JAVA

SOURCE CODE

Client Program

```
/*************************************************************************
**
*      Filename: ClientConnect.java                          *
*      Author: Tejas Dwarkaram                               *
*      Date: 20 July 2012, 10:10                            *
*      Operating System: Windows XP Professional                 *
*      Java Version: JDK 1.5 Update 9                        *
*      Description: Class to establish a connection to the server  *
**************************************************************************
*/

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;
import java.net.UnknownHostException;
import javax.swing.JOptionPane;

/**
 * Class used to establish a connection to the database server for the client side
application
 * @author Tejas Dwarkaram
 * @since JDK 1.5
 * @version 0.1 07/08/2012
 */
public class ClientConnect
{
   //creating the static variables that will be used globally
   static Socket clientSocket = null;
   static PrintWriter out = null;
   static BufferedReader in = null;
   /**
    * Creates a new instance of ClientConnect
    */
   public ClientConnect()
   {
      int portNumber = getPortNumber();
      try
      {
         //assigining a new client socket
         clientSocket = new Socket("127.0.0.1", portNumber);
         //creating a new print writer
         out = new PrintWriter(clientSocket.getOutputStream(), true);
         in = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
      }
      catch(UnknownHostException e)
      {
         System.err.println("Dont know about host 127.0.0.1");
         System.exit(1);
      }
      catch(IOException e)
      {
         JOptionPane.showMessageDialog(null, "No response from server! Please
ensure server is running, and port numbers correspond!", "Error",
JOptionPane.ERROR_MESSAGE);
         System.err.println("Couldnt get I/O");
         new ClientConnect();
      }
   }

   public int getPortNumber()
   {
      JOptionPane.showMessageDialog(null,"Hello");
```

```
        String port = JOptionPane.showInputDialog(null,"Enter port number to be
used for the server", "Port Requirement",
JOptionPane.INFORMATION_MESSAGE);
        int portNumber = Integer.parseInt(port);
        return portNumber;
    }

}
```

```
/********************************************************************
**
*      Filename: ClientGui.java                              *
*      Author: Tejas Dwarkaram                                *
*      Date: 20 July 2012, 09:10                             *
*      Operating System: Windows XP Professional                  *
*      Java Version: JDK 1.5 Update 9                          *
*      Description: Creating the Gui for the client application    *
*********************************************************************
*/

import java.io.IOException;
import java.util.regex.Pattern;
import javax.swing.JFrame;
import javax.swing.JOptionPane;

//importing the required packages

/**
 * Holds the constructors and methods required to create the Graphical User
Interface for the client side application
 * @author Tejas
 * @version 0.1 07/08/2012
 * @since JDK 1.5
 */
public class ClientGui extends javax.swing.JFrame
{
   /**
    * Creates new form ClientGui
    */
   public ClientGui()
   {
      //initializing the components
    initComponents();
}

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
// <editor-fold defaultstate="collapsed" desc=" Generated Code ">
private void initComponents() {
    headingLbl = new javax.swing.JLabel();
    allComponentPnl = new javax.swing.JPanel();
    allTbPne = new javax.swing.JTabbedPane();
    loginPnl = new javax.swing.JPanel();
    usernameFld = new javax.swing.JTextField();
    loginBtn = new javax.swing.JButton();
    infoLbl = new javax.swing.JLabel();
    passwordFld = new javax.swing.JPasswordField();
    addPnl = new javax.swing.JPanel();
    tabAdd = new javax.swing.JTabbedPane();
    addRecordPnl = new javax.swing.JPanel();
    addBtn = new javax.swing.JButton();
    movieidFld = new javax.swing.JTextField();
    movienameFld = new javax.swing.JTextField();
    genreidFld = new javax.swing.JTextField();
    movieDescripScrlPne = new javax.swing.JScrollPane();
    moviedescripTxtArea = new javax.swing.JTextArea();
    logoutBtn = new javax.swing.JButton();
    deleteRecordPnl = new javax.swing.JPanel();
    deleteBtn = new javax.swing.JButton();
    deleteMovieNameTxtFld = new javax.swing.JTextField();
    searchPnl = new javax.swing.JPanel();
    searchFld = new javax.swing.JTextField();
```

```
        searchBtn = new javax.swing.JButton();
        clientMnuBar = new javax.swing.JMenuBar();
        mainMnuItm = new javax.swing.JMenu();
        closeMnuItm = new javax.swing.JMenuItem();
        helpMnu = new javax.swing.JMenu();
        aboutMnuItm = new javax.swing.JMenuItem();


setDefaultCloseOperation(javax.swing.WindowConstants.DO_NOTHING_ON_CLO
SE);
        setTitle("Client Application");
        setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
        setName("clientFrame");
        addWindowListener(new java.awt.event.WindowAdapter() {
          public void windowClosing(java.awt.event.WindowEvent evt) {
            formWindowClosing(evt);
          }
        });

        headingLbl.setFont(new java.awt.Font("Monotype Corsiva", 0, 36));
        headingLbl.setForeground(new java.awt.Color(255, 0, 0));
        headingLbl.setText("Ins-Pirate-D Video Store");

        allComponentPnl.setBackground(new java.awt.Color(0, 0, 0));
        loginPnl.setBackground(new java.awt.Color(197, 198, 197));
        loginPnl.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0,
0, 0), 3, true));
        usernameFld.setText("Username");
        usernameFld.addMouseListener(new java.awt.event.MouseAdapter() {
          public void mouseClicked(java.awt.event.MouseEvent evt) {
            usernameFldMouseClicked(evt);
          }
          public void mouseExited(java.awt.event.MouseEvent evt) {
            usernameFldMouseExited(evt);
          }
        });

        loginBtn.setText("Login");
        loginBtn.addActionListener(new java.awt.event.ActionListener() {
          public void actionPerformed(java.awt.event.ActionEvent evt) {
            loginBtnActionPerformed(evt);
          }
        });

        infoLbl.setFont(new java.awt.Font("Tahoma", 1, 12));
        infoLbl.setText("Login as Administrator here");

        org.jdesktop.layout.GroupLayout loginPnlLayout = new
org.jdesktop.layout.GroupLayout(loginPnl);
        loginPnl.setLayout(loginPnlLayout);
        loginPnlLayout.setHorizontalGroup(

loginPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
          .add(org.jdesktop.layout.GroupLayout.TRAILING,
loginPnlLayout.createSequentialGroup()
            .addContainerGap(143, Short.MAX_VALUE)
            .add(infoLbl, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 181,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .add(165, 165, 165))
          .add(loginPnlLayout.createSequentialGroup()
            .add(157, 157, 157)

.add(loginPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILI
NG, false)
              .add(passwordFld)
              .add(usernameFld, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
154, Short.MAX_VALUE))
            .addContainerGap())
          .add(loginPnlLayout.createSequentialGroup()
            .add(182, 182, 182)
            .add(loginBtn, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 107,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(200, Short.MAX_VALUE))
        );
        loginPnlLayout.setVerticalGroup(

loginPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
          .add(loginPnlLayout.createSequentialGroup()
            .add(39, 39, 39)
            .add(infoLbl, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 29,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
            .add(usernameFld, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
28, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
            .add(passwordFld, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
29, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
            .add(loginBtn, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 35,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(137, Short.MAX_VALUE))
        );
        allTbPne.addTab("Login", loginPnl);

        addPnl.setBackground(new java.awt.Color(197, 198, 197));
        addPnl.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0,
0, 0), 3, true));
        addPnl.addComponentListener(new java.awt.event.ComponentAdapter() {
            public void componentShown(java.awt.event.ComponentEvent evt) {
                addPnlComponentShown(evt);
            }
        });

        tabAdd.setName("adminTab");
        addRecordPnl.setName("addTab");
        addRecordPnl.addComponentListener(new
java.awt.event.ComponentAdapter() {
            public void componentShown(java.awt.event.ComponentEvent evt) {
                addRecordPnlComponentShown(evt);
            }
        });

        addBtn.setText("Add");
        addBtn.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                addBtnActionPerformed(evt);
            }
        });

        movieidFld.setText("Movie ID");
        movieidFld.setEnabled(false);
        movieidFld.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                movieidFldMouseClicked(evt);
            }
        });

        movienameFld.setText("Movie Name");
        movienameFld.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                movienameFldMouseClicked(evt);
            }
            public void mouseExited(java.awt.event.MouseEvent evt) {
                movienameFldMouseExited(evt);
            }
        });

        genreidFld.setText("Genre ID");
        genreidFld.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                genreidFldMouseClicked(evt);
            }
            public void mouseExited(java.awt.event.MouseEvent evt) {
                genreidFldMouseExited(evt);
            }
        });

        moviedescripTxtArea.setColumns(20);
        moviedescripTxtArea.setRows(5);
        moviedescripTxtArea.setText("Movie Description\n");
        moviedescripTxtArea.addMouseListener(new
java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                moviedescripTxtAreaMouseClicked(evt);
            }
            public void mouseExited(java.awt.event.MouseEvent evt) {
                moviedescripTxtAreaMouseExited(evt);
            }
        });

        movieDescripScrlPne.setViewportView(moviedescripTxtArea);

        logoutBtn.setText("Logout");
        org.jdesktop.layout.GroupLayout addRecordPnlLayout = new
org.jdesktop.layout.GroupLayout(addRecordPnl);
        addRecordPnl.setLayout(addRecordPnlLayout);
        addRecordPnlLayout.setHorizontalGroup(

addRecordPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADI
