**《JAVA程序设计》课程设计**

**实验报告**

课程名称： 《JAVA》程序设计

设计题目： JAVA课程签到系统

指导老师： 余文君

专 业： 计算机科学与技术

班 级： 1803

组 员： 邢广杰 陈雨情

2019年12月19日

目录

[一、 选题与分工 - 2 -](#_Toc1196125787)

[1.1选题背景 - 2 -](#_Toc1141748000)

[1.2 详细分工 - 2 -](#_Toc925327330)

[1.3 开发环境 - 2 -](#_Toc217219377)

[二、 设计与实现 - 3 -](#_Toc33938665)

[2.1 设计目的 - 3 -](#_Toc1479000725)

[2.2 系统设计 - 3 -](#_Toc521239773)

[2.2.1 系统目标 - 3 -](#_Toc765425379)

[2.2.2 主题目标 - 4 -](#_Toc73226988)

[2.2.3 客户端功能 - 4 -](#_Toc1131027196)

[2.2.4 服务端功能 - 5 -](#_Toc1794706486)

[2.3 实现 - 6 -](#_Toc56213154)

[2.3.1 涉及知识点 - 6 -](#_Toc537138875)

[2.3.2 类图实现 - 6 -](#_Toc1636305273)

[2.4 源程序清单 - 9 -](#_Toc251999190)

[2.4.1 Server.java - 9 -](#_Toc2127433994)

[2.4.2 MultithreadSocketServer.java - 12 -](#_Toc615360791)

[2.4.3 ServerLogin.java - 13 -](#_Toc897950312)

[2.4.4 ServerAfterLogin.java - 18 -](#_Toc877288948)

[2.4.5 ClientLogin.java - 28 -](#_Toc1066439064)

[2.4.6 ClientAfterLogin.java - 33 -](#_Toc604611204)

[2.4.7 MySqlConnect.java - 40 -](#_Toc745518995)

## 选题与分工

### 1.1选题背景

《JAVA程序设计》一直是信息学院的同学们非常喜爱的一门课程，在这门课程上可以学到Java程序设计语言的方法和思想。《JAVA程序设计》课程配套的实验课程内容丰富，同学们可以在这门课上动手实践，掌握真正的程序设计技巧。由于同学们的热情，这门课一直以来都是座无虚席，特别是上实验课的时候。但由于学院信息技术教室的系统比较古老且不易于维护，学生管理就成为了教授这门课程的老师们的一项难题。同学们敬爱的余文君老师为考验同学们的开发能力，希望同学们分组开发出一套Java课程管理系统，帮助老师们实现一些需求的管理功能。于是，经过本小组内成员反复讨论、设计，决定开发一套安全、实用且友好的Java课程管理系统，希望能为老师们管理课堂提供一点帮助。

### 1.2 详细分工

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 主要任务 |
| 2017308210404 | 邢广杰 | 项目规划、框架设计、GUI设计与实现、网络通讯实现、部分功能实现 |
| 2018317210323 | 陈雨情 | 功能设计、数据库设计、数据库实现、部分功能实现 |

### 1.3 开发环境

编程工具： IntelliJ IDEA 2019.3 (Ultimate Edition)

JDK版本： jdk1.7.0\_80

操作系统： 开发：Ubuntu18.04 LST(amd\_64) 、macOS

测试：Windows7 Professional(x64)、Windows 10 Enterprise 2016 LTSB (x64)

数 据 库： Mysql（腾讯云远程数据库）

## 设计与实现

### 2.1 设计目的

本课程设计的目的就是通过一次训练，使学生能巩固自己的学识，补充未学但有必要的内容。进行课程设计的主要目的在于加深对程序设计基础中理论和基本知识的理解，促进理论与实践的结合，进一步提高程序设计的能力。同时，在程序设计方法及上机操作方面受到训练，培养软件工作所需要的动手能力。

1. 使学生更深入的理解和掌握课程中有关基本概念，程序设计的思想和方法。
2. 培养学生综合运用所学只是独立完成课题的能力。
3. 培养学生用于探索、严谨推理、实事求是、有错必改，用实践来验证理论全方位考虑问题等科学技术人员应具有的素质。
4. 提高学生对工作认真负责，一丝不苟，对同学团结友爱，协作攻关的基本素质。
5. 培养学生从资料文献、科学实验中获得知识的能力，提高学生从别人经验中找到解决问题的新途径的悟性，初步培养工程意识和创新能力。
6. 对学生掌握知识的深度，运用理论去处理问题的能力、实验能力、课程设计能力、书面及口头表达能力进行考核。

### 2.2 系统设计

#### 2.2.1 系统目标

编写Java程序代码设计一个一对多通讯课程管理系统。

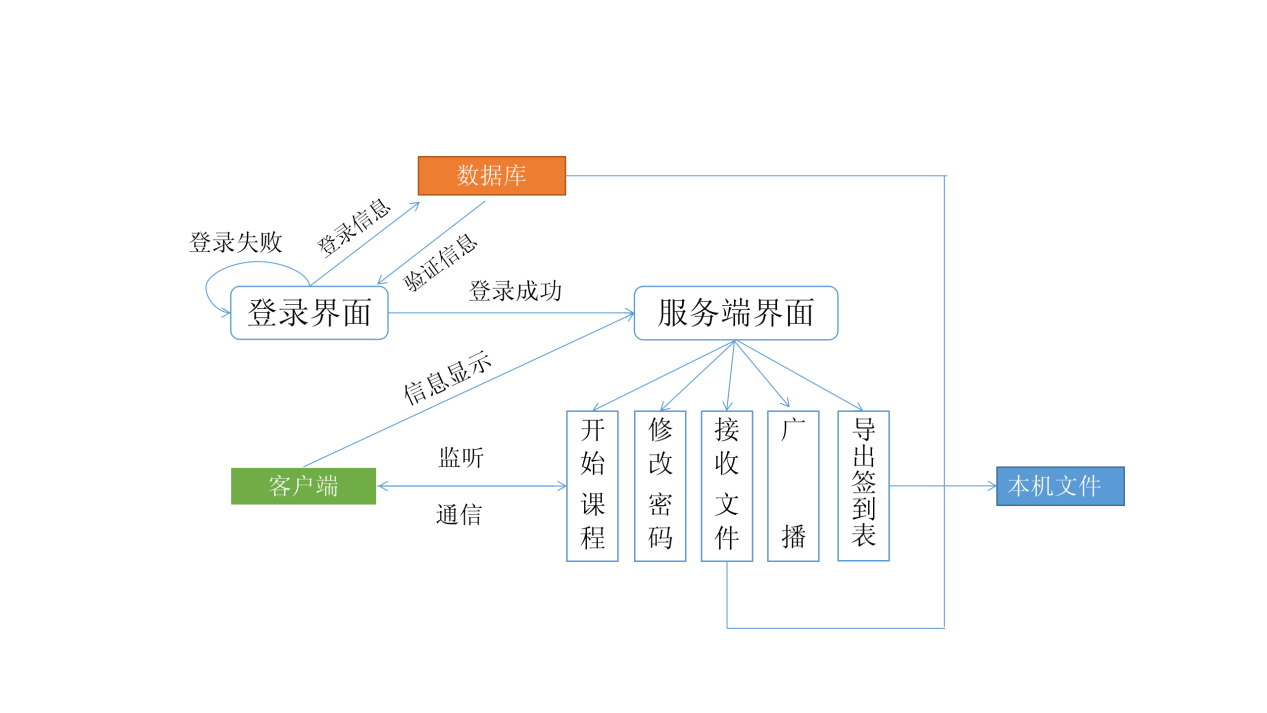
#### 2.2.2 主题目标

编写的程序能够划分为客户端和服务端，需要具备友好的图形用户界面，需要能够进行一对多的网络通信，需要能够进行文件传输，需要能够链接数据库，需要能够在数据库中插入、修改、查询和删除数据；能够导出文件。

#### 2.2.3 客户端功能

1. 可以使用帐号密码进行登录，有登录验证机制；
2. 登录之后能够记录签到时间并发送至服务端；
3. 登录之后可以看到课程信息及个人信息，能够实时接受来自服务端的信息并且显示在客户端的用户界面上；
4. 能够实现选择文件并发送文件至服务端；
5. 可以主动登出，记录登出时间并发送至服务端。

其模式图形如图2.1:

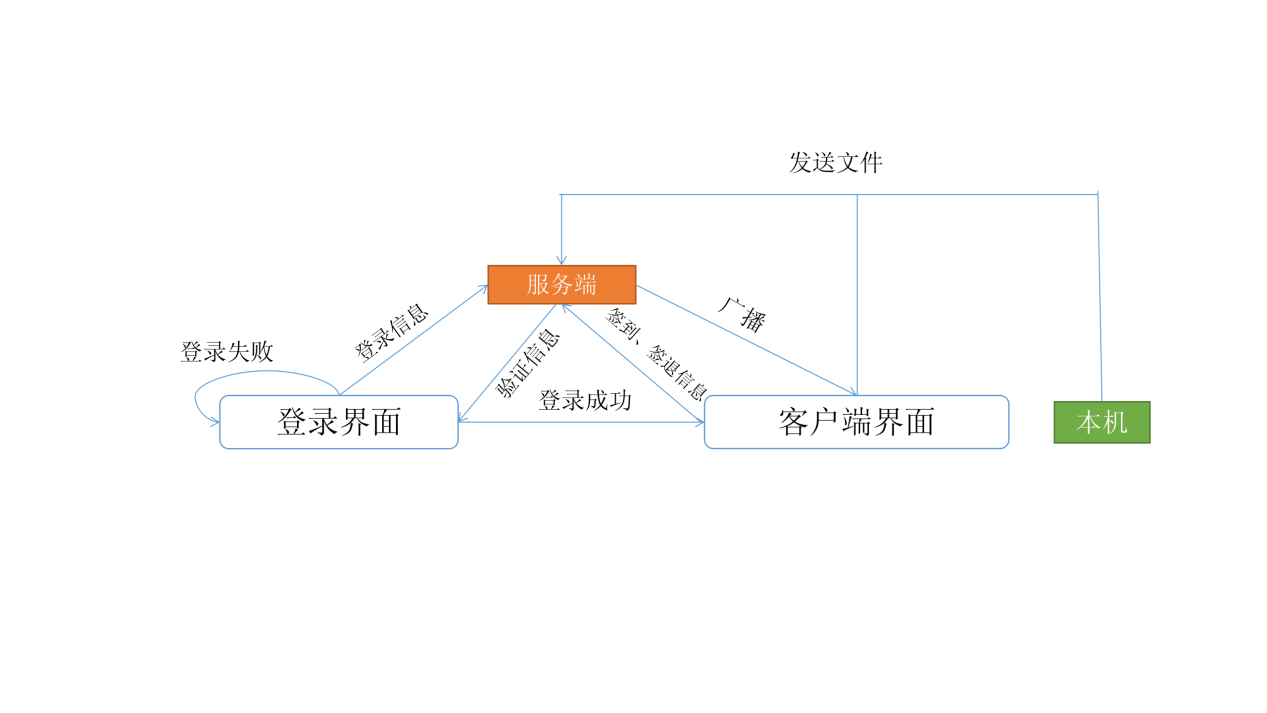


*图2.1*

#### 2.2.4 服务端功能

1. 服务端：可以使用管理员密码进行登录，有登录验证机制；
2. 登录之后可以实现设置课程信息，并启动课程（启动服务）；
3. 启动服务之后可以实时接受客户端的链接，实时显示链接所有建立链接客户端的信息在服务端用户界面上；
4. 可以实时广播消息给所有客户端；
5. 可以实时接受客户端发送来的文件，可以实时显示哪些客户端已经发送文件；
6. 可以更改管理员密码；
7. 可以对该堂课程的签到表导出到一个表格文件（.csv)。

