#AboutDialog.java

```
package Frame;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.GroupLayout;
import javax.swing.JButton;
import javax.swing.JComponent;

import javax.swing.JFrame;
import javax.swing.JTextPane;
import javax.swing.LayoutStyle;
import javax.swing.SwingUtilities;


/**
* This code was edited or generated using CloudGarden's Jigloo
* SWT/Swing GUI Builder, which is free for non-commercial
* use. If Jigloo is being used commercially (ie, by a corporation,
* company or business for any purpose whatever) then you
* should purchase a license for each developer using Jigloo.
* Please visit www.cloudgarden.com for details.
* Use of Jigloo implies acceptance of these licensing terms.
* A COMMERCIAL LICENSE HAS NOT BEEN PURCHASED FOR
* THIS MACHINE, SO JIGLOO OR THIS CODE CANNOT BE USED
* LEGALLY FOR ANY CORPORATE OR COMMERCIAL PURPOSE.
*/
public class AboutDialog extends javax.swing.JDialog {
	/**
	 * @uml.property  name="aboutDialogOK"
	 * @uml.associationEnd  multiplicity="(1 1)"
	 */
	private JButton AboutDialogOK;
	/**
	 * @uml.property  name="aboutText"
	 * @uml.associationEnd  multiplicity="(1 1)"
	 */
	private JTextPane AboutText;

	/**
	* Auto-generated main method to display this JDialog
	*/

	public AboutDialog(JFrame frame) {
		super(frame);
		initGUI();
	}

	private void initGUI() {
		try {
			GroupLayout thisLayout = new GroupLayout((JComponent)getContentPane());
			getContentPane().setLayout(thisLayout);
			getContentPane().setBackground(new java.awt.Color(255,255,255));
			this.setTitle("About this program..");
			/*
			 * This program is journal recommendation system.
```

It is connected to KAIST's student DB so that it can access users' personal data once you logged in by their student ID.

Based on users' academic data along with their answers on a series of qeustionnaires, it can provide them with recommendations for the most suitable journals for them to submit.
This will improve users' ability to choose right journals as well as minimize the cost of finding them.

```
                 *
                 * */
                this.setFont(new java.awt.Font("Consolas",0,12));
                {
                    AboutDialogOK = new JButton();
                    AboutDialogOK.setText("OK");
                    AboutDialogOK.addMouseListener(new MouseAdapter() {
                        public void mouseClicked(MouseEvent evt) {
                            AboutDialogOKMouseClicked(evt);
                        }
                    });
                }
                {
                    AboutText = new JTextPane();
                    AboutText.setText("This program is journal recommendation system. " +
                            "It is connected to KAIST's student DB so that it can " +
                            "access users' personal data once you logged in by their student ID. Based on users'
academic data along with their answers on a series of questionnaires, it can provide them with recommendations for
the most suitable journals for them to submit. This will improve users' ability to choose right journals as well as
minimize the cost of finding them.");
                    AboutText.setEditable(false);
                    AboutText.setFont(new java.awt.Font("Consolas",0,12));
                }
                thisLayout.setVerticalGroup(thisLayout.createSequentialGroup()
                    .addContainerGap(24, 24)
                    .addComponent(AboutText, GroupLayout.PREFERRED_SIZE, 180,
GroupLayout.PREFERRED_SIZE)
                    .addGap(17)
                    .addComponent(AboutDialogOK, GroupLayout.PREFERRED_SIZE,
GroupLayout.PREFERRED_SIZE, GroupLayout.PREFERRED_SIZE)
                    .addContainerGap(17, Short.MAX_VALUE));
                thisLayout.setHorizontalGroup(thisLayout.createSequentialGroup()
                    .addContainerGap(24, 24)
                    .addGroup(thisLayout.createParallelGroup()
                       .addGroup(thisLayout.createSequentialGroup()
                          .addComponent(AboutText, GroupLayout.PREFERRED_SIZE, 317,
GroupLayout.PREFERRED_SIZE)
                          .addGap(0, 0, Short.MAX_VALUE))
                       .addGroup(GroupLayout.Alignment.LEADING, thisLayout.createSequentialGroup()
                          .addGap(137)
                          .addComponent(AboutDialogOK, GroupLayout.PREFERRED_SIZE, 60,
GroupLayout.PREFERRED_SIZE)
                          .addGap(0, 120, Short.MAX_VALUE)))
                    .addContainerGap(43, 43));
                setSize(400, 300);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

        private void AboutDialogOKMouseClicked(MouseEvent evt) {
            this.dispose();
            //TODO add your code for AboutDialogOK.mouseClicked
        }
```

}

```
//AnalyzerUtils.java
package Lucene;

import java.io.IOException;
import java.io.StringReader;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

import org.apache.lucene.analysis.Analyzer;
import org.apache.lucene.analysis.SimpleAnalyzer;
import org.apache.lucene.analysis.TokenStream;
import org.apache.lucene.analysis.standard.StandardAnalyzer;
import org.apache.lucene.analysis.tokenattributes.OffsetAttribute;
import org.apache.lucene.analysis.tokenattributes.TermAttribute;
import org.apache.lucene.analysis.tokenattributes.TypeAttribute;
import org.apache.lucene.util.AttributeSource;
import org.apache.lucene.util.Version;

/**
 * Class provided as source for 'Lucene In Action, 2nd Edition'
 */
public class AnalyzerUtils {
	public static void displayTokens(Analyzer analyzer,
			String text) throws IOException {
		displayTokens(analyzer.tokenStream("contents", new StringReader(text)));  //A
	}

	public static void displayTokens(TokenStream stream)
	throws IOException {

		TermAttribute term = (TermAttribute) stream.addAttribute(TermAttribute.class);
		while(stream.incrementToken()) {
			System.out.print("[" + term.term() + "] ");    //B
		}
		System.out.println();
	}



	public static List<String> getAnalyzedTokens(Analyzer analyzer, String text, boolean sortedAlphabetically)
throws IOException
	{
		return getAnalyzedTokens(analyzer.tokenStream("contents", new StringReader(text)),
sortedAlphabetically);
	}

	public static List<String> getAnalyzedTokens(Analyzer analyzer, String text) throws IOException {
		return getAnalyzedTokens(analyzer.tokenStream("contents", new StringReader(text)), false);  //A
	}

	public static List<String> getAnalyzedTokens(TokenStream stream) throws IOException
	{
		return getAnalyzedTokens(stream, false);
	}
	public static List<String> getAnalyzedTokens(TokenStream stream, boolean sortedAlphabetically) throws
IOException {
```

```
        List<String> result = new ArrayList<String>();
        TermAttribute term = (TermAttribute) stream.addAttribute(TermAttribute.class);
        while(stream.incrementToken())
        {
            result.add(term.term());
        }

        if(sortedAlphabetically)
        {
            Collections.sort(result);
        }

        return result;
    }

    public static String getTerm(AttributeSource source) {
        TermAttribute attr = (TermAttribute) source.addAttribute(TermAttribute.class);
        return attr.term();
    }

    public static String getType(AttributeSource source) {
        TypeAttribute attr = (TypeAttribute) source.addAttribute(TypeAttribute.class);
        return attr.type();
    }
    public static void setTerm(AttributeSource source, String term) {
        TermAttribute attr = (TermAttribute) source.addAttribute(TermAttribute.class);
        attr.setTermBuffer(term);
    }

    public static void setType(AttributeSource source, String type) {
        TypeAttribute attr = (TypeAttribute) source.addAttribute(TypeAttribute.class);
        attr.setType(type);
    }

    public static void displayTokensWithPositions
    (Analyzer analyzer, String text) throws IOException {

        TokenStream stream = analyzer.tokenStream("contents",
                    new StringReader(text));
        TermAttribute term = (TermAttribute) stream.addAttribute(TermAttribute.class);

        int position = 0;
        while(stream.incrementToken()) {
            System.out.print("[" + term.term() + "] ");
        }
        System.out.println();
    }

    public static void displayTokensWithFullDetails(Analyzer analyzer,
            String text) throws IOException {

        TokenStream stream = analyzer.tokenStream("contents",                    // #A
                    new StringReader(text));

        TermAttribute term = (TermAttribute)                          // #B
        stream.addAttribute(TermAttribute.class);                     // #B
        OffsetAttribute offset = (OffsetAttribute)                    // #B
```

```
        stream.addAttribute(OffsetAttribute.class);                 // #B
        TypeAttribute type = (TypeAttribute)                        // #B
        stream.addAttribute(TypeAttribute.class);                   // #B

        int position = 0;
        while(stream.incrementToken()) {                            // #C
            System.out.print("[" +                                  // #E
                    term.term() + ":" +                             // #E
                    offset.startOffset() + "->" +                   // #E
                    offset.endOffset() + ":" +                      // #E
                    type.type() + "] ");                            // #E
        }
        System.out.println();
    }
}
```

```
//databaseManager.java
package Database;

import java.io.BufferedWriter;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.Writer;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;

import com.mysql.jdbc.Blob;

public class databaseManager {
 rivate static Connection con = null;
 rivate static java.sql.Statement st;
 rivate static ResultSet rs;
	public static void connectionDatabase(){
 try {
	 con = DriverManager.getConnection("jdbc:mysql://110.76.81.24/kse643",
			"jhkim", "1234!@");
	st = con.createStatement();

		catch (SQLException sqex) {
	System.out.println("SQLException: " + sqex.getMessage());
	System.out.println("SQLState: " + sqex.getSQLState());
 }
	}

 ublic int getNumOfRow() throws SQLException{
 int totalNum = 0;
 rs = st.executeQuery("SELECT count(*) FROM journalknowledgebase;");
 if (st.execute("SELECT count(*) FROM journalknowledgebase;")){
	rs = st.getResultSet();
 }
		while(rs.next()){
	totalNum = rs.getInt(1);
 }
		return totalNum;



	public static int getNumOfJournal() throws SQLException{
 int totalNum = 0;
 rs = st.executeQuery("SELECT count(*) FROM journal_image;");
 if (st.execute("SELECT count(*) FROM journal_image;")){
	rs = st.getResultSet();
 }
		while(rs.next()){
	totalNum = rs.getInt(1);
 }
		return totalNum;


	public static byte[] readImage(String stdNum) throws SQLException{
```

```
try {
	 rs = st.executeQuery("SELECT image FROM student_image where id = "'+stdNum+"';");
} catch (SQLException e) {
	// TODO Auto-generated catch block
	e.printStackTrace();
}
		try {
	if (st.execute("SELECT image FROM student_image where id = "'+stdNum+"';")){
		rs = st.getResultSet();
	}
		} catch (SQLException e) {
	// TODO Auto-generated catch block
	e.printStackTrace();
}
		//System.out.println("Here : "+stdNum);
Blob blob;
while (rs.next()) {
	 try {
		 blob = (Blob) rs.getBlob(1);
		 int blobLength = (int) blob.length();
	  byte[] blobAsBytes = blob.getBytes(1, blobLength);
	  return blobAsBytes;
	} catch (SQLException e) {
		// TODO Auto-generated catch block
		e.printStackTrace();
	}
		}
return null;


	public static byte[] readJournalImage(String isbn) throws SQLException{
try {
	 rs = st.executeQuery("SELECT journal_imagecol FROM kse643.journal_image where id = "'+isbn+"';");
} catch (SQLException e) {
	// TODO Auto-generated catch block
	e.printStackTrace();
}
		try {
	if (st.execute("SELECT journal_imagecol FROM kse643.journal_image where id = "'+isbn+"';")){
		rs = st.getResultSet();
	}
		} catch (SQLException e) {
	// TODO Auto-generated catch block
	e.printStackTrace();
}
		//System.out.println("Here : "+isbn);
Blob blob;
while (rs.next()) {
	 try {
		 blob = (Blob) rs.getBlob(1);
		 int blobLength = (int) blob.length();
	  byte[] blobAsBytes = blob.getBytes(1, blobLength);
	  return blobAsBytes;
	} catch (SQLException e) {
		// TODO Auto-generated catch block
		e.printStackTrace();
	}
```

```
        }
return null;


    public static  boolean  initialSetup() throws SQLException{
StringBuffer buffer = new StringBuffer();

Writer writer = null;

        try {
    writer = new BufferedWriter(new OutputStreamWriter(
          new FileOutputStream("knowledgebase.pl"), "utf-8"));

            String inputSQL = "SELECT knowledgeInfo FROM journalknowledgebase;";

            rs = st.executeQuery(inputSQL);
    if (st.execute(inputSQL)){
        rs = st.getResultSet();
    }
            while(rs.next()){
        //System.out.println(rs.getString(1));
        //buffer.append(rs.getString(1)+"\n");
                writer.write(rs.getString(1)+"\n");
            }



            inputSQL = "SELECT knowledgeInfo FROM userknowledgebase;";

            rs = st.executeQuery(inputSQL);
            if (st.execute(inputSQL)){
                rs = st.getResultSet();
            }
            while(rs.next()){

                //System.out.println(rs.getString(1));
                writer.write(rs.getString(1)+"\n");
                //buffer.append(rs.getString(1)+"\n");
            }


            inputSQL = "SELECT weight FROM journal_weight;";

            rs = st.executeQuery(inputSQL);
            if (st.execute(inputSQL)){
                rs = st.getResultSet();
            }
            while(rs.next()){

                //System.out.println(rs.getString(1));
                writer.write(rs.getString(1)+"\n");
                //buffer.append(rs.getString(1)+"\n");
            }
```

```
            inputSQL = "SELECT knowledgebase FROM rulebase1;";

            rs = st.executeQuery(inputSQL);
            if (st.execute(inputSQL)){
                rs = st.getResultSet();
            }
            while(rs.next()){

                //System.out.println(rs.getString(1));
                writer.write(rs.getString(1)+"\n");
                //buffer.append(rs.getString(1)+"\n");
            }


        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            try {
                if (writer != null)writer.close();
            } catch (IOException ex) {
                ex.printStackTrace();
            }
        }
        //load all data from DB
        //concat
        //store to knowledgeBase.pl

        return false;
    }
}
```

```
//EnglishSpeaker.java
package EnglishSpeech;

import java.beans.PropertyVetoException;
import java.util.Locale;
import javax.speech.AudioException;
import javax.speech.Central;
import javax.speech.EngineException;
import javax.speech.EngineStateError;
import javax.speech.synthesis.Synthesizer;
import javax.speech.synthesis.SynthesizerModeDesc;
import javax.speech.synthesis.Voice;
public class EnglishSpeaker {
 SynthesizerModeDesc desc;
 Synthesizer synthesizer;
 Voice voice;
 public void init(String voiceName)
     throws EngineException, AudioException, EngineStateError,
     PropertyVetoException {
  if (desc == null) {
   System.setProperty("freetts.voices",
       "com.sun.speech.freetts.en.us.cmu_us_kal.KevinVoiceDirectory");
   desc = new SynthesizerModeDesc(Locale.US);
   Central.registerEngineCentral("com.sun.speech.freetts.jsapi.FreeTTSEngineCentral");
   synthesizer = Central.createSynthesizer(desc);
   synthesizer.allocate();
   synthesizer.resume();
   SynthesizerModeDesc smd =
       (SynthesizerModeDesc) synthesizer.getEngineModeDesc();
   Voice[] voices = smd.getVoices();
   Voice voice = null;
   for (int i = 0; i < voices.length; i++) {
    if (voices[i].getName().equals(voiceName)) {
     voice = voices[i];
     break;
    }
   }
   synthesizer.getSynthesizerProperties().setVoice(voice);
  }
 }
 public void terminate() throws EngineException, EngineStateError {
  synthesizer.deallocate();
 }
 public void doSpeak(String speakText)
     throws EngineException, AudioException, IllegalArgumentException,
     InterruptedException {
  synthesizer.speakPlainText(speakText, null);
  synthesizer.waitEngineState(Synthesizer.QUEUE_EMPTY);
 }


 public static void main(String[] args) throws Exception {
  EnglishSpeaker su = new EnglishSpeaker();
  su.init("kevin16");
  su.doSpeak("selected journal is Alpha!");
  su.terminate();
 }
```

