left leg electrode:

$$aVR = RA - \frac{1}{2}(LA + LL) = \frac{3}{2}(RA - V_w) \quad (4)$$

Lead augmented vector left (aVL) has the positive electrode on the left arm. The negative pole is a combination of the right arm electrode and the left leg electrode:

$$\boldsymbol{aVL = LA - \frac{1}{2}(RA + LL) = \frac{3}{2}(LA - V_w)} \quad (5)$$

Lead augmented vector foot (aVF) has the positive electrode on the left leg. The negative pole is a combination of the right arm electrode and the left arm electrode:

$$aVF = LL - \frac{1}{2}(RA + LA) = \frac{3}{2}(LL - V_w) \quad (6)$$

The Wilson's formula V_W is the average of the calibration of the electrodes RA, LA and LL to give an average potential throughout the body:

$$V_w = \frac{1}{3}(RA + LA + LL) \quad (7)$$

Precordial leads (V1, V2, V3, V4, V5, V6)

A lot of information about heart attacks and other heart problems can be found from the six precordial leads that are V1, V2, V3, V4, V5 and V6. The electrodes record positive waves on these leads. Only the T-wave in lead V1 is negative. From V1 to V6 the R-wave becomes taller and the S-wave becomes smaller.

ECG Standard formats

According to the International Organization of Standards, the stored electronic medical data have to be accessible by many authorized users and to transmit in secure [26]. Many digital ECG formats have been implemented so far but all require political commitment and international cooperation.

Digital ECG formats and standards are divided into seven groups [30]. Formats that are supported by the Standard Development Organizations (SDOs) belong to the first group, irrespective of the nature of the data format. The most known formats in this category are the Standard Communications Protocol for computer-assisted electrocardiography (SCP-ECG, European standard), the Health Level 7 annotated ECG (HL7, American standard), the Digital Imaging and Communication in Medicine (DICOM) and the Medical waveform Format Encoding Rules (MFER, Japanese standard). These formats are related to each other and can be either binary or XML-based.

SCP-ECG is one of the most widespread standards for exchanging digital ECGs medical informatics standardization and supported by the European Committee for Standardization (CEN). It is a binary encoded format and intended for short term diagnostic ECGs. Many

proposals for promoting the SCP-ECG standard had been done such as release of open source SCP-ECG tools under GNU General Public License, the development of implementation guides and providing programming contests and tools related to SCP-ECG. The need of the Food and Drug Administration (FDA) for digitalization a great number of varieties of formats annotated ECGs created an XML-based format for digital ECGs, which was the HL7 aECG. DICOM standard was developed for exchanging medical images. A DICOM extension is the handling of biomedical signals such as the ECG. Despite the advantages of DICOM for viewing, interchange, and archiving the ECG signals some users claim that the use of DICOM is limited. The last standard in this category is the MFER, which is a preliminary complementary standard, specializes in medical waveforms. Moreover the MFER standard is expected to integrate into the 11073 group and some improvements have to be done such as a specification for standard 12-lead ECG.

The X73 Family of Standards in Digital ECG contains the less known formats that are supported by SDO. The Vital Signs Information Representation (VSIR) format was used in cardiology included an object-oriented domain information and service model. The File Exchange Format for vital signs (FEF) is a format that leveraged VSIR and biomedical measurements. The next format is the X73-Point (X73PoC) of Care specialization IEEE P11073-10306 for ECG devices and describes the data transfer between ECG Virtual Medical Devices. The X73–Personal Health Devices (X73PHD) standard refers to transmission of ECG data, 1–3 leads, between personal ECG devices.

The second group contains the existing binary encoded formats. The holter applications contain a large amount of data and have different requirements that are covered by the International Society for Holter and Noninvasive Electrocardiology (ISHNE). The large amount of numerical data requires a different file format and organization that the Hierarchical Data Format (HDF) can cover. The last standard in that group is the improvement of SCP-ECG protocol that is called enhance SCP-ECG (e-SCP-ECG+) which overrides the limitations that exist in the SCP-ECG by creating new sections. This format is compatible to the SCP-ECG and handles more vital signs but also demographic data.

The third group is referred to format on eXtensible Markup Language (XML). This group is divided into two proposals, which are the general-purpose proposals such as PhilipsXML, ecgML, XML-ECG and I-Med and specific use case proposals such as mECGML, ECGaware,

Unisens and BSPM-XML. Philips created the PhilipsXML format that was used by its own electrocardiographs, was written in the W3C XML Schema Language and it was made available including documentation and software tools for easy access. The ECG data are compressed using an algorithm without loss and encoded into ASCII using a base 64-encoding scheme. Philips XML format uses Scalable Vector Graphics (SVG) as display format and can connect with other standards like HL7 aECG. The ElectroCardioGraphy Markup Language (ecgML) format is suggested as a solution for integrating ECG data into electronic medical records also provides applications for easier use such as an ecgML generator and an ecgML browser. Its creators refer to the XML-ECG format as a simpler structure, which provides more readability. I-Med standard provides the opportunity to exchange several types of medical data but also basic features such as QRS duration. The Mobile ElectroCardioGraphy Markup Language (mECGML) format is used for exchanging and storing ECG data on mobile devices. The ECGaware extends ECG standards to cover a patient's heart tele monitoring during daily activities. The UNIversal data format for multiSENSor data (UNISENS) provides recording and storing data from different types of sensors such as ECG, blood pressure and respiration rate. The XML-Body Surface Potential Map (BSPM) supports less prominent methods.

The fourth group contains formats that intended for neurophysiology but is also used for ECG signals. Given the fact that the ECG signal has close structure with the neurophysiological signals, such as electromyogram and electroencephalogram, the standards that are contained in that group provides managing all of these signals. The formats in this group are separated into two categories, which are the Data Format family such as EDF, EDF+, GDF, BDF and OpenXDF and the formats initially intended for neurophysiology such as E1467-92, SIGIF, EBS, SignalML and IFFPHYS. The European Data Format (EDF) is a 16-bit format suitable for exchanging time series by supporting multiple sampling rates. EDF+ is an improvement of EDF, which provides interrupted and time-stamped recordings. The General Data Format (GDF) designed to provide some extra modifications to cover some limitations that EDF had, such as coding scheme for events. The BioSemi Data Format (BDF) is a 24-bit version of the 16-bit EDF format, which was designed for electroencephalography applications but is applied in similar signals. The Open eXchange Data Format (OpenXDF) is referred to as an XML-based extension of the EDF format. The E1467-92 format is a format that was designed to transfer digital neurophysiological data but later through expandability was transferring more types of digital

data such as ECG. The SIGnal Interchange Format (SIGIF) is characterized by versatility and adaptability according to its designers and can store both raw and processed data. The Extensible BioSignal EBS file format is a binary file format for storing multichannel time-series recordings. The Signal Markup Language (SignalML) was used to avoid problems of incompatibility of different formats of digital data. The Interleaved File Format for PHYSiological data format (IFFPHYS) is an extension of the IFF format and provides storing ECG signals and other physiological signals.

The fifth group is referred to main existing database formats. Many organizations have created their databases for researches and experiments. Open data format, software and reference materials are available by those organizations to handle their formats. Moreover, the databases include ECG data in a form compatible with the Standard formats such as EDF, BDF and SCP-ECG. Some of the existing databases are the Massachusetts Institute of Technology, Beth Israel Hospital (MIT-BIH) database, the Multiparameter Intelligent Monitoring in Intensive Care (MIMIC) database and the Physikalisch-Technische Bundesanstalt (PTB) database, which are supported by the Physionet component and its Waveform Database (WFDB) software. More databases are the American Heart Association (AHA) database, and the Common Standards for Electrocardiography (CSE) database.

The sixth group contains the Integrating the Healthcare Enterprise (IHE) profiles that are referred to ECG domain. IHE Cardiology is related to information sharing, workflow, and patient care in cardiology and its profiles contain stable documents, such as the Retrieve ECG for display, draft for trial implementation, such as the Resting ECG Workflow (REWF), and draft for public comment such as the Waveform Communication Management (WCM). IHE Cardiology Framework integrates existing standards, usually HL7 and DICOM.

The seventh group contains existing and ongoing works on ECG ontologies. Ontologies are embraced to define controlled vocabularies for shared use across different medical domains. Such ontologies are the SCP-ECG Ontology (SEO), the National Center for Biomedical Ontology (NCBO) and the NEMO.

There is one more group, which contains the standards that were created by manufacturers to use them in their own ECG devices. Such standards are the Siemens Interchange Format for medical records (SIFOR) by Siemens, the Unipro by Mortara and the

ECG-9x by Nihon Kohden. Many manufacturers have declared that their standrads are compatible with the SCP-ECG.

In the current thesis, the PTB database is used. PTB database contains a compilation of digitized ECGs supporting the SCP-ECG format and is provided by the National Metrology Institute of Germany. It is supported by the Physionet and the WFDB software and is widely used for research purposes. Healthy control ECG signals are provided but also pathological ECG signals such as myocardial infarction and heart failure for experimental setup.

ECG Signal Filtering

Electrocardiography (ECG) studies the electrical activity of the heart that occurs over a period from an electrode attached to the skin. The ECG signal consists of low amplitude voltages and noise. The desired signal is ±0.5mV but the maximum offset voltage of electrode is ±300mV. Some sources of noise are the baseline displacement, which creates low frequency noise, the power line interference, which creates 50Hz or 60Hz noise, the muscle noise and other interference such as radio frequency noise that is created from other equipment in the environment but also from within the ECG equipment itself. To cover all the causes that affect the ECG signal, different types of filters are used [31]. A filter removes or reduces frequencies where noise occurs, allowing the signal frequency through. This can be prevented either by hardware or by software. There are many methods for removing noise from ECG signal. The classical filters for ECG are shown below.

There are two types of digital filters on basis of impulse response that are used on ECG signal processing, the FIR (Finite Impulse Response) and the IIR (Infinite Impulse Response). The FIR filter has the equation

$$y(n) = \sum_{k=0}^{M} b_k x[n-k] \tag{8}$$

and its output depends only on present and previous inputs and does not depend on previous outputs. The IIR filter has the equation

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=1}^{M} b_k x(n-k) \tag{9}$$

and its output depends on previous inputs, present inputs and on previous outputs [32].

FIR filters are separated into two types. The first one is the window method and the second one is the frequency sampling domain method. The window method selection is determined by the required attenuation value in the cut-off zone and includes four types, which are the Rectangular window, the Kaiser window, the Hanning window, the Hamming window and the Blackmann window. The most usual IIR filters are the Butterworth filter, the Chebychev Type I and II filters, and the Elliptic filter.

Breathing or electrode issues affect baseline (isoelectric line or isoline). The first method to remove the alternations is to use a FIR (Finite Impulse Response) filter design with Hamming window. Another filter that is used is the Butterworth IIR (Infinite Impulse Response) HP (High Pass) filter, which its feature is the nonlinearity of phase frequency characteristic of IIR filter. The IIR filter with zero-phase frequency characteristic and at the output is zero-phase IIR filter. Another method that can remove the alternations is to detect the regions that have zero potential. Using the points of these regions, the line that is interpolated is the baseline and so is reconstructed. Moreover, a method that is also used is the median filter, which is usually used for removing impulse noise. The window width for median evaluation must be such that the waves are not deleted [33].

The noise from the ECG signal can be removed using more than one filter. The fact that the upper and lower frequencies of the pass band are apart we can use separately low pass and high pass filters. One method for removing noise is the use of FIR filters. Notch filter combines both high and low pass filters, attenuates several singular frequencies and the rest remains. Another method for removing that noise is to use an adaptive filter using an LMS (Least Mean Square) algorithm and correct adaptation coefficients. The adaptive filter is used in periodic distortions and when the spectrum of noise interferes with the spectrum of useful signal.

In the current thesis, we used the interpolation method to remove the alternations of the baseline. We detected the region with zero potential, which is from the end of a beat to the beginning of the next beat, and we remade the baseline. A low-pass FIR filter was used and the rectangular window method, or “boxcar” filter, to remove the noise of the ECG signals.

Chapter 3 - Physionet

Physionet allows free web access to a large collection of recorded biomedical signals, which are stored in a database, PhysioBank. It is also related with open source software for research the PhysioToolkit. Physionet funded by National Institute of General Medical Sciences (NIGMS) and the National Institute of Biomedical Imaging and Bioengineering (NIBIB) and managed by members of MIT's Lab for Computational Physiology [34].

PhysioBank

PhysioBank contains collections of Electrocardiographic (ECG) signals, Neuroelectric signals, Myoelectric signals and other signals from patients who have different healthy problems like congestive heart failure, epilepsy and sleep apnea. The databases are separated into two archives. The first one includes Clinical databases and the second one, Waveform databases. Clinical databases contain data from intensive patient care, which may include demographics, vitals measurements, laboratory test results, drugs and notes that the caregiver has observed. Waveform databases are high-resolution records of physiological signals, are categorized according to their signal and annotation types, and are listed below.

Multi-Parameter Databases include signals that related to Electrocardiogram (ECG), blood pressure, respiration, oxygen saturation and Electroencephalogram (EEG). ECG Databases contain ECG signals. Interbeat (RR) Interval Databases contain beat annotations received from ECG recordings. Cardiovascular Databases contain blood pressure signals before and after a salt diet. Gait and Balance Databases contain gait duration in Neuro-Degenerative Diseases like Parkinson’s disease. Neuroelectric and Myoelectric Databases include Electroencephalographic and Electrohysterographic signals. Image Database has magnetic angiography images. Synthetic Databases contain synthetic fetal phonocardiographic signals. In addition, other databases contain signals, with and without Parkinson’s disease, in order to indicate the interaction with computer keyboards to detect signs in the early stages of Parkinson’s.

The database that used in the current thesis is the PTB Diagnostic ECG Database that is part of ECG Databases in the archive Waveform databases. This database contains 549 high-

resolution ECG signals from 290 patients aged 17 to 87, 209 of them are men and 81 women. Each patient has one to five records and each record contains 15 measured signals which are the typical 12 leads (I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, V6) together with the 3 Frank lead ECGs (VX, VY, VZ). For typical leads, ten electrodes are used and three more electrodes for Frank lead. Each signal is digitized at 1000 samples per second, with 16-bit resolution over a range of ± 16.384 mV.

Records contain three files. The binary signals file (.dat) that contains the signal, which will be read by the PhysioToolkit. The header file (.hea) includes some information about patient such as age, gender, medical history, medication and details about Electrocardiogram like echocardiography and ventriculography. The binary signal file (.xyz) contains the three Frank lead and is optional.

Each record has different diagnoses, which are Myocardial infarction, Cardiomyopathy/Heart failure Bundle branch block, Dysrhythmia, Myocardial hypertrophy, Valvular heart disease, Myocarditis, Miscellaneous and Healthy controls.

PhysioToolkit

PhysioToolkit is a collection of open source software using for viewing, analyzing, and simulating physiologic signals. The software consists of Waveform Database (WFDB) Functions and WFDB Applications. Both Functions and Applications are developed in C language and take pointer arguments so allocation of memory is necessary.

The functions are divided into categories. To select database records we use functions that are in that category like *annopen, isigopen, osigopen, osigfopen* and *wfdbinit*. To set or return the input sampling frequency, functions that are used are in the category Special I/O Modes, like *setifreq, getifreq, setgvmode, getgvmode, getspf* and *getafreq*. The next category is Reading and Writing Signals and Annotations. These functions are *getvec, getframe, putvec, getann, ungetann* and *putann*. There are also functions that provide Non-Sequential access to WFDB files. Such functions are *isigsettime, isgsettime, tnextvec, iannsettime, sample* and *sample_valid*. The functions that are used for conversion are *annstr, anndesc, ecgstr, strann, strecg, setannstr, setanndesc, setecgstr, timstr, strtim, datstr, strdat, aduphys, physadu, adumuv*

and *muvadu*. For calibration, we use the functions *calopen, getcal, putcal, newcal* and *flushcal*. The memory allocation functions are *memerr, sfree, sualloc, salloc, srealloc* and *sstrcpy*. There are also Miscellaneous Functions such as *newheader* or *setheader*, which create header files, *wfdbquit* that closes all open WFDB files and frees commited memory by other WFDB functions.

The applications, using the above functions, produce some results through operations. The applications that were used for the current implementation are the *Rdsamp,* which reads WFDB signal files and writes the samples as decimal numbers into a text file. WFDB signal files are binary, and usually contain either 16-bit amplitudes (format 16), pairs of 12-bit amplitudes bit-packed into byte triplets (format 212), or 8-bit first differences (format 8). They have also the suffix .dat. The text file contains 13 columns separated by tabs. The first one contains the elapsed time of the ECG and the next 12 columns contain the ECG leads (I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5 and V6) in decimal numbers which represent the mVolts.

Wrsamp reads a text file that contains the samples and writes them into WFDB signal files. Wrsamp also creates a header file that contains the attributes of the signal and some information such as sampling frequency. We use wrsamp to multiply the data, which are now decimal, by 1000. This processing is important because the filter receives integers as input. To multiply the data we use the wrsamp application. Wrsamp takes a text file as input, multiply all columns by 1000 and produces two files. The one is binary with the suffix .dat and the other is the header file with the suffix .hea.

Fir application is a FIR filter that is using for WFDB records. It can be used to apply any desired finite impulse response filter. The filter that was used in the current implementation is a low-pass boxcar filter that attenuates the high frequencies and reduces the noise that an ECG signal may have. The outputs of the fir application are also two files. The first one is the binary file that contains the signal and the second one is the header file. The binary file needs to be readable for processing and the using of rdsamp is required. The data that are used and display in the platform are filtered.

Sqrs detects QRS complexes in an ECG signal. The input of sqrs application is the binary file and the header file of the ECG signal. The output of sqrs is an annotation file, which is also a binary file. The output is an annotation file with the suffix .qrs that is used as input by ann2rr

application and ihr application. *Rdann* reads WFDB annotation files and writes the results into a text file.

Ann2rr is used to obtain R-R intervals from ECG annotation files and writes them into a text file. R-R interval is the distance between two heartbeats. Ann2rr application takes the location of the QRS complexes that sqrs has estimated and determines the R-R intervals in the ECG signal. The text file, which is produced from ann2rr, contains three columns. The first one has the start time of each QRS complex, the third column has the stop time of the corresponding QRS complex and the second column contains the interval from start time to stop time.

Ihr application takes also the location of the QRS complexes and calculates the instantaneous heart rate, which is stored in a text file. To find the heart rate of the whole ECG signal we calculate the average of the instantaneous heart rates.

The last processing is to define the ECG waves in the ECG. The ECG waves are the P-wave, the QRS complex and the T-wave. To read the file we use *ecgpuwave* application. For each lead we use the ecgpuwave once and every time we determine which signal to process. We have 12 leads, so ecgpuwave will process each lead separately. The output for this application is an annotation file that is has to be readable. To convert the annotation file into text file we use the *rdann* application. We call ecgpuwave 12 times for each signal so we have to call 12 times the rdann to convert into text file each signal. Rdann takes the annotation file as input and produces a text file. Each text file contains two columns. The first column contains the start time of each wave, the location time and the stop time. The second column has the number of sample that each wave starts, its placement and the number of sample that it stops. The third column contains the waves. When a wave starts, an open parenthesis is used '('. To determine the waves, the letters p, N, t are used. The letter 'p' is referred to the P-wave, the letter 'N' referred to QRS complex and the letter 't' referred to T-wave. When a wave stops, a close parenthesis is used ')'. As we can see from the table 5-1, we determine the interval of each wave. Using these intervals and the filtered data, we also can calculate the amplitudes of each wave. Taking the start and stop time of this text file, we search into the text file after filter the corresponding heights.

Time	Sample	Type of wave
0:00.542	542	(
0:00.572	572	p
0:00.608	608	)
0:00.651	651	(

0:00.710	710	N
0:00.747	747	)
0:00.870	870	(
0:00.941	941	t
0:01.030	1030	)

Table 3-1 ECG Waves after ecgpuwave Application

Chapter 4 - OpenMRS – Sana

History of OpenMRS - Sana

In '90s, students from Indiana University School of Medicine in the United States and Moi University in Eldoret, Kenya, cooperate for a health care system project. The information system that was used was Microsoft Access. Because of the large amounts of data, a new system has to be created. At the same time, a new non-profit project was created in Boston with the name of Partners in Health (PIH), which is web-based.

At the World Congress on Medical and Health Informatics (MedInfo) conference in San Francisco, the director of the program in Boston and the students from Indiana and Moi Universities met and found that they had the same needs and decided to create a new information management system together. That system was OpenMRS.

OpenMRS is an electronic medical record system (EMR) that established in 2004 to provide healthcare in developing countries. The system has evolved into a medical informatics platform that is used all over the world to provide healthcare and avoid wasting of time and money providing effective cure and prevention. OpenMRS is an open-source project, which with a compilation of other open-source projects or mobile phone applications can be extended or modified [35].

OpenMRS satisfies functionalities such as patient registration and retrieval clinical notes in the system and secure drug prescription. In addition, there are no specific hardware requirements [25] and a dedicated implementers Wiki where the members direct anyone new developer or implementer [36]. Furthermore, it is compatible with mobile devices such as Sana platform, which allows users to send medical data that the doctors can view, make a diagnosis and add a medication [22] using a simple mobile phone.

Sana is an open source telehealth platform that was developed by MIT (Massachusetts Institute of Technology) to provide health care facilities in remote areas and increase the quality of care. Sana uses OpenMRS for data storage and supports the connectivity with mobile device. Healthcare assistants collect medical data via Sana platform and these data are sent to OpenMRS where doctors can view and make a diagnosis. Sana supports remote patient monitoring and decision making [37].

An example of the using of OpenMRS is at the Amani clinic. A European nongovernmental organization provided funding for the establishment of a new health care facility in Kisiizi, a small town in southwest Uganda. Amani Clinic aims to cover the needs of mother and child in the city and surrounding areas. The agency is seeking the funding to be applied in a clinical information system for management to better monitor patients and suggests OpenMRS.

Each patient, who went to the clinic, was given his personal folder, which was kept in the archive. Each doctor who examined the patient added the necessary documents required for each visit by completing the folder. At the end of the visit, the patient returned the file back to the archive. OpenMRS implements the above scenario. The patient's folder is no longer in a clinic room but is an information system accessible to the clinic's manager, doctors and others users.

Technical Specifications of OpenMRS

OpenMRS can be installed in two ways, either Standalone or Enterprise. It requires Java 6 or higher to run OpenMRS.

Standalone is a simplified installation that contains a web server, embedded database and allowed to run on localhost. It is ideal for data less than 10,000 patient records. Enterprise is used in the case of more than one subscription when there are already installed Java Servlets and a database and only OpenMRS needs to be set up on them.

In the current thesis, the standalone version was used. Enterprise is suitable for a larger number of records, but has greater demands on computing resources such as CPU, RAM, and Storage. The hardware depends on the size of the implementation. If the implementation is small, it can work on a laptop. 100s of patients require a 1 GHz processor or better, 256 MB of memory or more, 40 GB hard drive or larger. 10,000 patients are required 1.5 GHz processor or better, 2 GB of memory, 150 GB hard drive with RAID (Redundant Array of Independent Disks) and backup facilities. RAID provides reliability, availability, performance and capacity. Over 250.000 patients 2.26 GHz quad processors, 16 GB of memory, 500 GB hard drive with RAID and backup facilities are required [38]. Standalone version is also more user-friendly. Lastly, the database that is provided and the web server are running without needing any pre-installed.

To run the Standalone version we need to execute the command java -jar standalone-1.1.jar from the command line inside the folder where OpenMRS is stored on Windows or the command ./run-on-linux.sh in the terminal on Linux. By default, OpenMRS runs the MySQL database on port 3316 and the Tomcat server on port 8081.

At Amani Clinic, the Enterprise version was used. Although there were not many registrations, the operators were more familiar with Apache Tomcat and MySQL.

Architecture of OpenMRS

OpenMRS is a Framework built upon Java and other related frameworks. It is based on modular architecture, which provides additional functionality. A module is a packaged Java code that can be installed into a running OpenMRS instance and is able to modify almost all aspects of OpenMRS. It can provide web pages, operation of the services and add new functionality [39].

There are six basic types of modules. Core Modules, Bundled Modules, Community Supported Modules, Community Owned Modules, Community Modules and Commercial Modules.

Core Modules are necessary for the functionality of OpenMRS. We use Core Modules when we want to take extra functionality into the core of OpenMRS. If a Core Module is missing, the System cannot work [40].

Bundled Modules are modules that are pre-packaged into OpenMRS. We also have functionality with these modules but if a user wants to delete one, the system will run correctly without the specific module. Most of them are required by other modules to run. An example is the HTML Form Entry module that needs the Form Entry to run [41].

Community Supported Modules are modules that community of OpenMRS will ensure their maintained [42].

Community Owned Modules are modules that will be developed and maintained as long as the community needs them.

Community Modules are modules created by anyone and share with the community of OpenMRS.

Commercial Modules needs to be purchased or has a subscription fee to be used.

The file extension for modules is ".omod". However, a module is simply a jar underneath and a jar is a special zip file. Modules can add their own jsp web pages and also add and modify tables in the database.

Figure 4-1 depicts the architecture of OpenMRS.

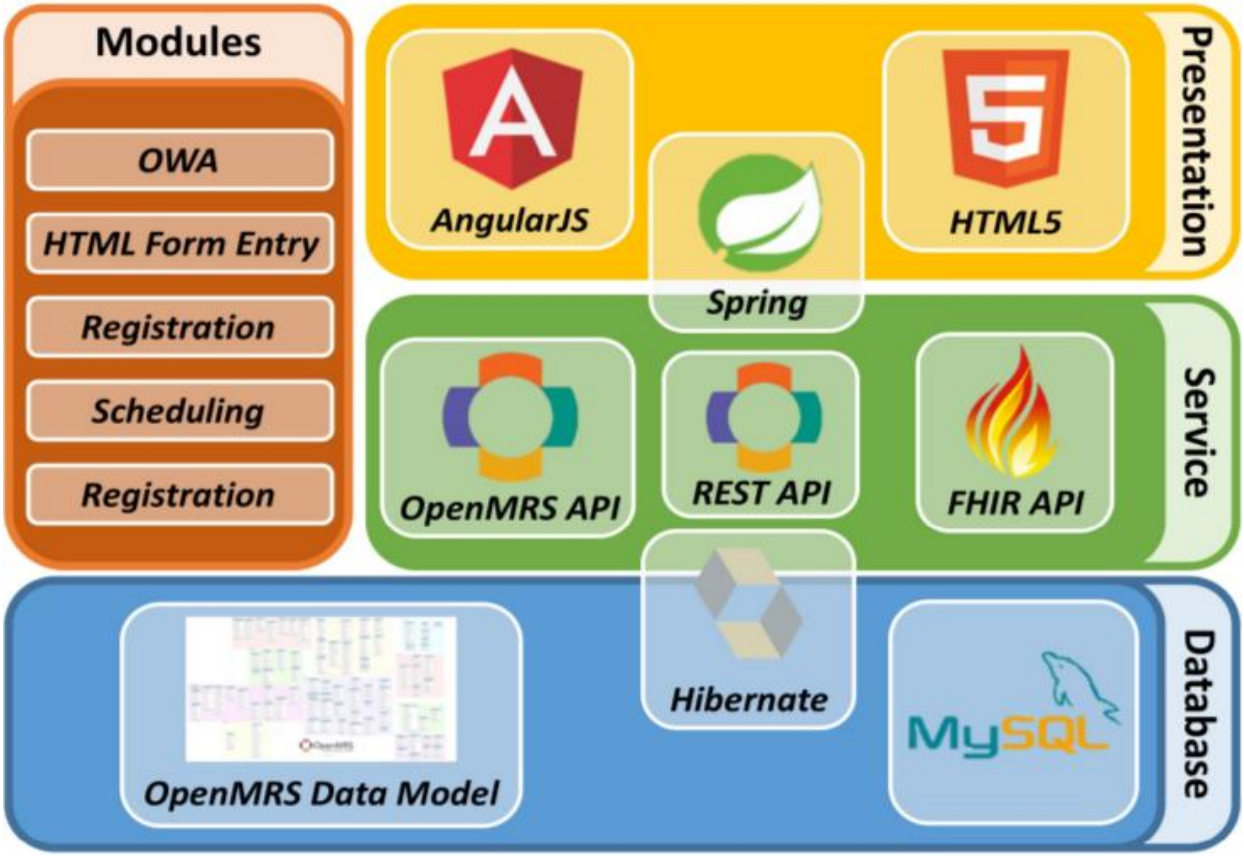


Figure 4-1 Architecture of OpenMRS

The basic part is the OpenMRS API [43]. This API has methods that are called classes, for all of the basic functions like adding a patient, adding a concept and more. An example of classes is the *PatientService* that can create, delete and display a patient. The functionality of these classes is on Service Layer and is developed on Java.

There are three levels on OpenMRS architecture. These levels are the *Data Access* Layer, the *Service* Layer and the *Presentation* Layer or *User Interface*.

The Data Access Layer is the layer that is related to the database. It uses Hibernate as the Object Relational mapping tool. Hibernate describes the relationship between tables of the database and the domain objects using XML mapping files. In this layer, Liquibase is also used

to manage and apply database schema changes in a database-independent way. The Data Access Layer is exposed to the Service Layer through interfaces.

The Service Layer is built around the Spring Framework and is responsible for the business logic of the application and to manage the transactions between the classes.

The Presentation Layer is built upon AnjularJS, JSP, JavaScript and JQuery for interaction between JavaScript and browser.

OpenMRS Data Model

In the data model, we can see how the information is stored. The data model is divided into ten domains. Each domain has different color as we can see in Figure 4-2.

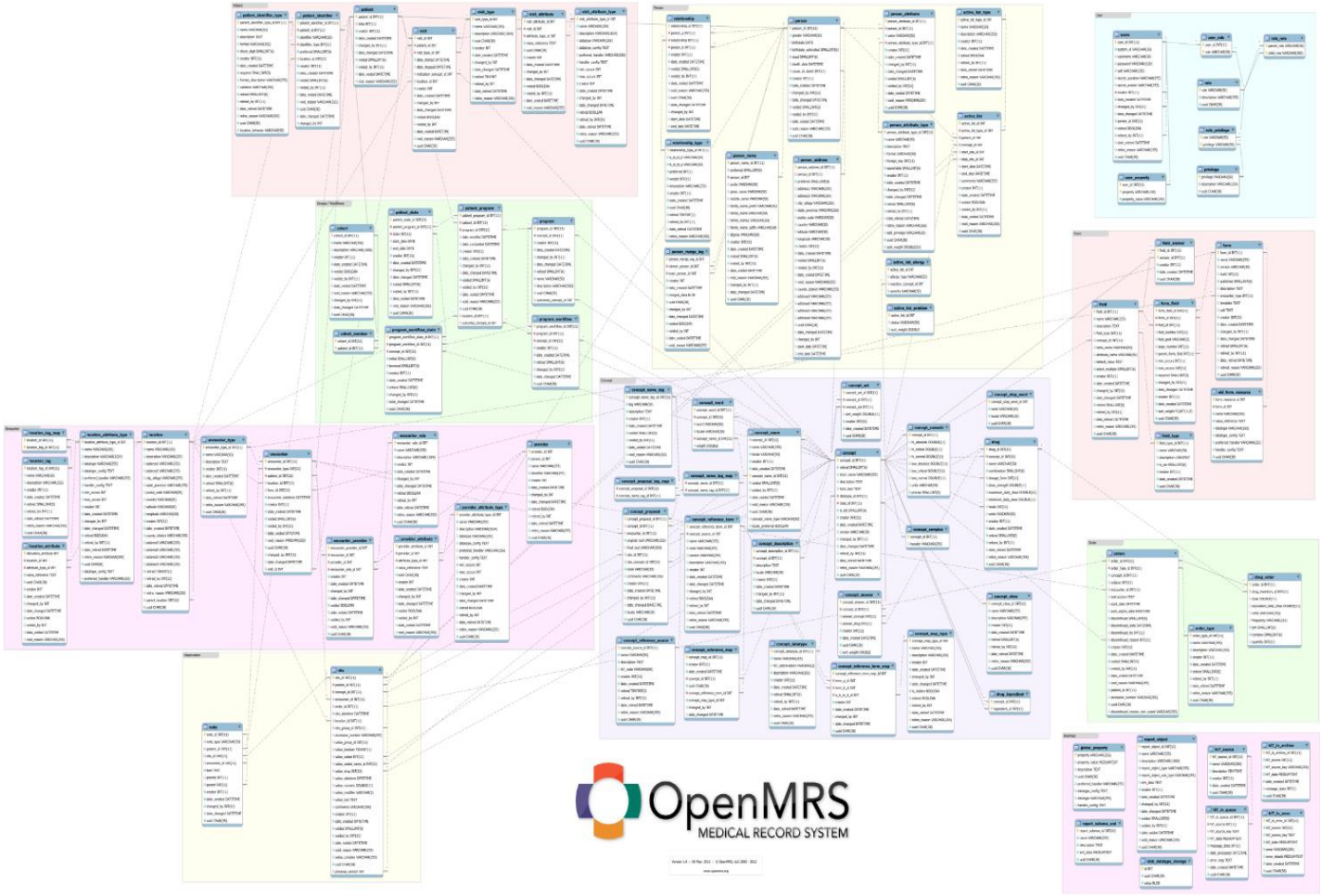


Figure 4-2 Domains in Data Model

The domains are Concept, Encounter, Form, Observation, Order, Patient, User, Person, Business, and Groups/Workflow. Figure 4-3 depicts the domain Patient and the tables that are included.

