FINAL YEAR DISSERTATION
2004

HALLUR GUNNARSSON
Samningur milli nemenda Háskólans á Akureyri og bókasafns háskólans um meðferð lokaverkefna

Ég undirrituð/aðurður...nemandi við Háskólanum á Akureyri áhendi her með bókasafi háskólanum þrú eintök af lokaverkefni minu, eitt prentað og innbundið, annað prentað og öinnbundið og það þrója á geisladiski.

Prentuð eintök:

 Lokaverkefnið er lokað □ til □

<table>
<thead>
<tr>
<th>Ef lokaverkefnið er opið er bókasafninu heimilt að:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>já</strong></td>
</tr>
<tr>
<td>□</td>
</tr>
<tr>
<td>□</td>
</tr>
<tr>
<td>□</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ef lokaverkefnið er opið er heimilt:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>já</strong></td>
</tr>
<tr>
<td>□</td>
</tr>
<tr>
<td>□</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ef lokaverkefnið er opið er heimilt:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>já</strong></td>
</tr>
<tr>
<td>□</td>
</tr>
<tr>
<td>□</td>
</tr>
</tbody>
</table>

| □ | **nei** | Bókasafninu er heimilt að ljósrita lokaverkefnið til viðhalds á snjáðum eintökum sinum, þó aldrei svo að það eigi fleiri en tvö eintök í senn |


Stafrént eintak:

Lokaverkefnið er lokað ☐ til ____________________

Þó að lokaverkefnið sé lokað er bókasafninu heimilt að leyfa aðgang á vefnum að:

☐ ☐ efnisyfirlití
☐ ☐ útdrátti
☒ ☐ heimildaskrám

Lokaverkefnið er opið og bókasafninu heimilt að:

☐ ☐ bjóða opinu aðgang að því á vefnum í heild sinni til allra
☐ ☐ bjóða adeins aðgang að því af staðarnetí háskólan
d
☐ ☐ leyfa fjarnemum og starfsmönnum háskólan aðgang utan staðarnets háskólan með aðgangs- og lykilordum

Akureyri 16.6.20

[Signature]

nemandi

bókavörður
KUNST
the art of awareness

Hallur Gunnarsson

Module Supervisor: Dr. Mark O’Brien
Project Supervisor: Adam Bridgen

Submitted April 2004, in partial fulfilment of the conditions of the award of degree BSc.

I hereby declare that this interim report is all my own work, except as indicated in the text:

Hallur Gunnarsson
16.04.2004
Abstract

This dissertation describes my final year project in the University of Akureyri, Faculty of Information Technology. This project is about taking records of heart rate from individuals, store them in a database retrieve them again and display them on the internet. The information is to be viewed in a graph representation.
Table of contents

Preface.................................................................................................................. 1
Motivation for the work..................................................................................... 2
Description of the work.................................................................................. 3
Related work ..................................................................................................... 4
  MobiHealth .................................................................................................... 4
Design ............................................................................................................. 5
  Database ....................................................................................................... 5
  Server ........................................................................................................... 5
  Java .............................................................................................................. 6
  XML .............................................................................................................. 6
Interface ......................................................................................................... 7
Security ............................................................................................................ 8
  Pattern Matching .......................................................................................... 8
Evaluation ....................................................................................................... 10
  Testing ........................................................................................................ 10
Conclusion ...................................................................................................... 11
Reference ......................................................................................................... 12
Table of figures

Figure 1. XML + XSLT = SVG.................................................................7
Figure 2. Website for heart-rate graphs ...........................................8
Preface

This dissertation is intended to describe various parts of this final year project. It will start in the motivation for the work, after that it will go direct into the description of the work. Having gone through those issues the related work will be discussed. The system design is explained and illustrated. The implementation of the software is also discussed as the reasons for the decisions made. This dissertation ends with a section about the evaluation of the software.
Motivation for the work

In this section the motivation for the work will be addressed. It all started with a crazy idea. By letting the idea cultivating for some time and discussing it with people with different background it improved and became more realistic. But the general motivation and the reason why I choose this instead of something else, was that I believe that this could make the world a better place.