}

```
//GetInfo.java


package Frame;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JOptionPane;
import javax.swing.JRadioButton;
import javax.swing.JTextPane;

import javax.swing.WindowConstants;


//import info.clearthought.layout.TableLayout;
import java.awt.BorderLayout;
import java.awt.Canvas;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.Graphics;
import java.awt.GridLayout;
import java.awt.Image;
import java.awt.TextArea;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.IOException;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Hashtable;

import javax.imageio.ImageIO;
import javax.swing.*;

import Database.databaseManager;
import Database.prologManager;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

/**
 * This code was edited or generated using CloudGarden's Jigloo SWT/Swing GUI Builder, which is free for non-commercial use. If Jigloo is being used commercially (ie, by a corporation, company or business for any purpose whatever) then you should purchase a license for each developer using Jigloo. Please visit www.cloudgarden.com for details. Use of Jigloo implies acceptance of these licensing terms. A COMMERCIAL LICENSE HAS NOT BEEN PURCHASED FOR THIS MACHINE, SO JIGLOO OR THIS CODE CANNOT BE USED LEGALLY FOR ANY CORPORATE OR COMMERCIAL PURPOSE.
 * @uml.dependency   supplier="Lucene.LuceneAnalyzer"
 * @uml.dependency   supplier="Database.prologManager"
 */
```

```
public class GetInfo extends JFrame {
	private JButton tab_1_next;
 rivate JButton tab_2_back;
 rivate JButton tab_2_next;
 rivate JButton tab_3_back;
 rivate JButton tab_3_next;
 rivate JButton tab_4_back;
 private JButton tab_4_next;
	private JButton tab_5_back;
	private JButton tab_5_next;
	private JButton tab_6_back;
	private JButton tab_6_finish;
	private JRadioButton q1RadioButton1;
	private JRadioButton q1RadioButton2;
	private JRadioButton q2RadioButton1;
	private JRadioButton q2RadioButton2;
	private JRadioButton q2RadioButton3;
   private JTextField q3TextField;
   private JComboBox q4ComboBox;
	private JRadioButton q5RadioButton1;
	private JRadioButton q5RadioButton2;
   private JTextField q6TextField;
   private TextArea q6TextArea;
   private String stdNum;
   private String recommendedList[] = new String[5];
   private Image img;

   private Hashtable<String, String> explainList;

   private RecommendationWindow rw;
   private JTabbedPane jtp = new JTabbedPane();



	GetInfo(){
		UserInterface();
		initEventListener();
	}

	GetInfo(String t){
		stdNum = t;
		explainList = new Hashtable<String, String>();
		UserInterface();
		initEventListener();
	}

	private void UserInterface(){


	//Here we are creating the object
	//JTabbedPane jtp = new JTabbedPane();
	//This creates the template on the windowed application that we will be using
	getContentPane().add(jtp);


	JPanel tab_1= new JPanel();
	JPanel tab_2 = new JPanel();
	JPanel tab_3 = new JPanel();
	JPanel tab_4 = new JPanel();
	JPanel tab_5 = new JPanel();
	JPanel tab_6 = new JPanel();
```

```
//This creates a non-editable label, sets what the label will read
//and adds the label to the first tab

jtp.addTab("Question1", tab_1);
jtp.addTab("Question2", tab_2);
jtp.addTab("Question3", tab_3);
jtp.addTab("Question4", tab_4);
jtp.addTab("Question5", tab_5);
jtp.addTab("Question6", tab_6);

Toolkit tk = Toolkit.getDefaultToolkit();
img = tk.getImage("survey4.png");
Canvas cvs = new Canvas(){
@Override
public void paint(Graphics g)     {
   g.drawImage(img, 0, 0, this);
}
};
cvs.setSize(1000,300);
cvs.repaint();



tab_1.setLayout(new BorderLayout());


tab_1_next = new JButton("NEXT >");
q1RadioButton1 = new JRadioButton();
q1RadioButton1.setText("YES");
q1RadioButton2 = new JRadioButton();
q1RadioButton2.setText("NO");
ButtonGroup group1 =new ButtonGroup();
group1.add(q1RadioButton1);
group1.add(q1RadioButton2);

JLabel RadioButtonGroup1 = new JLabel();
RadioButtonGroup1.setLayout(new FlowLayout(FlowLayout.CENTER, 10, 15));
RadioButtonGroup1.add(new JLabel("Select :  "));
RadioButtonGroup1.add(q1RadioButton1);
RadioButtonGroup1.add(q1RadioButton2);


JLabel tab_1_center_text1 = new JLabel();
   JLabel tab_1_center_text2 = new JLabel();
tab_1_center_text1.setFont(new java.awt.Font("arial",0,22));
   tab_1_center_text2.setFont(new java.awt.Font("arial",0,22));
tab_1_center_text1.setText("                    Is new paper related to the recent paper you
submitted?");
tab_1_center_text2.setText("                    ( In case you have never submitted, please select
No. )");



JPanel tab_1_inside_panel = new JPanel();
tab_1_inside_panel.setLayout(new GridLayout(5,1));
tab_1_inside_panel.setBackground(Color.white);
tab_1_inside_panel.add(new JLabel("                         Question 1:"));
tab_1_inside_panel.add(tab_1_center_text1);
tab_1_inside_panel.add(tab_1_center_text2);
```

```
        tab_1_inside_panel.add(RadioButtonGroup1);

    q1RadioButton1.setBackground(Color.white);
    q1RadioButton2.setBackground(Color.white);

    q1RadioButton1.setSelected(true);//
    JPanel tab_1_bottomPanel = new JPanel();//
    tab_1_bottomPanel.setLayout(new BorderLayout());
    tab_1_bottomPanel.add(tab_1_next, BorderLayout.EAST);

    JPanel tab_1_in = new JPanel();//tab_1
    tab_1_in.setLayout(new BorderLayout());
    //tab_1_in.add(Profile, BorderLayout.NORTH);
    tab_1_in.add(tab_1_inside_panel, BorderLayout.CENTER);
    tab_1_in.add(tab_1_bottomPanel, BorderLayout.SOUTH);
    tab_1_in.setBackground(Color.WHITE);
    tab_1.add(tab_1_in, BorderLayout.CENTER);


    //add(jtp, BorderLayout.NORTH);
    //add(bottomPanel, BorderLayout.SOUTH);


    //================================================ tab_2
==================================================//

    tab_2.setLayout(new BorderLayout());


    //----------------------tab_2_in  ---------------------//
    tab_2_next = new JButton("NEXT >");//
    tab_2_back = new JButton("< BACK");
    q2RadioButton1 = new JRadioButton();
    q2RadioButton1.setText("management");
    q2RadioButton2 = new JRadioButton();
    q2RadioButton2.setText("computer_science");
    q2RadioButton3 = new JRadioButton();
    q2RadioButton3.setText("economics");

    ButtonGroup group2 =new ButtonGroup();//
        group2.add(q2RadioButton1);
    group2.add(q2RadioButton2);
    group2.add(q2RadioButton3);

    q2RadioButton1.setBackground(Color.white);// Backgrounds Color
    q2RadioButton2.setBackground(Color.white);
    q2RadioButton3.setBackground(Color.white);

    JLabel ButtonGroup2 = new JLabel();
    ButtonGroup2.setLayout(new FlowLayout(FlowLayout.CENTER, 10, 15));
    ButtonGroup2.add(new JLabel("Select :  "));
    ButtonGroup2.add(q2RadioButton1);
    ButtonGroup2.add(q2RadioButton2);
    ButtonGroup2.add(q2RadioButton3);

    JLabel tab_2_center_text1 = new JLabel();//??       JLabel tab_2_center_text2 = new JLabel();
    tab_2_center_text1.setFont(new java.awt.Font("arial",0,22));//  ??- ??.... ?
tab_2_center_text2.setFont(new java.awt.Font("arial",0,22));
    tab_2_center_text1.setText("                    What's the research area related to topic of the
paper?");
    tab_2_center_text2.setText("                    ( ex. Management, Computer Science, Economics
```

```
)");


        JPanel tab_2_inside_panel = new JPanel();
        tab_2_inside_panel.setLayout(new GridLayout(5,1));
        tab_2_inside_panel.setBackground(Color.white);
        tab_2_inside_panel.add(new JLabel("                              Question 2:"));
        tab_2_inside_panel.add(tab_2_center_text1);
        tab_2_inside_panel.add(tab_2_center_text2);
            tab_2_inside_panel.add(ButtonGroup2);

        q2RadioButton1.setSelected(true);//1 ??
        JPanel q2_bottomPanel = new JPanel();//
        q2_bottomPanel.setLayout(new BorderLayout());
        q2_bottomPanel.add(tab_2_back, BorderLayout.WEST);
        q2_bottomPanel.add(tab_2_next, BorderLayout.EAST);

        JPanel tab_2_in = new JPanel();//tab_2
        tab_2_in.setLayout(new BorderLayout());
        //tab_2_in.add(Profile, BorderLayout.NORTH);
        tab_2_in.add(tab_2_inside_panel, BorderLayout.CENTER);
        tab_2_in.add(q2_bottomPanel, BorderLayout.SOUTH);
        tab_2_in.setBackground(Color.WHITE); //��
        tab_2.add(tab_2_in, BorderLayout.CENTER);//



        //================================================ tab_3
==================================================//

        tab_3.setLayout(new BorderLayout()); //tab_3


        //----------------------tab_3_in  ----------------------//
        tab_3_next = new JButton("NEXT >");//
        tab_3_back = new JButton("< BACK");

        //ComboBoxModel q3ComboBoxModel = new DefaultComboBoxModel(new String[] { "Knowledge
Engineeering", "HCI", "Database", "User Expierience" });
        //q3ComboBox = new JComboBox();
        //q3ComboBox.setModel(q3ComboBoxModel);
            q3TextField = new JTextField();
            q3TextField.setText("");
            //q3ComboBox.setBackground(Color.gray);


        JLabel ButtonGroup3 = new JLabel();
        ButtonGroup3.setLayout(new FlowLayout(FlowLayout.CENTER, 10, 15));
        ButtonGroup3.add(new JLabel("Type your keyword :  "));
        //ButtonGroup3.add(q3ComboBox);
        ButtonGroup3.add(q3TextField);
        q3TextField.setPreferredSize(new java.awt.Dimension(250, 24));
        //q3ComboBox.setPreferredSize(new java.awt.Dimension(353, 24));
        //q3ComboBox.setOpaque(true);
        //q3ComboBox.setHorizontalAlignment(CENTER);

        JLabel tab_3_center_text1 = new JLabel();//??       JLabel tab_3_center_text2 = new JLabel();
        tab_3_center_text1.setFont(new java.awt.Font("arial",0,22));//  ??- ??.... ?
tab_3_center_text2.setFont(new java.awt.Font("arial",0,22));
        tab_3_center_text1.setText("                              What is the first important keyword of your
```

```
paper?");
      tab_3_center_text2.setText("                          ");


      JPanel tab_3_inside_panel = new JPanel();
      tab_3_inside_panel.setLayout(new GridLayout(5,1));
      tab_3_inside_panel.setBackground(Color.white);
      tab_3_inside_panel.add(new JLabel("                          Question 3:"));
      tab_3_inside_panel.add(tab_3_center_text1);
      tab_3_inside_panel.add(tab_3_center_text2);
         tab_3_inside_panel.add(ButtonGroup3);


      //q3RadioButton1.setSelected(true);//1 ??
      JPanel q3_bottomPanel = new JPanel();//
      q3_bottomPanel.setLayout(new BorderLayout());
      q3_bottomPanel.add(tab_3_back, BorderLayout.WEST);
      q3_bottomPanel.add(tab_3_next, BorderLayout.EAST);

      JPanel tab_3_in = new JPanel();//tab_3
      tab_3_in.setLayout(new BorderLayout());
      //tab_3_in.add(Profile, BorderLayout.NORTH);
      tab_3_in.add(tab_3_inside_panel, BorderLayout.CENTER);
      tab_3_in.add(q3_bottomPanel, BorderLayout.SOUTH);
      tab_3_in.setBackground(Color.WHITE); //��
      tab_3.add(tab_3_in, BorderLayout.CENTER);//


      //================================================ tab_4
================================================//

      tab_4.setLayout(new BorderLayout()); //tab_4

      //----------------------tab_4_in  ----------------------//
      tab_4_next = new JButton("NEXT >");//
      tab_4_back = new JButton("< BACK");

      ComboBoxModel q4ComboBoxModel = new DefaultComboBoxModel(new String[] {
"article_influence_score", "cited_half-life", "five-year_impact_factor", "immediacy_index", "total_cites",
"none" });
      q4ComboBox = new JComboBox();
         q4ComboBox.setModel(q4ComboBoxModel);
         //q4ComboBox.setBackground(Color.gray);


      JLabel ButtonGroup4 = new JLabel();
      ButtonGroup4.setLayout(new FlowLayout(FlowLayout.CENTER, 10, 15));
      ButtonGroup4.add(new JLabel("Select :  "));
      ButtonGroup4.add(q4ComboBox);


      JLabel tab_4_center_text1 = new JLabel();//??       JLabel tab_4_center_text2 = new JLabel();
      tab_4_center_text1.setFont(new java.awt.Font("arial",0,22));//  ??- ??.... ?
tab_4_center_text2.setFont(new java.awt.Font("arial",0,22));
      tab_4_center_text1.setText("                          Besides Impact Factor, is there a Measurement you
specially consider?");
      tab_4_center_text2.setText("                          ");


      JPanel tab_4_inside_panel = new JPanel();
      tab_4_inside_panel.setLayout(new GridLayout(5,1));
```

```
    tab_4_inside_panel.setBackground(Color.white);
    tab_4_inside_panel.add(new JLabel("                    Question 4:"));
    tab_4_inside_panel.add(tab_4_center_text1);
    tab_4_inside_panel.add(tab_4_center_text2);
       tab_4_inside_panel.add(ButtonGroup4);



    //q4RadioButton1.setSelected(true);//1 ??
    JPanel q4_bottomPanel = new JPanel();//
    q4_bottomPanel.setLayout(new BorderLayout());
    q4_bottomPanel.add(tab_4_back, BorderLayout.WEST);
    q4_bottomPanel.add(tab_4_next, BorderLayout.EAST);

    JPanel tab_4_in = new JPanel();//tab_4
    tab_4_in.setLayout(new BorderLayout());
    //tab_4_in.add(Profile, BorderLayout.NORTH);
    tab_4_in.add(tab_4_inside_panel, BorderLayout.CENTER);
    tab_4_in.add(q4_bottomPanel, BorderLayout.SOUTH);
    tab_4_in.setBackground(Color.WHITE);
    tab_4.add(tab_4_in, BorderLayout.CENTER);//


    //================================================ tab_5
==================================================//

    tab_5.setLayout(new BorderLayout()); //tab_5

    //----------------------tab_5_in  ----------------------//
    tab_5_next = new JButton("NEXT >");//
    tab_5_back = new JButton("< BACK");

    q5RadioButton1 = new JRadioButton();
    q5RadioButton1.setText("YES");
    q5RadioButton2 = new JRadioButton();
    q5RadioButton2.setText("NO");
    ButtonGroup group5 =new ButtonGroup();
    group5.add(q5RadioButton1);
    group5.add(q5RadioButton2);

    JLabel ButtonGroup5 = new JLabel();
    ButtonGroup5.setLayout(new FlowLayout(FlowLayout.CENTER, 10, 15));
    ButtonGroup5.add(new JLabel("Select :  "));
    ButtonGroup5.add(q5RadioButton1);
    ButtonGroup5.add(q5RadioButton2);


    JLabel tab_5_center_text1 = new JLabel();
       JLabel tab_5_center_text2 = new JLabel();
    tab_5_center_text1.setFont(new java.awt.Font("arial",0,22));
       tab_5_center_text2.setFont(new java.awt.Font("arial",0,22));
    tab_5_center_text1.setText("                    Would you prefer a Journal that has long history?");
    tab_5_center_text2.setText("                    ");



    JPanel tab_5_inside_panel = new JPanel();
    tab_5_inside_panel.setLayout(new GridLayout(5,1));
    tab_5_inside_panel.setBackground(Color.white);
    tab_5_inside_panel.add(new JLabel("                    Question 5:"));
    tab_5_inside_panel.add(tab_5_center_text1);
```

```
    tab_5_inside_panel.add(tab_5_center_text2);
       tab_5_inside_panel.add(ButtonGroup5);

    q5RadioButton1.setBackground(Color.white);
    q5RadioButton2.setBackground(Color.white);

    q5RadioButton1.setSelected(true);//1 ??
    JPanel tab_5_bottomPanel = new JPanel();//
    tab_5_bottomPanel.setLayout(new BorderLayout());
    tab_5_bottomPanel.add(tab_5_back, BorderLayout.WEST);
    tab_5_bottomPanel.add(tab_5_next, BorderLayout.EAST);

    JPanel tab_5_in = new JPanel();//tab_5
    tab_5_in.setLayout(new BorderLayout());
    //tab_5_in.add(Profile, BorderLayout.NORTH);
    tab_5_in.add(tab_5_inside_panel, BorderLayout.CENTER);
    tab_5_in.add(tab_5_bottomPanel, BorderLayout.SOUTH);
    tab_5_in.setBackground(Color.WHITE);
    tab_5.add(tab_5_in, BorderLayout.CENTER);//


    //============================================== tab_6
==============================================//

    tab_6.setLayout(new BorderLayout()); //tab_6

    //----------------------tab_6_in  ----------------------//
    tab_6_back = new JButton("< BACK");//
    tab_6_finish = new JButton("FINISH");

    //ComboBoxModel q6ComboBoxModel = new DefaultComboBoxModel(new String[] { "Management
Information Systems Quartly", "Science", "Nature", "Total Cites", "None" });
    //q6ComboBox = new JComboBox();
       //q6ComboBox.setModel(q6ComboBoxModel);
       //q6ComboBox.setBackground(Color.gray);
       q6TextField = new JTextField();
       q6TextField.setText("");
