King Abdul-Aziz university
Faculty of Computing and Information Technology
Computer Science Department

February 13th, 2023

CPCS-371: Computer Networks

Simulating a Remote Health Monitoring System (RHMS) for elderly patients using client-server TCP Sockets

Instructors: Dr. Etimad Fadel and Dr.Ohoud Alzamzami

ID	Name	Section
[redacted]	[redacted]	B1
[redacted]	Amani Khalid Biraik	B1
[redacted]	[redacted]	B1
[redacted]	[redacted]	B1
[redacted]	Logain Ezzat Sendi	B2
[redacted]	[redacted]	B1

Table of Contents

Table of Figures

1 Introduction

1.1 Client-server applications

The client-server application consists of two parts which involve a client and a server. The client sends a message request, and the server responds with a message containing the services that were requested. This procedure is accomplished using a request-response pattern. Additionally, a communication protocol between the client and server is required to ensure the success of this procedure. Beyond that, the client and server can either run on the same computer or other computers and connect through a network.

1.2 Java TCP sockets

Transmission Control Protocol (TCP) is a transport layer protocol that provides a connection-oriented reliable data transfer. The sensor client application functions as a client in this program and uses TCP sockets to transfer data to the personal server. The personal server, however, has both a client and a server role. Prior to sending the data to the medical server, the personal server works as a server and accepts the data from the sensor client application. If the data was unusual, it would be forwarded to the medical server, where the personal server would act as the client, open a new TCP socket, and make the connection.

1.3 Selected GUIs elements

In our application, we used the java swing GUI application to create the interfaces for all applications (sensor client application, personal server, and medical server). We used labels, text fields, text areas, and a button.

1.4 RHMS application

A Remote health monitoring system (RHMS) is a system that is thoroughly used in the health department to solve medical issues, mostly used with elderly patients. Three distinct sensors in this device can detect oxygen saturation, heart rate, and temperature. Every five sends, the sensor senses data, and if a problem or abnormal state arises, the personal server must alert the medical server so that it may take the necessary steps to save the patient's life.

1.5 The role of the clients and servers in RHMS application

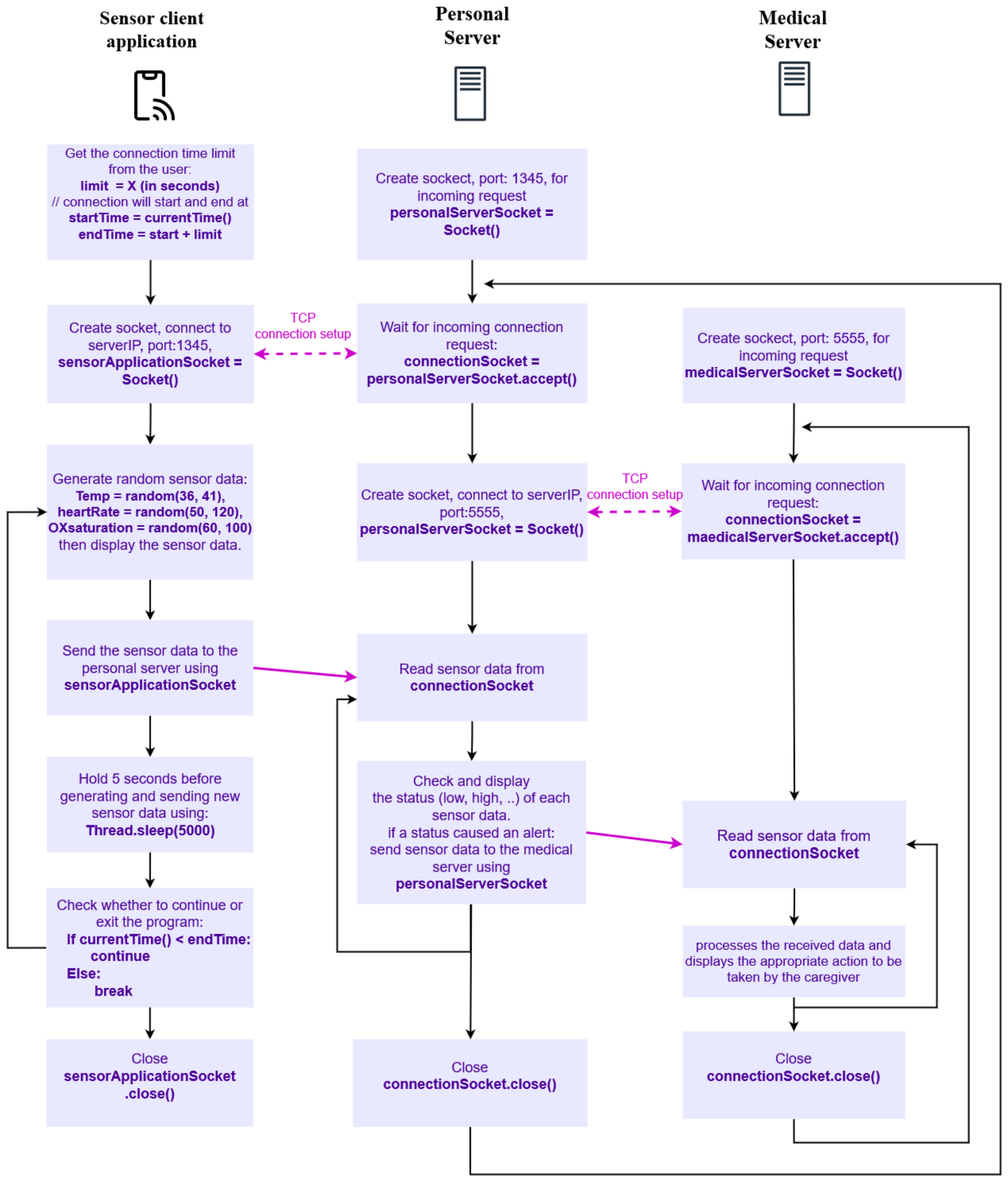
The sensor client application acts as a client in the RHMS application. It generates and collects the user's medical data every five seconds through sensors and sends them to the personal server using a TCP connection that is open for a minimum of 60 seconds. The personal server acts as both a client and a server. Its role is to receive the data from the sensor client application and process it to figure out whether the data sent is normal or abnormal (here, the personal server acts as a server). If the data is anomalous, a message is sent to the medical server (here, the personal server acts as a client). The medical server acts as a server only. Its role is to receive the messages sent from the personal server and display them to the caregiver while also processing the message to decide what action needs to be taken.

1.6 Overview

Section two presents the interaction diagram and pseudocode for the RHMS application. In section three, the implementation of our projection is displayed and explained. Section four, snapshots of the RHMS application being run on one machine and two machines, will be shown with details explaining each. Section five is about how the project was divided, what each girl did, and the lessons learned are mentioned. And finally, in section six, the conclusion will be presented.

2 Patient Monitoring Application interaction diagram

2.1 Interaction diagram


Sensor client application
Personal Server
Medical Server
Get the connection time limit from the user:
limit = X (in seconds)
// connection will start and end at
startTime = currentTime()
endTime = start + limit
Create sockect, port: 1345, for incoming request
personalServerSocket = Socket()
Create socket, connect to serverIP, port:1345,
sensorApplicationSocket = Socket()
TCP connection setup
Wait for incoming connection request:
connectionSocket = personalServerSocket.accept()
Create sockect, port: 5555, for incoming request
medicalServerSocket = Socket()
Generate random sensor data:
Temp = random(36, 41),
heartRate = random(50, 120),
OXsaturation = random(60, 100)
then display the sensor data.
Create socket, connect to serverIP, port:5555,
personalServerSocket = Socket()
TCP connection setup
Wait for incoming connection request:
connectionSocket = maedicalServerSocket.accept()
Send the sensor data to the personal server using sensorApplicationSocket
Read sensor data from connectionSocket
Hold 5 seconds before generating and sending new sensor data using:
Thread.sleep(5000)
Check and display the status (low, high, ..) of each sensor data.
if a status caused an alert: send sensor data to the medical server using
personalServerSocket
Read sensor data from connectionSocket
Check whether to continue or exit the program:
If currentTime() < endTime:
continue
Else:
break
processes the received data and displays the appropriate action to be taken by the caregiver
Close sensorApplicationSocket.close()
Close connectionSocket.close()
Close connectionSocket.close()