The first idea for this system was to give people a change to monitor and have an influence on my live. This was to by possible by following my body status constantly online. People would then be able to contact my and ask me to do some tasks like running, and then they would observe the body status changing. I could also make a schedule about some activities as attending school, go in the gym, and people could monitor how well I did in following the schedule.

When I addressed this idea my supervisor pointed out that this could come in more use having more people connected to the system. Designing a system that would take these data and then after some time of data input the system could figure out what the individual is doing. At first the person would type in current activity as well as the sensors would record the body status. After some time the user would not have to type in the activity, the system would recognise the values of the data and make a assumption about what the person is doing.

This could even be expanded further. Having the people connected categorised in groups by some rules like place of work. A comparison could be made between groups, looking at patterns in peoples live, not only individually but also in a group.

There are many potential ways that this system could help people. Doctors could for an example have continuous monitor on their patients or the system could issue an alert if the state of the patient was about to become. The information this system would collect would be extremely sensitive and therefore security is an important factor.

It became clear early on that the full implementation of this idea would reach far beyond the possible scope of a project of this size. The idea can be implemented on the work produced at this point. The idea can therefore be thought of as a potential future work, if this project will be taken further.
Description of the work

In this section, I will state what this system is meant to achieve, how it is meant to function to reach the previous stated goal. This section will give an overall description of the work.

Definition of scope

The overall goal can be described as the following. This system is meant to give the user the ability to monitor person's heart rate. The user is supposed to be able to view the information in a graph presentation. The data is saved and historic data can be compared by using the graph to inspect trends and patterns.

Simple system to monitor and view peoples status online

The data is stored in a database. That data is then retrieved from the database and transformed to a format that is used to represent the data in a graph on an internet site.

The information about the heart rate, time of the record and some individual information about the person are gathered and stored in a IBM DB2 Universal database. These data are then retrieved from the database using java servlet by using the Apache Tomcat 5.0 web server. Once retrieved from the database the data is presented in XML format containing the in-lined SVG formatted information. The SVG information can then be presented on a website in a graph using a SVG viewer plug-in.
Related work
In this section I will address related work, what has been done before and how my work differs from that. Some say that nothing is new under the sun, but I assure those that in the field of computer science this old saying does not apply.

Searching on the internet I merely found system that where focusing on the Hardware and the communication part. That is the sensors for recording the persons body status and how the communication would take place sending the information from the person to the system computer

MobiHealth
Though this system has different goal it does use some aspect of the technique of sending information from the user located in different areas to the main system. This system will be briefly described and points that can be applied to the KUNST will be emphasised.

Though the project MobiHealth is in the domain of healthcare it has many shared functionality as the KUNST system. MobiHealth is a project of 14 partners from five European countries. It aims at developing and trialling new mobile services in the area of healthcare, thus bringing healthcare to the patient. The MobiHealth system allows patients to be fully mobile whilst undergoing health monitoring.

The patient wears a lightweight monitoring system – the MobiHealth Body Area Network (BAN) – which is customized to their individual health needs. Physical measurements such as blood pressure or electrocardiogram (ECG) are measured by the MobiHealth BAN and transmittedwirelessly from the BAN to their doctor, the hospital or their health call centre. (mobilHelath.org)

In this project a technology is used for transmitting data over GSM network General Packet Radio Service (GPRS). This technology provides high end-to-end IP packet service over the GSM network. GPRS even also allows SMS transfer over GPRS radio channels. (University of Twente, et.al.)

There is not much stated about the sensors. Their documentation is more focusing on networking. For an example how they will transmit the data from the patient (user) to the doctor (end terminal). There work is similar concerning hardware issues and transmitting the data. MobiHealth is about doctors being able to monitor their patents constantly. The KUNST system is designed with more wide-ranging idea in mind, it does not only limit it self in the health sector.