其模式图形如图2.2:



*图2.2*

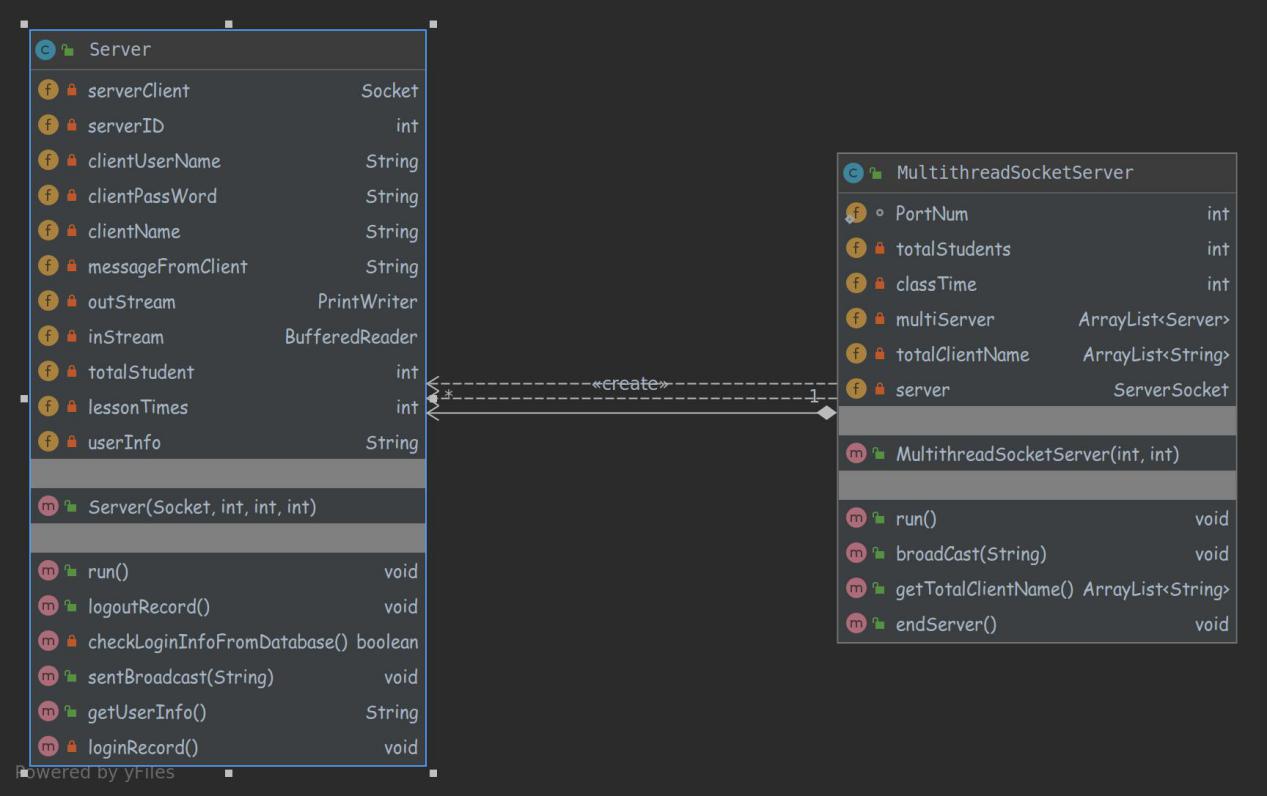
### 2.3 实现

#### 2.3.1 涉及知识点

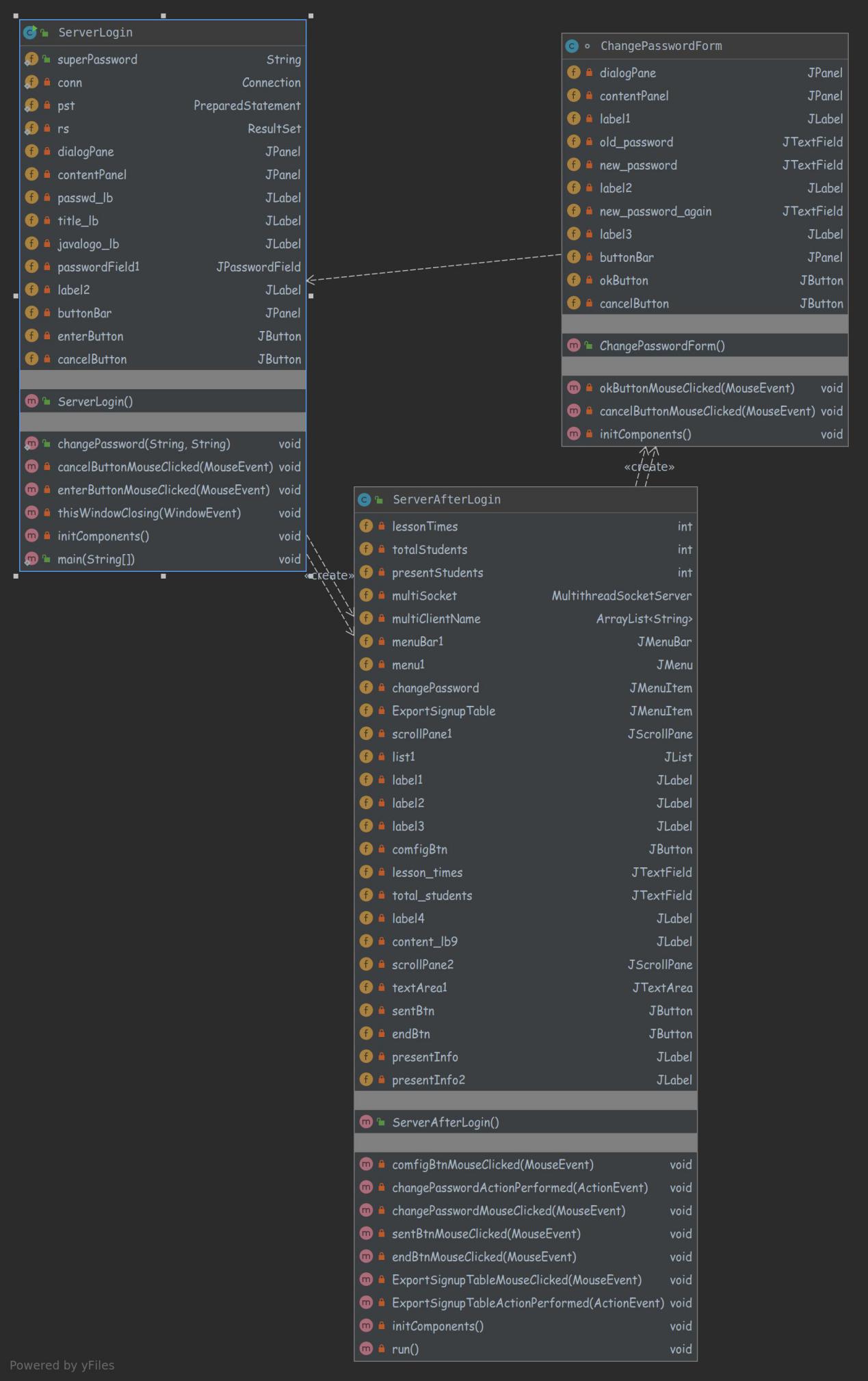
Java编程、面向对象程序设计思想与方法、javax.swing GUI编程、socket网络通信编程、多线程技术、Mysql数据库技术

