

Imam Mohammad Bin Saud Islamic University

College of Computer and Information Sciences

**A Client Server Based Application**

**for HeartBeat Analysis**

**By:**

Abdullah Askar, 432011125, CS

Abdulrahman Abahussin, 431006799, CS

**Supervised by:**

A.Ahmed Biyabani

Project Submitted in Fulfillment for the CS493 Course requirements

Second-1436



Imam Mohammad Bin Saud Islamic University

College of Computer and Information Sciences

**Android Based Application for Heart-Beat Analysis**

**By:**

Abdullah Askar, 432011125, CS

Abdulrahman Abahussin, 431006799, CS

We hereby certify that this project satisfies the project requirements:

|  |  |
| --- | --- |
| **Approved by** | A.Ahmed Biyabani |
| **Date of Approval** |  |
| **Signature** |  |

|  |  |
| --- | --- |
| **Approved by** |  |
| **Date of Approval** |  |
| **Signature** |  |

**Declaration**

We Abdullah Askar and Abdulrahman being members of final year project group number CS 02 declare that this report contains only work completed by members of our group except for information obtained in a legitimate way from literature, company or university sources. All information from these other sources has been duly referenced and acknowledged in accordance with the University Policy on Plagiarism.

Furthermore, we declare that in completing the project, the individual group members had the following responsibilities and contributed in the following proportions to the final outcomes of the project:

|  |  |  |  |
| --- | --- | --- | --- |
| **Student ID** | **Responsibility** | **Contributed** | **Signature** |
| **432011125** |  |  |  |
| **431006799** |  |  |  |

**Acknowledgment**

Express your appreciation to who have helped during your work…

Sample:

First and foremost, we would like to present my deepest gratitude to Almighty ALLAH for his bounties and blessings and for giving us the ability to finish this project

We would like to express our deep appreciation and our sincere gratitude to our supervisor A.Ahmed Biyabani for his valuable advice guidance throughout this project.

Finally, we would like to thank our families and friends for continued encouragements and support during this project and along the years of study.

**Abstract:**

**Client Server** based application for heartbeat analysis is system to detect arrhythmia in real time. The System after retrieving the heartbeats signal applies certain signal processing algorithms then apply some anomaly detection algorithms that classify the heartbeat for detection of arrhythmia disease. This system uses android mobile application as front-end, and PHP, MySQL and C++ as back-end. MITBIH public database to obtain heartbeats and test the accuracy of the extracted results. The web server applies Pan-Tompkins algorithm and p-wave detection to feed hierarchical classification decision tree, which is used to classify type of heartbeat in order to detect arrhythmia type if existed then update database. The android application is used to keep track of patient heart and feed report for the caretaker about the his patients. It also provide side function for the patient that can help him get help such as, calling the ambulance and the caretaker. At the end all functions that has been implemented shall be tested through real patient caretaker scenarios.

**Abstract (in Arabic):**

Rewrite your abstract in Arabic.

**Keywords:**

Android application.

Server.

Arrhythmia.

Sensor.

Pan-tompkins algorithm.

MySQL database.

Php.

RR-interval.

Hierarchical algorithm.

Beats per minute (bpm).

P-Wave.

Anomoly Detection

QRS

T-Wave

**List of abbreviation:**

List the abbreviations you have used in your project if there are any and what are they stands for…

Samples:

SOS:  Save Our Souls.

bpm: Beats Per Minute

**Table of Contents**

Declaration ..................................................................................................................................iv

Acknowledgment ..........................................................................................................................v

Abstract .......................................................................................................................................vi

Abstract (in Arabic) ...................................................................................................................vii

Keywords ..................................................................................................................................viii

List of abbreviation .....................................................................................................................ix

Table of Contents ..........................................................................................................................x

List of Figures .............................................................................................................................xi

List of Tables .............................................................................................................................xii

Chapter One: Introduction ............................................................................................................1

1.1 Introduction .......................................................................................................................3

1.2 Problem statement .............................................................................................................3

1.3 ………...............................................................................................................................3

1.4 ………...............................................................................................................................3

1.5 Summary ...........................................................................................................................3

Chapter Two: Background ............................................................................................................4

Chapter Three: Analysis and Methodology ..................................................................................6

3.1 Introduction .......................................................................................................................8

3.2 …………………...............................................................................................................8

3.2.1 …………………… ......................................................................................................8

3.2.2 …………………… ......................................................................................................8

3.2.2.1 ……………...........................................................................................................8

3.2.2.2 ……………….......................................................................................................8

3.3 …………………...............................................................................................................8

3.4 ………………………….................................................................................................10

3.5 Summary .........................................................................................................................10

Chapter Four: Design and Implementation .................................................................................11

Chapter Five: Testing ..................................................................................................................13

Chapter Six: Conclusion and Future Work .................................................................................15

References ...................................................................................................................................17

Appendix ...................................................................................................................................A.1

**List of Figures**

Figure 1 Gantt chart .................................................................................................9

**List of Tables**

Table 1 Student’s Responsibilities .........................................................................10

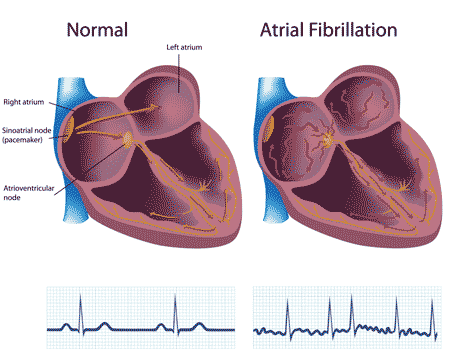
**Chapter One: Introduction**

* 1. **Abstract:**

In this chapter, we will give information about arrhythmia, different between normal heartbeat signal and abnormal. Also, objective, problem statement, and scope of the project.

* 1. **Background study:**

Arrhythmia is any of a group of conditions in which the electrical activity of the heart is irregular, means it could be faster or slower than normal. Arrhythmia diseases are of many types. In abroad study that has been applied to United States of America showed the occurrence of arrhythmia is common; as many as 2.2 million Americans are living with atrial fibrillation (one type of arrhythmia). Moreover, a recent study of same candidates has also suggested that 1 in 4 Adult Americans over the age of 40 could develop an irregular heartbeat. Another study of same country suggest that the number of people who death is caused by (SCD) Sudden cardiac death is 325,000 each year. [2]



* 1. **Problem statement:**

**Implement a Client-Server Based Application** that classifies the signal of a heartbeat in order to detect weather normal of abnormal heartbeat.

**In this project the critical point of view** is to extract heartbeat data and inspect weather normal or abnormal heartbeat.

**In order to identify the signals data**, a Pan-Tompkins algorithm is used for QRS detection, then QRS signal is removed in order to detect P and T wave, finally apply decision tree and classification algorithms in order to inspect heartbeat abnormality.

* 1. **Objective of the project:**

This project’s importance arises for:

* Detection of arrhythmia in real time.
* Making connection between patients and caretakers.
* Reduction of time and cost for both caretakers and hospitals.
  1. **Scope of the project:**

Our project scope:

* This system uses MIT-BIH as a measurement of the performance of the detection algorthims.
* This system focuses on software to make it accept any resources from any sensor.
* This system requires to builds android application for its ease of access.
* Integrating emergency function into one application
* Improving the method of long distance care taking.
* Building an integrated system of deferent layers.
  1. **Summary of this chapter:**

As you read, arrhythmia is occur a lot. It needs so much cost and time to detect. Some type of arrhythmia cause death. You read objective and our scope.

**Chapter Two: Background**

**2.1 Abstract:**

This chapter will discuss system architecture, the techniques used include the algorithm as well as the literature reviews when doing research for this project.

**2.2 Model of project:**

In this Project we’ve selected Plan-Driven agile for the ambiguity of some of the requirement and for limited number of workers. Also the lack of experience..[3]

**2.3 Background development:**

**2.3.1 Programming languages:**

1. Java .
2. Php.
3. C++.

**2.3.2 Technology:**

1. Android application.
2. REST Web Service.
3. JSON Parser.

**2.3.3 Tools:**

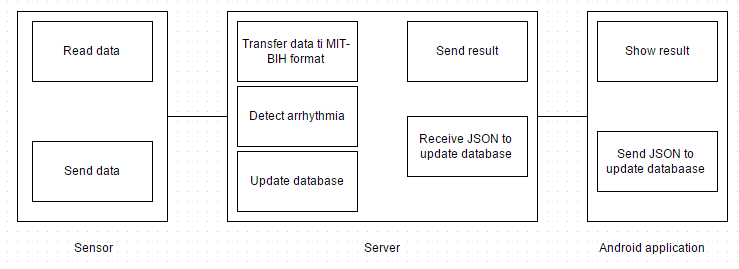
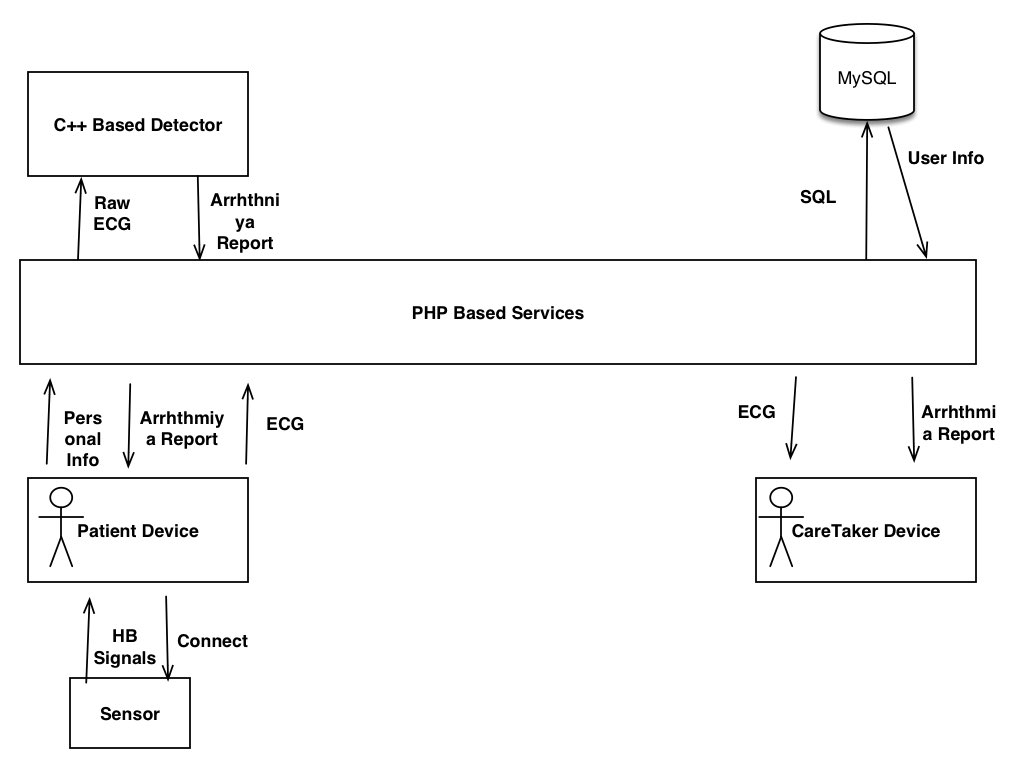
1. Android studio.
2. PhpMyAdmin.