3 RHMS implementation

3.1 Sensor (Client) Application

```
import java.io.BufferedWriter;
 import java.io.IOException;
 import java.io.OutputStreamWriter;
 import java.net.Socket;
 import java.text.SimpleDateFormat;
 import java.util.Date;
 import java.util.Scanner;
 import java.util.concurrent.ThreadLocalRandom;

public class Sensor_client_application {

// initiate variables
 static int heartRateSensor = 0;
 static double TempSensorDouble = 0;
 static String TempSensorString = "";
 static int O2Sensor = 0;
 static int limit = 0;
 static Socket socket = null;
 static long startTime = 0;
 static long endTime = 0;

public static void main(String[] args) throws IOException, InterruptedException {
 Scanner input = new Scanner(System.in);
 // ask the user to enter the desired running time of the program
 System.out.print("Enter connection time wanted in seconds, 60s minimum: ");
 limit = input.nextInt();

// the minimum time is 60 seconds, therefore, check if the user entered a time less than 60s
 if (limit < 60) {
 limit = 60; }

// determine the start and end time of the program
 startTime = System.currentTimeMillis();
 endTime = startTime + limit * 1000;

// create the client TCP socket
socket = new Socket("localhost", 1345);

// for sending the data to the personal server
OutputStreamWriter output = new OutputStreamWriter(socket.getOutputStream());
 BufferedWriter writer = new BufferedWriter(output);

while (System.currentTimeMillis() < endTime) {
```

```
// generate random data
 // Heart rate sensor is between 50 and 120 heart beats each minute
 heartRateSensor = Data(50, 120);
 // Temperature sensor data is between 36 and 41
 TempSensorDouble = DataDouble(36, 41);
 // Oxygen level in the blood data is sent by the sensor with values between 60 to 100
 O2Sensor = Data(60, 100);
 // formating the date and time
 SimpleDateFormat formatter = new SimpleDateFormat("dd MMM yy 'time' HH:mm:ss");
 Date date = new Date();

// coverting the double value of temperature to a string value
 TempSensorString = String.valueOf(TempSensorDouble);
 // take only one digit after the decimal point
 TempSensorString = TempSensorString.substring(0, 4);

// print data
 System.out.println("");
 System.out.println("At date: " + formatter.format(date) + ",sensed temperature is " + TempSensorString);
 System.out.println("At date: " + formatter.format(date) + ",sensed heart rate is " + heartRateSensor);
 System.out.println("At date: " + formatter.format(date) + ",sensed oxygen saturation is " + O2Sensor);

// send the data to the personal server
 writer.write(heartRateSensor);
 writer.write(O2Sensor);
 writer.write(TempSensorString);
 writer.newLine();
 writer.flush();

// the program must hold for a 5s before the next iteration
 Thread.sleep(5000); }

socket.close(); }

// this method generates an integer random number between the two values min and max
 public static int Data(int min, int max) {

int random_int = (int) Math.floor(Math.random() * (max - min + 1) + min);

return random_int; }

// this method generates a double random number between the two values min and max
 public static double DataDouble(int min, int max) {
 return (ThreadLocalRandom.current().nextDouble() * (max - min)) + min; } }
```

3.2 Personal Server

```
// personal server: recieves sensored data from the client application, then checks the user's medical status and send data to medical server if neeeded.
 package personal_server;

import java.io.BufferedReader;
 import java.io.BufferedWriter;
 import java.io.IOException;
 import java.io.InputStreamReader;
 import java.io.OutputStreamWriter;
 import java.net.ServerSocket;
 import java.net.Socket;
 import java.text.SimpleDateFormat;
 import java.util.Date;

public class Personal_server {

public static void main(String[] args) throws IOException, InterruptedException {

// create the personal server TCP socket
 ServerSocket serverSocket = new ServerSocket(1345);
 SimpleDateFormat formatter = new SimpleDateFormat("dd MMM yy 'time' HH:mm:ss");

//1st loop
 while (true) { // the server always on

// accept the socket CONNECTION
 Socket socket = serverSocket.accept();

// medical server socket
 Socket medicalSocket = new Socket("localhost", 5555);

// to write data to medical server
 OutputStreamWriter medicalOutput = new OutputStreamWriter(medicalSocket.getOutputStream());
 BufferedWriter medicalWriter = new BufferedWriter(medicalOutput);

// initiate variables
 double receiveTemp;
 String TempString = "";
 int receiveHeartRate;
 int receiveOxygenSaturation;

// an alert variable is used to determine whether to send data to medical server or not
 boolean alert = false;

// 2nd loop
 while (true) {
```

```
// to read from the client application class
 InputStreamReader input = new InputStreamReader(socket.getInputStream());
 BufferedReader reader = new BufferedReader(input);

// read data from client application class
 receiveHeartRate = reader.read();
 receiveOxygenSaturation = reader.read();
 TempString = reader.readLine();


// if the client application is terminated then the default data to be send is -1, so this method will check whether
to end the connection or not
 if (receiveHeartRate == -1) {
 break; }

// make sure to take only one digit after the decimal point of the temperature value
 TempString = TempString.substring(0, 4);


// convert the String value of temperature to a double value
 receiveTemp = Double.valueOf(TempString);

Date date = new Date();

// check temperature
 if (checkTemp(receiveTemp)) {

System.out.println("At date: " + formatter.format(date) + ", Temperature is normal");

} else {
 alert = true;
 System.out.println("At date: " + formatter.format(date) + ", Temperature is high " + receiveTemp + ". An alert
message is sent to the Medical Server."); }
 //-------------------------------------------------------------------------------

// check heart rate
 String heartRate = checkHeartRate(receiveHeartRate);
 if (heartRate.equalsIgnoreCase("Above")) {

alert = true;
 System.out.println("At date: " + formatter.format(date) + ", Hear rate is above normal " + receiveHeartRate +
". An alert message is sent to the Medical Server.");

} else if (heartRate.equalsIgnoreCase("Below")) {

alert = true;
 System.out.println("At date: " + formatter.format(date) + ", Hear rate is below normal " + receiveHeartRate +
". An alert message is sent to the Medical Server.");

} else {

System.out.println("At date: " + formatter.format(date) + ", Hear rate is normal"); }
```

```
//-----------------------------------------------------------------------------------------
 // check oxygen saturation
 if (checkOxygenSaturation(receiveOxygenSaturation)) {

System.out.println("At date: " + formatter.format(date) + ", Oxygen Saturation is normal");

} else {

alert = true;

System.out.println("At date: " + formatter.format(date) + ", Oxygen Saturation is low " +
receiveOxygenSaturation + ". An alert message is sent to the Medical Server."); }
 //-----------------------------------------------------------------------------------------

System.out.println("");
 // check if the alert is true to send data to medical server
 if (alert) {
medicalWriter.write(1);
 medicalWriter.write(receiveHeartRate);
 medicalWriter.write(receiveOxygenSaturation);
 medicalWriter.write(TempString);
 medicalWriter.newLine();
 medicalWriter.flush(); }
// the program must hold for a 5s before the next iteration
 Thread.sleep(5000); }//end of 2nd loop

socket.close(); }// end of 1st loop  }

  public static boolean checkTemp(double temp) {

if (temp > 37) {
 return false;
 } else {
 return true; } }

// check heart rate method
 public static String checkHeartRate(int heartRate) {

if (heartRate > 100) {
 return "Above";
 } else if (heartRate < 60) {
 return "Below";
 } else {
 return "Normal"; } }

// check oxygen saturation method
 public static boolean checkOxygenSaturation(int oxygenSaturation) {

if (oxygenSaturation < 75) {
 return false;
 } else {
 return true; } } }
```