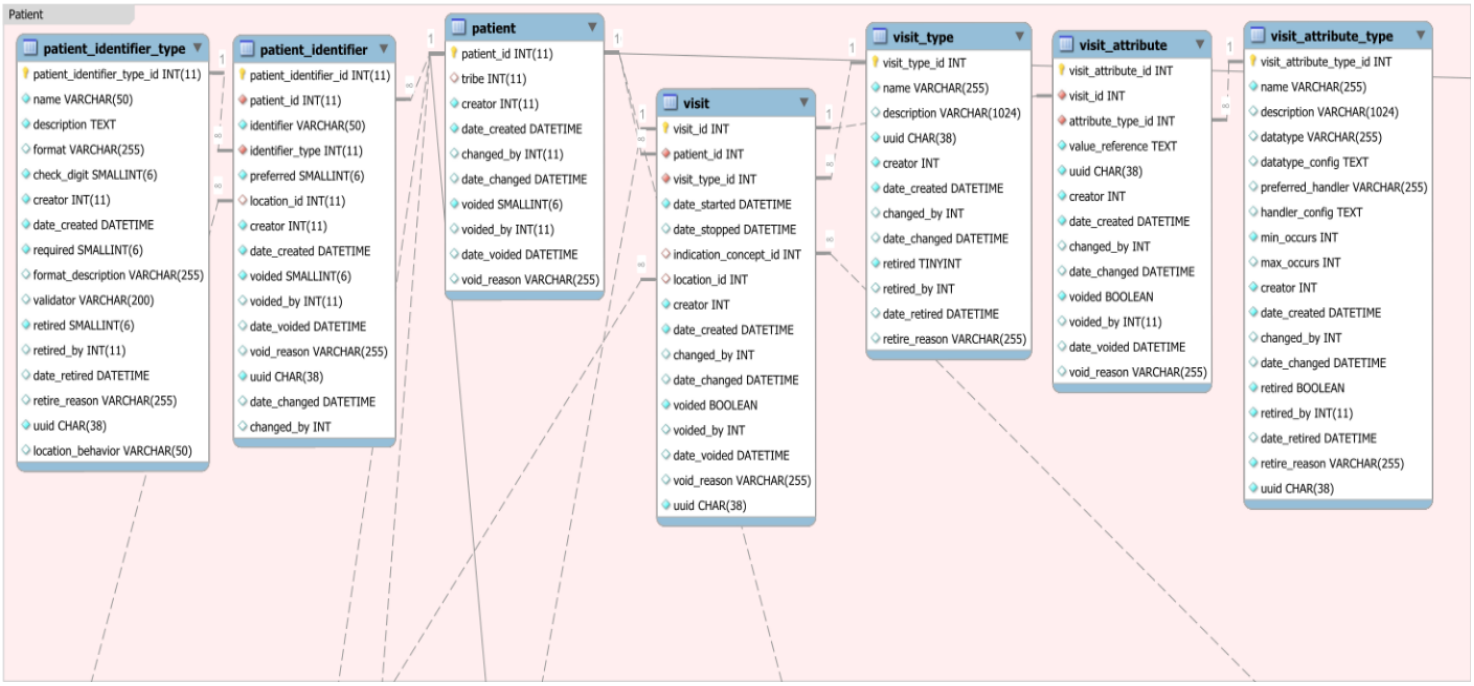


Figure 4-3 Patient Domain

Definitions

The records stored in OpenMRS are called Data. Patients, Encounters and Observations are an example of Data. All Data have an ID or UUID to guarantee their uniqueness. The UUID value will be hexadecimal digits with 36 characters in the form of 8chars-4-4-4-12chars for example 0febc204-bca1-11de-913d-0010c6dffd0f [44].

OpenMRS ensures the stored Data in the database by not erase them. If we delete a record accidently, the value *voided* is placed in the corresponding database field. Data is not deleted but they do not appear in the platform in this way, and in the case of inversion, the corresponding value becomes *unvoid*. In order to delete data permanently, an administrator must delete them from the base where the operators do not have such special access.

Metadata: OpenMRS designed with extensibility capabilities. Depending on the requirements, OpenMRS can be customized. Some changes that need to be made are, for example, the locations of the hospitals. Another example is that in a clinic it may be necessary to record the weight of patients. That parameter is useful and we need to adjust and add. Electrocardiogram is also considered as Metadata, which will be dealt with in this thesis. Such parameters, like location, weight and electrocardiogram, are called metadata. In each hospital, different data are collected in the personal folder for every patient. As it happens with Data, Metadata are not deleted from the database of OpenMRS. Some metadata can be removed for future use but they are not deleted. For example, a doctor may be transferred to another hospital. Records that have made by the same doctor will not be deleted but they will continue to appear in the platform, but in the future the name of the doctor does not appear to the active doctors but it may be found as retired. In case of a doctor or a location set as *retire*, the administrator has the ability to *unretire*.

Concepts and concept dictionary: Concept Dictionary is a list with medical terms and anything else related to questions and answers about Observations. Any data that we want to store for a patient must be defined as a Concept. An observation is a single unit of clinical information. An example is the data we collect for a patient's Electrocardiogram. The Electrocardiogram is the question and contains information that has been pronounced as Concepts, such as pulses. The Pulse Concept is called Bpm and the answer is a Numerical Value that gets a value after running an algorithm. In the case of the patient's blood type, the question is the blood type. The answer can be answered by A, B, AB and O, which are fixed. These responses can be defined as four different Concepts each of them contains a response that is referred to the blood type.

Persons: Each person who mentioned in OpenMRS is called Person. Persons can be patients, doctors, nurses, users, and whoever is related to the patient. All Persons have the same attributes that must be defined such their name and address. In addition to these features, we can also define some additional features such as their phone. This feature is ranked as Metadata. Such Metadata that refer to extra features of Persons are called *Person Attributes Types*.

Patients: Each Person who gets medical care in OpenMRS is Patient. Patients are Persons so they must have, at least, name and address. Each Patient has also at least one Identifier. Patient Identifier or Patient ID is a medical record number, which is created during registration of the patient in OpenMRS and is used for searching the patient into the database and identify him in a future visit to the hospital. There is a possibility to define a Patient Identifier Type that specifies how the Patient ID will be created, that is the format the ID will have. The OpenMRS IDs are produced by a variation of the Luhn algorithm [45].

Relationships: Relationships that we can declare to OpenMRS are Metadata and refer to the relationship for example patient-doctor and parent-child. At Amani Clinic, the clinic was serving mothers with their children so it was necessary to create a Relationship that would involve each mother with her child. Relationship can also exist between a nurse and the patient or a caregiver and the patient. Metadata that are describing a particular type of relationship is a *Relationship Type*.

Visits: Visit is when a patient shortly interacts with the system in a specific location. Metadata differentiate each visit to *Visit Type*. If the visit is for example in the patient's home or via telephone, then we have a Visit Type that specifies the type of visit. One patient checks in at the clinic and starts his Visit until the checkout time when Visit completes. The Visit contains Encounters that are added to the patient's card from each Provider patient visits.

Encounters: The data that are collected each time a patient visits a Provider. Encounters are part of Visits. Observations are collected and grouped together into one Encounter (one Visit). Each Provider, during a patient visit, completes a form with his / her observations. This form creates an Encounter that is contained in a Visit. Metadata defines the *Encounter Type*. The Electrocardiogram is an Encounter Type that is completed during the patient's visit, whenever is necessary.

Providers: Provider is any person serving a patient like a doctor, a nurse or even a social worker. Every Person, who can produce an encounter, is a Provider. The Provider is also a Person who must have a name, address, and ID.

Locations: Any place that the patient can visit, within a clinic, is referred to as a Location. Such example may be a laboratory or external medicine. There may also be another location outside the clinic, which should refer to the visit as a provincial hospital.

Observations: Observations can be seen as a piece of information that is being imported for a patient. Each Observation has a Concept as Question and a Numeric Value or Value Text or Concept as Answer. Most data that are stored in OpenMRS are Observations and are created when there is Encounter. An Encounter contains from one to dozens of Observations. Each Observation is valid for the time that created and until the next time. For example, a patient's pulse may be increased in a visit. In the next visit, this data will change. An allergy cannot be recorded as Observation.

In Figure 4-4, we can view the contents of a Visit. The Encounter is part of the Visit consisting of a variety of Observations. A Visit consists of many Encounters and an Encounter of many Observations. Each Observation has a Concept as a question and response. Concepts can take the following type depending on the storing data. *Value_boolean* stores boolean values, *value_coded* is used in a drop-down menu and contains all possible answers, *value_drug* stores an ID that represents the Concept which is used for saving a drug, *value_datetime* stores date, *value_numeric* stores a double number, *value_text* stores a text, *value_complex* allows the uploading of a file or an image and *concept*, which stores an ID which represents another concept. Each Visit, Encounter, Observation και Concept has an ID which is unique.

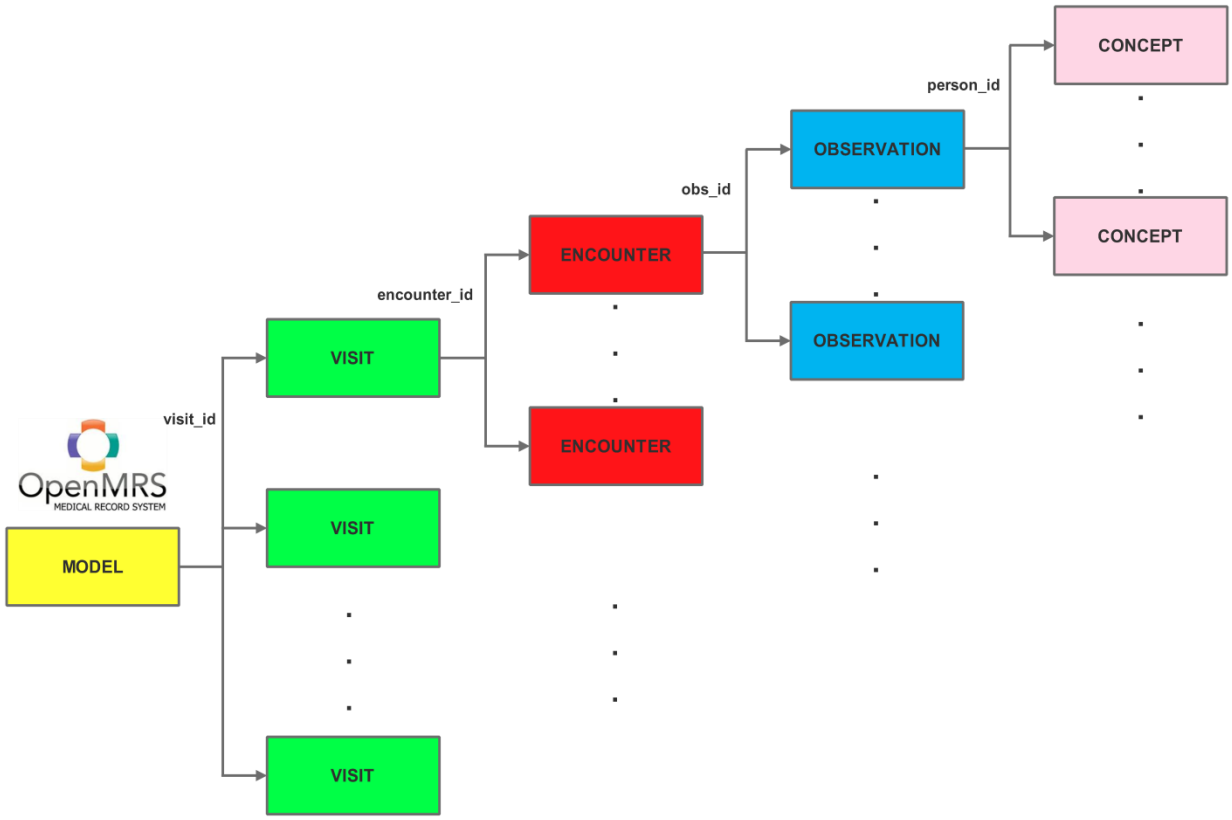


Figure 4-4 Medical Information Structure

Allergy lists: In OpenMRS it is possible to register allergies for each patient. We do not rank allergies in the Observations, because allergies last for a long time.

Appointments: It allows patients to schedule appointments but also check the availability of each doctor.

Forms: Forms are the way in which we can import and display our data. OpenMRS does not have a specific way in which data entry can be done. It can either be with Form Entry (Infopath) or Xforms or HTML Form Entry. Their differences will be considered later.

Users, roles, and privileges: User is anyone who is registered into the system and can log in. A record of a Person at OpenMRS, which may have an account, represents a person. If a

patient wants to have access to the system and see his logs, an account must be created for him to have access using a user name and a password. A Role represents some privileges that users have in the system. A Role can be a doctor who has different privileges from a patient such as data entry in the system. A user may have more than one role. Privilege is the permission of a user to make any changes to the system. An example is a nurse that has the right to import the results from the patient's hematological examinations, but the patient is only allowed to see them. The Administrator has all the privileges in the system.

Scenario Analysis

A patient visits the hospital for the first time. He / She makes a registration at the hospital reception desk using with his / her details, his / her name, address, father's and mother's name and the place of birth. He then visits the doctor he wants. His doctor makes an electrocardiogram and he inserts it into his personal envelope inside the system. Once the electrocardiogram uploads, the processing starts. When the procedure ends, the results are stored in a database and the doctor can view them using the OpenMRS platform from his personal computer or his mobile device and make a diagnosis.

In the above example, the hospital is the Location, which we will input. The hospital reception is a User who creates the folder of the patient and completes it with his personal information. The father and mother's name and the place of birth are some extra information of the patient that the current hospital needs. That information is Metadata and referred to personal information so they are called Person Attributes Types. The doctor who makes the entry of the electrocardiogram is the Provider. Electrocardiogram is an encounter type and contains many Observations that consist of many concepts. Such a concept is the bpm in which we will store the beats per minute of a patient and takes a numerical value.

User Interface

OpenMRS provides also a web application. Anyone who is User has a username and a password that can use to have access into the platform, insert data and view patient medical information. The privileges, that each User has, depend on his role. In Figure 4-5, we can view the login screen. User enters his username and password, chooses the location where the patient did the medical examinations and logs in the system. The username and password have been given by the administrator, In case of a user cannot log in, the field "*Can't log in?*" advices him to contact with the Administrator of the system.

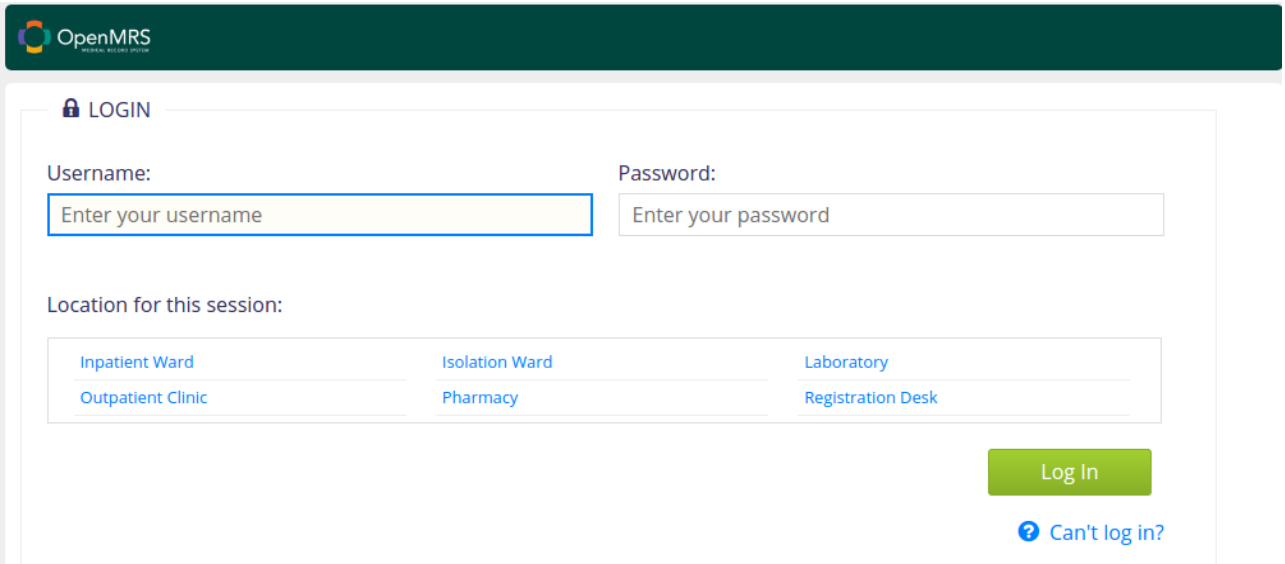


Figure 4-5 Login Screen

Once the user logs in, the system recognizes him and according to the role and the privileges that the user has, the system displays the corresponding fields. In Figure 4-6, the administrator logs in and has access in all over the platform. Subsequently, a nurse and a doctor with different privileges will log into the platform and we view the options that each user has.

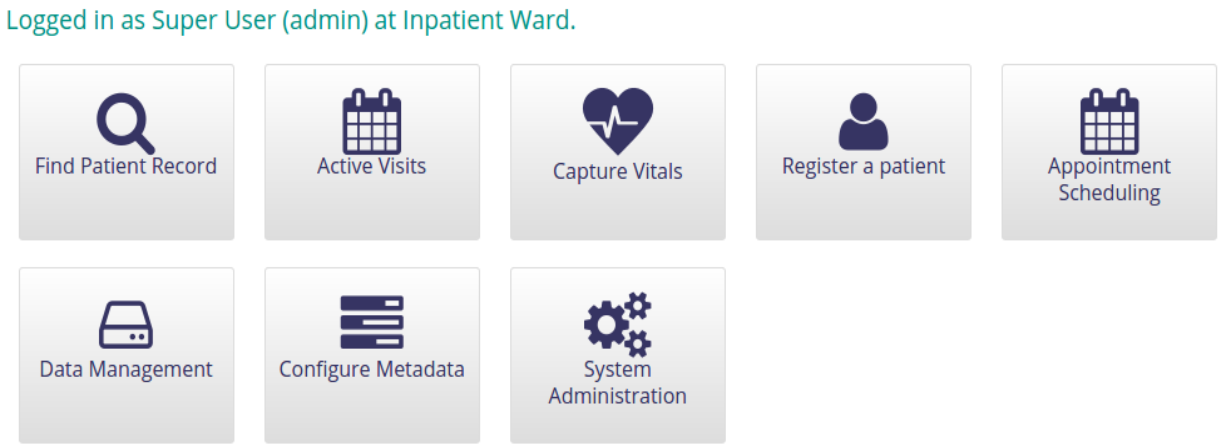


Figure 4-6 Administrator's Login

Figure 4-7 depicts doctor's privileges.

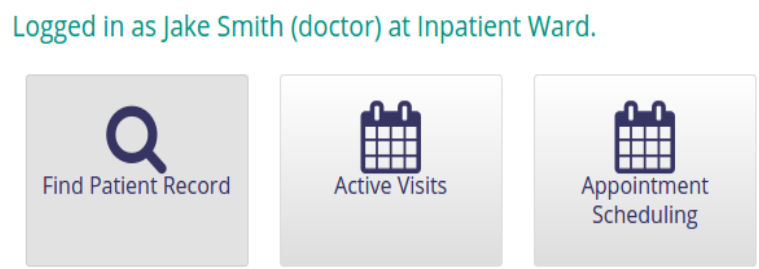


Figure 4-7 Doctor's Login

In Figure 4-8, we can view nurse's privileges.

Figure 4-8 Nurse's Login

The field *Find Patient Record* displays a search bar in which the user types the patient he searches. Then he chooses and displays the patient tab with all the information he has. The patient's tab is analyzed below.

The next field *Active Visits*, shows if there is any active Visit, which gives us the opportunity to register Encounters. In Figure 4-9, we are informed that the particular patient has an active visit. Therefore, by choosing his name we can go to his card.

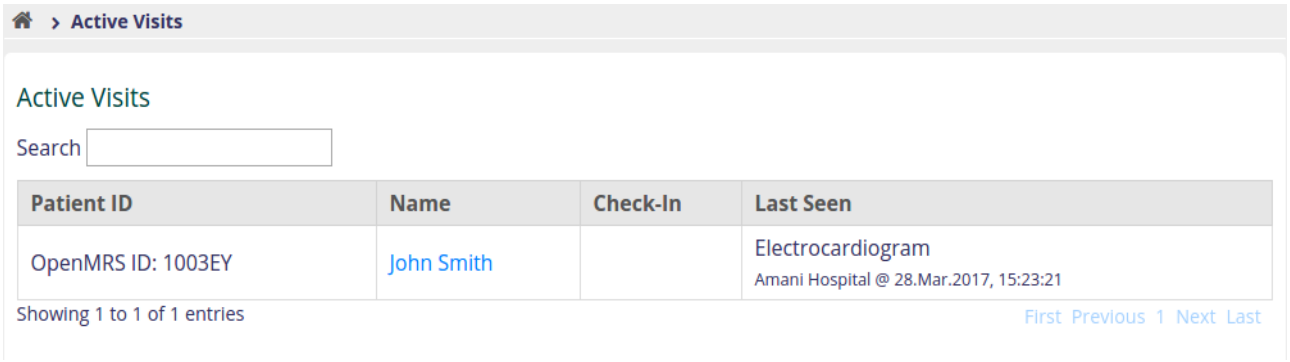


Figure 4-9 Active Visit

The field *Capture Vitals* gives us the opportunity to choose a patient and entry some information about his condition. Vitals are listed below on the patient's tab.

In the field *Register a Patient*, anyone who has privileges, registers a new patient by completing his or her personal information as it is shown in Figure 4-10.

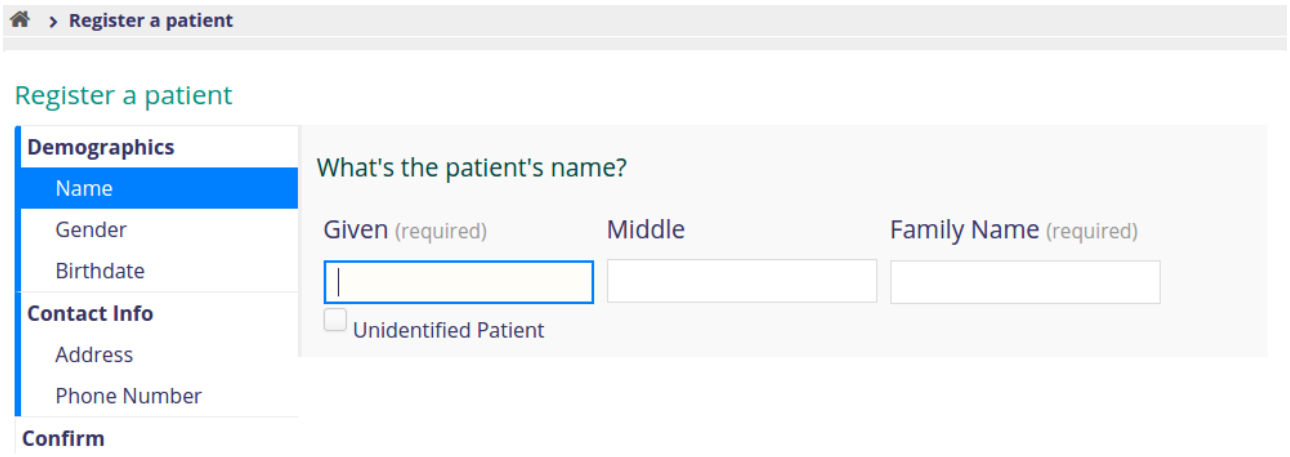


Figure 4-10 Patient Registration

The field *Appointment Scheduling* provides information on the doctors' schedule and on patient appointments. It is a plan with all the appointments in existing clinics.

The *Data Management*, the *Configure Metadata* and the *System Administration* are fields that are appeared to the Administrator and define some settings made on the platform. The *Data Management* provides the possibility of merging some records if considered necessary. In the field *Configure Metadata,* we can add more information. Such information is a new location or a new encounter type. If the admin needs a reprocessing on the roles and the privileges, he can go to that field.

In the field *System Administration,* some General Settings are provided. Administrator can activate some operations but also disable them in case they do not need it now. It can also create a new User and also change the appearance of the platform. In addition to General Settings, there are also the *Advanced Administration Settings.*

Using the *Advanced Administration Settings,* administrator adds new data. These are the Users, Patients, Persons, Visits, Encounters, Providers, Locations, Observations, Scheduler, Programs, Concepts, Forms και Appointments. He can also make some settings for the proper operation of the platform. OpenMRS has modular architecture, a special functionality that allows adding and removing modules without affecting the system. The way they are made is mentioned below. They have full access to the system and behave according to the operation that we have assigned them to do. We use them in case we want to add some additional functionality that in the first place did not exist. One such example is the HTML Form Entry Module, which enables us to collect data through an HTML form. Another example is the Visit Document Module, which we use to upload an electrocardiogram signal.

Some modules are necessary for the platform to work and others that we add to it. The modules that are placed at the initial installation of the platform are as follows. The HTML Form Entry module allows the user to create an HTML form and imports data into the platform. The Xforms is an XML form that allows data entry to be done directly from any JavaScript enabled browser. It is commonly used on mobiles because they are easier to access by these devices. The FormEntry module uses a form that is designed in Infopath. Infopath is a software application for designing, distributing, filling and submitting forms containing data. Users complete it and submit it to OpenMRS. InfoPath forms are avoided because they are not very easy to use and are difficult to troubleshoot. The HTML Widgets module provides reusable HTML form field

widgets that programmers can use into their code. The Reporting module provides a feature-rich and a user-friendly web interface for managing reports within OpenMRS. The Serialization Xstream module provides an implementation of serialization/deserialization strategy using the XStream library. Xstream is a Java library to serialize objects to XML or JSON and back again. The ID Generation module creates IDs that are used in every new data entry. The Metadata Sharing module enables the exchange of data between other platforms. The Rest Web Services module adds a new functionality to OpenMRS, which is the Rest API. REpresentational State Transfer (REST) or RESTful Web Services are one way of providing interoperability between computer systems.

Patient Dashboard

On the patient's dashboard (Figure 4-11), we can see his personal information such as the age, the address, the mobile phone and more.

The *Diagnoses* tab provides information for previously diagnosed diagnostics and until the last visit.

The *Vitals* tab provides information about the patient's most recent condition. Such information is his temperature, his pressure, his oxygen measurement, his breathing rate and also his height and weight. Using the height and weight, we calculate the Body Mass Index (BMI) which measures body fat relative to height and weight. BMI is given by the equation:

$$\text{BMI}=\frac{weight}{\left(\frac{height}{100}\right)^2} \tag{10}$$

In the *Appointments* tab, the appointments of the patients are displayed.

In the *Recent Visits* the last five Visits, that the patient has made, are appeared. By clicking on each date, the user can see information about that visit. Next to every date, we can see if a Visit is active or not. Active Visit is when the patient has not yet completed his visit.

The *Allergies* tab displays the allergies that the patient has on a substance as well as the symptoms that are caused by each allergy.

The *Visit Documents* tab displays if a file has been uploaded and is used for uploading the Electrocardiogram signal file.

The fields in the left blue box show some actions that we can do in either Visits or general actions. In the *Current Visit Actions* we can end a Visit or to complete Encounters into a Visit. In *General Actions*, we can add or edit some Visits or Appointments.

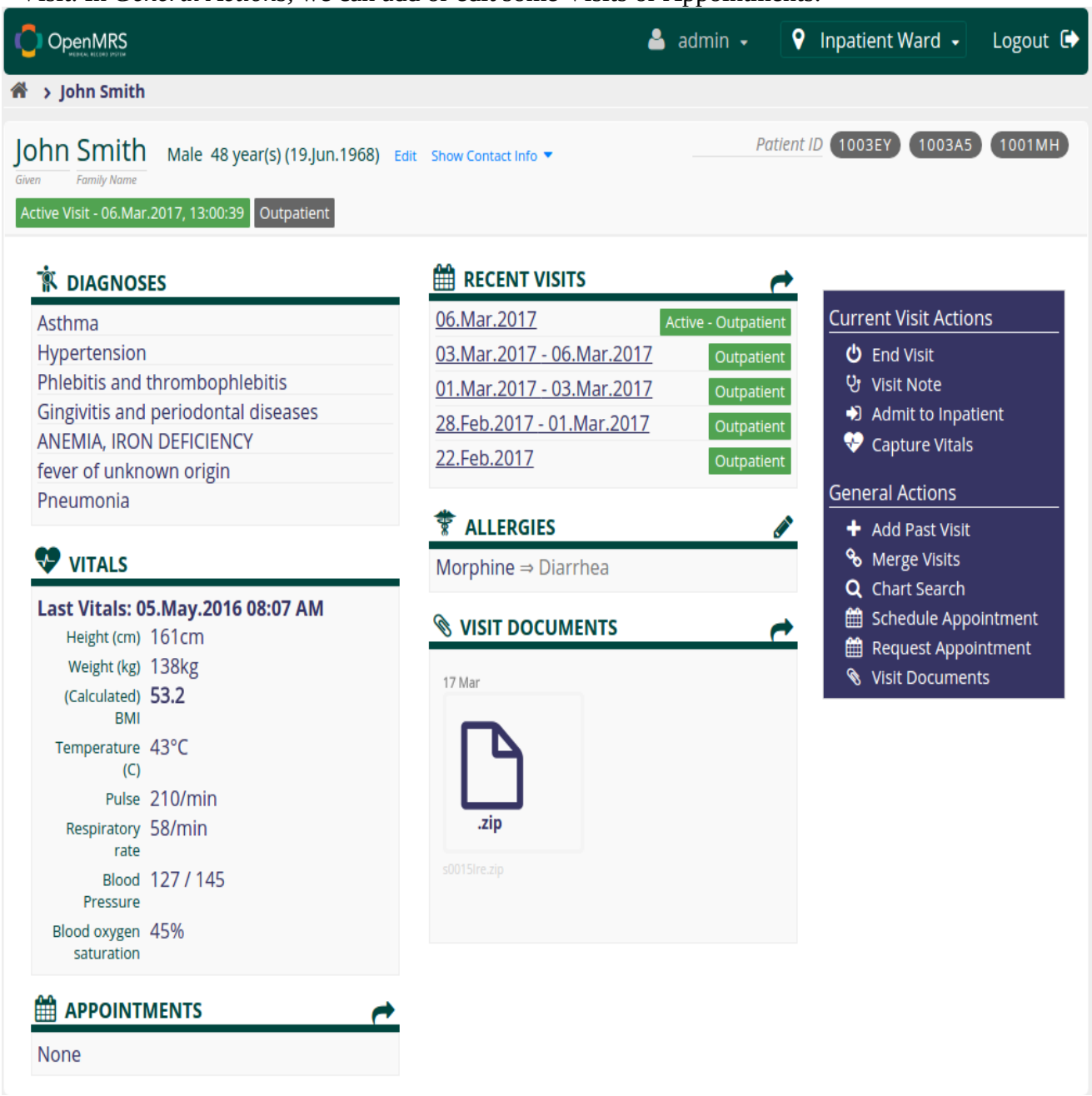


Figure 4-11 Patient's Dashboard

The patient's Visits are displayed in Figure 4-12. We can select a date from the left and view the Encounters that are recorded. On the Visits tab, we can view the start date of a visit as well as the expiration date if it is over. In addition, a diagnosis is displayed for that date. One

visit can contain more than one Encounter. In the below example, we have three encounters. The Visit Note, which contains some diagnosis for the patient, the Vitals include information about the patient's condition and the Electrocardiogram that contains the patient's electrocardiogram as well as some additional information related to an electrocardiogram. Encounters can be processed or deleted by clicking the buttons that have a pencil and an *x* respectively.

Clicking on the *Show Details* field displays a form that contains the entries that have been entered for the corresponding Encounter. Some Encounters may not be edited or deleted. This depends on the creator of the form. The Electrocardiogram is not editable because the data is generated by an algorithm and is not manually placed by the user. A user can edit the Visit Note because another Provider may want to add an additional diagnosis that may not have been added.

Encounters, that someone can add, appear in the blue frames. In our example, there is the *End Visit* field that is ending the visit, the *Visit Note* field, the *Admit to Inpatient* filed and the *Capture Vitals* field. By tapping on one of these, except for the End Visit, a user-friendly form appears which stores data and appears as Encounter in the list that appears below. Apart from the Visits, one can also see the Appointments that the patient has.

The blue *Actions* panel displays the General Actions that are mentioned above.

