UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

SMART CITY TOOL

(ANNEXE)

A Degree's Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

ANTONI OBIOLS SERRA

In partial fulfilment

of the requirements for the degree in

***ELECTRONIC SYSTEMS* ENGINEERING**

Advisor: JORDI FORNÉS

Barcelona, February 2015

Table of contents

Annexe A:

//Code of the first window

```
package my.enviornmentcomfort;

import java.io.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Properties;

public class EnviornmentComfort extends javax.swing.JFrame {

    public EnviornmentComfort() {
        initComponents();
        try{
        Properties props = new Properties();
                                                                    props.load(new
FileInputStream("src\\my\\enviornmentcomfort\\resources\\config.properties"));

        String setDesktop = props.getProperty("office");
        office.setText(setDesktop);
        String setName = props.getProperty("name");
        name.setText(setName);

        } catch (IOException ex){

        }
    }


    String principal = "";
```

```
String sub = "";
EnviornmentComfort2 envcomf = new EnviornmentComfort2();
static UtilsEC utils = new UtilsEC();
int a = 0;

public int getConfirmation() {
        return a;
}

public void setConfirmation(int a) {
        this.a = a;
}


@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    buttonGroup1 = new javax.swing.ButtonGroup();
    okButton = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    office = new javax.swing.JTextField();
    name = new javax.swing.JTextField();
    officeLabel = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    okButton.setText("OK");
    okButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            okButtonActionPerformed(evt);
        }
    });
```

```
                                                    jLabel1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/my/enviornmentcomfort/resources/ait_si
mbol.png"))); // NOI18N

        office.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                officeActionPerformed(evt);
            }
        });

        officeLabel.setText("Office:");

        jLabel3.setText("Name:");

                                        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(27, 27, 27)
                    .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 306,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
46, Short.MAX_VALUE)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                                                    .addComponent(officeLabel,
javax.swing.GroupLayout.PREFERRED_SIZE,                 51,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addGap(18, 18, 18)
                                .addComponent(office, javax.swing.GroupLayout.PREFERRED_SIZE,
262, javax.swing.GroupLayout.PREFERRED_SIZE))
                                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
```

```
                    .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE,
51, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                    .addComponent(name, javax.swing.GroupLayout.PREFERRED_SIZE,
262, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addComponent(okButton, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    80,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(40, 40, 40))
        );
        layout.setVerticalGroup(
           layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
           .addGroup(layout.createSequentialGroup()
              .addGap(33, 33, 33)
               .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING, false)
                    .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE,
108, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.BASELINE)
                        .addComponent(office, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                                .addComponent(officeLabel,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    20,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGap(18, 18, 18)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.BASELINE)
                        .addComponent(name, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                                .addComponent(jLabel3,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    20,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(okButton)))
```

```
                .addContainerGap(36, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void okButtonActionPerformed(java.awt.event.ActionEvent evt) {
        principal = office.getText();
        sub = name.getText();
        try{
            Properties props = new Properties();
                                                                    props.load(new
FileInputStream("src\\my\\enviornmentcomfort\\resources\\config.properties"));

            String user = props.getProperty("user");
            String password = props.getProperty("password");
               String url = props.getProperty("url");
            String driver = props.getProperty("driver");
            Class.forName(driver);

            Connection con;

            con = DriverManager.getConnection(url, user, password);
            String query = "SELECT * FROM user";

            Statement myStmt = con.createStatement();
            ResultSet rs = myStmt.executeQuery(query);

            while(rs.next()){
               String officeDatabase = rs.getString("office");
               String nameDatabase = rs.getString("user_name");
               if (principal.equals(officeDatabase) && sub.equals(nameDatabase)){
                  a = 1;
                  setConfirmation(a);
               }
```

```
            }
        } catch (IOException | ClassNotFoundException | SQLException ex){
            System.out.println(ex);
        }

        if(getConfirmation() == 0){
            new chooseMap().setVisible(true);
        }
        else{
            String workPlace = utils.sumString(principal, sub);
            envcomf.setVisible(true);
            envcomf.setWorkPlace(workPlace);
            try{
                                                    FileInputStream in = new
FileInputStream("src\\my\\enviornmentcomfort\\resources\\config.properties");
                Properties props = new Properties();
                props.load(in);
                in.close();

                                                    FileOutputStream out = new
FileOutputStream("src\\my\\enviornmentcomfort\\resources\\config.properties");
                props.setProperty("office",principal);
                props.setProperty("name", sub);
                props.store(out,null);
                out.close();

            } catch (IOException ex){
                System.out.println(ex);
            }
            dispose();
        }
    }

    private void officeActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
```

```
    }


    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional)
">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
feel.
                                                                * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
        */
        try {
                                        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(EnviornmentComfort.class.getName()).log(java
.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(EnviornmentComfort.class.getName()).log(java
.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(EnviornmentComfort.class.getName()).log(java
.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(EnviornmentComfort.class.getName()).log(java
.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>


        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
```

```
            public void run(){
                new EnviornmentComfort().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.ButtonGroup buttonGroup1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JTextField name;
    private javax.swing.JTextField office;
    private javax.swing.JLabel officeLabel;
    private javax.swing.JButton okButton;
    // End of variables declaration
}
```