3.3 Medical Server

```
// medical server: receives the data from the personal server to process abnormal cases.

package medical_server;

import java.io.BufferedReader;
 import java.io.IOException;
 import java.io.InputStreamReader;
 import java.net.ServerSocket;
 import java.net.Socket;
 import java.text.SimpleDateFormat;
 import java.util.Date;

public class Medical_server {

public static void main(String[] args) throws IOException, InterruptedException {

// create the medical server TCP socket
 ServerSocket MedicalServerSocket = new ServerSocket(5555);

SimpleDateFormat formatter = new SimpleDateFormat("dd MMM yy 'time' HH:mm:ss");
 //1st loop
 while (true) {//the server always on

Socket socket = MedicalServerSocket.accept(); // accept the socket

// initiate the variables
 int alert = -1;
double receiveTemp;
 String TempString = "";
 int receiveHeartRate;
 int receiveOxygenSaturation;

while (true) {

// to read from the personal server class
 InputStreamReader medicalInput = new InputStreamReader(socket.getInputStream());
 BufferedReader medicalReader = new BufferedReader(medicalInput);


// if there is an alert from the personal server this value must be read
 alert = medicalReader.read();


// if the personal server detect any abnormal condition the alert will be send to the medical server (this server)
 if (alert == 1) {


// read the temperature, heart rate, and oxygen saturation that sent by the personal server
receiveHeartRate = medicalReader.read();
 receiveOxygenSaturation = medicalReader.read();
 TempString = medicalReader.readLine();
// if the client application is terminated then the default data to be send is -1, so this method will check whether
to end the connection or not
 if (receiveHeartRate == -1) {
 break;
 }

//take only one digit after the decimal point of the temperature value
 TempString = TempString.substring(0, 4);

// convert the String value of temperature to a double value
 receiveTemp = Double.valueOf(TempString);
 Date date = new Date();

// If the temperature exceeds 39 and heart rate is above 100 and oxygen is below 95
 if ((receiveTemp > 39) && (receiveHeartRate > 100) && (receiveOxygenSaturation < 95)) {
 System.out.println("At date: " + formatter.format(date) + ", Temperature is high " + receiveTemp + ".");
 System.out.println("At date: " + formatter.format(date) + ", Hear rate is above normal " + receiveHeartRate +
".");
 System.out.println("At date: " + formatter.format(date) + ", Oxygen Saturation is low" +
receiveOxygenSaturation + ".");
 System.out.println("ACTION: Send an ambulance to the patient!");
 System.out.println("");
 } // If the temperature is between 38 and 38.9, and the heart rate is between 95 and 98, and oxygen is below 80
 else if (((receiveTemp >= 38) && (receiveTemp <= 38.9)) && ((receiveHeartRate >= 95) &&
(receiveHeartRate <= 98)) && (receiveOxygenSaturation < 80)) {
 System.out.println("At date: " + formatter.format(date) + ", Temperature is high " + receiveTemp + ".");
 System.out.println("At date: " + formatter.format(date) + ", Hear rate is normal " + receiveHeartRate + ".");
 System.out.println("At date: " + formatter.format(date) + ", Oxygen Saturation is low" +
receiveOxygenSaturation + ".");
 System.out.println("ACTION: Call the patient's family!");
 System.out.println("");
 } // default message for the caregiver
 else {
 System.out.println("At date: " + formatter.format(date) + ", Temperature is " + receiveTemp + ".");
 System.out.println("At date: " + formatter.format(date) + ", Hear rate is " + receiveHeartRate + ".");
 System.out.println("At date: " + formatter.format(date) + ", Oxygen Saturation is " + receiveOxygenSaturation
+ ".");
 System.out.println("ACTION: Warning, advise patient to make a checkup appointment!");
 System.out.println(""); } }

// the program must hold for a 5s before the next iteration
 Thread.sleep(5000);  }

// closing the connections
 socket.close();  }   } }
```

3.4 Where are TCP sockets used?

TCP sockets are used in the client sensor application, to connect to the servers. TCP sockets are also used in the personal and medical servers to listen to the connection, as well as request and send messages. Personal server sends messages to the medical server through the TCP socket, and the medical server receives those messages through its TCP socket. (colored blue in the code comments above)

3.5 Where are threads used? For what reason are they used?

A program's single sequential control flow is known as a thread. A single thread is how most programs operate. We may run many processes simultaneously by using Java's multithreading feature, which enables the server to support multiple clients, by accepting a client connection, opening a thread for that conversation, and keeping an eye out for requests from other clients.

3.6 What are the functions used for sending and receiving information using sockets?

(Colored pink in code comments above)

1) OutputStreamWriter and BufferedWriter 2) writer.write(data)	In client sensor for sending the data to the personal server, and in personal for sending the data to the medical server
1) InputStreamReader and BufferedReader 2) reader.read(data)	in personal server for reading data from sensor client application class, and in medical server for reading data from personal server

3.7 What are the GUIs elements used in your application?

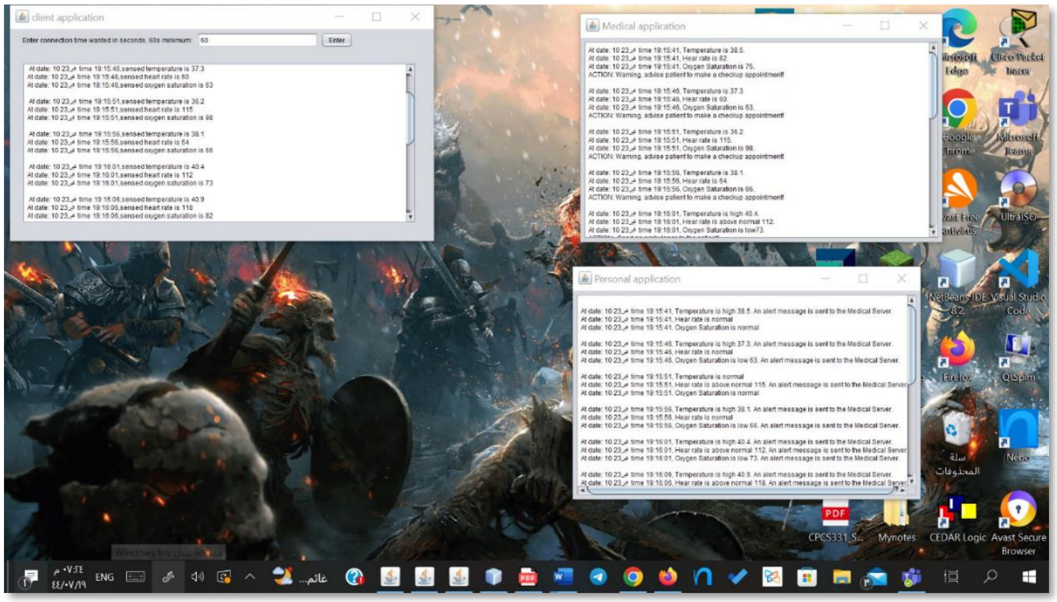
- Labels
- Text field
- Button
- Text Area

```
// this methos is responsible for taken the action of the input after the user clicks it
private void EnterActionPerformed(java.awt.event.ActionEvent evt)
// this line is used to get the entered input from the user
  limit = Integer.parseInt(jTextField1.getText());
        if (limit < 60) {
           limit = 60; }
// is to print the result into the text area as it appears in Image1
   object.append
```

4 RHMS Application runs snapshots.

4.1 run of the program on one machine

1. the user will enter the required time in the text field.
2. The Sensor will start to send the information every 5 seconds to Personal server as it's appearing in Image 1
3. The personal server will print the result and send the alert to the medical server.
4. The Medical server will output the result and print it on the screen.