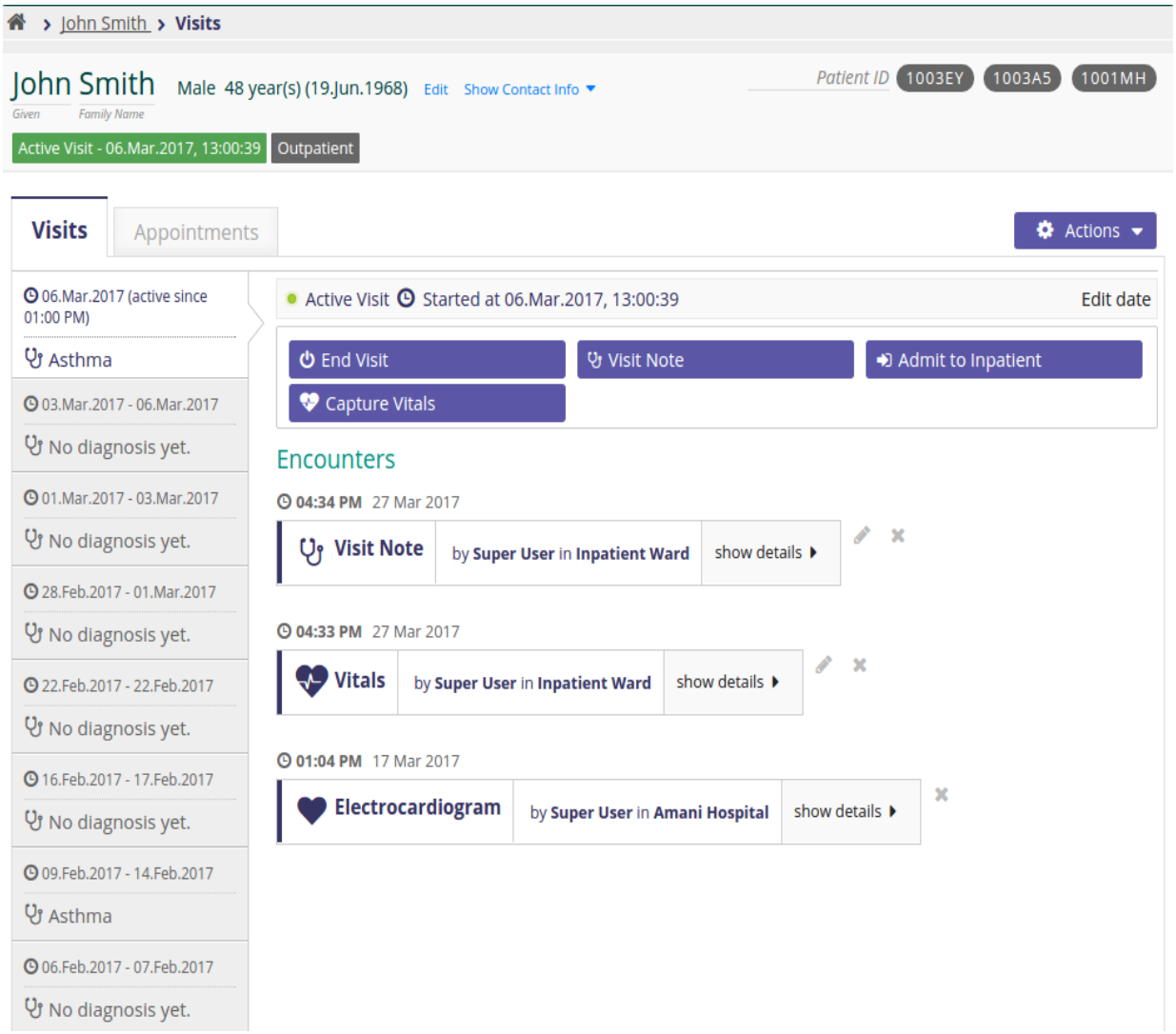


Figure 4-12 Patient's Visit

Access Control

Administrator is responsible for creating roles and privileges (figure 4-13). This is very helpful because the system is more easy to be managed. Roles are used to group privileges. For example, a role is a nurse. Nurse has access to view patients and complete patient's dashboard with personal data but she cannot change settings on platform. An assistant can also registers patients. Roles depend on clinic's operation. When we create a user, it is necessary to add him a role. If we create users who are doctors, they will have all the same privileges.

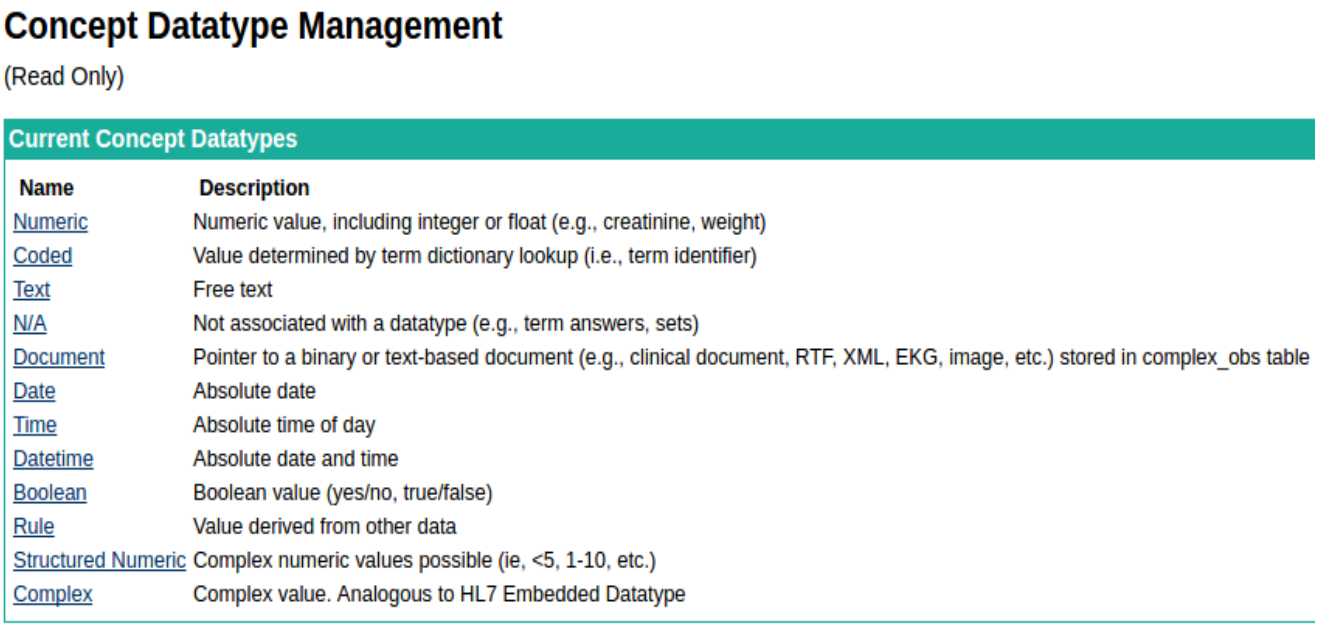
Organizational: Doctor	Doctor	Application: Sees Appointment Schedule , Application: Uses Patient Summary ...
Organizational: Hospital Administrator	Hospital Administrator	Application: Requests Appointments , Application: Configures Appointment Scheduling
Organizational: Nurse	Nurse	Application: Enters Vitals , Application: Uses Capture Vitals App ...
Organizational: Registration Clerk	Registration Clerk	Application: Schedules Appointments , Application: Registers Patients ...
Organizational: System Administrator	System Administrator	Application: Configures Metadata , Application: Configures Forms ...
Privilege Level: Full	A role that has all API privileges	
Provider	All users with the 'Provider' role will appear as options in the default Infopath	
System Developer	Developers of the OpenMRS .. have additional access to change fundamental structure of the database model.	[Has all roles and privileges]

Figure 4-13 Roles and Privileges

Management of Concepts

The most important issue in OpenMRS is to create well-defined concepts. Concepts are basic elements in the system. Before creating a concept, we need to define the Concept Classes. The Classes are details of how a concept will be represented when the information is stored [46].

Another important fact is the type of the data that we will entry. When we create a concept, we have to determine the type of the data that we will store as we can see from Figure 4-14 [47].



Concept Datatype Management

(Read Only)

Current Concept Datatypes

Name	Description
Numeric	Numeric value, including integer or float (e.g., creatinine, weight)
Coded	Value determined by term dictionary lookup (i.e., term identifier)
Text	Free text
N/A	Not associated with a datatype (e.g., term answers, sets)
Document	Pointer to a binary or text-based document (e.g., clinical document, RTF, XML, EKG, image, etc.) stored in complex_obs table
Date	Absolute date
Time	Absolute time of day
Datetime	Absolute date and time
Boolean	Boolean value (yes/no, true/false)
Rule	Value derived from other data
Structured Numeric	Complex numeric values possible (ie, <5, 1-10, etc.)
Complex	Complex value. Analogous to HL7 Embedded Datatype

Figure 4-14 Type of Data

Concepts are stored in the Concept Dictionary. To create a Concept we click on the *Add new Concept* inside the *View Concept Dictionary*. An example of how we create Concepts depicts in Figure 4-15. Some fields are necessary to be filled such as the name, the datatype and the version. These fields have been filled below. Concept Mappings are used when some concepts have the same meaning and are related each other also, they are not necessary to be filled. After the creation of the concept, an ID is created. Using those IDs of all the concepts, we will take and display the values of each concept.

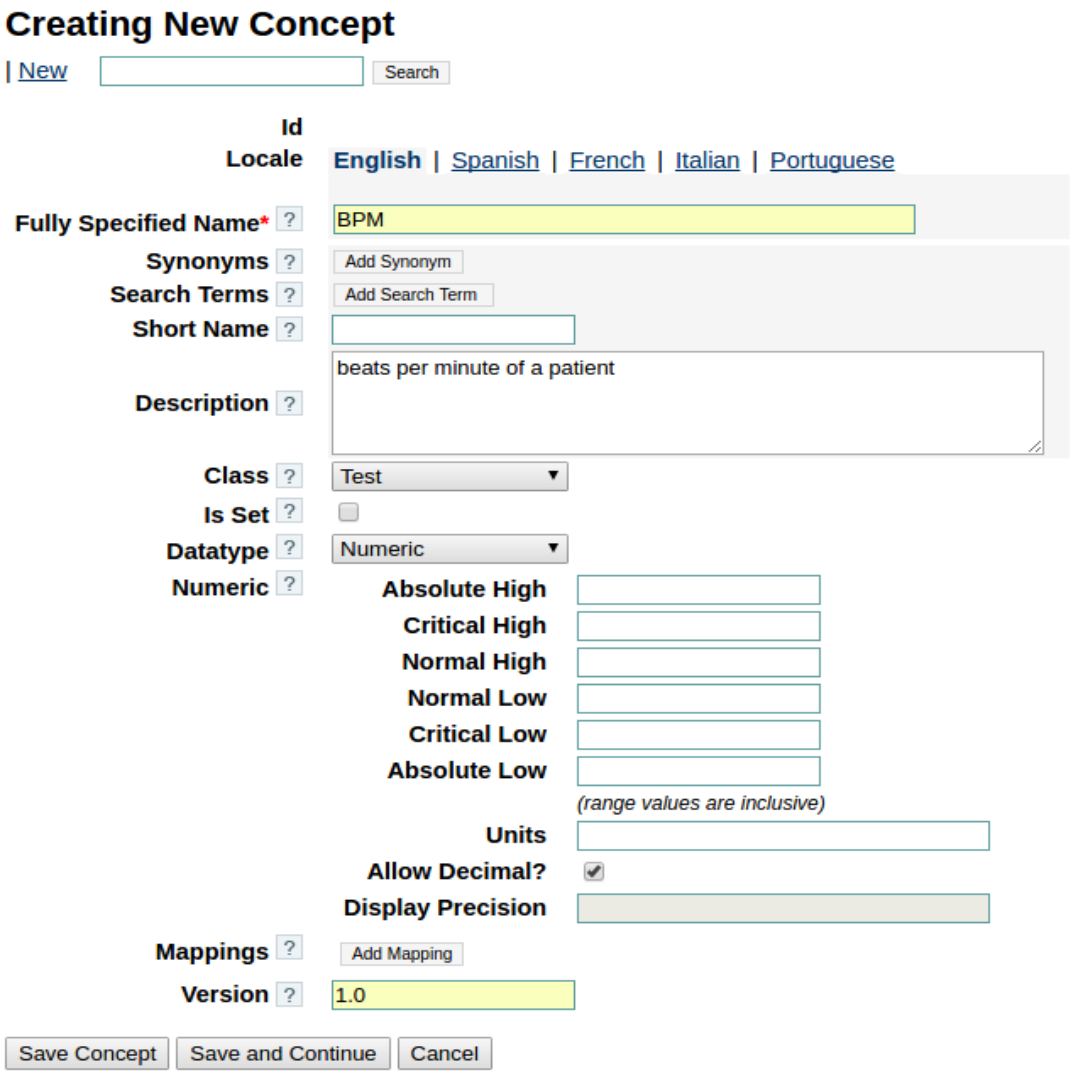


Figure 4-15 Creation of a Concept

Chapter 5 - Technical Implementation

System Design

In the current implementation, OpenMRS is the main component of our system. In conventional research approaches, OpenMRS is used only for storing medical data into the platform in image format. The main steps of previous research are depicted in Figure 5-1. Medical data such as electrocardiogram are exported via health monitoring devices (electrocardiograph). This information is uploaded by a doctor or medical staff to the OpenMRS platform as images by using a computer or a mobile device. This platform collects medical data for all patients and doctors can access this information using also a computer or a smartphone in order to form the diagnosis. The diagnosis relies on empirical findings by observing the visualized results that OpenMRS provides. OpenMRS supports all types of files but it provides visualization capability only in cases of image uploaded data. In case that the diagnosis indicates a health misoperation, doctor should provide the appropriate treatment to the patient.

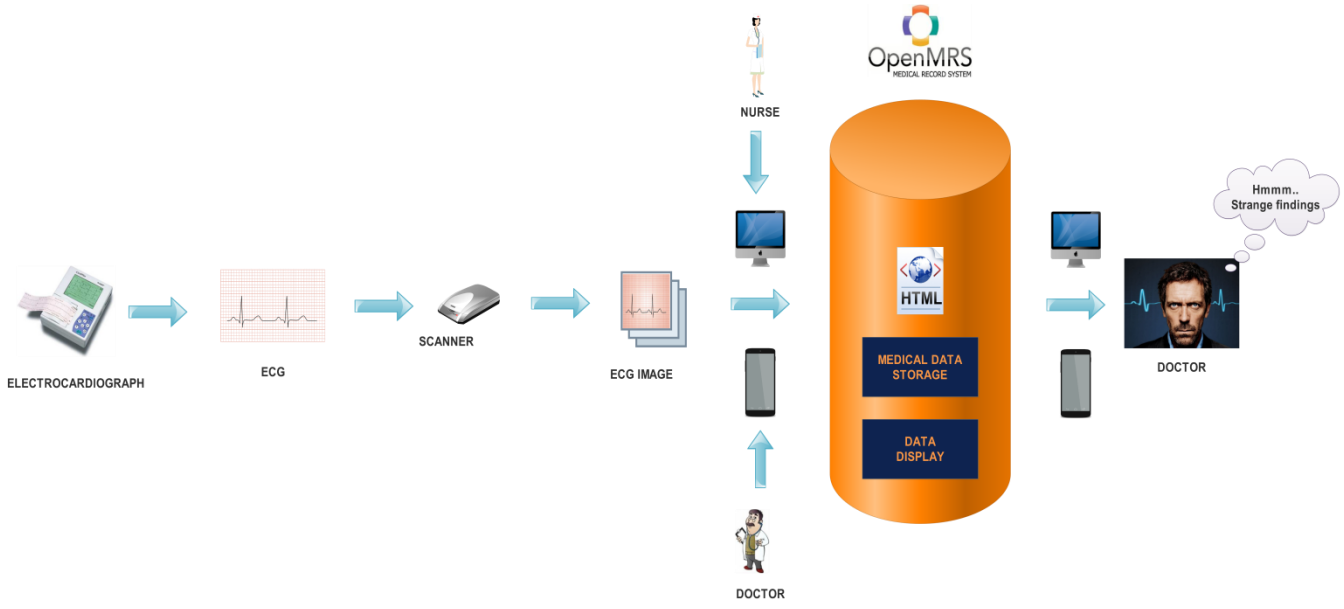


Figure 5-1 Conventional OpenMRS Functionality

This approach allows doctors to study medical exams remotely using hardware but the diagnosis is based on observing the results of the tests. In this case, constant connection to the OpenMRS platform is required for the doctors. Also, doctors should periodically check the recent upload files by the patients and this is not practical in regular basis due to the workload that is assigned to the doctor.

In the current implemented system, we try to extend OpenMRS functionality by expanding its duties. Apart from the previously described operation, OpenMRS analyses the input signals (ECG waveforms) that are uploaded by the medical staff. The analysis process is based on algorithms that use ECG background. The outcomes from this process are used as indicators by our system in order to create the diagnosis for the patient. In case that our system detects health anomalies, it uses appropriate notation to inform the doctor for instant treatment. Additionally, our system allows direct connectivity with the health monitoring devices without requiring any medical staff. This approach tries to embed the workload for the diagnosis process form the doctor to OpenMRS and generally decrease the degree of human intervention in the whole operation. The overview of the proposed schema is contained in Figure 5-2. Also, the system allows real time measurements transmission from wearable health monitoring devices using TCP/IP protocol. Finally, the system receives measurements in binary form for the analysis, diagnosis and visualization process on contrast with previous implementations that display only uploaded data images.

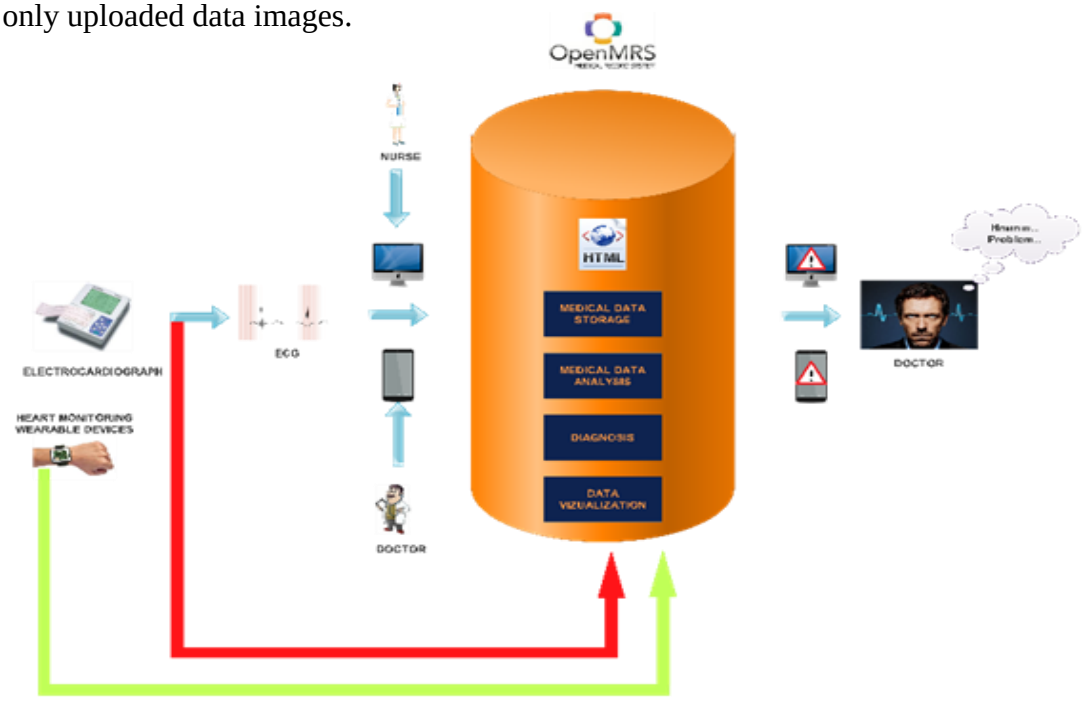


Figure 5-2 Proposed Schema

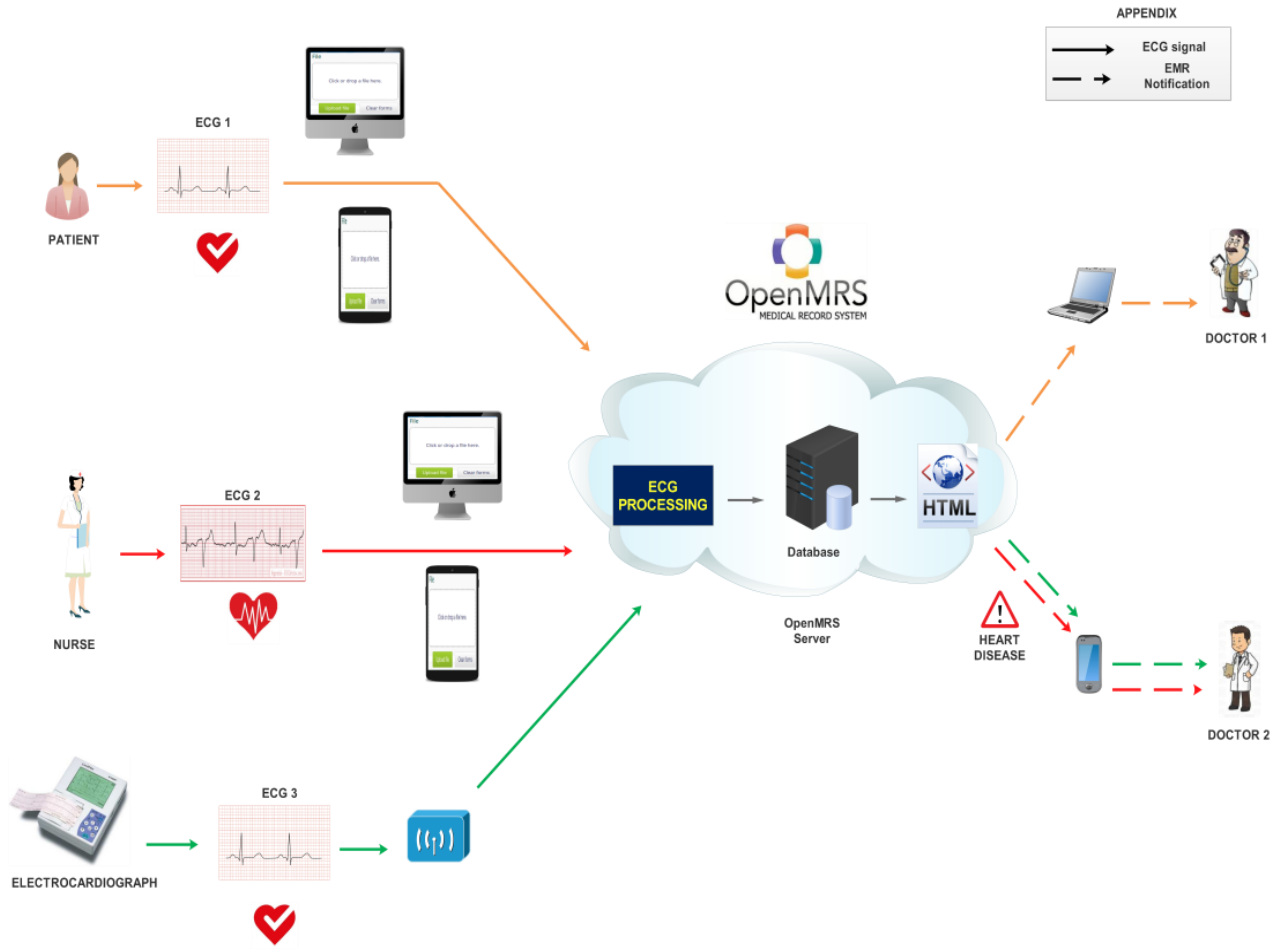


Figure 5-3 System Architecture

Figure 5-3 depicts an abstract view of our architecture. A patient, who has access in the OpenMRS platform, uploads his Electrocardiogram file or other health in-formation on the platform. Apart from the patient, medical staff such as nurses can upload ECG files using a computer or a mobile device. Finally, ECG file can be transmitted directly to the platform by the electrocardiograph using Internet connectivity. The platform is installed into a server and there is access via mobile phones through an application and computers via a web browser. In server side, WFDB applications and algorithms are implemented to analyze the ECG file and export the necessary results that a doctor must check. When processing is accomplished, the doctor or the nurse can view the results whenever is needed.

Initially, patient needs to have saved the ECG file into his personal device. The ECG file is 12 - lead WFDB signal file and it is binary. It is accompanied by a header file, which contains information about the signal and is necessary to exist for the functionality of the WFDB applications. For better ease, user can upload a compressed folder that includes both files. The compressed folder is checked for non-existence of files. If one file does not exist or a file with different type is uploaded, the execution of the algorithms does not start. User can also upload different files, such as images, that are stored in the database and can be viewed.

A module that provides the ability to upload files has been installed into the platform. User navigates to the specific field and selects the compressed folder for uploading. Once the folder is loaded, WFDB applications and algorithms start the execution. The results are stored in the database and can be displayed by other users.

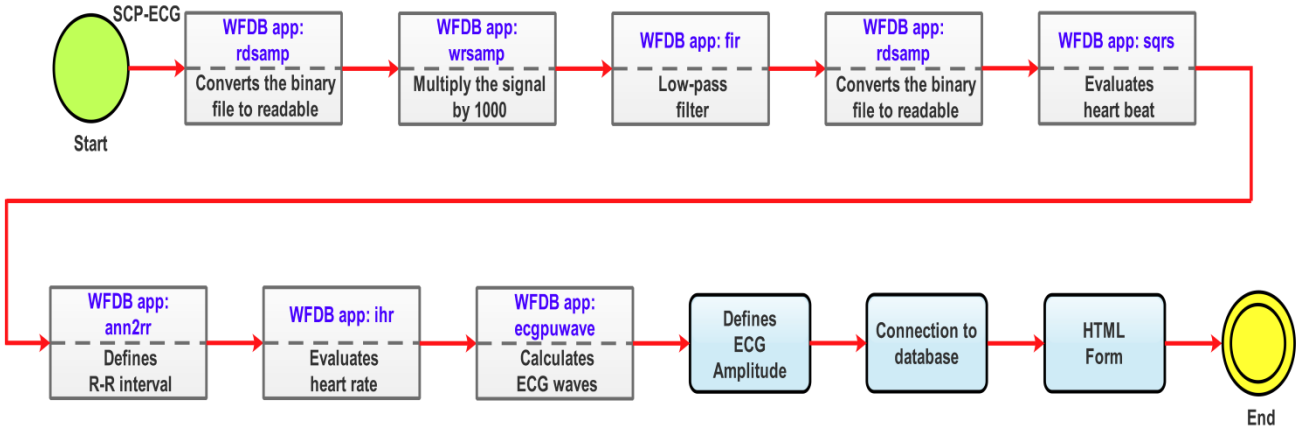


Figure 5-4 System Architecture

As we can see from the Figure 5-4, there is a use of different technologies. Through an OpenMRS module that is called Visit Document Module, a user can upload an ECG signal file in SCP-ECG format that is stored into the database. Using Java, we check if the file is a WFDB signal file. Then, algorithms that are written in Python and WFDB applications are executed to process the signal file, using its destination and name.

The function *os.system(command),* in Python, is used for the execution of the WFDB applications. The first operation is to convert the binary file into text file by executing the command *os.system('rdsamp -r '+file_name+' >samples.txt')*. The *file_name* is the name of the

ECG signal file and the text file *samples.txt* the file that will be created and contain the signal in decimal numbers. The next operation is to filter the signal using a low-pass "boxcar" fir filter. After filtering, the *ann2rr* application is used to determine the R-R intervals. By executing the *ihr* application, the instantaneous heart rate is calculated. Finally, *ecgpuwave* locates the waveforms of the ECG signal.

When the execution of the WFDB applications is terminated, the execution of the functions in python is started in order to calculate the ECG values. Those values are described in the next section. When the operation is completed, the ECG values are stored into the database using SQL Queries. Each value is stored into specific table in the database that all the observations are stored and uses a different *id*. Each value is a different concept and is distinguished from the rest values by the corresponding id.

The OpenMRS platform provides the opportunity to create HTML Forms. Those forms are written on HTML and also use CSS, JavaScript and JQuery [48]. To display the values from the database we use the id that we had determined when storing the data into database. Using the *id* attributes and JavaScript on an *obs* tag, we take the content from a specific value. For example, the command *<obs id ="bpm" conceptId="163142"/>* is used to take the value that is stored in the concept with the id 163142. That concept contains the beats per minute of each patient. When we create a concept, a different id is created for each concept. In addition, when a concept is stored into the database, other characteristics are also saved like the *visit_id*, which referred to the visit that the concept is belonging to and the *patient_id*, which is referred to the patient.

The values are retrieved from the database and displayed as an HTML form using JavaScript and JQueries [49]. For example, the command *<obs id="col_i" conceptId="163168"/>*, gives a name to the corresponding concept, which is the *col_i*. The name will be used to recognize which observation we will display into the HTML page. When we get the value, using the JavaScript command

jQuery(function() { var val=getValue("col_i.value") });

we can also process it and then display it into the HTML form.

ECG Findings

According to cardiologists, in order to evaluate an electrocardiogram, they have to extract empirically the required metrics. Those metrics are the ECG findings such as heart rate and amplitude of the ECG waves and are described below.

The electrocardiogram is imprinted on a graph paper, which makes it easier to measure the height of each lead and the duration of the waves. The heights are the distances between amplitudes of the peaks, positive or negative, and a baseline. The graph paper is divided into dark lines that have 5 mm distance between them and lighter lines that have 1 mm distance. The horizontal axis displays the time that is 1 mm = 0.04 sec = 40 msec. The vertical axis displays the amplitude of the waves that is 1 mm = 100 μV = 0.1 mV [50].

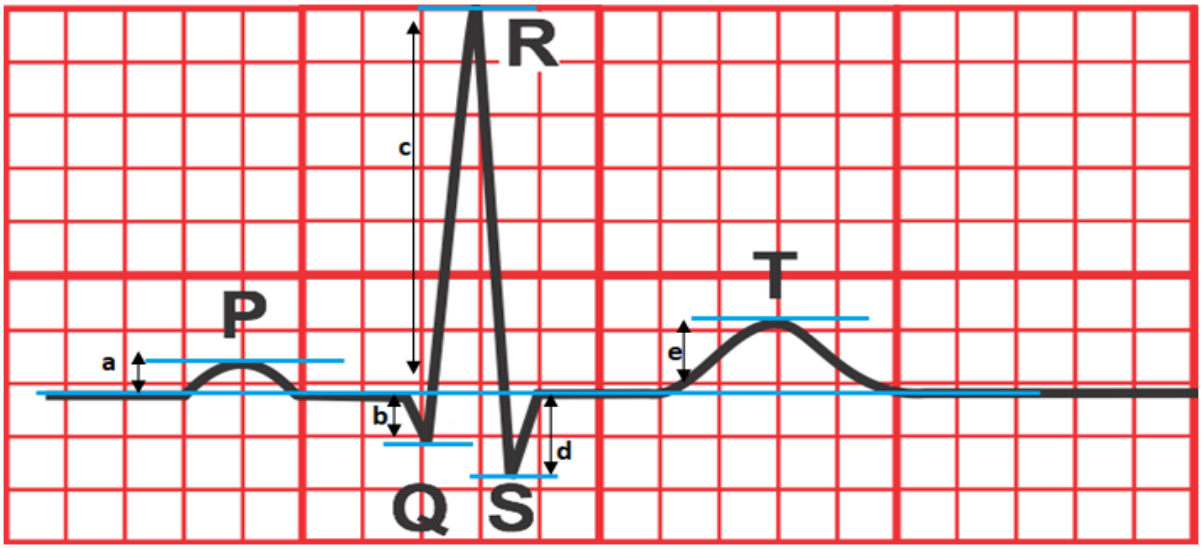


Figure 5-5 Amplitudes in a Heartbeat

From the Figure 5-5, the variables are defined as a = P-wave amplitude, b = S-wave amplitude, c = R-wave amplitude, d = S-wave amplitude and e = T-wave amplitude.

To measure a beat, it must be complete with the P-wave, QRS complex and T-wave. If a wave is missing, we measure the next beat.

Ideally a heartbeat contains one P-wave, one QRS complex and one T-wave. The excitation wave may be change by some heart disease and that affects the QRS complex. A disease may cause the non-appearance of the Q-wave or the amplitude of the Q-wave is greater than S-wave. Moreover, an abnormality appears when the R-wave is missing. In addition, the S-wave may be not displayed.

From the electrocardiogram, we can estimate some findings that are necessary for the human health. Firstly, we calculate the heart rate using R-R interval. The formula to calculate the heart rate is

$$bpm = \frac{60}{R-R\ interval} \tag{11}$$

The number 60 represents the seconds that a minute has. The limits of the heart rate are 60 beats per minute (bpm) to 100 bpm. If the heart rate is below 60 bpm there is bradycardia. When the heart rate is more than 100, there is tachycardia [51].

The intervals in a heartbeat are also very important to be calculated. P-wave's duration has to be less than 120 milliseconds. QRS complex in a normal heartbeat lasts for about 70 msec to 110 msec. T-wave extends about 300 msec after the QRS complex and lasts about 160 msec. P-R interval lasts about 120 msec to 200 msec and Q-T interval lasts from 350 msec to 440 msec [52]. Using Q-T interval we calculate the QTc, which is a corrected Q-T. QTc depends on the R-R interval and we calculate it using the Bazett's formula [53]

$$QTc = \frac{QT}{\sqrt{R-R\ interval}} \tag{12}$$

The ECG intervals are summarized in the Table 5-1.

P wave duration	< 120 msec
QRS complex duration	70 - 110 msec
T wave duration	< 160 msec
P-R interval	120 - 200 msec
QT interval	350 - 440 msec
QTc interval	350 - 440 msec
R-R interval	600-1000 msec

Table 5-1 ECG Intervals

Some extra values that need to be estimated are the amplitudes of some waves in the ECG signal. The amplitudes are calculated from the top of each wave to a baseline and give important information about health condition of patients. For example, the amplitude of P-wave should be up to 2.5 mm in Limp Leads (I, II, III, AVR, AVL and AVF) and up to 1.5 mm in Precordial Leads (V1, V2, V3, V4, V5 and V6) [54] and it also has to be preceded from the QRS complex. P-wave has to be positive in all leads except in lead AVR, which is negative. The amplitude of the QRS complex is the largest of the other waves and contains the Q-wave, the R-wave and the S-wave. The R-wave is positive and the Q-wave and S-wave are negative on most leads except in lead AVR that all the waves have to be inverted. T-wave is after the QRS complex and is also positive in all leads except in lead AVR. Table 5-2 summarizes the values of the ECG waves that are needed [55] [56] [57].