       q6TextField.setPreferredSize(new java.awt.Dimension(500, 25));
       q6TextArea = new TextArea();
       q6TextArea.setText("");
       q6TextArea.setPreferredSize(new java.awt.Dimension(500, 50));

    JLabel Buttongroup6 = new JLabel();
    Buttongroup6.setLayout(new FlowLayout(FlowLayout.CENTER, 10, 15));
    Buttongroup6.add(new JLabel("Topic :       "));
    //Buttongroup6.add(q6ComboBox);
    Buttongroup6.add(q6TextField);

    JLabel Textgroup6 = new JLabel();
    Textgroup6.setLayout(new FlowLayout(FlowLayout.CENTER, 10, 15));
    Textgroup6.add(new JLabel("Abstract :  "));
    Textgroup6.add(q6TextArea);


    JLabel tab_6_center_text1 = new JLabel();//??       //JLabel tab_6_center_text2 = new JLabel();
    tab_6_center_text1.setFont(new java.awt.Font("arial",0,22));//
       //tab_6_center_text2.setFont(new java.awt.Font("arial",0,22));
    tab_6_center_text1.setText("                        Input your papers topic and abstract.");
    //tab_6_center_text2.setText("                          ( ex. Management Information System Quartly,
Science, Nature )");
```

```
    JPanel tab_6_inside_panel = new JPanel();
    tab_6_inside_panel.setLayout(new GridLayout(5,1));
    tab_6_inside_panel.setBackground(Color.white);
    tab_6_inside_panel.add(new JLabel("                    Question 6:"));
    tab_6_inside_panel.add(tab_6_center_text1);
    //tab_6_inside_panel.add(tab_6_center_text2);
        tab_6_inside_panel.add(Buttongroup6);
        tab_6_inside_panel.add(Textgroup6);


    //q6RadioButton1.setSelected(true);//1 ??
    JPanel q6_bottomPanel = new JPanel();//
    q6_bottomPanel.setLayout(new BorderLayout());
    q6_bottomPanel.add(tab_6_back, BorderLayout.WEST);
    q6_bottomPanel.add(tab_6_finish, BorderLayout.EAST);

    JPanel tab_6_in = new JPanel();//tab_6
    tab_6_in.setLayout(new BorderLayout());
    //tab_6_in.add(Profile, BorderLayout.NORTH);
    tab_6_in.add(tab_6_inside_panel, BorderLayout.CENTER);
    tab_6_in.add(q6_bottomPanel, BorderLayout.SOUTH);
    tab_6_in.setBackground(Color.WHITE);
    tab_6.add(tab_6_in, BorderLayout.CENTER);



    //ButtonHandler phandler = new ButtonHandler();
//tab_1_next.addActionListener((ActionListener) phandler);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    //JFrame main = new JFrame();
    setLayout(new BorderLayout());
    //setBackground(Color.WHITE);
    add(cvs, BorderLayout.NORTH);
    add(jtp, BorderLayout.CENTER);
    //add(Profile, BorderLayout.SOUTH);



    setTitle("Input your information!");
    setSize(1000,800);
        setVisible(true);


        Dimension frameSize = getSize();
        Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
        setLocation((screenSize.width - frameSize.width)/2, (screenSize.height - frameSize.height)/2);

  }




     private void initEventListener(){


            tab_1_next.addActionListener(new ActionListener(){
```

```
		@Override
		public void actionPerformed(ActionEvent arg0) {
			// TODO Auto-generated method stub

			if(!(q1RadioButton1.isSelected()||q1RadioButton2.isSelected())){
				JOptionPane.showConfirmDialog(null, "Please select one button!!", "error",
JOptionPane.CLOSED_OPTION);
			}else{
				jtp.setSelectedIndex(1);
			}
		}});

	tab_2_next.addActionListener(new ActionListener(){

		@Override
		public void actionPerformed(ActionEvent arg0) {
			// TODO Auto-generated method stub
			jtp.setSelectedIndex(2);
		}});

	tab_3_next.addActionListener(new ActionListener(){

		@Override
		public void actionPerformed(ActionEvent arg0) {
			// TODO Auto-generated method stub
			jtp.setSelectedIndex(3);
		}});

	tab_4_next.addActionListener(new ActionListener(){

		@Override
		public void actionPerformed(ActionEvent arg0) {
			// TODO Auto-generated method stub
			jtp.setSelectedIndex(4);
		}});

	tab_5_next.addActionListener(new ActionListener(){

		@Override
		public void actionPerformed(ActionEvent arg0) {
			// TODO Auto-generated method stub
			jtp.setSelectedIndex(5);
		}});

	tab_2_back.addActionListener(new ActionListener(){

		@Override
		public void actionPerformed(ActionEvent arg0) {
			// TODO Auto-generated method stub
			jtp.setSelectedIndex(0);
		}});

	tab_3_back.addActionListener(new ActionListener(){

		@Override
		public void actionPerformed(ActionEvent arg0) {
			// TODO Auto-generated method stub
			jtp.setSelectedIndex(1);
		}});

	tab_4_back.addActionListener(new ActionListener(){
```

```
		@Override
		public void actionPerformed(ActionEvent arg0) {
			// TODO Auto-generated method stub
			jtp.setSelectedIndex(2);
		}});

	tab_5_back.addActionListener(new ActionListener(){

		@Override
		public void actionPerformed(ActionEvent arg0) {
			// TODO Auto-generated method stub
			jtp.setSelectedIndex(3);
		}});

	tab_6_back.addActionListener(new ActionListener(){

		@Override
		public void actionPerformed(ActionEvent arg0) {
			// TODO Auto-generated method stub
			jtp.setSelectedIndex(4);
		}});

	tab_6_finish.addActionListener(new ActionListener(){

		@Override
		public void actionPerformed(ActionEvent arg0) {

			dispose();


			//prologManager.doRecommendation(stdNum, measureName, interesting,
userkeyword, recentJournalFlag, numOfRecentSubmitFlag, considerHistory, token)

			//q1 q2 q3 q4 q5 q6
			////////////////////////////////////////////////////////////////////////////////

			String measureName = ((String) q4ComboBox.getSelectedItem()).toLowerCase();
			System.out.println("measureName = "+measureName);

			////////////////////////////////////////////////////////////////////////////////
			String interesting = "";

			if(q2RadioButton1.isSelected()==true) interesting =
q2RadioButton1.getText().toLowerCase();
			if(q2RadioButton2.isSelected()==true) interesting =
q2RadioButton2.getText().toLowerCase();
			if(q2RadioButton3.isSelected()==true) interesting =
q2RadioButton3.getText().toLowerCase();

			System.out.println("interesting = "+interesting);

			////////////////////////////////////////////////////////////////////////////////

			String userkeyword = q3TextField.getText();

			System.out.println("userkeyword = "+userkeyword);


			////////////////////////////////////////////////////////////////////////////////
```

```
                        String recentJournalFlag = "";
                        if(q1RadioButton1.isSelected()==true) recentJournalFlag = "yes";
                        if(q1RadioButton2.isSelected()==true) recentJournalFlag = "no";
                        System.out.println("recentJournalFlag : "+recentJournalFlag);

                        ////////////////////////////////////////////////////////////////////////////

                        String numOfRecentSubmitFlag = "yes";
                        System.out.println("numOfRecentSubmitFlag : "+numOfRecentSubmitFlag);

                        ////////////////////////////////////////////////////////////////////////////

                        String considerHistory = "";

                        if(q5RadioButton1.isSelected()==true) considerHistory = "yes";
                        if(q5RadioButton2.isSelected()==true) considerHistory = "no";


                        System.out.println("considerHistory : "+considerHistory);

                        ////////////////////////////////////////////////////////////////////////////

                        String[] token = Lucene.LuceneAnalyzer.analyze(q6TextField.getText() + " "+
                                q6TextField.getText()+" "+
                                q6TextField.getText()+ " "+
                                q6TextArea.getText()
                                );

                        for(int i = 0; i < token.length; i++){

                              System.out.println("token["+i+"] = "+token[i]);
                        }
                        ////////////////////////////////////////////////////////////////////////////
                        recommendedList = prologManager.doRecommendation(stdNum, measureName,
interesting, userkeyword, recentJournalFlag, numOfRecentSubmitFlag, considerHistory, token,
explainList);
                        ////////////////////////////////////////////////////////////////////////////
                        //prologManager.getResult();


                        RecommendationWindow rw = new RecommendationWindow(stdNum,
recommendedList, explainList);
                        rw.setLocationRelativeTo(null);
                        rw.setVisible(true);
                  }});
      }
}


class ButtonHandler implements ActionListener{

   public void actionPerformed(ActionEvent e){

         JOptionPane.showMessageDialog(null, "I've been pressed", "What happened?", JOptionPane.
INFORMATION_MESSAGE);
   }

}
```

```
//LoginFrame.java
package Frame;
import java.awt.event.ActionEvent;
import java.io.File;
import java.io.IOException;
import java.sql.SQLException;

import javax.imageio.ImageIO;
import javax.swing.AbstractAction;
import javax.swing.GroupLayout;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComponent;
import javax.swing.JDialog;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JTextPane;
import javax.swing.LayoutStyle;

import javax.swing.WindowConstants;
import javax.swing.SwingUtilities;

import Database.databaseManager;
import Database.prologManager;


/**
* This code was edited or generated using CloudGarden's Jigloo
* SWT/Swing GUI Builder, which is free for non-commercial
* use. If Jigloo is being used commercially (ie, by a corporation,
* company or business for any purpose whatever) then you
* should purchase a license for each developer using Jigloo.
* Please visit www.cloudgarden.com for details.
* Use of Jigloo implies acceptance of these licensing terms.
* A COMMERCIAL LICENSE HAS NOT BEEN PURCHASED FOR
* THIS MACHINE, SO JIGLOO OR THIS CODE CANNOT BE USED
* LEGALLY FOR ANY CORPORATE OR COMMERCIAL PURPOSE.
*/
public class LoginFrame extends javax.swing.JFrame {
	/**
	 * @uml.property  name="submitButton"
	 * @uml.associationEnd
	 */
	private JButton SubmitButton;
	/**
	 * @uml.property  name="login_ID"
	 * @uml.associationEnd
	 */
	private JTextPane Login_ID;
	/**
	 * @uml.property  name="inputLoginID"
	 * @uml.associationEnd
	 */
	private JTextField InputLoginID;
	/**
	 * @uml.property  name="inputLoginPw"
	 * @uml.associationEnd
```

```
 */
private JTextField InputLoginPw;
/**
 * @uml.property  name="pushSubmitButton"
 * @uml.associationEnd
 */
private AbstractAction PushSubmitButton;
/**
 * @uml.property  name="jLabel_IL"
 * @uml.associationEnd  multiplicity="(1 1)"
 */
private JLabel jLabel_IL;
/**
 * @uml.property  name="oKPutton"
 * @uml.associationEnd
 */
private AbstractAction OKPutton;
/**
 * @uml.property  name="okButton"
 * @uml.associationEnd
 */
private JButton okButton;
/**
 * @uml.property  name="warningMessage"
 * @uml.associationEnd
 */
private JTextPane warningMessage;
/**
 * @uml.property  name="idpwWarning"
 * @uml.associationEnd
 */
private JDialog idpwWarning;

private MainFrame p;

/**
* Auto-generated main method to display this JFrame
*/
public static void main(String[] args) {
	SwingUtilities.invokeLater(new Runnable() {
		public void run() {

			LoginFrame inst = null;
			//ImageIcon img = null;
			try {
				inst = new LoginFrame();

			} catch (IOException e1) {
				// TODO Auto-generated catch block
				e1.printStackTrace();
			}

			inst.setLocationRelativeTo(null);
			inst.setVisible(true);

		}
	});
```

```
	}

	public LoginFrame() throws IOException {
		super();
		try {
			initialize();
		} catch (SQLException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}

		this.setIconImage(ImageIO.read(new File("Penguin_4.ico")));

		this.setContentPane(getJLabel_IL());
		this.setFont(new java.awt.Font("Consolas",0,12));

		//inst.pack();


		initGUI();
	}

	private void initialize() throws SQLException {
		databaseManager.connectionDatabase();
		databaseManager.initialSetup();
		prologManager.connectionProlog();

	}

	private void initGUI() {
		try {
			setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
			GroupLayout thisLayout = new GroupLayout((JComponent)jLabel_IL);
			jLabel_IL.setLayout(thisLayout);
			thisLayout.setVerticalGroup(thisLayout.createSequentialGroup()
				.addContainerGap(50, 50)
				.addGroup(thisLayout.createParallelGroup()
				   .addGroup(thisLayout.createSequentialGroup()
				      .addGap(0, 0, Short.MAX_VALUE)
				      .addComponent(getSubmitButton(), GroupLayout.PREFERRED_SIZE, 145,
GroupLayout.PREFERRED_SIZE))
				   .addGroup(GroupLayout.Alignment.LEADING, thisLayout.createSequentialGroup()
				      .addGap(0, 35, Short.MAX_VALUE)
				      .addComponent(getLogin_ID(), GroupLayout.PREFERRED_SIZE, 24,
GroupLayout.PREFERRED_SIZE)
				      .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
				      .addComponent(getInputLoginID(), GroupLayout.PREFERRED_SIZE, 27,
GroupLayout.PREFERRED_SIZE)
				      .addComponent(getInputLoginPw(), GroupLayout.PREFERRED_SIZE, 28,
GroupLayout.PREFERRED_SIZE)
				      .addGap(25)))
				.addContainerGap(105, 105));
			thisLayout.setHorizontalGroup(thisLayout.createSequentialGroup()
				.addContainerGap(125, 125)
				.addGroup(thisLayout.createParallelGroup()
				   .addGroup(GroupLayout.Alignment.LEADING, thisLayout.createSequentialGroup()
				      .addComponent(getLogin_ID(), GroupLayout.PREFERRED_SIZE, 137,
```

```
GroupLayout.PREFERRED_SIZE)
                                .addGap(6))
                            .addComponent(getInputLoginID(), GroupLayout.Alignment.LEADING,
GroupLayout.PREFERRED_SIZE, 143, GroupLayout.PREFERRED_SIZE)
                            .addComponent(getInputLoginPw(), GroupLayout.Alignment.LEADING,
GroupLayout.PREFERRED_SIZE, 143, GroupLayout.PREFERRED_SIZE))
                        .addGap(85)
                        .addComponent(getSubmitButton(), GroupLayout.PREFERRED_SIZE, 43,
GroupLayout.PREFERRED_SIZE)
                        .addContainerGap(38, Short.MAX_VALUE));
                this.setTitle("20112371");
                jLabel_IL.setBackground(new java.awt.Color(255,255,255));
                this.setPreferredSize(new java.awt.Dimension(450, 338));
                pack();
                this.setSize(450, 338);
            } catch (Exception e) {
                //add your error handling code here
                e.printStackTrace();
            }
        }



        /**
         * @return
         * @uml.property  name="submitButton"
         */
        private JButton getSubmitButton() {
            if(SubmitButton == null) {
                SubmitButton = new JButton();
                SubmitButton.setText("Submit");

                SubmitButton.setFont(new java.awt.Font("Consolas",0,12));
                SubmitButton.setAction(getPushSubmitButton());
                SubmitButton.setBackground(new java.awt.Color(0,0,0));
            }
            return SubmitButton;
        }

        /**
         * @return
         * @uml.property  name="login_ID"
         */
        private JTextPane getLogin_ID() {
            if(Login_ID == null) {
                Login_ID = new JTextPane();
                Login_ID.setText("Put your ID/PW");
                Login_ID.setFont(new java.awt.Font("Consolas",0,12));
                Login_ID.setEditable(false);
                Login_ID.setBackground(new java.awt.Color(255,255,255));
                Login_ID.setFont(new java.awt.Font("Aharoni",0,16));
            }
            return Login_ID;
        }

        /**
         * @return
```

```
	 * @uml.property  name="inputLoginID"
	 */
	private JTextField getInputLoginID() {
		if(InputLoginID == null) {
			InputLoginID = new JTextField();

			InputLoginID.setText("20112371");
			InputLoginID.setFont(new java.awt.Font("Consolas",0,12));
		}
		return InputLoginID;
	}

	/**
	 * @return
	 * @uml.property  name="inputLoginPw"
	 */
	private JTextField getInputLoginPw() {
		if(InputLoginPw == null) {
			InputLoginPw = new JTextField();
			InputLoginPw.setText("20112371");
			InputLoginPw.setFont(new java.awt.Font("Consolas",0,12));
		}
		return InputLoginPw;
	}

	/**
	 * @return
	 * @uml.property  name="pushSubmitButton"