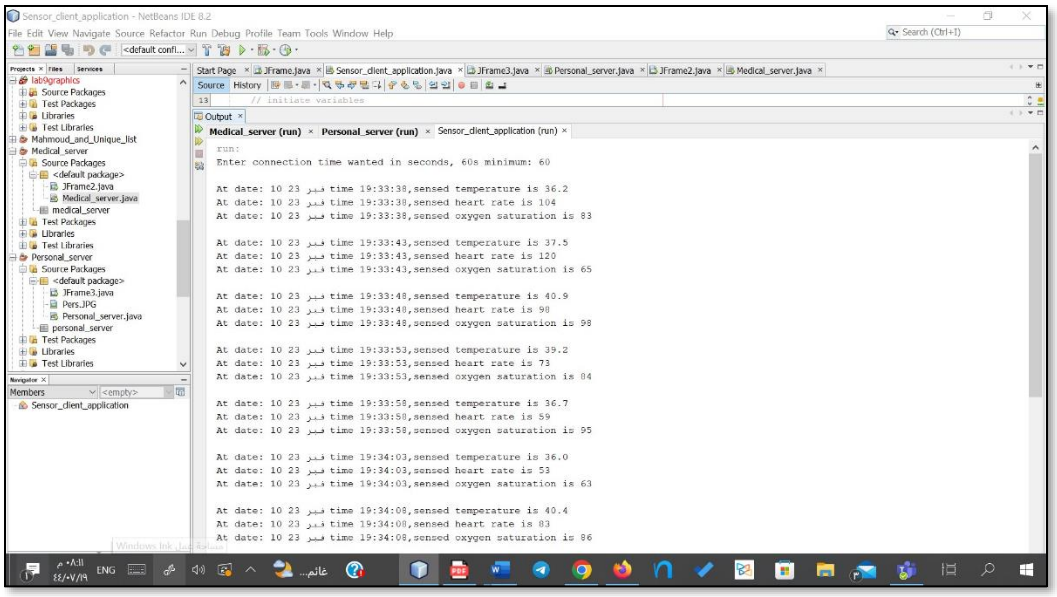


```
At date: 10 23 فبر time 19:15:46,sensed temperature is 37.3
At date: 10 23 فبر time 19:15:46,sensed heart rate is 60
At date: 10 23 فبر time 19:15:46,sensed oxygen saturation is 63

At date: 10 23 فبر time 19:15:51,sensed temperature is 36.2
At date: 10 23 فبر time 19:15:51,sensed heart rate is 115
At date: 10 23 فبر time 19:15:51,sensed oxygen saturation is 98

At date: 10 23 فبر time 19:15:56,sensed temperature is 38.1
At date: 10 23 فبر time 19:15:56,sensed heart rate is 64
At date: 10 23 فبر time 19:15:56,sensed oxygen saturation is 66

At date: 10 23 فبر time 19:16:01,sensed temperature is 40.4
At date: 10 23 فبر time 19:16:01,sensed heart rate is 112
At date: 10 23 فبر time 19:16:01,sensed oxygen saturation is 73

At date: 10 23 فبر time 19:16:06,sensed temperature is 40.9
At date: 10 23 فبر time 19:16:06,sensed heart rate is 118
At date: 10 23 فبر time 19:16:06,sensed oxygen saturation is 82
```

```
At date: 10 23 فبر time 19:15:41, Temperature is 38.5.
At date: 10 23 فبر time 19:15:41, Hear rate is 82.
At date: 10 23 فبر time 19:15:41, Oxygen Saturation is 75.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 23 فبر time 19:15:46, Temperature is 37.3.
At date: 10 23 فبر time 19:15:46, Hear rate is 60.
At date: 10 23 فبر time 19:15:46, Oxygen Saturation is 63.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 23 فبر time 19:15:51, Temperature is 36.2.
At date: 10 23 فبر time 19:15:51, Hear rate is 115.
At date: 10 23 فبر time 19:15:51, Oxygen Saturation is 98.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 23 فبر time 19:15:56, Temperature is 38.1.
At date: 10 23 فبر time 19:15:56, Hear rate is 64.
At date: 10 23 فبر time 19:15:56, Oxygen Saturation is 66.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 23 فبر time 19:16:01, Temperature is high 40.4.
At date: 10 23 فبر time 19:16:01, Hear rate is above normal 112.
At date: 10 23 فبر time 19:16:01, Oxygen Saturation is low73.
```

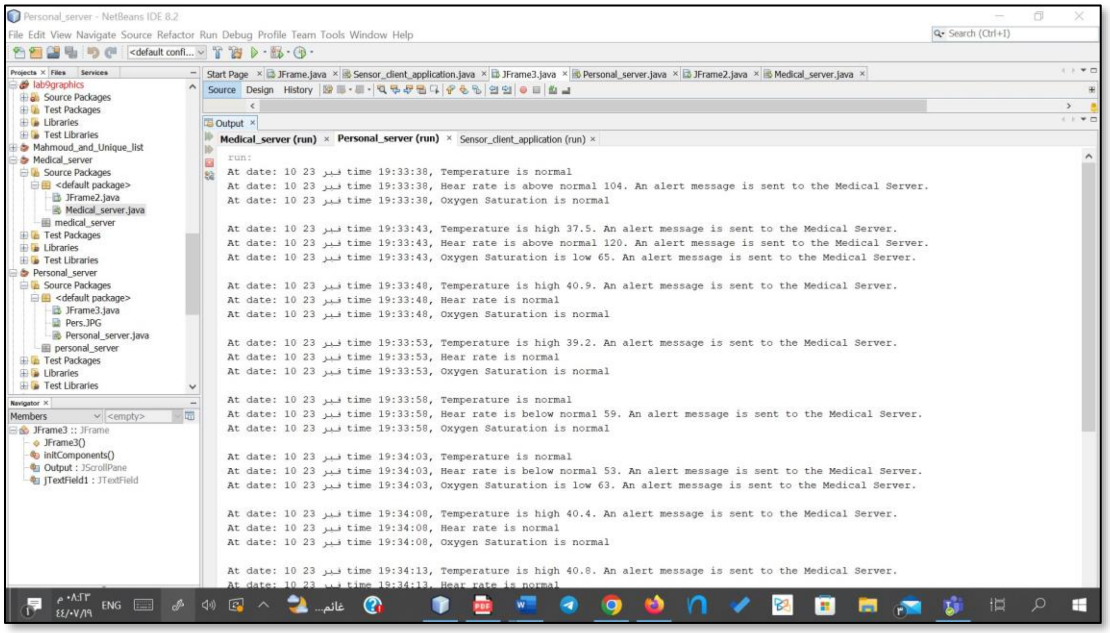
```
At date: 10 23 فبر time 19:15:41, Temperature is high 38.5. An alert message is sent to the Medical Server.
At date: 10 23 فبر time 19:15:41, Hear rate is normal
At date: 10 23 فبر time 19:15:41, Oxygen Saturation is normal

At date: 10 23 فبر time 19:15:46, Temperature is high 37.3. An alert message is sent to the Medical Server.
At date: 10 23 فبر time 19:15:46, Hear rate is normal
At date: 10 23 فبر time 19:15:46, Oxygen Saturation is low 63. An alert message is sent to the Medical Server.

At date: 10 23 فبر time 19:15:51, Temperature is normal
At date: 10 23 فبر time 19:15:51, Hear rate is above normal 115. An alert message is sent to the Medical Server
At date: 10 23 فبر time 19:15:51, Oxygen Saturation is normal

At date: 10 23 فبر time 19:15:56, Temperature is high 38.1. An alert message is sent to the Medical Server.
At date: 10 23 فبر time 19:15:56, Hear rate is normal
At date: 10 23 فبر time 19:15:56, Oxygen Saturation is low 66. An alert message is sent to the Medical Server.

At date: 10 23 فبر time 19:16:01, Temperature is high 40.4. An alert message is sent to the Medical Server.
At date: 10 23 فبر time 19:16:01, Hear rate is above normal 112. An alert message is sent to the Medical Server
At date: 10 23 فبر time 19:16:01, Oxygen Saturation is low 73. An alert message is sent to the Medical Server.

At date: 10 23 فبر time 19:16:06, Temperature is high 40.9. An alert message is sent to the Medical Server.
At date: 10 23 فبر time 19:16:06, Hear rate is above normal 118. An alert message is sent to the Medical Server
```

Figure 1: run in GUI 3 classes.