#### 2.3.2 类图实现

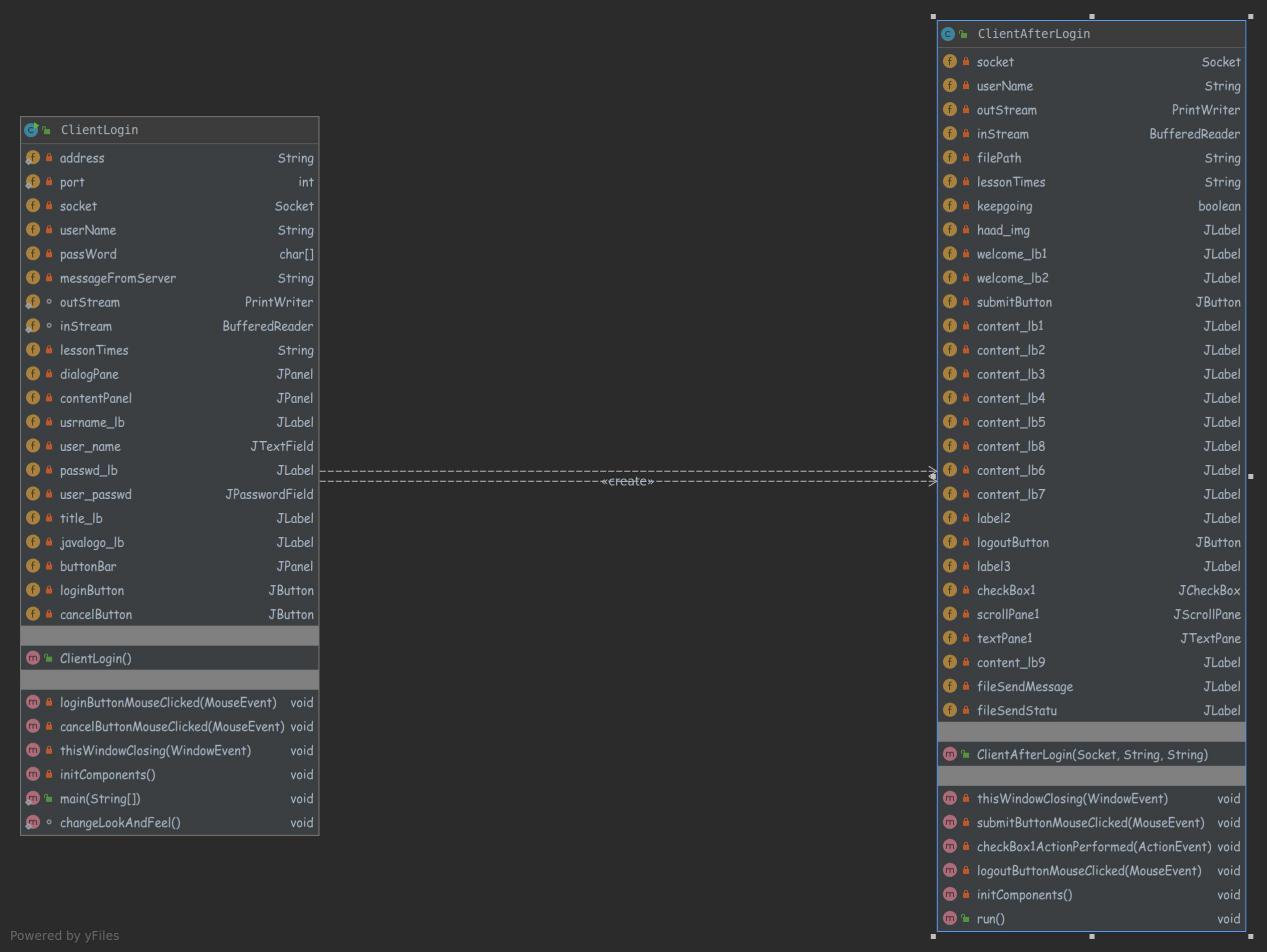
1. 服务端后端类图



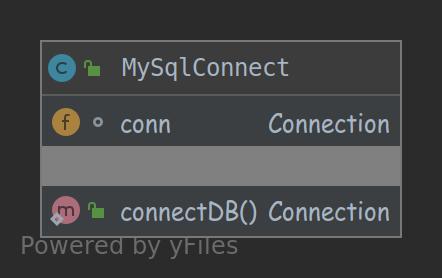
1. 服务端前端类图



1. 客户端前端类图



1. 数据库类图



### 2.4 源程序清单

2.4.1 Server.java**package** backend;  
  
**import** connection.MySqlConnect;  
  
**import** javax.swing.filechooser.FileSystemView;  
**import** java.io.\*;  
**import** java.net.Socket;  
**import** java.sql.Connection;  
**import** java.sql.PreparedStatement;  
**import** java.sql.ResultSet;  
**import** java.sql.SQLException;  
**import** java.text.SimpleDateFormat;  
**import** java.util.Date;  
  
**public class** Server **extends** Thread{  
 **private** Socket **serverClient**;  
 **private int serverID**;  
 **private** String **clientUserName**;  
 **private** String **clientPassWord**;  
 **private** String **clientName**;  
 **private** String **messageFromClient**;  
 **private** PrintWriter **outStream**;  
 **private** BufferedReader **inStream**;  
 **private int totalStudent**;  
 **private int lessonTimes**;  
 **private** String **userInfo**;  
 **public** Server(Socket client\_socket, **int** clientNum, **int** totalStudent, **int** lessonTimes) {  
 **serverClient** = client\_socket;  
 **serverID** = clientNum;  
 **this**.**totalStudent** = totalStudent;  
 **this**.**lessonTimes** = lessonTimes;  
 System.***out***.println(**"Connect from : \n"** + **serverClient**.toString());  
 }  
  
 @Override  
 **public void** run() {  
 *//* ***TODO Auto-generated method stub* try** {  
 **inStream** = **new** BufferedReader(**new** InputStreamReader(**serverClient**.getInputStream()));  
 **outStream** = **new** PrintWriter(**serverClient**.getOutputStream());  
 **outStream**.println(**lessonTimes**);  
 **outStream**.flush();  
 **boolean** keepGoing = **true**;  
 **while**(keepGoing) {  
 *//System.out.println("waiting for " + serverID + "'s" + "message");* **try**{  
 **messageFromClient** = **inStream**.readLine();  
 System.***out***.println(**"Message from Client "** + **serverID** + **": "** + **messageFromClient** + **"..."**);  
 **if**(**messageFromClient**.equals(**"Login Request"**)){  
 **clientUserName** = **inStream**.readLine();  
 **clientPassWord** = **inStream**.readLine();  
 System.***out***.println(**"Received user name : "** + **clientUserName** + **"\npassword : "** + **clientPassWord**);  
 **if**(checkLoginInfoFromDatabase()){  
 System.***out***.println(**"Client "** + **serverID** + **" Password test successfully!"**);  
 **try**{  
 **outStream**.println(**"Login Passed"**);  
 **outStream**.flush();  
 **outStream**.println(**clientName**);  
 **outStream**.flush();  
 loginRecord();  
 **userInfo** = **clientUserName** + **" "** + **clientName**;  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 **else**{  
 System.***out***.println(**"Client "** + **serverID** + **"Password test failed."**);  
 **try**{  
 **outStream**.println(**"Login Failed"**);  
 **outStream**.flush();  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
 **if**(**messageFromClient**.equals(**"exit"**)){  
 System.***out***.println(**"Message from Client "** + **serverID** + **": "** + **messageFromClient** + **"..."**);  
 System.***out***.println(**"Client "** + **serverID** + **" has disturbed."**);  
 logoutRecord();  
 keepGoing = **false**;  
 }  
  
 **if**(**messageFromClient**.equals(**"logout"**)){  
 System.***out***.println(**"Message from Client "** + **serverID** + **": "** + **messageFromClient** + **"..."**);  
 System.***out***.println(**"Client "** + **serverID** + **" has logged out."**);  
 logoutRecord();  
 keepGoing = **false**;  
 }  
 **if**(**messageFromClient**.equals(**"sendfile"**)){  
 System.***out***.println(**"Receiving File..."**);  
 DataInputStream dis =**new** DataInputStream(**serverClient**.getInputStream());  
 **long** size =dis.readLong();  
 String suffix =dis.readUTF();  
 FileSystemView fsv = FileSystemView.*getFileSystemView*();  
 File path = **new** File(fsv.getHomeDirectory().getPath() + **"/temp/"** + **userInfo**);  
 **if**(!path.exists()){  
 path.mkdir();  
 System.***out***.println(**"Creat dir "** + path.getPath());  
 }  
 File dest = **new** File(path.getPath() + **"/"** + **userInfo** + **"."** + suffix);  
 BufferedInputStream bis = **new** BufferedInputStream(**serverClient**.getInputStream());  
 BufferedOutputStream bos = **new** BufferedOutputStream(**new** FileOutputStream(dest));  
 **long** readSize = 0;  
 **while**(**true**){  
 **if**(readSize > size){  
 **break**;  
 }  
 bos.write(bis.read());  
 readSize++;  
 }  
 System.***out***.println(**"Upload Competed"**);  
 bos.flush();  
 **userInfo** += **"\*"**;  
 }  
  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 System.***out***.println(**"Disconnect from : \n"** + **serverClient**.toString());  
 **inStream**.close();  
 **outStream**.close();  
 **serverClient**.close();  
 }  
 **catch** (IOException e) {  
 *//* ***TODO: handle exception*** e.printStackTrace();  
 }  
 }  
  
 **public void** logoutRecord() {  
 String sql = **"insert into logOutRecord values(null, ?, ?, ?)"**;  
 Connection conn = MySqlConnect.*connectDB*();  
 **try**{  
 SimpleDateFormat df = **new** SimpleDateFormat(**"yyyy-MM-dd HH:mm:ss"**);  
 String recordDate = df.format(**new** Date());  
 **assert** conn != **null**;  
 PreparedStatement pst = conn.prepareStatement(sql);  
 pst.setString(1, **clientName**);  
 pst.setString(2, recordDate);  
 pst.setString(3, **clientUserName**);  
 pst.executeUpdate();  
 } **catch** (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
  
 **private boolean** checkLoginInfoFromDatabase(){  
 String sql = **"select student\_password, student\_name from students where student\_id = ?"**;  
 Connection conn = MySqlConnect.*connectDB*();  
 **try**{  
 String correctPassWord = **""**;  
 PreparedStatement pst = conn.prepareStatement(sql);  
 pst.setString(1, **clientUserName**);  
 ResultSet rs = pst.executeQuery();  
 **while**(rs.next()){  
 correctPassWord = rs.getString(1);  
 **clientName**= rs.getString(2);  
 }  
 **if**(**clientPassWord**.equals(correctPassWord)){  
 **return true**;  
 }  
 **else return false**;  
 } **catch** (SQLException e) {  
 e.printStackTrace();  
 **return false**;  
 }  
 }  
  
 **public void** sentBroadcast(String broadcast){  
 **try**{  
 **outStream**.println(**"broadcast"**);  
 **outStream**.flush();  
 **outStream**.println(broadcast);  
 **outStream**.flush();  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 **public** String getUserInfo(){  
 **return userInfo**;  
 }  
  
 **private void** loginRecord(){  
 String sql = **"insert into signInRecord values(null, ?, ?, ?)"**;  
 Connection conn = MySqlConnect.*connectDB*();  
 **try**{  
 SimpleDateFormat df = **new** SimpleDateFormat(**"yyyy-MM-dd HH:mm:ss"**);  
 String recordDate = df.format(**new** Date());  
 **assert** conn != **null**;  
 PreparedStatement pst = conn.prepareStatement(sql);  
 pst.setString(1, **clientName**);  
 pst.setString(2, recordDate);  
 pst.setString(3, **clientUserName**);  
 pst.executeUpdate();  
 } **catch** (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
}

#### 2.4.2 MultithreadSocketServer.java

**package** backend;  
  
**import** com.sun.xml.internal.bind.v2.model.annotation.RuntimeAnnotationReader;  
  
**import** java.io.IOException;  
**import** java.net.ServerSocket;  
**import** java.net.Socket;  
**import** java.util.ArrayList;  
**import** java.util.Iterator;  
**import** java.util.Scanner;  
  
**public class** MultithreadSocketServer **implements** Runnable {  
 **static int** *PortNum* = 9090;  
 **private int totalStudents**;  
 **private int classTime**;  
 **private** ArrayList<Server> **multiServer**;  
 **private** ArrayList<String> **totalClientName**;  
 **private** ServerSocket **server**;  
  
 **public** MultithreadSocketServer(**int** totalStudents, **int** classTime) {  
 **this**.**totalStudents** = totalStudents;  
 **this**.**classTime** = classTime;  
 **this**.**multiServer** = **new** ArrayList<>();  
 **this**.**totalClientName** = **new** ArrayList<>();  
 }  
  
 @Override  
 **public void** run() {  
 **int** clientSum = 0;  
 **try**{  
 **server** = **new** ServerSocket(*PortNum*);  
 System.***out***.println(**"Waiting for client on port "** + **server**.getLocalPort() + **"..."**);  
  