//Code from the second window

```
package my.enviornmentcomfort;
import java.io.FileInputStream;
import java.sql.*;
import java.util.Properties;
import java.util.logging.Level;
import java.util.logging.Logger;



public class EnviornmentComfort2 extends javax.swing.JFrame {

        public EnviornmentComfort2() {
```

```
    initComponents();
}

String thermal_comfort = "Comfortable";
double type_of_activity = 1.2;
double type_of_clothes = 0.748;
String window = "Closed";
String door = "Closed";
String air_conditioning = "Off";
String heating = "Off";
String tolerance = "-";

String workPlace;

public void setWorkPlace(String workPlace){
    this.workPlace = workPlace;
}



@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    buttonGroup1 = new javax.swing.ButtonGroup();
    exit = new javax.swing.JButton();
    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    Clothes = new javax.swing.JComboBox();
    Activity = new javax.swing.JComboBox();
    jPanel2 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    cool = new javax.swing.JRadioButton();
    cold = new javax.swing.JRadioButton();
```

```
        slightlyCool = new javax.swing.JRadioButton();
        comfortable = new javax.swing.JRadioButton();
        warm = new javax.swing.JRadioButton();
        slightlyWarm = new javax.swing.JRadioButton();
        hot = new javax.swing.JRadioButton();
        jPanel3 = new javax.swing.JPanel();
        jLabel3 = new javax.swing.JLabel();
        windowOpen = new javax.swing.JRadioButton();
        doorOpen = new javax.swing.JRadioButton();
        airOn = new javax.swing.JRadioButton();
        heatOn = new javax.swing.JRadioButton();
        jPanel4 = new javax.swing.JPanel();
        jLabel4 = new javax.swing.JLabel();
        tooCold = new javax.swing.JButton();
        tooHot = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        exit.setText("Save & Exit");
        exit.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                exitActionPerformed(evt);
            }
        });

                jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("Personal
Information"));

        jLabel1.setText("Select the clothes you are wearing and the activity you are doing:");

                Clothes.setModel(new javax.swing.DefaultComboBoxModel(new String[]
{ "(default)", "Jacket, shirt with long sleeves, trousers/dress, tie, shoes", "Jacket, open
neck shirt, trousers/dress, shoes", "Shirt with long sleeves, trousers/dress, tie, shoes",
"Shirt with long sleeves, trousers/dress, shoes", "Sweater, shirt, trousers, shoes", "Shirt
with short sleeves, trousers, shoes", "Shirt with short sleeves, trousers, sandals", "Shirt
with short sleeves, short/skirt, sandals", "Shirt with short sleeves, short/skirt, sandals" }));
```

```
        Clothes.setSelectedIndex(4);
        Clothes.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                ClothesActionPerformed(evt);
            }
        });

        Activity.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "(default)", "Seated, relaxed", "Sedentary activity (ex. computer, reading)", "Standing, light activity (ex. oral presentation)", "Standing, medium activity", "Walking", "Cycling, running" }));
        Activity.setSelectedIndex(2);
        Activity.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                ActivityActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(Clothes, 0, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 328, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(0, 0, Short.MAX_VALUE))
                    .addComponent(Activity, 0, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addContainerGap())
        );
```

```
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATE
D)
                .addComponent(Clothes, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(Activity, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())
        );


                jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder("Thermal
Comfort"));


        jLabel2.setText("Choose one of the options:");


        buttonGroup1.add(cool);
        cool.setText("Cool");
        cool.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                coolActionPerformed(evt);
            }
        });


        buttonGroup1.add(cold);
        cold.setText("Cold");
        cold.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                coldActionPerformed(evt);
```

```
    }
});

buttonGroup1.add(slightlyCool);
slightlyCool.setText("Slightly Cool");
slightlyCool.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        slightlyCoolActionPerformed(evt);
    }
});

buttonGroup1.add(comfortable);
comfortable.setSelected(true);
comfortable.setText("Comfortable");
comfortable.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        comfortableActionPerformed(evt);
    }
});

buttonGroup1.add(warm);
warm.setText("Warm");
warm.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        warmActionPerformed(evt);
    }
});

buttonGroup1.add(slightlyWarm);
slightlyWarm.setText("Slightly Warm");
slightlyWarm.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        slightlyWarmActionPerformed(evt);
    }
```

```
        });

        buttonGroup1.add(hot);
        hot.setText("Hot");
        hot.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                hotActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel2)
                                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
                                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
                .addContainerGap(10, Short.MAX_VALUE)
                .addComponent(cold)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(cool)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(slightlyCool)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(comfortable)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(slightlyWarm)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(warm)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
            .addComponent(hot)
            .addContainerGap())
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
          .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(7, 7, 7)
              .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE)
             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATE
D)
             .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
               .addComponent(cold)
               .addComponent(cool)
               .addComponent(slightlyCool)
               .addComponent(comfortable)
               .addComponent(slightlyWarm)
               .addComponent(warm)
               .addComponent(hot))
            .addGap(0, 0, Short.MAX_VALUE))
        );

        jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder("Actions"));

        jLabel3.setText("Interaction with the office:");

        windowOpen.setText("The window is open");
        windowOpen.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                windowOpenActionPerformed(evt);
            }
        });
```

```
        doorOpen.setText("The door is open");
        doorOpen.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                doorOpenActionPerformed(evt);
            }
        });

        airOn.setText("The air conditioning is on");
        airOn.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                airOnActionPerformed(evt);
            }
        });

        heatOn.setText("Heating is on");
        heatOn.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                heatOnActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
        jPanel3.setLayout(jPanel3Layout);
        jPanel3Layout.setHorizontalGroup(
            jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                    .addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGroup(jPanel3Layout.createSequentialGroup()
                    .addContainerGap()
                    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
```

```
                    .addComponent(windowOpen)
                    .addComponent(doorOpen)
                    .addComponent(airOn))
                .addGap(0, 0, Short.MAX_VALUE)))
            .addContainerGap())
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(heatOn)
                                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
    jPanel3Layout.setVerticalGroup(
        jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addComponent(jLabel3)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(windowOpen)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(doorOpen)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(airOn)
              .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(heatOn))
    );

    jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder("Tolerance"));

    jLabel4.setText("Is your temperature tolerance over?");

    tooCold.setText("Too Cold");
    tooCold.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            tooColdActionPerformed(evt);
```

```
        }
    });

    tooHot.setText("Too Hot");
    tooHot.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            tooHotActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
    jPanel4.setLayout(jPanel4Layout);
    jPanel4Layout.setHorizontalGroup(
        jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
        .addGroup(jPanel4Layout.createSequentialGroup()
                .addComponent(jLabel4,  javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addContainerGap())
        .addGroup(jPanel4Layout.createSequentialGroup()
            .addContainerGap()
                .addComponent(tooCold, javax.swing.GroupLayout.PREFERRED_SIZE, 83,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATE
D)
                .addComponent(tooHot, javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(28, Short.MAX_VALUE))
    );
    jPanel4Layout.setVerticalGroup(
        jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
        .addGroup(jPanel4Layout.createSequentialGroup()
            .addContainerGap()
                .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATE
D)
            .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                .addComponent(tooCold, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(tooHot, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addContainerGap())
        );


                javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                    .addGroup(layout.createSequentialGroup()
                        .addContainerGap()
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING, false)
                            .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
                    .addGroup(layout.createSequentialGroup()
                        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                        .addComponent(exit, javax.swing.GroupLayout.PREFERRED_SIZE, 151,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(49, 49, 49)))