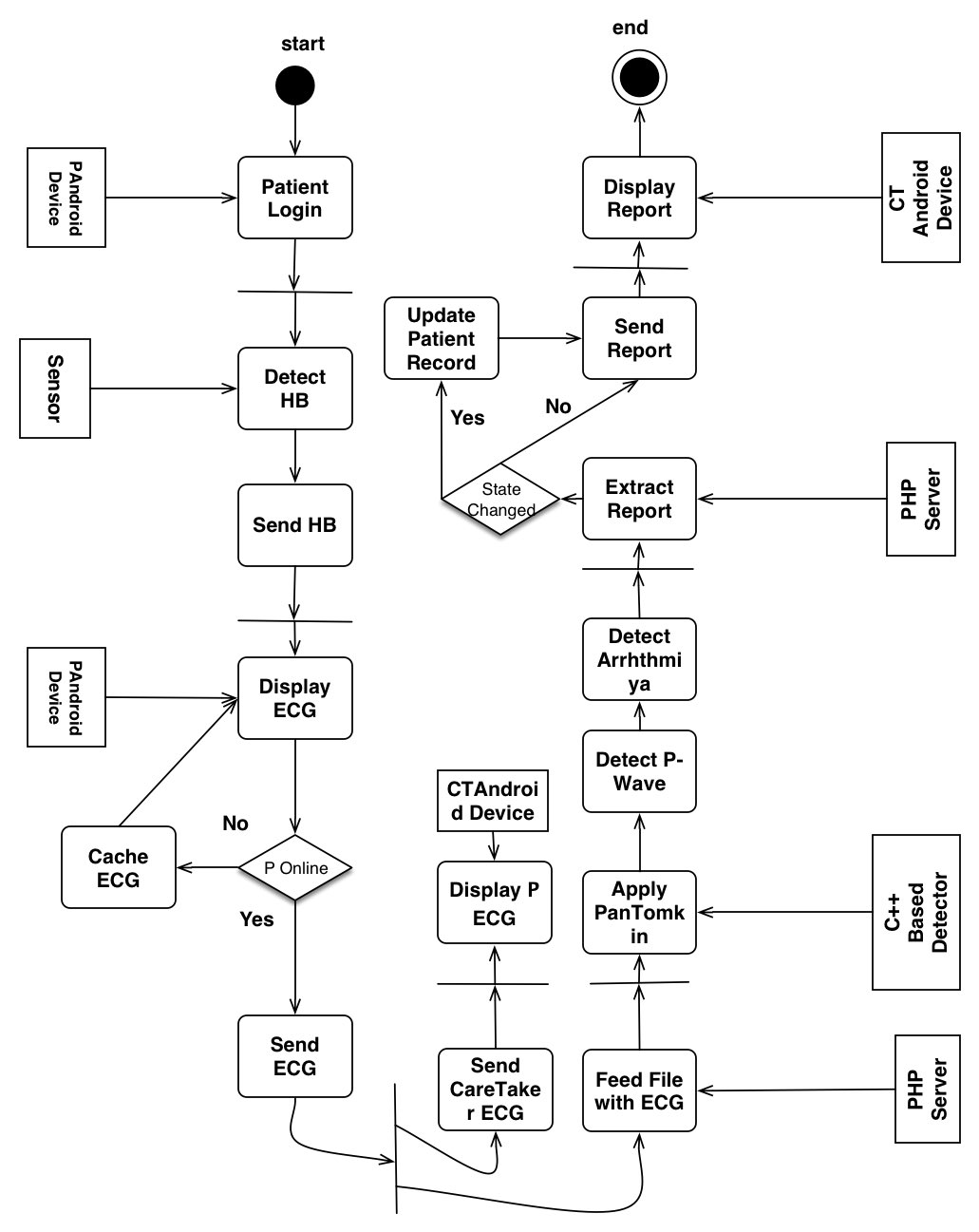
**2.3.4 Algorithm:**

1. Pan-Tomking.
2. Hierarchical.
3. P-wave detecting.
4. Time Based Detection Algorthim

**2.4 System architecture:**

This project includes three main components, which are component are sensor, web server, and android application. The sensor is the starting point of the process of this project. It reads heartbeat data from patient then send it to the android application. The Application then sends the detected HB to the web server. The server after receiving the detected HB transfers its format into MIT-BIH format then use Pan-Tomkins algorithm to detect QRS complex and p-wave, after that it feeds the outputted data into the hierarchical algorithm which classify each heartbeat basing on the algorithm to detect if normal or abnormal heartbeat then if it abnormal heartbeat what kind of arrhythmia is it. The next process is where the database is been updated by determining weather patient is in serious or unserious case. Finally the server updates the care taker with the result through it mobile application.



****

**2.5 The server:**

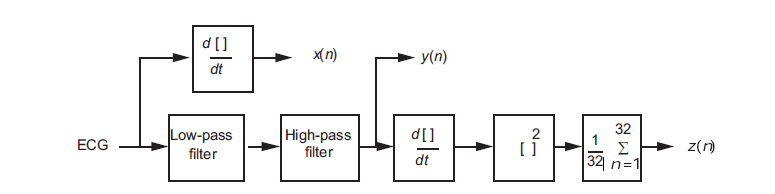
The server has three processing phases. First, transfer received data to MIT-BIH format. Second, detecting arrhythmia. Third, update database by the detected results.

**2.5.1Transfer data to MIT-BIH format:**

**2.5.2 Detecting arrhythmia:**

The server uses three algorithms for signal processing and abnormality detection. It uses Pan-Tomkins algorithm for QRS complex detection. Secondly it uses p-wave detection to extract the number of p-wave. Finally a hierarchical algorithm, which is the key algorithm to detect arrhythmia.

**2.5.2.1 Pan-tompkins algorithm:**



Pan-tomkins gives us QRS complex in real time. It is sequence of function start from low-pass filter end to moving window. Each function need data from previses function. Its functions are low-pass filter, high-pass filter, Derivative, Squaring function, and Moving window integral. Low-pass and high-pass filter are band-pass integer filter.

**Band-pass integer filter**

The band-pass filter for the QRS detection algorithm reduces noise in the ECG signal by matching the spectrum of the average QRS complex. Thus, it attenuates noise due to muscle noise, 60-Hz interference, baseline wander, and T-wave interference. The passband that maximizes the QRS energy is approximately in the 5–15 Hz range. The filter implemented in this algorithm is a recursive integer filter in which poles are located to cancel the zeros on the unit circle of the *z* plane. A low-pass and a high-pass filter are cascaded to form the band-pass filter.

**Derivative**

After the signal has been filtered, it is then differentiated to provide information about the slope of the QRS complex.

**Squaring function**

The previous processes and the moving-window integration, which is explained in the next section, are linear processing parts of the QRS detector. The squaring function that the signal now passes through is a nonlinear operation.

**Moving window integral**

The slope of the R wave alone is not a guaranteed way to detect a QRS event. Many abnormal QRS complexes that have large amplitudes and long durations (not very steep slopes) might not be detected using information about slope of the R wave only. Thus, we need to extract more information from the signal to detect a QRS event.

**Thresholding**

The set of thresholds that Pan and Tompkins (1985) used for this stage of the QRS detection algorithm were set such that signal peaks (i.e., valid QRS complexes) were detected. Signal peaks are defined as those of the QRS complex, while noise peaks are those of the T waves, muscle noise, etc. After the ECG signal has passed through the bandpass filter stages, its signal-to-noise ratio increases. This permits the use of thresholds that are just above the noise peak levels. Thus, the overall sensitivity of the detector improves.

Two sets of thresholds are used, each of which has two threshold levels. The set of thresholds that is applied to the waveform from the moving window integrator is

*SPKI* = 0.125\**PEAKI* + 0.875\**SPKI* if *PEAKI* is the signal peak

*NPKI* = 0.125\**PEAKI* + 0.875\**NPKI* if *PEAKI* is the noise peak

*THRESHOLD I1* = *NPKI* + 0.25 (*SPKI* – *NPKI* )

*THRESHOLD I2* = 0.5 *THRESHOLD I1*

All the variables in these equations refer to the signal of the integration waveform and are described below:

*PEAKI* is the overall peak.

*SPKI* is the running estimate of the signal peak.

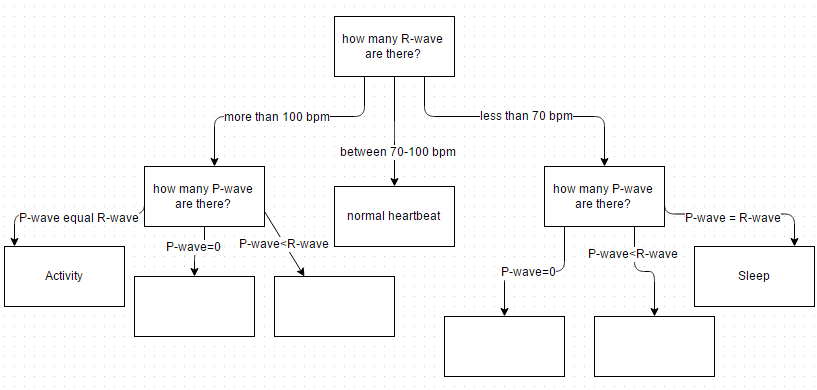
*NPKI* is the running estimate of the noise peak.

*THRESHOLD I1* is the first threshold applied.

*THRESHOLD I2* is the second threshold applied.

**2.5.2.2 P-wave detecting:**

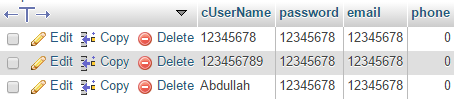
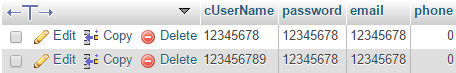
**2.5.2.3 Hierarchical algorithm:**

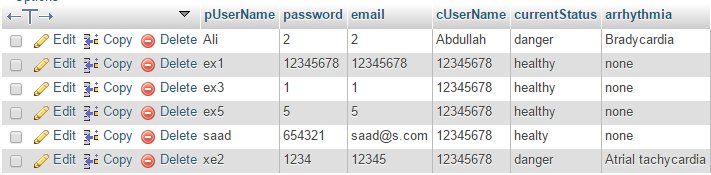
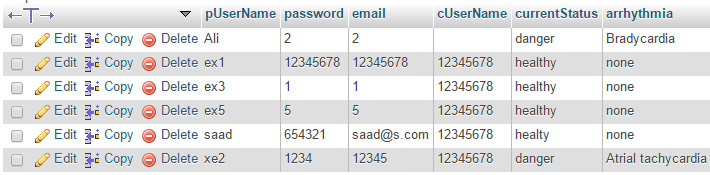


Hierarchical algorithm is tree. It has two iteration. First iteration, root has three children which these children are more than 100 beat per minute, normal heartbeat, less than 70 beat per minute. Second iteration is divide children to type of arrhythmia. More than 100 bpm has child Atrial tachycardia or activity. Normal heartbeat child does not has children. Less than 70 bpm has bradycardia or sleep. Each iteration need result from other algorithm. First iteration needs result from pan-tomkins algorithm to children. Second iteration needs result from p-wave detecting to choose type of arrhythmia.