P height II	≤ 2.5 mm
R amplitude I	15 mm $< R \leq 20$ mm
R amplitude II	≤ 20 mm
R amplitude III	≤ 20 mm
R amplitude AVF	≤ 20 mm
R amplitude AVL	≤ 13 mm
R amplitude AVR	≤ 3 mm
R amplitude V1	≤ 26 mm
R amplitude V2	≤ 26 mm
R amplitude V5	≤ 27 mm
R amplitude V6	≤ 27 mm
S amplitude I	≤ 8 mm
S amplitude V1	≤ 30 mm
S amplitude V2	≤ 30 mm
S amplitude V5	≤ 17 mm
S amplitude V6	≤ 4 mm

Table 5-2 Amplitudes of ECG Waves

Using the amplitudes of the QRS waves, we calculate the R/S ratio. With the R amplitudes in leads V1, V2 and V6 and the S amplitudes in the V1, V2 and V6, we estimate the ratios (Table 5-3) [28] [58].

V1 Ratio: R-V1/S-V1	≥ 1
V2 Ratio: R-V2/S-V2	≥ 1.5
V6 Ratio: R-V6/S-V6	≤ 3

Table 5-3 R/S Ratio

Electrical Axis

The heart has axes that describe the direction of the electrical wave (Figure 5-6). The most important axis is the QRS Axis that shows the direction of electrical current propagation through the myocardium. The normal QRS Axis has ranges from -30^0 to $+90^0$.

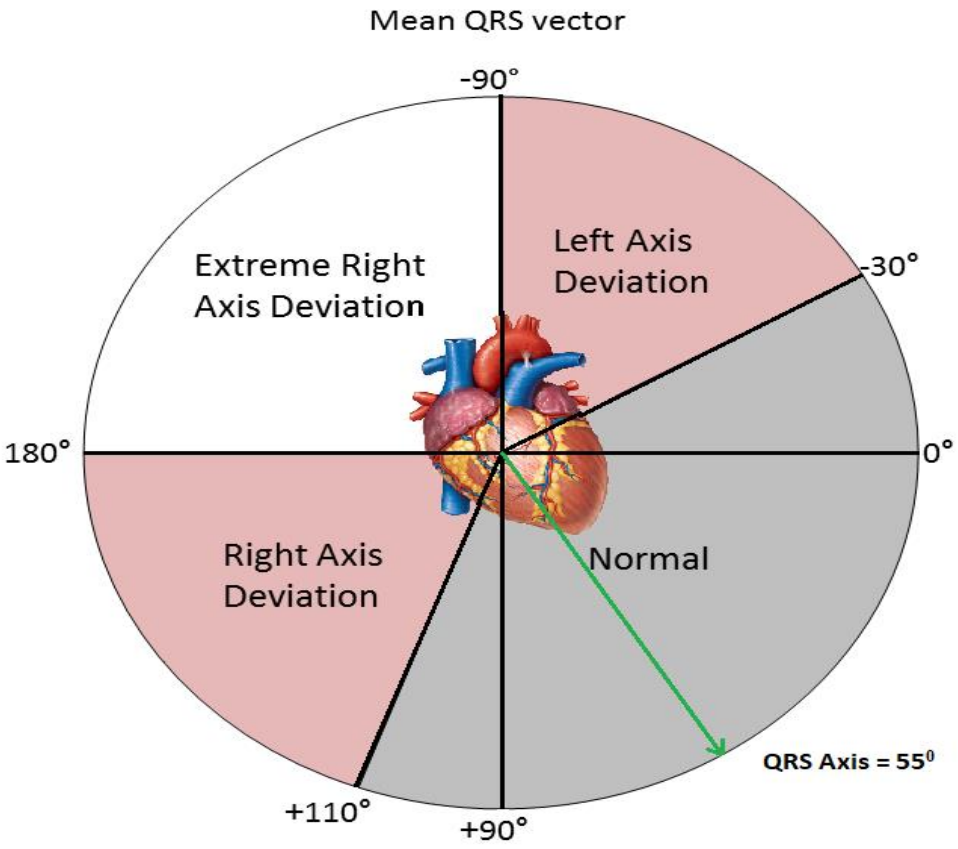


Figure 5-6 QRS Axis

There is also the P Axis, which represents the P wave and the T Axis that represents the T wave. The P Axis has ranges from 0^0 to 75^0. Using QRS Axis and T Axis, we estimate the QRS-Tangle. QRS Tangle equals to the difference of QRS axis to the T axis (Figure 5-7). QRS Tangle has to be from 20^0 to 116^0 in women and from 30^0 to 130^0 in men [59].

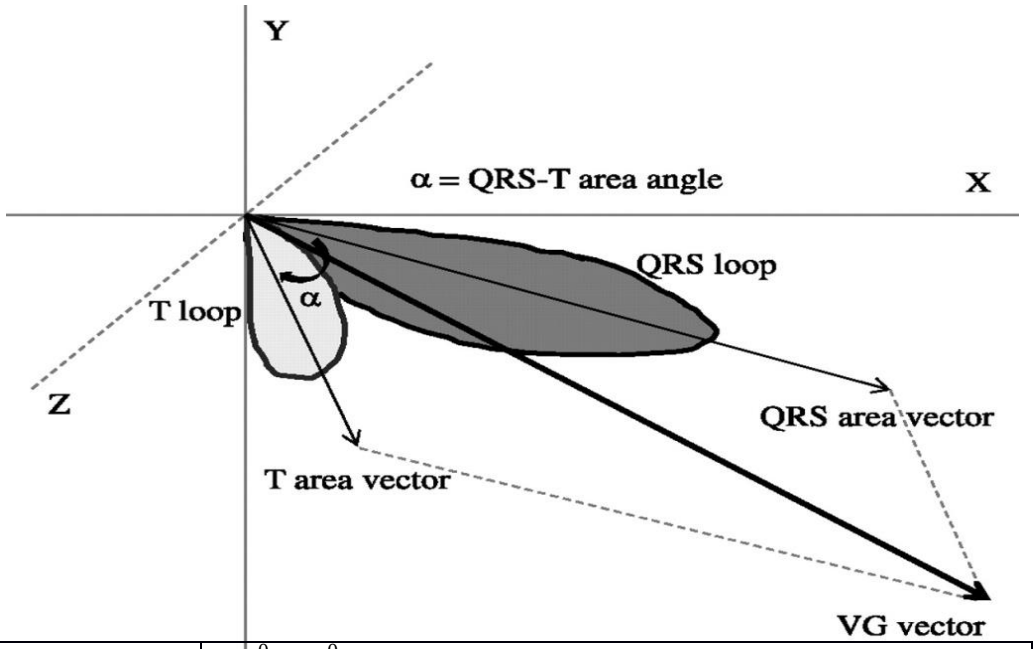


Figure 5-7 QRS Tangle

QRS Axis	20^0 - 90^0
P Axis	
QRS Tangle	20^0 - 116^0 in women and 30^0 - 130^0 in men

Uploading ECG file

Implementation consists of three steps. The first step is the uploading of the ECG file. The second one is the elaboration of the ECG signal and the connection to database for storing the results and the third one is the display of the results to user.

For uploading the ECG file, a module that is called Visit Documents Module is used. A user is navigated into the platform and uploads a compressed folder to the corresponding field (Figure 5-8). The compressed folder contains the ECG binary signal file and the header file, which contains information about the signal. Without the header file, the binary file is not readable and editable.

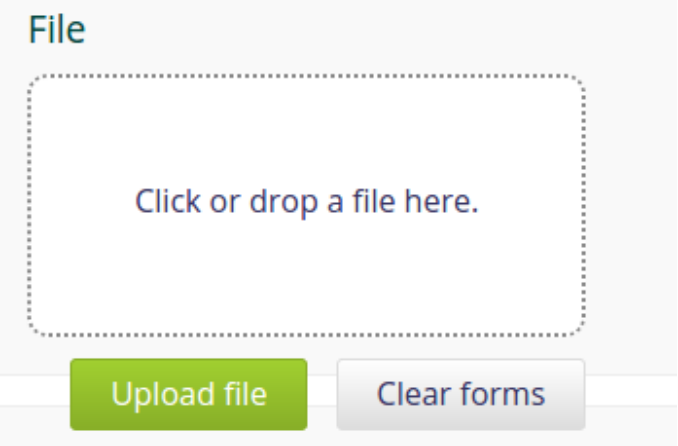


Figure 5-8 Upload dialog box

The ECG upload process in OpenMRS platform can be provided via a mobile device as Figure 5-9 shows.

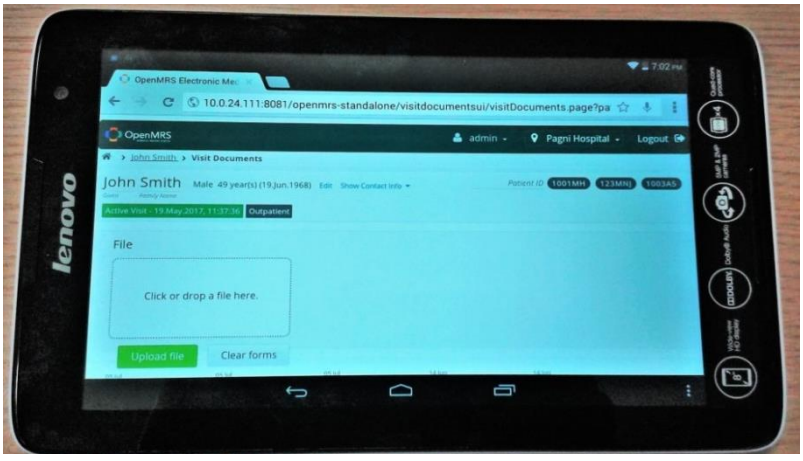


Figure 5-9 ECG upload process via mobile device

The compressed folder is uploaded through the Spring Framework and we have access to that folder using HTTP Requests and Responses. With OpenMRS API and the use of some classes, we take the name of the compressed folder, as it is stored into database.

The interface ObsService is involved with the storing and getting Observations in/from the database. Using the class Obs and the method *getObsByUuid (String uuid);* we take the Observation by its UUID. As we have already mentioned, an observation is part of the record of an Encounter. Observations are grouped together into one Encounter. The method *getComplexObs (Integer obsId, String view);* gets a complex Observation with the complex data of the Observation. ComplexObs is a permanent Object that extends Obs but it is not persisted in the database. It has a data Object and a title. Using the method *getComplexData();* we take the data Object. Using the constructor *ComplexData(String title, Object data);* and the method *getTitle();* we get the title for this ComplexData which is the name of the compressed file that is stored in the database (Figure 5-10).

```
view = VisitDocumentsConstants.DOC_VIEW_ORIGINAL;

// Getting the Core/Platform complex data object
Obs obs = context.getObsService().getObsByUuid(obsUuid);
Obs complexObs = context.getObsService().getComplexObs(obs.getObsId(), view);
ComplexData complexData = complexObs.getComplexData();

// Switching to our complex data object
ValueComplex valueComplex = new ValueComplex(obs.getValueComplex());
DocumentComplexData docComplexData = context.getComplexDataHelper().build(valueComplex.getInstructions(), complexData);
String name = docComplexData.getTitle();
```

Figure 5-10 Get name using OpenmMRS API

When we get the filename, we make some successive checks in Java to check if the compressed folder contains the binary file and the header file. All the files are checked for their suffixes one by one. The binary file has the suffix .dat and the header file has the suffix .hea. If a file is missing, the operation is terminated. The counter *count* is used to run through the files that are in the compressed folder and keeps the position of each file for the next check (Figure 5-11).

```
if(entry.getName().equals(filenames.get(count)+".dat")){
    datIndex = count;
}

if(entry.getName().equals(filenames.get(count)+".hea")){
    heaIndex = count;
}
count++;
```

Figure 5-11 Check existence of ECG files

An extra check that is done is the name. The WFDB libraries need the binary file and the header file to have the same name but different suffix. Using the variables *datIndex* and *heaIndex* we keep the position of the files. Therefore, we can check if the filenames are the same (Figure 5-12).

```
if(datIndex>=0 && heaIndex>=0){
    if(filenames.get(datIndex).equals(filenames.get(heaIndex))){
        isValid = true;
        break;
    }
}
```

Figure 5-12 Check filenames

A Boolean variable is used in case of the checks are correct. If the value of the variable is true, the files are extracted from the compressed folder using the method *doUnZip (String ZipName)* where ZipName is the name of the compressed folder into the database (Figure 5-13). The extract of the files is done into the server into the same folder that the python files are stored.

```
private void doUnZip(String ZipName){
    List<String> fileList;
    String INPUT_ZIP_FILE = "//home//katerina//Desktop//openmrs-standalone-1.11.6//appdata//complex_obs//"+ZipName;
    String OUTPUT_FOLDER = "//home//katerina//Desktop//openmrs-standalone-1.11.6";
    UnZip unZip = new UnZip();
    unZip.unZipIt(INPUT_ZIP_FILE,OUTPUT_FOLDER,ZipName);
}
```

Figure 5-13 Extract ECG files

After extracting the files, the last operation is to start the python execution through Java (Figure 5-14). The arguments that the method *call_python (String filename, String zipname)* needs, are the name of the ECG files and the name of the compressed folder in the database. Then the processing of the ECG signal starts using algorithms and functions that are written on python.

```
public void call_python(String filename,String zipname) {

    try {
    String pythonScriptPath = "//home//katerina//Desktop//openmrs-standalone-1.11.6//final.py";
    String[] cmd = new String[4];
    cmd[0] = "python";
    cmd[1] = pythonScriptPath;
    cmd[2] = filename;
    cmd[3] = zipname;
    Runtime rt = Runtime.getRuntime();
    Process pr = rt.exec(cmd);
```

Figure 5-14 Call Python Script from Java

ECG Signal Processing

WFDB Applications allow ECG processing for evaluating useful findings, which we have already described. The processing of the ECG file starts after upload phase is completed. Figure 5-15 depicts the steps of ECG processing. The ECG signal passes through filtering phase and then ECG analyze stage takes place. The results of this analysis are stored into the database. Lastly, the doctor can view the ECG signal from the OpenMRS platform and make a diagnosis.

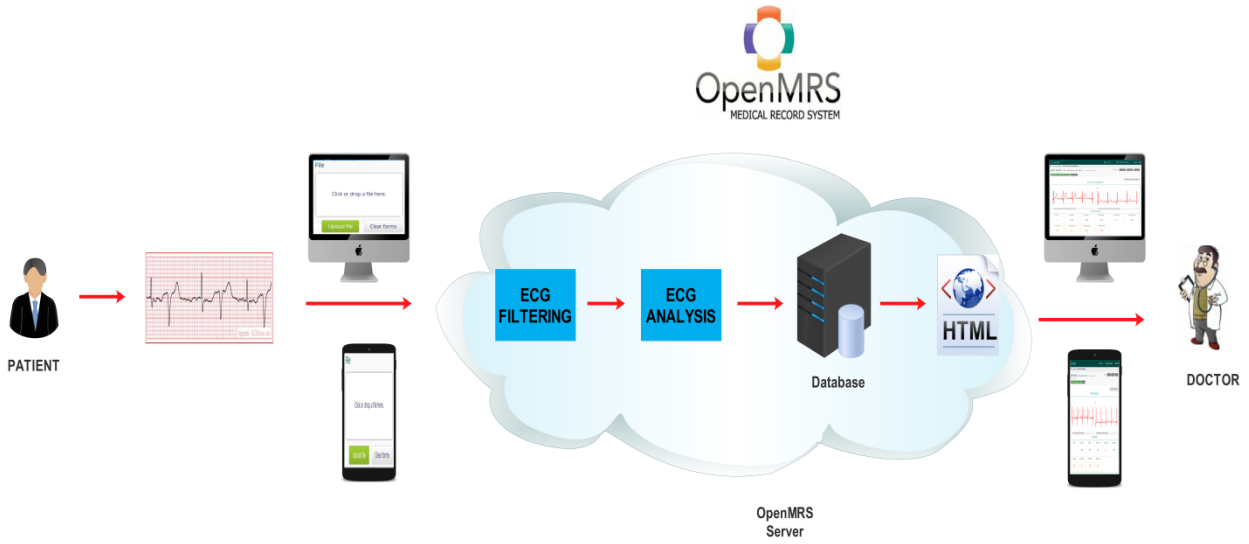


Figure 5-15 ECG Signal Processing

Using WFDB applications

Before the processing of the ECG signal starts, we need to know the two arguments. Those arguments are the name of the ECG files, which must be the same for the binary file and the header file, and the name of the compressed folder as it is stored into the database. The name of the ECG files, from now we call it *file_name* in the WFDB applications, helps us to process the signal with WFDB applications. As we mentioned, the WFDB applications need a binary file that contains the signal and a header file that contains information about the signal. Both files are

necessary for the WFDB applications execution. The name of the compressed folder, from now we call it *zip_name*, helps us to distinguish each records.

Firstly, we convert the binary signal, using the header file, into a text file. The conversion is done using the *rdsamp* application. To produce the desired outcome we have to specify some options in the application.

```
command1='rdsamp -r '+file_name+' -H -t 0:20 -v -pd >'+zip_name+'_sampl.txt'
os.system(command1)
```

The required option is the *-r* that is followed by the name of the record. The next options are optional. The options that we used are -H that reads the signal in high-resolution mode. The option *-t 0:20* represents the time that the record starts and stops and the duration of the ECG signal in seconds. We use the first 20 seconds for each record. The option *-v* prints the column headings. The option *-pd* prints the time of the day and the date as [hh:mm:ss DD/MM/YYYY]. The last option *>'+zip_name+'_sampl.txt'* saves the record in decimal into a text file using the name of the compressed folder as it is stored in the database. For example, the name *s0552_re_sampl.txt* consists of the name of the compressed folder into the database, which is the *s0552_re*, and the *_sampl.txt*, which is an extra parameter to help us to recognize all the files. The command *os.system(command1)* executes the above command using the Standard C function system and the *os* module which provides extra functionality.

An example of the text file that contains the samples is displayed in Figure 5-16. The samples are decimals and are stored in columns for each lead every 1 msec.

Elapsed time	i	ii	iii	avr	avl	avf	v1	v2	v3	v4	v5	v6
hh:mm:ss.mmm	(mV)	(mV)	(mV)	(mV)	(mV)	(mV)	(mV)	(mV)	(mV)	(mV)	(mV)	(mV)
0:00.000	0.146	-0.343	-0.490	0.099	0.319	-0.417	0.012	-0.218	-0.096	-0.429	-0.073	-0.138
0:00.001	0.147	-0.342	-0.489	0.098	0.319	-0.416	0.013	-0.217	-0.095	-0.427	-0.072	-0.137
0:00.002	0.144	-0.340	-0.485	0.098	0.315	-0.412	0.013	-0.218	-0.096	-0.429	-0.072	-0.139
0:00.003	0.143	-0.339	-0.483	0.098	0.313	-0.411	0.013	-0.218	-0.094	-0.428	-0.070	-0.139
0:00.004	0.143	-0.345	-0.489	0.102	0.316	-0.417	0.015	-0.215	-0.093	-0.425	-0.070	-0.140
0:00.005	0.143	-0.346	-0.490	0.102	0.317	-0.419	0.015	-0.215	-0.093	-0.426	-0.070	-0.140
0:00.006	0.144	-0.345	-0.490	0.101	0.318	-0.418	0.015	-0.215	-0.093	-0.425	-0.070	-0.141
0:00.007	0.144	-0.348	-0.492	0.102	0.319	-0.420	0.014	-0.215	-0.092	-0.423	-0.070	-0.142
0:00.008	0.144	-0.347	-0.492	0.102	0.319	-0.420	0.015	-0.214	-0.092	-0.423	-0.071	-0.141
0:00.009	0.145	-0.346	-0.492	0.101	0.319	-0.419	0.017	-0.213	-0.092	-0.422	-0.070	-0.141
0:00.010	0.145	-0.344	-0.490	0.100	0.318	-0.417	0.016	-0.215	-0.092	-0.423	-0.070	-0.142
0:00.011	0.145	-0.344	-0.489	0.099	0.318	-0.416	0.015	-0.215	-0.094	-0.423	-0.072	-0.143
0:00.012	0.145	-0.344	-0.489	0.100	0.317	-0.417	0.015	-0.214	-0.093	-0.423	-0.072	-0.142
0:00.013	0.144	-0.344	-0.488	0.100	0.317	-0.416	0.015	-0.213	-0.093	-0.423	-0.071	-0.142
0:00.014	0.144	-0.345	-0.490	0.101	0.318	-0.418	0.013	-0.213	-0.094	-0.421	-0.072	-0.141
0:00.015	0.145	-0.342	-0.487	0.099	0.317	-0.414	0.013	-0.214	-0.095	-0.422	-0.072	-0.141
0:00.016	0.145	-0.340	-0.485	0.098	0.316	-0.413	0.013	-0.214	-0.094	-0.422	-0.071	-0.141
0:00.017	0.147	-0.342	-0.489	0.098	0.318	-0.415	0.013	-0.214	-0.094	-0.422	-0.070	-0.141
0:00.018	0.145	-0.341	-0.487	0.098	0.317	-0.414	0.013	-0.214	-0.095	-0.423	-0.071	-0.141
0:00.019	0.144	-0.342	-0.486	0.099	0.316	-0.414	0.013	-0.214	-0.095	-0.422	-0.072	-0.141
0:00.020	0.146	-0.340	-0.486	0.097	0.317	-0.413	0.013	-0.214	-0.095	-0.423	-0.072	-0.139

Figure 5-16 ECG samples for 12 leads

The next command is *wrsamp*, which multiplies the samples by 1000. The filter, which is the next command, needs integers and not decimals. The output of this command is a binary file and a header file with the same name *input_'+file_name+'*. The required options are the *-i* option that is followed by the name of the text file, which contains the samples, and the *-o* option which is the name of the files that will be created. The *-z* option does not copy column 0 that contains the milliseconds. The *-x 1000* option multiplies all input samples by 1000 before writing them to the output signal file.

command2='wrsamp -i '+zip_name+'_sampl.txt -o input_'+file_name+' -z -x 1000'

os.system(command2)

After multiplying the samples and converted them into integers we can filter them using a low-pass boxcar filter. The required options for the *fir* application are the *-i* option and the *-n* option that represent the signal which will be filtered and the name of the output files. The output is also two files, a binary and a header file. It is necessary to determine the coefficients using *-c* option, which is followed by numbers.

command3='fir -i input_'+file_name+' -n output_'+file_name+' -c .2 .2 .2 .2 .2'

os.system(command3)

A low-pass “boxcar” or rectangular window filter is a finite response filter (FIR), whose impulse response has non-zero values and has finite duration in a “window” containing of N samples. It also allows signals to pass with low frequency. The coefficients, which are equal to 0.2, are also equal to

$$\frac{1}{M+1} \tag{13}$$

where M=4, which represents the grade of the filter. The output of the filter is the average of the (M+1) samples that we take each time from the input. The output equals to

$$y(n) = \sum_{k=0}^{M} h(k)x(n-k) = \frac{x(n)+x(n-1)+\cdots+x(n-M)}{(M+1)} = \frac{x(n)+x(n-1)+x(n-2)+x(n-3)+x(n-4)}{(4+1)} \tag{14}$$

where h(k) are the coefficients of the filter and are equal to h(k)=[0.2 0.2 0.2 0.2 0.2].

In Figure 5-17, we can see the ECG signal before applying the filter.

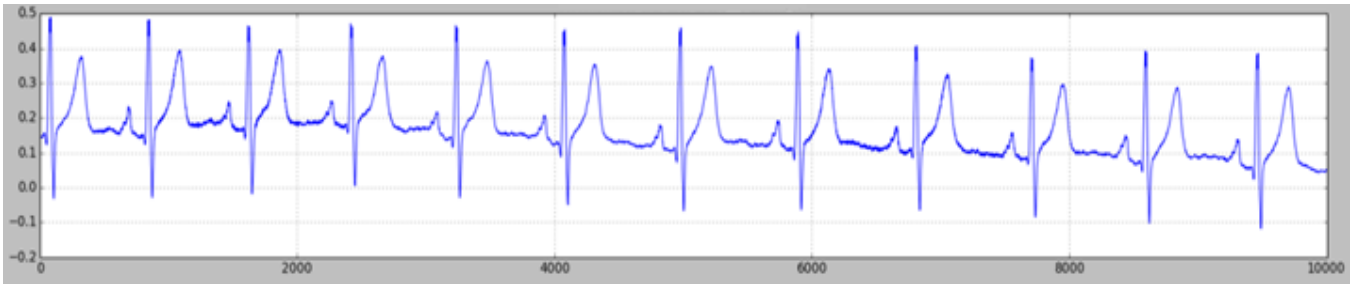


Figure 5-17 ECG signal before filter

Figure 5-18 displays the ECG signal after applying the filter. As we can see, the filter is smoother and the noise has removed.

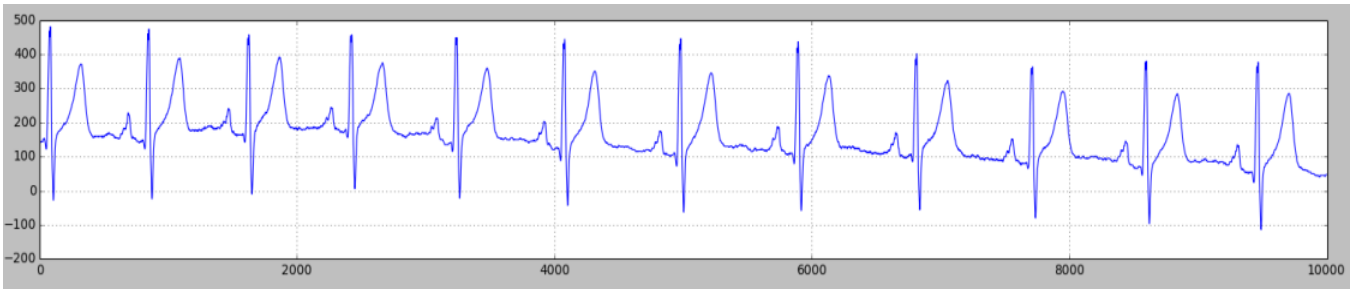


Figure 5-18 ECG signal after FIR filter

To read the ECG files, that the fir application exported, we use the *rdsamp* application. The option *-r* is followed by the name of the record, which is the output of the fir filter. The samples are saved in a text file with the name *'+zip_name+'_samples.txt'*.

```
command4='rdsamp -r output_'+file_name+' >'+zip_name+'_samples.txt'
os.system(command4)
```

An example of the samples after filter is displayed in Figure 5-19.

0	146	-343	-490	99	319	-417	12	-218	-96	-429	-73	-138
1	146	-342	-489	98	319	-416	13	-218	-96	-429	-72	-138
2	144	-342	-489	98	316	-416	13	-218	-95	-428	-72	-139
3	143	-342	-489	98	316	-416	13	-217	-94	-427	-70	-139
4	143	-345	-489	101	316	-417	15	-215	-93	-426	-70	-140
5	143	-345	-490	102	317	-418	15	-215	-93	-425	-70	-140
6	144	-346	-490	102	318	-419	15	-215	-93	-425	-70	-141
7	144	-346	-492	102	319	-419	15	-215	-92	-423	-70	-141
8	144	-346	-492	101	319	-419	15	-215	-92	-423	-70	-141
9	145	-346	-492	101	319	-419	15	-215	-92	-423	-70	-142
10	145	-344	-490	100	318	-417	15	-214	-92	-423	-71	-142
11	145	-344	-489	100	318	-417	15	-214	-93	-423	-71	-142
12	145	-344	-489	100	318	-417	15	-214	-93	-423	-72	-142
13	145	-344	-489	100	317	-416	15	-214	-94	-423	-72	-142
14	145	-344	-488	100	317	-416	13	-214	-94	-422	-72	-141
15	145	-342	-488	99	317	-415	13	-214	-94	-422	-71	-141
16	145	-342	-487	98	317	-414	13	-214	-94	-422	-71	-141
17	145	-342	-487	98	317	-414	13	-214	-95	-422	-71	-141
18	145	-341	-486	98	317	-414	13	-214	-95	-422	-71	-141
19	146	-341	-487	98	317	-414	13	-214	-95	-423	-72	-141
20	146	-340	-486	97	317	-413	13	-214	-95	-423	-72	-139

Figure 5-19 ECG samples after filter

The next command is to determine the QRS complexes. The required option is the *-r* option, which is followed by the name of the file. The output of that application is an annotation

file with the same name as filename and with the suffix .qrs. It is used by the next two applications.

command5='sqrs -r '+file_name

os.system(command5)

After detecting the QRS complexes, we use the *ann2rr* application that determines the R-R intervals. The required options are the *-r* option and the *-a* option. The first one is followed by the name of the annotation file and the second one is followed by the suffix of the annotation file, which is *qrs*. Both of them are the output of the *sqrs* application. The *-v T* option displays the time that each interval stops. The format of the time is [hh:mm:ss dd/mm/yyyy]. The *-i s3* option prints the intervals using a specific format. The format that is used includes the time in seconds and the number 3, which has to be an integer between 0 and 15, represents the number of decimal places. The *-V T* option prints the initial times using the format [hh:mm:ss dd/mm/yyyy]. The *-w* option prints the type of the final annotations and the *-W* option prints the type of the initial annotations. The annotation types [60] are symbolized by some letters and symbols and some of them are: N for normal beat, L for left bundle branch block beat, R for right bundle branch block beat, B for bundle branch block beat and A for atrial premature beat. Figure 5-20 depicts the start and stop time (first and last columns) of the heartbeats and the difference between them.

command6='ann2rr -r '+file_name+' -a qrs -v T -i s3 -V T -w -W >'+zip_name+'_rr_intervals.txt'

os.system(command6)

0:00.000	[0]	0.068	N	0:00.068
0:00.068	N	0.764	N	0:00.832
0:00.832	N	0.776	N	0:01.608
0:01.608	N	0.800	N	0:02.408
0:02.408	N	0.812	N	0:03.220
0:03.220	N	0.840	N	0:04.060
0:04.060	N	0.900	N	0:04.960
0:04.960	N	0.912	N	0:05.872
0:05.872	N	0.920	N	0:06.792
0:06.792	N	0.900	N	0:07.692
0:07.692	N	0.884	N	0:08.576
0:08.576	N	0.868	N	0:09.444
0:09.444	N	0.836	N	0:10.280
0:10.280	N	0.824	N	0:11.104
0:11.104	N	0.820	N	0:11.924
0:11.924	N	0.828	N	0:12.752
0:12.752	N	0.856	N	0:13.608
0:13.608	N	0.864	N	0:14.472
0:14.472	N	0.856	N	0:15.328
0:15.328	N	0.856	N	0:16.184
0:16.184	N	0.864	N	0:17.048
0:17.048	N	0.864	N	0:17.912
0:17.912	N	0.848	N	0:18.760
0:18.760	N	0.820	N	0:19.580
0:19.580	N	0.796	N	0:20.376
0:20.376	N	0.804	N	0:21.180

Figure 5-20 R-R Intervals

The last application before the applications for detecting the waveforms is the *ihr*. Ihr reads an annotation file and produces an instantaneous heart rate signal. The annotation file is the output of the *sqrs* application. The required options are the *–r* option and the *–a* option. The first option is followed by the name of the annotation file and the second one is followed by the suffix of the annotation file, which is *qrs.* The option *>'+zip_name+'_beats.txt'* writes the beats into a text file (Figure 5-21).

```
command7='ihr -r '+file_name+' -a qrs >'+zip_name+'_beats.txt'
os.system(command7)
```

22.896	78.534
23.660	78.125
24.428	77.3196
25.204	78.534
25.968	78.9474
26.728	78.125
27.496	78.534
28.260	79.7872
29.012	79.7872
29.764	79.3651
30.520	78.9474
31.280	79.3651
32.036	79.3651
32.792	78.534
33.556	78.125
34.324	78.9474
35.084	78.534

Figure 5-21 Instantaneous heart rate

The last applications that are used are the *ecgpuwave* and *rdann*. Ecgpuwave analyses an ECG signal, detecting the QRS complexes and locating P, QRS, and T waveforms (Figure 5-22). The options that required are the *–r* option which is followed by the filename and the *–a* option which is followed by the WFDB-format annotation file. The output is a file with the name same as the binary signal file and the suffix *annotator*. We analyze the ECG signal for every lead. An extra option that we have to set is the number of the signal that we want to analyze. For every signal, we have to set the *–s* option that represents which signal will be analyzed. For the first signal, which is lead I, it is optional to set the –s option because it is 0.

```
command8='ecgpuwave -r '+file_name+' -a annotator'
os.system(command8)
```

To make the output of the ecgpuwave readable we use the *rdann* application. The required options are the *–r* and the *–a* options. The first option is followed by the name of the output of *ecgpuwave* and the second one is followed by the suffix of the file that is the output of the ecgpuwave. The *–v* option prints the column headings. The option *>'+zip_name+'_pqrst_i.txt'* saves the analyzed signal into text file.

```
command9='rdann -r '+file_name+' -a annotator -v >'+zip_name+'_pqrst_i.txt'
os.system(command9)
```

To analyzed the 12 leads ECG signal we set the –s option for every lead. To analyze the second signal, which is lead II, we set *–s 1*. For the lead III we set *–s 2*. After the *ecgpuwave* we call *rdann* to make the files readable. The *ecgpuwave* is executed 12 times so *rdann* does.

```
command10='ecgpuwave -r '+file_name+' -a annotator -s 1'
os.system(command10)

command11='rdann -r '+file_name+' -a annotator -v >'+zip_name+'_pqrst_ii.txt'
os.system(command11)
```

Time	Sample #	Type
0:00.542	542	(
0:00.572	572	p
0:00.608	608	)
0:00.651	651	(
0:00.710	710	N
0:00.747	747	)
0:00.870	870	(
0:00.941	941	t
0:01.030	1030	)
0:01.269	1269	(
0:01.299	1299	p
0:01.334	1334	)
0:01.379	1379	(
0:01.437	1437	N
0:01.474	1474	)
0:01.583	1583	(
0:01.670	1670	t
0:01.755	1755	)

Figure 5-22 PQRST waveforms in lead I

The files that are produced by the WFDB application are saved into the same folder that the ECG files are. Using those files, we have implemented some algorithms to determine the amplitudes, the intervals and the other findings that we need. In the end of the ECG signal processing all the files are deleted for saving memory. The deletion is done after storing the findings into the database.