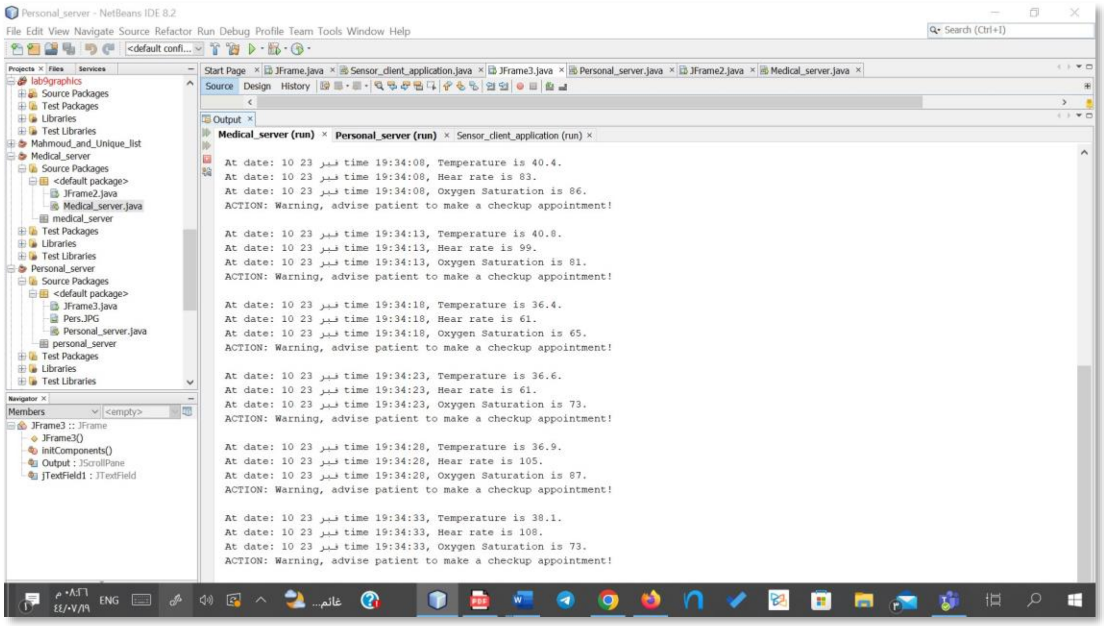
```
run:
Enter connection time wanted in seconds, 60s minimum: 60

At date: 10 23 فبر time 19:33:38,sensed temperature is 36.2
At date: 10 23 فبر time 19:33:38,sensed heart rate is 104
At date: 10 23 فبر time 19:33:38,sensed oxygen saturation is 83

At date: 10 23 فبر time 19:33:43,sensed temperature is 37.5
At date: 10 23 فبر time 19:33:43,sensed heart rate is 120
At date: 10 23 فبر time 19:33:43,sensed oxygen saturation is 65

At date: 10 23 فبر time 19:33:48,sensed temperature is 40.9
At date: 10 23 فبر time 19:33:48,sensed heart rate is 98
At date: 10 23 فبر time 19:33:48,sensed oxygen saturation is 98

At date: 10 23 فبر time 19:33:53,sensed temperature is 39.2
At date: 10 23 فبر time 19:33:53,sensed heart rate is 73
At date: 10 23 فبر time 19:33:53,sensed oxygen saturation is 84

At date: 10 23 فبر time 19:33:58,sensed temperature is 36.7
At date: 10 23 فبر time 19:33:58,sensed heart rate is 59
At date: 10 23 فبر time 19:33:58,sensed oxygen saturation is 95

At date: 10 23 فبر time 19:34:03,sensed temperature is 36.0
At date: 10 23 فبر time 19:34:03,sensed heart rate is 53
At date: 10 23 فبر time 19:34:03,sensed oxygen saturation is 63

At date: 10 23 فبر time 19:34:08,sensed temperature is 40.4
At date: 10 23 فبر time 19:34:08,sensed heart rate is 83
At date: 10 23 فبر time 19:34:08,sensed oxygen saturation is 86
```

Figure 2: run in Sensor console.



```
run:
At date: 10 23 فبر time 19:33:38, Temperature is normal
At date: 10 23 فبر time 19:33:38, Hear rate is above normal 104. An alert message is sent to the Medical Server.
At date: 10 23 فبر time 19:33:38, Oxygen Saturation is normal

At date: 10 23 فبر time 19:33:43, Temperature is high 37.5. An alert message is sent to the Medical Server.
At date: 10 23 فبر time 19:33:43, Hear rate is above normal 120. An alert message is sent to the Medical Server.
At date: 10 23 فبر time 19:33:43, Oxygen Saturation is low 65. An alert message is sent to the Medical Server.

At date: 10 23 فبر time 19:33:48, Temperature is high 40.9. An alert message is sent to the Medical Server.
At date: 10 23 فبر time 19:33:48, Hear rate is normal
At date: 10 23 فبر time 19:33:48, Oxygen Saturation is normal

At date: 10 23 فبر time 19:33:53, Temperature is high 39.2. An alert message is sent to the Medical Server.
At date: 10 23 فبر time 19:33:53, Hear rate is normal
At date: 10 23 فبر time 19:33:53, Oxygen Saturation is normal

At date: 10 23 فبر time 19:33:58, Temperature is normal
At date: 10 23 فبر time 19:33:58, Hear rate is below normal 59. An alert message is sent to the Medical Server.
At date: 10 23 فبر time 19:33:58, Oxygen Saturation is normal

At date: 10 23 فبر time 19:34:03, Temperature is normal
At date: 10 23 فبر time 19:34:03, Hear rate is below normal 53. An alert message is sent to the Medical Server.
At date: 10 23 فبر time 19:34:03, Oxygen Saturation is low 63. An alert message is sent to the Medical Server.

At date: 10 23 فبر time 19:34:08, Temperature is high 40.4. An alert message is sent to the Medical Server.
At date: 10 23 فبر time 19:34:08, Hear rate is normal
At date: 10 23 فبر time 19:34:08, Oxygen Saturation is normal

At date: 10 23 فبر time 19:34:13, Temperature is high 40.8. An alert message is sent to the Medical Server.
At date: 10 23 فبر time 19:34:13, Hear rate is normal
```

Figure 3:run in Personal Server console.



```
At date: 10 23 فبر time 19:34:08, Temperature is 40.4.
At date: 10 23 فبر time 19:34:08, Hear rate is 83.
At date: 10 23 فبر time 19:34:08, Oxygen Saturation is 86.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 23 فبر time 19:34:13, Temperature is 40.8.
At date: 10 23 فبر time 19:34:13, Hear rate is 99.
At date: 10 23 فبر time 19:34:13, Oxygen Saturation is 81.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 23 فبر time 19:34:18, Temperature is 36.4.
At date: 10 23 فبر time 19:34:18, Hear rate is 61.
At date: 10 23 فبر time 19:34:18, Oxygen Saturation is 65.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 23 فبر time 19:34:23, Temperature is 36.6.
At date: 10 23 فبر time 19:34:23, Hear rate is 61.
At date: 10 23 فبر time 19:34:23, Oxygen Saturation is 73.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 23 فبر time 19:34:28, Temperature is 36.9.
At date: 10 23 فبر time 19:34:28, Hear rate is 105.
At date: 10 23 فبر time 19:34:28, Oxygen Saturation is 87.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 23 فبر time 19:34:33, Temperature is 38.1.
At date: 10 23 فبر time 19:34:33, Hear rate is 108.
At date: 10 23 فبر time 19:34:33, Oxygen Saturation is 73.
ACTION: Warning, advise patient to make a checkup appointment!
```

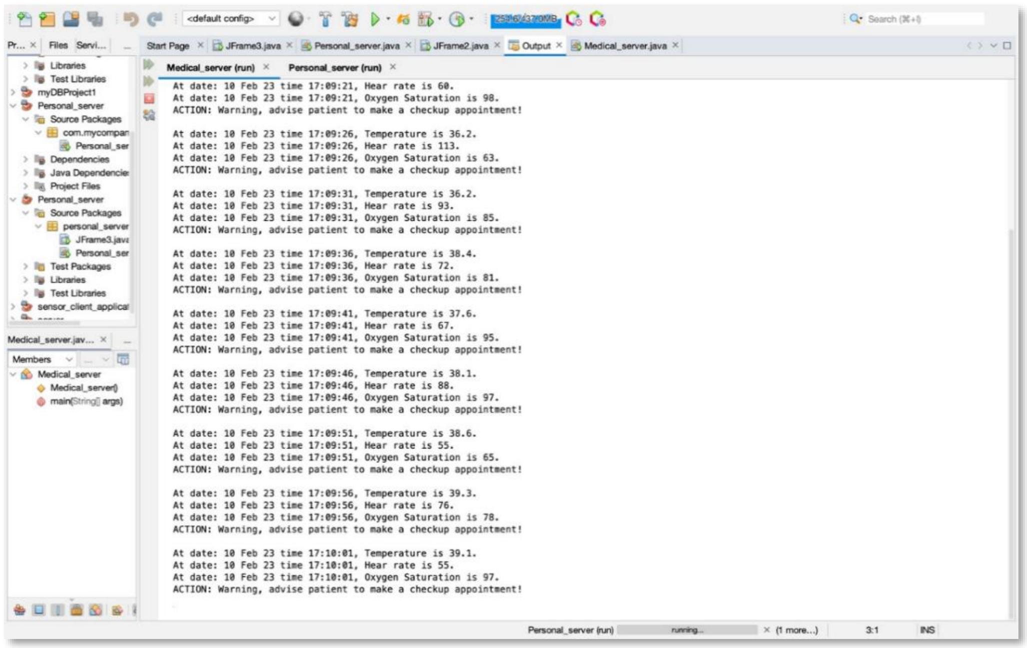
Figure 4:run in Medical Server console.