**2.5.3 Update database:**

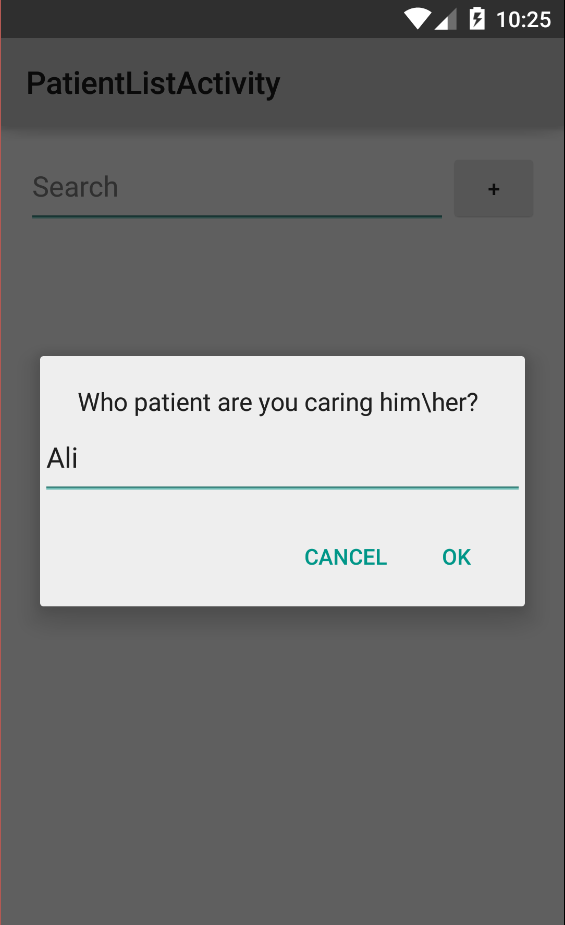
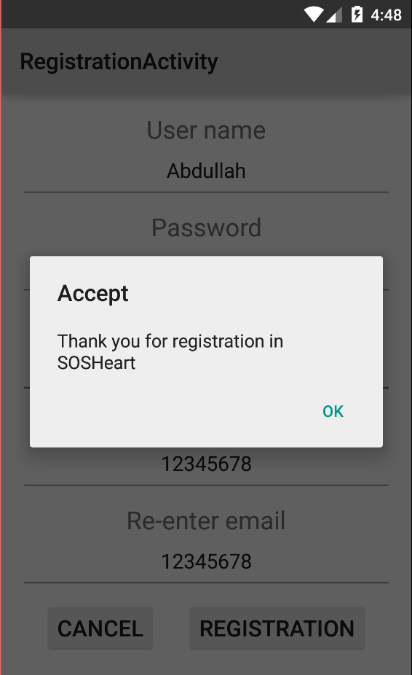
The server can update database from detecting arrhythmia or android application. From detecting arrhythmia, it only update current state and arrhythmia. From android, it updates caretaker table with new row and patient table in caretaker.

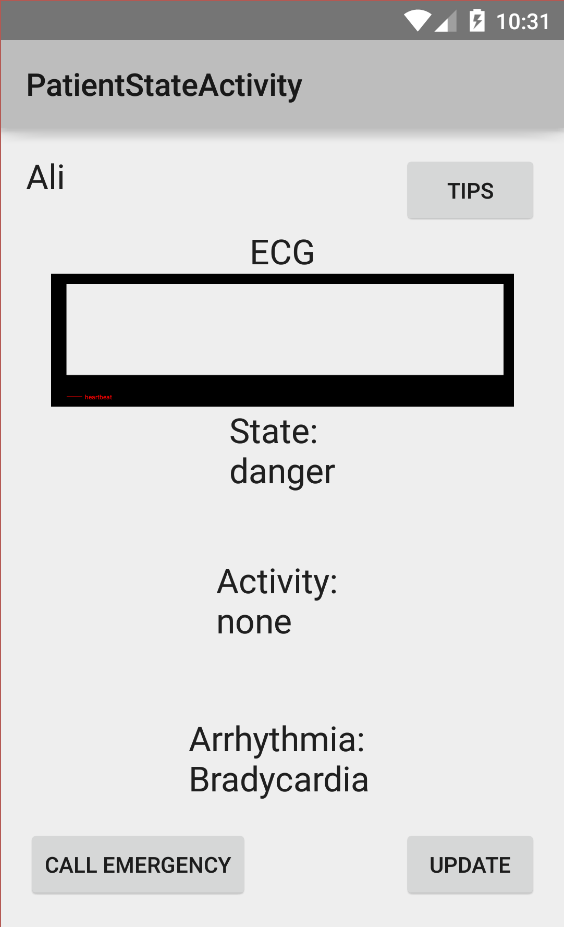
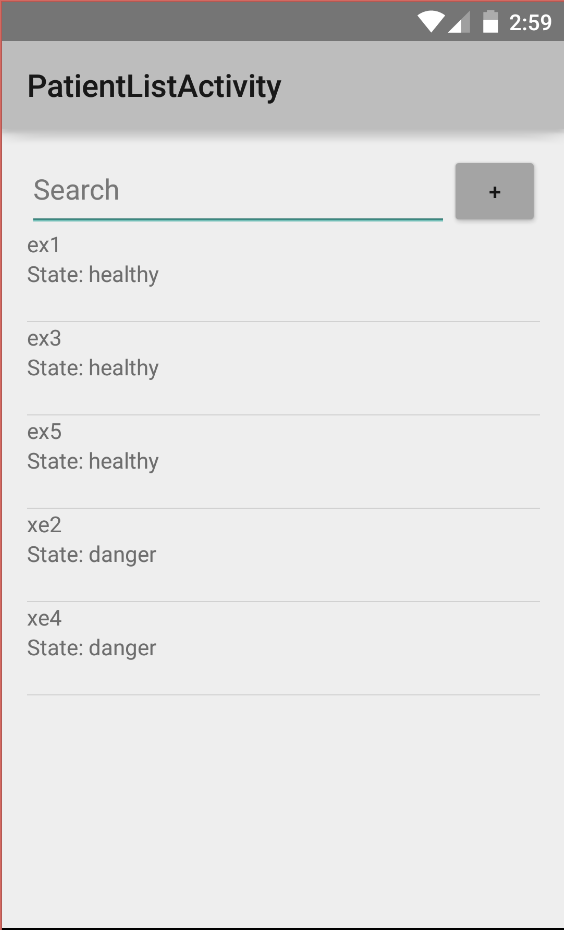




**2.6 Android application:**

Android base application for heartbeat analysis use android for show result or update database as previse section. The result show in this figure.





**Chapter Three: Analysis and Methodology**

* 1. **Methodology:**

**3.2.1Definition and reasoning:**

* + - 1. **Arrhythmia:**

**Arrhythmia** is any of a group of conditions in which the electrical activity of the heart is irregular, means it could be faster or slower than normal.[1]

**Why our project need it?**

Because arrhythmia is core of our project.

* + - 1. **Finite state automata (FSA):**

**Finite state automata** is state has input and output expect goal state.[8]

**Why our project need it?**

Because one of common method to detect anomaly.

* + - 1. **Hierarchical clustering:**

**Hierarchical clustering** is a method of cluster analysis which seeks to build a hierarchy of clusters. [9]

**Why our project need it?**

Because it used in relate work and comparing it with FSA.

* + - 1. **Pan-Tomking algorithm:**

**Pan-Tomking algorithm** is algorithm to translate single heartbeat from data to ECG. [7]

**Why our project need it?**

Becauseit is only way to detect arrhythmia by using ECG.

* + - 1. **Electrocardiograph (ECG):**

**Electrocardiograph (ECG)** is the recording the heartbeat activity. [12]

**Why our project need it?**

Because caretaker know what type of arrhythmia.

* + - 1. **MIT-BIH database:**

**MIT-BIH database** is public database focus and study arrhythmia. Also it has many sample of different type of arrhythmia.

**Why our project need it?**

Because MIT-BIH database has sample of arrhythmia.

* + - 1. **Patients:**

**Patients** is user who will carry sensor.

* + - 1. **Caretakers:**

**Caretakers** is user who will have mobile application.

* + - 1. **Sensor:**

**Sensor** is hardware have ability to read heartbeat, accepting new application, and sending data to web service.

**Why our project need it?**

Because sensor has ability read heartbeat and sending them to web service.

* + - 1. **Web service:**

**Web service** is a method of communication between two electronic devices over a network.[?]

**Why our project need it?**

Because web service has very fast process and we need this process quickly to detection arrhythmia. Also if sensor application and mobile application in different platform.

* + - 1. **Mobile application:**

**Mobile application** is software in mobile.

**Why our project need it?**

Because mobile application has ability to send location and receive notification.

**3.2.2 Model of project:**

Our project select plan-driven agile because we focus in coding and testing more than requirement. Also we have database and web service may be change every time.[3]

* + 1. **Working plan:**
       1. **Implementation Plan:**
* Download MIT-BIH database.
* Learning develop android application.
* Learning develop php web service.
* Learning create SQL database.
* Creating web service.
* Creating android application.
* Creating database.
* Testing with public database.
  + - 1. **Time-frame:**

**3.2.3.3 How do we divide the project?**

We divide the project between us by someone take android and the other take server. Abdullah take android and Abdurhman take server because Abdullah has basic in android and Abdurhman has basic in Php and database. When someone finish has job help the other.

**3.2.3.4 How do start the project?**

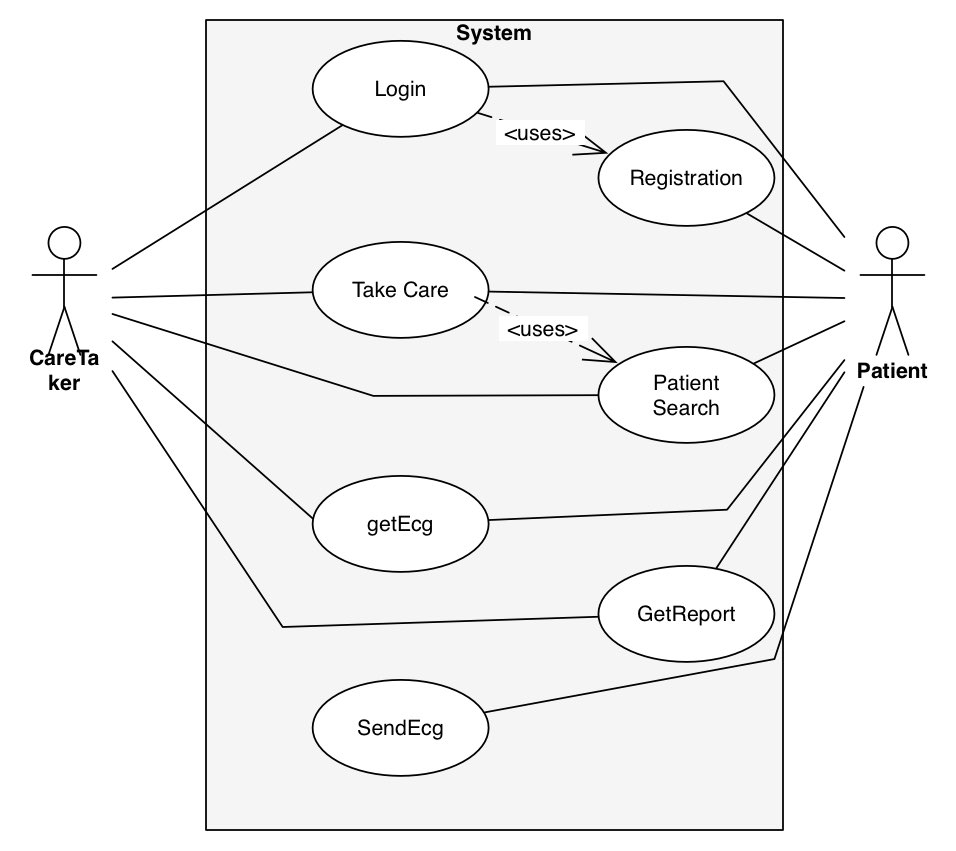
Before we begin the project, we have problem how to start the project. We ask group of programming about the project how to start it? They say start from what user need. We don’t understand that but they explained it. Starting from login and registration then make link between caretaker and patient then show the information of patient to caretaker without detecting then detecting arrhythmia.