NG)
            .add(addRecordPnlLayout.createSequentialGroup()
                .addContainerGap()

.add(addRecordPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.L
EADING)
                    .add(addRecordPnlLayout.createSequentialGroup()
                        .add(movieDescripScrlPne,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 332,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

.add(addRecordPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.T
RAILING)
                            .add(addBtn, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
91, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                            .add(logoutBtn, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
106, Short.MAX_VALUE)))
                    .add(addRecordPnlLayout.createSequentialGroup()

.add(addRecordPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.T
RAILING, false)
                            .add(org.jdesktop.layout.GroupLayout.LEADING, genreidFld)
                            .add(org.jdesktop.layout.GroupLayout.LEADING, movienameFld)
                            .add(org.jdesktop.layout.GroupLayout.LEADING, movieidFld,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 149, Short.MAX_VALUE))
                        .add(180, 180, 180)))
                .addContainerGap())
        );
        addRecordPnlLayout.setVerticalGroup(

addRecordPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADI
NG)
            .add(org.jdesktop.layout.GroupLayout.TRAILING,
addRecordPnlLayout.createSequentialGroup()
                .addContainerGap()
                .add(movieidFld, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 28,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                .add(movienameFld, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
30, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                .add(genreidFld, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 30,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, 7,
Short.MAX_VALUE)

.add(addRecordPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.L
EADING, false)
                    .add(addRecordPnlLayout.createSequentialGroup()
                        .add(addBtn, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 35,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .add(logoutBtn, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
34, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                    .add(movieDescripScrlPne,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 136,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                .addContainerGap())
```

```
    );
    tabAdd.addTab("Add Record", null, addRecordPnl, "Add a record");

    deleteBtn.setText("Delete");
    deleteBtn.setMaximumSize(new java.awt.Dimension(51, 23));
    deleteBtn.setMinimumSize(new java.awt.Dimension(51, 23));
    deleteBtn.setPreferredSize(new java.awt.Dimension(51, 23));
    deleteBtn.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        deleteBtnActionPerformed(evt);
      }
    });

    deleteMovieNameTxtFld.setText("Enter the name of the movie you wish to
delete");

    org.jdesktop.layout.GroupLayout deleteRecordPnlLayout = new
org.jdesktop.layout.GroupLayout(deleteRecordPnl);
    deleteRecordPnl.setLayout(deleteRecordPnlLayout);
    deleteRecordPnlLayout.setHorizontalGroup(

deleteRecordPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEA
DING)
      .add(deleteRecordPnlLayout.createSequentialGroup()
        .addContainerGap(98, Short.MAX_VALUE)
        .add(deleteMovieNameTxtFld,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 277,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
        .add(89, 89, 89))
      .add(deleteRecordPnlLayout.createSequentialGroup()
        .add(185, 185, 185)
        .add(deleteBtn, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 91,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
```

```
            .addContainerGap(188, Short.MAX_VALUE))
        );
        deleteRecordPnlLayout.setVerticalGroup(

deleteRecordPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEA
DING)
          .add(deleteRecordPnlLayout.createSequentialGroup()
            .add(89, 89, 89)
            .add(deleteMovieNameTxtFld,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 30,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .add(24, 24, 24)
            .add(deleteBtn, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 35,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(87, Short.MAX_VALUE))
        );
        tabAdd.addTab("Delete Record", null, deleteRecordPnl, "Delete a record");

        org.jdesktop.layout.GroupLayout addPnlLayout = new
org.jdesktop.layout.GroupLayout(addPnl);
        addPnl.setLayout(addPnlLayout);
        addPnlLayout.setHorizontalGroup(

addPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
          .add(addPnlLayout.createSequentialGroup()
            .addContainerGap()
            .add(tabAdd, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 469,
Short.MAX_VALUE)
            .addContainerGap())
        );
        addPnlLayout.setVerticalGroup(

addPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
```

```
          .add(addPnlLayout.createSequentialGroup()
             .addContainerGap()
             .add(tabAdd, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 293,
Short.MAX_VALUE)
             .addContainerGap())
       );
       allTbPne.addTab("Admin", addPnl);

       searchPnl.setBackground(new java.awt.Color(197, 198, 197));
       searchFld.setFont(new java.awt.Font("Tahoma", 0, 12));
       searchFld.setText("Enter your search word here");
       searchFld.addMouseListener(new java.awt.event.MouseAdapter() {
          public void mouseClicked(java.awt.event.MouseEvent evt) {
             searchFldMouseClicked(evt);
          }
          public void mouseExited(java.awt.event.MouseEvent evt) {
             searchFldMouseExited(evt);
          }
       });

       searchBtn.setText("Search");
       searchBtn.addActionListener(new java.awt.event.ActionListener() {
          public void actionPerformed(java.awt.event.ActionEvent evt) {
             searchBtnActionPerformed(evt);
          }
       });

       org.jdesktop.layout.GroupLayout searchPnlLayout = new
org.jdesktop.layout.GroupLayout(searchPnl);
       searchPnl.setLayout(searchPnlLayout);
       searchPnlLayout.setHorizontalGroup(

searchPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
```

```
            .add(searchPnlLayout.createSequentialGroup()
                .addContainerGap(161, Short.MAX_VALUE)
                .add(searchFld, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 184,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                .add(150, 150, 150))
            .add(searchPnlLayout.createSequentialGroup()
                .add(210, 210, 210)
                .add(searchBtn, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 75,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(210, Short.MAX_VALUE))
        );
        searchPnlLayout.setVerticalGroup(

searchPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
            .add(searchPnlLayout.createSequentialGroup()
                .add(119, 119, 119)
                .add(searchFld, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 28,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                .add(17, 17, 17)
                .add(searchBtn, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 27,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(130, Short.MAX_VALUE))
        );
        allTbPne.addTab("Search", searchPnl);

        org.jdesktop.layout.GroupLayout allComponentPnlLayout = new
org.jdesktop.layout.GroupLayout(allComponentPnl);
        allComponentPnl.setLayout(allComponentPnlLayout);
        allComponentPnlLayout.setHorizontalGroup(

allComponentPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LE
ADING)
            .add(allComponentPnlLayout.createSequentialGroup()
            .addContainerGap()
            .add(allTbPne, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 500,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(28, Short.MAX_VALUE))
      );
      allComponentPnlLayout.setVerticalGroup(

allComponentPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LE
ADING)
         .add(allComponentPnlLayout.createSequentialGroup()
            .addContainerGap()
            .add(allTbPne, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 349,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
      );

      mainMnuItm.setText("Menu");
      closeMnuItm.setText("Close");
      closeMnuItm.addActionListener(new java.awt.event.ActionListener() {
         public void actionPerformed(java.awt.event.ActionEvent evt) {
            closeMnuItmActionPerformed(evt);
         }
      });

      mainMnuItm.add(closeMnuItm);

      clientMnuBar.add(mainMnuItm);

      helpMnu.setText("Help");
      aboutMnuItm.setText("About");
      aboutMnuItm.addActionListener(new java.awt.event.ActionListener() {
         public void actionPerformed(java.awt.event.ActionEvent evt) {
            aboutMnuItmActionPerformed(evt);
          }
        });

        helpMnu.add(aboutMnuItm);

        clientMnuBar.add(helpMnu);

        setJMenuBar(clientMnuBar);

        org.jdesktop.layout.GroupLayout layout = new
org.jdesktop.layout.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
          layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
          .add(layout.createSequentialGroup()
            .addContainerGap(org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
              .add(org.jdesktop.layout.GroupLayout.TRAILING, allComponentPnl,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
              .add(org.jdesktop.layout.GroupLayout.TRAILING, headingLbl,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 334,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
            .addContainerGap())
        );
        layout.setVerticalGroup(
          layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
          .add(layout.createSequentialGroup()
            .addContainerGap()
            .add(headingLbl)
            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
            .add(allComponentPnl,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(25, Short.MAX_VALUE))
      );
      pack();
   }// </editor-fold>

   private void deleteBtnActionPerformed(java.awt.event.ActionEvent evt) {
      protocolSend(3);
   }

   //method called when the Add button is clicked
   private void addBtnActionPerformed(java.awt.event.ActionEvent evt) {
      if(Pattern.matches("[A-Z]{1}", movienameFld.getText()) &
Pattern.matches("\\d", genreidFld.getText()))
      {
         protocolSend(2);
      }
      else
      {
         JOptionPane.showMessageDialog(null,"Movie name should start with a