 **while**(**true**){  
 clientSum++;  
 Socket serverClient = **server**.accept();  
 System.***out***.println(**"Just connected to"** + serverClient.getRemoteSocketAddress());  
 System.***out***.println(**" >> "** + **"Client No: "** + clientSum + **" Started!"**);  
 Server s = **new** Server(serverClient, clientSum, **totalStudents**, **classTime**);  
 **multiServer**.add(s);  
 s.start();  
 }  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 **public void** broadCast(String message){  
 **for**(Iterator<Server> item = **multiServer**.iterator(); item.hasNext(); ){  
 item.next().sentBroadcast(message);  
 }  
 }  
  
 **public** ArrayList<String> getTotalClientName(){  
 **for**(Iterator<Server> item = **multiServer**.iterator(); item.hasNext();){  
 String infoItem = item.next().getUserInfo();  
 **if**(infoItem != **null**){  
 **if**(**totalClientName**.contains(infoItem)){  
 **continue**;  
 }  
 **if**(**totalClientName**.contains(infoItem.substring(0, infoItem.length()-1))){  
 **totalClientName**.remove(infoItem.substring(0, infoItem.length()-1));  
 }  
 **totalClientName**.add(infoItem);  
 }  
 }  
 **return totalClientName**;  
 }  
  
 **public void** endServer() {  
 **for**(Iterator<Server> item = **multiServer**.iterator(); item.hasNext();){  
 item.next().logoutRecord();  
 }  
 **try**{  
 **server**.close();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

#### 2.4.3 ServerLogin.java

*/\*  
 \* Created by JFormDesigner on Thu Dec 12 19:13:07 CST 2019  
 \*/***package** view;  
  
**import** backend.MultithreadSocketServer;  
**import** connection.MySqlConnect;  
  
**import** java.awt.\*;  
**import** java.awt.event.\*;  
**import** java.sql.Connection;  
**import** java.sql.PreparedStatement;  
**import** java.sql.ResultSet;  
**import** java.sql.SQLException;  
**import** javax.swing.\*;  
**import** javax.swing.GroupLayout;  
**import** javax.swing.border.\*;  
  
  
*/\*\*  
 \** ***@author*** *jiengup  
 \*/***public class** ServerLogin **extends** JFrame {  
 **public static** String *superPassword*;  
 **private static** Connection *conn* = **null**;  
 **private static** PreparedStatement *pst* = **null**;  
 **private static** ResultSet *rs* = **null**;  
 **public** ServerLogin() **throws** SQLException {  
 initComponents();  
 String sql = **"select super\_password from super where super\_password\_id = (select** *max***(super\_password\_id) from super)"**;  
 String correctPassword = **""**;  
 *conn* = MySqlConnect.*connectDB*();  
 **try**{  
 *pst* = *conn*.prepareStatement(sql);  
 *rs* = *pst*.executeQuery();  
 **while**(*rs*.next()){  
 correctPassword = *rs*.getString(1);  
 System.***out***.println(**"Return password : "** + correctPassword);  
 }  
 } **catch** (SQLException e) {  
 e.printStackTrace();  
 }**finally** {  
 *pst*.close();  
 }  
 *superPassword* = correctPassword;  
 }  
  
 **public static void** changePassword(String newPassword, String oldPassword) **throws** SQLException {  
 **if**(!oldPassword.equalsIgnoreCase(*superPassword*)){  
 JOptionPane.*showMessageDialog*(**null**, **"Old Password Incorrect"**, **"Alert"**, JOptionPane.***ERROR\_MESSAGE***);  
 }  
 **else**{  
 *conn* = MySqlConnect.*connectDB*();  
 String sql = **"insert into super values(null, ?)"**;  
 **try** {  
 *pst* = *conn*.prepareStatement(sql);  
 *pst*.setString(1, newPassword);  
 *pst*.executeUpdate();  
 *superPassword* = newPassword;  
 } **catch** (SQLException e) {  
 e.printStackTrace();  
 } **finally** {  
 *pst*.close();  
 }  
 }  
 }  
  
 **private void** cancelButtonMouseClicked(MouseEvent e) {  
 System.*exit*(0);  
 }  
  
 **private void** enterButtonMouseClicked(MouseEvent e) {  
 String inputPassword = **new** String(**passwordField1**.getPassword());  
 **if**(inputPassword.equalsIgnoreCase(**""**)){  
 JOptionPane.*showMessageDialog*(**this**, **"Please input your password"**, **"Warning"**, JOptionPane.***WARNING\_MESSAGE***);  
 }  
 **else**{  
 **if**(inputPassword.equals(*superPassword*)){  
  
 EventQueue.*invokeLater*(**new** Runnable() {  
 @Override  
 **public void** run() {  
 ServerAfterLogin sal = **new** ServerAfterLogin();  
 sal.setVisible(**true**);  
 }  
 });  
 **this**.dispose();  
 }  
 **else**{  
 JOptionPane.*showMessageDialog*(**this**, **"Incorrect Password\nPlease Enter Again."**, **"Message"**, JOptionPane.***WARNING\_MESSAGE***);  
 }  
 }  
 }  
  
 **private void** thisWindowClosing(WindowEvent e) {  
 System.*exit*(0);  
 }  
  
 **private void** initComponents() {  
 *// JFormDesigner - Component initialization - DO NOT MODIFY //GEN-BEGIN:initComponents* **dialogPane** = **new** JPanel();  
 **contentPanel** = **new** JPanel();  
 **passwd\_lb** = **new** JLabel();  
 **title\_lb** = **new** JLabel();  
 **javalogo\_lb** = **new** JLabel();  
 **passwordField1** = **new** JPasswordField();  
 **label2** = **new** JLabel();  
 **buttonBar** = **new** JPanel();  
 **enterButton** = **new** JButton();  
 **cancelButton** = **new** JButton();  
  
 *//======== this ========* setResizable(**false**);  
 setTitle(**"Server Login"**);  
 addWindowListener(**new** WindowAdapter() {  
 @Override  
 **public void** windowClosing(WindowEvent e) {  
 thisWindowClosing(e);  
 }  
 });  
 Container contentPane = getContentPane();  
 contentPane.setLayout(**new** BorderLayout());  
  
 *//======== dialogPane ========* {  
 **dialogPane**.setBorder(**new** EmptyBorder(12, 12, 12, 12));  
 **dialogPane**.setLayout(**new** BorderLayout());  
  
 *//======== contentPanel ========* {  
  
 *//---- passwd\_lb ----* **passwd\_lb**.setText(**"Super Administor's Password please:"**);  
 **passwd\_lb**.setFont(**passwd\_lb**.getFont().deriveFont(**passwd\_lb**.getFont().getSize() + 3f));  
  
 *//---- title\_lb ----* **title\_lb**.setText(**"Java Class Management"**);  
 **title\_lb**.setFont(**new** Font(**"MathJax\_Typewriter"**, Font.***BOLD***, 22));  
 **title\_lb**.setForeground(Color.***darkGray***);  
  
 *//---- javalogo\_lb ----* **javalogo\_lb**.setIcon(**new** ImageIcon(getClass().getResource(**"/resource/javalogo2.png"**)));  
  
 *//---- passwordField1 ----* **passwordField1**.setToolTipText(**"your password here"**);  
  
 *//---- label2 ----* **label2**.setText(**"Dear Mr Yu: "**);  
 **label2**.setFont(**new** Font(**"Segoe Print"**, Font.***BOLD***, 15));  
  
 GroupLayout contentPanelLayout = **new** GroupLayout(**contentPanel**);  
 **contentPanel**.setLayout(contentPanelLayout);  
 contentPanelLayout.setHorizontalGroup(  
 contentPanelLayout.createParallelGroup()  
 .addGroup(contentPanelLayout.createSequentialGroup()  
 .addGap(0, 121, Short.***MAX\_VALUE***)  
 .addComponent(**passwordField1**, GroupLayout.***PREFERRED\_SIZE***, 329, GroupLayout.***PREFERRED\_SIZE***)  
 .addContainerGap(114, Short.***MAX\_VALUE***))  
 .addGroup(contentPanelLayout.createSequentialGroup()  
 .addGap(88, 88, 88)  
 .addComponent(**label2**)  
 .addContainerGap(376, Short.***MAX\_VALUE***))  
 .addGroup(contentPanelLayout.createSequentialGroup()  
 .addGap(135, 135, 135)  
 .addComponent(**passwd\_lb**)  
 .addContainerGap(149, Short.***MAX\_VALUE***))  
 .addGroup(contentPanelLayout.createSequentialGroup()  
 .addGap(120, 120, 120)  
 .addComponent(**javalogo\_lb**)  
 .addGap(18, 18, 18)  
 .addComponent(**title\_lb**, GroupLayout.***PREFERRED\_SIZE***, 256, GroupLayout.***PREFERRED\_SIZE***)  
 .addContainerGap(125, Short.***MAX\_VALUE***))  
 );  
 contentPanelLayout.setVerticalGroup(  
 contentPanelLayout.createParallelGroup()  
 .addGroup(contentPanelLayout.createSequentialGroup()  
 .addGap(32, 32, 32)  
 .addGroup(contentPanelLayout.createParallelGroup()  
 .addGroup(contentPanelLayout.createSequentialGroup()  
 .addGap(24, 24, 24)  
 .addComponent(**title\_lb**))  
 .addComponent(**javalogo\_lb**))  
 .addGap(46, 46, 46)  
 .addComponent(**label2**)  
 .addGap(18, 18, 18)  
 .addComponent(**passwd\_lb**)  
 .addGap(29, 29, 29)  
 .addComponent(**passwordField1**, GroupLayout.***PREFERRED\_SIZE***, GroupLayout.***DEFAULT\_SIZE***, GroupLayout.***PREFERRED\_SIZE***)  
 .addContainerGap(73, Short.***MAX\_VALUE***))  
 );  
 }  
 **dialogPane**.add(**contentPanel**, BorderLayout.***CENTER***);  
  
 *//======== buttonBar ========* {  
 **buttonBar**.setBorder(**new** EmptyBorder(12, 0, 0, 0));  
 **buttonBar**.setLayout(**new** GridBagLayout());  
 ((GridBagLayout)**buttonBar**.getLayout()).**columnWidths** = **new int**[] {0, 85, 80};  
 ((GridBagLayout)**buttonBar**.getLayout()).**columnWeights** = **new double**[] {1.0, 0.0, 0.0};  
  