	 */
	private AbstractAction getPushSubmitButton() {

		if(PushSubmitButton == null) {
			PushSubmitButton = new AbstractAction("Push", null) {
				public void actionPerformed(ActionEvent evt) {
					String id = """+InputLoginID.getText()+""";
					String pw = """+InputLoginPw.getText()+""";
					//20128192
					if(InputLoginID.getText().equals("") || InputLoginPw.getText().equals("") ||
prologManager.checkUser(id, pw) == false){
						idpwWarning = getIdpwWarning();
					idpwWarning.setLocationRelativeTo(null);
					idpwWarning.pack();
					idpwWarning.setVisible(true);
					}
					else{
						dispose();
						p = new MainFrame(InputLoginID.getText());

						ImageIcon img = new ImageIcon("Penguin_4.ico");
						p.setIconImage(img.getImage());


						//System.out.println("Login part = "+InputLoginID.getText());
						p.setLocationRelativeTo(null);
						p.setVisible(true);
					}
				}
```

```
                };
            }
            return PushSubmitButton;
        }

        /**
         * @return
         * @uml.property  name="idpwWarning"
         */
        private JDialog getIdpwWarning() {
            if(idpwWarning == null) {
                idpwWarning = new JDialog(this);
                GroupLayout idpwWarningLayout = new
GroupLayout((JComponent)idpwWarning.getContentPane());
                idpwWarning.setLayout(idpwWarningLayout);
                idpwWarning.setPreferredSize(new java.awt.Dimension(243, 158));
                idpwWarning.setEnabled(true);
                idpwWarning.getContentPane().setBackground(new java.awt.Color(255,255,255));
                idpwWarning.setSize(243, 158);
                idpwWarningLayout.setHorizontalGroup(idpwWarningLayout.createSequentialGroup()
                    .addContainerGap(21, 21)
                    .addGroup(idpwWarningLayout.createParallelGroup()
                       .addGroup(idpwWarningLayout.createSequentialGroup()
                          .addGap(0, 0, Short.MAX_VALUE)
                          .addComponent(getWarningMessage(), GroupLayout.PREFERRED_SIZE, 194,
GroupLayout.PREFERRED_SIZE))
                       .addGroup(GroupLayout.Alignment.LEADING,
idpwWarningLayout.createSequentialGroup()
                          .addGap(50)
                          .addComponent(getOkButton(), GroupLayout.PREFERRED_SIZE, 82,
GroupLayout.PREFERRED_SIZE)
                          .addGap(0, 62, Short.MAX_VALUE)))
                    .addContainerGap());
                idpwWarningLayout.setVerticalGroup(idpwWarningLayout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(getWarningMessage(), GroupLayout.PREFERRED_SIZE, 30,
GroupLayout.PREFERRED_SIZE)
                    .addGap(0, 27, Short.MAX_VALUE)
                    .addComponent(getOkButton(), GroupLayout.PREFERRED_SIZE, 40,
GroupLayout.PREFERRED_SIZE)
                    .addContainerGap());
            }
            return idpwWarning;
        }

        /**
         * @return
         * @uml.property  name="warningMessage"
         */
        private JTextPane getWarningMessage() {
            if(warningMessage == null) {

                warningMessage = new JTextPane();
                warningMessage.setText("Check your id / password please");
                warningMessage.setEditable(false);
            }
            return warningMessage;
```

```
	}

	/**
	 * @return
	 * @uml.property  name="okButton"
	 */
	private JButton getOkButton() {
		if(okButton == null) {
			okButton = new JButton();
			okButton.setText("OK");
			okButton.setAction(getOKPutton());

		}
		return okButton;
	}

	/**
	 * @return
	 * @uml.property  name="oKPutton"
	 */
	private AbstractAction getOKPutton() {
		if(OKPutton == null) {
			OKPutton = new AbstractAction("OK", null) {
				public void actionPerformed(ActionEvent evt) {
					idpwWarning.dispose();
				}
			};
		}
		return OKPutton;
	}

	private JLabel getJLabel_IL() {
		if(jLabel_IL == null) {
			try {
				jLabel_IL = new JLabel(new ImageIcon(ImageIO.read(new File("background.jpg"))));
			} catch (IOException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			}
			jLabel_IL.setSize(300, 265);
		}
		return jLabel_IL;
	}
}
```

```
//LuceneAnalyzer.java

package Lucene;

import org.apache.lucene.index.IndexReader;
import org.apache.lucene.index.IndexWriter;
import org.apache.lucene.analysis.Analyzer;
import org.apache.lucene.analysis.TokenStream;
import org.apache.lucene.analysis.standard.StandardAnalyzer;
import org.apache.lucene.document.Document;
import org.apache.lucene.document.Field;
import org.apache.lucene.store.FSDirectory;
import org.apache.lucene.store.Directory;
import org.apache.lucene.util.Version;

import java.io.File;
import java.io.FileFilter;
import java.io.IOException;
import java.io.FileReader;
import java.io.Reader;
import java.util.Arrays;
import java.util.Iterator;
import java.util.List;



public class LuceneAnalyzer {
 rivate static final String examples = "The main disadvantage of the k-means algorithm is that the number of clusters,
 K, must be supplied as a parameter "+
"In this paper we present a simple validity measure based on the intra-cluster and inter-cluster distance measures "+
"which allows the number of clusters to be determined automatically. The basic procedure involves producing all "+
"the segmented images for 2 clusters up to Kmax clusters, where Kmax represents an upper limit on the number of
"+
"clusters. Then our validity measure is calculated to determine which is the best clustering by finding the minimum
"+
"value for our measure. The validity measure is tested for synthetic images for which the number of clusters in "+
"known, and is also implemented for natural images.";
      private static final  Analyzer analyzers= new StandardAnalyzer(Version.LUCENE_30);
      public static void main(String[] args){
 String t = examples;
 analyze(t);


      public static String[] analyze(String t ){
 Analyzer analyzer = new StandardAnalyzer(Version.LUCENE_30);
 String name = analyzer.getClass().getSimpleName();
 List<String> lst = null;
 try {
       //AnalyzerUtils.displayTokens(analyzer, t);
      lst = AnalyzerUtils.getAnalyzedTokens(analyzer, t);
 } catch (IOException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
 }

            String[] ll = new String[lst.size()];
 int i = 0;
```

```
 for(Iterator it = lst.iterator(); it.hasNext(); ){
       String ttt = (String) it.next();
       ll[i++] = ttt;
       System.out.println(ttt);
 }

             System.out.println("\n");
             return ll;
       }
}
```

```
//MainFrame.java
package Frame;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.io.File;
import java.io.IOException;
import java.sql.SQLException;

import javax.imageio.ImageIO;
import javax.swing.AbstractAction;
import javax.swing.BorderFactory;
import javax.swing.GroupLayout;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComponent;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextPane;
import javax.swing.LayoutStyle;

import javax.swing.WindowConstants;
import javax.swing.border.LineBorder;

import Database.databaseManager;
import Database.prologManager;


/**
* This code was edited or generated using CloudGarden's Jigloo
* SWT/Swing GUI Builder, which is free for non-commercial
* use. If Jigloo is being used commercially (ie, by a corporation,
* company or business for any purpose whatever) then you
* should purchase a license for each developer using Jigloo.
* Please visit www.cloudgarden.com for details.
* Use of Jigloo implies acceptance of these licensing terms.
* A COMMERCIAL LICENSE HAS NOT BEEN PURCHASED FOR
* THIS MACHINE, SO JIGLOO OR THIS CODE CANNOT BE USED
* LEGALLY FOR ANY CORPORATE OR COMMERCIAL PURPOSE.
*/
public class MainFrame extends javax.swing.JFrame {
 **
       * @uml.property  name="userPrivacyInfo"
 * @uml.associationEnd  multiplicity="(1 1)"
 */
      private JPanel UserPrivacyInfo;
 **
       * @uml.property  name="picture"
 * @uml.associationEnd  multiplicity="(1 1)"
 */
      private JPanel Picture;
 **
       * @uml.property  name="userPicture"
```

```
* @uml.associationEnd
*/
    private JLabel UserPicture;
**
     * @uml.property  name="jPhotoPanel"
* @uml.associationEnd
*/
    private JPanel JPhotoPanel;
**
     * @uml.property  name="journalLabel"
* @uml.associationEnd
*/
    private JLabel JournalLabel;
**
     * @uml.property  name="userInfoLabel"
* @uml.associationEnd
*/
    private JLabel UserInfoLabel;
**
     * @uml.property  name="jScrollPane1"
* @uml.associationEnd
*/
    private JScrollPane jScrollPane1;
**
     * @uml.property  name="journalList"
* @uml.associationEnd
*/
    private JTextPane JournalList;
**
     * @uml.property  name="helpAction"
* @uml.associationEnd
*/
    private AbstractAction HelpAction;
**
     * @uml.property  name="jMenuItem2"
* @uml.associationEnd
*/
    private JMenuItem jMenuItem2;
**
     * @uml.property  name="fileAction"
* @uml.associationEnd
*/
    private AbstractAction FileAction;
**
     * @uml.property  name="jMenuItem1"
* @uml.associationEnd  multiplicity="(1 1)"
*/
    private JMenuItem jMenuItem1;
**
     * @uml.property  name="userInfoName"
* @uml.associationEnd  multiplicity="(1 1)"
*/
    private JTextPane userInfoName;
**
     * @uml.property  name="publishedJournal"
* @uml.associationEnd  multiplicity="(1 1)"
*/
    private JPanel publishedJournal;
**
     * @uml.property  name="jLabel_IL"
* @uml.associationEnd
```

```
*/
	private JLabel jLabel_IL;
**
	 * @uml.property  name="getInfoStart"
* @uml.associationEnd  multiplicity="(1 1)"
*/
	private JButton GetInfoStart;
**
	 * @uml.property  name="helpManu"
* @uml.associationEnd  multiplicity="(1 1)"
*/
	private JMenu HelpManu;
**
	 * @uml.property  name="jMenu1"
* @uml.associationEnd  multiplicity="(1 1)"
*/
	private JMenu jMenu1;
**
	 * @uml.property  name="mainManuBar"
* @uml.associationEnd  multiplicity="(1 1)"
*/
	private JMenuBar MainManuBar;
**
	 * @uml.property  name="stdNum"
*/
	private String stdNum;
**
	* Auto-generated main method to display this JFrame
/
	private AboutDialog ad;
	private GetInfo inst;



	public MainFrame() {
super();
initGUI();


	public MainFrame(String p) {

		super();
this.stdNum = p;
this.setContentPane(getJLabel_IL());


initGUI();

	}

	private void initGUI() {
try {
	 GroupLayout thisLayout = new GroupLayout((JComponent)getContentPane());
	setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
	getContentPane().setLayout(thisLayout);
	this.setTitle("Journal Recommendation System");
	{
			MainManuBar = new JMenuBar();
		setJMenuBar(MainManuBar);
		{
			jMenu1 = new JMenu();
```

```
                MainManuBar.add(jMenu1);
                jMenu1.setText("File");
                {
                    jMenuItem1 = new JMenuItem();
                    jMenu1.add(jMenuItem1);
                    jMenuItem1.setText("Exit");
                    jMenuItem1.setAction(getFileAction());
                }
            }
                    {
                HelpManu = new JMenu();
                MainManuBar.add(HelpManu);
                HelpManu.setText("Help");
                HelpManu.add(getJMenuItem2());
            }
                    }
        {
                    UserPrivacyInfo = new JPanel();
            GroupLayout UserPrivacyInfoLayout = new GroupLayout((JComponent)UserPrivacyInfo);
            UserPrivacyInfo.setLayout(UserPrivacyInfoLayout);
            UserPrivacyInfo.setBackground(new java.awt.Color(215,230,251));
            UserPrivacyInfo.setBorder(new LineBorder(new java.awt.Color(0,0,0), 1, false));
            {
                 userInfoName = new JTextPane();
                System.out.println("StdUserID " + this.stdNum);
                String t = "  Name  \t" + prologManager.getUserName(stdNum)+ "\n" +
                           "  Sex  \t" + prologManager.getUserSex(stdNum)+ "\n" +
                           "  Nat\t" + prologManager.getUserNationality(stdNum)+ "\n" +
                           "  Phone  \t" +  prologManager.getUserPhone(stdNum).replaceAll("\'", "")+
"\n" +
                           "  Email \t" + prologManager.getUserEmail(stdNum).replaceAll("\'", "")+ "\n"
+
                           "  Major \t" + prologManager.getUserMajor(stdNum)+ "\n" +
                           "  Status \t" +prologManager.getUserStatus(stdNum)+ "\n" +
                           "  Gpa \t" +prologManager.getUserGpa(stdNum).replaceAll("\'", "")+ "\n" +
                           "  Lang \t" +prologManager.getUserLanguage(stdNum).replaceAll("\'", "")+
"\n";


                           userInfoName.setText(t);
                userInfoName.setFont(new java.awt.Font("Consolas",0,11));

                userInfoName.setEditable(false);
                userInfoName.setBorder(new LineBorder(new java.awt.Color(0,0,0), 1, false));
                userInfoName.setBackground(new java.awt.Color(234,251,240));
            }

serPrivacyInfoLayout.setHorizontalGroup(UserPrivacyInfoLayout.createSequentialGroup()
                .addContainerGap()
                .addGroup(UserPrivacyInfoLayout.createParallelGroup()
                    .addComponent(getUserInfoLabel(), GroupLayout.Alignment.LEADING,
GroupLayout.PREFERRED_SIZE, 177, GroupLayout.PREFERRED_SIZE)
                    .addGroup(UserPrivacyInfoLayout.createSequentialGroup()
                        .addGap(20)
                        .addGroup(UserPrivacyInfoLayout.createParallelGroup()
                            .addComponent(getJPhotoPanel(), GroupLayout.Alignment.LEADING,
GroupLayout.PREFERRED_SIZE, 135, GroupLayout.PREFERRED_SIZE)
                            .addGroup(GroupLayout.Alignment.LEADING,
UserPrivacyInfoLayout.createSequentialGroup()
                                .addGap(44)
                                .addComponent(getUserPicture(), GroupLayout.PREFERRED_SIZE, 67,
```

```
GroupLayout.PREFERRED_SIZE)
                                .addGap(24)))
                        .addGap(22)))
                .addComponent(userInfoName, GroupLayout.PREFERRED_SIZE, 241,
GroupLayout.PREFERRED_SIZE)
                .addContainerGap(27, Short.MAX_VALUE));
        UserPrivacyInfoLayout.setVerticalGroup(UserPrivacyInfoLayout.createSequentialGroup()
                .addContainerGap()
                .addComponent(getUserInfoLabel(), GroupLayout.PREFERRED_SIZE,
GroupLayout.PREFERRED_SIZE, GroupLayout.PREFERRED_SIZE)
                .addGroup(UserPrivacyInfoLayout.createParallelGroup()
                    .addComponent(userInfoName, GroupLayout.Alignment.LEADING,
GroupLayout.PREFERRED_SIZE, 166, GroupLayout.PREFERRED_SIZE)
                    .addGroup(GroupLayout.Alignment.LEADING,
UserPrivacyInfoLayout.createSequentialGroup()
                        .addGap(12)
                        .addComponent(getJPhotoPanel(), GroupLayout.PREFERRED_SIZE, 148,
GroupLayout.PREFERRED_SIZE)))
                .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(getUserPicture(), GroupLayout.PREFERRED_SIZE, 21,
GroupLayout.PREFERRED_SIZE)
                .addContainerGap(22, Short.MAX_VALUE));
    }
                {
        GetInfoStart = new JButton();
        GetInfoStart.setText("Start");
        GetInfoStart.setFont(new java.awt.Font("Calibri",0,20));
        GetInfoStart.setBounds(100,0,900,735);
        GetInfoStart.addMouseListener(new MouseAdapter() {
            public void mouseClicked(MouseEvent evt) {
                GetInfoStartMouseClicked(evt);
            }
         });
            }
    {
                publishedJournal = new JPanel();
        GroupLayout publishedJournalLayout = new GroupLayout((JComponent)publishedJournal);
        publishedJournal.setLayout(publishedJournalLayout);
        publishedJournal.setBackground(new java.awt.Color(215,230,251));
        publishedJournal.setBorder(new LineBorder(new java.awt.Color(0,0,0), 1, false));
        publishedJournalLayout.setHorizontalGroup(publishedJournalLayout.createSequentialGroup()
            .addContainerGap()
            .addGroup(publishedJournalLayout.createParallelGroup()
                .addGroup(publishedJournalLayout.createSequentialGroup()