capital letter", "Error", JOptionPane.ERROR_MESSAGE);
      }
   }

   //method called if the mouse is clicked on this Text Field
   private void movieidFldMouseClicked(java.awt.event.MouseEvent evt) {
      if((movieidFld.getText()).equals("Movie ID"))
      {
}


        movieidFld.setText("");
    }
}

//method called if
private void movienameFldMouseClicked(java.awt.event.MouseEvent evt) {
    if((movienameFld.getText()).equals("Movie Name"))
    {
        movienameFld.setText("");
    }
}

private void movienameFldMouseExited(java.awt.event.MouseEvent evt) {
    if((movienameFld.getText()).equals(""))
    {
        movienameFld.setText("Movie Name");
    }
}

private void genreidFldMouseClicked(java.awt.event.MouseEvent evt) {
    if((genreidFld.getText()).equals("Genre ID"))
    {
        genreidFld.setText("");
    }
}

private void genreidFldMouseExited(java.awt.event.MouseEvent evt) {
    if(((genreidFld.getText()).equals("")) || ((genreidFld.getText()).equals(" ")))
    {
        genreidFld.setText("Genre ID");
    }
}
```

```
    private void moviedescripTxtAreaMouseClicked(java.awt.event.MouseEvent
evt) {
        if((moviedescripTxtArea.getText()).equals("Movie Description"))
        {
            moviedescripTxtArea.setText("");
        }
    }

    private void moviedescripTxtAreaMouseExited(java.awt.event.MouseEvent evt)
{
        if((moviedescripTxtArea.getText()).equals(" "))
        {
            moviedescripTxtArea.setText("Movie Description");
        }
    }

    private void addRecordPnlComponentShown(java.awt.event.ComponentEvent
evt) {
        protocolSend(4);
    }

    private void addPnlComponentShown(java.awt.event.ComponentEvent evt) {
        protocolSend(4);
    }

    //method to execute when mouse exits the search text field
    private void searchFldMouseExited(java.awt.event.MouseEvent evt) {
        if((searchFld.getText()).equals(""))
        {
            searchFld.setText("Enter your search word here");
        }
    }
```

```
//method to execute when the user clicks on the search text field
private void searchFldMouseClicked(java.awt.event.MouseEvent evt) {
   if((searchFld.getText()).equals("Enter your search word here"))
   {
      searchFld.setText("");
   }
}

//method to execute if the mouse is exited from the movie name
//method to execute when the login button is pressed
private void loginBtnActionPerformed(java.awt.event.ActionEvent evt) {
   protocolSend(5);
}

//method to execute if the menu item '1' is clicked
private void aboutMnuItmActionPerformed(java.awt.event.ActionEvent evt) {
   new HelpFrame().setVisible(true);
}

//creating a new object array to store the 'close' options
Object[] options = {"Minimize", "Exit", "Cancel"};

//method to execute when the frame is closing
private void formWindowClosing(java.awt.event.WindowEvent evt) {
   windowCheck();
}

//method to exeucte when the menu item 3 is clicked
private void closeMnuItmActionPerformed(java.awt.event.ActionEvent evt) {
   windowCheck();
}

//creating the method to run when the exit command is called
```

```
    public void windowCheck()
    {
        int opt = JOptionPane.showOptionDialog(null, "What would you like to do ?",
"You sure you want to leave ?", JOptionPane.YES_NO_CANCEL_OPTION,
JOptionPane.QUESTION_MESSAGE, null, options, options[2]);
        if(opt==1)
        {
            dispose();
        }
        else if(opt==0)
        {
          this.setState(JFrame.ICONIFIED);
        }
    }

    //method to execute if the mouse is exited from the username field
    private void usernameFldMouseExited(java.awt.event.MouseEvent evt) {
        if((usernameFld.getText()).equals(""))
        {
            usernameFld.setText("Username");
        }
    }

    //method to execute id the username text field is clicked
    private void usernameFldMouseClicked(java.awt.event.MouseEvent evt) {

if(((usernameFld.getText()).equals("Username"))||((usernameFld.getText()).equal
s("")))
        {
            usernameFld.setText("");
        }
    }
    //method to execute if the search button is clicked
    private void searchBtnActionPerformed(java.awt.event.ActionEvent evt) {
        protocolSend(1);
    }

    /**
     * Creates a new instance of the constructor
     * @param args the command line arguments
     */
    public static void main(String args[])
    {

        java.awt.EventQueue.invokeLater
        (
            new Runnable()
            {
                public void run()
                {
                    new ClientConnect();
                    new ClientGui().setVisible(true);
                }
            }
        );

    }

    /**
     * Method to convert the characters retrieved from the JPasswordField into
readable String characters
     * @param pass Character Array
     * @return String variable
     */
    public String convertPassword(char[] pass)
```

```
{
    String password ="";
    for(int x=0;x<pass.length;x++)
    {
        password = password + Character.toString(pass[x]);
    }
    return password;
}

/**
 * Method used to send the required
 * @param function int
 */
public void protocolSend(int function)
{

    String fromServer;
    String fromUser;
    try
    {
        if((fromServer = ClientConnect.in.readLine()) != null)
        {
            fromUser = "0";
            if(function == 1)
            {
                fromUser = "1" + searchFld.getText();
            }
            else if(function == 2)
            {
                fromUser = 6 + genreidFld.getText();
                if(fromServer.equalsIgnoreCase("yes"))
                {
                  fromUser = "2" + "'" + movienameFld.getText() + "','"  + "'" +
moviedescripTxtArea.getText() + "','" + "'" + genreidFld.getText() + "'" ;
               }
               else
               {
                  JOptionPane.showMessageDialog(null, "Genre ID does not exist!");
               }
            }
            else if(function == 4)
            {
               fromUser = "4";
               movieidFld.setText(fromServer);
            }
            else if(function == 5)
            {
               fromUser = "5" + usernameFld.getText() + "+" +
convertPassword(passwordFld.getPassword());
            }
            else if(function == 3)
            {
               fromUser = "3" + deleteMovieNameTxtFld.getText();
            }

            System.out.println(fromServer);

            if(function == 2 && fromServer.equals("yes"))
            {
               JOptionPane.showMessageDialog(null,"Record added successfully",
"Success", JOptionPane.INFORMATION_MESSAGE);
            }
            else if(function == 1)
            {
               if(!fromServer.equals("Welcome"))
```

```
          {
            if(!fromServer.equalsIgnoreCase("No"))
            {
              String search[] = fromServer.split("[+]");

              Object[][] data = new Object[search.length][4];
              String new_row[];

              for(int _row = 0; _row != search.length; _row++)
              {
                  if(!search[_row].equals(null))
                  {
                      new_row = search[_row].split("[,]");

                      for(int _cell = 0; _cell != new_row.length; _cell++)
                      {
                        data[_row][_cell] = new_row[_cell];
                      }
                  }
              }
              SearchFrame results = new SearchFrame();
              results.setVisible(true);
              results.showResults(data);
            }
            else
            {
              JOptionPane.showMessageDialog(null,"No Results found for that
search ! Please try again", "Error", JOptionPane.ERROR_MESSAGE);
            }
          }
           }
        else if(function==5)
        {
          if(fromServer.equalsIgnoreCase("success"))
          {
            JOptionPane.showMessageDialog(null, "Welcome " +
usernameFld.getText());
            usernameFld.setEnabled(false);
            passwordFld.setEnabled(false);
            loginBtn.setVisible(false);
            logoutBtn.setVisible(true);
          }
          else
          {
            JOptionPane.showMessageDialog(null, "Error in log in ! try again !",
"Error", JOptionPane.ERROR_MESSAGE);
          }
        }

        if(fromServer.equals("Bye"))
        {
          System.exit(1);
        }
        if(fromUser != null)
        {
          System.out.println("\nClient: " + fromUser);
          ClientConnect.out.println(fromUser);
        }
      }
    }
    catch(IOException ioe){}
  }

  // Variables declaration - do not modify
  private javax.swing.JMenuItem aboutMnuItm;
  static javax.swing.JButton addBtn;
```

```
    public static javax.swing.JPanel addPnl;
    private javax.swing.JPanel addRecordPnl;
    private javax.swing.JPanel allComponentPnl;
    public static javax.swing.JTabbedPane allTbPne;
    private javax.swing.JMenuBar clientMnuBar;
    private javax.swing.JMenuItem closeMnuItm;
    static javax.swing.JButton deleteBtn;
    private javax.swing.JTextField deleteMovieNameTxtFld;
    private javax.swing.JPanel deleteRecordPnl;
    private javax.swing.JTextField genreidFld;
    private javax.swing.JLabel headingLbl;
    private javax.swing.JMenu helpMnu;
    private javax.swing.JLabel infoLbl;
    javax.swing.JButton loginBtn;
    javax.swing.JPanel loginPnl;
    public static javax.swing.JButton logoutBtn;
    private javax.swing.JMenu mainMnuItm;
    private javax.swing.JScrollPane movieDescripScrlPne;
    private javax.swing.JTextArea moviedescripTxtArea;
    private javax.swing.JTextField movieidFld;
    private javax.swing.JTextField movienameFld;
    private javax.swing.JPasswordField passwordFld;
    javax.swing.JButton searchBtn;
    public static javax.swing.JTextField searchFld;
    private javax.swing.JPanel searchPnl;
    public static javax.swing.JTabbedPane tabAdd;
    private javax.swing.JTextField usernameFld;
    // End of variables declaration
}
```

```
/*****************************************************************
**
*      Filename: HelpFrame.java                              *
*      Author: Tejas Dwarkaram                               *
*      Date: 25 July 2012, 03:30                            *
*      Operating System: Windows XP Professional                 *
*      Java Version: JDK 1.5 Update 9                          *
*      Description: Class to create a new frame to show help on the*
*                program                               *
*****************************************************************
*/

import javax.swing.JOptionPane;

/**
 * Class used to create the interface to give the user help
 * @author Tejas Dwarkaram
 * @since JDK 1.5
 * @version 0.1 07/08/2012
 */
public class HelpFrame extends javax.swing.JFrame
{

   /**
    * Creates new form HelpFrame
    */
   public HelpFrame()
   {
      initComponents();
   }

   /** This method is called from within the constructor to
    * initialize the form.