Processing Algorithms

Figure 5-23 depicts the steps that form the procedure of ECG analysis. Initially, the ECG signal is uploaded using SCP-ECG format by the patient. This signal passes through filtering process for noise removal (ECG filtering step). Then the signal is used for extracting the required metrics (amplitude, duration and axis) of the ECG waveforms via the corresponding processing steps that are used for disease detection. This medical data is kept into the OpenMRS database. The last procedure before ECG visualization is that the system detects a heart disease and informs the patient's doctor in case of abnormal findings.

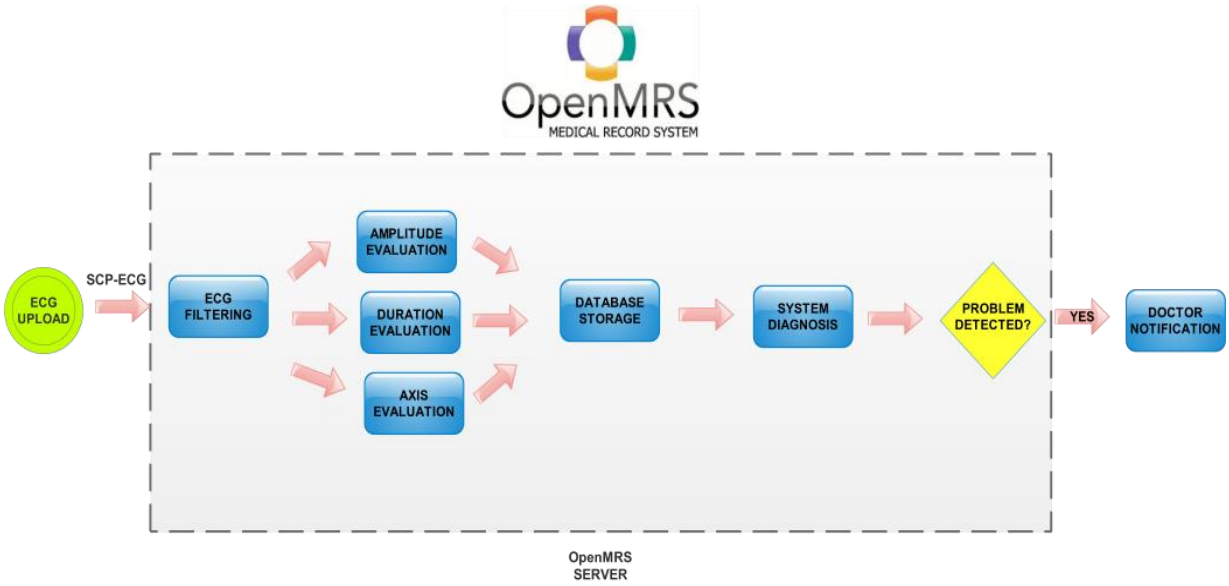


Figure 5-23 ECG Analysis flowchart

Baseline Calculation

Using the results of the WFDB applications, we calculate the ECG findings. Firstly, we have to determine the baseline. The samples of the ECG signals are not evenly distributed on the axes. Some of them are above the baseline and the waveforms that have to be below the baseline, are above, like Q-waves and S-waves. To avoid that issue we determine the area that we know we have zero potential in an ECG signal. That area is between the end of T-wave to the beginning of P-wave so we calculate the average of the amplitudes in that interval and subtract it from each sample from the entire signal. To detect all amplitudes of the signal we process two of the files that the WFDB applications export. The first file is the file with all samples after filter and the second one is that which displays the ECG waveforms. From the second file, we take the intervals that are from end of the T-wave to the beginning of the P-wave. Using those intervals, we take the samples from the first file between them.

Time	Sample #	Type
0:00.542	542	(
0:00.572	572	p
0:00.608	608	)
0:00.651	651	(
0:00.710	710	N
0:00.747	747	)
0:00.870	870	(
0:00.941	941	t
0:01.030	1030	)
0:01.269	1269	(
0:01.299	1299	p
0:01.334	1334	)
0:01.379	1379	(
0:01.437	1437	N
0:01.474	1474	)
0:01.583	1583	(
0:01.670	1670	t
0:01.755	1755	)
0:01.995	1995	(
0:02.025	2025	p
0:02.060	2060	)
0:02.104	2104	(
0:02.162	2162	N
0:02.208	2208	)
0:02.304	2304	(
0:02.395	2395	t
0:02.487	2487	)
0:02.708	2708	(
0:02.737	2737	p
0:02.772	2772	)

Figure 5-24 Intervals to detect baseline

As we can see from the Figure 5-24, the intervals that we need are those between the red lines. With *ecgpuwave* we have located the waveforms. The parentheses describe when a wave starts and stops. From the first column, we take the times of all the intervals. Then we take all the samples between those intervals from the file that has the filtered samples. The next step is to find the average of the samples and subtract the average from all the samples of the signals. The samples, after *wrsamp,* are multiplied by 1000 so we have to divide them by 1000 but the results have to be between single-digit and two-digit integers to be estimated by the doctors, so we divide them by 100. All samples for 12 leads are distributed in the same way.

To take the intervals we use a function that reads each column of the file that contains the waveforms. The function searches the third column for the P-waves and T-waves. When the letter ‘p’ is detected, we take the previous time and when the letter ‘t’ is detected, we take the next time. The times that represent the start or the stop of the wave, are stored into a table with the name wave (figure 5-25).

```
if var_name=='p':
    prev_sample=file_data_pqrst[i-1]
    splitted_pr=prev_sample.split('    ')
    wave.append(int(splitted_pr[2]))
    #print 'start_wave %s' %wave
elif var_name=='t':
    next_samp=file_data_pqrst[i+1]
    splitt_pr=next_samp.split('    ')
    wave.append(int(splitt_pr[2]))
    #print 'end_wave %s' %wave
else:
    break
```

Figure 5-25 Take previous and next times

In a normal ECG signal, the number of the P-waves is the same as the number of the QRS complexes and the number of the T-waves. In the case, the number of the P-waves is the same as the number of T-waves, to match the waves we remove the first P-wave and the last T-wave. Then we process the rest waves. From the below tables we take the first P-wave with the first T-wave, the second P-wave with the second T-wave etc. Those waves are the intervals that we use to find the samples.

P-waves = [1030.0, 1755.0, 2487.0, 3192.0, 3892.0, 4590.0, 5300.0, 6025.0]

T-waves = [1269.0, 1995.0, 2708.0, 3404.0, 4103.0, 4809.0, 5535.0, 6272.0]

In an abnormal ECG signal, the number of the waves may be different. In this case, we have to find the corresponding pair for each wave. For every pair the start time, which is in the table P-waves, has to be less than the corresponding stop time, which is in the table T-waves. For the example where P-waves = [1031, 1758, 2485, 3192] and T-waves = [554, 1281, 2005, 2716, 3414], the first P-wave is greater than the first T-wave, so we have to erase it. The example where P-waves = [1043.0, 1774.0, 2500.0, 3211.0] and T-waves = [0.0, 0.0, 1245.0, 2680.0] we have to match the P-waves with the available T-waves. To match them we use a threshold that

checks the distance between the waves. The threshold equals to 300 because is the maximum distance between the P-wave and T-wave. The first P-wave does not match with the first T-wave so we move on the next T-wave until we find the corresponding, if there is. Neither the second T-wave matches with the first P-wave, so we take the third one. The third T-wave matches so we keep it. Then we continue matching. Another issue is to match the right pair. For example the P-wave=1774.0 matches with the T-wave=2680.0 but also the P-wave=2500.0 matches. In that case we take the number with the less distance which is P-wave=2500.0. The numbers that do not match are erased. The rest process is the same as in a normal ECG.

In case of there are not any P-waves or T-waves we use the QRS complex. For example if P-waves are not detected, we take the beginnings of the QRS complexes and match them with the T-waves. If T-waves are not detected, we use the ends of the QRS complexes and match them with the P-waves.

In figures below, we can see the ECG signal above the baseline (figure 5-26) and the removal after processing (figure 5-27).

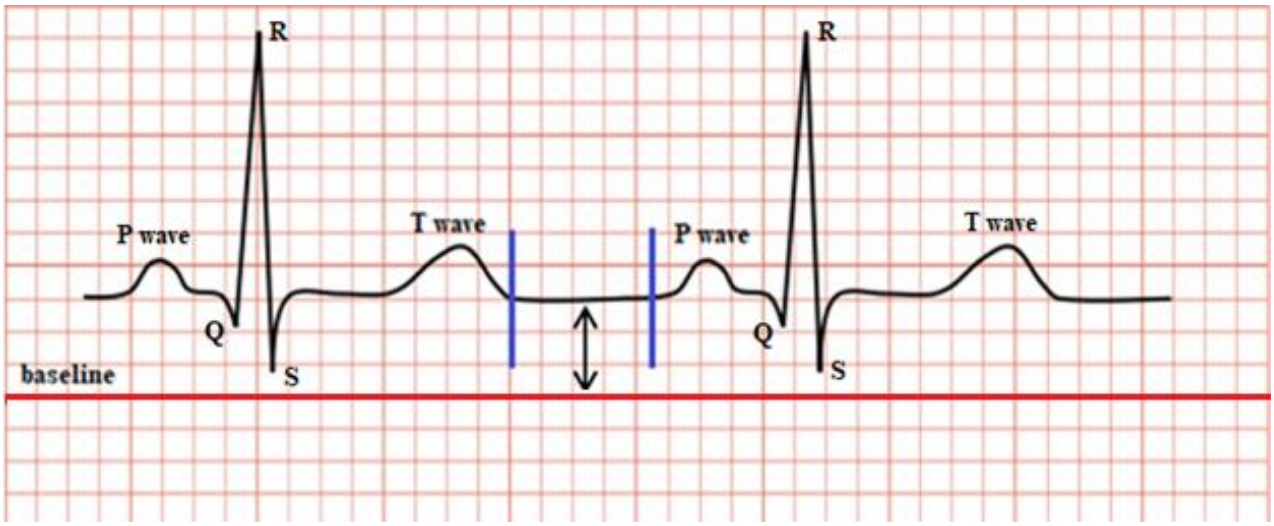


Figure 5-26 ECG signal above the baseline

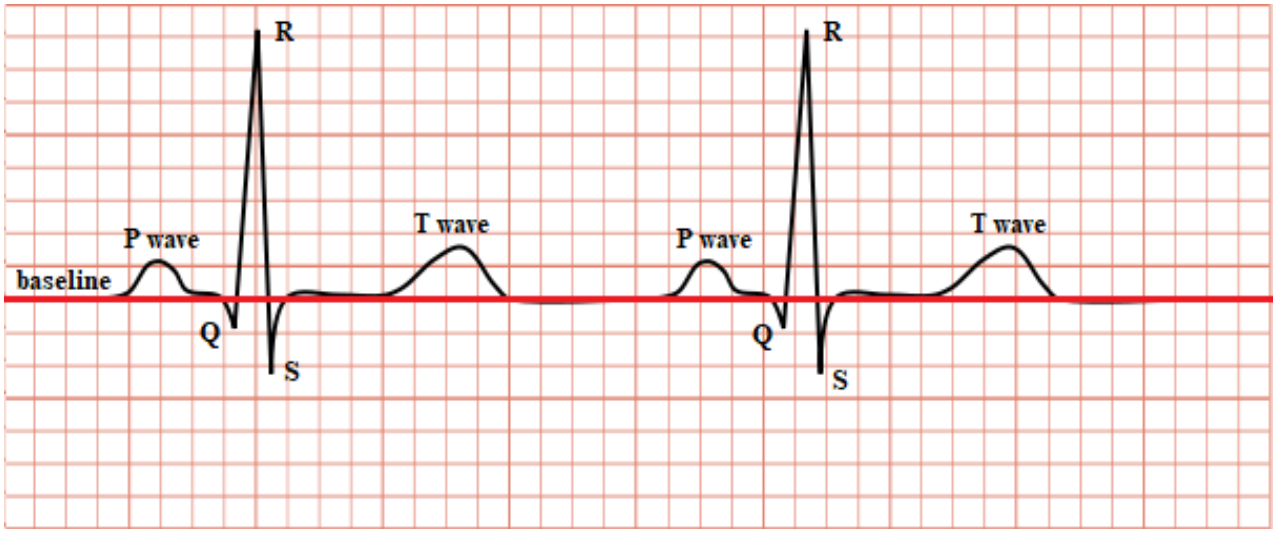


Figure 5-27 ECG signal on the baseline

Intervals Calculation

Using the files that *ecgpuwave* exports, we also estimate the duration and the heights of each wave. For all the leads, we calculate the duration and the heights of the second pulse randomly. Firstly, we estimate the duration of each wave calculating the interval for the left parenthesis to right containing the corresponding wave. Figure 5-31 shows that the duration of the second P-wave is the interval from the start time 0:01.439 to the end time 0:01.502. For better convenience, we can also calculate the number of samples. The duration of the wave calculating the samples is the subtracting between the samples that the wave starts from the sample that the wave ends and multiplying by 0.001 to find the seconds. For example in lead I, P-wave duration is (1502-1439)*0.001=63*0.001=0.063 sec (Figure 5-28). The QRS duration is (1675-1585)*0.001=0.090 sec (Figure 5-29) and the T-wave-duration is (1943-1771)*0.001=0.172 sec (Figure 5-30).

Time	Sample #	Type
0:00.659	659	(
0:00.687	687	p
0:00.722	722	)
0:00.812	812	(
0:00.844	844	N
0:00.897	897	)
0:00.992	992	(
0:01.081	1081	t
0:01.164	1164	)
0:01.439	1439	(
0:01.465	1465	p
0:01.502	1502	)
0:01.585	1585	(
0:01.621	1621	N
0:01.675	1675	)
0:01.771	1771	(
0:01.859	1859	t
0:01.943	1943	)

Figure 5-31 Duration of ECG waves

0:01.439	1439	(
0:01.465	1465	p
0:01.502	1502	)

Figure 5-28 P-wave duration

0:01.585	1585	(
0:01.621	1621	N
0:01.675	1675	)

Figure 5-29 QRS complex duration

0:01.771	1771	(
0:01.859	1859	t
0:01.943	1943	)

Figure 5-30 T-wave duration

The amplitudes of each wave are calculated using the start and stop time of the corresponding wave and then searching into the file that contains the filtered samples to find the maximum or the minimum value. To find the height of the P-wave in lead II we first estimate the start/stop time (Figure 5-32).

0:01.397	1397	(
0:01.456	1456	p
0:01.501	1501	)

Figure 5-32 P-wave duration in lead II

Amplitudes calculation

The amplitudes of each wave are calculated after determining the ECG intervals. The next step is to find the samples that are contained between those intervals. In the Figure 5-33, we can view all the samples that the P-wave contains. For that example we want the samples in lead II, so we take the samples from the third column (the first column contains the number of samples and the second the values in lead I). To find the amplitudes we calculate the maximum value of the specific samples.

Samples	i	ii	iii	avr	avl	avf	v1	v2	v3	v4	v5	v6
	(mV)	(mV)	(mV)	(mV)	(mV)	(mV)	(mV)	(mV)	(mV)	(mV)	(mV)	(mV)
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
1397	185	-248	-433	32	309	-341	-1	-191	-66	-379	-50	-116
1398	185	-248	-433	32	309	-341	-2	-191	-66	-379	-50	-116
1399	185	-248	-433	32	309	-341	0	-191	-66	-378	-50	-116
1400	185	-248	-433	32	309	-341	1	-190	-64	-377	-50	-116
1401	185	-244	-433	29	309	-339	1	-187	-62	-371	-46	-115
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
1497	174	-268	-445	47	309	-356	-17	-206	-80	-383	-53	-124
1498	174	-272	-446	49	309	-359	-16	-205	-81	-383	-55	-126
1499	172	-273	-447	50	310	-359	-15	-205	-81	-384	-56	-126
1500	172	-275	-447	51	310	-361	-15	-205	-82	-384	-56	-126
1501	172	-279	-452	53	312	-365	-15	-205	-83	-385	-59	-125

Figure 5-33 Samples in P-wave

The next example is to locate the amplitudes of the QRS complex in lead I (Figure 5-34). With similar way, we find the interval.

0:01.585	1585	(
0:01.621	1621	N
0:01.675	1675	)

Figure 5-34 QRS duration in lead I

As we can see from the Figure 5-35, the sample that begins the QRS complex is the 1585 and the sample that ends is the 1675, so we take the samples between them.

Samples	i (mV)	ii (mV)	iii (mV)	avr (mV)	avl (mV)	avf (mV)	v1 (mV)	v2 (mV)	v3 (mV)	v4 (mV)	v5 (mV)	v6 (mV)
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
1585	167	-318	-482	77	323	-401	34	-161	-56	-381	-79	-154
1586	164	-328	-494	81	331	-411	48	-147	-49	-382	-84	-161
1587	161	-346	-507	93	334	-427	68	-128	-40	-386	-98	-172
1588	152	-370	-524	110	338	-448	102	-93	-22	-388	-115	-189
1589	152	-401	-553	124	353	-477	141	-51	1	-388	-135	-210
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
1671	166	-196	-362	15	264	-279	2	-189	-50	-356	-25	-107
1672	166	-196	-362	14	264	-279	11	-175	-34	-347	-23	-107
1673	167	-196	-362	14	265	-279	20	-164	-23	-340	-21	-107
1674	172	-197	-369	12	271	-282	32	-149	-9	-328	-18	-107
1675	176	-200	-376	11	276	-287	39	-140	0	-322	-15	-108

Figure 5-36 Samples in QRS Complex

In QRS complex, we have to locate the Q-wave, the R-wave and the S-wave. Firstly, we locate the R-wave. The maximum value of that interval is the R-wave. The next wave that we will find is the Q-wave. The Q-wave is the minimum value between the samples that the QRS complex starts to the sample that we have locate the R-wave. The S-wave is the minimum value between the location of the R-wave, and the end of the QRS complex (Figure 5-36).

Samples	i (mV)	ii (mV)	iii (mV)	avr (mV)	avl (mV)	avf (mV)	v1 (mV)	v2 (mV)	v3 (mV)	v4 (mV)	v5 (mV)	v6 (mV)	
.	.	.	.	.	.	.	.	.	.	.	.	.	
.	.	.	.	.	.	.	.	.	.	.	.	.	
.	.	.	.	.	.	.	.	.	.	.	.	.	
1585	167	-318	-482	77	323	-401	34	-161	-56	-381	-79	-154	
1586	164	-328	-494	81	331	-411	48	-147	-49	-382	-84	-161	
1587	161	-346	-507	93	334	-427	68	-128	-40	-386	-98	-172	Q-wave
1588	152	-370	-524	110	338	-448	102	-93	-22	-388	-115	-189	
1589	152	-401	-553	124	353	-477	141	-51	1	-388	-135	-210	
.	.	.	.	.	.	.	.	.	.	.	.	.	
.	.	.	.	.	.	.	.	.	.	.	.	.	
.	.	.	.	.	.	.	.	.	.	.	.	.	
1625	457	1446	988	-951	-265	1216	-1146	-1383	-386	566	1416	910	
1626	459	1504	1044	-982	-291	1274	-1180	-1408	-387	645	1476	946	
1627	459	1504	1044	-982	-291	1274	-1210	-1419	-390	712	1528	966	→ R-wave
1628	459	1504	1044	-982	-291	1274	-1210	-1438	-390	712	1573	979	
1629	456	1469	1012	-962	-278	1241	-1210	-1438	-431	712	1573	979	
.	.	.	.	.	.	.	.	.	.	.	.	.	
.	.	.	.	.	.	.	.	.	.	.	.	.	
.	.	.	.	.	.	.	.	.	.	.	.	.	
1671	166	-196	-362	15	264	-279	2	-189	-50	-356	-25	-107	
1672	166	-196	-362	14	264	-279	11	-175	-34	-347	-23	-107	
1673	167	-196	-362	14	265	-279	20	-164	-23	-340	-21	-107	S-wave
1674	172	-197	-369	12	271	-282	32	-149	-9	-328	-18	-107	
1675	176	-200	-376	11	276	-287	39	-140	0	-322	-15	-108	

Figure 5-35 Locate Q, R, S waves

As we can see from the above image the maximum value, which is the amplitude of the R-wave in lead I, is the 459. As we have mentioned the samples have multiplied by 1000 so we have to divide them by 100 to convert them into mm. Another processing that should be done is to subtract the value that baseline has, from the amplitude of R-wave. The baseline has calculated and is equals to 176.65. We make a rounding and the value that will store into the database will be (459-176.65)/100 = (282.35/100)=2.8 mm.

R_amp_i=abs (R_amplitude_i[0]-x_axis_i)

R_ampl_i=round (R_amp_i/100,1)

To estimate the Q-wave we take the sample from beginning to the sample that contains the R-wave and calculate the minimum value in that interval. From the previous example, the R-wave was on the 1627 sample. Therefore, we search from 1585 sample to 1627 for the minimum value. We also subtract from axis, make a rounding and divide by 100.

Q_amp_i=abs (Q_amplitude_i[2]-x_axis_i)

Q_ampl_i=round (Q_amp_i/100,1)

To estimate the S-wave we search the samples from the sample that the R-wave is, to the sample that ends the QRS complex. For example, the interval that we will search for the minimum value is now from 1627 to 1675. After that, we subtract by the axis, round the value and divide by 100.

S_amp_i=abs (S_amplitude_i[2]-x_axis_i)

S_ampl_i=round (S_amp_i/100,1)

According to the amplitudes that we have to calculate, we search in different columns every time. In addition, we have to subtract by the corresponding axis for every value. To estimate the QRS complex in lead V1 we search the samples in eighth column but we have to subtract by the axis that we found using the corresponding file, that *ecgpuwave* has exported.

To calculate the P-R duration we use the sample that P-wave starts and the sample that the Q-wave is located. We find the subtracting and then we multiply by 0.001 to calculate the duration in msec. The R-R interval is given by the *ann2rr* application so we have not to calculate it using the samples. We take the average of R-R intervals and then rounding. The Q-T duration is from the beginning of the QRS complex to the end of the T-wave. Therefore, we calculate the interval from the sample in the beginning of QRS to the sample that the T-wave ends. To

calculate the QTc, which is the Bazzett's formula, we calculate QT/sqrt(R-R) and multiply by 0.001 to find the duration in msec.

To calculate the R/S ratio we estimate the amplitude of the R-waves and the amplitude in S-waves in leads V1, V2 and V6. Then we divide the R-waves by the corresponding S-waves.

The beats per minute are calculated using the file that *ihr* application exports. We find the average of all the instantaneous beats and we round them. Then we check if there is bradycardia or tachycardia. If the beats per minute are less than 60 bpm we have bradycardia, if the beats are more than 100 bpm we have tachycardia and if the beats are between 60 bpm and 100 bpm they are normal.

Heart Axes Calculation

The last values that we have to estimate are the QRS-axis, P-axis and QRS Tangle. For all axis values, we have to calculate two vectors. For the QRS axis, the first vector is the sum of the Q-wave, the R-wave and the S-wave in lead I. The second vector is the sum of the Q-wave, the R-wave and the S-wave in lead III. In the below cycle (Figure 5-37) each axis represents a limb lead. The sum in lead I is located on the axis that represents the lead I in the below figure. The sum in lead III is located on the axis that represents the lead III. The next step is to draw a vertical line that starts from lead I and a vertical line from lead III until those lines will be intersected (the red lines on the Figure 5-37). From the intersection point, we draw a line to the start of axes and we evaluate the angle in degrees clockwise. For the vector in lead I we know the coordinates. The y-coordinate is zero and the x-coordinate is the sum of the waves. To find the coordinates for the vector in lead III we use the formula of sine and cosine in the yellow right triangle, the angle between the leads, which equals to 30 degrees and the sum of waves, which is the hypotenuse of that right triangle.

$$\text{x-coordinate} = \text{hypotenuse} * \cos\frac{pi}{3} \tag{15}$$

$$\text{y-coordinate} = \text{hypotenuse} * \sin\frac{pi}{3} \tag{16}$$

After the above process, we calculate the slope of the line that is vertical to lead III. We know by definition that when we multiply the slopes of two vertical lines the result equals to -1.

The slope of the line that is vertical on lead I equals to λ_1 =y_coordinate/x_coordinate. So the slope of the line that is vertical on lead III is λ_2 = (-1)* λ_1. Using λ_2 we can estimate the linear equation of the vertical line, which is the left red line. Then we use the sum of the waves in lead I, which is the point that the vertical lines are connected, and solve the linear equation.

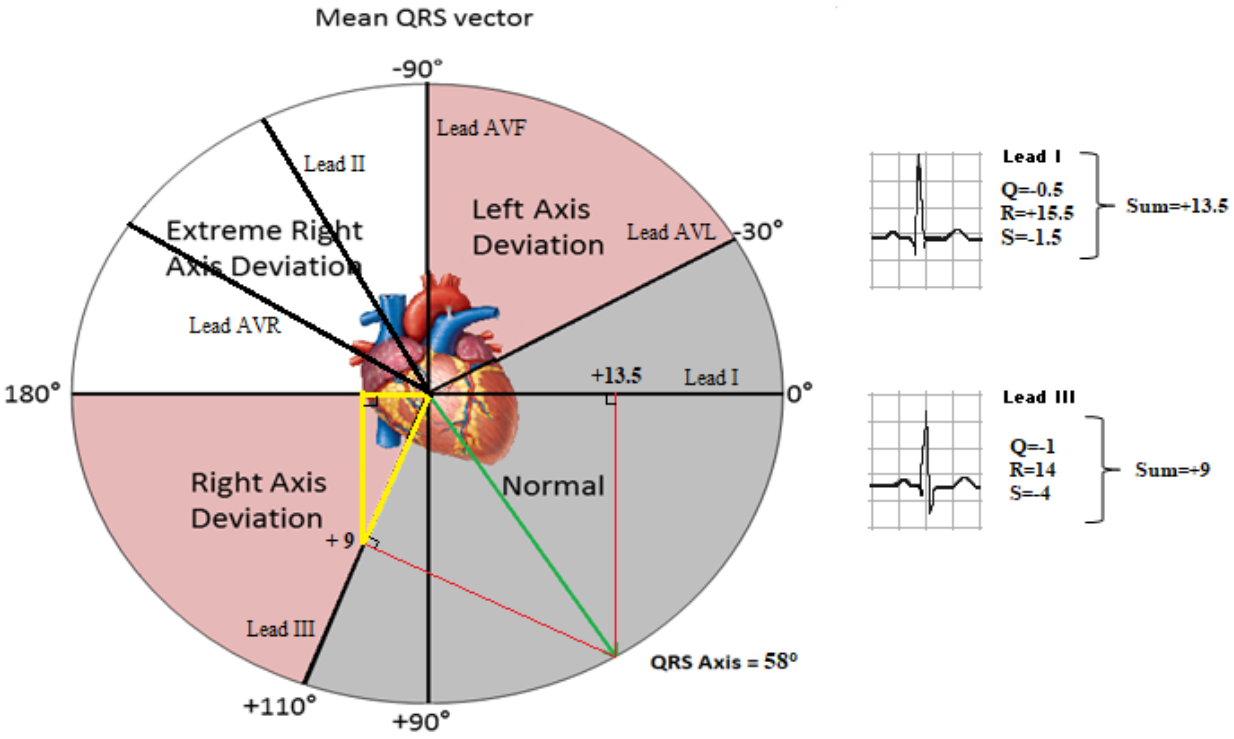


Figure 5-37 QRS Axis Calculation

With the same way, we find the P-axis and T-axis by calculating the amplitudes of the P-waves and T-waves in lead I and in lead III. We use T-axis to calculate the QRS Tangle. We subtract T-axis from QRS-axis. We have to check if the QRS-Tangle's angle is greater than 180 degrees and if it is, we subtract from 360^o.

All the samples are stored in lists separately. Before the connection to database, we convert each list, which contains the samples of each lead, to a string and send them all together separating by underscore. The strings that contain the samples of each lead are stored separately into the database.

Connection to Database

After collecting all the ECG findings we connect to database to send the values and then to display them in the OpenMRS platform. The platform runs on localhost so the arguments we need are the IP of localhost, the username and the password that the platform uses and the port that runs the MySQL.

```
db = MySQLdb.connect("127.0.0.1","openmrs","nkJe5jnCcoHA","openmrs",3316)
cur = db.cursor()
```

When the compressed folder is uploaded into the database, an Observation is created. This creation means that the table, with the name *obs* into the database, will be updated with new fields. Except the *obs* table, the tables *encounter* and *encounter_provider* will be updated with new fields. The table *encounter_provider* is not visible in the platform but it is necessary to be filled for the correct operation of the platform.

As we have mentioned, an Observation is part of an Encounter and an Encounter part of a Visit. When we want to add an Observation, we have ensured that there is an Encounter and a Visit. A Visit may contain one or more Encounters and Encounter one or more Observations. During uploading the compress folder, a Visit has already created to allow the uploading. An Encounter has also created which contains the information that a file has uploaded. For the current implementation, we will store our data in the same Visit but we will create a new Encounter, with the name Electrocardiogram, which contains forty-two Observations. Each Observation is a Concept, which has created in the platform, and it contains each value we want to store into the database. For example to store the value that the QRS axis has, we have created a concept named QRS_axis.

To connect the Encounter of the uploading file with the Observations we want to store, we have to take some values from the database. Those values are taken using queries in SQL. Firstly, we take the person_id, encounter_id, obs_datetime, location_id and the creator from the table *obs* after the file uploading. Then we take the visit_id and the last encounter_id from the table *encounter*. To create our Encounter, which is the Electrocardiogram, we have to update the table *encounter_provider* which for every insertion of an Encounter it is updated. After that, we create the Electrocardiogram and we take again the last encounter_id. Those values are useful

because we know exactly which file will be processed, who patient is with all his medical examinations, the location where the patient is and the date.

After the collection of the data, we start the insertion of the processing data, which are the ECG findings. For each insertion, we use different queries. Each insertion in the database has a UUID, which is unique. To create uuids we use a function in python, which guarantees that each uuid is unique. Moreover, when a concept is created, except its UUID, a number is created which uses to store the concepts into the concept dictionary in increasing series. We will use that number to store the data into the corresponding concept. The following example is a query, which is used for the insertion of the beats per minute. The field *person_id* uses to distinguish the patients. The field *concept_id* represents the number of the concept in which we will store the data. The *encounter_id* is the id of the encounter Electrocardiogram. The field *obs_datetime* equals to *date_created.* The *location_id* is the id of the current location and the *creator* is the user who makes the insertion. The *uuid* value is the unique value, which each concept must have to store into the database. In case of double uuids the insertion is cancelled. The last value named *value_numeric* contains the value we want to store. For the current example the beats per minute is a number so we defined the concept as numeric. In case of tachycardia or bradycardia, we defined a concept as *value_text* because it contains strings.

query = ("INSERT into openmrs.obs (person_id, concept_id, encounter_id, obs_datetime, location_id, value_numeric, creator, date_created, uuid) values (%s, '163142', %s, %s, %s, %s, %s, %s, %s) ")

bpm_insert=cur.execute (query, (person_id, last_encounter_id_after_new_obs, obs_datetime, location_id, bpm, creator, obs_datetime, uuid_1))

Display Results

After collecting all data and store them into the database we will use HTML Forms to display them. HTML Forms allow us to write in HTML but also in JavaScript and CSS. It is located into the encounter we want and has the appearance that helps us to display the results.

The HTML form, that we will create, includes the encounter Electrocardiogram. We declare, in the beginning of the html form, the id of the encounter we will use. *EncounterLocation*, which represents the current location of the medical examination, *encounterProvider*, which is the current user and *encounterDate*, which is the current date, should be defined in every HTML form.

Using JavaScript we take the samples that are stored as a string into the database, we convert them into numbers and saved them into a table. The declaration *obs id=”col_i” conceptId=”163168”* means that the concept with the id 163168 will be called col_i. That concept contains the samples in lead I. With JQuery and the command *getValue* we take the value of the specific concept which is a string. Then we split the string by underscore and convert them into numbers. After that, we save them into a table (Figure 5-38).

```
<div class="hidden"><obs id="col_i" conceptId="163168"/></div>
<script type="text/javascript">
jQuery(function() {

var str=getValue("col_i.value")
var temp = new Array();
temp = str.split("_").map(parseFloat);

var arr1=new Array();

var arr2=new Array();
arr2.push("TIME");
arr2.push("VALUE");
arr1.push(arr2);

for (var i=0; i&lt;(temp.length); i++){
    arr2=new Array();
    arr2.push(i);
    arr2.push(temp[i]);
    arr1.push(arr2);
}
```

Figure 5-38 Get samples from database

Using google charts, we make the graph, which contains the values that are stored in the table. With similar way, we make the 12 graphs of the Electrocardiogram. The values of each graph are stored in different concepts and using their conceptId, we take the values.