4.2 run of the program on different machines.

Using the code below

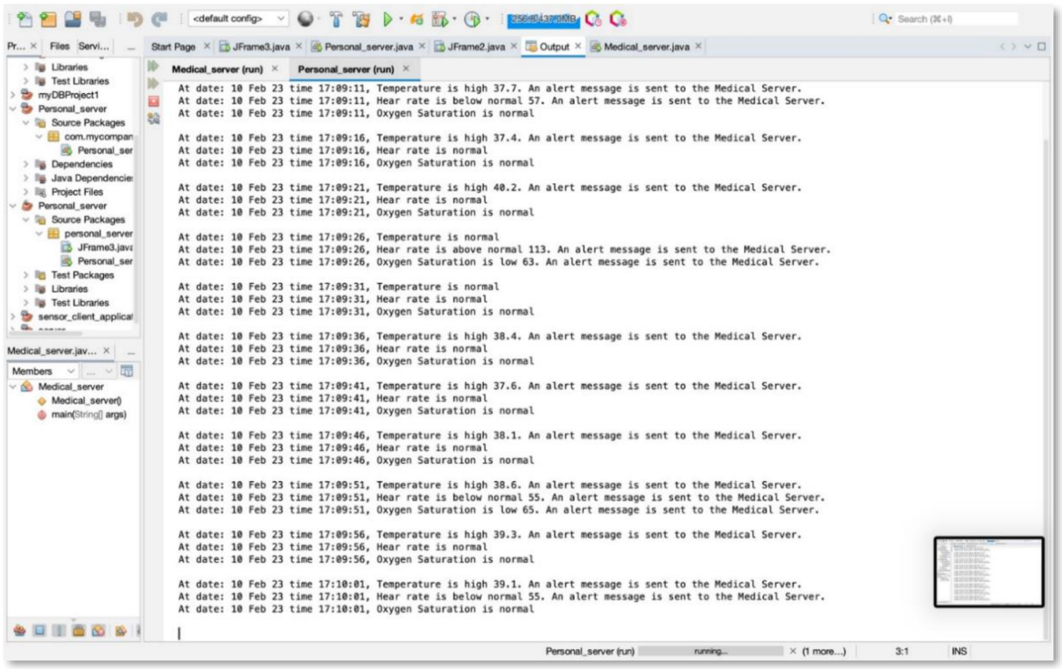
```
// to get the address from another device
InetAddress addresses = InetAddress.getByName("x.x.x.x");
// save the address in a String variable
String hostName = addresses.getHostName();
// create the socket based on the specific port and address
Socket socket = new Socket(hostName, port);
```

We could run the code on two different machines based on its IP address.



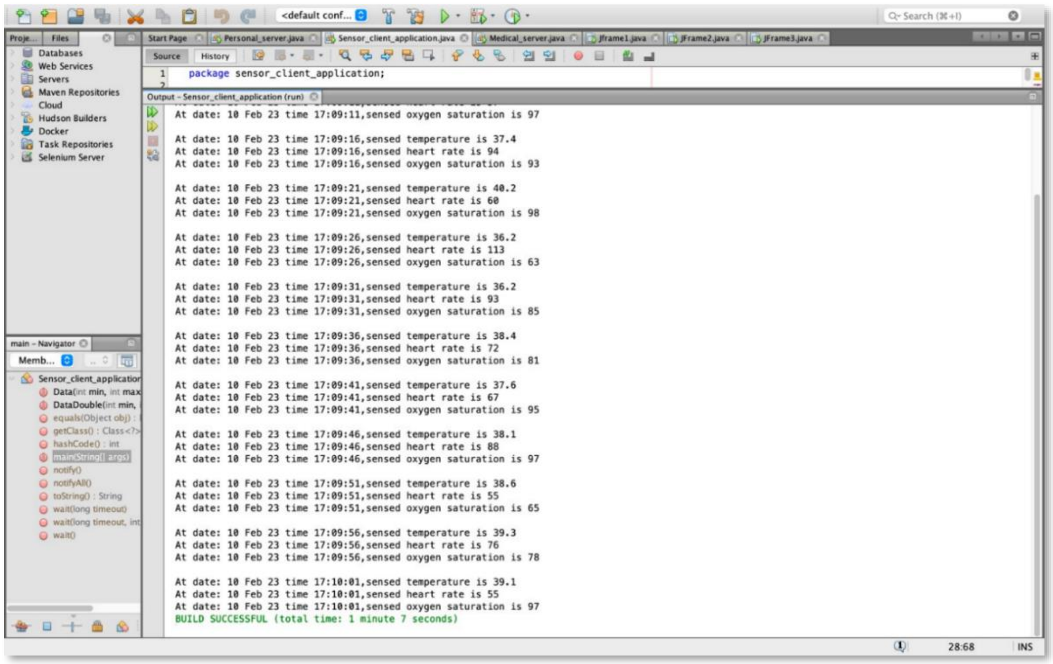
```
At date: 10 Feb 23 time 17:09:21, Hear rate is 60.
At date: 10 Feb 23 time 17:09:21, Oxygen Saturation is 98.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 Feb 23 time 17:09:26, Temperature is 36.2.
At date: 10 Feb 23 time 17:09:26, Hear rate is 113.
At date: 10 Feb 23 time 17:09:26, Oxygen Saturation is 63.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 Feb 23 time 17:09:31, Temperature is 36.2.
At date: 10 Feb 23 time 17:09:31, Hear rate is 93.
At date: 10 Feb 23 time 17:09:31, Oxygen Saturation is 85.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 Feb 23 time 17:09:36, Temperature is 38.4.
At date: 10 Feb 23 time 17:09:36, Hear rate is 72.
At date: 10 Feb 23 time 17:09:36, Oxygen Saturation is 81.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 Feb 23 time 17:09:41, Temperature is 37.6.
At date: 10 Feb 23 time 17:09:41, Hear rate is 67.
At date: 10 Feb 23 time 17:09:41, Oxygen Saturation is 95.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 Feb 23 time 17:09:46, Temperature is 38.1.
At date: 10 Feb 23 time 17:09:46, Hear rate is 88.
At date: 10 Feb 23 time 17:09:46, Oxygen Saturation is 97.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 Feb 23 time 17:09:51, Temperature is 38.6.
At date: 10 Feb 23 time 17:09:51, Hear rate is 55.
At date: 10 Feb 23 time 17:09:51, Oxygen Saturation is 65.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 Feb 23 time 17:09:56, Temperature is 39.3.
At date: 10 Feb 23 time 17:09:56, Hear rate is 76.
At date: 10 Feb 23 time 17:09:56, Oxygen Saturation is 78.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 Feb 23 time 17:10:01, Temperature is 39.1.
At date: 10 Feb 23 time 17:10:01, Hear rate is 55.
At date: 10 Feb 23 time 17:10:01, Oxygen Saturation is 97.
ACTION: Warning, advise patient to make a checkup appointment!
```