My feeling for the work done so far is that my system takes the idea further. Though I have not completed the vision and some might say that I have not produced anything new. As I have stated earlier that these are only the first steps of the future KUNST system.
Design
This section covers an overview of the design. It will give a description of the design that I chose and why I designed the system in that way. The problem of the design will be stated and how they where overcome.

On early stages of the project I decided to use IBM DB2 Universal database and Apache Tomcat 5.0 server. The reason for this chose is simple the server is an open source freeware and I got the database which was an educational copy from the supervisor.

Database
As stated earlier IBM DB2 Universal database was used in this project. It took some time to get acquainted with the program, but managed to do some basic procedures. Like creating tables, importing data, setting key constraints, viewing data in the tables and things like that.

The tables that I created where:
- person (ssn, firstname, lastname, birthday, address, city, postcode, country, phone, email)
- pulse (ssn, ordinal, rate)
- frequency (ssn, starttime, interval)

The ssn is the social security number and it is a primary key in person and a foreign key in pulse and frequency. Ordinal is the number of the record, it starts in 1 and then counts as the records are taken. The interval is the time period between each heart rate record but it is always the same within each file record. For an example today the interval is 10 seconds but tomorrow it could be 5 second.

By taking the interval and multiply it with the ordinal and add the sum of that to the starttime, then an individual record time can be found in the file. ((ordinal * interval) + starttime) = actual time of a specific record in the pulse table.

I imported data in the tables using the import feature provided in the IBM DB2. To be able to import the heart rate data I had to make small changes to the file. I had to take out the header and all the other information apart from the actual heart rate data. I also had to adjust the Column delimiter as the file was not in the default mode regarding to delimiters.

Server
The server that I used was Apache Tomcat 5.0. “Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies.” (The Apache Software Foundation). Tomcat is running on my own computer and I managed to run all the example servlets and the JSP’s that are in the package. I also produced a simple arithmetic servlet, which I used to gain
knowledge in this field. Then I tried to access the IBM DB2 Universal database using a servlet but I did not manage to complete that task.

**Java**

Before I produced the Java Servlets, I compiled it and tried it out in Java. Doing that I managed to connect to the IBM DB2 Universal database using jdbc odbc bridge. I used basic SQL statements to retrieve, import, deleted and manipulate data in the tables I had produced. The next step would be to retrieve the data in a XML format, but that step is not completed. I also wanted to transform the data from the XML format in SVG format which is also a valid XML, this step was not completed.

**XML**

The information, the hear rate measurements need to be displayed on the internet. The easiest way for the user to look at data like this is in a graph. EXtensible Markup Language (XML) was designed to describe data and using another format of XML which is Scalable Vector Graphics (SVG) those data can be described in a graph format. XML is good to us as a data file because the designer can set his own tags which can be used for a good categorising of the data.

Basic XML file looks like this:

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<QuarterlySales>
  <Year>2002</Year>
  <MaxValue>900</MaxValue>
  <Quarter name="Q1">450</Quarter>
  <Quarter name="Q2">600</Quarter>
  <Quarter name="Q3">575</Quarter>
  <Quarter name="Q4">800</Quarter>
</QuarterlySales>
(Moharil, Avinash & Zad, Rajesh)
Where this information is about quarter sales figures in the year 2002.

The XML file that I designed look like this with some example data:

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<record>
  <ssn>3008764989</ssn>
  <date>01.04.2004</date>
  <ordinal="1">125</ordinal>
  <ordinal="2">128</ordinal>
  <ordinal="3">136</ordinal>
  <ordinal="4">144</ordinal>
  <ordinal="5">154</ordinal>
</record>
This is only a small example file but the overall look can be seen. Where the ssn for the person who has this heart rate. The date of the record and then the ordinal and its heart rate value.

As stated earlier SVG is also a valid XML format. It can be described as another look in the data. I manually produced a SVG graph file which is presented on the project website and it can be seen in appendix I.