To display the ECG findings we use the *lookup expression*, which allows the evaluation of velocity expressions [61]. The variable we want to have access to in the velocity context is the concept. To take the name of a concept we use the below expression.

<lookup expression="fn.getConcept('163142').name"/>

After getting the name, we also use JQuery to take each value from the database. Before display, them in the HTML we check if the values are normal or not. We compare them with the corresponding limits (Figure 5-39). If the values are normal they are displayed in black letters and if they are not they are displayed in red letters. Changing the conceptId, we display the remaining values until we display them all.

The HTML form uses the date that has been taken, the user who did the insertion of the data, the locationId, the current visitId, the current encounterId and the patientId to display exactly the corresponding data for each patient at that time.

```
<div class="hidden"><obs id="bpm" conceptId="163142"/></div>
<table class="ex1">
    <td><fieldset disabled="disabled">
        <legend><lookup expression="fn.getConcept('163142').name"/></legend>

        <h3 align="center" id="val26"></h3>

<script type="text/javascript">
jQuery(function() {
var kl=getValue("bpm.value");
//console.log(kl);
var red = kl.fontcolor("red");
var black = kl.fontcolor("black");
if (kl&lt;60 || kl&gt;100){
document.getElementById("val26").innerHTML = red;
}
else{
document.getElementById("val26").innerHTML = black;
}
});
</script>
</td></table>
```

Figure 5-39 Get and display values on HTML form

For the current implementation, we created an encounter called Electrocardiogram. A visit may include more than once the same encounter and the number of occurrence each encounter is met depends on the frequency that the encounter is repeated during the Visit. The encounters can be displayed or deleted if the doctor selects the corresponding fields.

For ECG visualization, we select the corresponding visit from patient's dashboard as we can see in Figure 5-40.

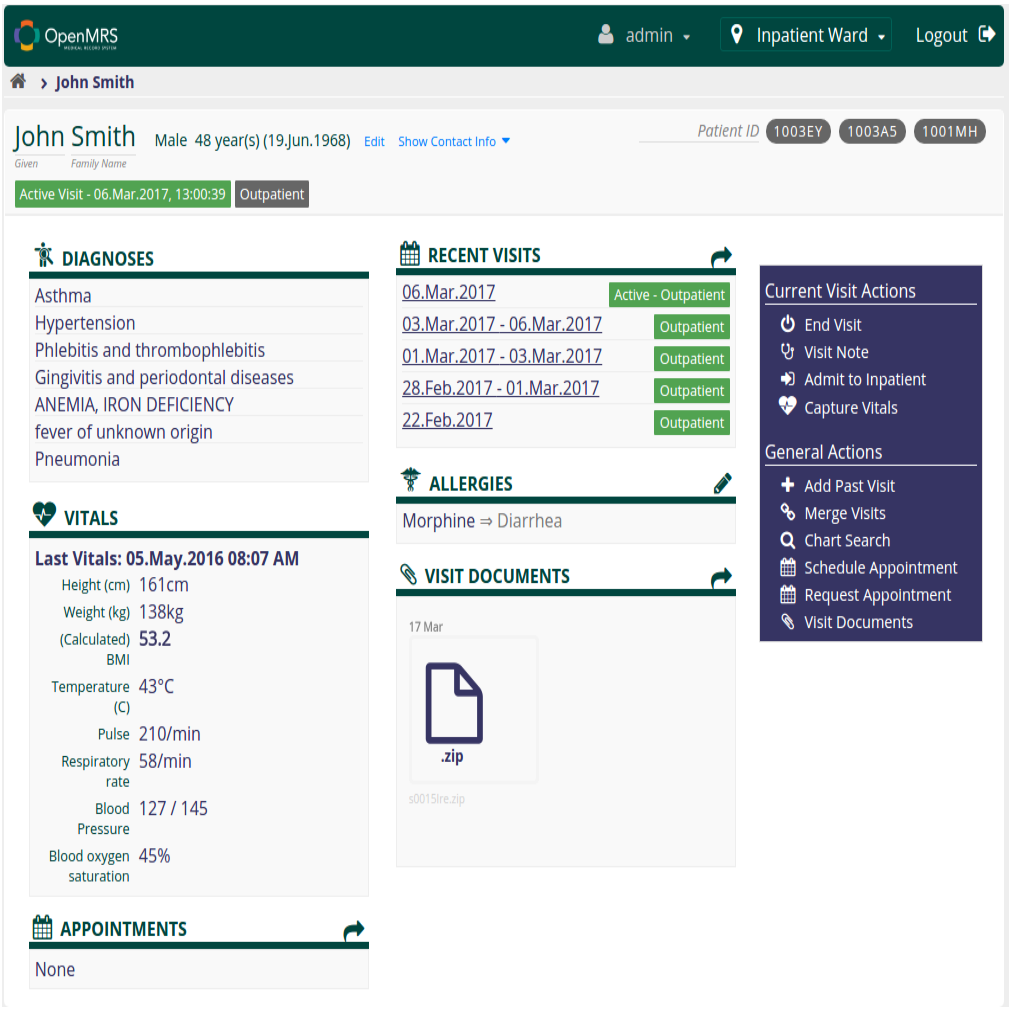


Figure 5-40 Patient's Dashboard

We can navigate into the patent's dashboard also via a mobile device as we see in Figure 5-41.

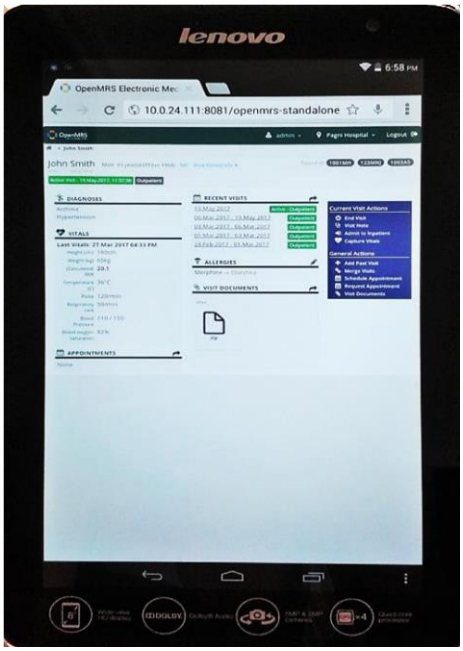


Figure 5-41 Patient's Dashboard via mobile device

Figure 5-42 depicts the encounters that a visit includes.

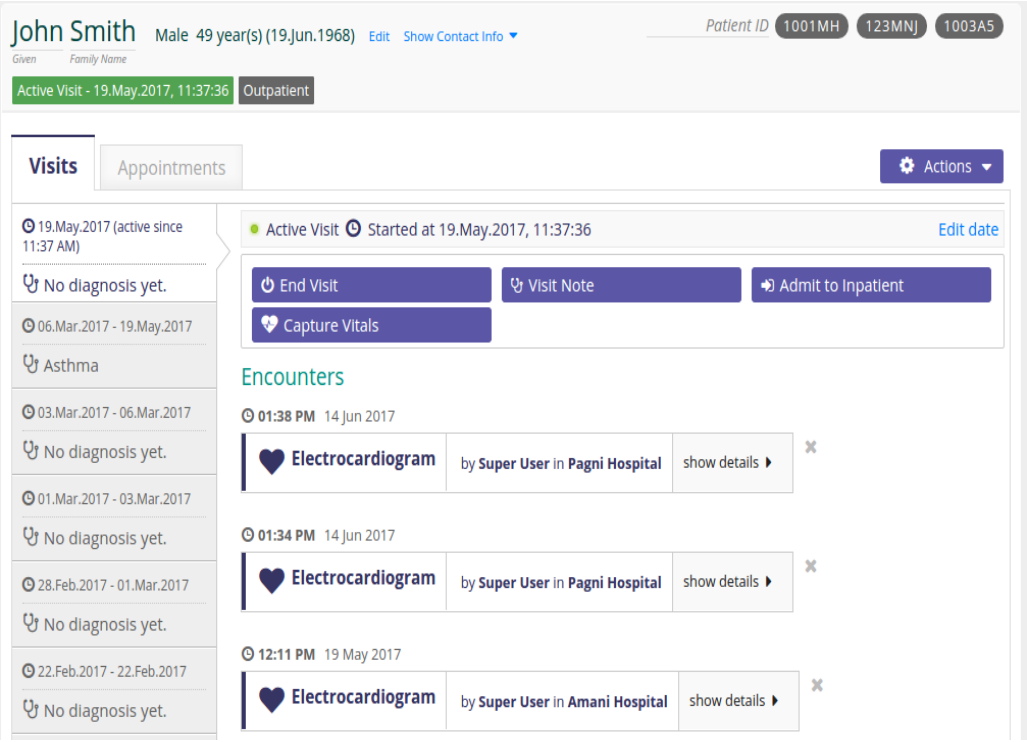


Figure 5-42 Patient's Encounters

Figure 5-43 shows an example of ECG visualization saved into OpenMRS platform.

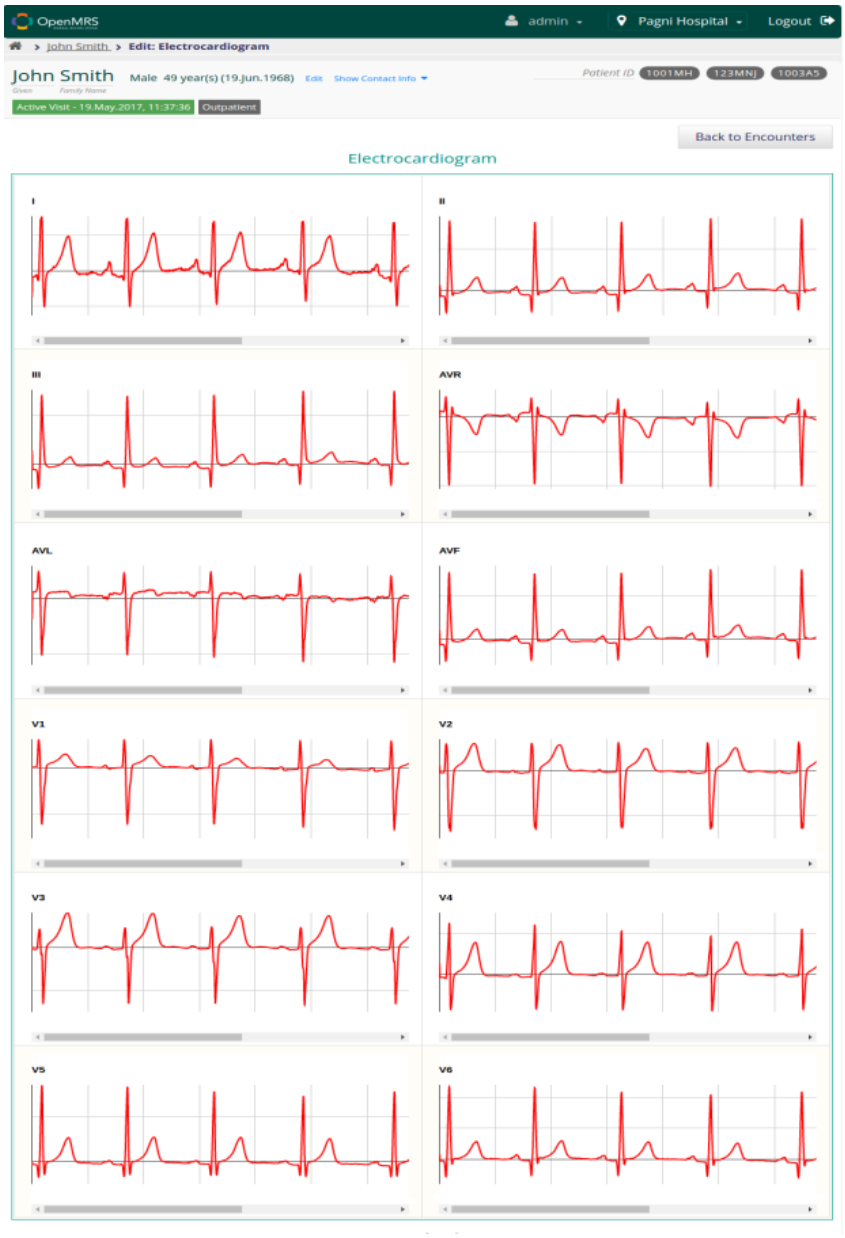


Figure 5-43 ECG Visualization

In the figure 5-44, we study an example of abnormal heart operation. The indicators in red color (R amplitude in lead I, V1 ratio, V2 ratio, V6 ratio and RR interval) are the warnings shown to the doctor that patient needs treatment.

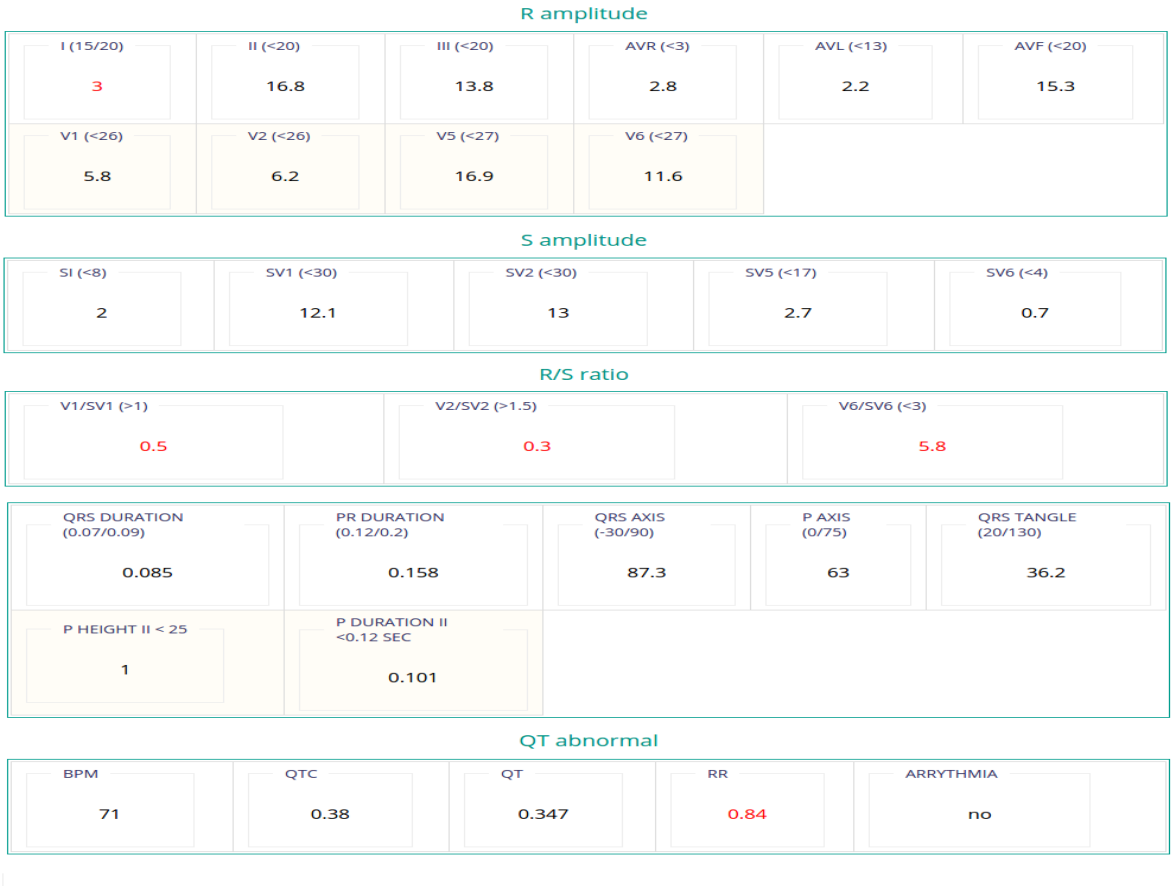


Figure 5-44 ECG metrics and warnings

ECG visualization is also feasible via a mobile device as Figure 5-45 shows

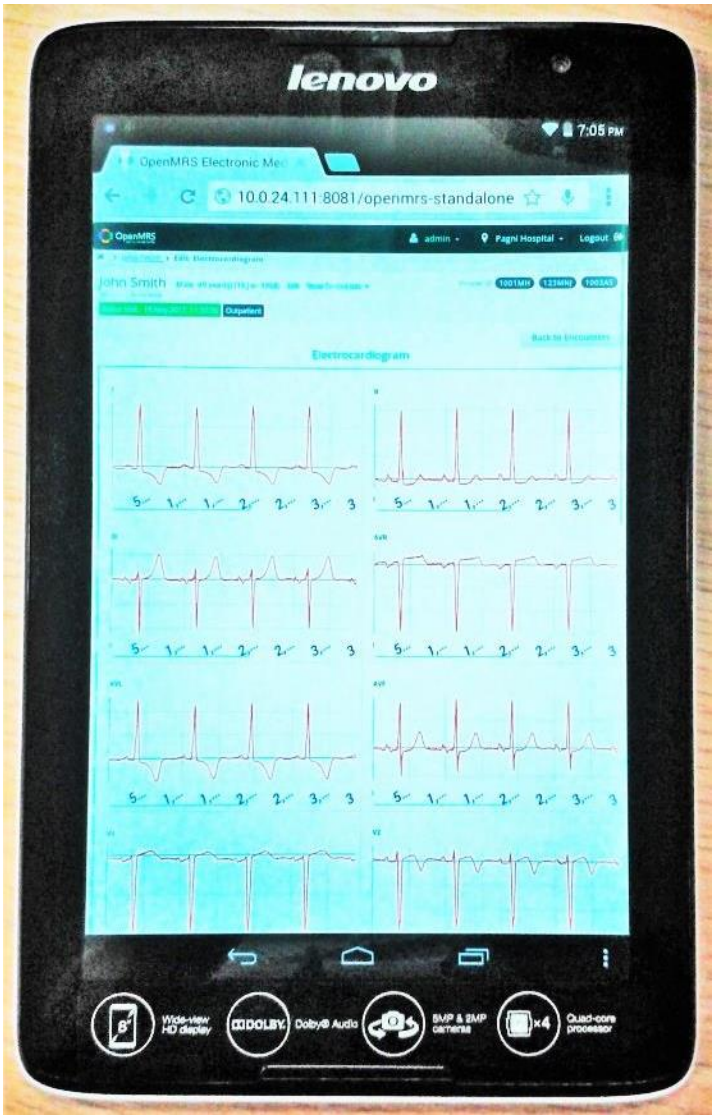


Figure 5-45 ECG Visualization in a mobile device

Chapter 6 - Experimental Evaluation

In this chapter, we will describe some tests with abnormal ECGs to verify our system's validity. For testing algorithms' accuracy, we check the ECG findings in three cases and compare them with the findings of a normal ECG. According to Mr. Germanaki's book [62], we should follow seven steps to evaluate an ECG and decide whether it is normal or pathological. The first step is to calculate the heart rate and check if it is normal. After that, we check if the P-waves in the 12 graphs precede to the QRS complexes. The next step is the intervals estimation. The fourth checkpoint concerns the amplitude values of the required waves. After that, we investigate the electrical axis values (QRS-axis, P-axis and QRS-Tangle). Additionally, we search for tachycardia and bradycardia and as last step, we make a conclusion about the type of the ECG (normal or pathological). After completing the seven steps that Mr. Germanakis mentions we will make a last check to test algorithm's validity by comparing the arithmetic values of the waves that algorithm exports with the values that we take empirically. Before finding the amplitudes of each wave, we determine if a wave is positive (above the baseline) or negative (under the baseline). If the algorithm exports a negative value for a wave and empirically we view that the wave is under the baseline, the algorithm works correctly. The experimental evaluation uses ECGs with Myocardial Infarction, Myocardial Hypertrophy and Cardiomyopathy Heart Failure. By using the graphs of the abnormal ECGs, we detect the ECG findings and we contradict them with the results that the WFDB applications and the algorithms give.

6.1 Myocardial Infarction

Firstly, we will detect the durations of the waves that we need. Randomly, the beat that we will analyze, is the second one. From Figure 6-1, we can see that the QRS complex starts about from the 1050^{th} sample and ends at approximately the 1200^{th} sample. So the duration, according to visual detection, equals to (1200-1050)*0.001=0.15 sec. As we can check from our platform, the QRS complex duration after the processing equals to 0.157 sec.

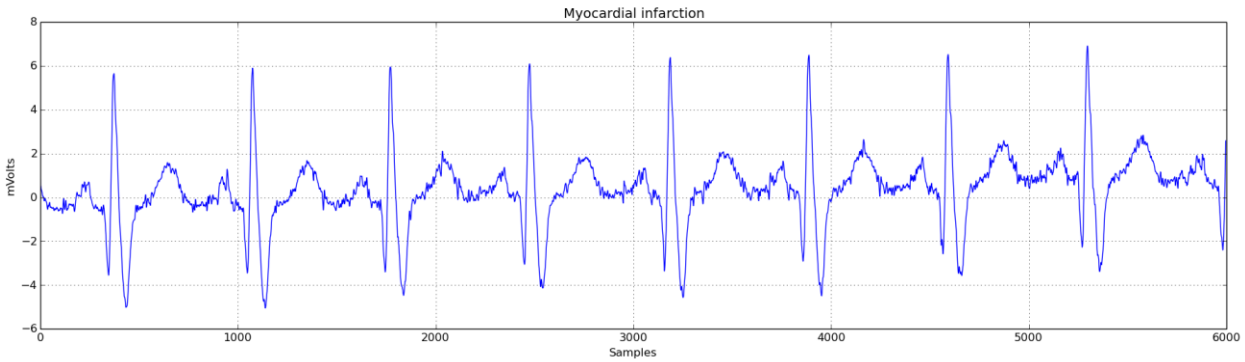


Figure 6-1 Myocardial Infarction in lead I

The second duration, which we will detect, is P-R duration and equals to about (870 - 1050)*0.001=0.18 sec. From our platform, we see that it equals to 0.15 sec. The next duration is P-wave's in lead II. From the graph, we see that it equals to (980-860)*0.001=0.12 sec and from the platform, it is 0.111 sec. After that, we will estimate the Q-T duration in order to calculate QTc. The Q-T duration is from about 1050th sample to 1500th sample so it equals to 450*0.001=0.45 sec and from our platform, the duration equals to 0.427 sec. The last duration is R-R, which starts from 1070th sample to 1750th sample. After the calculation, the R-R duration equals to 0.68 sec and from the platform it is 0.7 sec.

After estimating the durations, we have to check the amplitudes of the waves in the appropriate leads. Firstly, we will estimate the amplitudes in lead I. Figure 6-2 depicts the R-wave amplitude that equals to 5.9 mm and the S-wave amplitude that equals to 5.05 mm. Our algorithms calculate the R-wave amplitude to 5.9 mm and the S-wave amplitude to 5.1 mm.

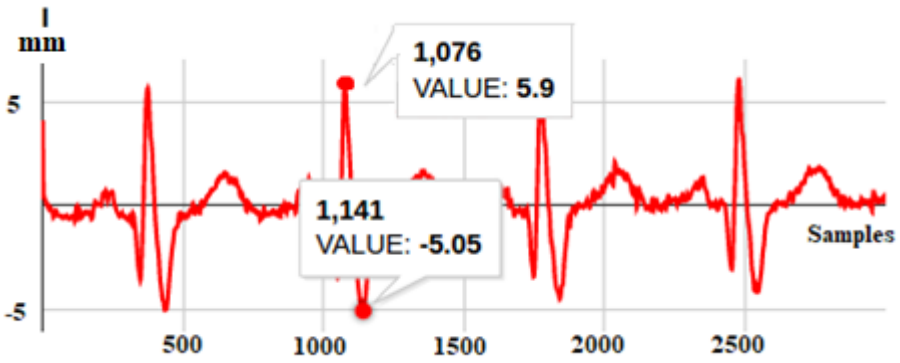


Figure 6-2 R-wave and S-wave amplitudes in lead I

The next amplitudes, which we will calculate, are the P-wave amplitude and the R-wave amplitude in lead II. From Figure 6-3, we can see that the P-wave amplitude equals to 2.81 mm and the R-wave amplitude equals to 0.72 mm. From our algorithms, these values equal to 2.8 mm and 0.7 mm respectively.

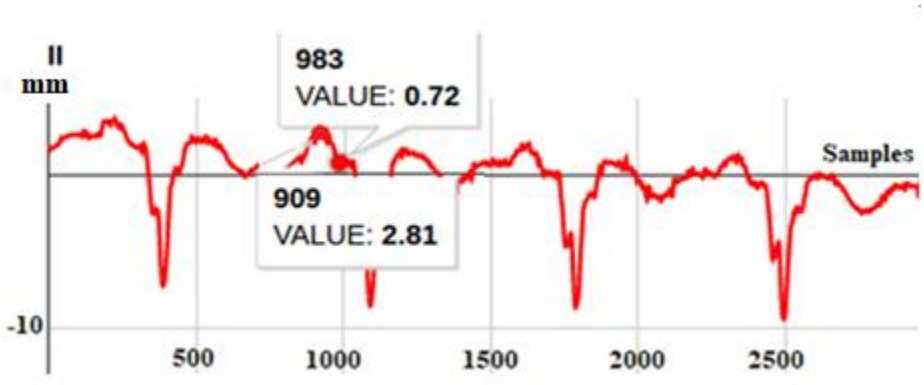


Figure 6-3 P-wave and R-wave amplitudes in lead II

The next amplitude is the R-wave amplitude in lead III. From Figure 6-4, we can see that it equals to 3.69 mm. Our algorithm calculated its value to 3.7 mm.

Figure 6-4 R- wave amplitude in lead III

The amplitude for the R-wave in lead AVR, as it is shown in Figure 6-5, equals to 3.78 mm. From our algorithm, its value is calculated to 3.8 mm. We also see that two R-wave amplitudes are appeared. In case of abnormality, there is such a phenomenon.

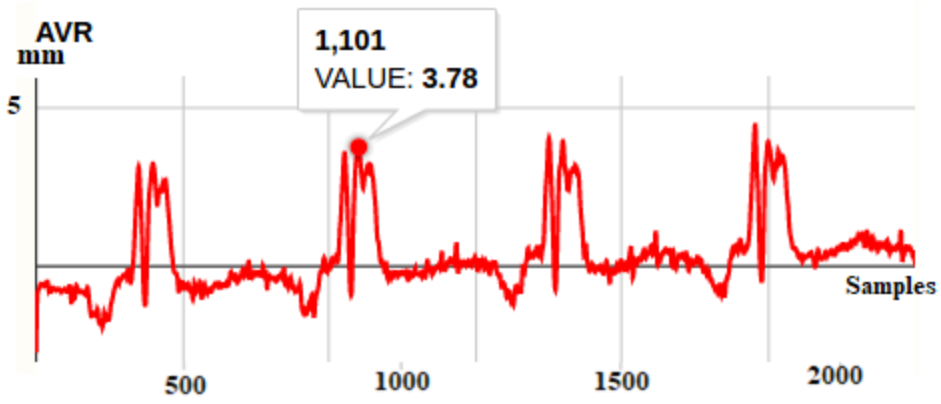


Figure 6-5 R-wave amplitude in lead AVR

The R-wave amplitude in lead AVL equals to 8.01 mm (Figure 6-6). Our algorithm calculated its value to 8 mm.

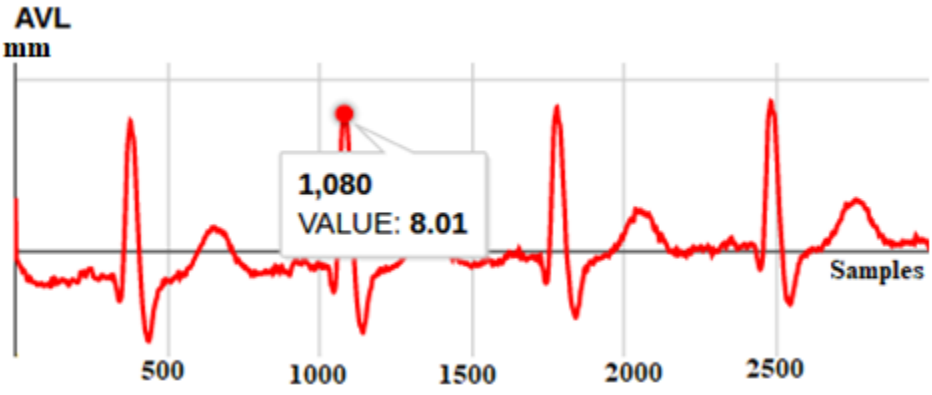


Figure 6-6 R-wave amplitude in lead AVL

The R-wave amplitude equals to 1.13 mm (Figure 6-7) and our algorithm calculated its value to 1.1 mm.

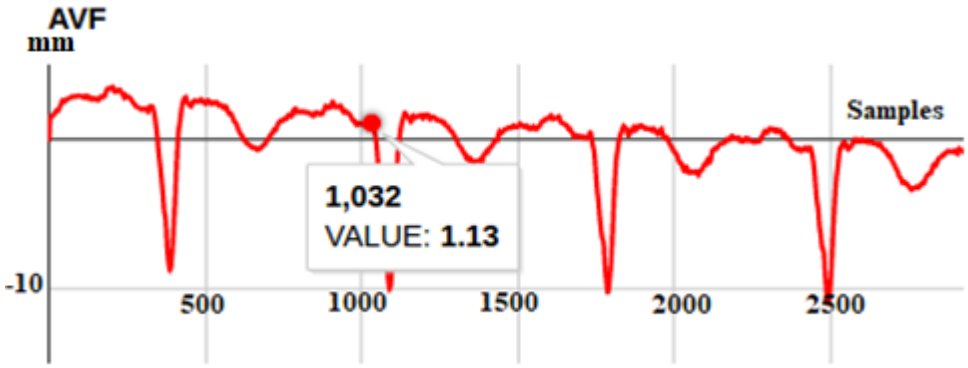


Figure 6-7 R-wave amplitude in lead AVF

The R-wave amplitude from Figure 6-8 equals to 9.72 mm and the S-wave amplitude to 0.56 mm. Our algorithm calculated these values to 9.7 mm and 0.5 mm respectively.

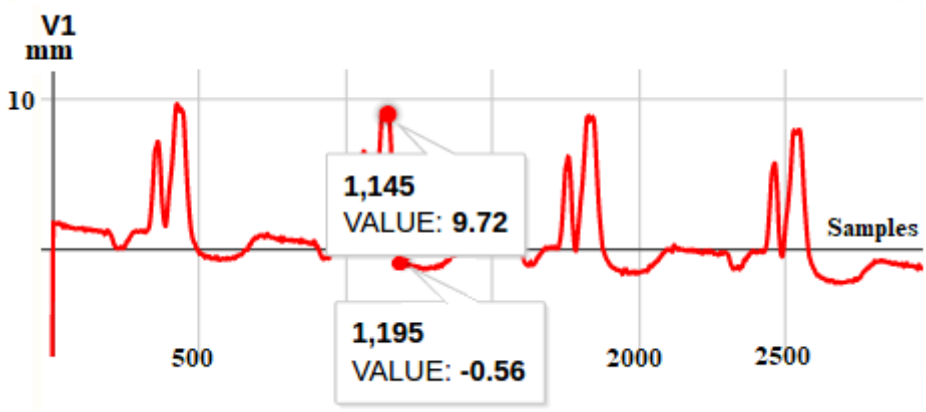


Figure 6-8 R-wave and S-wave amplitudes in lead V1

The R-wave amplitude from figure 6-9 equals to 13.98 mm and the S-wave amplitude to 0.67 mm. From our algorithms, we computed these values to 14 mm and 0.7 mm respectively.

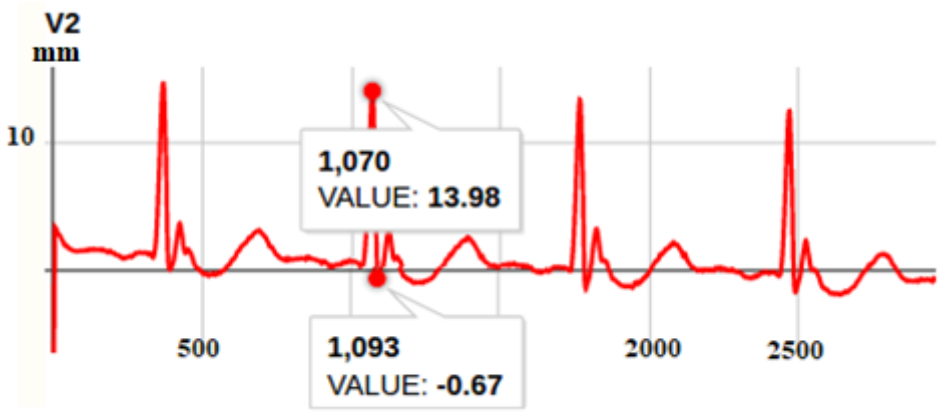


Figure 6-9 R-wave and S-wave amplitude in lead V2

The R-wave amplitude in lead V5 equals to 3.91 mm and the S-wave amplitude to 11.47 mm from Figure 6-10. The results from our algorithm are 3.9 mm and 11.5 mm respectively.