```

```
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    // <editor-fold defaultstate="collapsed" desc=" Generated Code ">
    private void initComponents() {
        helpTbdPne = new javax.swing.JTabbedPane();
        aboutScrlPne = new javax.swing.JScrollPane();
        aboutTxtArea = new javax.swing.JTextArea();
        helpMnuBar = new javax.swing.JMenuBar();
        helpMnu = new javax.swing.JMenu();
        backMnuItm = new javax.swing.JMenuItem();


setDefaultCloseOperation(javax.swing.WindowConstants.DO_NOTHING_ON_CLO
SE);
        setTitle("Help ?");
        setName("helpFrame");
        setResizable(false);
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent evt) {
                formWindowClosing(evt);
            }
        });

        aboutTxtArea.setColumns(20);
        aboutTxtArea.setEditable(false);
        aboutTxtArea.setLineWrap(true);
        aboutTxtArea.setRows(5);
        aboutTxtArea.setText("This is a client application of a client/server
application created in Java. This application is for use in the Ins-pirate-d Video
Store. The client application enables a user to search for movies even if they dont
know the full title of the movie.\n\n\n All information is stored in a database, and
the database can only be accessed by the Administrator, who has administrator
```

```
privileges. Administrators have the ability to easily add or delete records of the
database. \n\n_________________________________\n\nCreated by Tejas
Dwarkaram \nDate : /07/12\n\nVersion 1.0.0.1.1\n\nLicense Agreement  - Still
Pending . . . . \n\n");
    aboutTxtArea.setWrapStyleWord(true);
    aboutTxtArea.setOpaque(false);
    aboutScrlPne.setViewportView(aboutTxtArea);

    helpTbdPne.addTab("About", aboutScrlPne);

    helpMnu.setText("Menu");
    backMnuItm.setText("Back");
    backMnuItm.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        backMnuItmActionPerformed(evt);
      }
    });

    helpMnu.add(backMnuItm);

    helpMnuBar.add(helpMnu);

    setJMenuBar(helpMnuBar);

    org.jdesktop.layout.GroupLayout layout = new
org.jdesktop.layout.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
      layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
      .add(org.jdesktop.layout.GroupLayout.TRAILING, helpTbdPne,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 374, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
            layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
            .add(org.jdesktop.layout.GroupLayout.TRAILING,
layout.createSequentialGroup()
                .add(helpTbdPne, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 474,
Short.MAX_VALUE)
                .addContainerGap())
        );
        pack();
    }// </editor-fold>

    //method to execute if the frame is closing
    private void formWindowClosing(java.awt.event.WindowEvent evt) {
        //creating the object to store an array of possible options
        Object[] options = {"Go back to app", "Cancel"};

        //displaying an option dialog to allow the user to choose an option and
storing the response in an int variable
        int opt = JOptionPane.showOptionDialog(null, "What would you like to do ?",
"You sure you want to leave ?", JOptionPane.YES_NO_OPTION,
JOptionPane.QUESTION_MESSAGE, null, options, options[1]);
        //checking if the option is Zero
        if(opt==0)
        {
            //if user chooses to close the application the process is disposed
            dispose();
        }
    }

    //method to be executed if the menu item is clicked
    private void backMnuItmActionPerformed(java.awt.event.ActionEvent evt) {
        //closing this frame if user clicks close menu item
        this.setVisible(false);
    }
    /**
     * Main method used to create the JFrame for the help interface
     * @param args the command line arguments
     */
    public static void main(String args[])
    {
        java.awt.EventQueue.invokeLater
        (
            new Runnable()
            {
                public void run()
                {
                    //setting the frame visible
                    new HelpFrame().setVisible(true);
                }
            }
        );
    }

    // Variables declaration - do not modify
    private javax.swing.JScrollPane aboutScrlPne;
    private static javax.swing.JTextArea aboutTxtArea;
    private javax.swing.JMenuItem backMnuItm;
    private javax.swing.JMenu helpMnu;
    private javax.swing.JMenuBar helpMnuBar;
    private javax.swing.JTabbedPane helpTbdPne;
    // End of variables declaration

}
```

```
/************************************************************************
**
*      Filename: searchFrame.java                              *
*      Author: Tejas Dwarkaram                                 *
*      Date: 14 August 2012, 03:30                             *
*      Operating System: Windows XP Professional               *
*      Java Version: JDK 1.5 Update 9                          *
*      Description: Class used to create a new JFrame to house the *
*              table for the search results                    *
*************************************************************************
*/

import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;

/**
 * Class used to display the results of a search
 * @author Tejas Dwarkaram
 * @since JDK 1.5
 * @version 0.1 07/08/2012
 */
public class SearchFrame extends javax.swing.JFrame
{
   static JPanel searchResultsPnl = new JPanel();

   /**
    * Creates new form SearchFrame
    */
   public SearchFrame()
   {
      initComponents();
      setContentPane(searchResultsPnl);
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    // <editor-fold defaultstate="collapsed" desc=" Generated Code ">
    private void initComponents() {


setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Search Results Frame");
        setName("searchFrame");
        org.jdesktop.layout.GroupLayout layout = new
org.jdesktop.layout.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
            .add(0, 400, Short.MAX_VALUE)
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
            .add(0, 300, Short.MAX_VALUE)
        );
        pack();
    }// </editor-fold>

    /**
     * Main method which runs a new instance of the constructor
     * @param args the command line arguments
     */
    public static void main(String args[])
```

```
   {
      java.awt.EventQueue.invokeLater
      (
         new Runnable()
         {
            public void run()
            {
               new SearchFrame().setVisible(true);
            }
         }
      );
   }


   /**
    * Method used to retrieve the results and display them on a table
    * @param data Object[][]
    */
   public static void showResults(Object[][] data)
   {
      String columnNames[] = {"Movie ID","Movie Name","Movie
Description","Genre ID" };
      JTable table = new JTable(data, columnNames);
      JScrollPane t00 = new JScrollPane();
      t00.getViewport().add(table);
      searchResultsPnl.add(t00);
   }

   // Variables declaration - do not modify
   // End of variables declaration

}
```

JAVA
SOURCE CODE

Server Program

```
/*****************************************************************
**
*	Filename: searchFrame.java				*
*	Author: Tejas Dwarkaram				*
*	Date: 19 July 2012, 08:44				*
*	Operating System: Windows XP Professional		*
*	Java Version: JDK 1.5 Update 9				*
*	Description: Class to establish a connection to the database*
******************************************************************
*/

package servergui;


import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;



/**
 * Class created to establish a connection for the server between the application and the database server
 * @author Tejas Dwarkaram
 * @since JDK 1.5
 * @version 0.1 07/08/2012
 */
public class ServerConnect
{

    static Connection con;
    /** Creates a new instance of ServerConnect */
    public ServerConnect()
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            //getting the connection to the database
            con = DriverManager.getConnection("jdbc:odbc:video_store");
        }
        catch(SQLException e)
        {
            System.out.println("Error" + e.toString());
        }
        catch(ClassNotFoundException e)

        {

           System.out.println("Error" + e.toString());

        }

    }


}
```

```
/*****************************************************************
**

*       Filename: Time.java                                    *

*       Author: Tejas Dwarkaram                                  *

*       Date: 12 August 2012, 03:23                              *

*       Operating System: Windows XP Professional                *

*       Java Version: JDK 1.5 Update 9                           *

*       Description: Class used to create the thread to run the    *

*               timer for the server                       *
******************************************************************
*/


package servergui;


/**

 * Class used to create the functions that run a thread to keep the timer running

 * @author Tejas Dwarkaram

 * @since JDK 1.5

 * @version 0.1 07/08/2012

 */

public class Time implements Runnable