**3.2.4 Simulation**

For the limitation/lack of budget the sensor availability was shot. Therefore, we have proposed a simulated result which occurred through building an android application that have cached a HB data which has been downloaded manually from the MIT-BIH database then added to this application which will allow the user to select any information then select which starting and ending time the user wants the simulate on then it will return the detected result which the user can verify its accuracy through checking the actual result in the MIT-BIH website.

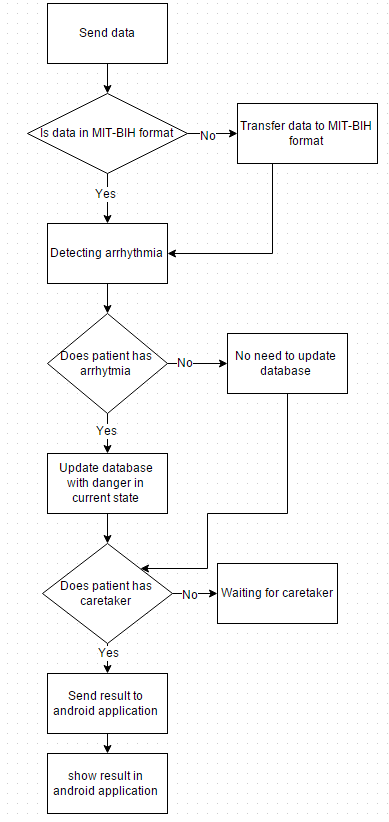
**3.3 System analysis:**

**3.3.1 Use case:**

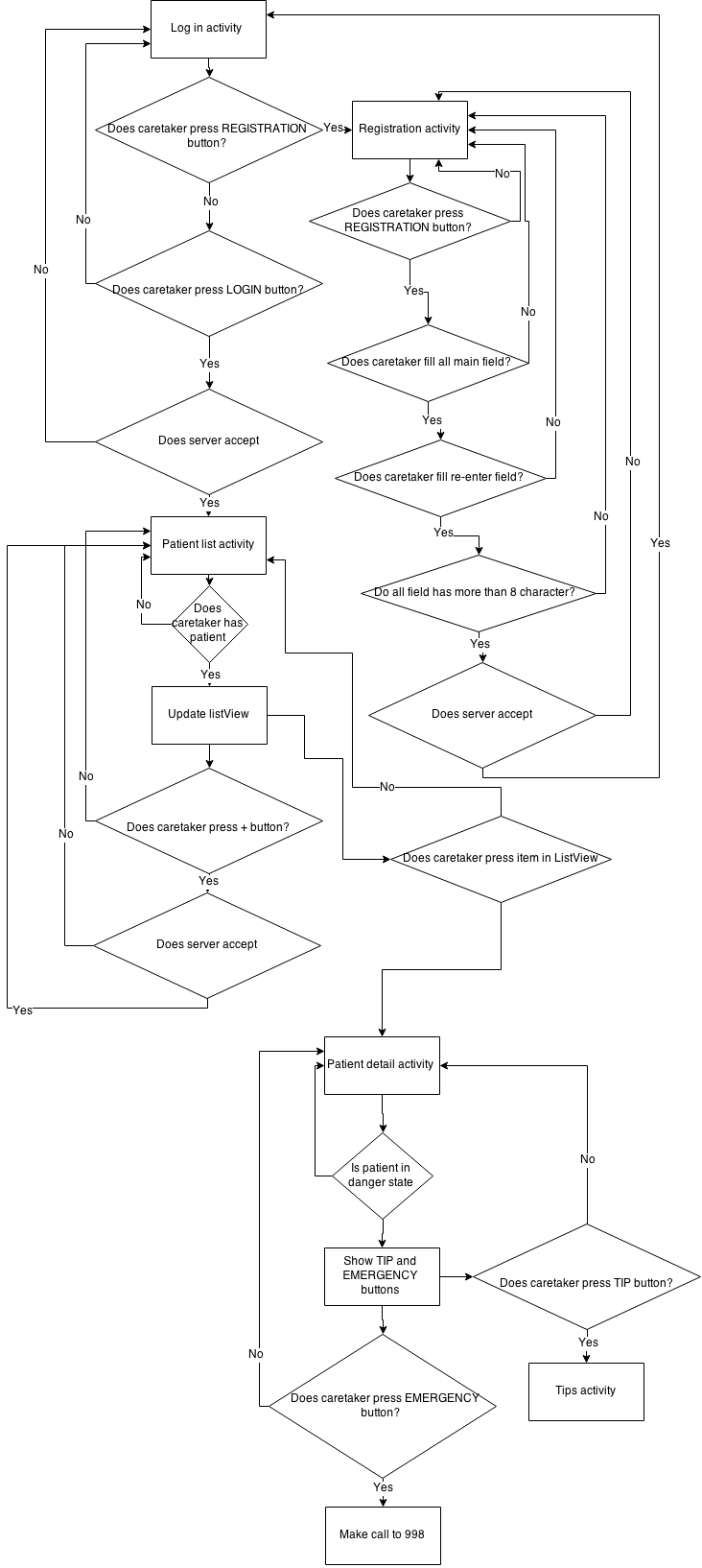
****

**3.3.2 Process flow chart**

**3.3.2.1 Form patient:**

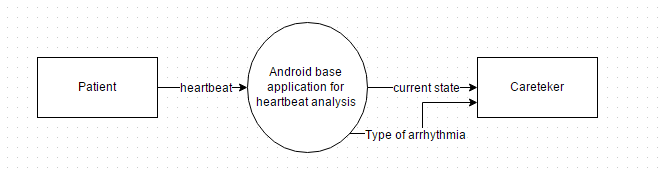


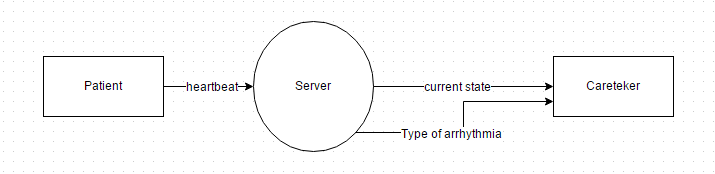
**3.3.2.2 From caretaker:**

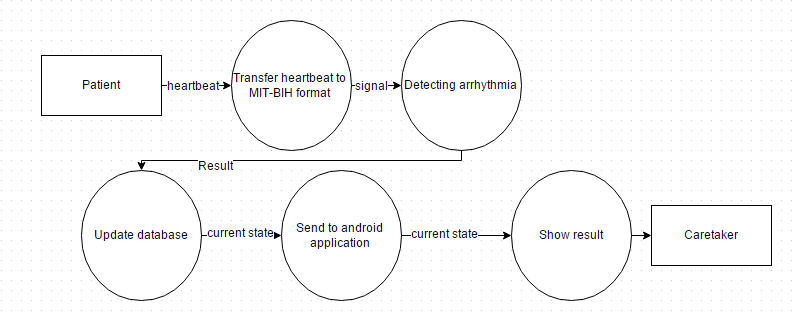


**3.3.3 Data flow:**

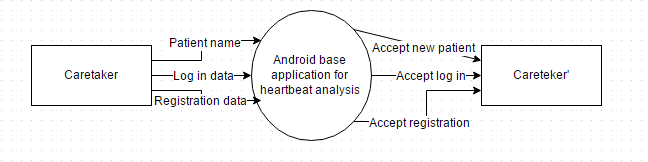
**3.3.3.1 From Patient to Caretaker:**

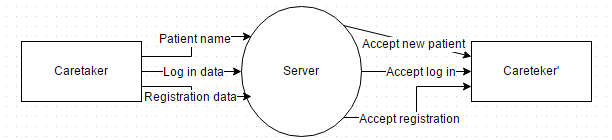


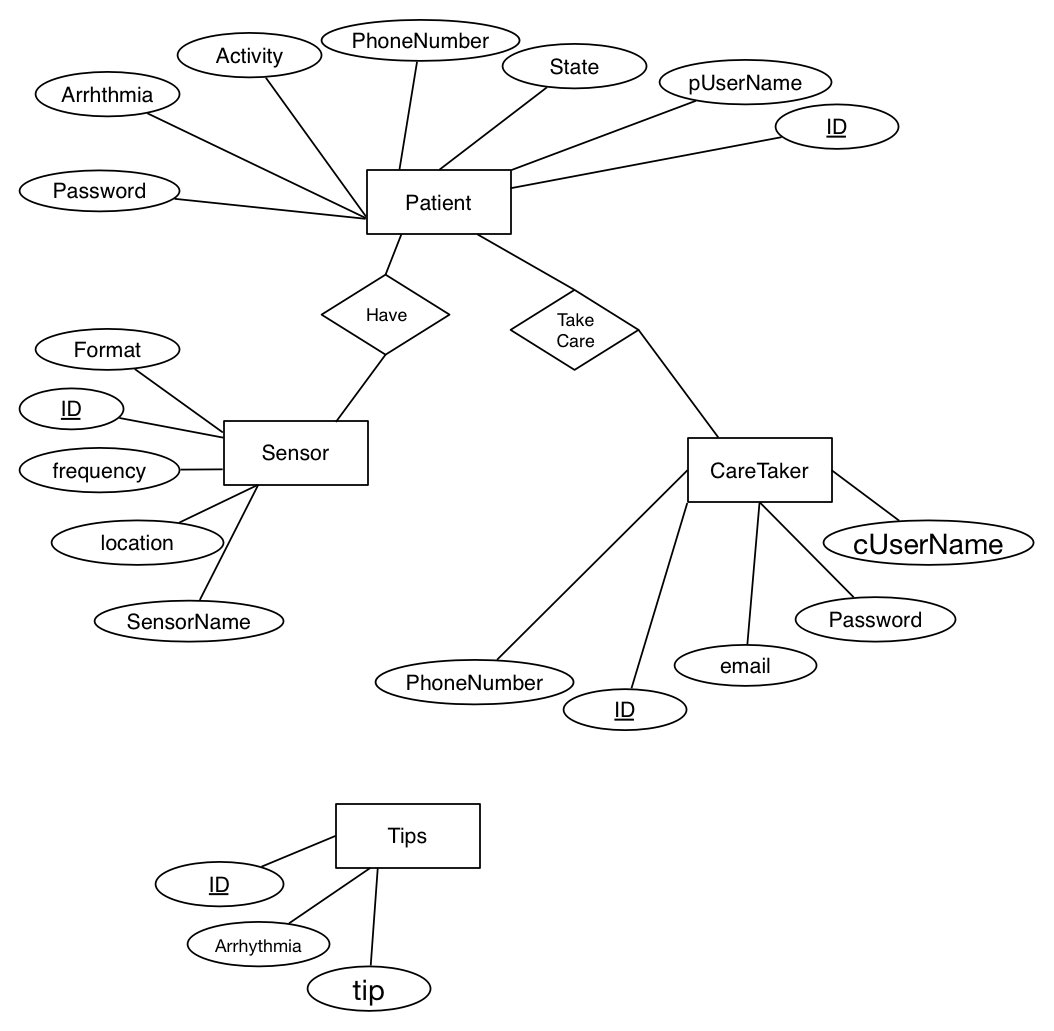




**3.3.3.2 From Caretaker to Caretaker’:**





**3.3.4 Database design:**

**Chapter Four: Design and Implementation**

Implementation:

**Abstract:**

In this section, we show you how we write pan-tompkins algorithm. Also, we will show how to make different process work together. As you see in analysis section, we show different process like detection arrhythmia and send JSON data to server by android.