                    .addComponent(getJScrollPane1(), GroupLayout.PREFERRED_SIZE, 612,
GroupLayout.PREFERRED_SIZE)
                    .addGap(0, 0, Short.MAX_VALUE))
                .addGroup(GroupLayout.Alignment.LEADING,
publishedJournalLayout.createSequentialGroup()
                    .addComponent(getJournalLabel(), GroupLayout.PREFERRED_SIZE, 166,
GroupLayout.PREFERRED_SIZE)
                    .addGap(0, 446, Short.MAX_VALUE)))
            .addContainerGap(22, 22));
        publishedJournalLayout.setVerticalGroup(publishedJournalLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(getJournalLabel(), 0, 25, Short.MAX_VALUE)
            .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED, 0,
GroupLayout.PREFERRED_SIZE)
            .addComponent(getJScrollPane1(), GroupLayout.PREFERRED_SIZE, 165,
GroupLayout.PREFERRED_SIZE)
            .addContainerGap());
```

```
    }
                {
            final ImageIcon icon;
            byte[] b = databaseManager.readImage(stdNum);
            Image img = Toolkit.getDefaultToolkit().createImage(b);
            img = img.getScaledInstance(150, 235, Image.SCALE_DEFAULT);
            ImageIcon imgc = new ImageIcon(ImageIO.read(new File("expert.jpg")));
            icon = imgc;
            Picture = new JPanel(){
           public void paintComponent(Graphics g) {
               //  Approach 1: Dispaly image at at full size
               g.drawImage(icon.getImage(), 0, 0, null);
               //  Approach 2: Scale image to size of component
               // Dimension d = getSize();
               // g.drawImage(icon.getImage(), 0, 0, d.width, d.height, null);
               // Approach 3: Fix the image position in the scroll pane
               // Point p = scrollPane.getViewport().getViewPosition();
               // g.drawImage(icon.getImage(), p.x, p.y, null);
               setOpaque(false);
               super.paintComponent(g);
           }
        };
            Picture.setBackground(new java.awt.Color(255,255,255));
            Picture.setBorder(new LineBorder(new java.awt.Color(0,0,0), 1, false));
            Picture.setSize(10, 10);

    }
                    thisLayout.setVerticalGroup(thisLayout.createSequentialGroup()
            .addContainerGap(35, 35)
            .addGroup(thisLayout.createParallelGroup()
                .addComponent(UserPrivacyInfo, GroupLayout.Alignment.LEADING,
GroupLayout.PREFERRED_SIZE, 243, GroupLayout.PREFERRED_SIZE)
                .addGroup(GroupLayout.Alignment.LEADING, thisLayout.createSequentialGroup()
                    .addGap(9)
                    .addComponent(Picture, GroupLayout.PREFERRED_SIZE, 235,
GroupLayout.PREFERRED_SIZE)
                    .addGap(9)))
            .addGap(0, 47, Short.MAX_VALUE)
            .addComponent(publishedJournal, GroupLayout.PREFERRED_SIZE, 222,
GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(GetInfoStart, GroupLayout.PREFERRED_SIZE, 47,
GroupLayout.PREFERRED_SIZE)
            .addContainerGap(58, 58));
       thisLayout.setHorizontalGroup(thisLayout.createSequentialGroup()
            .addContainerGap(58, 58)
            .addGroup(thisLayout.createParallelGroup()
                .addGroup(GroupLayout.Alignment.LEADING, thisLayout.createSequentialGroup()
                    .addComponent(Picture, GroupLayout.PREFERRED_SIZE, 150,
GroupLayout.PREFERRED_SIZE)
                    .addGap(27)
                    .addGroup(thisLayout.createParallelGroup()
                        .addGroup(GroupLayout.Alignment.LEADING, thisLayout.createSequentialGroup()
                            .addComponent(GetInfoStart, GroupLayout.PREFERRED_SIZE, 275,
GroupLayout.PREFERRED_SIZE)
                            .addGap(0, 241, Short.MAX_VALUE))
                        .addGroup(GroupLayout.Alignment.LEADING, thisLayout.createSequentialGroup()
                            .addPreferredGap(GetInfoStart, UserPrivacyInfo,
LayoutStyle.ComponentPlacement.INDENT)
                            .addComponent(UserPrivacyInfo, GroupLayout.PREFERRED_SIZE, 459,
GroupLayout.PREFERRED_SIZE)
```

```
                        .addGap(0, 0, Short.MAX_VALUE))))
                .addComponent(publishedJournal, GroupLayout.Alignment.LEADING, 0, 648,
Short.MAX_VALUE))
                .addContainerGap(46, 46));
        pack();
        this.setSize(768, 729);
 } catch (Exception e) {
     //add your error handling code here
        e.printStackTrace();
 }

                /*
  *
  * */
        }
        private void GetInfoStartMouseClicked(MouseEvent evt) {
 dispose();
 inst = new GetInfo(stdNum);
 inst.setLocationRelativeTo(null);
 inst.setVisible(true);


        }

 **
         * @return
 * @uml.property  name="fileAction"
 */
        private AbstractAction getFileAction() {
 if(FileAction == null) {
        FileAction = new AbstractAction("Exit", null) {
                public void actionPerformed(ActionEvent evt) {
                        dispose();
                }
                        };
 }
                return FileAction;


        private JMenuItem getJMenuItem2() {
 if(jMenuItem2 == null) {
        jMenuItem2 = new JMenuItem();
        jMenuItem2.setText("jMenuItem2");
        jMenuItem2.setAction(getHelpAction());
 }
                return jMenuItem2;


        /**
 * @return
 * @uml.property  name="helpAction"
 */
        private AbstractAction getHelpAction() {
 if(HelpAction == null) {
        HelpAction = new AbstractAction("About", null) {
                public void actionPerformed(ActionEvent evt) {
                        ad = new AboutDialog(null);
                        ad.setLocationRelativeTo(null);
                        ad.setVisible(true);
                }
                        };
```

```
}
		return HelpAction;


	/**
* @return
* @uml.property  name="journalList"
*/
	private JTextPane getJournalList() {
if(JournalList == null) {
	JournalList = new JTextPane();
	String t = prologManager.getJournalList(stdNum);
	JournalList.setText(t.replace('_', ' '));
	JournalList.setPreferredSize(new java.awt.Dimension(609, 153));
	JournalList.setEditable(false);
	JournalList.setBorder(new LineBorder(new java.awt.Color(0,0,0), 1, false));
	JournalList.setSize(605, 117);
	JournalList.setBackground(new java.awt.Color(234,251,240));

}
		return JournalList;


	private JScrollPane getJScrollPane1() {
if(jScrollPane1 == null) {
	jScrollPane1 = new JScrollPane();
	jScrollPane1.setViewportView(getJournalList());
}
		return jScrollPane1;


	/**
* @return
* @uml.property  name="userInfoLabel"
*/
	private JLabel getUserInfoLabel() {
if(UserInfoLabel == null) {
	UserInfoLabel = new JLabel();
	UserInfoLabel.setText("User Information");
	UserInfoLabel.setFont(new java.awt.Font("맑은 고딕",1,14));
}
		return UserInfoLabel;


	/**
* @return
* @uml.property  name="journalLabel"
*/
	private JLabel getJournalLabel() {
if(JournalLabel == null) {
	JournalLabel = new JLabel();
	JournalLabel.setText("Published JournalList");
	JournalLabel.setFont(new java.awt.Font("맑은 고딕",1,14));
}
		return JournalLabel;


	private JLabel getJLabel_IL() {

		if(jLabel_IL == null) {
	try {
```

```
            jLabel_IL = new JLabel(new ImageIcon(ImageIO.read(new File("mainbackground.jpg"))));
            jLabel_IL.setPreferredSize(new java.awt.Dimension(735, 664));
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
                //jLabel_IL.setSize(300, 265);
    }
            return jLabel_IL;


        /**
     * @return
     * @throws SQLException
     * @uml.property  name="jPhotoPanel"
     */
        private JPanel getJPhotoPanel() throws SQLException {
    if(JPhotoPanel == null) {



                final ImageIcon icon;
        byte[] b = databaseManager.readImage(stdNum);
        Image img = Toolkit.getDefaultToolkit().createImage(b);
        img = img.getScaledInstance(150, 150, Image.SCALE_DEFAULT);
        icon =new ImageIcon(img);
                JPhotoPanel = new JPanel(){
               public void paintComponent(Graphics g) {
                //  Approach 1: Dispaly image at at full size
                g.drawImage(icon.getImage(), 0, 0, null);
                //  Approach 2: Scale image to size of component
                // Dimension d = getSize();
                // g.drawImage(icon.getImage(), 0, 0, d.width, d.height, null);
                // Approach 3: Fix the image position in the scroll pane
                // Point p = scrollPane.getViewport().getViewPosition();
                // g.drawImage(icon.getImage(), p.x, p.y, null);
                setOpaque(false);
                super.paintComponent(g);
               }
            };
            JPhotoPanel.setBackground(new java.awt.Color(255,255,255));
            JPhotoPanel.setBorder(new LineBorder(new java.awt.Color(0,0,0), 1, false));
            }
            return JPhotoPanel;
        }

        /**
         * @return
         * @uml.property  name="userPicture"
         */
        private JLabel getUserPicture() {
            if(UserPicture == null) {
                UserPicture = new JLabel();
                UserPicture.setText("Picture");
            }
            return UserPicture;
        }

}
```

```
//Popup.java
package Frame;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.Toolkit;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.util.Hashtable;

import javax.swing.GroupLayout;
import javax.swing.JButton;
import javax.swing.JComponent;

import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JTextPane;
import javax.swing.LayoutStyle;
import javax.swing.SwingUtilities;

import Database.databaseManager;
import Database.prologManager;


/**
* This code was edited or generated using CloudGarden's Jigloo
* SWT/Swing GUI Builder, which is free for non-commercial
* use. If Jigloo is being used commercially (ie, by a corporation,
* company or business for any purpose whatever) then you
* should purchase a license for each developer using Jigloo.
* Please visit www.cloudgarden.com for details.
* Use of Jigloo implies acceptance of these licensing terms.
* A COMMERCIAL LICENSE HAS NOT BEEN PURCHASED FOR
* THIS MACHINE, SO JIGLOO OR THIS CODE CANNOT BE USED
* LEGALLY FOR ANY CORPORATE OR COMMERCIAL PURPOSE.
*/
public class Popup extends javax.swing.JDialog {
 **
        * @uml.property  name="aboutDialogOK"
 * @uml.associationEnd  multiplicity="(1 1)"
 */
       private JButton AboutDialogOK;
 rivate JTextPane Text;
 rivate JPanel Imageggg;
 rivate String isbn;
 rivate String score;
 rivate boolean isScore;
 rivate String expr;
 **
       * Auto-generated main method to display this JDialog
 /

       public Popup(JFrame frame) {
 super(frame);
 initGUI();
```

```
    public Popup(String t, String s){
super();
isScore = false;
isbn = t;
score = s;
initGUI();


    public Popup(String t, String s, String exp){
super();
isScore = true;
isbn = t;
score = s;
expr = exp;
System.out.println("expr = "+expr);
initGUI();


    private void initGUI() {
try {
     GroupLayout thisLayout = new GroupLayout((JComponent)getContentPane());
    getContentPane().setLayout(thisLayout);
    getContentPane().setBackground(new java.awt.Color(255,255,255));
    this.setTitle("Popup!..");

            this.setFont(new java.awt.Font("Consolas",0,12));
    {
            AboutDialogOK = new JButton();
        AboutDialogOK.setText("OK");
        AboutDialogOK.addMouseListener(new MouseAdapter() {
            public void mouseClicked(MouseEvent evt) {
            AboutDialogOKMouseClicked(evt);
            }
         });
            }
    {
            Text = new JTextPane();
        if(isScore == false){
            Text.setText("Journal Name : "+prologManager.getJournalName(isbn)+"\n"+
                        "Score : "+score+"\n"+
                        "Publisher : "+prologManager.getJournalPublisher(isbn)+"\n"+
                        "Publication_history : "+prologManager.getJournalPublicationHistory(isbn)+"\n"+
                        "Language : "+prologManager.getJournalLanguage(isbn)+"\n"+
                        "Impact factor : "+prologManager.getJournalImpactFactor(isbn)+"\n"+
                        "Homepage : "+prologManager.getJournalHomepage(isbn)+"\n"+
                        "Cited_half_life : "+prologManager.getJournalCitedHalfLife(isbn)+"\n"+
                        "Article : "+prologManager.getJournalArticle(isbn)+"\n"
                        );
            }
            else{
                Text.setText(expr);
            }

            Text.setEditable(false);
        }
        {
            final ImageIcon icon;
```

```
                    System.out.println("isbn = "+isbn);
                    byte[] b = databaseManager.readJournalImage(isbn);
                    Image img = Toolkit.getDefaultToolkit().createImage(b);
                    img = img.getScaledInstance(216, 275, Image.SCALE_DEFAULT);
                    //ImageIcon imgc = new ImageIcon(ImageIO.read(new File("expert.jpg")));
                    icon =new ImageIcon(img);
                    Imageggg = new JPanel(){
                        public void paintComponent(Graphics g) {
                            //  Approach 1: Dispaly image at at full size
                            g.drawImage(icon.getImage(), 0, 0, null);
                        //  Approach 2: Scale image to size of component
                        // Dimension d = getSize();
                        // g.drawImage(icon.getImage(), 0, 0, d.width, d.height, null);
                        // Approach 3: Fix the image position in the scroll pane
                        // Point p = scrollPane.getViewport().getViewPosition();
                        // g.drawImage(icon.getImage(), p.x, p.y, null);
                        setOpaque(false);
                        super.paintComponent(g);
                        }
                    };
                    Imageggg.setSize(216, 275);
                }
                thisLayout.setVerticalGroup(thisLayout.createSequentialGroup()
                    .addContainerGap(20, 20)
                    .addComponent(Imageggg, GroupLayout.PREFERRED_SIZE, 275,
GroupLayout.PREFERRED_SIZE)
                    .addGap(19)
                    .addComponent(Text, 0, 182, Short.MAX_VALUE)
                    .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(AboutDialogOK, GroupLayout.PREFERRED_SIZE, 46,
GroupLayout.PREFERRED_SIZE)
                    .addContainerGap());
                thisLayout.setHorizontalGroup(thisLayout.createSequentialGroup()
                    .addContainerGap(35, 35)
                    .addGroup(thisLayout.createParallelGroup()
                       .addGroup(thisLayout.createSequentialGroup()
                          .addComponent(Text, GroupLayout.PREFERRED_SIZE, 485,
GroupLayout.PREFERRED_SIZE)
                          .addGap(0, 0, Short.MAX_VALUE))
                       .addGroup(thisLayout.createSequentialGroup()
                          .addGap(128)
                          .addGroup(thisLayout.createParallelGroup()
                             .addGroup(thisLayout.createSequentialGroup()
                                .addComponent(Imageggg, GroupLayout.PREFERRED_SIZE, 216,
GroupLayout.PREFERRED_SIZE)
                                .addGap(0, 0, Short.MAX_VALUE))
                             .addGroup(GroupLayout.Alignment.LEADING, thisLayout.createSequentialGroup()
                                .addGap(60)
                                .addComponent(AboutDialogOK, GroupLayout.PREFERRED_SIZE, 97,
GroupLayout.PREFERRED_SIZE)
                                .addGap(0, 59, Short.MAX_VALUE)))
                          .addGap(141)))
                    .addContainerGap(41, 41));
                this.setSize(577, 700);
            } catch (Exception e) {
                e.printStackTrace();
            }
```

```
    }

    private void AboutDialogOKMouseClicked(MouseEvent evt) {
        this.dispose();
        //TODO add your code for AboutDialogOK.mouseClicked
    }

}
```

```
//prologManager.java
package Database;


import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.Comparator;
import java.util.Enumeration;