```

```
{
    static int seconds=0;
    static int minutes=0;
    static int hours=0;
    static Thread runner;

    /** Creates a new instance of Time */
    public Time()
    {
        if(runner == null)
        {
            //starting the runner
            runner = new Thread(this);
            runner.start();
        }
    }

    /**
     * Method used to keep the thread running
     */
```

```
public void run()
{
    while(runner == Thread.currentThread())
    {
        try
        {
            //intiating the thread to sleep
            Thread.sleep(1000);
            seconds++;
        }
        catch(InterruptedException ie){}

        if(seconds<10)
        {
            ServerGui.secondsLbl.setText("0" + Integer.toString(seconds));
        }
        else
        {
            ServerGui.secondsLbl.setText(Integer.toString(seconds));
        }
```

```
if(seconds==59)
{
   minutes++;
   if(minutes<10)
   {
      ServerGui.minutesLbl.setText("0" + Integer.toString(minutes));
   }
   else
   {
      ServerGui.minutesLbl.setText(Integer.toString(minutes));
   }
   seconds=0;

   if(minutes==59)
   {
      hours++;
      if(hours<10)
      {
```

```
                        ServerGui.hoursLbl.setText("0" + Integer.toString(hours));
                    }
                    else
                    {
                        ServerGui.hoursLbl.setText(Integer.toString(hours));
                    }
                    minutes=0;
                }
            }
        }
    }

}
```

```
/************************************************************************
**
*	Filename: Session.java				*
*	Author: Tejas Dwarkaram				*
*	Date: 19 July 2012, 10:48			*
*	Operating System: Windows XP Professional		*
*	Java Version: JDK 1.5 Update 9			*
*	Description: Class used to create to establish new sessions *
*		for each client connection that is established *
*		and interpret the commands to be sent to the	*
*		protocol class				*
*************************************************************************
*/

package servergui;


import java.io.BufferedOutputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;


/**
 * Class created to run the sessions that need to be created when ever a client application is enabled
 * @author Tejas Dwarkaram
 * @since JDK 1.5
 * @version 0.1 07/08/2012
 */
public class Session implements Runnable
{
    Socket soc;
    BufferedReader br;
    PrintWriter pw;
    Thread runner;


    /**
     * Creates a new instance of Session
     * @param s Socket
     */
    public Session(Socket s)
    {
        soc = s;
        try
        {
            br = new BufferedReader(new InputStreamReader(soc.getInputStream()));
            pw = new PrintWriter(new BufferedOutputStream(soc.getOutputStream()),
true);
            pw.println("Welcome");
        }
        catch(IOException ioe)
        {
            System.out.println("G" + ioe.toString());
        }

        if(runner == null)
        {
            //starting the runner
            runner = new Thread(this);
            runner.start();
        }
    }
```

```
/**
 * Method used to keep the thread running for the sessions class
 */
public void run()
{
    try
    {
        while(runner == Thread.currentThread())
        {
            //reading the input into the buffered reader
            String input = br.readLine();
            if(input != null)
            {
                String output;
                output = "nothing";
                //sending the input to the protocol class
                if((input.substring(0,1)).equals("1"))
                {
                    output = Protocol.searchRecord(input);
```

```
}
else if((input.substring(0,1)).equals("2"))
{
   output = Protocol.addRecord(input);
}
else if((input.substring(0,1)).equals("3"))
{
   output = Protocol.deleteRecord(input);
}
else if((input.substring(0,1)).equals("4"))
{
   output = Protocol.getMovieId(input);
}
else if((input.substring(0,1)).equals("5"))
{
   output = Protocol.adminLogin(input);
}
else if((input.substring(0,1)).equals("6"))
{
   output = Protocol.checkGenre(input);
```

```
            }
            pw.println(output);
            if(output.equalsIgnoreCase("BYE!"))
            {
                //closing everything if the word BYE! is parsed
                runner = null;
                pw.close();
                br.close();
                soc.close();
            }
        }
        try
        {
            //intiating the thread to sleep
            Thread.sleep(100);
        }
        catch(InterruptedException ie){}

    }
}
```

```
        catch(IOException ioe)
        {
            System.out.println("F" + ioe.toString());
        }
    }

}
```

```
/*****************************************************************
**
*       Filename: ServerGui.java                                  *
*       Author: Tejas Dwarkaram                                    *
*       Date: 19 July 2012, 10:50                                 *
*       Operating System: Windows XP Professional                 *
*       Java Version: JDK 1.5 Update 9                            *
*       Description: Class used to create the GUI for the server  *
******************************************************************
*/

package servergui;

import java.awt.Cursor;
import javax.swing.JOptionPane;

/**
 * Class to hold all of the components required to create the Graphical User Interface for the server application
 * @author Tejas
 * @version 0.1 07/08/2012
 * @since JDK 1.5
 */
public class ServerGui extends javax.swing.JFrame
{
    /**
     * Creates new form ServerGui
     */
    public ServerGui()
    {
        initComponents();
    }

     /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    // <editor-fold defaultstate="collapsed" desc=" Generated Code ">
    private void initComponents() {
        jLabel1 = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
```

```
loginPnl = new javax.swing.JPanel();
userFld = new javax.swing.JTextField();
passFld = new javax.swing.JPasswordField();
usernameLbl = new javax.swing.JLabel();
passwordLbl = new javax.swing.JLabel();
serverBtnsPnl = new javax.swing.JPanel();
serverStartBtn = new javax.swing.JButton();
stopServerBtn = new javax.swing.JButton();
serverPnl = new javax.swing.JPanel();
serverAreaScrlPne = new javax.swing.JScrollPane();
serverArea = new javax.swing.JTextArea();
timePnl = new javax.swing.JPanel();
secondsLbl = new javax.swing.JLabel();
minutesLbl = new javax.swing.JLabel();
hoursLbl = new javax.swing.JLabel();
timeSemiLbl2 = new javax.swing.JLabel();
timeSemiLbl = new javax.swing.JLabel();
timerHeadingLbl = new javax.swing.JLabel();

jLabel1.setText("jLabel1");
```

```
    org.jdesktop.layout.GroupLayout jPanel2Layout = new
org.jdesktop.layout.GroupLayout(jPanel2);
    jPanel2.setLayout(jPanel2Layout);
    jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
      .add(0, 100, Short.MAX_VALUE)
    );
    jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
      .add(0, 100, Short.MAX_VALUE)
    );

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Server Application");
    setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
    setForeground(java.awt.Color.black);
    setName("serverFrame");
    setResizable(false);
    loginPnl.setBackground(new java.awt.Color(0, 0, 0));
```

```
        loginPnl.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(153, 153, 153), 2, true));

        loginPnl.setOpaque(false);

        userFld.addMouseListener(new java.awt.event.MouseAdapter() {

            public void mouseClicked(java.awt.event.MouseEvent evt) {

                userFldMouseClicked(evt);

            }

            public void mouseEntered(java.awt.event.MouseEvent evt) {

                userFldMouseEntered(evt);

            }

        });


        passFld.addMouseListener(new java.awt.event.MouseAdapter() {

            public void mouseEntered(java.awt.event.MouseEvent evt) {

                passFldMouseEntered(evt);

            }

        });


        usernameLbl.setFont(new java.awt.Font("Book Antiqua", 1, 12));

        usernameLbl.setText("Username");
```

```
    passwordLbl.setFont(new java.awt.Font("Bookman Old Style", 1, 12));

    passwordLbl.setText("Password");


    org.jdesktop.layout.GroupLayout loginPnlLayout = new
org.jdesktop.layout.GroupLayout(loginPnl);

    loginPnl.setLayout(loginPnlLayout);

    loginPnlLayout.setHorizontalGroup(

loginPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

      .add(loginPnlLayout.createSequentialGroup()

        .addContainerGap()

.add(loginPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADI
NG)

          .add(passwordLbl)

          .add(usernameLbl))

        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

.add(loginPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILI
NG)

          .add(passFld, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 171,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
```

```
          .add(userFld, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 171,
Short.MAX_VALUE))
        .addContainerGap())
    );
    loginPnlLayout.setVerticalGroup(

loginPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