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                    .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
```

```
            .addGap(22, 22, 22))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addContainerGap()
             .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                        .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATE
D)
             .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING, false)
                        .addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATE
D)
                    .addComponent(exit, javax.swing.GroupLayout.PREFERRED_SIZE, 55,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(7, 7, 7))
        );

        pack();
    }// </editor-fold>

    private void exitActionPerformed(java.awt.event.ActionEvent evt) {
          //If the activity, the clothes and the thermal comfort are selected, the data will be
stored.
          //Until then, the program will keep asking to be choosen, unless toHot or toCold
buttons are pressed.
        //All the data will be safed in the datauser table of thermal_comfort database
```

```
        try{

            Properties props = new Properties();
                props.load(new
FileInputStream("src\\my\\enviornmentcomfort\\resources\\config.properties"));

                String user = props.getProperty("user");
                String password = props.getProperty("password");
                String url = props.getProperty("url");
            String driver = props.getProperty("driver");
            Class.forName(driver);

            Connection con;

            con = DriverManager.getConnection(url, user, password);

              String query = "INSERT INTO datauser (datum, work_place, type_of_clothes,
type_of_activity," +
            "thermal_comfort, window, door, air_conditioning, heating, tolerance)" +
            "VALUES(?,?,?,?,?,?,?,?,?,?)";

        try (PreparedStatement myStmt = con.prepareStatement(query)) {
            // set params
            //myStmt.setInt(1, 1);
            myStmt.setTimestamp(1, new Timestamp(System.currentTimeMillis()));
            myStmt.setString(2, workPlace);
            myStmt.setDouble(3, type_of_clothes);
            myStmt.setDouble(4, type_of_activity);
            myStmt.setString(5, thermal_comfort);
            myStmt.setString(6, window);
            myStmt.setString(7, door);
            myStmt.setString(8, air_conditioning);
            myStmt.setString(9, heating);
            myStmt.setString(10, tolerance);
```

```
            // execute SQL
            myStmt.executeUpdate();
            myStmt.close();
        }

        con.close();

    } catch (ClassNotFoundException ex){
        System.out.println(ex);
       } catch (Exception e) {
            System.out.println(e);
    }
        System.exit(0);

}

private void slightlyWarmActionPerformed(java.awt.event.ActionEvent evt) {
    thermal_comfort = "Slightly Warm";
}

private void hotActionPerformed(java.awt.event.ActionEvent evt) {
    thermal_comfort = "Hot";
}

private void coldActionPerformed(java.awt.event.ActionEvent evt) {
    thermal_comfort = "Cold";
}

private void ClothesActionPerformed(java.awt.event.ActionEvent evt) {
    //Give the value needed for the PMV algorithm
    switch(Clothes.getSelectedIndex()){
        case 1:
            type_of_clothes = 0.948;
```

```
            break;
        case 2:
            type_of_clothes = 0.903;
            break;
        case 3:
            type_of_clothes = 0.800;
            break;
        case 4:
            type_of_clothes = 0.748;
            break;
        case 5:
            type_of_clothes = 0.703;
            break;
        case 6:
            type_of_clothes = 0.600;
            break;
        case 7:
            type_of_clothes = 0.516;
            break;
        case 8:
            type_of_clothes = 0.400;
            break;
        case 9:
            type_of_clothes = 0.303;
            break;
        default:
            break;
    }
}

private void coolActionPerformed(java.awt.event.ActionEvent evt) {
    thermal_comfort = "Cool";
}
```

```
private void slightlyCoolActionPerformed(java.awt.event.ActionEvent evt) {
    thermal_comfort = "Slightly Cool";
}

private void comfortableActionPerformed(java.awt.event.ActionEvent evt) {
    thermal_comfort = "Comfortable";
}

private void warmActionPerformed(java.awt.event.ActionEvent evt) {
    thermal_comfort = "Warm";
}

private void ActivityActionPerformed(java.awt.event.ActionEvent evt) {
    //Give the value needed for the PMV algorithm
    switch(Activity.getSelectedIndex()){
        case 1:
            type_of_activity = 1.0;
            break;
        case 2:
            type_of_activity = 1.2;
            break;
        case 3:
            type_of_activity = 1.6;
            break;
        case 4:
            type_of_activity = 2.0;
            break;
        case 5:
            type_of_activity = 2.4;
            break;
        case 6:
            type_of_activity = 3.95;
            break;
        default:
```

```
            break;
        }
    }

    private void airOnActionPerformed(java.awt.event.ActionEvent evt) {
        air_conditioning = "On";
    }

    private void windowOpenActionPerformed(java.awt.event.ActionEvent evt) {
        window = "Open";
    }

    private void doorOpenActionPerformed(java.awt.event.ActionEvent evt) {
        door = "Open";
    }

    private void tooColdActionPerformed(java.awt.event.ActionEvent evt) {
        //if tooCold botton is pressed, all the other data doesn't matter
        thermal_comfort = "-";
        type_of_activity = 0;
        type_of_clothes = 0;
        window = "-";
        door = "-";
        air_conditioning = "-";
        heating = "-";
        tolerance = "Too Cold";

        //Start the conection
        try{
            Class.forName("com.mysql.jdbc.Driver");

            Connection con;
```

```
                                                                    con = DriverManager.getConnection("jdbc:mysql://localhost:3306/thermal_comfort", "root", "0810l5.serra_");

        //Enter new row of data to the datauser table
            String query = "INSERT INTO datauser (datum, work_place, type_of_clothes, type_of_activity," +
            "thermal_comfort, window, door, air_conditioning, heating, tolerance)" +
            "VALUES(?,?,?,?,?,?,?,?,?,?)";

        // set params
        try (PreparedStatement myStmt = con.prepareStatement(query)) {

            myStmt.setTimestamp(1, new Timestamp(System.currentTimeMillis()));
            myStmt.setString(2, workPlace);
            myStmt.setDouble(3, type_of_clothes);
            myStmt.setDouble(4, type_of_activity);
            myStmt.setString(5, thermal_comfort);
            myStmt.setString(6, window);
            myStmt.setString(7, door);
            myStmt.setString(8, air_conditioning);
            myStmt.setString(9, heating);
            myStmt.setString(10, tolerance);

            // execute SQL
            myStmt.executeUpdate();
        }
            //Close the conection
            con.close();


    } catch (ClassNotFoundException ex){
        System.out.println("No class found");
        } catch (SQLException e) {
            System.out.println("Error in SQL");
```

```
            Logger.getLogger(EnviornmentComfort2.class.getName()).log(Level.SEVERE,
null, e);
        }

        System.exit(0);
    }

    private void tooHotActionPerformed(java.awt.event.ActionEvent evt) {
        //if tooHot botton is pressed, all the other data doesn't matter
        thermal_comfort = "-";
        type_of_activity = 0;
        type_of_clothes = 0;
        window = "-";
        door = "-";
        air_conditioning = "-";
        heating = "-";
        tolerance = "Too Hot";

        try{
            //Start connection
            Class.forName("com.mysql.jdbc.Driver");

            Connection con;

                                                                    con        =
DriverManager.getConnection("jdbc:mysql://localhost:3306/thermal_comfort",        "root",
"0810l5.serra_");

            //Add new row to the datauser table
                String query = "INSERT INTO datauser (datum, work_place, type_of_clothes,
type_of_activity," +
                "thermal_comfort, window, door, air_conditioning, heating, tolerance)" +
                "VALUES(?,?,?,?,?,?,?,?,?,?)";

            try (PreparedStatement myStmt = con.prepareStatement(query)) {
```

```
            // set params
            myStmt.setTimestamp(1, new Timestamp(System.currentTimeMillis()));
            myStmt.setString(2, workPlace);
            myStmt.setDouble(3, type_of_clothes);
            myStmt.setDouble(4, type_of_activity);
            myStmt.setString(5, thermal_comfort);
            myStmt.setString(6, window);
            myStmt.setString(7, door);
            myStmt.setString(8, air_conditioning);
            myStmt.setString(9, heating);
            myStmt.setString(10, tolerance);

            // execute SQL
            myStmt.executeUpdate();
        }
            //close the conection
            con.close();


    } catch (ClassNotFoundException ex){
        System.out.println("No class found");
        } catch (SQLException e) {
            System.out.println("Error in SQL");
             Logger.getLogger(EnviornmentComfort2.class.getName()).log(Level.SEVERE,
null, e);
    }

    System.exit(0);
  }

  private void heatOnActionPerformed(java.awt.event.ActionEvent evt) {
    heating = "On";
  }

  public static void main(String args[]) {
```

```
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(EnviornmentComfort2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(EnviornmentComfort2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(EnviornmentComfort2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(EnviornmentComfort2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            @Override
            public void run() {
                new EnviornmentComfort2().setVisible(true);
            }
```

```
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JComboBox Activity;
    private javax.swing.JComboBox Clothes;
    private javax.swing.JRadioButton airOn;
    private javax.swing.ButtonGroup buttonGroup1;
    private javax.swing.JRadioButton cold;
    private javax.swing.JRadioButton comfortable;
    private javax.swing.JRadioButton cool;
    private javax.swing.JRadioButton doorOpen;
    private javax.swing.JButton exit;
    private javax.swing.JRadioButton heatOn;
    private javax.swing.JRadioButton hot;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanel4;
    private javax.swing.JRadioButton slightlyCool;
    private javax.swing.JRadioButton slightlyWarm;
    private javax.swing.JButton tooCold;
    private javax.swing.JButton tooHot;
    private javax.swing.JRadioButton warm;
    private javax.swing.JRadioButton windowOpen;
    // End of variables declaration
}
```