Figure 5:run in Medical Server console on the first device.



```
At date: 10 Feb 23 time 17:09:11, Temperature is high 37.7. An alert message is sent to the Medical Server.
At date: 10 Feb 23 time 17:09:11, Hear rate is below normal 57. An alert message is sent to the Medical Server.
At date: 10 Feb 23 time 17:09:11, Oxygen Saturation is normal

At date: 10 Feb 23 time 17:09:16, Temperature is high 37.4. An alert message is sent to the Medical Server.
At date: 10 Feb 23 time 17:09:16, Hear rate is normal
At date: 10 Feb 23 time 17:09:16, Oxygen Saturation is normal

At date: 10 Feb 23 time 17:09:21, Temperature is high 40.2. An alert message is sent to the Medical Server.
At date: 10 Feb 23 time 17:09:21, Hear rate is normal
At date: 10 Feb 23 time 17:09:21, Oxygen Saturation is normal

At date: 10 Feb 23 time 17:09:26, Temperature is normal
At date: 10 Feb 23 time 17:09:26, Hear rate is above normal 113. An alert message is sent to the Medical Server.
At date: 10 Feb 23 time 17:09:26, Oxygen Saturation is low 63. An alert message is sent to the Medical Server.

At date: 10 Feb 23 time 17:09:31, Temperature is normal
At date: 10 Feb 23 time 17:09:31, Hear rate is normal
At date: 10 Feb 23 time 17:09:31, Oxygen Saturation is normal

At date: 10 Feb 23 time 17:09:36, Temperature is high 38.4. An alert message is sent to the Medical Server.
At date: 10 Feb 23 time 17:09:36, Hear rate is normal
At date: 10 Feb 23 time 17:09:36, Oxygen Saturation is normal

At date: 10 Feb 23 time 17:09:41, Temperature is high 37.6. An alert message is sent to the Medical Server.
At date: 10 Feb 23 time 17:09:41, Hear rate is normal
At date: 10 Feb 23 time 17:09:41, Oxygen Saturation is normal

At date: 10 Feb 23 time 17:09:46, Temperature is high 38.1. An alert message is sent to the Medical Server.
At date: 10 Feb 23 time 17:09:46, Hear rate is normal
At date: 10 Feb 23 time 17:09:46, Oxygen Saturation is normal

At date: 10 Feb 23 time 17:09:51, Temperature is high 38.6. An alert message is sent to the Medical Server.
At date: 10 Feb 23 time 17:09:51, Hear rate is below normal 55. An alert message is sent to the Medical Server.
At date: 10 Feb 23 time 17:09:51, Oxygen Saturation is low 65. An alert message is sent to the Medical Server.

At date: 10 Feb 23 time 17:09:56, Temperature is high 39.3. An alert message is sent to the Medical Server.
At date: 10 Feb 23 time 17:09:56, Hear rate is normal
At date: 10 Feb 23 time 17:09:56, Oxygen Saturation is normal

At date: 10 Feb 23 time 17:10:01, Temperature is high 39.1. An alert message is sent to the Medical Server.
At date: 10 Feb 23 time 17:10:01, Hear rate is below normal 55. An alert message is sent to the Medical Server.
At date: 10 Feb 23 time 17:10:01, Oxygen Saturation is normal
```

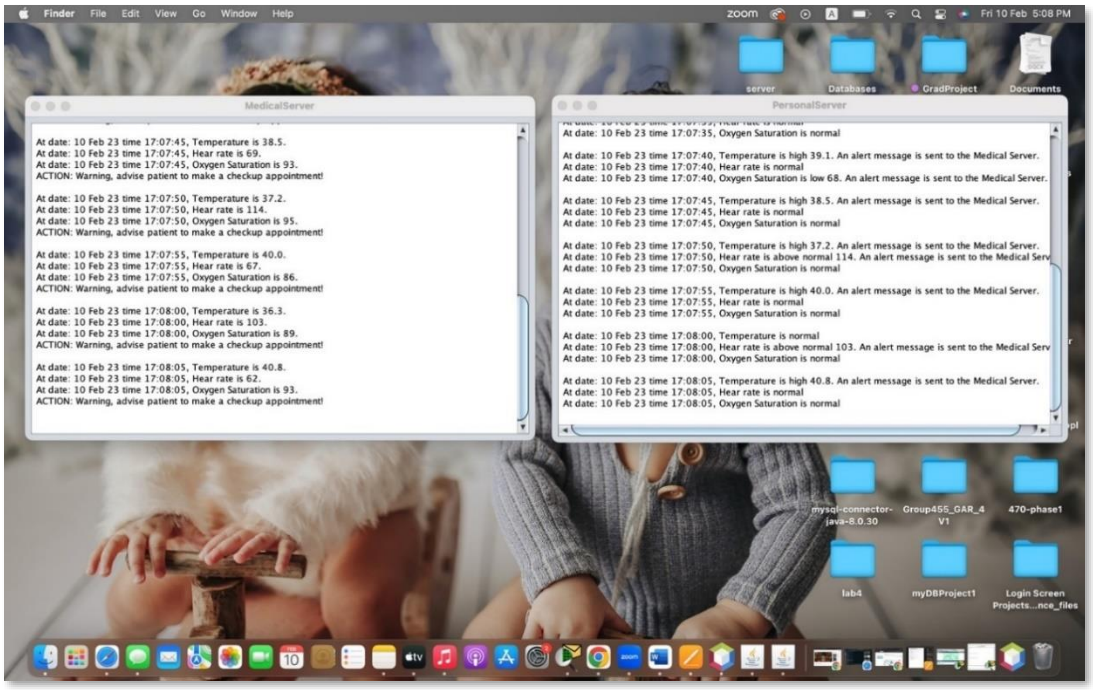
Figure 6:run in Personal Server console on the first device.



```
package sensor_client_application;
```

```
At date: 10 Feb 23 time 17:09:11,sensed oxygen saturation is 97

At date: 10 Feb 23 time 17:09:16,sensed temperature is 37.4
At date: 10 Feb 23 time 17:09:16,sensed heart rate is 94
At date: 10 Feb 23 time 17:09:16,sensed oxygen saturation is 93

At date: 10 Feb 23 time 17:09:21,sensed temperature is 40.2
At date: 10 Feb 23 time 17:09:21,sensed heart rate is 60
At date: 10 Feb 23 time 17:09:21,sensed oxygen saturation is 98

At date: 10 Feb 23 time 17:09:26,sensed temperature is 36.2
At date: 10 Feb 23 time 17:09:26,sensed heart rate is 113
At date: 10 Feb 23 time 17:09:26,sensed oxygen saturation is 63

At date: 10 Feb 23 time 17:09:31,sensed temperature is 36.2
At date: 10 Feb 23 time 17:09:31,sensed heart rate is 93
At date: 10 Feb 23 time 17:09:31,sensed oxygen saturation is 85

At date: 10 Feb 23 time 17:09:36,sensed temperature is 38.4
At date: 10 Feb 23 time 17:09:36,sensed heart rate is 72
At date: 10 Feb 23 time 17:09:36,sensed oxygen saturation is 81

At date: 10 Feb 23 time 17:09:41,sensed temperature is 37.6
At date: 10 Feb 23 time 17:09:41,sensed heart rate is 67
At date: 10 Feb 23 time 17:09:41,sensed oxygen saturation is 95

At date: 10 Feb 23 time 17:09:46,sensed temperature is 38.1
At date: 10 Feb 23 time 17:09:46,sensed heart rate is 88
At date: 10 Feb 23 time 17:09:46,sensed oxygen saturation is 97

At date: 10 Feb 23 time 17:09:51,sensed temperature is 38.6
At date: 10 Feb 23 time 17:09:51,sensed heart rate is 55
At date: 10 Feb 23 time 17:09:51,sensed oxygen saturation is 65

At date: 10 Feb 23 time 17:09:56,sensed temperature is 39.3
At date: 10 Feb 23 time 17:09:56,sensed heart rate is 76
At date: 10 Feb 23 time 17:09:56,sensed oxygen saturation is 78

At date: 10 Feb 23 time 17:10:01,sensed temperature is 39.1
At date: 10 Feb 23 time 17:10:01,sensed heart rate is 55
At date: 10 Feb 23 time 17:10:01,sensed oxygen saturation is 97
BUILD SUCCESSFUL (total time: 1 minute 7 seconds)
```