      .add(org.jdesktop.layout.GroupLayout.TRAILING,
loginPnlLayout.createSequentialGroup()

.add(loginPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADI
NG)
          .add(loginPnlLayout.createSequentialGroup()
            .addContainerGap(25, Short.MAX_VALUE)
            .add(usernameLbl)
            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED))
          .add(loginPnlLayout.createSequentialGroup()
            .addContainerGap()
            .add(userFld, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 30,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)))
```

```
.add(loginPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
          .add(passwordLbl)
          .add(passFld, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 28, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
        .add(15, 15, 15))
    );

    serverBtnsPnl.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(204, 204, 204), 1, true));
    serverStartBtn.setText("Start Server");
    serverStartBtn.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        serverStartBtnActionPerformed(evt);
      }
    });
    serverStartBtn.addMouseListener(new java.awt.event.MouseAdapter() {
      public void mouseEntered(java.awt.event.MouseEvent evt) {
        serverStartBtnMouseEntered(evt);
      }
    });
    stopServerBtn.setText("Stop Server");
    stopServerBtn.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        stopServerBtnActionPerformed(evt);
      }
    });
    stopServerBtn.addMouseListener(new java.awt.event.MouseAdapter() {
      public void mouseEntered(java.awt.event.MouseEvent evt) {
        stopServerBtnMouseEntered(evt);
      }
    });

    org.jdesktop.layout.GroupLayout serverBtnsPnlLayout = new
org.jdesktop.layout.GroupLayout(serverBtnsPnl);
    serverBtnsPnl.setLayout(serverBtnsPnlLayout);
    serverBtnsPnlLayout.setHorizontalGroup(

serverBtnsPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADI
NG)
      .add(serverBtnsPnlLayout.createSequentialGroup()
        .addContainerGap()
.add(serverBtnsPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
          .add(serverStartBtn, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 121, Short.MAX_VALUE)
          .add(org.jdesktop.layout.GroupLayout.TRAILING, stopServerBtn, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 121, Short.MAX_VALUE))
        .addContainerGap())
    );
    serverBtnsPnlLayout.setVerticalGroup(

serverBtnsPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
      .add(serverBtnsPnlLayout.createSequentialGroup()
        .addContainerGap()
        .add(serverStartBtn, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 32, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
        .add(stopServerBtn, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 31, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(12, Short.MAX_VALUE))
    );
    serverArea.setColumns(20);
    serverArea.setEditable(false);
    serverArea.setRows(5);
    serverAreaScrlPne.setViewportView(serverArea);

    org.jdesktop.layout.GroupLayout serverPnlLayout = new
org.jdesktop.layout.GroupLayout(serverPnl);
    serverPnl.setLayout(serverPnlLayout);
    serverPnlLayout.setHorizontalGroup(

serverPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
      .add(org.jdesktop.layout.GroupLayout.TRAILING,
serverPnlLayout.createSequentialGroup()
        .addContainerGap()
        .add(serverAreaScrlPne, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
389, Short.MAX_VALUE)
        .addContainerGap())
    );
    serverPnlLayout.setVerticalGroup(

serverPnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
      .add(serverPnlLayout.createSequentialGroup()
        .addContainerGap()
        .add(serverAreaScrlPne, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
103, Short.MAX_VALUE)
        .addContainerGap())
    );

    secondsLbl.setFont(new java.awt.Font("Tahoma", 1, 14));
    secondsLbl.setText("00");

    minutesLbl.setFont(new java.awt.Font("Tahoma", 1, 14));
    minutesLbl.setText("00");

    hoursLbl.setFont(new java.awt.Font("Tahoma", 1, 14));
    hoursLbl.setText("00");

    timeSemiLbl2.setFont(new java.awt.Font("Tahoma", 1, 14));
    timeSemiLbl2.setText(":");

    timeSemiLbl.setFont(new java.awt.Font("Tahoma", 1, 14));
    timeSemiLbl.setText(":");
```

```
    timerHeadingLbl.setFont(new java.awt.Font("Bookman Old Style", 1, 14));

    timerHeadingLbl.setText("Server Up-Time");


    org.jdesktop.layout.GroupLayout timePnlLayout = new
org.jdesktop.layout.GroupLayout(timePnl);

    timePnl.setLayout(timePnlLayout);

    timePnlLayout.setHorizontalGroup(


timePnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

      .add(timePnlLayout.createSequentialGroup()

        .addContainerGap(org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)


.add(timePnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADIN
G)

          .add(org.jdesktop.layout.GroupLayout.TRAILING,
timePnlLayout.createSequentialGroup()

            .add(hoursLbl, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
21, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)

            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

            .add(timeSemiLbl)

            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
            .add(minutesLbl,  org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
18, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)

            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

            .add(timeSemiLbl2)

            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

            .add(secondsLbl,  org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
18, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))

          .add(org.jdesktop.layout.GroupLayout.TRAILING, timerHeadingLbl))

        .addContainerGap())

    );

    timePnlLayout.setVerticalGroup(


timePnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

      .add(org.jdesktop.layout.GroupLayout.TRAILING,
timePnlLayout.createSequentialGroup()

        .addContainerGap()

        .add(timerHeadingLbl)

        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)


.add(timePnlLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELI
NE)
```

```
          .add(secondsLbl, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
23, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
          .add(timeSemiLbl2)
          .add(minutesLbl, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
22, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
          .add(timeSemiLbl)
          .add(hoursLbl, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 20,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
        .addContainerGap())
    );

    org.jdesktop.layout.GroupLayout layout = new
org.jdesktop.layout.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
      layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
      .add(layout.createSequentialGroup()
        .addContainerGap()

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING,
false)
```

```
        .add(org.jdesktop.layout.GroupLayout.LEADING,          serverPnl,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .add(org.jdesktop.layout.GroupLayout.LEADING,
layout.createSequentialGroup()

          .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)


.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING)

            .add(timePnl,   org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)

            .add(layout.createSequentialGroup()

              .add(loginPnl,   org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

              .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

              .add(serverBtnsPnl,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)))))

      .addContainerGap(org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

   );

   layout.setVerticalGroup(

     layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

     .add(layout.createSequentialGroup()
}

                .addContainerGap()

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING, false)

                    .add(loginPnl,              org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                    .add(serverBtnsPnl,     org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

                .add(timePnl,              org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)

                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

                .add(serverPnl,              org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

        );

        pack();

    }// </editor-fold>


    //method to execute if the button is clicked

    private void stopServerBtnActionPerformed(java.awt.event.ActionEvent evt) {

        Server.startServer(userFld.getText(),
convertPassword(passFld.getPassword()), 2);

    }
```

```
//method to execute if the mouse moves over the password field
private void passFldMouseEntered(java.awt.event.MouseEvent evt) {
   passFld.setCursor(new Cursor(Cursor.HAND_CURSOR));
}

//method to execute if the mouse is moved over the user text field