import java.util.Hashtable;
import java.util.Map;
import java.util.Set;
import java.lang.Integer;

import jpl.*;

public class prologManager {
 ashtable<String, Double> htResultSet = new Hashtable<String, Double>();
 /Connection prolog
 ublic prologManager(){



      public static void test(){


      String t1 = "consult('family.pl')";
      Query q1 = new Query(t1);
      System.out.println( t1 + " " + (q1.hasMoreSolutions() ? "success!" : "fail!") );


      String t2 = "father(jim, melody)";
      Query q2 = new Query(t2);

      System.out.println( t2 + " " + q2.hasMoreSolutions() );


      String t3 = "father(X, melody)";
      Query q3 = new Query(t3);

      Hashtable ht[] = q3.allSolutions();

      System.out.println( t3 + "'s Solution is ");
      for(int i = 0; i < ht.length; i++){
       ystem.out.println(" getSol is " + ht[i].get("X"));
      }


      String t4 = "father(X, Y)";
      Query q4 = new Query(t4);


      System.out.println( "Each Solution is ");
      while (q4.hasMoreElements()){
       ashtable ht3 = q4.nextSolution();
       ystem.out.println("X = "+ht3.get("X")+"  Y = "+ht3.get("Y"));
```

```
    }

    public static boolean checkUser(String name, String pw){

        String statement = "user("+name+","+pw+").";
System.out.println(statement);
Query q = new Query(statement);

        return q.hasMoreSolutions();

    public static boolean connectionProlog(){

    String t1 = "consult('knowledgebase.pl')";
    Query q1 = new Query(t1);
    System.out.println( t1 + " " + (q1.hasMoreSolutions() ? "success!" : "fail!") );
    return true;

    public static String getUserName(String stdNum){
    String t3 = "name('"+stdNum+"',X)";
    Query q3 = new Query(t3);

    Hashtable ht[] = q3.allSolutions();

    if(ht.length != 0){
     eturn ht[0].get("X").toString();              }
    return null;

    public static String getUserSex(String stdNum){
    String t3 = "sex('"+stdNum+"',X)";
    Query q3 = new Query(t3);

    Hashtable ht[] = q3.allSolutions();

    if(ht.length != 0){
     eturn ht[0].get("X").toString();              }
    return null;

    public static String getUserNationality(String stdNum){
    String t3 = "nationality('"+stdNum+"',X)";
    Query q3 = new Query(t3);

    Hashtable ht[] = q3.allSolutions();

    if(ht.length != 0){
     eturn ht[0].get("X").toString();              }
    return null;

    public static String getUserPhone(String stdNum){
    String t3 = "phone('"+stdNum+"',X)";
    Query q3 = new Query(t3);
```

```
Hashtable ht[] = q3.allSolutions();

if(ht.length != 0){
 eturn ht[0].get("X").toString();                }
return null;


public static String getUserEmail(String stdNum){
String t3 = "email('"+stdNum+"',X)";
Query q3 = new Query(t3);

Hashtable ht[] = q3.allSolutions();

if(ht.length != 0){
 eturn ht[0].get("X").toString();                }
return null;


public static String getUserMajor(String stdNum){
String t3 = "major('"+stdNum+"',X)";
Query q3 = new Query(t3);

Hashtable ht[] = q3.allSolutions();

if(ht.length != 0){
 eturn ht[0].get("X").toString();                }
return null;


public static String getUserStatus(String stdNum){
String t3 = "status('"+stdNum+"',X)";
Query q3 = new Query(t3);

Hashtable ht[] = q3.allSolutions();

if(ht.length != 0){
 eturn ht[0].get("X").toString();                }
return null;


public static String getUserGpa(String stdNum){
String t3 = "gpa('"+stdNum+"',X)";
Query q3 = new Query(t3);

Hashtable ht[] = q3.allSolutions();

if(ht.length != 0){
 eturn ht[0].get("X").toString();                }
return null;


public static String getUserLanguage(String stdNum){
String t3 = "language('"+stdNum+"',X)";
Query q3 = new Query(t3);

Hashtable ht[] = q3.allSolutions();
```

```
    if(ht.length != 0){
     eturn ht[0].get("X").toString();                }
    return null;




ublic static String getJournalList(String stdNum) {
    String t3 = "submit('"+stdNum+"',X)";
    Query q3 = new Query(t3);

    Hashtable ht[] = q3.allSolutions();
    String t = "";
    for(int i = 0; i < ht.length; i++){
      += "  "+ht[i].get("X").toString()+"\n";
    }
        return t;
    }

    public static String getJournalName(String isbn) {
    String t3 = "name('"+isbn.replace("\'", "")+"',X)";
    Query q3 = new Query(t3);

    Hashtable ht[] = q3.allSolutions();
    String t = "";
    for(int i = 0; i < ht.length; i++){
        t += "  "+ht[i].get("X").toString();
    }
        return t;
    }

    public static String getJournalPublisher(String isbn){
    String t3 = "publisher('"+isbn.replace("\'", "")+"',X)";
    Query q3 = new Query(t3);

    Hashtable ht[] = q3.allSolutions();
    String t = "";
    for(int i = 0; i < ht.length; i++){
        return ht[i].get("X").toString();
    }
    return null;
    }

    public static String getJournalPublicationHistory(String isbn){
    String t3 = "publication_history('"+isbn.replace("\'", "")+"',X)";
    Query q3 = new Query(t3);

    Hashtable ht[] = q3.allSolutions();
    String t = "";
    for(int i = 0; i < ht.length; i++){
        return ht[i].get("X").toString();
    }
    return null;
    }

    public static String getJournalDiscipline(String isbn){
```

```
String t3 = "discipline('"+isbn.replace("\'", "")+"',X)";
Query q3 = new Query(t3);

Hashtable ht[] = q3.allSolutions();
String t = "";
for(int i = 0; i < ht.length; i++){
	return ht[i].get("X").toString();
}
return null;
}

public static String getJournalLanguage(String isbn){
String t3 = "language('"+isbn.replace("\'", "")+"',X)";
Query q3 = new Query(t3);

Hashtable ht[] = q3.allSolutions();
String t = "";
for(int i = 0; i < ht.length; i++){
	return ht[i].get("X").toString();
}
return null;
}

public static String getJournalImpactFactor(String isbn){
String t3 = "impact_factor('"+isbn.replace("\'", "")+"',X)";
Query q3 = new Query(t3);

Hashtable ht[] = q3.allSolutions();
String t = "";
for(int i = 0; i < ht.length; i++){
	return ht[i].get("X").toString();
}
return null;
}

public static String getJournalHomepage(String isbn){
String t3 = "journal_homepage('"+isbn.replace("\'", "")+"',X)";
Query q3 = new Query(t3);

Hashtable ht[] = q3.allSolutions();
String t = "";
for(int i = 0; i < ht.length; i++){
	return ht[i].get("X").toString();
}
return null;
}

public static String getJournalCitedHalfLife(String isbn){
String t3 = "cited_half-life('"+isbn.replace("\'", "")+"',X)";
Query q3 = new Query(t3);

Hashtable ht[] = q3.allSolutions();
String t = "";
for(int i = 0; i < ht.length; i++){
	return ht[i].get("X").toString();
}
return null;
```

```
    }

    public static String getJournalArticle(String isbn){
    String t3 = "articles('"+isbn.replace("\"", "")+"',X)";
    Query q3 = new Query(t3);

    Hashtable ht[] = q3.allSolutions();
    String t = "";
    for(int i = 0; i < ht.length; i++){
        return ht[i].get("X").toString();
    }
    return null;
    }


    public static String[] doRecommendation(String stdNum, String measureName, String interesting,
            String userkeyword, String recentJournalFlag,String numOfRecentSubmitFlag, String
considerHistory, String[] token, Hashtable explainList){

        System.out.println("Parameter = ");
        System.out.println("stdNum = "+ stdNum +
                "  measureName = "+measureName+
                "  interesting = "+interesting+
                "  userkeyword = "+userkeyword+
                "  recentJournalFlag = "+recentJournalFlag+
                "  numOfRecentSubmitFlag = "+numOfRecentSubmitFlag+
                "  considerHistory = "+considerHistory
                );




        ruleInitializeScoring();
        //getResult(explainList, "ruleInitializeScoring");
        interestingMeasureAccent(measureName);
        //getResult(explainList, "interestingMeasureAccent");
        journalScoreInitializeScoring();
        getResult(explainList, "While we aggregated all the journal metrics, ");
        majorWeightScoring(stdNum);
        getResult(explainList, "While we compared your major with this journal, ");
        interestingScoring(interesting);
        getResult(explainList, "While we compared your area of interest with this journal, ");
        keywordScoring(userkeyword);
        getResult(explainList, "While we compared your keywords with keywords of this journal, ");
        recentSubmitScoring(recentJournalFlag, stdNum);
        getResult(explainList, "While we assigned a score to journals where you submitted in the past, ");
        numOfrecentSubmitScoring(numOfRecentSubmitFlag, stdNum);
        getResult(explainList, "While we analyzed your research capability and assigned a score based on this, ");
        considerHistoryScoring(considerHistory);
        getResult(explainList, "While we analyzed historical value of each journal, ");
        abstractInput(token);

        return getResult(explainList, "Finally, ");

    }
```

```
    private static void initializeExplainList() {


    }

    public static void interestingMeasureAccent(String measureName){
    String t3 = "start0("+measureName.replace("\'", "")+").";
    Query q3 = new Query(t3);
    System.out.println( t3 + " " + (q3.hasMoreSolutions() ? "measure update success!" : " measure update fail!") );
    }

    public static void ruleInitializeScoring(){
          String t3 = "initialize.";
    Query q3 = new Query(t3);
    Hashtable ht[] = q3.allSolutions();
    System.out.println( t3 + " " + (ht.length != 0 ? "rule initialize success!" : "rule initialize fail!") );

       //System.out.println(ht.length);
   }


    public static void journalScoreInitializeScoring(){
          String t3 = "start1.";
    Query q3 = new Query(t3);
    Hashtable t[] = q3.allSolutions();

    System.out.println( t3 + " " + (t.length >= 1 ? "journal score initialize success!" : "journal score initialize fail!")
);
    }

    public static void majorWeightScoring(String stdNum){
          //String major = getUserMajor(stdNum);
          //int s = Integer.parseInt(stdNum);

          String t3 = "start2('"+stdNum+"').";
    Query q3 = new Query(t3);
    Hashtable t[] = q3.allSolutions();
    System.out.println( t3 + " " + (t.length >= 1 ? "major scoring success!" : "major scoring fail!") );
    }

    public static void interestingScoring(String interesting){
          String t3 = "start3('"+interesting.replace("\'", "")+"').";
    Query q3 = new Query(t3);
    Hashtable t[] = q3.allSolutions();
    System.out.println( t3 + " " + (t.length >= 1 ? "interestingScoring success!" : "interestingScoring fail!") );
    }

    public static void keywordScoring(String userkeyword){
          if(userkeyword.equals("")){
                System.out.println("Keyword is empty!");
                return;
          }
          String t3 = "start4('"+userkeyword.replace("\'", "")+"').";
    Query q3 = new Query(t3);
    Hashtable t[] = q3.allSolutions();
    System.out.println( t3 + " " + (t.length >= 1 ?  "keywordScoring success!" : "keywordScoring fail!") );
```

```
    }

    public static void recentSubmitScoring(String recentJournalFlag, String stdNum){
        String t3 = "start5("+recentJournalFlag+",'"+stdNum.replace("\'", "")+"').";
    Query q3 = new Query(t3);
    Hashtable t[] = q3.allSolutions();
    System.out.println( t3 + " " + (t.length >= 1 ? "recentSubmitScoring success!" : "recentSubmitScoring fail!") );
    }

    public static void numOfrecentSubmitScoring(String numOfRecentSubmitFlag, String stdNum){
        String t3 = "start6("+numOfRecentSubmitFlag+",'"+stdNum.replace("\'", "")+"').";
    Query q3 = new Query(t3);
    Hashtable t[] = q3.allSolutions();
    System.out.println( t3 + " " + (t.length >= 1 ?"numOfrecentSubmitScoring success!" :
"numOfrecentSubmitScoring fail!") );
    }

    public static void considerHistoryScoring(String considerHistory){
        String t3 = "start7('"+considerHistory+"').";
    Query q3 = new Query(t3);
    Hashtable t[] = q3.allSolutions();
    System.out.println( t3 + " " + (t.length >= 1 ?  "considerHistoryScoring success!" : "considerHistoryScoring
fail!") );
    }

    public static void abstractInput(String[] tokenList){

        for(int i = 0 ; i < tokenList.length; i++){
            String t3 = "start8("+tokenList[i]+")";
        Query q3 = new Query(t3);
        System.out.println( t3 + " " + (q3.hasMoreSolutions() ? "abstractInput success!" : "abstractInput fail!") );

        }
    }

    public static String[] getResult(Hashtable explainList, String question)  {

        String t4 = "result(X, Y).";
    Query q4 = new Query(t4);
    Hashtable<String, Double> ht = new Hashtable<String, Double>();
    int i = 0;
    System.out.println( "Each Solution is ");




    while (q4.hasMoreElements()){
        Hashtable ht3 = q4.nextSolution();
//      System.out.println("X = "+ht3.get("X")+"  Y = "+ht3.get("Y").toString());
        String k = ht3.get("Y").toString();
        ht.put(ht3.get("X").toString(), Double.parseDouble(k));
      if(!explainList.containsKey(ht3.get("X").toString())){
        explainList.put(ht3.get("X").toString(),"");
      }
      else{
//      System.out.println("explainList.get(ht3.get(X).toString() is "+explainList.get(ht3.get("X").toString()));
        //System.out.println("k is "+k);
```

```
            explainList.put(ht3.get("X").toString(), explainList.get(ht3.get("X").toString()).toString() + question
+"this journal got a score of " + Math.floor(Double.parseDouble(k) * Math.pow(10, 2) + 0.5) / Math.pow(10, 2)  +".
(accumulated) \n");
            System.out.println("EXPLAINLIST's Key = "+ht3.get("X") + " and value = "+
explainList.get(ht3.get("X").toString()).toString());
          }

      }

      return sortValue(ht);

      }

      public static String[] sortValue(Hashtable<String, Double> t){
                String strList[] = new String[5];
          //Transfer as List and sort it
          ArrayList<Map.Entry<?, Double>> l = new ArrayList(t.entrySet());
          Collections.sort(l, new Comparator<Map.Entry<?, Double>>(){

            public int compare(Map.Entry<?, Double> o1, Map.Entry<?, Double> o2) {
              return o1.getValue().compareTo(o2.getValue());
           }});

          System.out.println(l);

          for(int m = 0; m < 5; m++){
             strList[m] = l.get(l.size()+(-m-1)).toString();
             System.out.println(strList[m]);
          }
          return strList;
      }
}
```

```
//RecommendationWindow.java
package Frame;
import java.awt.Component;
import java.awt.Graphics;
import java.awt.Dimension;
import java.awt.Image;
import java.awt.Toolkit;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.beans.PropertyVetoException;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Hashtable;
import java.util.List;

import javax.speech.AudioException;
import javax.speech.EngineException;
import javax.speech.EngineStateError;
import javax.swing.BorderFactory;

import javax.swing.GroupLayout;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComponent;
import javax.swing.JDialog;
import javax.swing.JInternalFrame;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.JTextPane;
import javax.swing.LayoutStyle;
import javax.swing.SwingConstants;

import javax.swing.WindowConstants;
import javax.swing.border.LineBorder;

import org.apache.commons.mail.DefaultAuthenticator;
import org.apache.commons.mail.Email;
import org.apache.commons.mail.EmailException;
import org.apache.commons.mail.SimpleEmail;

import Database.databaseManager;
import Database.prologManager;
import EnglishSpeech.EnglishSpeaker;


/**
* This code was edited or generated using CloudGarden's Jigloo
* SWT/Swing GUI Builder, which is free for non-commercial
* use. If Jigloo is being used commercially (ie, by a corporation,
* company or business for any purpose whatever) then you
* should purchase a license for each developer using Jigloo.
* Please visit www.cloudgarden.com for details.
* Use of Jigloo implies acceptance of these licensing terms.
* A COMMERCIAL LICENSE HAS NOT BEEN PURCHASED FOR
* THIS MACHINE, SO JIGLOO OR THIS CODE CANNOT BE USED
* LEGALLY FOR ANY CORPORATE OR COMMERCIAL PURPOSE.
*/
```