First I tried to transform a sample XML file to SVG using eXtensible Stylesheet Language Transform (XSLT) and worked on it for a quite long time without success, though I am at this point well informed about these file standards. I had one particular article as a guideline and I tried to manipulate the example given in that article. The reason for the time spent on that particular article was that is matched very well to what my system was meant to achieve. The overall design from the example in the article can be viewed at this figure.

![Diagram of application and client tier](image)

Figure 1. XML + XSLT = SVG

Looking at this figure you can see that XML and XSLT combined should produce SVG. I tried this without a success until I took it to the supervisor and he used an external transformer then it worked. There was no mentioned about somekind of transformer in the article, therfore I did not search for one. I did not like the idea of using an external transformer as this should be done on the fly and the user should not have to have any special software apart from the SVG viewer to look at the graphs.

(Moharil, Avinash & Zad, Rajesh)

**Interface**

I created a website using one of my favourite editors NetObjectFusion. I could have done this as well in basic HTML, but I did not because I work much faster and it is more efficient using this program. The website is both an interface and the project description. At this moment the website is static and it is hosted on the University server (my local space). I have registered for a domain name where it can be looked at and the URL is http://here.is/kunft. The project part shows what this project is about as well as all the work that has been handed in.
I look at the interface part as the main part for this project. When the user enters the website, he can see the heart rate of the individual connected to the system. The information can be viewed in a graph for a different time periods, the current day, last day, last week and the last month. The information is still just an example at this stage, since I did not manage to finish the code. They are only at the website for a demonstration purpose, for the user so that he can get a glimpse of how the website will work.

![heart-rate graph]

**Figure 2. Website for heart-rate graphs**

On this figure the heart-rate graphs can be seen as it is displayed on the website.

**Security**

I realised it when this idea came to me that this could not be true unless some security aspects would be discussed and addressed. Having data that are of this delicate, the system has to be secured. Issues like who should have access to the information and what kind of security should be installed need to be taken.

At this stage I decided not to spend much time in security issues. I knew from the beginning that this version would only be a demo version and that security aspects could be discussed at later steps in the implementation.

**Pattern Matching**

If this project would have worked out as it was planned when I wrote the interim report. I would have spent the last month that is March trying to do some pattern matching on the recorded data. Having that in mind I registered for the Pattern Recognition course at the University. I did that so that I would be able to require
some basic knowledge in that field. Though I did not reach that stage in the project I do not regret the time spent on this field, it will without a doubt become handy some day. Pattern Matching could be done on the data using for an example the ID3 Decision Tree Induction Algorithm. I did an assignment in this course where I used example data which were closely related to this project. This assignment can be viewed as small introduction for the future pattern matching part for this project.
Evaluation

In this section I will discuss testing, what testing was conducted and that should be done when the system is finished.

Testing

Since I did not manage to finish the code for the system, I did not perform tests in the Java Servlet retrieving the data and in the transformation to the SVG format.

The XML format is very strict and since SVG is valid XML, SVG is also strict. I did produce SVG file manually and it is viewed on the website. I used the Adobe SVG viewer and it is stated on the website that it is the appropriate viewer to use. I did therefore not test that file to any other viewer.

Importing data using the embedded feature of IBM DB2 from an external file. If the file was corrupted in any way then the import was not successful. Corrupted means that the values where not valid, that is characters instead of numbers and errors like that.

Manipulate data using the Java program was done using basic SQL. I did not produce any serious tests on the data. I tried to retrieve some data that did not exist, import invalid data. As suspected I failed and got error message indicating that.