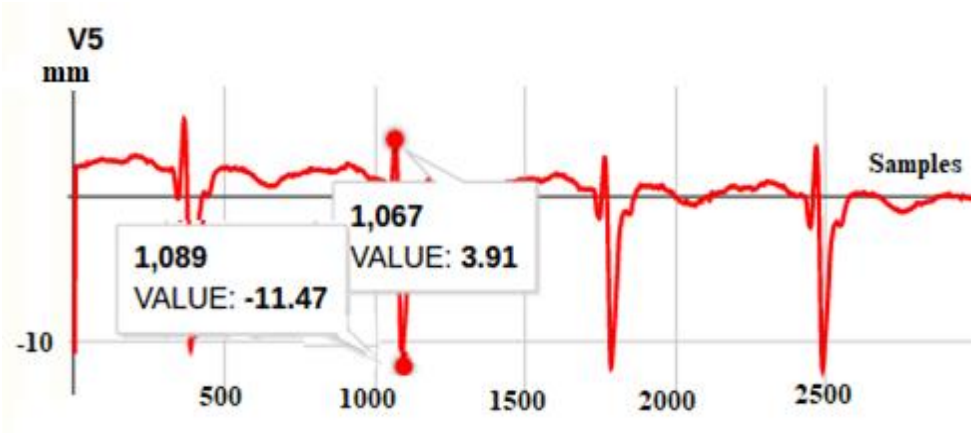


Figure 6-10 R-wave and S-wave amplitudes in lead V5

The R-wave amplitude from Figure 6-11 equals to 0.46 mm and the S-wave amplitude to 8.38 mm. The algorithm has calculated these values to 0.6 mm and 8.4 mm respectively.

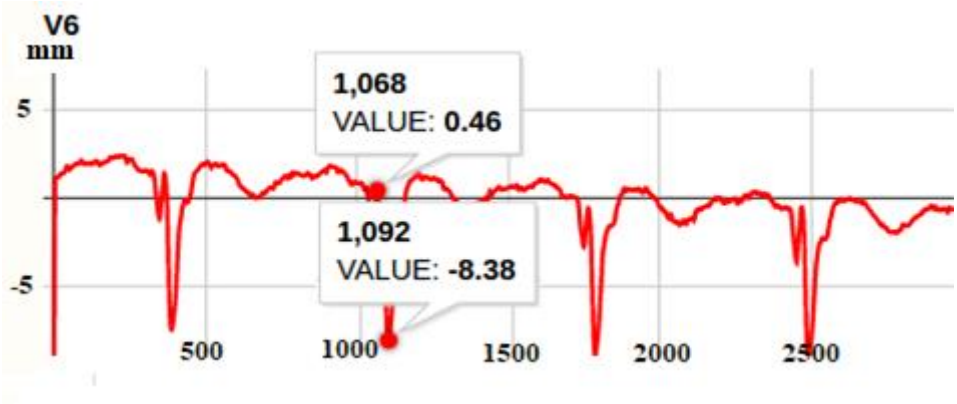


Figure 6-11 R-wave and S-wave amplitudes in lead V6

Results

In order to decide whether an ECG is normal or pathological, the seven steps that are mentioned by Mr Germanakis have to be followed. The first step is to estimate if the heart rate is normal. In the current situation, the heart rate equals to 86 beats per minute and according to the required limits, the value is normal.

The P-wave precedes to QRS complexes in 12 graphs but in some cases, they are negative instead of positive and also indiscernible. In leads I, II, III, AVR, AVL, AVF, V5 and V6 the P-waves are located correctly and are positive except in lead AVR where is correctly negative. In leads V1 and V2, the P-waves are negative instead of positive. In lead I, we can see a deep Q-wave and a deep S-wave, which are not normal. The QRS complex is negative in lead II and in lead III we can see a very deep Q. In lead AVR, we can see two R-waves and in lead AVF, we have a small R-wave and a deep S-wave. In leads V1 and V2, we also have two R waves and in leads V5 and V6, we have deep S-waves and small R-waves.

After P-waves, we have to check the required intervals. The first wave is the P-wave duration, which equals to 0.117 sec and it is normal. The P-R duration equals to 0.168 sec, which

is also normal. The QRS complex duration is 0.157 sec, the Q-T duration is 0.449 sec and the QTc is 0.54 sec. These durations are not normal but the R-R duration which equals to 0.7 sec is normal. In Table 6-1, we can see the duration values that we have calculated empirically and the calculated values from the algorithms. Therefore, we compare the results in order to find the deviation, which ranges from 0.22% to 6.66%.

Duration of the waves	**Duration values from graphs (sec)**	**Calculated values of the durations (sec)**	**Error (%)**
P-wave duration	0.12	0.117	2.5
P-R duration	0.18	0.168	6.66
QRS complex duration	0.15	0.157	4.66
Q-T duration	0.45	0.449	0.22
R-R duration	0.68	0.7	2.94

Table 6-1 Myocardial Infarction - Durations of the waves

As we can see from the previous tests, our algorithms detect the ECG waves with a deviation that ranges from 0% to 30.43% as it is summarized in Table 6-2. According to these results, we can see that some amplitude values are normal and others are abnormal. Except the amplitude of each wave, we have to check if they are positive or negative. Therefore, a combination has to be done with the results of the amplitude values and the figures for each lead.

Amplitudes in leads	**Values from graphs (mm)**	**Calculated values (mm)**	**Error (%)**
R-wave amplitude in lead I	5.9	5.9	0
S-wave amplitude in lead I	-5.05	-5.1	0.99
P-wave amplitude in lead II	2.81	2.8	0.35
R-wave amplitude in lead II	0.72	0.7	2.77
R-wave amplitude in lead III	3.69	3.7	0.27
R-wave amplitude in lead AVR	3.78	3.8	0.53
R-wave amplitude in lead AVL	8.01	8	0.12

R-wave amplitude in lead AVF	1.13	1.1	2.65
R-wave amplitude in lead V1	9.72	9.7	0.20
S-wave amplitude in lead V1	-0.56	-0.5	10.71
R-wave amplitude in lead V2	13.98	14	0.14
S-wave amplitude in lead V2	-0.67	-0.7	4.47
R-wave amplitude in lead V5	3.91	3.9	0.25
S-wave amplitude in lead V5	-11.47	-11.5	0.26
R-wave amplitude in lead V6	0.46	0.6	30.43
S-wave amplitude in lead V6	-8.38	-8.4	0.23

Table 6-2 Myocardial Infarction - Amplitudes in leads

The Table 6-3 contains the values of the waves that the algorithms export, if a wave is above or under the baseline and if the waves are correctly positive or negative.

Amplitudes in leads	**Values from algorithms**	**Position of the wave**	**Result**
R-wave amplitude in lead I	5.9	Above the baseline	correct
S-wave amplitude in lead I	-5.05	Under the baseline	correct
P-wave amplitude in lead II	2.81	Above the baseline	correct
R-wave amplitude in lead II	0.72	Above the baseline	correct
R-wave amplitude in lead III	3.69	Above the baseline	correct
R-wave amplitude in lead AVR	3.78	Above the baseline	correct
R-wave amplitude in lead AVL	8.01	Above the baseline	correct
R-wave amplitude in lead AVF	1.13	Above the baseline	correct
R-wave amplitude in lead V1	9.72	Above the baseline	correct
S-wave amplitude in lead V1	-0.56	Under the baseline	correct
R-wave amplitude in lead V2	13.98	Above the baseline	correct
S-wave amplitude in lead V2	-0.67	Under the baseline	correct
R-wave amplitude in lead V5	3.91	Above the baseline	correct
S-wave amplitude in lead V5	-11.47	Under the baseline	correct

R-wave amplitude in lead V6	0.46	Above the baseline	correct
S-wave amplitude in lead V6	-8.38	Under the baseline	correct

Table 6-3 Check position of the waves in myocardial infarction

As we can see from the Table 6-3, the algorithms define the position of the waves correctly. The R-wave amplitudes and the P-wave amplitude in the required leads are correctly positive and the S-wave amplitudes are correctly negative.

The QRS-axis equals to -106.6^{o}, the P-axis equals to 65.5^{o} and the QRS-Tangle equals to 145.2^{o}. From the axis values, only the P-axis has normal value. The last check is if there is tachycardia or bradycardia. In the current experiment the beats per minute are 86 so, there is no tachycardia or bradycardia.

Summarizing, we can see that many values are not normal and according to the figures, some graphs are abnormal too. Therefore, our algorithms highlight this ECG as pathological, confirming thus its abnormal nature, since it has been taken from the Physiobank database of ECGs with myocardial infarction.

6.2 Myocardial Hypertrophy

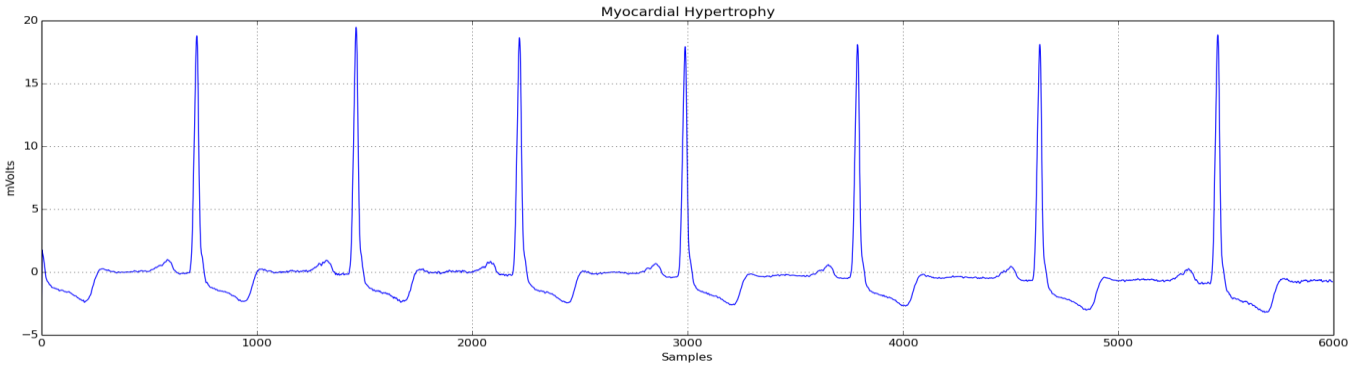


Figure 6-12 Myocardial Hypertrophy in lead I

Similarly, as we did with myocardial infarction, we will first estimate the ECG durations that we need. The first duration is the QRS complex duration. From Figure 6-12, we calculate the QRS complex duration that starts about from the 690th sample and ends at the 760th sample. So the duration equals to (760-690)*0.001= 0.07 sec. From our algorithms, the value is calculated to 0.072 sec.

The P-R duration equals to (670-530)*0.001=0.14 sec and our algorithm calculated the value to 0.127 sec. The P-wave duration in lead II equals to (1360-1230)*0.001=0.13 sec and from our algorithms, we take that the value equals to 0.127 sec. The next duration is the Q-T duration, which equals to (1810-1430)*0.001=0.38 sec and from our algorithms, we take that this value equals to 0.334 sec. The last value is the R-R interval. From the graph we take that the value equals to (2220-1460)*0.001=0.76 sec and the calculation from our algorithm equals it to 0.77 sec.

Having estimated the durations of the waves, we will calculate the amplitudes. From Figure 6-13 we can see that the R-wave amplitude in lead I equals to 18.59 mm and that the S-wave amplitude equals to 0.73 mm. The results from our algorithm equal them to 18.8 mm and 0.9 mm respectively.

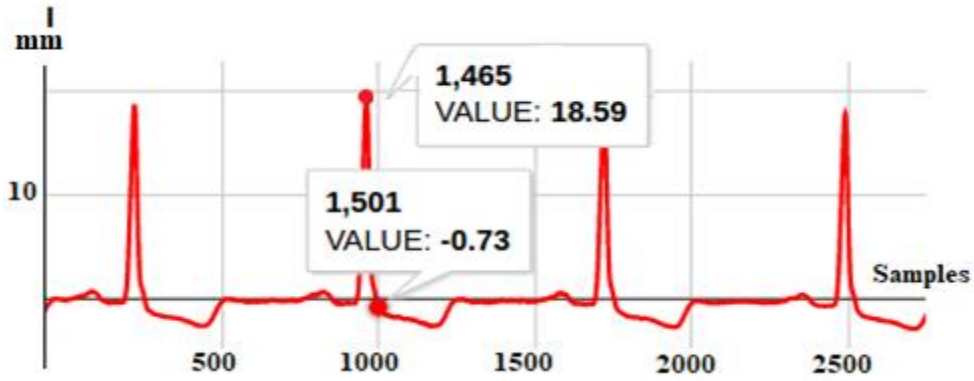


Figure 6-13 R-wave and S-wave amplitudes in lead I

The R-wave amplitude in lead II equals to 13.74 mm and the P-wave amplitude to 1.55 mm (Figure 6-14). After calculating, we determine these values to 13.7 mm for the R-wave amplitude and to 1.5 mm for the P-wave amplitude.

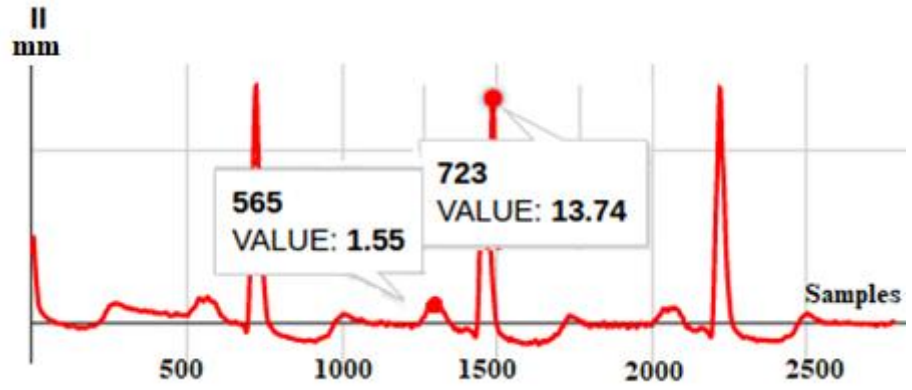


Figure 6-14 R-wave and P-wave amplitudes n lead II

The R-wave amplitude in lead III equals to 3.04 mm (Figure 6-15) and after the calculation by our algorithms, we take its value to 3 mm.

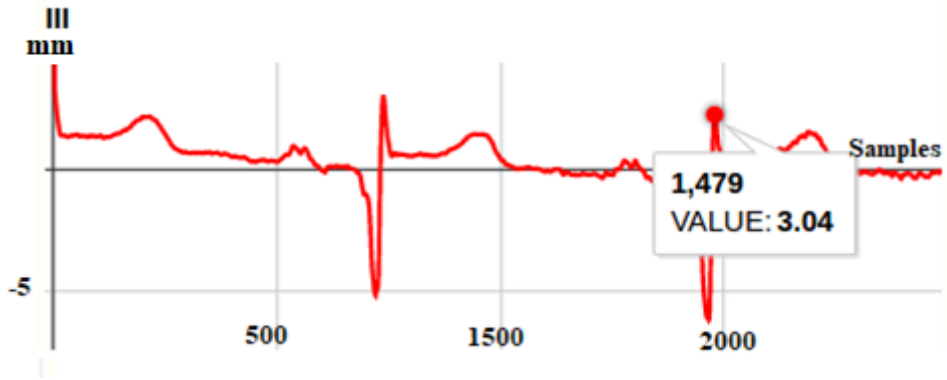


Figure 6-15 R-wave amplitude in lead III

The R-wave amplitude as we can see from Figure 6-16 equals to 0.21 mm and our algorithm calculates its value to 0.2 mm.

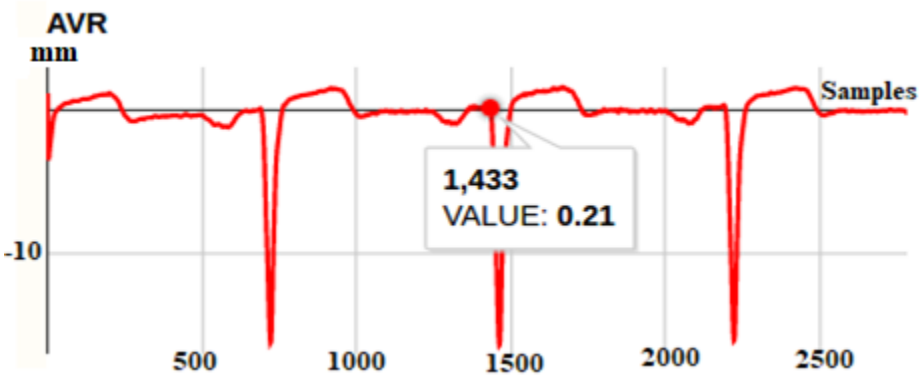


Figure 6-16 R-wave amplitude in lead AVR

The R-wave amplitude in lead AVL equals to 12.18 mm (Figure 6-17) and after calculating it its value equals to 12.2 mm.

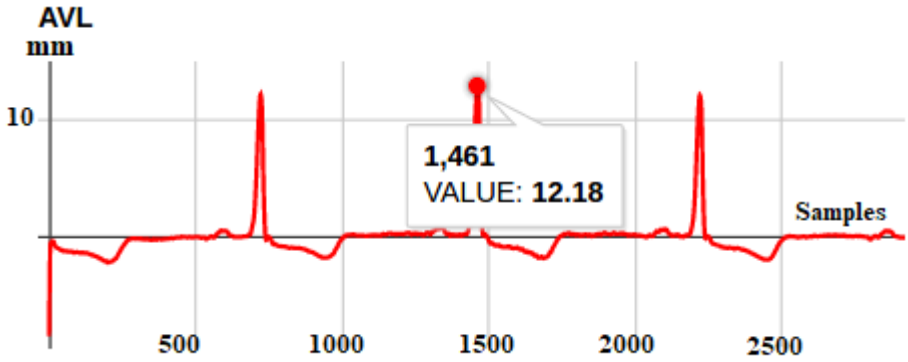


Figure 6-17 R-wave amplitude in lead AVL

The R-wave amplitude in lead AVF is 5.1 mm (Figure 6-18) and by using our algorithms a value equals to 5.1 mm is calculated.

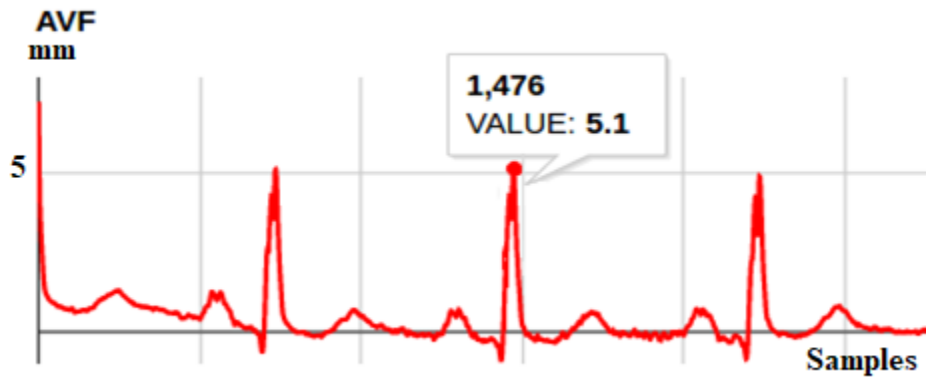


Figure 6-18 R-wave amplitude in lead AVF

The R-wave amplitude in lead V1 equals to 4.26 mm and the S-wave amplitude to 27.5 mm (Figure 6-19). After the execution of the algorithm, values equal to 4.3 mm and to 27.6 mm are calculated respectively.

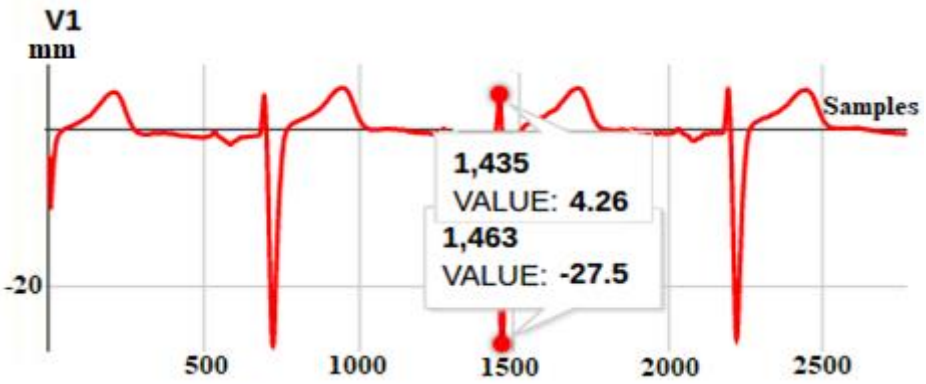


Figure 6-19 R-wave and S-wave amplitudes in lead V1

The R-wave amplitude in lead V2 equals to 7.47 mm and the S-wave amplitude to 15.23 mm (Figure 6-20). From our algorithms, we calculated the values to 7.6 mm for the R-wave amplitude and to 15.4 mm for the S-wave amplitude.

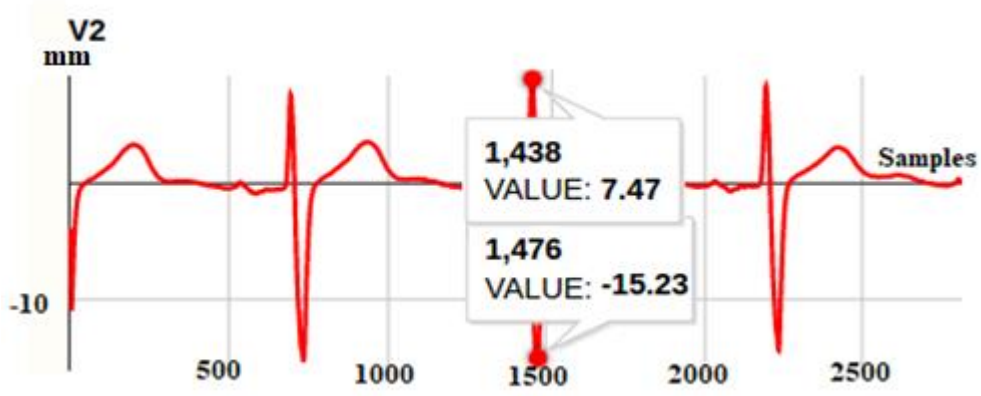


Figure 6-20 R-wave and S-wave amplitudes in lead V2

The R-wave amplitude as we can see from Figure 6-21 equals to 18.5 mm and the S-wave amplitude equals to 0.18 mm. After the execution of algorithms, the values are calculated equal to 18.7 mm and to 0.12 mm respectively.

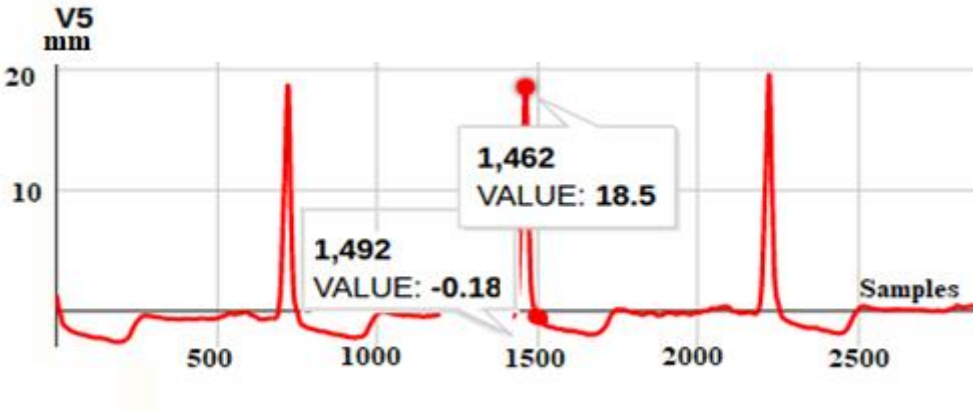


Figure 6-21 R-wave and S-wave amplitudes in lead V5

The R-wave amplitude in lead V6 equals to 15.42 mm and the S-wave amplitude to 0.19 mm (Figure 6-22). The algorithm calculated these results to 15.4 mm for the R-wave amplitude and to 0.2 mm for the S-wave amplitude.

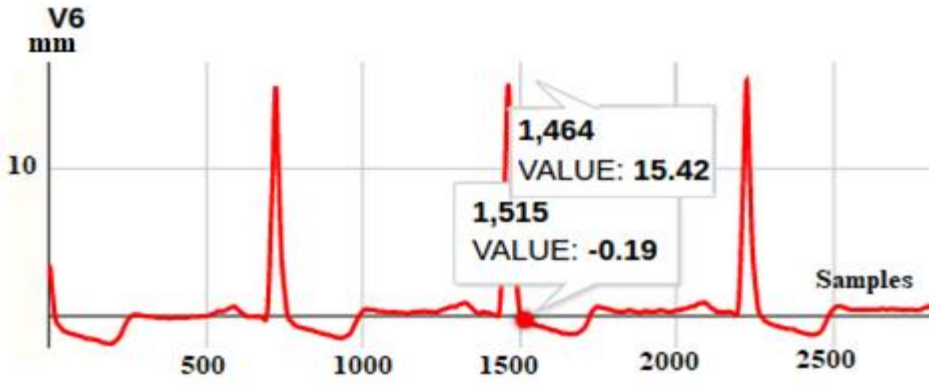


Figure 6-22 R-wave and S-wave amplitudes in lead V6

Results

The heart rate equals to 78 beats per minute, which is a normal value. The P-waves are preceded to QRS complexes in 12 graphs and they are positive in all leads except in AVR, which is correctly negative. In lead III, the S-wave amplitude is almost twice the R-wave amplitude, which is not normal.

The Q-T interval, which equals to 0.334 sec and the QTc, which equals to 0.39 sec are abnormal values. The rest intervals are between the normal limits. The duration of the waves are summarized in the Table 6-4 in order to estimate the deviation, which ranges from 1.31% to 12.10%.

Duration of the waves	**Duration values from graphs (sec)**	**Calculated values of the durations (sec)**	**Error (%)**
P-wave duration	0.13	0.127	2.3
P-R duration	0.14	0.127	9.28

QRS complex duration	0.07	0.072	2.85
Q-T duration	0.38	0.334	12.10
R-R duration	0.76	0.77	1.31

Table 6-4 Myocardial Hypertrophy - Durations of the waves

From the previous processing, we can see that our amplitude calculations have a small deviation that ranges from 0% to 33.33% as it is summarized in Table 6-5, but the location of the waves is correct. Comparing the calculated amplitudes with the limits of a normal ECG, we find that some values are not normal.

Amplitudes in leads	**Values from graphs (mm)**	**Calculated values (mm)**	**Error (%)**
R-wave amplitude in lead I	18.59	18.8	1.12
S-wave amplitude in lead I	-0.73	-0.9	23.28
P-wave amplitude in lead II	1.55	1.5	3.22
R-wave amplitude in lead II	13.74	13.7	0.29
R-wave amplitude in lead III	3.04	3	1.31
R-wave amplitude in lead AVR	0.21	0.2	4.76
R-wave amplitude in lead AVL	12.18	12.2	0.16
R-wave amplitude in lead AVF	5.1	5.1	0
R-wave amplitude in lead V1	4.26	4.3	0.93
S-wave amplitude in lead V1	-27.5	-27.6	0.36
R-wave amplitude in lead V2	7.47	7.6	1.74
S-wave amplitude in lead V2	-15.23	-15.4	1.11
R-wave amplitude in lead V5	18.5	18.7	1.08
S-wave amplitude in lead V5	-0.18	-0.12	33.33
R-wave amplitude in lead V6	15.42	15.4	0.12
S-wave amplitude in lead V6	-0.19	-0.2	5.26

Table 6-5 Myocardial Hypertrophy - Amplitudes in leads

In Table 6-6, we can see the values of the waves that the algorithms export, the position of each wave and if the waves are correctly positive or negative.

Amplitudes in leads	Values from algorithms	Position of the wave	Result
R-wave amplitude in lead I	18.59	Above the baseline	correct
S-wave amplitude in lead I	-0.73	Under the baseline	correct
P-wave amplitude in lead II	1.55	Above the baseline	correct
R-wave amplitude in lead II	13.74	Above the baseline	correct
R-wave amplitude in lead III	3.04	Above the baseline	correct
R-wave amplitude in lead AVR	0.21	Above the baseline	correct
R-wave amplitude in lead AVL	12.18	Above the baseline	correct
R-wave amplitude in lead AVF	5.1	Above the baseline	correct
R-wave amplitude in lead V1	4.26	Above the baseline	correct
S-wave amplitude in lead V1	-27.5	Under the baseline	correct
R-wave amplitude in lead V2	7.47	Above the baseline	correct
S-wave amplitude in lead V2	-15.23	Under the baseline	correct
R-wave amplitude in lead V5	18.5	Above the baseline	correct
S-wave amplitude in lead V5	-0.18	Under the baseline	correct
R-wave amplitude in lead V6	15.42	Above the baseline	correct
S-wave amplitude in lead V6	-0.19	Under the baseline	correct

Table 6-6 Check position of the waves in myocardial hypertrophy

As we can see from the Table 6-6, the R-wave amplitudes and the P-wave amplitude in the required leads are correctly positive and the S-wave amplitudes are correctly negative.

The QRS-axis equals to -38.1^{o}, which is out of limits. The P-axis equals to 47.5^{o} and the QRS-Tangle equals to 122.3^{o}. Both values are normal. Lastly, we have to check for tachycardia or bradycardia. The heart rate equals to 78, which is a normal value.

In this case, we have more normal values than in the previous experiment but we also have abnormal values, which we have to take into account. Therefore, our algorithms highlight

this ECG as pathological, confirming thus its abnormal nature, since it has been taken from the Physiobank database of ECGs with myocardial hypertrophy.

6.3 Cardiomyopathy Heart Failure

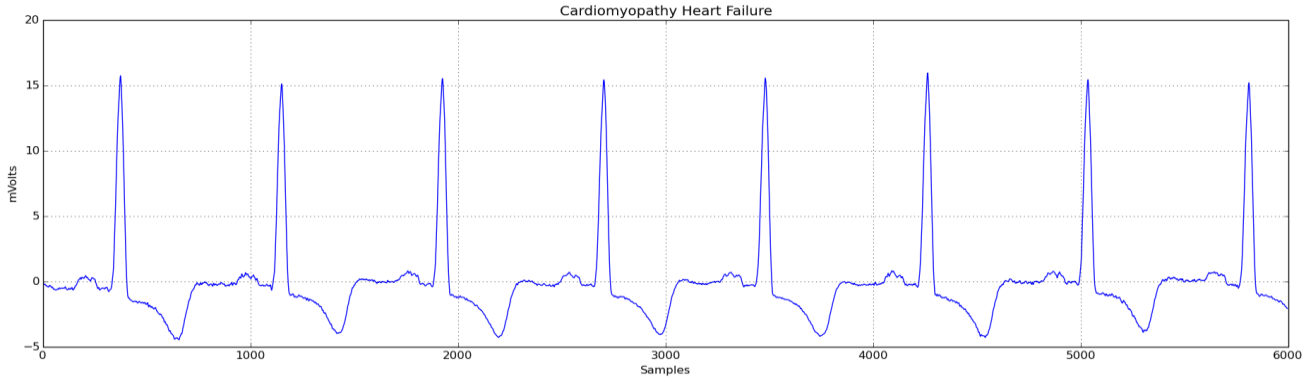


Figure 6-23 Cardiomyopathy Heart Failure in lead I

The next experiment that we will analyze is the cardiomyopathy heart failure. Firstly, we will estimate the durations of the waves and after that, their amplitudes. The QRS complex duration in lead I (Figure 6-23) equals to (1190-1100)*0.001=0.9 sec. After the calculation with our algorithm, the QRS duration equals to 0.89 sec.

The next duration that we will determine is the P-R duration in lead I. From Figure 6-23, we can see that it equals to (1100-940)*0.001=0.16 sec. Our algorithms calculated its value to 0.172 sec. The P-wave duration in lead II equals to (1030-940)*0.001=0.09 sec and from our algorithms, we have its value to 0.083 sec. The next duration is the Q-T duration, which equals to (1500-1100)*0.001=0.4 sec and from algorithms, we have its value 0.413 sec. The last duration that we have to calculate is R-R duration in lead I, which equals to (1930-1150)*0.001=0.78 sec and from algorithms, we have the same value that is 0.78 sec.

After estimating the durations, we have to calculate the amplitudes of the waveforms. The R-wave amplitude in lead I as we can see in Figure 6-24 equals to 15.1 mm and the S-wave amplitude equals to 0.99 mm. The algorithm calculates the value to 15.1 mm for the R-wave amplitude and to 0.9 mm for the S-wave amplitude.

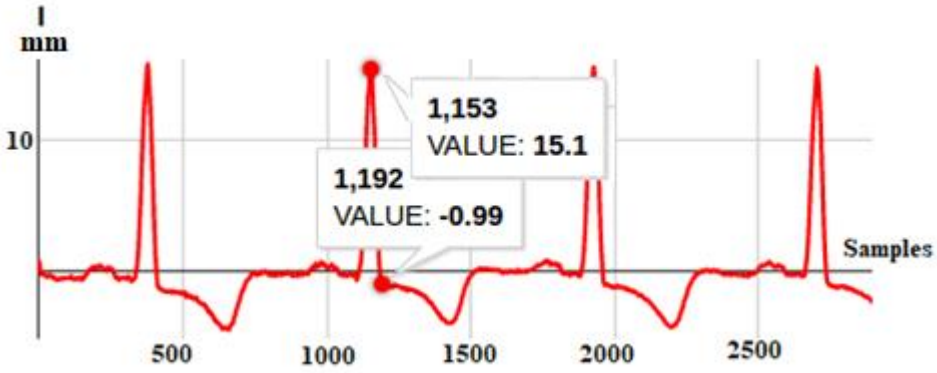


Figure 6-24 R-wave and S-wave amplitudes in lead I

The R-wave amplitude in lead II equals to 15.38 mm and the algorithm calculates it to 15.4 mm (Figure 6-25). The P-wave amplitude equals to 1.03 mm and after the processing, we calculated it to 0.9 mm.