// Code of 3rd window (fail to access)

```
package my.enviornmentcomfort;

public class chooseMap extends javax.swing.JFrame {

    public chooseMap() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jButton1 = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jButton1.setText("OK");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jLabel1.setText("Please, select your place of work");

                                            javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(79, 79, 79)
                 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
```

```
                                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                    .addComponent(jLabel1)
                    .addGap(73, 73, 73))
                                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                    .addGap(55, 55, 55)
                        .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addGap(129, 129, 129))))
        );
        layout.setVerticalGroup(
           layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
           .addGroup(layout.createSequentialGroup()
              .addGap(23, 23, 23)
                 .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
              .addComponent(jButton1)
              .addContainerGap(27, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }

    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
         //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional)
">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
feel.
```

```
                                                * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
                                            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(chooseMap.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(chooseMap.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(chooseMap.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(chooseMap.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new chooseMap().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
```

```
    // End of variables declaration
}
```

//Code of UtilsEC

```
package my.enviornmentcomfort;

public class UtilsEC {

    public String sumString(String string1, String string2){
        String string3 = string1 + " " + string2;
        return string3;
    }

    public static String getFirstString (String string1){
        String firstString = "";
        char c;
        int i;
        for (i = 0; i < string1.length(); i++){
            c = string1.charAt(i);
            if (c != ' '){
                firstString += Character.toString(c);
            }
            else
            {
                break;
            }
        }
        return firstString;
    }

}
```

// config.properties of the first application

user=root

name=Tony

url=jdbc\:mysql\://localhost\:3306/thermal_comfort

password=password

office=E407

driver=com.mysql.jdbc.Driver

Annexe B:

// Second application

// Algorithm

```
package algorithm;


public class Algorithm {


    double Ta; //air temperature ºC
    double Tr; // mean radiant temperature, ºC
    double Met; //metabolic rate, met (1 met = 58.2 W/m2 = 18.4 Btu/h ft2)
    double Icl;// insulation level of clothing, clo (1 clo = 0.155 m2.K/W = 0.88 ft2.F.h/Btu)
    double v = 0.1;
     double Patm = 1; // atmospheric pressure, atm (viena is at aproximately 171 meters above the sea
    double relhum; //humidity ratio
    double Wa;
    double Tcl;
    double Rcl;
    double fcl;
    double Pvw;
    double hc1;
    double hc2;
    double hc;
    double L;
    double work;
    double hr;
    double Tsk;
    double qSweat;
    double WSatSk;
    double qRespSen;
    double qRespLat;
    double qEvapDiff;
    double Mw;
```

```
int id;
String thermal_comfort;
String Window;
String Door;
String air_conditioning;
String heating;
String tolerance;
String datum;
String workplace;

static algortihmUtils prova = new algortihmUtils();

public double pmv(){
    Tr = Ta;
    Wa = prova.HumRatRH(Ta * 1.8 + 32, relhum/100, Patm);
       Mw = Met * 58.2;
    Rcl = Icl * 0.155;
    fcl = 1.05 + 0.1 * Icl;
    hc = 12.1 * Math.sqrt(v);
    hr = 4.2;
    Tsk = 35.7 - 0.0275 * Mw;
    qSweat = 0.42 * (Mw - 58.15);

    WSatSk = prova.HumRatRH(Tsk * 1.8 + 32, 1, Patm);
    qRespSen = 0.0014 * Mw * (34 - Ta);
    qRespLat = 2.78 * Mw * (0.0365 - Wa);
    qEvapDiff = 491 * (WSatSk - Wa);

    Tcl = (Tsk + Rcl * fcl * (hc * Ta + hr * Tr)) / (1 + Rcl * fcl * (hc + hr));

    if (Tcl < Ta){
       Tcl = Ta;
    }
    hc1 = 2.38 * Math.pow((Tcl - Ta), 0.24);
```

```
        hc2 = 12.1 * Math.sqrt(v);

        if (hc1 > hc2){
            hc = hc1;
        }
        else{
            hc = hc2;
        }

        Tcl = (Tsk + Rcl * fcl * (hc * Ta + hr * Tr)) / (1 + Rcl * fcl * (hc + hr));

         L = Mw - fcl * hr * (Tcl - Tr) - fcl * hc * (Tcl - Ta) - qEvapDiff - qSweat - qRespSen -
qRespLat;

        double PMV = L * (0.303 * Math.exp(-0.036 * Mw) + 0.028) - 0.5;
        return PMV;
    }

    public double ppd(){
            double PPD = 1 - 0.95 * Math.exp(-0.003353 * Math.pow(pmv(),4) - 0.2179 *
Math.pow(pmv(),2));
        PPD = PPD * 100; // %
            return PPD;
    }

    public double getTemperature() {
            return Ta;
    }

    public void setTemperature(double Ta) {
            this.Ta = Ta;
    }

    public double getHumidity() {
            return relhum;
```

```
    }

    public void setHumidity(double relhum) {
        this.relhum = relhum;
    }

    public double getActivity() {
        return Met;
    }

    public void setActivity(double Met) {
        this.Met = Met;
    }

     public double getClothes() {
        return Icl;
    }

    public void setClothes(double Icl) {
        this.Icl = Icl;
    }

    public double getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }
}
```

```
// Import data from excel file
package algorithm;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;

public class ChooseTemperature {

   static String numSensor;
   static String date;
   static String hour;
   static double temperature;
   static double relHumidity;
   static int rowNum;
   static String[][] sensorArray = new String[10000][5];

   public String[][] readFromExcel(){
      try{
         Properties props = new Properties();
         props.load(new FileInputStream("src\\utils\\config.properties"));

         String user = props.getProperty("user");
         String password = props.getProperty("password");
         String url = props.getProperty("url");
         String driver = props.getProperty("driver");
         String excelurl = props.getProperty("excelurl");
```

```
        String sheetname = props.getProperty("sheetname");

        Class.forName(driver);
        Connection con = DriverManager.getConnection(url, user, password);
        File excel = new File (excelurl);
        FileInputStream fis = new FileInputStream(excel);

        HSSFWorkbook wb = new HSSFWorkbook(fis);
        HSSFSheet ws = wb.getSheet(sheetname);

        rowNum = ws.getLastRowNum()+1;

        for (int i=1; i<rowNum; i++){
            HSSFRow row = ws.getRow(i);

            numSensor = row.getCell(0).getStringCellValue();
            date = row.getCell(1).getStringCellValue();
            hour = row.getCell(2).getStringCellValue();
            temperature = row.getCell(3).getNumericCellValue();
            relHumidity = row.getCell(4).getNumericCellValue();

            String temp = String.valueOf(temperature);
            String humid = String.valueOf(relHumidity);

            sensorArray[i][0] = numSensor;
            sensorArray[i][1] = date;
            sensorArray[i][2] = hour;
            sensorArray[i][3] = temp;
            sensorArray[i][4] = humid;
        }
    }
    catch (IOException | ClassNotFoundException | SQLException e)
    {
        System.out.println(e);
```

```
    }
    return sensorArray;
}

public static String getFirstString(String string1){
    String firstString = "";
    char c;
    int i;
    for (i = 0; i < string1.length(); i++){
        c = string1.charAt(i);
        if (c != ' '){
            firstString += Character.toString(c);
        }
        else
        {
            break;
        }
    }
    return firstString;
}

private static String getHour(String string2){
    string2 = string2.substring(0,2);
    return string2;
}

public double[] getTempHum(String workplace, String date, String time){
    int i = 1;
    int a = 0;
    double[] result = new double[2];
    String[][] read = readFromExcel();

    workplace = getFirstString(workplace);
    String time1 = getHour(time);
```

```
        while (a == 0 && i < rowNum){
            String timeExcel = getHour(read[i][2]);
                                if(date.equals(read[i][1])  &&  workplace.equals(read[i][0])  &&
time1.equals(timeExcel)){
                double temp = Double.parseDouble(read[i][3]);
                double humi = Double.parseDouble(read[i][4]);
                result[0] = temp;
                result[1] = humi;

                a = 1;
            }
            i++;
        }
        return result;
    }
}
```