 *//---- enterButton ----* **enterButton**.setText(**"Enter"**);  
 **enterButton**.setIcon(**new** ImageIcon(getClass().getResource(**"/resource/login.png"**)));  
 **enterButton**.setFont(**enterButton**.getFont().deriveFont(**enterButton**.getFont().getSize() + 3f));  
 **enterButton**.addMouseListener(**new** MouseAdapter() {  
 @Override  
 **public void** mouseClicked(MouseEvent e) {  
 enterButtonMouseClicked(e);  
 }  
 });  
 **buttonBar**.add(**enterButton**, **new** GridBagConstraints(1, 0, 1, 1, 0.0, 0.0,  
 GridBagConstraints.***CENTER***, GridBagConstraints.***BOTH***,  
 **new** Insets(0, 0, 0, 5), 0, 0));  
  
 *//---- cancelButton ----* **cancelButton**.setText(**"Cancel"**);  
 **cancelButton**.setIcon(**new** ImageIcon(getClass().getResource(**"/resource/cancel.png"**)));  
 **cancelButton**.setFont(**cancelButton**.getFont().deriveFont(**cancelButton**.getFont().getSize() + 3f));  
 **cancelButton**.addMouseListener(**new** MouseAdapter() {  
 @Override  
 **public void** mouseClicked(MouseEvent e) {  
 cancelButtonMouseClicked(e);  
 }  
 });  
 **buttonBar**.add(**cancelButton**, **new** GridBagConstraints(2, 0, 1, 1, 0.0, 0.0,  
 GridBagConstraints.***CENTER***, GridBagConstraints.***BOTH***,  
 **new** Insets(0, 0, 0, 0), 0, 0));  
 }  
 **dialogPane**.add(**buttonBar**, BorderLayout.***SOUTH***);  
 }  
 contentPane.add(**dialogPane**, BorderLayout.***CENTER***);  
 pack();  
 setLocationRelativeTo(getOwner());  
 *// JFormDesigner - End of component initialization //GEN-END:initComponents* }  
  
 **public static void** main(String[] args){  
 ClientLogin.*changeLookAndFeel*();  
  
 EventQueue.*invokeLater*(**new** Runnable() {  
 @Override  
 **public void** run() {  
 **try** {  
 **new** ServerLogin().setVisible(**true**);  
 } **catch** (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
 }  
 *// JFormDesigner - Variables declaration - DO NOT MODIFY //GEN-BEGIN:variables* **private** JPanel **dialogPane**;  
 **private** JPanel **contentPanel**;  
 **private** JLabel **passwd\_lb**;  
 **private** JLabel **title\_lb**;  
 **private** JLabel **javalogo\_lb**;  
 **private** JPasswordField **passwordField1**;  
 **private** JLabel **label2**;  
 **private** JPanel **buttonBar**;  
 **private** JButton **enterButton**;  
 **private** JButton **cancelButton**;  
 *// JFormDesigner - End of variables declaration //GEN-END:variables*}

#### 2.4.4 ServerAfterLogin.java

*/\*  
 \* Created by JFormDesigner on Fri Dec 13 20:53:48 CST 2019  
 \*/***package** view;  
  
**import** backend.MultithreadSocketServer;  
**import** connection.MySqlConnect;  
**import** java.awt.\*;  
**import** java.awt.event.\*;  
**import** java.io.\*;  
**import** java.sql.Connection;  
**import** java.sql.PreparedStatement;  
**import** java.sql.ResultSet;  
**import** java.sql.SQLException;  
**import** java.text.SimpleDateFormat;  
**import** java.util.ArrayList;  
**import** java.util.Date;  
**import** java.util.Iterator;  
**import** javax.swing.\*;  
**import** javax.swing.GroupLayout;  
**import** javax.swing.border.\*;  
**import** javax.swing.filechooser.FileSystemView;  
  
*/\*\*  
 \** ***@author*** *jiengup  
 \*/***public class** ServerAfterLogin **extends** JFrame {  
 **private int lessonTimes**;  
 **private int totalStudents**;  
 **private int presentStudents**;  
 **private** MultithreadSocketServer **multiSocket**;  
 **private** ArrayList<String> **multiClientName**;  
  
 **public** ServerAfterLogin() {  
 initComponents();  
 }  
 **private void** comfigBtnMouseClicked(MouseEvent e) {  
 *//* ***TODO add your code here* if**(!**comfigBtn**.isEnabled()) **return**;  
 **lessonTimes** = Integer.*parseInt*(**lesson\_times**.getText());  
 **totalStudents** = Integer.*parseInt*(**total\_students**.getText());  
 System.***out***.println(**"Lesson Times : "** + **lessonTimes** + **"Total Student : "** + **totalStudents**);  
 **multiSocket** = **new** MultithreadSocketServer(**totalStudents**, **lessonTimes**);  
 **new** Thread(**multiSocket**).start();  
 **sentBtn**.setEnabled(**true**);  
 **endBtn**.setEnabled(**true**);  
 **textArea1**.setEnabled(**true**);  
 **comfigBtn**.setEnabled(**false**);  
 **list1**.setEnabled(**true**);  
 run();  
 }  
  
  
 **private void** changePasswordActionPerformed(ActionEvent e) {  
 ChangePasswordForm cpf = **new** ChangePasswordForm();  
 cpf.setVisible(**true**);  
 }  
  
 **private void** changePasswordMouseClicked(MouseEvent e) {  
 *//* ***TODO add your code here*** }  
  
 **private void** sentBtnMouseClicked(MouseEvent e) {  
 **if**(!**sentBtn**.isEnabled()) **return**;  
 String broadcast = **textArea1**.getText();  
 **multiSocket**.broadCast(broadcast);  
 }  
  
 **private void** endBtnMouseClicked(MouseEvent e) {  
 System.***out***.println(**"Class End"**);  
 **multiSocket**.endServer();  
 System.*exit*(0);  
 }  
  
 **private void** ExportSignupTableMouseClicked(MouseEvent e) {  
  
 }  
  
 **private void** ExportSignupTableActionPerformed(ActionEvent e) {  
 String sql = **"select distinct a1.userId ID, a1.name, a1.time signinTime, a2.time logoutTime from(select userId, name, time from signInRecord where time in (select min(time) from signInRecord group by signInRecord.userId)) as a1, (select userId, name, time from logOutRecord where time in (select max(time) from logOutRecord group by logOutRecord.userId)) as a2 where a1.userId = a2.userId"**;  
 Connection conn = MySqlConnect.*connectDB*();  
 SimpleDateFormat df = **new** SimpleDateFormat(**"yyyy-MM-dd HH:mm:ss"**);  
 String date = df.format(**new** Date()).substring(0, 10);  
 FileSystemView fsv = FileSystemView.*getFileSystemView*();  
 String dirPath = fsv.getHomeDirectory() + **"/temp/"**;  
 File dir = **new** File(dirPath);  
 **if**(!dir.exists()){  
 dir.mkdir();  
 }  
 **try**{  
 File recordFile = **new** File(dir.getPath() + **"/"** + date + **"signin&logouttable.csv"**);  
 **if**(recordFile.exists()){  
 recordFile.createNewFile();  
 }  
 FileOutputStream bis = **new** FileOutputStream(recordFile.getPath());  
 bis.write(**"ID,name,signinTime,logoutTime\n"**.getBytes());  
 PreparedStatement pst = conn.prepareStatement(sql);  
 ResultSet rs = pst.executeQuery();  
 **while**(rs.next()){  
 String record = rs.getString(1) + **","** + rs.getString(2) + **","** + rs.getDate(3) + **","** + rs.getDate(4) + **"\n"**;  
 System.***out***.println(record);  
 bis.write(record.getBytes());  
 }  
 bis.write((**"present:"** + **presentStudents** + **"total:"** + **totalStudents**).getBytes());  
 System.***out***.println(**"Export Success at : "** + recordFile.getPath());  
 JOptionPane.*showMessageDialog*(**null**, **"Export Success at : "** + recordFile.getPath(), **"Message"**, JOptionPane.***PLAIN\_MESSAGE***);  
 } **catch** (SQLException | FileNotFoundException ex) {  
 ex.printStackTrace();  
 } **catch** (IOException ex) {  
 ex.printStackTrace();  
 }  
 }  
  
 **private void** initComponents() {  
 *// JFormDesigner - Component initialization - DO NOT MODIFY //GEN-BEGIN:initComponents* **menuBar1** = **new** JMenuBar();  
 **menu1** = **new** JMenu();  
 **changePassword** = **new** JMenuItem();  
 **ExportSignupTable** = **new** JMenuItem();  
 **scrollPane1** = **new** JScrollPane();  
 **list1** = **new** JList();  
 **label1** = **new** JLabel();  
 **label2** = **new** JLabel();  
 **label3** = **new** JLabel();  
 **comfigBtn** = **new** JButton();  
 **lesson\_times** = **new** JTextField();  
 **total\_students** = **new** JTextField();  
 **label4** = **new** JLabel();  
 **content\_lb9** = **new** JLabel();  
 **scrollPane2** = **new** JScrollPane();  
 **textArea1** = **new** JTextArea();  
 **sentBtn** = **new** JButton();  
 **endBtn** = **new** JButton();  
 **presentInfo** = **new** JLabel();  
 **presentInfo2** = **new** JLabel();  
  
 *//======== this ========* setTitle(**"Server"**);  
 setDefaultCloseOperation(WindowConstants.***EXIT\_ON\_CLOSE***);  
 setResizable(**false**);  
 Container contentPane = getContentPane();  
  
 *//======== menuBar1 ========* {  
  
 *//======== menu1 ========* {  
 **menu1**.setText(**"Tools"**);  
  
 *//---- changePassword ----* **changePassword**.setText(**"Change Password"**);  
 **changePassword**.addMouseListener(**new** MouseAdapter() {  
 @Override  
 **public void** mouseClicked(MouseEvent e) {  
 changePasswordMouseClicked(e);  
 }  
 });  
 **changePassword**.addActionListener(**new** ActionListener() {  
 @Override  
 **public void** actionPerformed(ActionEvent e) {  
 changePasswordActionPerformed(e);  
 }  
 });  
 **menu1**.add(**changePassword**);  
  
 *//---- ExportSignupTable ----* **ExportSignupTable**.setText(**"Export Signup Table"**);  
 **ExportSignupTable**.addMouseListener(**new** MouseAdapter() {  
 @Override  
 **public void** mouseClicked(MouseEvent e) {  
 ExportSignupTableMouseClicked(e);  
 }  
 });  
 **ExportSignupTable**.addActionListener(**new** ActionListener() {  
 @Override  
 **public void** actionPerformed(ActionEvent e) {  
 ExportSignupTableActionPerformed(e);  
 }  
 });  
 **menu1**.add(**ExportSignupTable**);  
 }  
 **menuBar1**.add(**menu1**);  
 }  
 setJMenuBar(**menuBar1**);  
  
 *//======== scrollPane1 ========* {  
 **scrollPane1**.setViewportView(**list1**);  
 }  
  