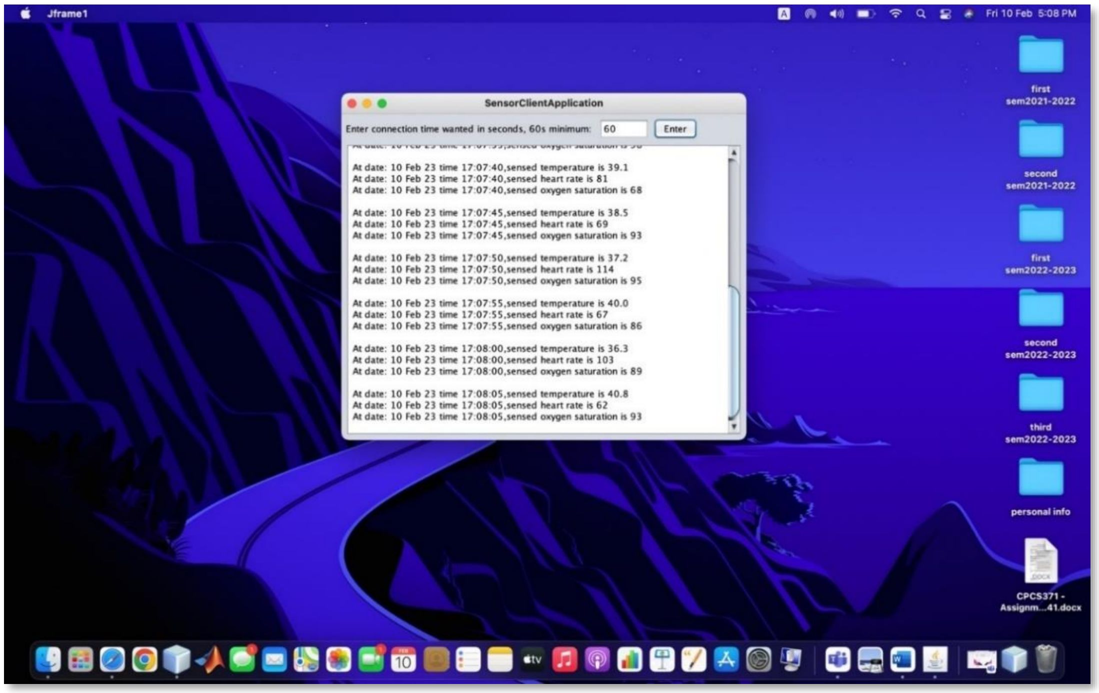
Figure 7:run in Sensor console on the second device.



```
At date: 10 Feb 23 time 17:07:45, Temperature is 38.5.
At date: 10 Feb 23 time 17:07:45, Hear rate is 69.
At date: 10 Feb 23 time 17:07:45, Oxygen Saturation is 93.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 Feb 23 time 17:07:50, Temperature is 37.2.
At date: 10 Feb 23 time 17:07:50, Hear rate is 114.
At date: 10 Feb 23 time 17:07:50, Oxygen Saturation is 95.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 Feb 23 time 17:07:55, Temperature is 40.0.
At date: 10 Feb 23 time 17:07:55, Hear rate is 67.
At date: 10 Feb 23 time 17:07:55, Oxygen Saturation is 86.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 Feb 23 time 17:08:00, Temperature is 36.3.
At date: 10 Feb 23 time 17:08:00, Hear rate is 103.
At date: 10 Feb 23 time 17:08:00, Oxygen Saturation is 89.
ACTION: Warning, advise patient to make a checkup appointment!

At date: 10 Feb 23 time 17:08:05, Temperature is 40.8.
At date: 10 Feb 23 time 17:08:05, Hear rate is 62.
At date: 10 Feb 23 time 17:08:05, Oxygen Saturation is 93.
ACTION: Warning, advise patient to make a checkup appointment!
```

```
At date: 10 Feb 23 time 17:07:35, Oxygen Saturation is normal

At date: 10 Feb 23 time 17:07:40, Temperature is high 39.1. An alert message is sent to the Medical Server.
At date: 10 Feb 23 time 17:07:40, Hear rate is normal
At date: 10 Feb 23 time 17:07:40, Oxygen Saturation is low 68. An alert message is sent to the Medical Server.

At date: 10 Feb 23 time 17:07:45, Temperature is high 38.5. An alert message is sent to the Medical Server.
At date: 10 Feb 23 time 17:07:45, Hear rate is normal
At date: 10 Feb 23 time 17:07:45, Oxygen Saturation is normal

At date: 10 Feb 23 time 17:07:50, Temperature is high 37.2. An alert message is sent to the Medical Server.
At date: 10 Feb 23 time 17:07:50, Hear rate is above normal 114. An alert message is sent to the Medical Serv
At date: 10 Feb 23 time 17:07:50, Oxygen Saturation is normal

At date: 10 Feb 23 time 17:07:55, Temperature is high 40.0. An alert message is sent to the Medical Server.
At date: 10 Feb 23 time 17:07:55, Hear rate is normal
At date: 10 Feb 23 time 17:07:55, Oxygen Saturation is normal

At date: 10 Feb 23 time 17:08:00, Temperature is normal
At date: 10 Feb 23 time 17:08:00, Hear rate is above normal 103. An alert message is sent to the Medical Serv
At date: 10 Feb 23 time 17:08:00, Oxygen Saturation is normal

At date: 10 Feb 23 time 17:08:05, Temperature is high 40.8. An alert message is sent to the Medical Server.
At date: 10 Feb 23 time 17:08:05, Hear rate is normal
At date: 10 Feb 23 time 17:08:05, Oxygen Saturation is normal
```

Figure 8:run in Personal and Medical Servers GUI on the first device.



Enter connection time wanted in seconds, 60s minimum: 60

```
At date: 10 Feb 23 time 17:07:40,sensed temperature is 39.1
At date: 10 Feb 23 time 17:07:40,sensed heart rate is 81
At date: 10 Feb 23 time 17:07:40,sensed oxygen saturation is 68

At date: 10 Feb 23 time 17:07:45,sensed temperature is 38.5
At date: 10 Feb 23 time 17:07:45,sensed heart rate is 69
At date: 10 Feb 23 time 17:07:45,sensed oxygen saturation is 93

At date: 10 Feb 23 time 17:07:50,sensed temperature is 37.2
At date: 10 Feb 23 time 17:07:50,sensed heart rate is 114
At date: 10 Feb 23 time 17:07:50,sensed oxygen saturation is 95

At date: 10 Feb 23 time 17:07:55,sensed temperature is 40.0
At date: 10 Feb 23 time 17:07:55,sensed heart rate is 67
At date: 10 Feb 23 time 17:07:55,sensed oxygen saturation is 86

At date: 10 Feb 23 time 17:08:00,sensed temperature is 36.3
At date: 10 Feb 23 time 17:08:00,sensed heart rate is 103
At date: 10 Feb 23 time 17:08:00,sensed oxygen saturation is 89

At date: 10 Feb 23 time 17:08:05,sensed temperature is 40.8
At date: 10 Feb 23 time 17:08:05,sensed heart rate is 62
At date: 10 Feb 23 time 17:08:05,sensed oxygen saturation is 93
```

Figure 9:run in Sensor GUI on the second device.

5 Teamwork and Lessons learned.

We divided our group of 6 girls into three pairs using an online spinner to decide which pair works on which class. Shaden and Logain worked on the sensor client application, Amani and Jetana on the personal server, Hasna and Wajd worked on the medical server. Amani debugged the final application code and fixed all minor details that were causing some issues with the connection of projects together. As for the report, the introduction and teamwork and lesson learned were done by Logain and Jetana, the patient monitoring application interaction diagram was done by Amani, the RHMS implementation was done by Wajd, the RHMS application run snapshots were done by Shaden, the word file template and the conclusion were done by Hasna.

As for the lessons learned, this project was new to all group members, and it explained the client-server architecture and taught us how it's implemented. We also learned how to manage and divide a big group to get the most out of everyone's time and effort.

6 Conclusion

To sum up, this project is based on the client-server architecture and how to implement an application to inspect elderly patients' health using TCP sockets. Additionally, the application consists of one client and two servers. So, the messages will flow throughout the 3 TCP sockets. One of the main ideas in this application is that the program could be run on up to three end systems. Therefore, the whole report was done by 6 group members cooperatively.

7 References

James F. Kurose, Keith W. Ross,, "Computer Networking", Pearson Education; 6 edition (2012)

Java Socket Programming (Java Networking Tutorial) - javatpoint. (n.d.). Retrieved January 22, 2023 from www.javatpoint.com. https://www.javatpoint.com/socket-programming

P. (2022, November 22). *Thread.sleep() in Java - Java Thread sleep*. Retrieved January 25, 2023 from DigitalOcean. https://www.digitalocean.com/community/tutorials/thread-sleep-java

NetBeans, A. (n.d.). *Designing a Swing GUI in NetBeans IDE*. https://netbeans.apache.org/kb/docs/java/quickstart-gui.html