**Implementation pan-tompkins algorithm:**

As you read in background section, pan-tompkins algorithm is detecting QRS complex in real time by sequence of steps. First, lowpass filter and highpass filter are filtering noise and make delay. Second, derivative is providing information about the slope of the QRS complex. Third, squaring is transferring nonlinear operation to linear operation. Fourth, moving window integral is increasing slope information of the R wave only. Last step, thresholding is splitting QRS complex from noise.

**Lowpass filter:**

int lowPassFilter(int data)

{

int y0;

static int y11 = 0, y2 = 0, x[26], n1 = 12;

x[n1] = x[n1 + 13] = data;

y0 = (y11 << 1) - y2 + x[n1] - (x[n1 + 6] << 1) + x[n1 + 12];

y2 = y11;

y11 = y0;

y0 >>= 5;

if(--n1 < 0)

n1 = 12;

return(y0);

}

**Highpass filter:**

int highPassFilter(int data)

{

int y0;

static int y1 = 0, x[66], n = 32;

x[n] = x[n + 33] = data;

y0 = y1 + x[n] - x[n + 32];

y1 = y0;

if(--n < 0)

n = 32;

return(x[n + 16] - (y0 >> 5));

}

**Derivative:**

int derivative(int data)

{

int y, i;

static int x\_derv[4];

/\*y = 1/8 (2x( nT) + x( nT - T) - x( nT - 3T) - 2x( nT - 4T))\*/

y = (data << 1) + x\_derv[3] - x\_derv[1] - ( x\_derv[0] << 1);

y >>= 3;

for (i = 0; i < 3; i++)

x\_derv[i] = x\_derv[i + 1];

x\_derv[3] = data;

return(y);

}

**Squaring:**

int squaring(int data){

//if (data\*data>8000)return(8000);

return(data\*data);

}

**Moving Window Integral:**

int movingWindowIntegral(int data)

{

static int x[32], ptr = -1;

static long sum = 0;

long ly;

int y;

if(++ptr == 32)

ptr = 0;

sum -= x[ptr];

sum += data;

x[ptr] = data;

ly = sum >> 5;

y = (int) ly;

return(y);

}

**Thresholding and detecting QRS complex:**

for(int j=0;j<size;j++){

byMean<<id[j]<<"\t";

byMean<<oldData[j]<<"\t";

byMean << data[j] <<"\t";

calculateThresholdI1ByMean();

calculateThresholdI2ByMean();

if(data[j]>thresholdI1ByMean){

if (data[j]>thresholdI2ByMean) {

string t;

stringstream convert2;

convert2 << data[j];

t= convert2.str();

byMean << t <<"\n";

QRS[j]=data[j];

calculateSPKIByMean(j);

qrs=true;

} else{

byMean <<"0\n";

QRS[j]=0;

calculateNPKIByMean(j);

noqrs=true;

}

} else{

QRS[j]=0;

byMean <<"0\n";

noqrs=true;

}

if(qrs==true&&noqrs==false){//in qrs looking for R which is max value

if(data[j]>maxR){

maxR=data[j];

maxPoint=j;

}

} else if(qrs==false&&noqrs==true){//going to next qrs maybe looking for P and T (no idea how)

} else if(qrs==true&&noqrs==true){//transfare to going to next qrs or enter qrs start form Q end by T

if(startqrs==false){//find next qrs

qrs=false;

noqrs=false;

startqrs=true;

int z=j;

qrsinfo<<id[z+1]<<"\t"<<time[1+z]<<"\t";

stfQRS=time[z+1];

count++;

}else{ //out from previse qrs

qrs=false;

noqrs=false;

startqrs=false;

maxR=0;

QRS[j]=data[j];

qrsinfo<<id[j]<<"\t"<<time[j]<<"\t"<<(time[j]-stfQRS)<<"\t";

qrsinfo<<id[maxPoint]<<"\t";

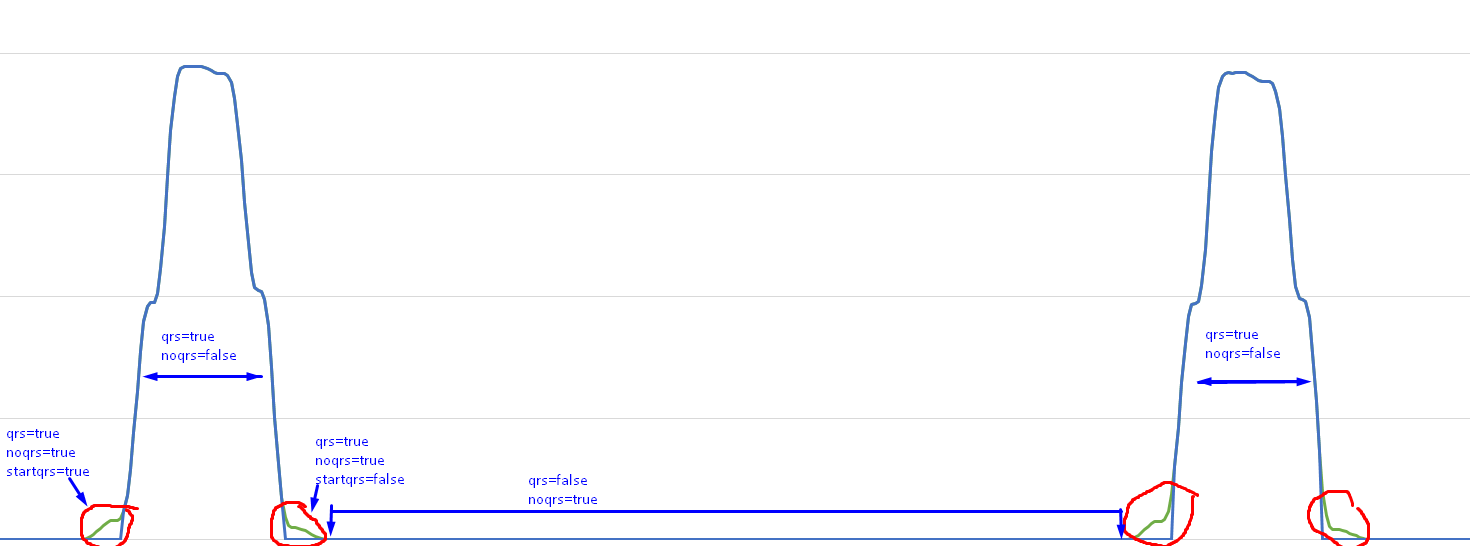
qrsinfo<<count-1<<"\n";

}

}

}

We don’t implementation pan-tompkins like in paper. Because it has backward search for miss single QRS complex. But we use three boolean to detect all QRS complex without missing anyone. In these code, qrs, noqrs, and startqrs are flag to detecting QRS complex by simple method. If one of qrs and noqrs is true the other false that mean in area. If both are true that mean there in transferring area or noise area which it is transferring from no QRS to QRS or out from QRS to no QRS. See the figure below:



**Execute c++ code from php:**

The previous process is local code which the mean is for testing not for real life. For execute c++ code from php which it is our server language, we need to write code execute line in php and make c++ accept input from other resource.

**Execute line in php:**

exec("mainv15.exe $name $fileName $qrsName $fileResult",$output);

main15.exe is file generating after compliec mainv15.cpp

$name is variable for having patient name.

$fileName, $qrsName, and $fileResult ara variable for having files name.

$output is variable for having output from c++ we mean from cout.

In this code, we write three variables for three files which these file are patientName\_data.txt, patientName\_qrs.txt, and patientName\_result.txt. patientName\_data.txt is storing data come from sensor in format (id of data, time, and data) then reading them from c++ to detect QRS complex. patientName\_qrs.txt is result from pan-tompkins and patientName\_data.txt then storing them in format (when the QRS start, when the QRS end , detecting max point for R-wave, time between start QRS to end, number of QRS) in every line. patientName\_result.txt is result from patientName\_qrs.txt to provide max time between start QRS and how many QRS in one minute.

**Updating c++ code:**

int main(int argc, char \*argsv[]) {

if (argc > 1) {

string arg1(argsv[1]);

patientName=arg1;

string arg2(argsv[2]);

dataFile=arg2;

string arg3(argsv[3]);

qrsFile=arg3;

string arg4(argsv[4]);

resultFile=arg4;

// do stuff with arg1

setupData();

updateFileResult();

}else{

cout<<"Error code 1"<<endl;//miss input

}

return 0;

}

In c++ code, there make main function like normal function which accept input from other resource. Argc variable is size for argsv array. Argsv array has input from php which these input are $name, $fileName, $qrsName, and $fileResult. Cout line return data to php code.

**Receiving data from sensor:**

We make acceptable format for detecting QRS is id data, time, and data. Our server can’t accepting other format.

if( isset($\_POST["json"]) ) {

$data = json\_decode($\_POST["json"]);

$name=$data->PatientName;

// include db connect class

require\_once \_\_DIR\_\_ . '/db\_connect.php';

// connecting to db

$db = new DB\_CONNECT();

$searchCTQuery = "SELECT \* FROM patienttable WHERE pUserName = '$name'";

//Check if User exist as patient

$result=mysql\_query($searchCTQuery);

if(mysql\_num\_rows($result) > 0){

$qrsName =$name."\_qrs.txt";

$fileResult=$name."\_Result.txt";

$fileName=$name."\_data.txt";

$fileN=fopen($fileName,"a");//for start writing form end of file

foreach( $data->lineData as $lineD ){

$endl="\r\n";

$realData2=$lineD;

fwrite($fileN,$realData2);

fwrite($fileN,$endl);

}

fclose($fileN);

exec("mainv15.exe $name $fileName $qrsName $fileResult",$output);

/\*

Hierarchical algorithm.

\*/

echo json\_encode($data);

}

In this code, our server receive data as JSON format then storing it in $data to make it easy use. It make sure the patient name are registration in the server by $ searchCTQuery. It store data to patientName\_data.txt. it execute c++ code as you read in previous process.

**Hierarchical algorithm:**

**Updating database after hierarchical algorithm:**

**Sending JSON and receiving it:**

**From android:**

In android, there is two method to make request to server which these request are post and get. We use post when we want send data to server and get response. For get method, we use it when there is no sending data to server. In our project, there is no get method because every activity need to send data to server.

public JSONObject requestToPHP(String url,String method,List<NameValuePair> key) throws IOException, JSONException {

HttpClient client=new DefaultHttpClient();

HttpEntity entity=null;

HttpResponse response=null;

if(method.equals("POST")){

HttpPost post=new HttpPost(url);

if (key != null) {

post.setEntity(new UrlEncodedFormEntity(key));

}

response= client.execute(post);

}else {

HttpGet get=new HttpGet(url);

if (key != null) {

String paramString = URLEncodedUtils

.format(key, "utf-8");

url += "?" + paramString;

}

response= client.execute(get);

}

StatusLine statusLine=response.getStatusLine();

if(statusLine.getStatusCode()== HttpStatus.SC\_OK){

InputStream inputStream=response.getEntity().getContent();

StringBuilder stringBuilder=new StringBuilder();

BufferedReader bufferedReader=new BufferedReader(new InputStreamReader(inputStream));

String line;

while ((line=bufferedReader.readLine())!=null)

if (!line.startsWith("<", 0)) {

if (!line.startsWith("(", 0)) {

stringBuilder.append(line + "\n");

}

}

String toString=stringBuilder.toString();

jsonObject=new JSONObject(toString);

jsonObject.put("stateLine",true);

}else {

response.getEntity().getContent().close();

jsonObject.put("stateLine",false);

}

return jsonObject;

}

In this code, requestToPhp is method for sending JSON in simple format and receiving JSON in any format. There is other code do the same job in other format like in our Simulator sensor. When we mean simple format we mean we don’t send JSON object inside JSON object like this.