 *//---- label1 ----* **label1**.setIcon(**new** ImageIcon(getClass().getResource(**"/resource/head.png"**)));  
  
 *//---- label2 ----* **label2**.setText(**"Lesson Times"**);  
 **label2**.setFont(**new** Font(**"\u5fae\u8f6f\u96c5\u9ed1"**, Font.***BOLD***, 16));  
  
 *//---- label3 ----* **label3**.setText(**"Total Students"**);  
 **label3**.setFont(**new** Font(**"\u5fae\u8f6f\u96c5\u9ed1"**, Font.***BOLD***, 16));  
  
 *//---- comfigBtn ----* **comfigBtn**.setText(**"Config and start class"**);  
 **comfigBtn**.setFont(**comfigBtn**.getFont().deriveFont(**comfigBtn**.getFont().getSize() + 3f));  
 **comfigBtn**.addMouseListener(**new** MouseAdapter() {  
 @Override  
 **public void** mouseClicked(MouseEvent e) {  
 comfigBtnMouseClicked(e);  
 }  
 });  
  
 *//---- lesson\_times ----* **lesson\_times**.setFont(**lesson\_times**.getFont().deriveFont(**lesson\_times**.getFont().getSize() + 2f));  
  
 *//---- total\_students ----* **total\_students**.setFont(**total\_students**.getFont().deriveFont(**total\_students**.getFont().getSize() + 2f));  
  
 *//---- label4 ----* **label4**.setText(**"Powered By javax.Swing"**);  
 **label4**.setForeground(Color.***gray***);  
 **label4**.setFont(**new** Font(**"DejaVu Sans"**, Font.***ITALIC***, 10));  
  
 *//---- content\_lb9 ----* **content\_lb9**.setFont(**new** Font(**"MathJax\_Typewriter"**, Font.***PLAIN***, 20));  
 **content\_lb9**.setText(**"Broadcast"**);  
  
 *//======== scrollPane2 ========* {  
  
 *//---- textArea1 ----* **textArea1**.setEnabled(**false**);  
 **scrollPane2**.setViewportView(**textArea1**);  
 }  
  
 *//---- sentBtn ----* **sentBtn**.setText(**"Sent"**);  
 **sentBtn**.setFont(**sentBtn**.getFont().deriveFont(**sentBtn**.getFont().getSize() + 3f));  
 **sentBtn**.setEnabled(**false**);  
 **sentBtn**.addMouseListener(**new** MouseAdapter() {  
 @Override  
 **public void** mouseClicked(MouseEvent e) {  
 sentBtnMouseClicked(e);  
 }  
 });  
  
 *//---- endBtn ----* **endBtn**.setText(**"End Class"**);  
 **endBtn**.setFont(**endBtn**.getFont().deriveFont(**endBtn**.getFont().getSize() + 3f));  
 **endBtn**.setForeground(Color.***red***);  
 **endBtn**.setEnabled(**false**);  
 **endBtn**.setBackground(Color.***red***);  
 **endBtn**.addMouseListener(**new** MouseAdapter() {  
 @Override  
 **public void** mouseClicked(MouseEvent e) {  
 endBtnMouseClicked(e);  
 }  
 });  
  
 *//---- presentInfo ----* **presentInfo**.setFont(**presentInfo**.getFont().deriveFont(**presentInfo**.getFont().getSize() + 2f));  
 **presentInfo**.setText(**"present:10/total:31"**);  
  
 *//---- presentInfo2 ----* **presentInfo2**.setFont(**presentInfo2**.getFont().deriveFont(**presentInfo2**.getFont().getSize() + 2f));  
 **presentInfo2**.setText(**"\*:task competed"**);  
  
 GroupLayout contentPaneLayout = **new** GroupLayout(contentPane);  
 contentPane.setLayout(contentPaneLayout);  
 contentPaneLayout.setHorizontalGroup(  
 contentPaneLayout.createParallelGroup()  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addGap(238, 238, 238)  
 .addComponent(**label1**)  
 .addContainerGap(GroupLayout.***DEFAULT\_SIZE***, Short.***MAX\_VALUE***))  
 .addGroup(GroupLayout.Alignment.***TRAILING***, contentPaneLayout.createSequentialGroup()  
 .addGap(313, 313, 313)  
 .addGroup(contentPaneLayout.createParallelGroup()  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addComponent(**label4**)  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***RELATED***, 247, Short.***MAX\_VALUE***)  
 .addComponent(**endBtn**)  
 .addContainerGap())  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addComponent(**sentBtn**)  
 .addGap(84, 410, Short.***MAX\_VALUE***))))  
 .addGroup(GroupLayout.Alignment.***TRAILING***, contentPaneLayout.createSequentialGroup()  
 .addGap(93, 93, 93)  
 .addGroup(contentPaneLayout.createParallelGroup()  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addGap(11, 11, 11)  
 .addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.***TRAILING***)  
 .addComponent(**comfigBtn**)  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addGroup(contentPaneLayout.createParallelGroup()  
 .addComponent(**label3**)  
 .addComponent(**label2**))  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***UNRELATED***)  
 .addGroup(contentPaneLayout.createParallelGroup()  
 .addComponent(**lesson\_times**, GroupLayout.***PREFERRED\_SIZE***, 86, GroupLayout.***PREFERRED\_SIZE***)  
 .addComponent(**total\_students**, GroupLayout.***PREFERRED\_SIZE***, 86, GroupLayout.***PREFERRED\_SIZE***)))))  
 .addComponent(**content\_lb9**)  
 .addComponent(**scrollPane2**, GroupLayout.***PREFERRED\_SIZE***, 294, GroupLayout.***PREFERRED\_SIZE***))  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***RELATED***, 145, Short.***MAX\_VALUE***)  
 .addGroup(contentPaneLayout.createParallelGroup()  
 .addComponent(**scrollPane1**, GroupLayout.***PREFERRED\_SIZE***, 204, GroupLayout.***PREFERRED\_SIZE***)  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addGap(6, 6, 6)  
 .addGroup(contentPaneLayout.createParallelGroup()  
 .addComponent(**presentInfo2**)  
 .addComponent(**presentInfo**))))  
 .addGap(52, 52, 52))  
 );  
 contentPaneLayout.setVerticalGroup(  
 contentPaneLayout.createParallelGroup()  
 .addGroup(GroupLayout.Alignment.***TRAILING***, contentPaneLayout.createSequentialGroup()  
 .addContainerGap()  
 .addComponent(**label1**)  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***RELATED***)  
 .addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.***TRAILING***)  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.***BASELINE***)  
 .addComponent(**label2**)  
 .addComponent(**lesson\_times**, GroupLayout.***PREFERRED\_SIZE***, GroupLayout.***DEFAULT\_SIZE***, GroupLayout.***PREFERRED\_SIZE***))  
 .addGap(18, 18, 18)  
 .addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.***BASELINE***)  
 .addComponent(**label3**)  
 .addComponent(**total\_students**, GroupLayout.***PREFERRED\_SIZE***, GroupLayout.***DEFAULT\_SIZE***, GroupLayout.***PREFERRED\_SIZE***))  
 .addGap(18, 18, 18)  
 .addComponent(**comfigBtn**)  
 .addGap(18, 18, 18)  
 .addComponent(**content\_lb9**)  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***RELATED***)  
 .addComponent(**scrollPane2**, GroupLayout.***PREFERRED\_SIZE***, 73, GroupLayout.***PREFERRED\_SIZE***)  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***UNRELATED***)  
 .addComponent(**sentBtn**)  
 .addGap(50, 50, 50)  
 .addComponent(**label4**))  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addComponent(**scrollPane1**, GroupLayout.***PREFERRED\_SIZE***, 295, GroupLayout.***PREFERRED\_SIZE***)  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***RELATED***)  
 .addComponent(**presentInfo**)  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***RELATED***)  
 .addComponent(**presentInfo2**)  
 .addGap(11, 11, 11)  
 .addComponent(**endBtn**, GroupLayout.***PREFERRED\_SIZE***, 51, GroupLayout.***PREFERRED\_SIZE***)))  
 .addContainerGap(7, Short.***MAX\_VALUE***))  
 );  
 setSize(790, 555);  
 setLocationRelativeTo(getOwner());  
 *// JFormDesigner - End of component initialization //GEN-END:initComponents* }  
 **private void** run(){  
 **new** Thread(){  
 **public void** run(){  
 **while**(**true**) {  
 **presentStudents** = 0;  
 **multiClientName** = **multiSocket**.getTotalClientName();  
 **if** (!**multiClientName**.isEmpty()) {  
 DefaultListModel<String> listModel = **new** DefaultListModel<>();  
 **for**(Iterator<String> item = **multiClientName**.iterator(); item.hasNext(); ){  
 listModel.addElement(item.next());  
 **presentStudents**++;  
 }  
 **list1**.setModel(listModel);  
 }  
 **presentInfo**.setText(**"present:"** + **presentStudents** + **"total:"** + **totalStudents**);  
 }  
 }  
 }.start();  
 }  
 *// JFormDesigner - Variables declaration - DO NOT MODIFY //GEN-BEGIN:variables* **private** JMenuBar **menuBar1**;  
 **private** JMenu **menu1**;  
 **private** JMenuItem **changePassword**;  
 **private** JMenuItem **ExportSignupTable**;  
 **private** JScrollPane **scrollPane1**;  
 **private** JList **list1**;  
 **private** JLabel **label1**;  
 **private** JLabel **label2**;  
 **private** JLabel **label3**;  
 **private** JButton **comfigBtn**;  
 **private** JTextField **lesson\_times**;  
 **private** JTextField **total\_students**;  
 **private** JLabel **label4**;  
 **private** JLabel **content\_lb9**;  
 **private** JScrollPane **scrollPane2**;  
 **private** JTextArea **textArea1**;  
 **private** JButton **sentBtn**;  
 **private** JButton **endBtn**;  
 **private** JLabel **presentInfo**;  
 **private** JLabel **presentInfo2**;  
 *// JFormDesigner - End of variables declaration //GEN-END:variables*}  
**class** ChangePasswordForm **extends** JFrame {  
 **public** ChangePasswordForm() {  
 initComponents();  
 }  
  
 **private void** okButtonMouseClicked(MouseEvent e) {  
 String oldPassword = **old\_password**.getText();  
 String newPassword = **new\_password**.getText();  
 String newPasswordAgain = **new\_password\_again**.getText();  
 **if**(newPassword.equalsIgnoreCase(newPasswordAgain)){  
 **try**{  
 System.***out***.println(ServerLogin.*superPassword*);  
 ServerLogin.*changePassword*(newPassword, oldPassword);  
 } **catch** (SQLException ex) {  
 ex.printStackTrace();  
 }  
 }  
 **else**{  
 JOptionPane.*showMessageDialog*(**this**, **"Two New Password Doesn't Match"**, **"Alert"**, JOptionPane.***WARNING\_MESSAGE***);  
 }  
 }  
  