private void userFldMouseEntered(java.awt.event.MouseEvent evt) {
   userFld.setCursor(new Cursor(Cursor.HAND_CURSOR));
}

//method to execute if the mouse is entered over the stop server button
private void stopServerBtnMouseEntered(java.awt.event.MouseEvent evt) {
   stopServerBtn.setCursor(new Cursor(Cursor.HAND_CURSOR));
}

//method to execute if the mouse moves over the server start button
private void serverStartBtnMouseEntered(java.awt.event.MouseEvent evt) {
   serverStartBtn.setCursor(new Cursor(Cursor.HAND_CURSOR));
}
```

```
  //method to execute if the button is clicked
  private void serverStartBtnActionPerformed(java.awt.event.ActionEvent evt) {

Server.startServer(userFld.getText(),convertPassword(passFld.getPassword()), 1);
  }

  //method to be executed if the username field is clicked
  private void userFldMouseClicked(java.awt.event.MouseEvent evt) {
    if(((userFld.getText()).equals("Username"))||((userFld.getText()).equals("")))
    {
      userFld.setText("");
    }
  }

  /**
   * Main method to create new instance of the constructor of the ServerGui class
   * @param args the command line arguments
   */
  public static void main(String args[])
```

```
    {
        java.awt.EventQueue.invokeLater
        (
            new Runnable()
            {
                public void run()
                {
                    new ServerGui().setVisible(true);
                    serverPnl.setVisible(false);
                    stopServerBtn.setVisible(false);
                    timePnl.setVisible(false);
                    JOptionPane.showMessageDialog(null, "Please enter your login details above \nOnly Admin can start/stop server", "Welcome", JOptionPane.INFORMATION_MESSAGE);
                }
            }
        );
    }

    /**
     * Method to convert the array of Characters found in the JPasswordField into String characters
```

```
 * @param pass Character Array
 * @return String
 */
public String convertPassword(char[] pass)
{
    String password ="";
    for(int x=0;x<pass.length;x++)
    {
        password = password + Character.toString(pass[x]);
    }
    return password;
}

// Variables declaration - do not modify
public static javax.swing.JLabel hoursLbl;
private javax.swing.JLabel jLabel1;
private javax.swing.JPanel jPanel2;
public static javax.swing.JPanel loginPnl;
public static javax.swing.JLabel minutesLbl;
public static javax.swing.JPasswordField passFld;
```

```
    private javax.swing.JLabel passwordLbl;

    public static javax.swing.JLabel secondsLbl;

    public static javax.swing.JTextArea serverArea;

    private javax.swing.JScrollPane serverAreaScrlPne;

    private javax.swing.JPanel serverBtnsPnl;

    public static javax.swing.JPanel serverPnl;

    public static javax.swing.JButton serverStartBtn;

    public static javax.swing.JButton stopServerBtn;

    public static javax.swing.JPanel timePnl;

    public static javax.swing.JLabel timeSemiLbl;

    public static javax.swing.JLabel timeSemiLbl2;

    private javax.swing.JLabel timerHeadingLbl;

    public static javax.swing.JTextField userFld;

    private javax.swing.JLabel usernameLbl;

    // End of variables declaration


}
```

```
/******************************************************************
**
*	Filename: Server.java				*
*	Author: Tejas Dwarkaram				*
*	Date: 19 July 2012, 10:48				*
*	Operating System: Windows XP Professional		*
*	Java Version: JDK 1.5 Update 9			*
*	Description: Class used to create the connections for the	*
*		server					*
*******************************************************************
*/


package servergui;


import java.io.IOException;
import java.net.ServerSocket;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JOptionPane;

/**
 * Class used to create the server side processes, and create a network connection
 * @author Tejas Dwarkaram
 * @since JDK 1.5
 * @version 0.1 07/08/2012
 */
public class Server implements Runnable
{
    /** Creates a new instance of Server */
    static ServerSocket ss;
    static boolean listening;
    Thread runner;
    static Date start = new Date();
    static SimpleDateFormat startTime = new SimpleDateFormat("HH:mm:ss");
    static String stTime = startTime.format(start.getTime());

    /**
     * Used to create the constructor for the server class
 */
public Server()
{
    String port = JOptionPane.showInputDialog(null,"Enter port number to be used for the server", "PortSettings", JOptionPane.INFORMATION_MESSAGE);

    int portNumber = Integer.parseInt(port);
    while(portNumber == 0)
    {
        JOptionPane.showMessageDialog(null,"An error has occured, please enter a valid port number of 4 digits!", "Error on port number", JOptionPane.ERROR_MESSAGE);
        port = JOptionPane.showInputDialog(null,"Enter port number to be used for the server", "PortSettings", JOptionPane.INFORMATION_MESSAGE);
        portNumber = Integer.parseInt(port);
    }
    try
    {
        //creating a new server socket
        ss = new ServerSocket(portNumber);
        ServerGui.serverArea.append("Server connection started.\nTime started: " + stTime + "\nAdministrator Name - " + ServerGui.userFld.getText() + "\nServer at port: " + portNumber + "\n*-*-*-*-*-*-*-*-*-*-*-*-*\n ");
        listening = true;

        new Time();

    }

    catch(IOException e)

    {

        System.out.println("T"+ e.toString());

        System.exit(1);

    }

    if(runner == null)

    {

        //starting the runner

        runner = new Thread(this);

        runner.start();

    }

}


/**

 * Method used to keep the thread running

 */

public void run()
```

```
{
  while(runner == Thread.currentThread())
  {
    while(listening)
    {
      try
      {
        //trying to create a new session
        new Session(ss.accept());
      }
      catch(IOException e)
      {
        System.out.println("C" + e.toString());
      }
    }
    try
    {
      //intiating the thread to sleep
      Thread.sleep(100);
    }
```

```
        catch(InterruptedException ie){}
    }
}

/**
 * Method used to start the server
 * @param logonName String
 * @param pass String
 * @param option int
 */
public static void startServer(String logonName, String pass, int option)
{
    int x=1;
    boolean log = false;
    ServerConnect connect = new ServerConnect();
    if(option==1)
    {
        try
        {
            //creating a statement to be used
            Statement st = connect.con.createStatement();
            //creating the statement to select all of the data from the artists table
            ResultSet rec = st.executeQuery("SELECT * FROM users");
            while(rec.next() && x==1)
            {
               if((logonName).equals(rec.getString(2))                                  &&
(pass).equals(rec.getString(3)))
               {
                  log = true;
                  x=0;
               }
            }
            st.close();
            if(log==true)
            {
               ServerGui.serverPnl.setVisible(true);
               ServerGui.serverArea.setVisible(true);
               ServerGui.serverStartBtn.setVisible(false);
               ServerGui.stopServerBtn.setVisible(true);
               ServerGui.userFld.setText("");
               ServerGui.passFld.setText("");
               ServerGui.timePnl.setVisible(true);
               new Server();
            }
            else
            {
               JOptionPane.showMessageDialog(null, "Error in login name or password, please try again", "Error", JOptionPane.ERROR_MESSAGE);
               ServerGui.userFld.setText("");
               ServerGui.passFld.setText("");
            }
         }
         catch(SQLException e)
         {
            System.out.println("Error" + e.toString());
         }
      }
      else
      {
         try
         {
            //creating a statement to be used
```

```
    Statement st = connect.con.createStatement();

    //creating the statement to select all of the data from the artists table

    ResultSet rec = st.executeQuery("SELECT * FROM users");

    while(rec.next() && x==1)

    {

      if((logonName).equals(rec.getString(2)) &&
(pass).equals(rec.getString(3)))

      {

        log = true;

        x=0;

      }

    }

    st.close();

    if(log==true)

    {

      try

      {

        Server.ss.close();

        Server.listening = false;

        ServerGui.serverArea.append("Server connection stopped.\nTime
ended: " + stTime + "\nServer Up-Time - " + ServerGui.hoursLbl.getText() + ":" +
ServerGui.minutesLbl.getText() + ":" + ServerGui.secondsLbl.getText() + "\n*-*-*-
*-*-*-*-*-*-*-*-*-*\n ");

        ServerGui.userFld.setText("");

        ServerGui.passFld.setText("");

        ServerGui.serverStartBtn.setVisible(true);

        ServerGui.stopServerBtn.setVisible(false);



        Time.runner = null;

        Time.seconds=0;

        Time.minutes=0;

        Time.hours=0;

        ServerGui.hoursLbl.setText("00");

        ServerGui.secondsLbl.setText("00");

        ServerGui.minutesLbl.setText("00");



        ServerGui.timePnl.setVisible(false);

      }

      catch(IOException ioe){}

    }

    else

    {
```

```