//Graphic window for the 2nd application

```
package algorithm;

public class ImportExcelAndCalculatePMV extends javax.swing.JFrame {

    public ImportExcelAndCalculatePMV() {
        initComponents();
    }

    SQLConnect connection = new SQLConnect();
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jImport = new javax.swing.JButton();
        close = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jImport.setText("Import");
        jImport.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jImportActionPerformed(evt);
            }
        });

        close.setText("Close");
        close.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                closeActionPerformed(evt);
            }
        });
```

```
jLabel1.setText("<html>Remember that the path from the config.properties file
<BR>from where the excel file will be imported has to be correct.</html>");


                                        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addGap(31, 31, 31)
         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
           .addGroup(layout.createSequentialGroup()
             .addComponent(jImport, javax.swing.GroupLayout.PREFERRED_SIZE,
120, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(36, 36, 36)
              .addComponent(close, javax.swing.GroupLayout.PREFERRED_SIZE,
113, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE,
290, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(24, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
        .addGap(26, 26, 26)
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING, false)
            .addComponent(jImport, javax.swing.GroupLayout.DEFAULT_SIZE, 34,
Short.MAX_VALUE)
              .addComponent(close, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
```

```
                .addContainerGap(25, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void jImportActionPerformed(java.awt.event.ActionEvent evt) {
        connection.connectAll();
    }

    private void closeActionPerformed(java.awt.event.ActionEvent evt) {
        dispose();
    }

    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
         //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional)
">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
feel.
                                                * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
                                    for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(ImportExcelAndCalculatePMV.class.getName()
).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
```

```
            java.util.logging.Logger.getLogger(ImportExcelAndCalculatePMV.class.getName()
).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(ImportExcelAndCalculatePMV.class.getName()
).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(ImportExcelAndCalculatePMV.class.getName()
).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new ImportExcelAndCalculatePMV().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton close;
    private javax.swing.JButton jImport;
    private javax.swing.JLabel jLabel1;
    // End of variables declaration
}
```

//Main

```
package algorithm;

import java.io.FileInputStream;
import java.io.IOException;
import java.sql.*;
import java.util.Properties;
import java.util.logging.Level;
```

```
import java.util.logging.Logger;

public class SQLConnect {

    static Algorithm algor = new Algorithm();
    static ChooseTemperature choose = new ChooseTemperature();
    static Utils util = new Utils();
    String thermal_comfort;
    double type_of_activity;
    double type_of_clothes;
    String window;
    String door;
    String air_conditioning;
    String heating;
    String tolerance;
    String workPlace;
    String date;
    String time;
    String datetime;
    static double pmv;
    static double ppd;
    static double temperature;
    static double humidity;
    int id;
    int[] arrayId;
    static int size;

    public static int[] getAllId(ResultSet resultSet) {
        int[] id = new int[1000];
        int i = 0;
        try {
            while(resultSet.next()){
                id[i] = resultSet.getInt("id");
                i++;
```

```
            }
        }
        catch(SQLException ex) {
        }
        return id;
    }

    public static int getRowCount(ResultSet resultSet) {
        int size = 0;
        try {
            resultSet.last();
            size = resultSet.getRow();
            resultSet.beforeFirst();
        }
        catch(SQLException ex) {
            return 0;
        }
        return size;
    }

    public void setIdArray(int[] arrayId){
        this.arrayId = arrayId;
    }

    public int[] getIdArray(){
        return arrayId;
    }

    public String getDate() {
            return date;
    }

    public void setDate(String date) {
            this.date = date;
```

```
}

public String getTime() {
	return time;
}

public void setTime(String time) {
	this.time = time;
}

public String getDateTime() {
	return datetime;
}

public void setDateTime(String datetime) {
	this.datetime = datetime;
}

public String getThermalComfort() {
	return thermal_comfort;
}

public void setThermalComfort(String thermal_comfort) {
	this.thermal_comfort = thermal_comfort;
}

public String getWindow() {
	return window;
}

public void setWindow(String Window) {
	this.window = Window;
}
```

```
public String getDoor() {
        return door;
}

public void setDoor(String Door) {
        this.door = Door;
}

public String getAirCond() {
        return air_conditioning;
}

public void setAirCond(String airCond) {
    this.air_conditioning = airCond;
}

public String getHeating() {
        return heating;
}

public void setHeating(String heating) {
        this.heating = heating;
}

public String getTolerance() {
        return tolerance;
}

public void setTolerance(String tolerance) {
        this.tolerance = tolerance;
}

public String getWorkPlace() {
        return workPlace;
```

```
}

public void setWorkPlace(String workplace) {
        this.workPlace = workplace;
}

public void connectAll(){
        try{
        Properties props = new Properties();
        props.load(new FileInputStream("src\\utils\\config.properties"));

        String user = props.getProperty("user");
        String password = props.getProperty("password");
           String url = props.getProperty("url");
        String driver = props.getProperty("driver");
        Class.forName(driver);

        Connection con;
        con = DriverManager.getConnection(url, user, password);
        String query = "SELECT id FROM datauser";

        Statement myStmt = con.createStatement();
        ResultSet rs = myStmt.executeQuery(query);

        setIdArray(getAllId(rs));
        size = getRowCount(rs);

        rs.close();
        myStmt.close();

    } catch (ClassNotFoundException ex){
        System.out.println(ex);
        } catch (IOException | SQLException e) {
        System.out.println(e);
```

```
}

for (int a = 0; a < size; a++){
try{
   Properties props = new Properties();
   props.load(new FileInputStream("src\\utils\\config.properties"));

   String user = props.getProperty("user");
   String password = props.getProperty("password");
      String url = props.getProperty("url");
   String driver = props.getProperty("driver");
   Class.forName(driver);

   Connection con;
   con = DriverManager.getConnection(url, user, password);

   int[] idi = getIdArray();
   String query = "SELECT * FROM datauser WHERE id = ?";

   PreparedStatement myStmt = con.prepareStatement(query);
   myStmt.setInt(1, idi[a]);
   ResultSet rs = myStmt.executeQuery();


   rs.next();
   //Retrieve by column name

   String datum = rs.getString("datum");
   setDateTime(datum);
   String date = util.separateDateTime(datum)[0];
   String time = util.separateDateTime(datum)[1];
   setDate(date);
   setTime(time);
   String thermal_comfort = rs.getString("thermal_comfort");
```

```
        setThermalComfort(thermal_comfort);
        double type_of_activity = rs.getDouble("type_of_activity");
        algor.setActivity(type_of_activity);
        double type_of_clothes = rs.getDouble("type_of_clothes");
        algor.setClothes(type_of_clothes);
        String window = rs.getString("window");
        setWindow(window);
        String door = rs.getString("door");
        setDoor(door);
        String air_conditioning = rs.getString("air_conditioning");
        setAirCond(air_conditioning);
        String heating = rs.getString("heating");
        setHeating(heating);
        String tolerance = rs.getString("tolerance");
        setTolerance(tolerance);
        String workPlace = rs.getString("work_place");
        setWorkPlace(workPlace);
        temperature = choose.getTempHum(workPlace, date, time)[0];
        humidity = choose.getTempHum(workPlace, date, time)[1];
        algor.setTemperature(temperature);
        algor.setHumidity(humidity);

        pmv = algor.pmv();
        ppd = algor.ppd();

        rs.close();
        myStmt.close();
        con.close();

    } catch (ClassNotFoundException ex){
        System.out.println(ex);
       } catch (IOException | SQLException e) {
        System.out.println(e);
    }
```

```
String datum = getDateTime();
String thermal_comfort = getThermalComfort();
double type_of_activity = algor.getActivity();
double type_of_clothes = algor.getClothes();
String window = getWindow();
String door = getDoor();
String air_conditioning = getAirCond();
String heating = getHeating();
String tolerance = getTolerance();
double temp = algor.getTemperature();
String workplace = getWorkPlace();
double humi = algor.getHumidity();

try{
   Properties props = new Properties();
   props.load(new FileInputStream("src\\utils\\config.properties"));

   String user = props.getProperty("user");
   String password = props.getProperty("password");
   String url = props.getProperty("url");
   String driver = props.getProperty("driver");
   Class.forName(driver);

   Connection con;
   con = DriverManager.getConnection(url, user, password);

        String query = "INSERT INTO thermal (datum, work_place, type_of_clothes,
type_of_activity," +
        "thermal_comfort, window, door, air_conditioning, heating, tolerance, pmv, ppd,
temperature, relative_humidity)" +
        "VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?)";

   if (humi != 0.0){
      try (PreparedStatement myStmt = con.prepareStatement(query)) {
```

```
            myStmt.setString(1, datum);
            myStmt.setString(2, workplace);
            myStmt.setDouble(3, type_of_clothes);
            myStmt.setDouble(4, type_of_activity);
            myStmt.setString(5, thermal_comfort);
            myStmt.setString(6, window);
            myStmt.setString(7, door);
            myStmt.setString(8, air_conditioning);
            myStmt.setString(9, heating);
            myStmt.setString(10, tolerance);
            myStmt.setDouble(11, pmv);
            myStmt.setDouble(12, ppd);
            myStmt.setDouble(13, temp);
            myStmt.setDouble(14, humi);

            myStmt.executeUpdate();

            try{

               int[] idi = getIdArray();
               String sql = "DELETE FROM datauser WHERE id = ?";
               PreparedStatement stmt = con.prepareStatement(sql);

               stmt.setInt(1,idi[a]);
               stmt.executeUpdate();
               stmt.close();

            }catch (SQLException e) {
               System.out.println(e);
            }
         }
      }
      con.close();
```

```
        } catch (IOException | SQLException |ClassNotFoundException e) {
            System.out.println(e);
        }
        }
    }

    public static void main(String args[]) {
        new ImportExcelAndCalculatePMV().setVisible(true);
    }
}
```