Though I did not perform any serious testing, I did consider testing issues and meant to apply them when the system would be completed. Tests like Cyclomatic Complexity and stress tests would be conducted, when the system would be ready for those kinds of tests.
Conclusion

The work in this project can be described in the following way. I came across many obstacles some I managed to break others I didn’t. I learned a lot from all of them. Working on this project was hard and disappointing at numerous moments, but when something worked it is so enjoyable that it pays off all the time spent on it.
Reference


Final Year Dissertation
2004

Appendix 1

Hallur Gunnarsson
# Table of Contents

Simple Java Servlet .................................................................................................................. 2  
Java DBConnection .................................................................................................................. 3  
Java transformer still in process ............................................................................................ 6  
DBConnection Java Servlet still in process ............................................................................ 7  
Example of files ...................................................................................................................... 9  
  Heart rate file ....................................................................................................................... 9  
  SVG, manually produced ....................................................................................................... 11  
Reference .................................................................................................................................. 14
Simple Java Servlet

```java
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class SqrtServlet extends HttpServlet {

    public void doGet(HttpServletRequest request,
            HttpServletResponse response) throws IOException, ServletException{
        response.setContentType("text/html");

        PrintWriter out = response.getWriter();
        out.println("<html>
        <head><title>Square roots</title></head>
        <body>

        String data = request.getParameter("number");
        double value = 0;

        try {
            value = Double.parseDouble(data);
        } catch (Exception e) {
            out.println("<p>Need NUMERIC data.");
            out.println("</body></html>");
        }

        if(value<0.0) {
            out.println("You are living in an imaginary world.");
            out.println("</body></html>");
            return;
        }

        value = Math.sqrt(value);
        out.println("Your square root value is "+value);
        out.println("</body></html>");
    }

    (Gray, Neil.)
```
Java DBCConnection

```java
import java.sql.*;
import java.io.*;
import java.lang.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;

class db2conn {

    private static final String dbDriverName = "sun.jdbc.odbc.JdbcOdbcDriver";
    private static final String dbURL = "jdbc:odbc:KUNST";
    private Connection theDBConnection;

    public void init() {
        String userName = "db2admin";
        String userPassword = "tiippi";
        try {
            Class.forName(dbDriverName);
            theDBConnection = DriverManager.getConnection(dbURL, userName, userPassword);
        }
        catch (Exception e) {
            System.out.println("hello the catch is on");
            System.out.println(e.toString());
            e.printStackTrace();
            System.err.println("Got an exception! ");
            System.err.println(e.getMessage());
        }
    }

    private void doInsert() {
        try {
            Statement st = theDBConnection.createStatement();
            st.executeUpdate("INSERT INTO pulse VALUES (3008764989, 1028, 22)");
        }
        catch (Exception e) {
            System.err.println("Got an exception! doInsert ");
            System.err.println(e.getMessage());
        }
    }
}
```
private void update() {
    try {
        Statement st = theDBConnection.createStatement();
        st.executeUpdate("UPDATE pulse SET ssn=2008764989 WHERE ssn=1008764989");
    } catch (Exception e) {
        System.err.println("Got an exception! update ");
        System.err.println(e.getMessage());
    }
}

private void select() {
    try {
        Statement st = theDBConnection.createStatement();
        System.out.println("Starting selecting");
        ResultSet rs = st.executeQuery("SELECT ssn, ordinal, rate FROM pulse WHERE ssn=3008764989");
        while(rs.next()) {
            System.out.println("ssn" + rs.getString("ssn") + " " + \\
            rs.getString("rate") + " " + rs.getString("ordinal"));
        }
    } catch (Exception e) {
        System.err.println("Got an exception! The select doesn’t work ");
        System.err.println(e.getMessage());
    }
}

public void destroy() {
    if (theDBConnection != null) {
        try {
            theDBConnection.close();
        } catch (Exception e) {
        }
    }
}

public db2conn() {
    init();
    doInsert();
    update();
    select();
    destroy();
}
public static void main(String args[]) {
    System.out.println("Starting db2conn...");
    db2conn mainFrame = new db2conn();
}

(Gray, Neil.)
Java transformer still in process
import javax.xml.transform.*;
import javax.xml.transform.stream.*;

.
.
.

private void transform() {
    StreamSource xml = new StreamSource(sourceURL.getFile());
    StreamSource xsl = new StreamSource(new
    File(request.getRealPath("input.xsl")));

    StreamResult result = new StreamResult(response.getWriter());