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

 rivate JButton SendToMailBtn;

 rivate JButton sendButton;

 rivate JTextField mailAdrsField;

 rivate JDialog mailSendDialog;

 rivate JPanel JournalPic;
 rivate JPanel Fourth;
 rivate JButton fifthQuestion;
 rivate JButton fourthQuestion;
 rivate JButton thirdQuestion;
 rivate JButton secondQuestion;
 rivate JButton Voice;
 rivate JButton ScoreWhy1;
 rivate JInternalFrame JinternalUpperMain;
 rivate JInternalFrame jInternalFrame1;
 rivate JPanel Fifth;
 rivate JPanel Thrid;
 rivate JPanel Second;
 rivate String stdNum;
 rivate JTextPane FirstJournalInfo;
 rivate String[] recommendedList;
 rivate Hashtable<String, String> explainList;

 rivate JButton mailButton;

 rivate EnglishSpeech.EnglishSpeaker sp;

       public RecommendationWindow() {
 super();
 initGUI();


       public RecommendationWindow(String stdNum, String[] recommendedList, Hashtable<String, String>
explainList) {

 super();
 this.sp = new EnglishSpeaker();
 try {
        sp.init("kevin16");
 } catch (EngineException e) {
       // TODO Auto-generated catch block
       e.printStackTrace();
 } catch (AudioException e) {
       // TODO Auto-generated catch block
       e.printStackTrace();
 } catch (EngineStateError e) {
       // TODO Auto-generated catch block
       e.printStackTrace();
 } catch (PropertyVetoException e) {
       // TODO Auto-generated catch block
       e.printStackTrace();
 }
```

```
            this.recommendedList = recommendedList;
this.stdNum = stdNum;
this.explainList = explainList;
initGUI();


      }

rivate void initGUI() {
try {
       setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
      getContentPane().setBackground(new java.awt.Color(255,255,255));
      GroupLayout thisLayout = new GroupLayout((JComponent)getContentPane());
      //GroupLayout thisLayout = new GroupLayout((JComponent)getContentPane());
      //GroupLayout thisLayout = new GroupLayout((JComponent)getContentPane());
      getContentPane().setLayout(thisLayout);
      this.setTitle("Recommendation Window");
      {
                        SendToMailBtn = new JButton();
            SendToMailBtn.setText("SendToMail");
            SendToMailBtn.addMouseListener(new MouseAdapter() {
                  public void mouseClicked(MouseEvent evt) {
                        SendToMailBtnMouseClicked(evt);
                  }
             });
                  }
      thisLayout.setVerticalGroup(thisLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(SendToMailBtn, GroupLayout.PREFERRED_SIZE, GroupLayout.PREFERRED_SIZE,
GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(getJinternalUpperMain(), GroupLayout.PREFERRED_SIZE, 374,
GroupLayout.PREFERRED_SIZE)
            .addGap(0, 34, Short.MAX_VALUE)
            .addComponent(getJInternalFrame1(), GroupLayout.PREFERRED_SIZE, 242,
GroupLayout.PREFERRED_SIZE));
      thisLayout.setHorizontalGroup(thisLayout.createParallelGroup()
            .addGroup(GroupLayout.Alignment.LEADING, thisLayout.createParallelGroup()
               .addComponent(getJinternalUpperMain(), GroupLayout.Alignment.LEADING, 0, 905,
Short.MAX_VALUE)
               .addComponent(getJInternalFrame1(), GroupLayout.Alignment.LEADING, 0, 905,
Short.MAX_VALUE))
            .addGroup(GroupLayout.Alignment.LEADING, thisLayout.createSequentialGroup()
               .addGap(764)
               .addComponent(SendToMailBtn, GroupLayout.PREFERRED_SIZE, 119,
GroupLayout.PREFERRED_SIZE)
               .addContainerGap(22, Short.MAX_VALUE)));
      pack();
      this.setSize(921, 735);
} catch (Exception e) {
   //add your error handling code here
      e.printStackTrace();
}
      }
```

```
**
	 * @return
* @uml.property  name="mailSendDialog"
*/
	private JDialog getMailSendDialog() {
if(mailSendDialog == null) {
	mailSendDialog = new JDialog(this);
	GroupLayout mailSendDialogLayout = new GroupLayout((JComponent)mailSendDialog.getContentPane());
	mailSendDialog.getContentPane().setLayout(mailSendDialogLayout);
	mailSendDialog.setTitle("mailSendDialog");
	mailSendDialog.setSize(305, 88);
	mailSendDialogLayout.setHorizontalGroup(mailSendDialogLayout.createSequentialGroup()
		.addContainerGap()
		.addComponent(getMailAdrsField(), GroupLayout.PREFERRED_SIZE, 172,
GroupLayout.PREFERRED_SIZE)
		.addPreferredGap(LayoutStyle.ComponentPlacement.UNRELATED)
		.addComponent(getSendButton(), GroupLayout.PREFERRED_SIZE, 79,
GroupLayout.PREFERRED_SIZE)
		.addContainerGap(22, Short.MAX_VALUE));
	mailSendDialogLayout.setVerticalGroup(mailSendDialogLayout.createSequentialGroup()
		.addContainerGap(17, 17)
		.addGroup(mailSendDialogLayout.createParallelGroup(GroupLayout.Alignment.BASELINE)
		   .addComponent(getMailAdrsField(), GroupLayout.Alignment.BASELINE,
GroupLayout.PREFERRED_SIZE, 31, GroupLayout.PREFERRED_SIZE)
		   .addComponent(getSendButton(), GroupLayout.Alignment.BASELINE,
GroupLayout.PREFERRED_SIZE, 26, GroupLayout.PREFERRED_SIZE))
		.addContainerGap(19, 19));
	mailSendDialogLayout.linkSize(SwingConstants.VERTICAL, new Component[] {getSendButton(),
getMailAdrsField()});
}
		return mailSendDialog;


	/**
* @return
* @uml.property  name="mailAdrsField"
*/
	private JTextField getMailAdrsField() {
if(mailAdrsField == null) {
	mailAdrsField = new JTextField();
	mailAdrsField.setEditable(false);
	mailAdrsField.setText(prologManager.getUserEmail(stdNum).replace("\'", ""));
		}
		return mailAdrsField;
	}

	/**
	 * @return
	 * @uml.property  name="sendButton"
	 */
	private JButton getSendButton() {
		if(sendButton == null) {
			sendButton = new JButton();
			sendButton.setText("SEND");
```

```
            sendButton.addMouseListener(new MouseAdapter() {
                public void mouseClicked(MouseEvent evt) {
                    sendButtonMouseClicked(evt);
                }
            });
        }
        return sendButton;
    }

    private void sendButtonMouseClicked(MouseEvent evt) {
        try{

            if(mailAdrsField.getText().compareTo("")==0){
                return ;
            }

            Email email = new SimpleEmail();
          email.setSmtpPort(587);
          email.setAuthenticator(new DefaultAuthenticator("kse.javahomework",
              "wkqktnrwp"));
          email.setDebug(false);
          email.setHostName("smtp.gmail.com");
          email.setFrom("kse.javahomework@gmail.com");
          email.setSubject("[KSE643] Recommended Result ");

          String sendMessage = "Junghoon is cute >. <";

          email.setMsg(sendMessage);
          email.addTo(mailAdrsField.getText());
          email.setTLS(true);
          email.send();
        }
        catch (EmailException e) {
            e.printStackTrace();
        }
    }

    private void SendToMailBtnMouseClicked(MouseEvent evt) {
        JDialog ad = getMailSendDialog();
        ad.setLocationRelativeTo(null);
        ad.setVisible(true);
    }

    private JPanel getSecond() throws SQLException {
        if(Second == null) {

            final ImageIcon icon;
            byte[] b = null;

            b = databaseManager.readJournalImage(recommendedList[1].substring(1, 10));
            Image img = Toolkit.getDefaultToolkit().createImage(b);
            img = img.getScaledInstance(140, 164, Image.SCALE_DEFAULT);
            //ImageIcon imgc = new ImageIcon(ImageIO.read(new File("expert.jpg")));
            icon =new ImageIcon(img);
            Second = new JPanel(){
          public void paintComponent(Graphics g) {
            //  Approach 1: Dispaly image at at full size
```

```
            g.drawImage(icon.getImage(), 0, 0, null);
            //  Approach 2: Scale image to size of component
            // Dimension d = getSize();
            // g.drawImage(icon.getImage(), 0, 0, d.width, d.height, null);
            // Approach 3: Fix the image position in the scroll pane
            // Point p = scrollPane.getViewport().getViewPosition();
            // g.drawImage(icon.getImage(), p.x, p.y, null);
            setOpaque(false);
            super.paintComponent(g);
          }
        };
            Second.setBorder(new LineBorder(new java.awt.Color(0,0,0), 1, false));
            Second.addMouseListener(new MouseAdapter() {
                public void mouseClicked(MouseEvent evt) {
                    SecondMouseClicked(evt);
                }
            });

        }
        return Second;
    }

    private JPanel getThrid() throws SQLException {
        if(Thrid == null) {
            final ImageIcon icon;
            byte[] b;

            b = databaseManager.readJournalImage(recommendedList[2].substring(1, 10));

            Image img = Toolkit.getDefaultToolkit().createImage(b);
            img = img.getScaledInstance(140, 164, Image.SCALE_DEFAULT);
            //ImageIcon imgc = new ImageIcon(ImageIO.read(new File("expert.jpg")));
            icon =new ImageIcon(img);
            Thrid = new JPanel(){
          public void paintComponent(Graphics g) {
            //  Approach 1: Dispaly image at at full size
            g.drawImage(icon.getImage(), 0, 0, null);
            //  Approach 2: Scale image to size of component
            // Dimension d = getSize();
            // g.drawImage(icon.getImage(), 0, 0, d.width, d.height, null);
            // Approach 3: Fix the image position in the scroll pane
            // Point p = scrollPane.getViewport().getViewPosition();
            // g.drawImage(icon.getImage(), p.x, p.y, null);
            setOpaque(false);
            super.paintComponent(g);
          }
        };
            Thrid.setBorder(new LineBorder(new java.awt.Color(0,0,0), 1, false));
            Thrid.addMouseListener(new MouseAdapter() {
                public void mouseClicked(MouseEvent evt) {
                    ThridMouseClicked(evt);
                }
            });
        }
        return Thrid;
    }
```

```
private JPanel getJPanel1() throws SQLException {
	if(Fourth == null) {
		final ImageIcon icon;
		byte[] b;

		b = databaseManager.readJournalImage(recommendedList[3].substring(1, 10));

		Image img = Toolkit.getDefaultToolkit().createImage(b);
		img = img.getScaledInstance(140, 164, Image.SCALE_DEFAULT);
		//ImageIcon imgc = new ImageIcon(ImageIO.read(new File("expert.jpg")));
		icon =new ImageIcon(img);
		Fourth = new JPanel(){
	    public void paintComponent(Graphics g) {
		//  Approach 1: Dispaly image at at full size
		g.drawImage(icon.getImage(), 0, 0, null);
		//  Approach 2: Scale image to size of component
		// Dimension d = getSize();
		// g.drawImage(icon.getImage(), 0, 0, d.width, d.height, null);
		// Approach 3: Fix the image position in the scroll pane
		// Point p = scrollPane.getViewport().getViewPosition();
		// g.drawImage(icon.getImage(), p.x, p.y, null);
		setOpaque(false);
		super.paintComponent(g);
	    }
	};
		Fourth.setBorder(new LineBorder(new java.awt.Color(0,0,0), 1, false));
		Fourth.addMouseListener(new MouseAdapter() {
			public void mouseClicked(MouseEvent evt) {
				FourthMouseClicked(evt);
			}
		});
	}
	return Fourth;
}

private JPanel getJPanel1x() throws SQLException {
	if(Fifth == null) {
		final ImageIcon icon;
		byte[] b;

		b = databaseManager.readJournalImage(recommendedList[4].substring(1, 10));

		Image img = Toolkit.getDefaultToolkit().createImage(b);
		img = img.getScaledInstance(140, 164, Image.SCALE_DEFAULT);
		//ImageIcon imgc = new ImageIcon(ImageIO.read(new File("expert.jpg")));
		icon =new ImageIcon(img);
		Fifth = new JPanel(){
	    public void paintComponent(Graphics g) {
		//  Approach 1: Dispaly image at at full size
		g.drawImage(icon.getImage(), 0, 0, null);
		//  Approach 2: Scale image to size of component
		// Dimension d = getSize();
		// g.drawImage(icon.getImage(), 0, 0, d.width, d.height, null);
		// Approach 3: Fix the image position in the scroll pane
		// Point p = scrollPane.getViewport().getViewPosition();
		// g.drawImage(icon.getImage(), p.x, p.y, null);
		setOpaque(false);
```

```
                super.paintComponent(g);
            }
        };
            Fifth.setBorder(new LineBorder(new java.awt.Color(0,0,0), 1, false));
            Fifth.addMouseListener(new MouseAdapter() {
                public void mouseClicked(MouseEvent evt) {
                    FifthMouseClicked(evt);
                }
            });
        }
        return Fifth;
    }

    private JTextPane getFirstJournalInfo() {
        if(FirstJournalInfo == null) {


            String idx = recommendedList[0].substring(1, 10);
            String score = recommendedList[0].substring(12);