//Utils

```
package algorithm;

public class Utils {

    public String[] separateDateTime(String datetime){
        String[] result = new String[2];
        result[0] = datetime.substring(0,10);
        result[1] = datetime.substring(11, datetime.length());
        return result;
    }


}
```

//Algorithm utils

```
package algorithm;

public class algortihmUtils {
    static final double NMol = 0.62198; //ratio of molecular weights, Mvap/MAir
    static final double RHMax = 1; //maximum relative humidity, 1 or 100 (if percent)
    static final double tolRel = 0.000001; //relative error tolerance for iteration


    //SI units
    static final double HfgRef = 2501000; //heat of vaporization at 0C, J/kg
    static final double CpVap = 1805; //specific heat of water vapor, J/kg C
    static final double CpWat = 4186; //specific heat of liquid water, J/kg C
    static final double CpAir = 1006; //specific heat of dry air, J/kg C
    static final double RAir = 0.002833; //gas constant for air, RAir*kPaMult*1000 = J/kg C
    static final double kPaMult = 101.325; //multiplier to get kPascals from user pressure
    static final double TAbs = 273.15; //add to user temperature to get absolute temp
    static final double TKelMult = 1; //multiplier to get Kelvin from user temp
    static final double TAmb = 25; //typical temperature in user units (initial value)

    //constants for vapor pressure correlations
    static final double C1 = -5674.5359;
    static final double C2 = -0.51523058;
    static final double C3 = -0.009677843;
    static final double C4 = 0.00000062215701;
    static final double C5 = 2.0747825 * (1E-9);
    static final double C6 = -9.484024 * (1E-13);
    static final double C7 = 4.1635019;
    static final double C8 = -5800.2206;
    static final double C9 = -5.516256;
    static final double C10 = -0.048640239;
    static final double C11 = 0.000041764768;
    static final double C12 = -0.000000014452093;
    static final double C13 = 6.5459673;
```

```
    public double SatPress(double TArg){
        double T;
        double kPa = 0;
        T = (TArg + TAbs) * TKelMult;
        if(T < 273.15){
            kPa = Math.exp((C1/T) + C2 + T * C3 + T * T * (C4 + T * (C5 + C6 * T)) + C7 *
Math.log10(T));
        }
        else if(T > 273.15){
                kPa = Math.exp((C8/T) + C9 + T * (C10 + T * (C11 + T * C12)) + C13 *
Math.log10(T));
        }
        double SatPress = kPa / kPaMult;
        return SatPress;
    }

    public double HumRatRH(double T, double RH, double PAtm){
        double pw;
        pw = SatPress(T) * RH / RHMax;
        double HumRatRH = NMol * pw / (PAtm - pw);
        return HumRatRH;
    }
}
```

//config.properties from second application

```
user=root
password=password
url=jdbc:mysql://localhost:3306/thermal_comfort
driver=com.mysql.jdbc.Driver
excelurl=C:\\Users\\ANTONI1\\Desktop\\ANTONI\\sensors.xls
sheetname=Sheet1
```

Annexe C:

// 3rd application export data

// From database to excel file

```
import java.io.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Properties;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.ss.usermodel.*;


public class DatabaseToExcel {
   public void exportAll(){
      try{
         Properties props = new Properties();
         props.load(new FileInputStream("src\\utils\\config.properties"));

         String user = props.getProperty("user");
         String password = props.getProperty("password");
         String url = props.getProperty("url");
         String driver = props.getProperty("driver");
         String excelurl = props.getProperty("excelurl");

         Class.forName(driver);
         Connection con = DriverManager.getConnection(url, user, password);
         Statement st = con.createStatement();
         ResultSet rs = st.executeQuery("Select * from thermal");
         HSSFWorkbook workbook = new HSSFWorkbook();
```

```
HSSFSheet sheet = workbook.createSheet("thermalData");
HSSFRow rowhead = sheet.createRow((short) 0);
rowhead.createCell((short) 0).setCellValue("Date");
rowhead.createCell((short) 1).setCellValue("Place of Work");
rowhead.createCell((short) 2).setCellValue("Clothes");
rowhead.createCell((short) 3).setCellValue("Kind of activity");
rowhead.createCell((short) 4).setCellValue("Thermal Comfort");
rowhead.createCell((short) 5).setCellValue("Window");
rowhead.createCell((short) 6).setCellValue("Door");
rowhead.createCell((short) 7).setCellValue("Air Conditioning");
rowhead.createCell((short) 8).setCellValue("Heating");
rowhead.createCell((short) 9).setCellValue("Tolerance");
rowhead.createCell((short) 10).setCellValue("PMV");
rowhead.createCell((short) 11).setCellValue("PPD");
rowhead.createCell((short) 12).setCellValue("Temperature");
int i = 1;
while (rs.next()){
    HSSFRow row = sheet.createRow((short) i);
    row.createCell((short) 0).setCellValue(rs.getString("datum"));
    row.createCell((short) 1).setCellValue(rs.getString("work_place"));
    row.createCell((short) 2).setCellValue(rs.getString("type_of_clothes"));
    row.createCell((short) 3).setCellValue(rs.getString("type_of_activity"));
    row.createCell((short) 4).setCellValue(rs.getString("thermal_comfort"));
    row.createCell((short) 5).setCellValue(rs.getString("window"));
    row.createCell((short) 6).setCellValue(rs.getString("door"));
    row.createCell((short) 7).setCellValue(rs.getString("air_conditioning"));
    row.createCell((short) 8).setCellValue(rs.getString("heating"));
    row.createCell((short) 9).setCellValue(rs.getString("tolerance"));
    row.createCell((short) 10).setCellValue(rs.getString("pmv"));
    row.createCell((short) 11).setCellValue(rs.getString("ppd"));
    row.createCell((short) 12).setCellValue(rs.getString("temperature"));
    i++;
}
FileOutputStream fileOut = new FileOutputStream(excelurl);
```

```
            workbook.write(fileOut);
            fileOut.close();
        } catch (ClassNotFoundException | SQLException | FileNotFoundException e1) {
            System.out.println(e1);
        }  catch (IOException e) {
            System.out.println(e);
        }
    }

    public void exportOneDay(String date){
        try{
            Properties props = new Properties();
            props.load(new FileInputStream("src\\utils\\config.properties"));

            String user = props.getProperty("user");
            String password = props.getProperty("password");
            String url = props.getProperty("url");
            String driver = props.getProperty("driver");
            String excelurl = props.getProperty("excelurl");

            Class.forName(driver);
            Connection con = DriverManager.getConnection(url, user, password);
            Statement st = con.createStatement();
            ResultSet rs = st.executeQuery("Select * from thermal");
            HSSFWorkbook workbook = new HSSFWorkbook();
            HSSFSheet sheet = workbook.createSheet("thermalData");
            HSSFRow rowhead = sheet.createRow((short) 0);
            rowhead.createCell((short) 0).setCellValue("Date");
            rowhead.createCell((short) 1).setCellValue("Place of Work");
            rowhead.createCell((short) 2).setCellValue("Clothes");
            rowhead.createCell((short) 3).setCellValue("Kind of activity");
            rowhead.createCell((short) 4).setCellValue("Thermal Comfort");
            rowhead.createCell((short) 5).setCellValue("Window");
            rowhead.createCell((short) 6).setCellValue("Door");
```

```
            rowhead.createCell((short) 7).setCellValue("Air Conditioning");
            rowhead.createCell((short) 8).setCellValue("Heating");
            rowhead.createCell((short) 9).setCellValue("Tolerance");
            rowhead.createCell((short) 10).setCellValue("PMV");
            rowhead.createCell((short) 11).setCellValue("PPD");
            rowhead.createCell((short) 12).setCellValue("Temperature");
            int i = 1;
            while (rs.next()){
                String date1 = rs.getString("datum");
                date1 = date1.substring(0,10);
                if (date.equals(date1)){
                    HSSFRow row = sheet.createRow((short) i);
                    row.createCell((short) 0).setCellValue(rs.getString("datum"));
                    row.createCell((short) 1).setCellValue(rs.getString("work_place"));
                    row.createCell((short) 2).setCellValue(rs.getString("type_of_clothes"));
                    row.createCell((short) 3).setCellValue(rs.getString("type_of_activity"));
                    row.createCell((short) 4).setCellValue(rs.getString("thermal_comfort"));
                    row.createCell((short) 5).setCellValue(rs.getString("window"));
                    row.createCell((short) 6).setCellValue(rs.getString("door"));
                    row.createCell((short) 7).setCellValue(rs.getString("air_conditioning"));
                    row.createCell((short) 8).setCellValue(rs.getString("heating"));
                    row.createCell((short) 9).setCellValue(rs.getString("tolerance"));
                    row.createCell((short) 10).setCellValue(rs.getString("pmv"));
                    row.createCell((short) 11).setCellValue(rs.getString("ppd"));
                    row.createCell((short) 12).setCellValue(rs.getString("temperature"));
                    i++;
                }
            }
            FileOutputStream fileOut = new FileOutputStream(excelurl);
            workbook.write(fileOut);
            fileOut.close();
        } catch (ClassNotFoundException | SQLException | FileNotFoundException e1) {
            System.out.println(e1);
        }  catch (IOException e) {
```

```
            System.out.println(e);
        }
    }

    public void exportPeriod(String date, String date2){
        String actualDate = date;
        date2 = calculateNextDay(date2);
        try{
            Properties props = new Properties();
            props.load(new FileInputStream("src\\utils\\config.properties"));

            String user = props.getProperty("user");
            String password = props.getProperty("password");
            String url = props.getProperty("url");
            String driver = props.getProperty("driver");
            String excelurl = props.getProperty("excelurl");

            Class.forName(driver);
            Connection con = DriverManager.getConnection(url, user, password);

            Statement st = con.createStatement();
            ResultSet rs = st.executeQuery("Select * from thermal");
            HSSFWorkbook workbook = new HSSFWorkbook();
            HSSFSheet sheet = workbook.createSheet("thermalData");

            HSSFRow rowhead = sheet.createRow((short) 0);
            rowhead.createCell((short) 0).setCellValue("Date");
            rowhead.createCell((short) 1).setCellValue("Place of Work");
            rowhead.createCell((short) 2).setCellValue("Clothes");
            rowhead.createCell((short) 3).setCellValue("Kind of activity");
            rowhead.createCell((short) 4).setCellValue("Thermal Comfort");
            rowhead.createCell((short) 5).setCellValue("Window");
            rowhead.createCell((short) 6).setCellValue("Door");
            rowhead.createCell((short) 7).setCellValue("Air Conditioning");
```

```
rowhead.createCell((short) 8).setCellValue("Heating");
rowhead.createCell((short) 9).setCellValue("Tolerance");
rowhead.createCell((short) 10).setCellValue("PMV");
rowhead.createCell((short) 11).setCellValue("PPD");
rowhead.createCell((short) 12).setCellValue("Temperature");
int i = 1;

while (rs.next() && !date2.equals(actualDate)){
   String date1 = rs.getString("datum");
   date1 = date1.substring(0,10);
   if (!actualDate.equals(date1)){
      actualDate = calculateNextDay(actualDate);
   }
   if (actualDate.equals(date1)){
      HSSFRow row = sheet.createRow((short) i);
      row.createCell((short) 0).setCellValue(rs.getString("datum"));
      row.createCell((short) 1).setCellValue(rs.getString("work_place"));
      row.createCell((short) 2).setCellValue(rs.getString("type_of_clothes"));
      row.createCell((short) 3).setCellValue(rs.getString("type_of_activity"));
      row.createCell((short) 4).setCellValue(rs.getString("thermal_comfort"));
      row.createCell((short) 5).setCellValue(rs.getString("window"));
      row.createCell((short) 6).setCellValue(rs.getString("door"));
      row.createCell((short) 7).setCellValue(rs.getString("air_conditioning"));
      row.createCell((short) 8).setCellValue(rs.getString("heating"));
      row.createCell((short) 9).setCellValue(rs.getString("tolerance"));
      row.createCell((short) 10).setCellValue(rs.getString("pmv"));
      row.createCell((short) 11).setCellValue(rs.getString("ppd"));
      row.createCell((short) 12).setCellValue(rs.getString("temperature"));
      i++;
   }
}

FileOutputStream fileOut = new FileOutputStream(excelurl);
workbook.write(fileOut);
```

```
            fileOut.close();
        } catch (ClassNotFoundException | SQLException | FileNotFoundException e1) {
            System.out.println(e1);
        }  catch (IOException e) {
            System.out.println(e);
        }
    }

    private String calculateNextDay(String date){
        int day = Integer.parseInt(date.substring(8));
        int month = Integer.parseInt(date.substring(5, 7));
        int year = Integer.parseInt(date.substring(0, 4));
        if (day < 28){
            day = day + 01;
        }
        else if (day == 28 && month == 02){
            day = 01;
            month = 03;
        }
        else if (day == 29){
            day = day + 01;
        }
        else if (day == 30){
            switch(month){
                case 4: case 6: case 9: case 11:
                    day = 01;
                    month = month + 01;
                    break;
                case 1: case 3: case 5: case 7: case 8: case 10: case 12:
                    day = day + 01;
                    break;
            }
        }
        else if (day == 31){
```

```
            switch(month){
                case 1: case 3: case 5: case 7: case 8: case 10:
                    day = 01;
                    month = month + 01;
                    break;
                case 12:
                    day = 01;
                    month = 01;
                    year ++;
                    break;
            }
        }
        String dai = Integer.toString(day);
        String mont = Integer.toString(month);
        if (dai.length() < 2){
            dai = "0" + dai;
        }
        if (mont.length() < 2){
            mont = "0" + mont;
        }
        String finaldate = year + "-" + mont + "-" + dai;
        return finaldate;
    }
}
```