    TransformerFactory tFactory = TransformerFactory.newInstance();
    Transformer transformer = tFactory.newTransformer(xsl);
    transformer.setOutputProperty(OutputKeys.METHOD, "svg");
    transformer.setOutputProperty(OutputKeys.ENCODING, "ISO-8859-1");
    transformer.setOutputProperty(OutputKeys.INDENT, "yes");
    transformer.transform(xml, result);
}
(Sourirajan, Sridhar)
import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class DBConnection extends HttpServlet {

private static final String dbDriverName = "sun.jdbc.odbc.JdbcOdbcDriver";
private static final String dbURL = "jdbc:odbc:KUNST";
private Connection theDBConnection;

public void init(HttpServletRequest req, HttpServletResponse res)
    throws ServletException, IOException {
    String userName = "db2admin";
    String userPassword = "rullah";
    res.setContentType("text/html");
    PrintWriter out = res.getWriter();

    try {
        Class.forName(dbDriverName);
        theDBConnection = DriverManager.getConnection(
            dbURL,
            userName, userPassword);
        Statement stmt = theDBConnection.createStatement();

        // Execute an SQL query, get a ResultSet
        ResultSet rs = stmt.executeQuery("SELECT ssn, rate FROM pulse");

        // Display the result set as a list
        out.println("<HTML><HEAD><TITLE>PULSE</TITLE></HEAD>");
        out.println("<BODY>");
        out.println("<UL>");
        while(rs.next()) {
            out.println("<LI> + rs.getString("ssn") + " + rs.getString("rate"));
        }
        out.println("</UL>");
        out.println("</BODY></HTML>");
    }
    catch(SQLException e) {
        out.println("Could't load database driver: " + e.getMessage());
    }
    catch(SQLException e) {
out.println("SQLException caught: "+e.getMessage());
}
finally {
// Always close the database connection.
try {
if (theDBConnection != null) theDBConnection.close();
}
catch (SQLException ignored) {}
Example of files

**Heart rate file**

[Params]
Version=105
Monitor=2
Mode=000
Date=20011118
StartTime=10:55:33.0
Length=0:25:18.0
Interval=5
Upper1=141
Lower1=80
Upper2=179
Lower2=141
Upper3=175
Lower3=145
Timer1=0:00:00.0
Timer2=0:00:00.0
Timer3=0:00:00.0
ActiveLimit=0
MaxHR=192
RestHR=70
StartDelay=0

[Note]
NV

[IntTimes]

[IntNotes]

[ExtraData]

[Summary-123]
50 0 5 45 0 0
192 141 80 70
1465 0 105 930 430 0
192 179 141 70
0 0 0 0 0 0
192 175 145 70
0 303
### Summary-TH

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3080</td>
<td>0</td>
<td>230</td>
<td>780</td>
<td>505</td>
<td>0</td>
</tr>
<tr>
<td>192</td>
<td>175</td>
<td>145</td>
<td>70</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>303</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SwapTimes

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0:00:54.5</td>
<td>1</td>
</tr>
</tbody>
</table>