                    JOptionPane.showMessageDialog(null, "Please enter your administrator details", "You have left some blanks", JOptionPane.INFORMATION_MESSAGE);

                    ServerGui.userFld.setText("");

                    ServerGui.passFld.setText("");

                }

            }

            catch(SQLException e)

            {

                System.out.println("Error" + e.toString());

            }

        }

    }

}
```

```
/****************************************************************
**

*       Filename: Protocol.java                         *

*       Author: Tejas Dwarkaram                          *

*       Date: 19 July 2012, 10:48                       *

*       Operating System: Windows XP Professional             *

*       Java Version: JDK 1.5 Update 9                    *

*       Description: Creating the class to execute commands        *

*                 recieved from the client application          *

*****************************************************************
*/


package servergui;


import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;


/**

 * Class created to handle various requests that are made on the client side
application

 * @author Tejas Dwarkaram
 * @since JDK 1.5
 * @version 0.1 07/08/2012
 */
public class Protocol
{
    /**
     * Method used to do a search of the database
     * @param input String
     * @return String
     */
    public static String searchRecord(String input)
    {
        input = input.substring(1);
        try
        {
            int x=0;
            Statement st = ServerConnect.con.createStatement();
            ResultSet rec = st.executeQuery("SELECT * FROM movie WHERE
movie_name LIKE '%" + input + "%'");

            String data = "";
```

```
while(rec.next())
{
   if(x==0)
   {
      String str = rec.getString(1) + "," + rec.getString(2) + "," + rec.getString(3) + "," + rec.getString(4);
      data = data.concat(str);
   }
   else if(x>0)
   {
      String str = "+" + rec.getString(1) + "," + rec.getString(2) + "," + rec.getString(3) + "," + rec.getString(4);
      data = data.concat(str);
   }
   x++;
}
if(data.equals(""))
{
   return "No";
}
```

```
        else
        {
            return data;
        }
    }
    catch(SQLException sql) {}
    return "No";
}

 /**
 * Method used to add a record into the database
 * @param data String
 * @return String
 */
public static String addRecord(String data)
{
    data = data.substring(1);
    try
    {
        int addRows;
```

```
            Statement st = ServerConnect.con.createStatement();
            addRows = st.executeUpdate("INSERT INTO movie (movie_name, movie_description, genre_id) VALUES (" + data + ")");
        }
        catch(SQLException sql)
        {
            return "SQL Exception" + sql.toString();
        }
        return "yes";
    }

    /**
     * Method used to delete a record from the database
     * @param input String
     * @return String
     */
    public static String deleteRecord(String input)
    {
        try
        {
            String h = "e" ;
            Statement st = ServerConnect.con.createStatement();

            ResultSet rec = st.executeQuery("SELECT * FROM movie WHERE movie_name LIKE '%" + input + "%'");

            while(rec.next())

            {

                String str = rec.getString(1) + "," + rec.getString(2) + "," + rec.getString(3) + "," + rec.getString(4);

                h = h.concat(str);

            }

            return h;

        }

        catch(SQLException sql)

        {

            return "SQL Exception" + sql.toString();

        }

    }


    /**

     * Method used to retrieve the next available id from the table

     * @param input String

     * @return String
```

```
 */
public static String getMovieId(String input)
{
    try
    {
        int id=0;
        Statement st = ServerConnect.con.createStatement();
        ResultSet rec = st.executeQuery("SELECT * FROM movie");
        while(rec.next())
        {
            id = Integer.parseInt(rec.getString(1));
        }
        return Integer.toString(id + 1);
    }
    catch(SQLException sql)
    {
        return "SQL Exception" + sql.toString();
    }
}
```

```
/**
 * Method used to check if the genre of a movie already exists
 * @param input String
 * @return String
 */
public static String checkGenre(String input)
{
    input = input.substring(1);
    int x=1;
    boolean log = false;
    try
    {
        //creating a statement to be used
        Statement st = ServerConnect.con.createStatement();
        //creating the statement to select all of the data from the artists table
        ResultSet rec = st.executeQuery("SELECT * FROM genre");
        while(rec.next() && x==1)
        {
            if(input.equals(rec.getString(1)))
            {
                log = true;
                x=0;
            }
        }
        st.close();
        if(log==true)
        {
            return "yes";
        }
        else
        {
            return "no";
        }
    }
    catch(SQLException e)
    {
        System.out.println("Error" + e.toString());
    }
    return "gelo";
}
```

```
    /**
     * Method used to verify that the user trying to access the admin panel is an administratorm and has corresponding username and password
     * @param input String
     * @return String
     */
    public static String adminLogin(String input)
    {
        input = input.substring(1);
        String[] login = input.split("[+]");
        String userName = login[0];
        String password = login[1];
        int x=1;
        boolean log = false;
        try
        {
            //creating a statement to be used
            Statement st = ServerConnect.con.createStatement();
            //creating the statement to select all of the data from the artists table
            ResultSet rec = st.executeQuery("SELECT * FROM users");
            while(rec.next() && x==1)

            {

               if(userName.equals(rec.getString(2))                                              &&
      (password.equals(rec.getString(3))))

               {

                  log = true;

                  x=0;

               }

            }

            st.close();

            if(log==true)

            {

               return "success";

            }

            else

            {

               return "fail";

            }

        }

        catch(SQLException e)

        {
           System.out.println("Error" + e.toString());
        }
        return "gelo";
     }
      }
```

SOURCE CODE

MSSQL

Database code

```sql
/*********************************************************************
*       Filename: Create_Database_Video_Store
      *
*       Author: Tejas Dwarkaram                                      *
*       Date: 20 July 2012
            *
*       Description: Creating the database for the video store,     *
*                             as well as adding the tables,and inserting the
*
*                             sample data
                  *
********************************************************************/
USE master
GO

IF EXISTS(SELECT name FROM master.dbo.sysdatabases
            WHERE name = 'video_store')
            DROP DATABASE video_store
            GO

--Creating the database
CREATE DATABASE video_store
ON PRIMARY
(
      NAME = 'video_store_data',
      FILENAME = 'C:\Video_Store_NS2012-0340\video_store.mdf',
      SIZE = 10,
      MAXSIZE = UNLIMITED,
      FILEGROWTH = 2
)
--creating the log file for the database
LOG ON
(

      NAME = 'video_store_log',
      FILENAME = 'C:\Video_Store_NS2012-0340\video_store_log.ldf',
      SIZE =10,
      MAXSIZE = UNLIMITED,
      FILEGROWTH = 2
)
GO

USE video_store
GO

--creating all of the tables for the database
CREATE TABLE genre
(
      genre_id int NOT NULL IDENTITY(1, 1),
      genre_name VARCHAR(40) NOT NULL,
      CONSTRAINT prim_genreId PRIMARY KEY(genre_id)
)
GO

CREATE TABLE movie
(
      movie_id INT NOT NULL IDENTITY(1, 1),
      movie_name VARCHAR(40) NOT NULL,
      movie_description VARCHAR(150) NULL,
      genre_id int NOT NULL,
      CONSTRAINT prim_movie_id PRIMARY KEY(movie_id),
      CONSTRAINT for_genre_id FOREIGN KEY(genre_id) REFERENCES
genre(genre_id)
)
GO

CREATE TABLE users
(
      users_id INT NOT NULL IDENTITY(1, 1),
      users_name VARCHAR(15) NOT NULL,
      users_password VARCHAR(20) NOT NULL,
      CONSTRAINT prim_users_id PRIMARY KEY(users_id)
)
GO

--inserting the sample data into the tables on the database
INSERT INTO genre(genre_name)
VALUES ('Action'), ('Comedy'), ('Drama'), ('Thrill'), ('Suspense')
GO

INSERT INTO movie(movie_name, movie_description, genre_id)
VALUES ('Die Hard 4.0','Intense action movie starring the esteemed Bruce
Willis', 1),
      ('Hang Over 2','Crazy comedy movie about a bachelor party gone wrong ',
2),
      ('The Expendables','Hardcore action movie with Salvestor Stallon', 1),
      ('Prometheus','Brilliant movie shown as the prequel to the Aliens of
AvP', 5),
      ('Kings Speech', 'Movie about a vocally challenged King who gives the
best speech of all time', 3)
GO

INSERT INTO users(users_name, users_password)
VALUES ('Tejas', 'gero_$000'),
      ('Roland', 'Iceangel4%100'),
      ('minal', 'bongani#g')
GO
```