**From php:**

if (isset($\_POST['patient']) && isset($\_POST['caretaker']) && isset($\_POST['sensor'])){

/\*

anycode code

\*/

echo json\_encode($response);

}

As you see, android need more code to send and receiving JSON not like php which few line.

**Connecting php with database:**

Our project need three query right now which these query are search, insert, and update.

**Query for check is patient name registration in database:**

$searchCTQuery = "SELECT \* FROM patienttable WHERE pUserName = '$patientName'";

//Check if User exist as patient

$result=mysql\_query($searchCTQuery);

if(mysql\_num\_rows($result) > 0){}

**Inserting new caretaker in database:**

//Insert

$quieryInsertCT = "INSERT INTO caretakertable (cUserName, password, email) VALUES('$name', '$password', '$email')";

$result = mysql\_query($quieryInsertCT);

// check if row inserted or not

if ($result){}

**Update single patient name for send data to caretaker:**

$updateQuery = "UPDATE patienttable SET cUserName= '$name' WHERE pUserName = '$patient2'";

$checkPExistQuery = "SELECT \* FROM patienttable WHERE pUserName = '$patient2' AND who\_will\_caring\_patient='$name'";

$result=mysql\_query($checkPExistQuery);

if(mysql\_num\_rows($result) > 0){

if(mysql\_query($updateQuery)){//Build Successed}

**Android:**

**1-How many activity?**

Because I don’t have good information in android I mean when it is good to have one activity or many activity. So I try to be in middle of them. I create five activities which theses activity are main activity you can call login activity also, registration activity, list activity, information activity and tips activity. Main activity has one function send user name and password to server to check if user name is registration in server and password match the user name. Registration activity has one function send user name, password, email, and what type of user caretaker or patient to server. List activity has two function show list view to caretaker include patient name and state and send patient name to server to add him\her to list view. Information activity show the current ECG, state, and activity and call emergency if patient has danger state. Tips activity has information about single type of arrhythmia and some tips to help patient if possible.

|  |  |
| --- | --- |
| MainActivity | RegistrationActivity |
|  |  |
| PatientListActivity | PatientDetailActivity |
|  |  |
| TipsActivity |  |
|  |  |

**2-Alert with edit text?**

I know how to make alert but with edit text no. so I looking in internet and find source code for it. There is no different between alert with edit text and without except write few code. Shown by red color

**Alert without edit text:**

AlertDialog.Builder alert = new AlertDialog.Builder(this);

alert.setTitle("connecting");

alert.setMessage("Who is the patient you want to take caring him\\her ");

alert.setPositiveButton("Ok", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int whichButton) {

// write your code

}

});

alert.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int whichButton) {

// Canceled.

}

});

alert.show();

**Alert with edit text:**

AlertDialog.Builder alert = new AlertDialog.Builder(this);

//alert.setTitle("connecting");

alert.setMessage("Who is the patient you want to take caring him\\her ");

// Set an EditText view to get user input

final EditText input = new EditText(this);

alert.setView(input);

alert.setPositiveButton("Ok", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int whichButton) {

String patientName = input.getText().toString();

// Do something with value!

}

});

alert.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int whichButton) {

// Canceled.

}

});

alert.show();

**3-Learning AsyncTask?**

Learning asyncTask was some kind difficult because it need to learn thread and process. Android’s thread sort job in thread if some job need more time the application crash. So, we need to handle this job by make new thread. There is different way to handle new thread like Handle, looper, AsyncTask, and Timer and TimerTask. I chose AsyncTask because that the suggestion form group and easy to understand but it has many code.

**AsyncTask:**

It has many functions but these are the main:

**onPreExecute:**Invoked before the task is executed ideally before doInBackground method is called on the UI thread. This method is normally used to setup the task like showing progress bar in the UI.

**doInBackground:**Code running for long lasting time should be put in doInBackground method. When execute method is called in UI main thread, this method is called with the parameters passed.

**onProgressUpdate:**Invoked by calling publishProgress at anytime from doInBackground. This method can be used to display any form of progress in the user interface.

**onPostExecute:**Invoked after background computation in doInBackground method completes processing. Result of the doInBackground is passed to this method.

For more information: http://programmerguru.com/android-tutorial/what-is-asynctask-in-android/

**4-Connecting to server and fetch data from database?**

I need to understand AsyncTask to connecting to server. Connecting to server by use get or post.

**Post:**

DefaultHttpClient httpClient = new DefaultHttpClient();

HttpPost httpPost = new HttpPost(url);

httpPost.setEntity(new UrlEncodedFormEntity(params));

HttpResponse httpResponse = httpClient.execute(httpPost);

**Get:**

DefaultHttpClient httpClient = new DefaultHttpClient();

HttpGet httpGet = new HttpGet(url);

HttpResponse httpResponse = httpClient.execute(httpGet);

Most code come from this link: http://programmerguru.com/android-tutorial/what-is-asynctask-in-android/

**5-Learning JSON:**

JSON is format to make to different device take to each other. There is other format like xml but JSON was easy to learn. I learn it from Abdullah Eid which I buy the course from his store. His store is in this link: https://asnadstore.com/abdullaheid.

**Make call from application?**

There is two different way. The first one need user permission. The second one does not need user permission.

**First one:**

Intent callIntent = new Intent(Intent.ACTION\_CALL, Uri.parse(uri));

startActivity(callIntent);

user permission

<uses-permission android:name="android.permission.CALL\_PHONE" />

**Second one:**

Intent dialIntent = new Intent(Intent.ACTION\_DIAL, Uri.parse(uri));

startActivity(dialIntent);

the different is first one call directly without change the number and the second one give user option to change the number.

**6-Draw chart in android?**

There is different chart like bar chart, line chart, and pie chart. The most important chart is line chart because ECG is line chart. Android by itself cannot draw chart. I need to download library for draw chart. I download aChartEngine. There are many option library but this was has more source and teaching. It has many class but these is more important.

**XYSeries:**

For create point in chart.

XYSeries row1=new XYSeries("row1");

row1.add(point,value);//add new point

**XYMultipleSeriesDataset:**

Add XYSeries point in chart.

XYMultipleSeriesDataset dataset=new XYMultipleSeriesDataset();

dataset.addSeries(row1);

**XYSeriesRenderer:**

Create line.

XYSeriesRenderer rendererRow1=new XYSeriesRenderer();

**XYMultipleSeriesRenderer:**

Create X and Y and add line in chrat.

XYMultipleSeriesRenderer multiRenderer=new XYMultipleSeriesRenderer();

multiRenderer.addSeriesRenderer(rendererRow1);

**GraphicalView:**

Install chart in view.

GraphicalView myChart= ChartFactory.getLineChartView(this, dataset, multiRenderer);

// Add chart to the layout

layout.addView(myChart);

also I need to add new line in **AndroidManifest**

<activity android:name="org.achartengine.GraphicalActivity"/>

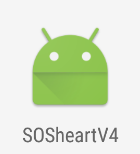
**Summary of this chapter:**

as you read, we don’t finish all project because the limitation in knowledge for heartbeat data and time. We still detect QRS complex which it is not good for detecting any kind of arrhythmia. We done connection between four resources by php which these resource are c++ code, android application, MySQL database, and sensor to make good system.

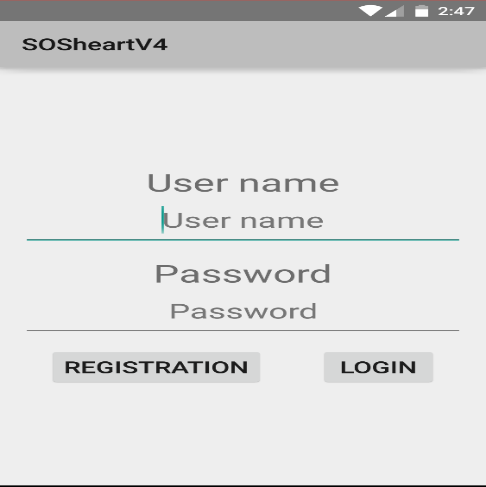
**Chapter Five: Testing**

**Between server, MySQL, and android application:**

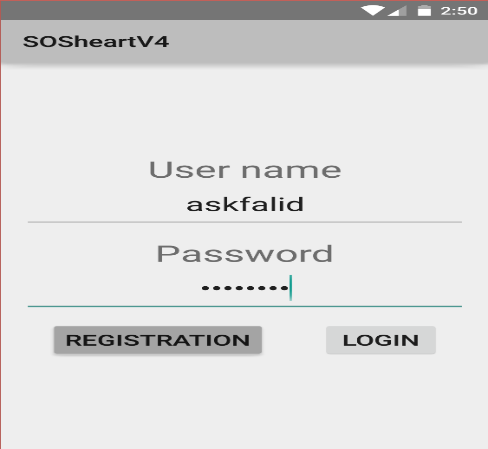
This is our icon when user click it:

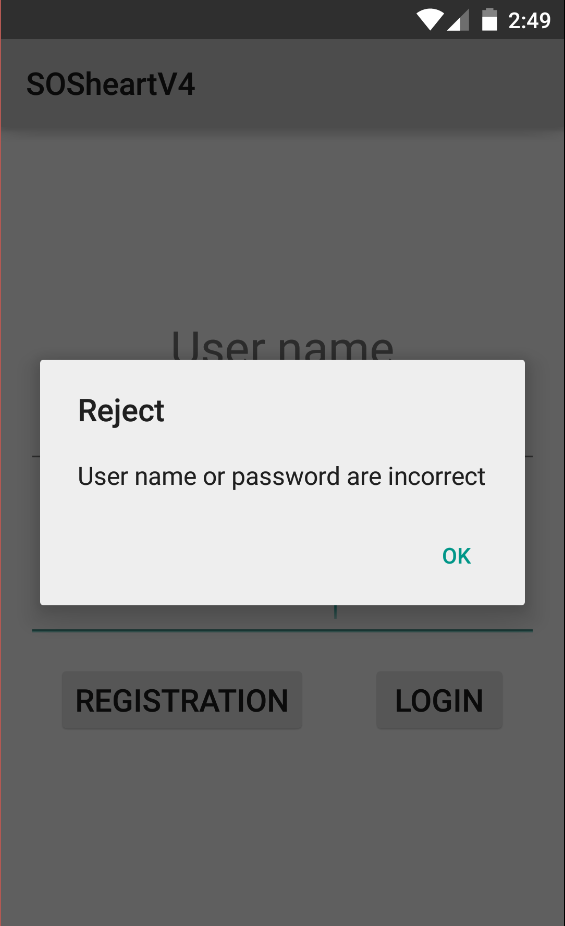


This is first Activity:

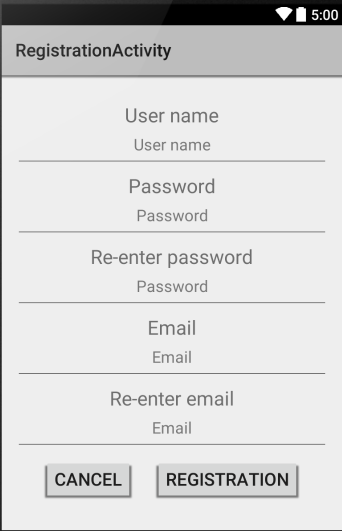


User enter wrong password or user name then click LOGIN button:

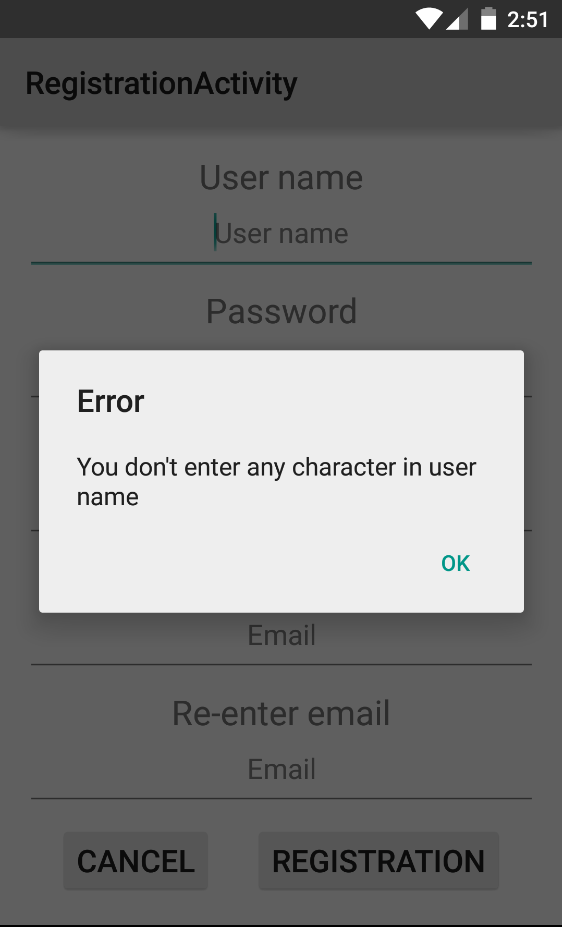




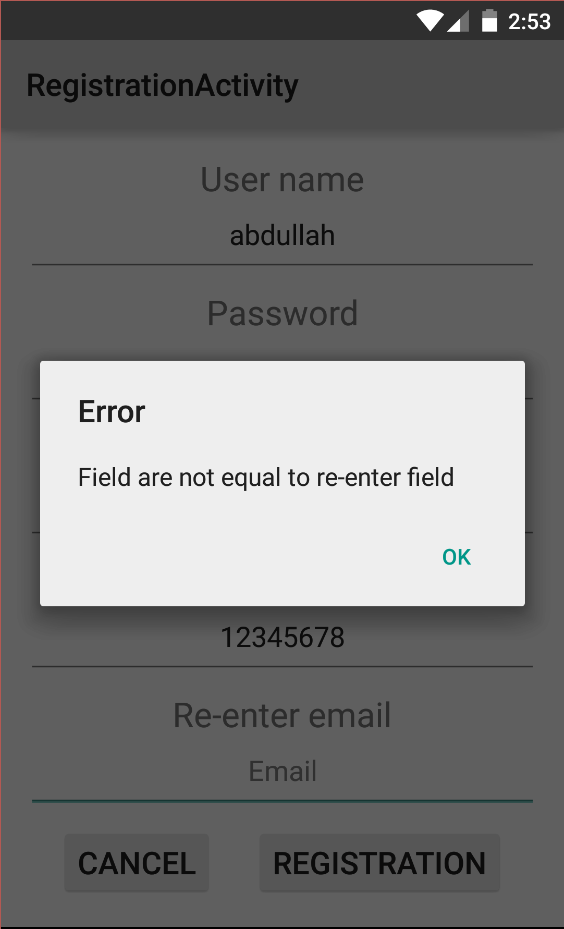
Click REGISTRATION button:



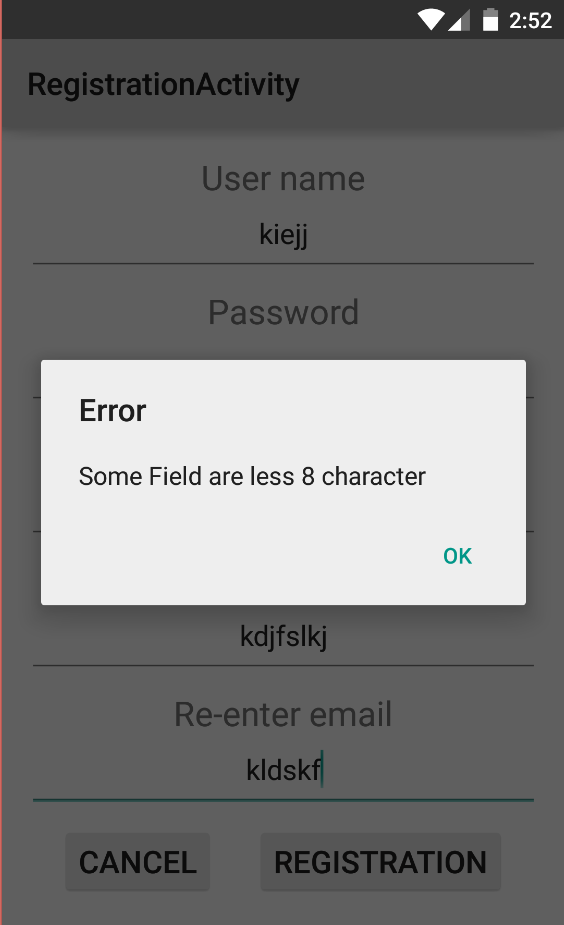
User don’t write any information in all field:



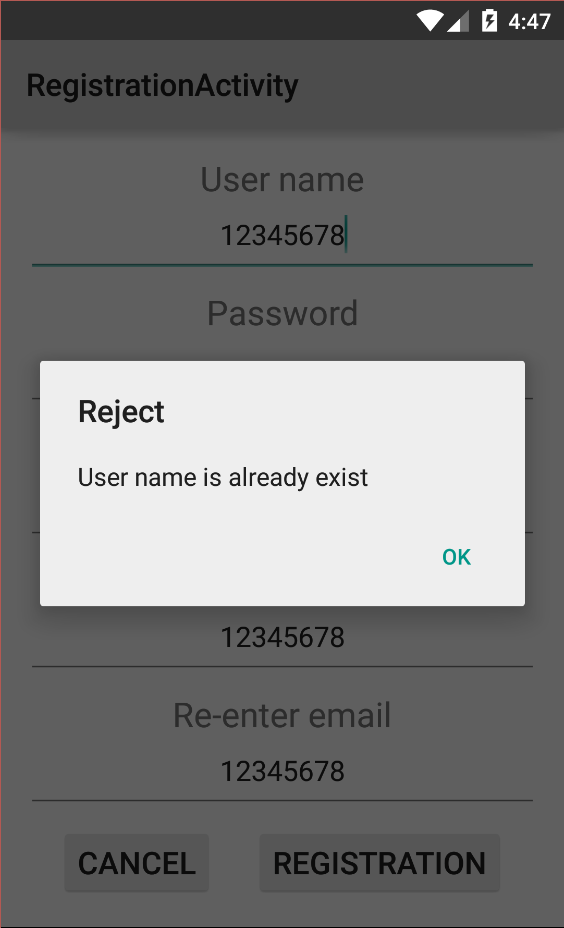
User don’t write in re-enter field:



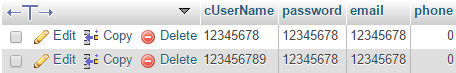
User write less than 8 character:



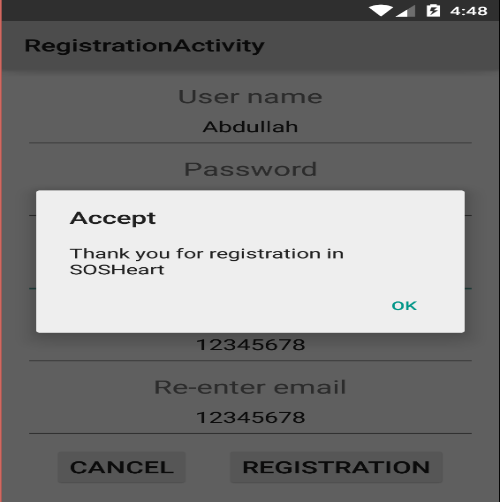
User enter user name is already exist:



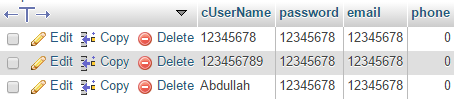
Database sosheartv1 in caretaker table at these point:



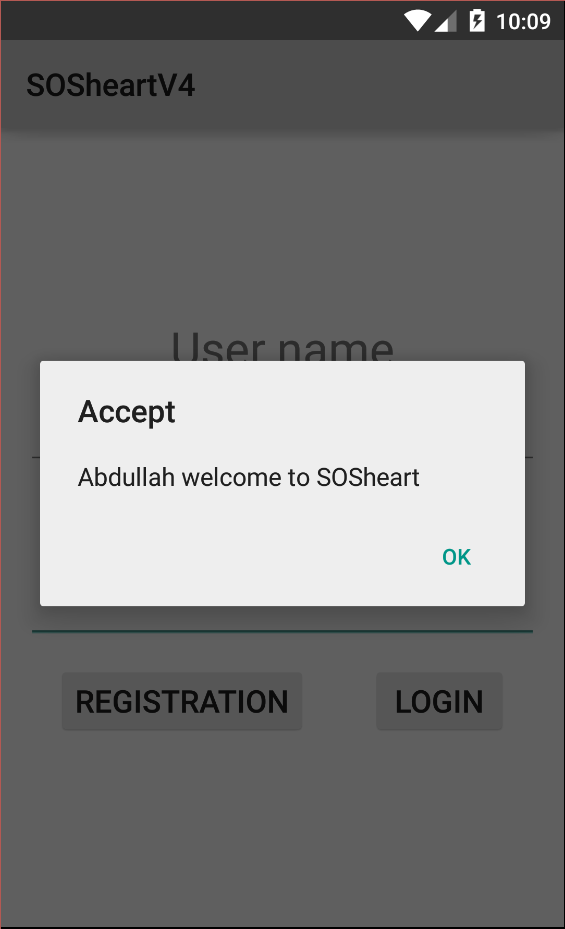
User write new user name and this name is Abdullah:



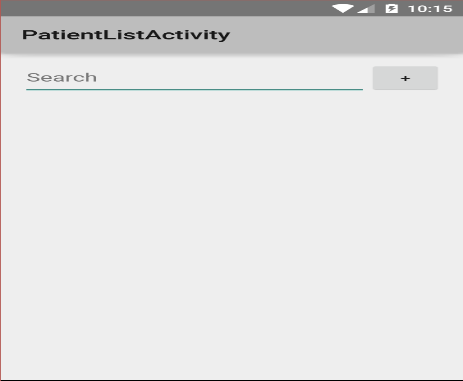
The database is update with new account:



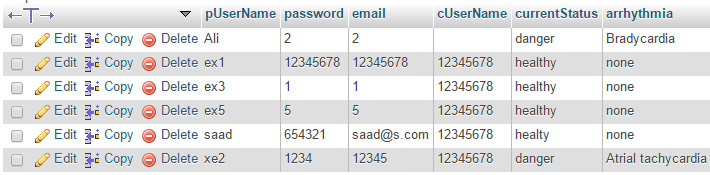
User write already user name with right password:



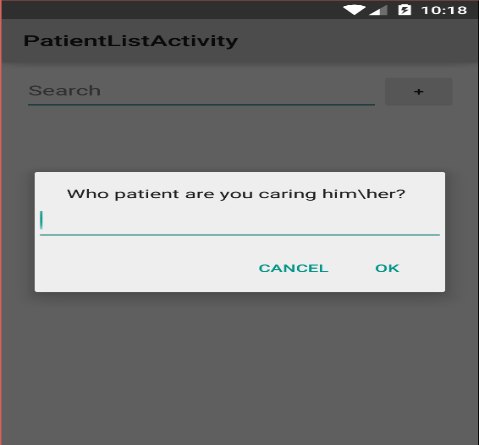
User Abdullah will have this activity:



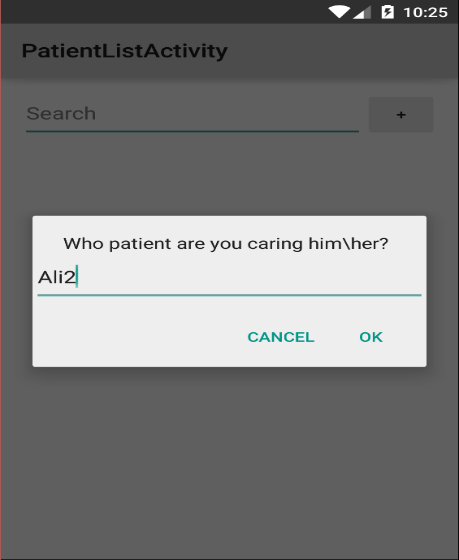
Since the database sosheartv1 in patient table:

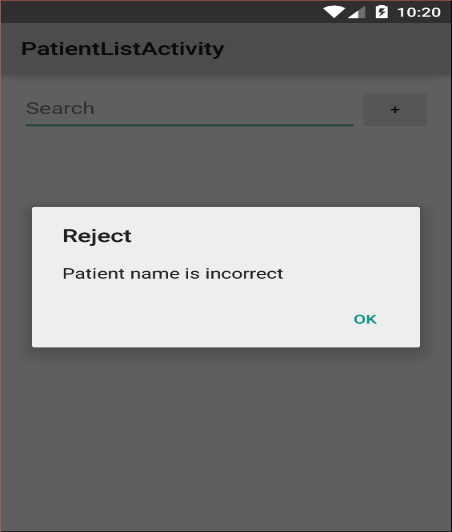


User Abdullah want to take care of patient:

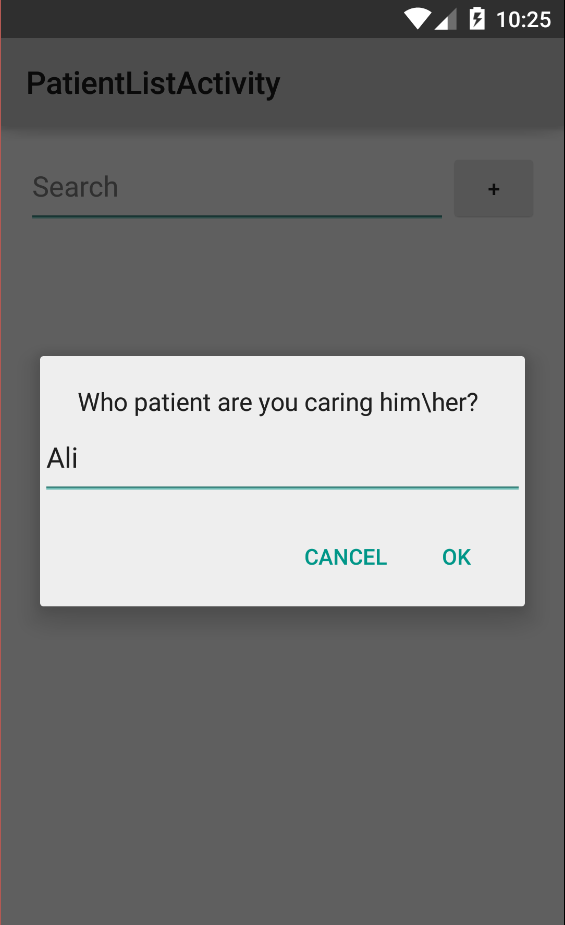


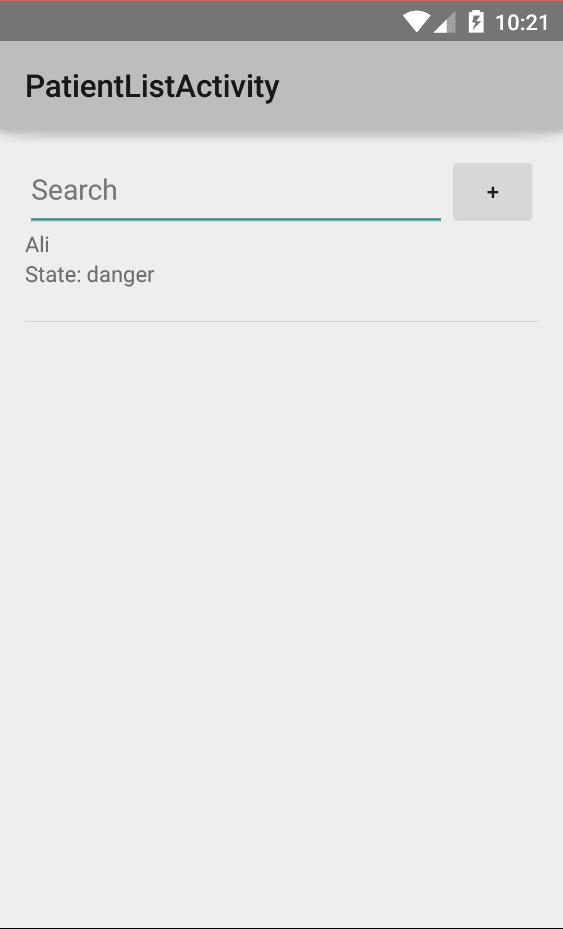
He enter wrong patient name:



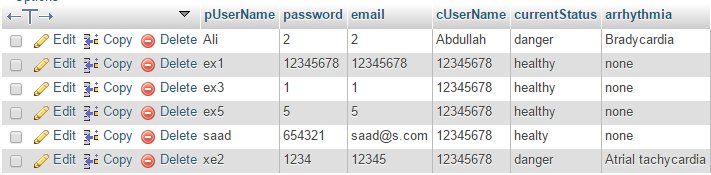


He enter patient name Ali:

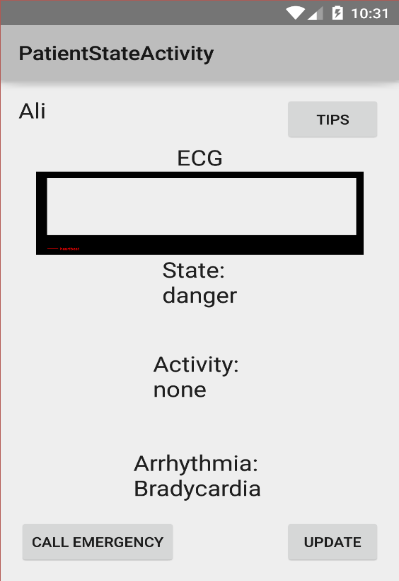




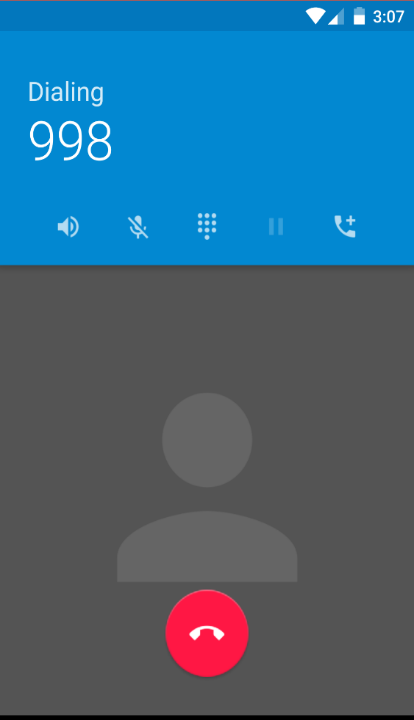
Update database in table patient:



He enter click in Ali to read Ali state:



Press CALL EMERGENCY button:

Local c++ code:

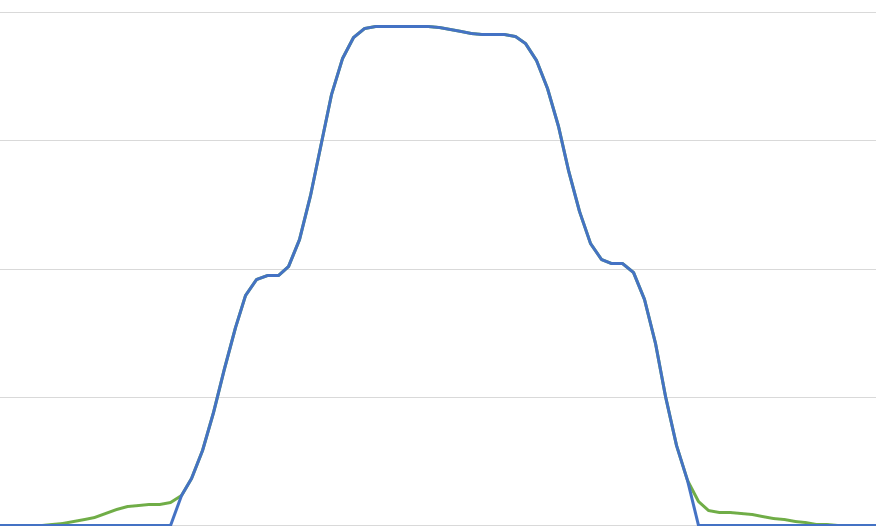
In this section we will show the result for detect QRS complex. As you read in previous chapter we show you how we make php execute c++. We show the result for QRS detect and comparing the result with matlab code in file 103 in total QRS.

|  |  |  |
| --- | --- | --- |
| Every minute | C++ | Matlab |
| 1 | 72 | 71 |
| 2 | 70 | 70 |
| 3 | 72 | 72 |
| 4 | 70 | 70 |
| 5 | 72 | 72 |
| 6 | 70 | 70 |
| 7 | 70 | 70 |
| 8 | 69 | 69 |
| 9 | 70 | 70 |
| 10 | 71 | 71 |

In this table, we comparing result every minute how our project detecting number of QRS. Comparing this result give us there is a problem in beginning of detect which this problem is detecting first p-wave in file 103. The error rate of our project is we just find error rate in different result.

Detecting single QRS complex:

Since we have problem in knowledge of heartbeat. We can’t calculate error rate of our project. This figure show different between after moving window integral and detecting QRS:



Green line is data after moving window integral

Blue line is data after detect QRS

There is data we can make it include to QRS complex because thresholds and noise.

Read date in real life:

Previous section show the result in local program. Our project make Simulator sensor to send data to QRS complex every minute. We build simulator by using android application and make it read data from file. We don’t know how to make read data then send frequently. We just make read data distance between two point or between two minute.