### HRData

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>166</td>
<td>173</td>
<td>173</td>
<td>151</td>
<td>164</td>
</tr>
<tr>
<td>91</td>
<td>163</td>
<td>175</td>
<td>174</td>
<td>151</td>
<td>166</td>
</tr>
<tr>
<td>91</td>
<td>165</td>
<td>172</td>
<td>174</td>
<td>149</td>
<td>167</td>
</tr>
<tr>
<td>91</td>
<td>167</td>
<td>171</td>
<td>175</td>
<td>148</td>
<td>168</td>
</tr>
<tr>
<td>91</td>
<td>168</td>
<td>164</td>
<td>175</td>
<td>148</td>
<td>168</td>
</tr>
<tr>
<td>91</td>
<td>166</td>
<td>161</td>
<td>174</td>
<td>147</td>
<td>168</td>
</tr>
<tr>
<td>114</td>
<td>167</td>
<td>166</td>
<td>174</td>
<td>147</td>
<td>168</td>
</tr>
<tr>
<td>136</td>
<td>168</td>
<td>169</td>
<td>174</td>
<td>141</td>
<td>169</td>
</tr>
<tr>
<td>136</td>
<td>168</td>
<td>173</td>
<td>173</td>
<td>146</td>
<td>170</td>
</tr>
<tr>
<td>143</td>
<td>169</td>
<td>175</td>
<td>173</td>
<td>145</td>
<td>170</td>
</tr>
<tr>
<td>148</td>
<td>169</td>
<td>176</td>
<td>172</td>
<td>143</td>
<td>170</td>
</tr>
<tr>
<td>148</td>
<td>170</td>
<td>176</td>
<td>172</td>
<td>141</td>
<td>169</td>
</tr>
<tr>
<td>151</td>
<td>170</td>
<td>177</td>
<td>172</td>
<td>140</td>
<td>170</td>
</tr>
<tr>
<td>151</td>
<td>169</td>
<td>177</td>
<td>171</td>
<td>140</td>
<td>171</td>
</tr>
<tr>
<td>152</td>
<td>169</td>
<td>176</td>
<td>171</td>
<td>140</td>
<td>171</td>
</tr>
<tr>
<td>153</td>
<td>170</td>
<td>176</td>
<td>171</td>
<td>138</td>
<td>171</td>
</tr>
<tr>
<td>153</td>
<td>170</td>
<td>175</td>
<td>171</td>
<td>130</td>
<td>172</td>
</tr>
<tr>
<td>156</td>
<td>169</td>
<td>175</td>
<td>170</td>
<td>128</td>
<td>172</td>
</tr>
<tr>
<td>157</td>
<td>168</td>
<td>174</td>
<td>170</td>
<td>137</td>
<td>172</td>
</tr>
<tr>
<td>156</td>
<td>169</td>
<td>175</td>
<td>153</td>
<td>143</td>
<td>172</td>
</tr>
<tr>
<td>159</td>
<td>170</td>
<td>174</td>
<td>154</td>
<td>145</td>
<td>172</td>
</tr>
<tr>
<td>158</td>
<td>170</td>
<td>173</td>
<td>157</td>
<td>146</td>
<td>173</td>
</tr>
<tr>
<td>155</td>
<td>170</td>
<td>173</td>
<td>159</td>
<td>146</td>
<td>173</td>
</tr>
<tr>
<td>163</td>
<td>168</td>
<td>173</td>
<td>160</td>
<td>148</td>
<td>173</td>
</tr>
<tr>
<td>163</td>
<td>169</td>
<td>173</td>
<td>159</td>
<td>148</td>
<td>173</td>
</tr>
<tr>
<td>159</td>
<td>172</td>
<td>173</td>
<td>149</td>
<td>174</td>
<td>167</td>
</tr>
<tr>
<td>159</td>
<td>173</td>
<td>173</td>
<td>156</td>
<td>153</td>
<td>175</td>
</tr>
<tr>
<td>164</td>
<td>173</td>
<td>173</td>
<td>156</td>
<td>155</td>
<td>175</td>
</tr>
<tr>
<td>162</td>
<td>175</td>
<td>173</td>
<td>157</td>
<td>156</td>
<td>175</td>
</tr>
<tr>
<td>162</td>
<td>174</td>
<td>174</td>
<td>157</td>
<td>157</td>
<td>176</td>
</tr>
<tr>
<td>163</td>
<td>174</td>
<td>175</td>
<td>155</td>
<td>159</td>
<td>176</td>
</tr>
<tr>
<td>164</td>
<td>174</td>
<td>175</td>
<td>153</td>
<td>162</td>
<td>177</td>
</tr>
<tr>
<td>164</td>
<td>174</td>
<td>174</td>
<td>153</td>
<td>163</td>
<td>176</td>
</tr>
</tbody>
</table>
SVG, manually produced

```xml
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
</svg>

<svg width="403px" height="223px"