 **private void** cancelButtonMouseClicked(MouseEvent e) {  
 **this**.dispose();  
 }  
  
 **private void** initComponents() {  
 *// JFormDesigner - Component initialization - DO NOT MODIFY //GEN-BEGIN:initComponents* **dialogPane** = **new** JPanel();  
 **contentPanel** = **new** JPanel();  
 **label1** = **new** JLabel();  
 **old\_password** = **new** JTextField();  
 **new\_password** = **new** JTextField();  
 **label2** = **new** JLabel();  
 **new\_password\_again** = **new** JTextField();  
 **label3** = **new** JLabel();  
 **buttonBar** = **new** JPanel();  
 **okButton** = **new** JButton();  
 **cancelButton** = **new** JButton();  
  
 *//======== this ========* Container contentPane = getContentPane();  
 contentPane.setLayout(**new** BorderLayout());  
  
 *//======== dialogPane ========* {  
 **dialogPane**.setBorder(**new** EmptyBorder(12, 12, 12, 12));  
 **dialogPane**.setLayout(**new** BorderLayout());  
  
 *//======== contentPanel ========* {  
  
 *//---- label1 ----* **label1**.setText(**"Old Password"**);  
 **label1**.setFont(**label1**.getFont().deriveFont(**label1**.getFont().getSize() + 3f));  
  
 *//---- label2 ----* **label2**.setText(**"New Password"**);  
 **label2**.setFont(**label2**.getFont().deriveFont(**label2**.getFont().getSize() + 3f));  
  
 *//---- label3 ----* **label3**.setText(**"New Password Again"**);  
 **label3**.setFont(**label3**.getFont().deriveFont(**label3**.getFont().getSize() + 3f));  
  
 GroupLayout contentPanelLayout = **new** GroupLayout(**contentPanel**);  
 **contentPanel**.setLayout(contentPanelLayout);  
 contentPanelLayout.setHorizontalGroup(  
 contentPanelLayout.createParallelGroup()  
 .addGroup(contentPanelLayout.createSequentialGroup()  
 .addGap(129, 129, 129)  
 .addGroup(contentPanelLayout.createParallelGroup(GroupLayout.Alignment.***TRAILING***, **false**)  
 .addGroup(contentPanelLayout.createSequentialGroup()  
 .addComponent(**label1**)  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***RELATED***, GroupLayout.***DEFAULT\_SIZE***, Short.***MAX\_VALUE***)  
 .addComponent(**old\_password**, GroupLayout.***PREFERRED\_SIZE***, 174, GroupLayout.***PREFERRED\_SIZE***))  
 .addGroup(GroupLayout.Alignment.***LEADING***, contentPanelLayout.createSequentialGroup()  
 .addComponent(**label2**)  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***RELATED***, GroupLayout.***DEFAULT\_SIZE***, Short.***MAX\_VALUE***)  
 .addComponent(**new\_password**, GroupLayout.***PREFERRED\_SIZE***, 174, GroupLayout.***PREFERRED\_SIZE***))  
 .addGroup(GroupLayout.Alignment.***LEADING***, contentPanelLayout.createSequentialGroup()  
 .addComponent(**label3**)  
 .addGap(18, 18, 18)  
 .addComponent(**new\_password\_again**, GroupLayout.***PREFERRED\_SIZE***, 174, GroupLayout.***PREFERRED\_SIZE***)))  
 .addContainerGap(149, Short.***MAX\_VALUE***))  
 );  
 contentPanelLayout.setVerticalGroup(  
 contentPanelLayout.createParallelGroup()  
 .addGroup(contentPanelLayout.createSequentialGroup()  
 .addGap(125, 125, 125)  
 .addGroup(contentPanelLayout.createParallelGroup(GroupLayout.Alignment.***BASELINE***)  
 .addComponent(**old\_password**, GroupLayout.***PREFERRED\_SIZE***, GroupLayout.***DEFAULT\_SIZE***, GroupLayout.***PREFERRED\_SIZE***)  
 .addComponent(**label1**))  
 .addGap(18, 18, 18)  
 .addGroup(contentPanelLayout.createParallelGroup(GroupLayout.Alignment.***BASELINE***)  
 .addComponent(**new\_password**, GroupLayout.***PREFERRED\_SIZE***, GroupLayout.***DEFAULT\_SIZE***, GroupLayout.***PREFERRED\_SIZE***)  
 .addComponent(**label2**))  
 .addGap(18, 18, 18)  
 .addGroup(contentPanelLayout.createParallelGroup()  
 .addGroup(contentPanelLayout.createSequentialGroup()  
 .addGap(1, 1, 1)  
 .addComponent(**label3**))  
 .addComponent(**new\_password\_again**, GroupLayout.***PREFERRED\_SIZE***, GroupLayout.***DEFAULT\_SIZE***, GroupLayout.***PREFERRED\_SIZE***))  
 .addContainerGap(160, Short.***MAX\_VALUE***))  
 );  
 }  
 **dialogPane**.add(**contentPanel**, BorderLayout.***CENTER***);  
  
 *//======== buttonBar ========* {  
 **buttonBar**.setBorder(**new** EmptyBorder(12, 0, 0, 0));  
 **buttonBar**.setLayout(**new** GridBagLayout());  
 ((GridBagLayout)**buttonBar**.getLayout()).**columnWidths** = **new int**[] {0, 85, 80};  
 ((GridBagLayout)**buttonBar**.getLayout()).**columnWeights** = **new double**[] {1.0, 0.0, 0.0};  
  
 *//---- okButton ----* **okButton**.setText(**"OK"**);  
 **okButton**.addMouseListener(**new** MouseAdapter() {  
 @Override  
 **public void** mouseClicked(MouseEvent e) {  
 okButtonMouseClicked(e);  
 }  
 });  
 **buttonBar**.add(**okButton**, **new** GridBagConstraints(1, 0, 1, 1, 0.0, 0.0,  
 GridBagConstraints.***CENTER***, GridBagConstraints.***BOTH***,  
 **new** Insets(0, 0, 0, 5), 0, 0));  
  
 *//---- cancelButton ----* **cancelButton**.setText(**"Cancel"**);  
 **cancelButton**.addMouseListener(**new** MouseAdapter() {  
 @Override  
 **public void** mouseClicked(MouseEvent e) {  
 cancelButtonMouseClicked(e);  
 }  
 });  
 **buttonBar**.add(**cancelButton**, **new** GridBagConstraints(2, 0, 1, 1, 0.0, 0.0,  
 GridBagConstraints.***CENTER***, GridBagConstraints.***BOTH***,  
 **new** Insets(0, 0, 0, 0), 0, 0));  
 }  
 **dialogPane**.add(**buttonBar**, BorderLayout.***SOUTH***);  
 }  
 contentPane.add(**dialogPane**, BorderLayout.***CENTER***);  
 pack();  
 setLocationRelativeTo(getOwner());  
 *// JFormDesigner - End of component initialization //GEN-END:initComponents* }  
  
 *// JFormDesigner - Variables declaration - DO NOT MODIFY //GEN-BEGIN:variables* **private** JPanel **dialogPane**;  
 **private** JPanel **contentPanel**;  
 **private** JLabel **label1**;  
 **private** JTextField **old\_password**;  
 **private** JTextField **new\_password**;  
 **private** JLabel **label2**;  
 **private** JTextField **new\_password\_again**;  
 **private** JLabel **label3**;  
 **private** JPanel **buttonBar**;  
 **private** JButton **okButton**;  
 **private** JButton **cancelButton**;  
 *// JFormDesigner - End of variables declaration //GEN-END:variables*}

#### 2.4.5 ClientLogin.java

*/\*  
 \* Created by JFormDesigner on Thu Dec 12 17:13:13 CST 2019  
 \*/***package** view;  
  
**import** com.sun.security.ntlm.Client;  
  
**import** java.awt.\*;  
**import** java.awt.event.\*;  
**import** java.io.\*;  
**import** java.net.Socket;  
**import** java.net.SocketException;  
**import** java.net.UnknownHostException;  
**import** java.nio.Buffer;  
**import** javax.swing.\*;  
**import** javax.swing.GroupLayout;  
**import** javax.swing.border.\*;  
  
*/\*\*  
 \** ***@author*** *jiengup  
 \*/***public class** ClientLogin **extends** JFrame {  
 **private static** String *address* = **"127.0.0.1"**;  
 **private static int** *port* = 9090;  
 **private** Socket **socket**;  
 **private** String **userName** = **""**;  
 **private char**[] **passWord**;  
 **private** String **messageFromServer**;  
 **static** PrintWriter *outStream*;  
 **static** BufferedReader *inStream*;  
 **private** String **lessonTimes**;  
  
 **public** ClientLogin() {  
 initComponents();  
 }  
  
 **private void** loginButtonMouseClicked(MouseEvent e){  
 **try**{  
 System.***out***.println(**"Connecting to "** + *address* + **" on port "** + *port*);  
 **socket** = **new** Socket(*address*, *port*);  
 System.***out***.println(**"Just Connected to "** + **socket**.getInetAddress());  
  
 *inStream* = **new** BufferedReader(**new** InputStreamReader(**socket**.getInputStream()));  
 *outStream* = **new** PrintWriter(**socket**.getOutputStream());  
 **lessonTimes** = *inStream*.readLine();  
 System.***out***.println(**"Received Lesson Times : "** + **lessonTimes**);  
 } **catch** (UnknownHostException ex) {  
 ex.printStackTrace();  
 JOptionPane.*showMessageDialog*(**null**, **"CONNECT FAILED!\nPLEASE CHECK SERVER"**, **"Alert"**, JOptionPane.***ERROR\_MESSAGE***);  
 System.*exit*(0);  
 } **catch** (IOException ex) {  
 ex.printStackTrace();  
 JOptionPane.*showMessageDialog*(**null**, **"CONNECT FAILED!\nPLEASE CHECK SERVER"**, **"Alert"**, JOptionPane.***ERROR\_MESSAGE***);  
 System.*exit*(0);  
 }  
 **userName** = **user\_name**.getText();  
 **passWord** = **user\_passwd**.getPassword();  
 String passWordTest = **new** String(**passWord**);  
 **if**(**userName**.equalsIgnoreCase(**""**)){  
 JOptionPane.*showMessageDialog*(**this**, **"Please input your user name"**, **"Alert"**, JOptionPane.***WARNING\_MESSAGE***);  
 }  
 **else**{  
 **try**{  
 *outStream*.println(**"Login Request"**);  
 *outStream*.flush();  
 *outStream*.println(**userName**);  
 *outStream*.flush();  
 *outStream*.println(passWordTest);  
 *outStream*.flush();  
 **messageFromServer** = *inStream*.readLine();  
 System.***out***.println(**"Receive from server : "** + **messageFromServer**);  
 **if**(**messageFromServer**.equals(**"Login Passed"**)){  
 System.***out***.println(**"Login successfully"**);  
 **final** String userName = *inStream*.readLine();  
 EventQueue.*invokeLater*(**new** Runnable() {  
 @Override  
 **public void** run() {  
 ClientAfterLogin cal = **new** ClientAfterLogin(**socket**, userName, **lessonTimes**);  
 cal.setVisible(**true**);  
 cal.run();  
 }  
 });  
 **this**.dispose();  
 }  
 **else if**(**messageFromServer**.equals(**"Login Failed"**)){  
 System.***out***.println(**"Login failed"**);  
 }  
 } **catch** (Exception ex) {  
 ex.printStackTrace();  
 }  
 }  
 }  
  