            FirstJournalInfo = new JTextPane();
            FirstJournalInfo.setText("Journal Name : \t"+prologManager.getJournalName(idx)+"\n"+
                    "Score : \t\t"+score+"\n"+
                    "Publisher : \t"+prologManager.getJournalPublisher(idx)+"\n"+
                    "Publication_history : \t"+prologManager.getJournalPublicationHistory(idx)+"\n"+
                    "Language : \t"+prologManager.getJournalLanguage(idx)+"\n"+
                    "Impact factor : \t"+prologManager.getJournalImpactFactor(idx)+"\n"+
                    "Homepage : \t"+prologManager.getJournalHomepage(idx)+"\n"+
                    "Cited_half_life : \t"+prologManager.getJournalCitedHalfLife(idx)+"\n"+
                    "Article : \t\t"+prologManager.getJournalArticle(idx)+"\n"
                    );
            FirstJournalInfo.setFont(new java.awt.Font("Arial",0,16));
            FirstJournalInfo.setBackground(new java.awt.Color(212,208,200));
            FirstJournalInfo.setEditable(false);

        }
        return FirstJournalInfo;
    }

    private JInternalFrame getJInternalFrame1() throws SQLException {
        if(jInternalFrame1 == null) {
            jInternalFrame1 = new JInternalFrame();
            GroupLayout jInternalFrame1Layout = new
GroupLayout((JComponent)jInternalFrame1.getContentPane());
            jInternalFrame1.getContentPane().setLayout(jInternalFrame1Layout);
            jInternalFrame1.setVisible(true);
            jInternalFrame1.setBounds(0, 455, 987, 242);
            jInternalFrame1.setTitle("Another Recommendation");
            try {
            } catch (Exception e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            jInternalFrame1Layout.setHorizontalGroup(jInternalFrame1Layout.createSequentialGroup()
                .addContainerGap(70, 70)
                .addGroup(jInternalFrame1Layout.createParallelGroup()
                    .addComponent(getSecond(), GroupLayout.Alignment.LEADING,
```

```
GroupLayout.PREFERRED_SIZE, 140, GroupLayout.PREFERRED_SIZE)
                        .addGroup(GroupLayout.Alignment.LEADING,
jInternalFrame1Layout.createSequentialGroup()
                            .addGap(38)
                            .addComponent(getSecondQuestion(), GroupLayout.PREFERRED_SIZE, 60,
GroupLayout.PREFERRED_SIZE)
                            .addGap(42)))
                    .addGap(73)
                    .addGroup(jInternalFrame1Layout.createParallelGroup()
                        .addComponent(getThrid(), GroupLayout.Alignment.LEADING,
GroupLayout.PREFERRED_SIZE, 140, GroupLayout.PREFERRED_SIZE)
                        .addGroup(GroupLayout.Alignment.LEADING,
jInternalFrame1Layout.createSequentialGroup()
                            .addGap(44)
                            .addComponent(getThirdQuestion(), GroupLayout.PREFERRED_SIZE, 60,
GroupLayout.PREFERRED_SIZE)
                            .addGap(36)))
                    .addGap(82)
                    .addGroup(jInternalFrame1Layout.createParallelGroup()
                        .addComponent(getJPanel1(), GroupLayout.Alignment.LEADING,
GroupLayout.PREFERRED_SIZE, 140, GroupLayout.PREFERRED_SIZE)
                        .addGroup(GroupLayout.Alignment.LEADING,
jInternalFrame1Layout.createSequentialGroup()
                            .addGap(39)
                            .addComponent(getFourthQuestion(), GroupLayout.PREFERRED_SIZE, 60,
GroupLayout.PREFERRED_SIZE)
                            .addGap(41)))
                    .addGap(88)
                    .addGroup(jInternalFrame1Layout.createParallelGroup()
                        .addGroup(jInternalFrame1Layout.createSequentialGroup()
                            .addComponent(getJPanel1x(), GroupLayout.PREFERRED_SIZE, 140,
GroupLayout.PREFERRED_SIZE)
                            .addGap(0, 0, Short.MAX_VALUE))
                        .addGroup(GroupLayout.Alignment.LEADING,
jInternalFrame1Layout.createSequentialGroup()
                            .addGap(47)
                            .addComponent(getFifthQuestion(), GroupLayout.PREFERRED_SIZE, 60,
GroupLayout.PREFERRED_SIZE)
                            .addGap(0, 33, Short.MAX_VALUE)))
                    .addContainerGap(30, 30));
                    jInternalFrame1Layout.setVerticalGroup(jInternalFrame1Layout.createSequentialGroup()
                    .addContainerGap()
                    .addGroup(jInternalFrame1Layout.createParallelGroup()
                        .addComponent(getSecond(), GroupLayout.Alignment.LEADING,
GroupLayout.PREFERRED_SIZE, 164, GroupLayout.PREFERRED_SIZE)
                        .addComponent(getThrid(), GroupLayout.Alignment.LEADING,
GroupLayout.PREFERRED_SIZE, 164, GroupLayout.PREFERRED_SIZE)
                        .addComponent(getJPanel1(), GroupLayout.Alignment.LEADING,
GroupLayout.PREFERRED_SIZE, 164, GroupLayout.PREFERRED_SIZE)
                        .addComponent(getJPanel1x(), GroupLayout.Alignment.LEADING,
GroupLayout.PREFERRED_SIZE, 164, GroupLayout.PREFERRED_SIZE))
                    .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(jInternalFrame1Layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
                        .addComponent(getSecondQuestion(), GroupLayout.Alignment.BASELINE,
GroupLayout.PREFERRED_SIZE, GroupLayout.PREFERRED_SIZE, GroupLayout.PREFERRED_SIZE)
                        .addComponent(getThirdQuestion(), GroupLayout.Alignment.BASELINE,
GroupLayout.PREFERRED_SIZE, GroupLayout.PREFERRED_SIZE, GroupLayout.PREFERRED_SIZE)
```

```
                        .addComponent(getFourthQuestion(), GroupLayout.Alignment.BASELINE,
GroupLayout.PREFERRED_SIZE, GroupLayout.PREFERRED_SIZE, GroupLayout.PREFERRED_SIZE)
                        .addComponent(getFifthQuestion(), GroupLayout.Alignment.BASELINE,
GroupLayout.PREFERRED_SIZE, GroupLayout.PREFERRED_SIZE, GroupLayout.PREFERRED_SIZE))
                    .addContainerGap());
        }
        return jInternalFrame1;
    }

    private JInternalFrame getJinternalUpperMain() throws SQLException {
        if(JinternalUpperMain == null) {
            JinternalUpperMain = new JInternalFrame();
            GroupLayout JinternalUpperMainLayout = new
GroupLayout((JComponent)JinternalUpperMain.getContentPane());
            JinternalUpperMain.getContentPane().setLayout(JinternalUpperMainLayout);
            JinternalUpperMain.setVisible(true);
            JinternalUpperMain.setBounds(0, 47, 905, 374);
            JinternalUpperMain.setTitle("Recommendation");
            {


                final ImageIcon icon;
                byte[] b = databaseManager.readJournalImage(recommendedList[0].substring(1, 10));
                Image img = Toolkit.getDefaultToolkit().createImage(b);
                img = img.getScaledInstance(216, 275, Image.SCALE_DEFAULT);
                //ImageIcon imgc = new ImageIcon(ImageIO.read(new File("expert.jpg")));
                icon =new ImageIcon(img);
                JournalPic = new JPanel(){
                    public void paintComponent(Graphics g) {
                        //  Approach 1: Dispaly image at at full size
                        g.drawImage(icon.getImage(), 0, 0, null);
                    //  Approach 2: Scale image to size of component
                    // Dimension d = getSize();
                    // g.drawImage(icon.getImage(), 0, 0, d.width, d.height, null);
                    // Approach 3: Fix the image position in the scroll pane
                    // Point p = scrollPane.getViewport().getViewPosition();
                    // g.drawImage(icon.getImage(), p.x, p.y, null);
                    setOpaque(false);
                    super.paintComponent(g);
                    }
                };









                //JournalPic = new JPanel();
                JournalPic.setBorder(new LineBorder(new java.awt.Color(0,0,0), 1, false));
            }
            JinternalUpperMainLayout.setHorizontalGroup(JinternalUpperMainLayout.createSequentialGroup()
                .addContainerGap(17, 17)
                .addGroup(JinternalUpperMainLayout.createParallelGroup()
                    .addComponent(JournalPic, GroupLayout.Alignment.LEADING,
```

```
GroupLayout.PREFERRED_SIZE, 216, GroupLayout.PREFERRED_SIZE)
                        .addGroup(GroupLayout.Alignment.LEADING,
JinternalUpperMainLayout.createSequentialGroup()
                            .addGap(130)
                            .addComponent(getVoice(), GroupLayout.PREFERRED_SIZE, 86,
GroupLayout.PREFERRED_SIZE)))
                    .addGap(50)
                    .addGroup(JinternalUpperMainLayout.createParallelGroup()
                        .addGroup(JinternalUpperMainLayout.createSequentialGroup()
                            .addComponent(getFirstJournalInfo(), GroupLayout.PREFERRED_SIZE, 565,
GroupLayout.PREFERRED_SIZE)
                            .addGap(0, 0, Short.MAX_VALUE))
                        .addGroup(GroupLayout.Alignment.LEADING,
JinternalUpperMainLayout.createSequentialGroup()
                            .addGap(206)
                            .addComponent(getScoreWhy1(), GroupLayout.PREFERRED_SIZE, 150,
GroupLayout.PREFERRED_SIZE)
                            .addGap(0, 209, Short.MAX_VALUE)))
                    .addContainerGap(55, 55));
            JinternalUpperMainLayout.setVerticalGroup(JinternalUpperMainLayout.createSequentialGroup()
                    .addContainerGap(25, 25)
                    .addGroup(JinternalUpperMainLayout.createParallelGroup()
                        .addComponent(getFirstJournalInfo(), GroupLayout.Alignment.LEADING,
GroupLayout.PREFERRED_SIZE, 282, GroupLayout.PREFERRED_SIZE)
                        .addGroup(GroupLayout.Alignment.LEADING,
JinternalUpperMainLayout.createSequentialGroup()
                            .addComponent(JournalPic, GroupLayout.PREFERRED_SIZE, 275,
GroupLayout.PREFERRED_SIZE)
                            .addGap(7)))
                    .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)

.addGroup(JinternalUpperMainLayout.createParallelGroup(GroupLayout.Alignment.BASELINE)
                        .addComponent(getScoreWhy1(), GroupLayout.Alignment.BASELINE,
GroupLayout.PREFERRED_SIZE, GroupLayout.PREFERRED_SIZE, GroupLayout.PREFERRED_SIZE)
                        .addComponent(getVoice(), GroupLayout.Alignment.BASELINE,
GroupLayout.PREFERRED_SIZE, 24, GroupLayout.PREFERRED_SIZE))
                    .addContainerGap());
        }
        return JinternalUpperMain;
    }

    private void SecondMouseClicked(MouseEvent evt) {
        Popup p = new Popup(recommendedList[1].substring(1, 10),
                recommendedList[1].substring(12));
        p.setLocationRelativeTo(null);
        p.setVisible(true);
    }

    private void ThridMouseClicked(MouseEvent evt) {
        Popup p = new Popup(recommendedList[2].substring(1, 10),recommendedList[2].substring(12));
        p.setLocationRelativeTo(null);
        p.setVisible(true);
    }

    private void FourthMouseClicked(MouseEvent evt) {
        Popup p = new Popup(recommendedList[3].substring(1, 10),recommendedList[3].substring(12));
        p.setLocationRelativeTo(null);
```

```
        p.setVisible(true);
    }

    private void FifthMouseClicked(MouseEvent evt) {
        Popup p = new Popup(recommendedList[4].substring(1, 10),recommendedList[4].substring(12));
        p.setLocationRelativeTo(null);
        p.setVisible(true);
    }

    private JButton getScoreWhy1() {
        if(ScoreWhy1 == null) {

            ScoreWhy1 = new JButton();
            ScoreWhy1.setSize(new Dimension(80, 25));
            ScoreWhy1.setText("Score Explanation");
            ScoreWhy1.addMouseListener(new MouseAdapter() {
                public void mouseClicked(MouseEvent evt) {
                    ScoreWhy1MouseClicked(evt);
                }
            });
        }
        return ScoreWhy1;
    }

    private void ScoreWhy1MouseClicked(MouseEvent evt) {
        Popup p = new Popup(recommendedList[0].substring(1, 10),recommendedList[0].substring(12),
explainList.get("\""+recommendedList[0].substring(1, 10)+"\"").toString());
        p.setLocationRelativeTo(null);
        p.setVisible(true);
    }

    private JButton getVoice() {
        if(Voice == null) {
            Voice = new JButton();
            Voice.setText("Voice");
            Voice.addMouseListener(new MouseAdapter() {
                public void mouseClicked(MouseEvent evt) {
                    VoiceMouseClicked(evt);
                }
            });
        }
        return Voice;
    }

    private void VoiceMouseClicked(MouseEvent evt) {
        String idx = recommendedList[0].substring(1, 10);

        try {
            sp.doSpeak("hello "+prologManager.getUserName(stdNum).replaceAll("_", "")+" our expert system
selected journal which name is "+prologManager.getJournalName(idx).replace("_", "")+". have a good luck ");
            //sp.terminate();
        } catch (EngineException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (AudioException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
```

```
        } catch (IllegalArgumentException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    private JButton getSecondQuestion() {
        if(secondQuestion == null) {
            secondQuestion = new JButton();
            secondQuestion.setText("?");
            secondQuestion.addMouseListener(new MouseAdapter() {
                public void mouseClicked(MouseEvent evt) {
                    secondQuestionMouseClicked(evt);
                }
            });
        }
        return secondQuestion;
    }

    private JButton getThirdQuestion() {
        if(thirdQuestion == null) {
            thirdQuestion = new JButton();
            thirdQuestion.setText("?");
            thirdQuestion.addMouseListener(new MouseAdapter() {
                public void mouseClicked(MouseEvent evt) {
                    thirdQuestionMouseClicked(evt);
                }
            });
        }
        return thirdQuestion;
    }

    private JButton getFourthQuestion() {
        if(fourthQuestion == null) {
            fourthQuestion = new JButton();
            fourthQuestion.setText("?");
            fourthQuestion.addMouseListener(new MouseAdapter() {
                public void mouseClicked(MouseEvent evt) {
                    fourthQuestionMouseClicked(evt);
                }
            });
        }
        return fourthQuestion;
    }

    private JButton getFifthQuestion() {
        if(fifthQuestion == null) {
            fifthQuestion = new JButton();
            fifthQuestion.setText("?");
            fifthQuestion.addMouseListener(new MouseAdapter() {
                public void mouseClicked(MouseEvent evt) {
                    fifthQuestionMouseClicked(evt);
                }
            });
```

```
            }
            return fifthQuestion;
        }

        private void fifthQuestionMouseClicked(MouseEvent evt) {
            Popup p = new Popup(recommendedList[4].substring(1, 10),recommendedList[4].substring(12),
explainList.get("\""+recommendedList[4].substring(1, 10)+"\"").toString());
            p.setLocationRelativeTo(null);
            p.setVisible(true);
        }

        private void fourthQuestionMouseClicked(MouseEvent evt) {
            Popup p = new Popup(recommendedList[3].substring(1, 10),recommendedList[3].substring(12),
explainList.get("\""+recommendedList[3].substring(1, 10)+"\"").toString());
            p.setLocationRelativeTo(null);
            p.setVisible(true);
        }

        private void thirdQuestionMouseClicked(MouseEvent evt) {
            Popup p = new Popup(recommendedList[2].substring(1, 10),recommendedList[2].substring(12),
explainList.get("\""+recommendedList[2].substring(1, 10)+"\"").toString());
            p.setLocationRelativeTo(null);
            p.setVisible(true);
        }

        private void secondQuestionMouseClicked(MouseEvent evt) {
            Popup p = new Popup(recommendedList[1].substring(1, 10),recommendedList[1].substring(12),
explainList.get("\""+recommendedList[1].substring(1, 10)+"\"").toString());
            p.setLocationRelativeTo(null);
            p.setVisible(true);
        }

}
```