//Window from 3rd application

```
public class SelectData extends javax.swing.JFrame {

    public SelectData() {
        initComponents();
    }
    private DatabaseToExcel datatoExcel;
    int a = 0;
    int b = 0;
```

```
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    selectDate = new javax.swing.JLabel();
    writeDate = new javax.swing.JTextField();
    writeDate2 = new javax.swing.JTextField();
    selectDate2 = new javax.swing.JLabel();
    close = new javax.swing.JButton();
    export = new javax.swing.JButton();
    jLabel2 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    selectDate.setText("Select first date (yyyy-mm-dd):");

    writeDate.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            writeDateActionPerformed(evt);
        }
    });

    writeDate2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            writeDate2ActionPerformed(evt);
        }
    });

    selectDate2.setText("Select final date (yyyy-mm-dd):");

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
```

```
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                .addComponent(selectDate)
                .addComponent(selectDate2))
            .addGap(42, 42, 42)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING, false)
                .addComponent(writeDate, javax.swing.GroupLayout.DEFAULT_SIZE, 211,
Short.MAX_VALUE)
                .addComponent(writeDate2))
                                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
                .addComponent(selectDate)
                .addComponent(writeDate, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
                .addComponent(writeDate2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(selectDate2))
                                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
```

```
close.setText("Close");
close.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        closeActionPerformed(evt);
    }
});

export.setText("Export");
export.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        exportActionPerformed(evt);
    }
});

jLabel2.setText("<html>Remember that once you select export, all the data that was previously in the file will <BR> be erased and substitute by the new data.</html>");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addGroup(layout.createSequentialGroup()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(layout.createSequentialGroup()
                .addGap(106, 106, 106)
```

```
                    .addComponent(export, javax.swing.GroupLayout.PREFERRED_SIZE,
87, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(50, 50, 50)
                .addComponent(close, javax.swing.GroupLayout.PREFERRED_SIZE, 85,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                    .addComponent(export, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(close, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void writeDateActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void writeDate2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
```

```
}

private void exportActionPerformed(java.awt.event.ActionEvent evt) {
    String writedate1 = writeDate.getText();
    String writedate2 = writeDate2.getText();
    if (writedate1 != null && writedate1.trim().length() > 0){
        a = 1;
    }
    if (writedate2 != null && writedate2.trim().length() > 0){
        a = 0;
        b = 1;
    }
    try {
        datatoExcel = new DatabaseToExcel();
        if (a == 0 && b == 0){
            datatoExcel.exportAll();
        }
        else if (a == 1){
            datatoExcel.exportOneDay(writedate1);
        }
        else if (b == 1){
            datatoExcel.exportPeriod(writedate1, writedate2);
        }
    }catch (Exception ex){
        System.out.println(ex);
    }
}

private void closeActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}

public static void main(String args[]) {
```

```
        try {
                                        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(SelectData.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(SelectData.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(SelectData.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(SelectData.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        }
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new SelectData().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton close;
    private javax.swing.JButton export;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JLabel selectDate;
    private javax.swing.JLabel selectDate2;
    private javax.swing.JTextField writeDate;
    private javax.swing.JTextField writeDate2;
```

```
    // End of variables declaration
}
```

//config.properties from 3rd application

```
user=root
password=password
url=jdbc:mysql://localhost:3306/thermal_comfort
driver=com.mysql.jdbc.Driver
excelurl=C:\\Users\\ANTONI1\\Desktop\\ANTONI\\thermalComfortData.xls
```

Annexe D:

//Scripts to create the tables of the database:

//user table:

```
CREATE TABLE user (
   id INT UNSIGNED NOT NULL AUTO_INCREMENT,
   office VARCHAR (100),
   user_name VARCHAR (100),
   PRIMARY KEY (id)
        );
```

//datauser table:

```
CREATE TABLE Datauser (
   id INT UNSIGNED NOT NULL AUTO_INCREMENT,
   datum DATE NOT NULL,
   work_place VARCHAR (100),
   type_of_clothes DOUBLE,
   type_of_activity DOUBLE,
   thermal_comfort VARCHAR (200),
   window VARCHAR (50),
   door VARCHAR (50),
   air_conditioning VARCHAR (50),
   heating VARCHAR (50),
   tolerance VARCHAR (50),
   PRIMARY KEY (id)
        );
```

//thermal table:

```
CREATE TABLE thermal (
   id INT UNSIGNED NOT NULL AUTO_INCREMENT,
   datum DATE NOT NULL,
   work_place VARCHAR (100),
   type_of_clothes DOUBLE,
   type_of_activity DOUBLE,
   thermal_comfort VARCHAR (100),
   window VARCHAR (50),
   door VARCHAR (50),
   air_conditioning VARCHAR (50),
   heating VARCHAR (50),
   tolerance VARCHAR (50),
   pmv DOUBLE,
   ppd DOUBLE,
   temperature DOUBLE,
   relative_humidity DOUBLE,
   PRIMARY KEY (id)
        );
```