xmlns="http://www.w3.org/2000/svg" version="1.1">
  <desc>Example line01 - lines expressed in user coordinates</desc>
  <rect x="1" y="1" width="400" height="220"
    fill="none" stroke="black" stroke-width="3" />

  <line x1="30" y1="200" x2="30" y2="30" />  <!-- Y axes-->
  <line x1="27.6" y1="200" x2="360" y2="200" />  <!-- X axes-->
</svg>

<svg>
  <g font-size="10" font-family="Verdana">
    <text x="10" y="25">(pulse)</text>
    <text x="365" y="203">(time)</text>
  </g>

  <g font-size="7" font-family="Verdana">
    <text x="9" y="200">(0)</text>
    <text x="6" y="180">(20)</text>
    <text x="6" y="160">(40)</text>
    <text x="6" y="140">(60)</text>
    <text x="6" y="120">(80)</text>
    <text x="4" y="100">(100)</text>
    <text x="4" y="80">(120)</text>
    <text x="4" y="60">(140)</text>
  </g>

  <g font-size="7" font-family="Verdana">
    <text x="45" y="210">(10)</text>
    <text x="85" y="210">(11)</text>
  </g>
</svg>
```
<!-- The lines that indicate the level X1 and Y1 on the X axes(time) X2 and Y2 indicate the level(pulse) -->
<line x1="35" y1="198" x2="35" y2="120" stroke-width="3" />
<line x1="40" y1="198" x2="40" y2="110" stroke-width="3" />
<line x1="45" y1="198" x2="45" y2="120" stroke-width="3" />
<line x1="50" y1="198" x2="50" y2="110" stroke-width="3" />
<line x1="55" y1="198" x2="55" y2="100" stroke-width="3" />
<line x1="60" y1="198" x2="60" y2="100" stroke-width="3" />
<line x1="65" y1="198" x2="65" y2="100" stroke-width="3" />
<line x1="70" y1="198" x2="70" y2="100" stroke-width="3" />
<line x1="75" y1="198" x2="75" y2="100" stroke-width="3" />
<line x1="80" y1="198" x2="80" y2="110" stroke-width="3" />
<line x1="85" y1="198" x2="85" y2="120" stroke-width="3" />
<line x1="90" y1="198" x2="90" y2="110" stroke-width="3" />
<line x1="95" y1="198" x2="95" y2="100" stroke-width="3" />
<line x1="100" y1="198" x2="100" y2="100" stroke-width="3" />
<line x1="105" y1="198" x2="105" y2="100" stroke-width="3" />
<line x1="110" y1="198" x2="110" y2="100" stroke-width="3" />
<line x1="115" y1="198" x2="115" y2="100" stroke-width="3" />
<line x1="120" y1="198" x2="120" y2="110" stroke-width="3" />
<line x1="125" y1="198" x2="125" y2="120" stroke-width="3" />
<line x1="130" y1="198" x2="130" y2="110" stroke-width="3" />
<line x1="135" y1="198" x2="135" y2="100" stroke-width="3" />
<line x1="140" y1="198" x2="140" y2="100" stroke-width="3" />
<line x1="145" y1="198" x2="145" y2="100" stroke-width="3" />
<line x1="150" y1="198" x2="150" y2="100" stroke-width="3" />
<line x1="155" y1="198" x2="155" y2="100" stroke-width="3" />
<line x1="160" y1="198" x2="160" y2="110" stroke-width="3" />
<line x1="165" y1="198" x2="165" y2="120" stroke-width="3" />
<line x1="170" y1="198" x2="170" y2="110" stroke-width="3" />
<line x1="175" y1="198" x2="175" y2="100" stroke-width="3" />
<line x1="180" y1="198" x2="180" y2="100" stroke-width="3" />
<line x1="185" y1="198" x2="185" y2="100" stroke-width="3" />
<line x1="190" y1="198" x2="190" y2="100" stroke-width="3" />
<line x1="195" y1="198" x2="195" y2="100" stroke-width="3" />
Reference