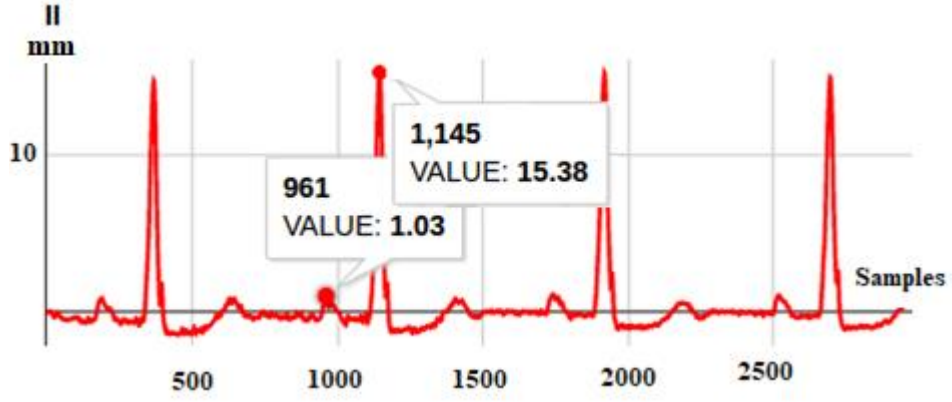


Figure 6-25 R-wave and P-wave amplitudes in lead II

The next amplitude we need to estimate is the R-wave amplitude in lead III, which equals to 2.23 mm (Figure 6-26), and our algorithms calculated its value to 2.2 mm.

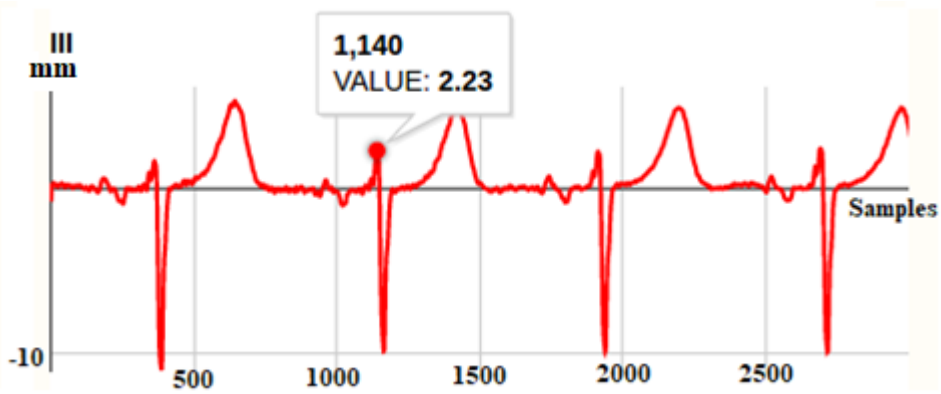


Figure 6-26 R-wave amplitude in lead III

The R-wave amplitude in lead AVR equals to 0.26 mm (Figure 6-27) and our algorithms calculated its value to 0.3 mm.

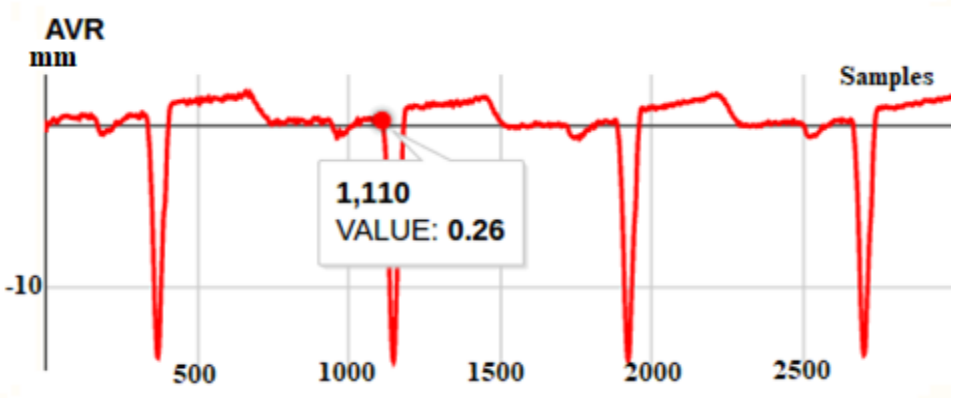


Figure 6-27 R-wave amplitude in lead AVR

The R-wave amplitude in lead AVL equals to 11.17 mm (Figure 6-28) and the execution of our algorithm calculates it to 11.2 mm.

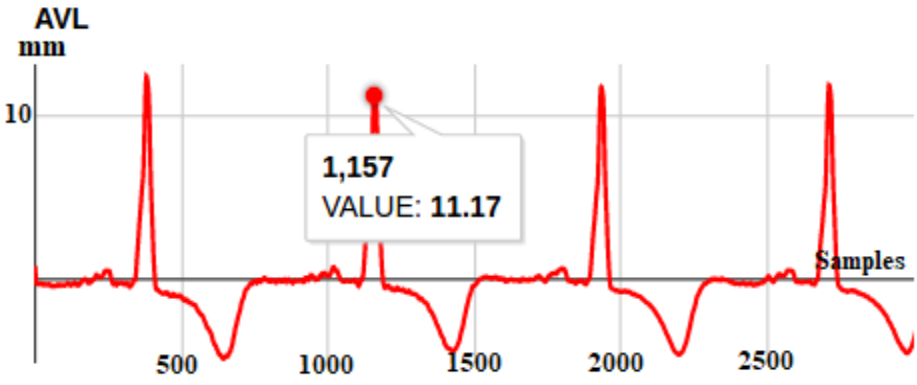


Figure 6-28 R-wave amplitude in lead AVL

As we can see from Figure 6-29, the R-wave amplitude in lead AVF equals to 8.71 mm and the processing with our algorithms calculates it to 8.7 mm.

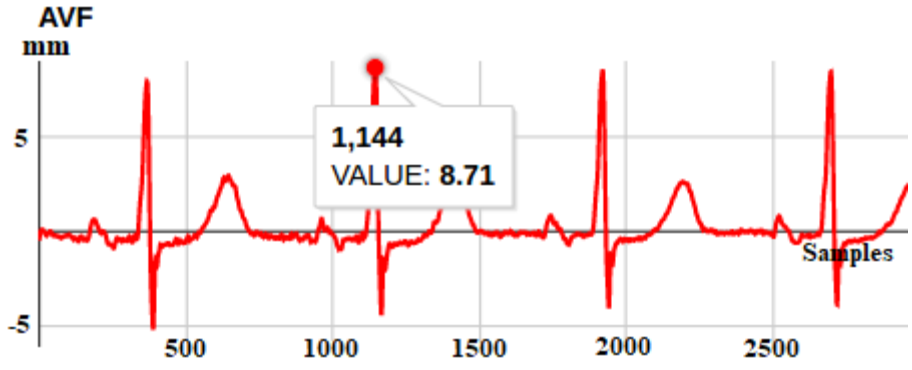


Figure 6-29 R-wave amplitude in lead AVF

The R-wave amplitude in lead V1 as we can see from Figure 6-30 equals to 0.31 mm and the S-wave value equals to 19.47 mm. After the execution of our algorithms, we take values equal to 0.3 mm and to 19.5 mm respectively.

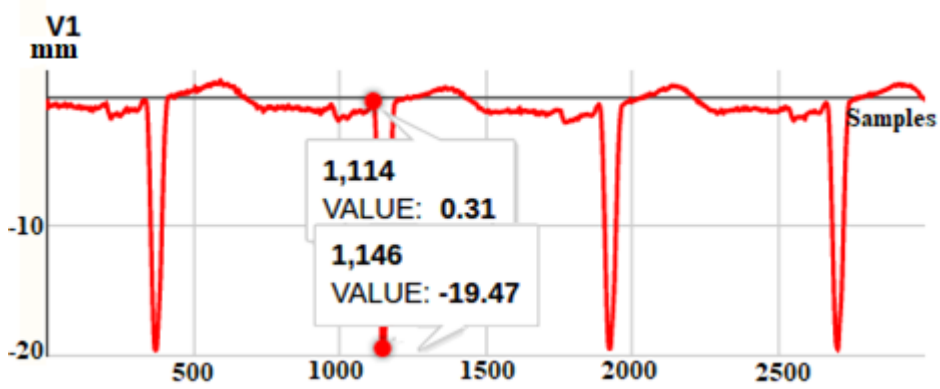


Figure 6-30 R-wave and S-wave amplitudes in lead V1

The next waves we have to estimate are the R-wave amplitude in lead V2 that equals to 2.19 mm and the S-wave amplitude, which equals to 13.88 mm (Figure 6-31). The processing of the signal calculated them to 2.2 mm and to 13.9 mm respectively.

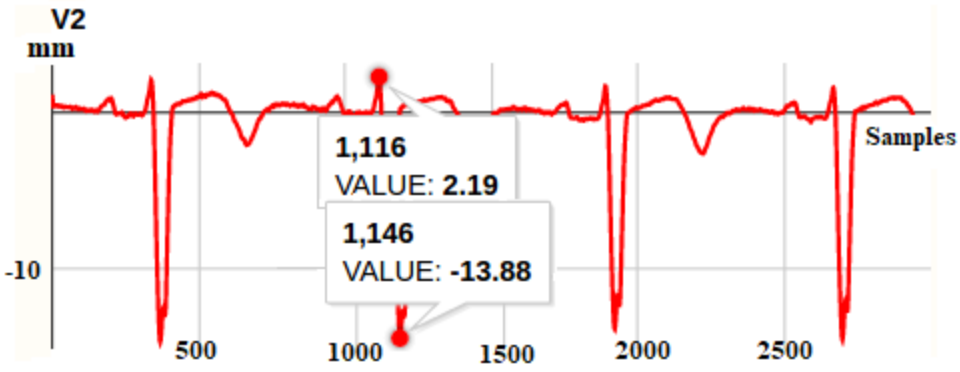


Figure 6-31 R-wave and S-wave amplitudes in lead V2

The R-wave amplitude in lead V5 equals to 23.61 mm and the S-wave amplitude equals to 0.71 mm (Figure 6-32). After the execution of our algorithms, we have values equal to 23.6 mm and to 0.7 mm respectively.

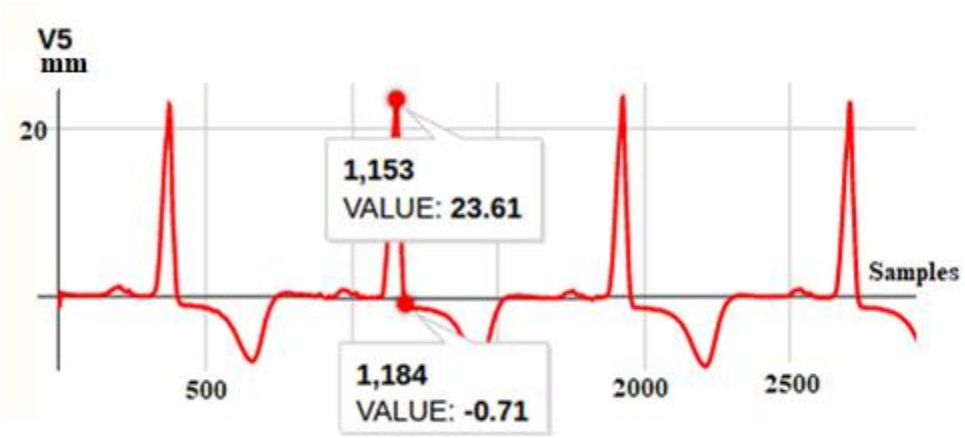


Figure 6-32 R-wave and S-wave amplitudes in lead V5

The R-wave amplitude in lead V6, as we can see in Figure 6-33, equals to 20.46 mm and the S-wave amplitude equals to 0.85 mm. Our algorithms calculate them to 20.5 mm and to 0.8 mm respectively.

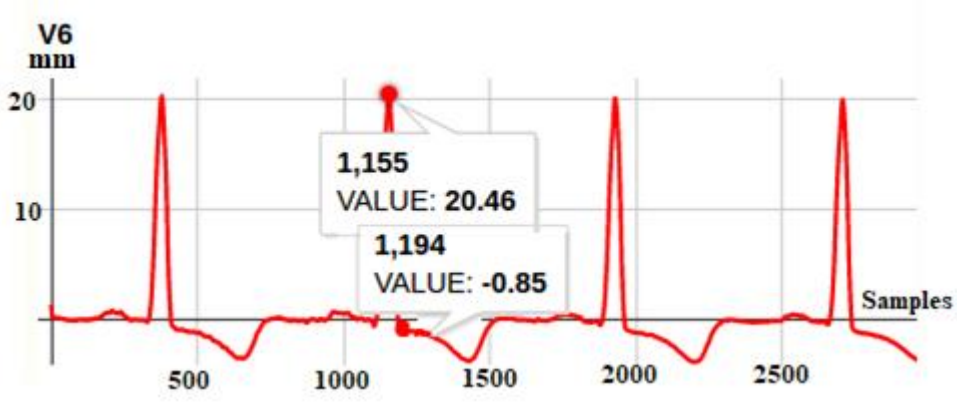


Figure 6-33 R-wave and S-wave amplitudes in lead V6

Results

The heart rate equals to 77 beats per minute, which is a normal value. The P-waves are preceded to QRS complexes in 12 graphs and they are correctly positive in all leads except in lead AVR, which is correctly negative.

In leads III, V1 and V2, a small R-wave and a very deep S-wave are displayed. In leads I, AVL, V2, V5 and V6, an inverted T-wave appears. The T-wave should be positive in all leads except in AVR.

The P-wave duration equals to 0.083 sec, the P-R interval equals to 0.172 sec, the QRS complex duration equals to 0.89 sec, the Q-T interval equals to 0.413 sec and the R-R interval equals to 0.78 sec. These values are normal except the QTc, which equals to 0.47 sec and it is out of normal limits. The duration of the waves are summarized in the Table 6-7 in order to estimate the deviation, which ranges from 0% to 7.77%.

Duration of the waves	**Duration values from graphs (sec)**	**Calculated values of the durations (sec)**	**Error (%)**
P-wave duration	0.09	0.083	7.77
P-R duration	0.16	0.172	7.5
QRS complex duration	0.9	0.89	1.11
Q-T duration	0.4	0.413	3.25
R-R duration	0.78	0.78	0

Table 6-7 Cardiomyopathy Heart Failure - Durations of the waves

With regard to the amplitudes of the ECG waves, we can see in Table 6-8 that our calculations present deviation that ranges from 0.04% to 15.38%. Comparing the R-wave amplitudes and the S-wave amplitudes in all leads with the normal limits, we can see that some values are out of limits.

Amplitudes in leads	Values from graphs (mm)	Calculated values (mm)	Error (%)
R-wave amplitude in lead I	15.1	15.1	0
S-wave amplitude in lead I	-0.99	-0.9	9.09
P-wave amplitude in lead II	1.03	0.9	12.62
R-wave amplitude in lead II	15.38	15.4	0.13
R-wave amplitude in lead III	2.23	2.2	1.34
R-wave amplitude in lead AVR	0.26	0.3	15.38
R-wave amplitude in lead AVL	11.17	11.2	0.26
R-wave amplitude in lead AVF	8.71	8.7	0.11
R-wave amplitude in lead V1	0.31	0.3	3.22
S-wave amplitude in lead V1	-19.47	-19.5	0.15
R-wave amplitude in lead V2	2.19	2.2	0.45
S-wave amplitude in lead V2	-13.88	-13.9	0.14
R-wave amplitude in lead V5	23.61	23.6	0.04
S-wave amplitude in lead V5	-0.71	-0.7	1.40
R-wave amplitude in lead V6	20.46	20.5	0.19
S-wave amplitude in lead V6	-0.85	-0.8	5.88

Table 6-8 Cardiomyopathy Heart Failure - Amplitudes in leads

In Table 6-9, we can see the arithmetic values of the waves that the algorithms export, if a wave is positive or negative wave and if the result is correct.

Amplitudes in leads	Values from algorithms	Position of the wave	Result
R-wave amplitude in lead I	15.1	Above the baseline	correct
S-wave amplitude in lead I	-0.99	Under the baseline	correct
P-wave amplitude in lead II	1.03	Above the baseline	correct
R-wave amplitude in lead II	15.38	Above the baseline	correct
R-wave amplitude in lead III	2.23	Above the baseline	correct

R-wave amplitude in lead AVR	0.26	Above the baseline	correct
R-wave amplitude in lead AVL	11.17	Above the baseline	correct
R-wave amplitude in lead AVF	8.71	Above the baseline	correct
R-wave amplitude in lead V1	0.31	Above the baseline	correct
S-wave amplitude in lead V1	-19.47	Under the baseline	correct
R-wave amplitude in lead V2	2.19	Above the baseline	correct
S-wave amplitude in lead V2	-13.88	Under the baseline	correct
R-wave amplitude in lead V5	23.61	Above the baseline	correct
S-wave amplitude in lead V5	-0.71	Under the baseline	correct
R-wave amplitude in lead V6	20.46	Above the baseline	correct
S-wave amplitude in lead V6	-0.85	Under the baseline	correct

Table 6-9 Check position of the waves in cardiomyopathy heart failure

As we can see from the Table 6-9, the R-wave amplitudes and the P-wave amplitude in the required leads are correctly positive and the S-wave amplitudes are correctly negative.

The QRS axis equals to -51.6^{o}, the P-axis equals to -36.6^{o} and the QRS-Tangle equals to 141.6^{o}. Neither of these values is normal. Lastly, we check if there is tachycardia or bradycardia. The heart rate equals to 77 and this is a normal value.

Concluding, we can see that many values are out of limits, which means that this ECG signal is not normal. This is confirmed because we used an ECG from the PhysioBank with cardiomyopathy heart failure.

Chapter 7 - Conclusion and Future Work

Due to the technological development during recent decades, new achievements are enabled into medical research area. IoT and computationally powerful mobile devices are assigned to execute operations that initially belonged to doctor's responsibilities. This evolution amplifies the role of healthcare in recent medical system. Using healthcare application allows doctors to save time and also have access to patients' health data without requiring physical presence into hospitals.

In the present work, we focus on the design and implementation of a health information system related to cardiovascular diseases. In more details, we describe a system that uses electrocardiogram (ECG) from a patient as input. Our system analyzes this signal and evaluates some metrics that help for an accurate diagnosis against heart diseases. Apart from the processing part, our system transmits this information to a centralized database using an open source platform called OpenMRS and makes it accessible to the doctor. He can have instant access to the data and examine the history of each patient. This model provides dynamic access to health data using a single laptop or a mobile device improving diagnosis conditions. Additionally, our implemented architecture virtualizes the data and also in emergency cases it informs the doctor with appropriate warning messages for instant treatment.

The implemented system uses open source software so this allows the research community to extend its capabilities and also to adjust to any possible packages. Furthermore, open source software has zero cost so our application amplifies the costless profile of the health system. Based on the financial circumstances that exist in many countries around the world, such applications can prove to be valuable in human life prevent people from dangerous health problems due to lack of hospital staff or equipment. It aims to the direction of a shared health system that supports all humanity.

OpenMRS platform allows the storage and manipulation of the medical data and it does not have any constraints against other software. Moreover, OpenMRS has modular structure, which means that it can increase its functionalities by embedding the proper module.

Apart from the additional packages that can be incorporated in OpenMRS platform, our system can accept any possible biomedical signal as input for storage or processing. This means

that except from electrocardiogram signal it accepts other signals representing health indicators such as temperature, pressure, oxygen consumption etc. This information can be kept in the database and be accessible to the doctor instantly. Following this direction, our system can be used for monitoring patients for multiple health indicators simultaneously reducing health cost and time spent in hospitals.

Additionally, the exponential growth of IoT industry and the evolution of wearable devices or Body Sensor Networks (BSNs) have the potential to lead to a next generation of health system using wireless technology. This is practical for older people that stay home for a long period without a specialist for treatment. It also limits the need for hospitalization for medical tests. This is practical for counties that lack of medical staff or equipment. The wearable sensors can transmit the measurements in real time because they contain transceivers (Bluetooth or ZigBee protocol [63], [64]) and this helps people that are not familiar with technological achievements and mobile devices. Moreover, based on the computational power that recent mobile devices, computers, wireless sensors and healthcare devices have, a possible future extension of our system is to move the workload for ECG processing from our centralized OpenMRS server to the mobile devices that patients use in order to upload the ECG signal. Each patient's device can extract the required information and then transmit it the server. This reduces dramatically the requirements that the server should have and the only duties that it keeps is monitoring this information and informing doctor in emergency cases. This approach increases the security and the redundancy of the system because patients can keep locally a copy of their ECG signals before uploading it to the server. In case of a DOS attack to the server, following this method, no data loss will occur.

All the above possible extensions have the potential to increase the flexibility and the effectiveness of the recent health system in order to respect the patients and reduce the obstacles that creates in their daily life. It offers the capability of connecting all hospitals by creating a global digital database that uses information by the local databases in each hospital. This is crucial for studying some exceptional cases that are met in medical science and exporting useful conclusions, supporting research in this area. Finally, this idea allows opinion exchange between doctors located in different hospitals around the world using a laptop or a mobile device. This approach saves time and eliminates distance, which are crucial factors that must overcome in health branch.

References

[1] S. Taylor and P. A. Todd, "Understanding information technology usage: A test of competing models," *Information systems research*, vol. 6, no. 2, pp. 144-176, 1995.

[2] S. S. Jones, R. S. Rudin, T. Perry, and P. G. Shekelle, "Health Information Technology: An Updated Systematic Review With a Focus on Meaningful Use," *Annals of Internal Medicine*, vol. 160, no. 1, pp. 48-54, January 2014.

[3] S. Ajami and T. Bagheri-Tadi, "Barriers for Adopting Electronic Health Records (EHRs) by Physicians," *Acta Informatica Medica*, vol. 21, no. 2, pp. 129–134, June 2013.

[4] L. Sörnmo and P. Laguna, "Electrocardiogram (ECG) Signal Processing," *Wiley encyclopedia of biomedical engineering*, 2006.

[5] J. Hatcliff, A. King, I. Lee, A. MacDonald, and A. Fernando, "Rationale and Architecture Principles for Medical Application Platforms," *ACM/EEE Third International Conference on Cyber-Physical Systems (ICCPS 2012)*, pp. 3-12, April 2012.

[6] L. Wang, R. Ranjan, J. Kołodziej, A. Zomaya, and L. Alem, "Software Tools and Techniques for Big Data Computing in Healthcare Clouds," *Future Generation Comp. Syst.*, vol. 43, pp. 38-39, 2015.

[7] S. Sahay, "Big Data and Public Health: Challenges and Opportunities for Low and Middle Income Countries," *Communications of the Association for Information Systems*, vol. 39, no. 20, 2016.

[8] M. Chen, Y. Ma, J. Song, C Lai, and B. Hu, "Smart Clothing: Connecting Human with Clouds and Big Data for Sustainable Health Monitoring," *Mobile Networks and Applications*, vol. 21, no. 5, pp. 825–845, 2016.

[9] J. T. Bram, B. Warwick-Clark, E. Obeysekar, and K Mehta, "Utilization and Monetization of Healthcare Data in Developing Countries," *Big data*, vol. 3, no. 2, pp. 59-66, 2015.

[10] S. Balamurugan, R. Madhukant, V. M. Prabhakaran, and R. Gokul Kruba Shanker, "Internet of Health: Applying IoT and Big Data to Manage Healthcare Systems," *International Research Journal of Engineering and Technology (IRJET)*, vol. 3, no. 10, October 2016.

[11] A. Amato and A. Coronato, "An IoT-Aware Architecture for Smart Healthcare Coaching Systems," *Advanced Information Networking and Applications (AINA), 2017 IEEE 31st International Conference on. IEEE*, pp. 1027-1034, March 2017.

[12] K. S. Velrani and G. Geetha, "Sensor Based Healthcare Information System," *Technological Innovations in ICT for Agriculture and Rural Development (TIAR), 2016 IEEE*, pp. 86-92, 2016.

[13] P. A. Laplante and N. L. Laplante, "A Structured Approach for Describing Healthcare Applications for The Internet of Things," *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pp. 621-625, 2015.

[14] P. A. Laplante and N. Laplante, "The internet of things in healthcare: Potential applications and challenges," *IT Professional*, vol. 18, no. 3, pp. 2-4, 2016.

[15] C. Puri and et al., "iCarMa: Inexpensive Cardiac Arrhythmia Management--An IoT Healthcare Analytics Solution," *Proceedings of the First Workshop on IoT-enabled Healthcare and Wellness Technologies and Systems*, pp. 3-8, 2016.

[16] C. A. Villamil, S. F. Landínez, D. M. López, and B. Blobel, "A Mobile ECG System for the Evaluation of Cardiovascular Risk," *MIE*, pp. 210-214, September 2016.

[17] C. L. Ventola, "Mobile Devices and Apps for Health Care Professionals:Uses and Benefits," vol. 39, no. 5, May 2014.

[18] N. Zarka, A. Saleh, and M. M. Mansour, "Mobile Healthcare System," 2016.

[19] S. H. AlMotiri, M. A. Khan, and M. A. AlGhamdi, "Mobile Health (m-health) System in the context of IoT," *2016 4th International Conference on Future Internet of Things and Cloud Workshops*, pp. 39-42, August 2016.

[20] Knowledge for Health. (2017) mHealthKnowledge. [Online]. http://www.mhealthknowledge.org/resource-type/applications-platforms

[21] J. Baran, A. J. Nimunkar, J. G. Webster, and N. A. Kalogriopoulos, "Electronic Medical Record Systems for Developing Countries: Review," *Engineering in Medicine and Biology Society, Annual International Conference of the IEEE*, pp. 1730-1733, 2009.

[22] (2010) Sana. [Online]. http://dev.sanamobile.org/

[23] L. A. Celi, L. Sarmenta, J. Rotberg, A. Marcelo, and G. Clifford, "Mobile Care (Moca) for

Remote Diagnosis and Screening," *Journal of health informatics in developing countries*, vol. 3, no. 1, pp. 17–21, 2009.

[24] J. L. Helbostad and et al., "Mobile Health Applications to Promote Active and Healthy Ageing," *Sensors*, vol. 17, no. 3, 2017, 622.

[25] P. S. Millard, J. Bru, and C. A. Berger, "Open-source point-of-care electronic medical records for use in resource-limited settings: systematic review and questionnaire surveys," *BMJ Open*, vol. 2, no. 4, 2012, e000690.

[26] M.L.M. Kiah, A. Haiqi, A. A. Zaidan, and B. B. Zaidan, "Open source EMR software: Profiling, insights and hands-on analysis," *Computer methods and programs in biomedicine*, vol. 117, no. 2, pp. 360-382, 2014.

[27] Y. Wang, F. Agrafioti, D. Hatzinakos, and K. N. Plataniotis, "Analysis of Human Electrocardiogram for Biometric Recognition," *EURASIP journal on Advances in Signal Processing*, vol. 19, 2008.

[28] R. J. Prineas, Z. Zhang, and R. S. Crow, *The Minnesota Code Manual of Electrocardiographic Findings.*: Springer Science & Business Media, 2010.

[29] Wikipedia. (2017, June) Electrocardiography. [Online]. https://en.wikipedia.org/wiki/Electrocardiography

[30] J. D. Trigo, A. Alesanco, I. Martinez, and J. Garcia, "A Review on Digital ECG Formats and the Relationships Between Them," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 3, pp. 432-444, May 2012.

[31] A. Bharadwaj and U. Kamath. (2011, February) EE Times Connecting the Global Electronics Community. [Online]. http://www.eetimes.com/document.asp?doc_id=1278571

[32] M. Kaur, B. Singh, J. S. Ubhi, and S. Rani, "Digital Filteration of ECG Signals for Removal of Baseline Drift," *Proceedings of International Conference on Computer Communication and Management (ICCCM 2011)*, vol. 5, 2011.

[33] O. Ondracek, J. Pucik, and E. Cocherová, "Filters for ECG digital signal processing," *International Conference Trends in Biomedical Engineering*, vol. 7, no. 9, Januay 2005.

[34] the National Institute of Biomedical Imaging and Bioengineering (NIBIB) National

Institute of General Medical Sciences (NIGMS). PhysioNet the research resource for complex physiologic signals. [Online]. https://physionet.org/

[35] (2004) OpenMRS. [Online]. http://openmrs.org/

[36] C. J. Seebregts and et al., "The OpenMRS Implementers Network," *International journal of medical informatics*, vol. 78, no. 11, pp. 711-720, 2009.

[37] C. M. Costa and et al., "S2DIA: A Diagnostic System for Diabetes mellitus using SANA platform," *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pp. 6078-6081, September 2012.

[38] J. Flowers. (2013, October) System Requirements. [Online]. https://wiki.openmrs.org/display/docs/System+Requirements

[39] A. Grzybkowski. (2016, December) For Module Developers. [Online]. https://wiki.openmrs.org/display/docs/For+Module+Developers

[40] D. Jazayeri. (2012, March) Core Modules. [Online]. https://wiki.openmrs.org/display/docs/Core+Modules

[41] B. Mamlin. (2015) Bundled Modules. [Online]. https://wiki.openmrs.org/display/docs/Bundled+Modules

[42] P. Brandt. (2016, August) Community Modules. [Online]. https://wiki.openmrs.org/display/docs/Community+Modules

[43] B. Mamlin. (2014, June) Technical Overview. [Online]. https://wiki.openmrs.org/display/docs/Technical+Overview

[44] M. Goodrich. (2011, July) UUIDs. [Online]. https://wiki.openmrs.org/display/docs/UUIDs

[45] B. Mamlin. (2016, August) Check Digit Algorithm. [Online]. https://wiki.openmrs.org/display/docs/Check+Digit+Algorithm

[46] L. Stanisic. (2012, October) Managing Concept Classes. [Online]. https://wiki.openmrs.org/display/docs/Managing+Concept+Classes

[47] L. Stanisic. (2012, October) Managing Concept Datatypes. [Online]. https://wiki.openmrs.org/display/docs/Managing+Concept+Datatypes

[48] J. Arbaugh. (2014, October) HTML Form Entry JavaScript Reference. [Online]. https://wiki.openmrs.org/display/docs/HTML+Form+Entry+JavaScript+Reference

[49] J. Riley. (2011, April) HTML Form Entry Module Getting Started. [Online]. https://wiki.openmrs.org/display/docs/HTML+Form+Entry+Module+Getting+Started

[50] W. J. Brady and et al., *The ECG in Prehospital Emergency Care*.: John Wiley & Sons, 2012.

[51] G. Papazaxos, *The electrocardiogram in clinical practice*.: Medical Publications of Litsas, 2000.

[52] P. R. Rijnbeek, G. Herpen, M. L. Bots, and N. Verweij, "Normal values of the electrocardiogram for ages 16–90 years," *Journal of Electrocardiology*, vol. 47, no. 6, pp. 914-921, August 2014.

[53] Wikipedia. (2017, May) QT interval. [Online]. https://en.wikipedia.org/wiki/QT_interval

[54] ECGpedia. (2011, January) P Wave Morphology. [Online]. http://en.ecgpedia.org/wiki/P_Wave_Morphology

[55] D. W. Romhilt and et al., "A critical appraisal of the electrocardiographic criteria for the diagnosis of left ventricular hypertrophy," *Circulation*, vol. 40, no. 2, pp. 185-196, August 1969.

[56] L. L. Vitelli and et al., "Electrocardiographic findings in a healthy biracial population," *The American journal of cardiology*, vol. 81, no. 4, pp. 453-459, 1998.

[57] H. Blackburn, A. Keys, E. Simonson, P. Rautaharju, and S. Punsar, "The Electrocardiogram in Population Studies," *Circulation*, vol. 21, no. 6, pp. 1160-1175, 1960.

[58] B. P. Betensky and et al., "The V 2 transition ratio: A new electrocardiographic criterion for distinguishing left from right ventricular outflow tract tachycardia origin," *Journal of the American College of Cardiology*, vol. 57, no. 22, pp. 2255-2262, 2011.

[59] A. Oehler, T. Feldman, C. A. Henrikso, and L. G. Tereshchenko, "QRS-T angle: A Review," *Annals of Noninvasive Electrocardiology*, vol. 19, no. 6, pp. 534-542, 2014.

[60] the National Institute of Biomedical Imaging and Bioengineering (NIBIB) National Institute of General Medical Sciences (NIGMS). (2016, july) PhysioBank Annotations. [Online]. https://physionet.org/physiobank/annotations.shtml

[61] S. S. Musoke. (2017, January) HTML Form Entry Module HTML Reference. [Online].

https://wiki.openmrs.org/display/docs/HTML+Form+Entry+Module+HTML+Reference#HTMLFormEntryModuleHTMLReference

[62] I. Germanakis, A. Vlachos, A. Giannopoulos, K. Papadopoulou Legmpelou, and G. Varlamis, *Introduction in Pediadric Cardiology*., 2015.

[63] P. More, K. Memane, T Londhe, and H. J. Thanki, "Advance IoT-based BSN Healthcare System for Emergency Response of Patient with Continuous Monitoring and Motion Detection," *International Journal for Modern Trends in Science and Technology*, vol. 2, no. 12, December 2016.

[64] A. Sangwa and P. P. Bhattacharya, "Wireless Body Sensor Networks: A Review," *International Journal of Hybrid Information Technology*, vol. 8, no. 9, pp. 105-120, 2015.