 **private void** cancelButtonMouseClicked(MouseEvent e) {  
 **try**{  
 *outStream*.println(**"exit"**);  
 *outStream*.flush();  
 *inStream*.close();  
 *outStream*.close();  
 } **catch** (IOException ex) {  
 ex.printStackTrace();  
 }  
 System.*exit*(0);  
 }  
  
  
 **private void** thisWindowClosing(WindowEvent e) {  
 **try**{  
 *outStream*.println(**"exit"**);  
 *outStream*.flush();  
 *inStream*.close();  
 *outStream*.close();  
 } **catch** (IOException ex) {  
 ex.printStackTrace();  
 }  
 System.*exit*(0);  
 }  
  
  
 **private void** initComponents() {  
 *// JFormDesigner - Component initialization - DO NOT MODIFY //GEN-BEGIN:initComponents* **dialogPane** = **new** JPanel();  
 **contentPanel** = **new** JPanel();  
 **usrname\_lb** = **new** JLabel();  
 **user\_name** = **new** JTextField();  
 **passwd\_lb** = **new** JLabel();  
 **user\_passwd** = **new** JPasswordField();  
 **title\_lb** = **new** JLabel();  
 **javalogo\_lb** = **new** JLabel();  
 **buttonBar** = **new** JPanel();  
 **loginButton** = **new** JButton();  
 **cancelButton** = **new** JButton();  
  
 *//======== this ========* setDefaultCloseOperation(WindowConstants.***EXIT\_ON\_CLOSE***);  
 setResizable(**false**);  
 setTitle(**"Client Login"**);  
 addWindowListener(**new** WindowAdapter() {  
 @Override  
 **public void** windowClosing(WindowEvent e) {  
 thisWindowClosing(e);  
 }  
 });  
 Container contentPane = getContentPane();  
 contentPane.setLayout(**new** BorderLayout());  
  
 *//======== dialogPane ========* {  
 dialogPane.setBorder(**new** EmptyBorder(12, 12, 12, 12));  
 dialogPane.setLayout(**new** BorderLayout());  
  
 *//======== contentPanel ========* {  
  
 *//---- usrname\_lb ----* usrname\_lb.setText(**"User name:"**);  
 usrname\_lb.setFont(**new** Font(**"\u5fae\u8f6f\u96c5\u9ed1"**, Font.PLAIN, 17));  
  
 *//---- user\_name ----* user\_name.setToolTipText(**"your user name here"**);  
 user\_name.setFont(user\_name.getFont().deriveFont(user\_name.getFont().getStyle() | Font.ITALIC));  
  
 *//---- passwd\_lb ----* passwd\_lb.setText(**"Password:"**);  
 passwd\_lb.setFont(**new** Font(**"\u5fae\u8f6f\u96c5\u9ed1"**, Font.PLAIN, 17));  
  
 *//---- user\_passwd ----* user\_passwd.setToolTipText(**"your password here"**);  
 user\_passwd.setFont(user\_passwd.getFont().deriveFont(user\_passwd.getFont().getStyle() | Font.ITALIC));  
  
 *//---- title\_lb ----* title\_lb.setText(**"Welcome To The Java Class"**);  
 title\_lb.setFont(**new** Font(**"MathJax\_Typewriter"**, Font.BOLD, 22));  
 title\_lb.setForeground(Color.darkGray);  
  
 *//---- javalogo\_lb ----* javalogo\_lb.setIcon(**new** ImageIcon(getClass().getResource(**"/resource/javalogo2.png"**)));  
  
 GroupLayout contentPanelLayout = **new** GroupLayout(contentPanel);  
 contentPanel.setLayout(contentPanelLayout);  
 contentPanelLayout.setHorizontalGroup(  
 contentPanelLayout.createParallelGroup()  
 .addGroup(contentPanelLayout.createSequentialGroup()  
 .addGroup(contentPanelLayout.createParallelGroup()  
 .addGroup(contentPanelLayout.createSequentialGroup()  
 .addGap(123, 123, 123)  
 .addComponent(javalogo\_lb)  
 .addPreferredGap(LayoutStyle.ComponentPlacement.UNRELATED)  
 .addComponent(title\_lb, GroupLayout.PREFERRED\_SIZE, 299, GroupLayout.PREFERRED\_SIZE))  
 .addGroup(contentPanelLayout.createSequentialGroup()  
 .addGap(101, 101, 101)  
 .addGroup(contentPanelLayout.createParallelGroup()  
 .addComponent(usrname\_lb)  
 .addComponent(passwd\_lb, GroupLayout.PREFERRED\_SIZE, 91, GroupLayout.PREFERRED\_SIZE))  
 .addGap(18, 18, 18)  
 .addGroup(contentPanelLayout.createParallelGroup(GroupLayout.Alignment.LEADING, **false**)  
 .addComponent(user\_name, GroupLayout.DEFAULT\_SIZE, 280, Short.MAX\_VALUE)  
 .addComponent(user\_passwd, GroupLayout.DEFAULT\_SIZE, 280, Short.MAX\_VALUE))))  
 .addContainerGap(99, Short.MAX\_VALUE))  
 );  
 contentPanelLayout.setVerticalGroup(  
 contentPanelLayout.createParallelGroup()  
 .addGroup(GroupLayout.Alignment.TRAILING, contentPanelLayout.createSequentialGroup()  
 .addGap(25, 25, 25)  
 .addGroup(contentPanelLayout.createParallelGroup()  
 .addGroup(GroupLayout.Alignment.TRAILING, contentPanelLayout.createSequentialGroup()  
 .addComponent(title\_lb)  
 .addGap(11, 11, 11))  
 .addComponent(javalogo\_lb, GroupLayout.Alignment.TRAILING))  
 .addGap(80, 80, 80)  
 .addGroup(contentPanelLayout.createParallelGroup(GroupLayout.Alignment.BASELINE)  
 .addComponent(usrname\_lb)  
 .addComponent(user\_name, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))  
 .addGap(21, 21, 21)  
 .addGroup(contentPanelLayout.createParallelGroup(GroupLayout.Alignment.BASELINE)  
 .addComponent(passwd\_lb)  
 .addComponent(user\_passwd, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))  
 .addContainerGap(123, Short.MAX\_VALUE))  
 );  
 }  
 dialogPane.add(contentPanel, BorderLayout.CENTER);  
  
 *//======== buttonBar ========* {  
 buttonBar.setBorder(**new** EmptyBorder(12, 0, 0, 0));  
 buttonBar.setLayout(**new** GridBagLayout());  
 ((GridBagLayout)buttonBar.getLayout()).columnWidths = **new int**[] {0, 85, 80};  
 ((GridBagLayout)buttonBar.getLayout()).columnWeights = **new double**[] {1.0, 0.0, 0.0};  
  
 *//---- loginButton ----* loginButton.setText(**"Sign up"**);  
 loginButton.setIcon(**new** ImageIcon(getClass().getResource(**"/resource/login.png"**)));  
 loginButton.setFont(loginButton.getFont().deriveFont(loginButton.getFont().getSize() + 3f));  
 loginButton.addMouseListener(**new** MouseAdapter() {  
 @Override  
 **public void** mouseClicked(MouseEvent e) {  
 loginButtonMouseClicked(e);  
 }  
 });  
 buttonBar.add(loginButton, **new** GridBagConstraints(1, 0, 1, 1, 0.0, 0.0,  
 GridBagConstraints.CENTER, GridBagConstraints.BOTH,  
 **new** Insets(0, 0, 0, 5), 0, 0));  
  
 *//---- cancelButton ----* cancelButton.setText(**"Cancel"**);  
 cancelButton.setIcon(**new** ImageIcon(getClass().getResource(**"/resource/cancel.png"**)));  
 cancelButton.setFont(cancelButton.getFont().deriveFont(cancelButton.getFont().getSize() + 3f));  
 cancelButton.addMouseListener(**new** MouseAdapter() {  
 @Override  
 **public void** mouseClicked(MouseEvent e) {  
 cancelButtonMouseClicked(e);  
 }  
 });  
 buttonBar.add(cancelButton, **new** GridBagConstraints(2, 0, 1, 1, 0.0, 0.0,  
 GridBagConstraints.CENTER, GridBagConstraints.BOTH,  
 **new** Insets(0, 0, 0, 0), 0, 0));  
 }  
 dialogPane.add(buttonBar, BorderLayout.SOUTH);  
 }  
 contentPane.add(dialogPane, BorderLayout.CENTER);  
 pack();  
 setLocationRelativeTo(**null**);  
 *// JFormDesigner - End of component initialization //GEN-END:initComponents* }  
  
 **public static void** main(String args[]){  
 changeLookAndFeel();  
  
 EventQueue.invokeLater(**new** Runnable() {  
 @Override  
 **public void** run() {  
 **new** ClientLogin().setVisible(**true**);  
 }  
 });  
 }  
 **static void** changeLookAndFeel() {  
 **try**{  
 **for**(UIManager.LookAndFeelInfo info : UIManager.getInstalledLookAndFeels()){  
 **if**(**"Windows"**.equals(info.getName())){  
 UIManager.setLookAndFeel(info.getClassName());  
 **break**;  
 }  
 }  
 }**catch**(ClassNotFoundException e){  
 e.printStackTrace();  
 }**catch** (InstantiationException e){  
 e.printStackTrace();  
 }**catch** (IllegalAccessException e) {  
 e.printStackTrace();  
 }**catch** (UnsupportedLookAndFeelException e) {  
 e.printStackTrace();  
 }  
 }  
  
 *// JFormDesigner - Variables declaration - DO NOT MODIFY //GEN-BEGIN:variables* **private** JPanel dialogPane;  
 **private** JPanel contentPanel;  
 **private** JLabel usrname\_lb;  
 **private** JTextField user\_name;  
 **private** JLabel passwd\_lb;  
 **private** JPasswordField user\_passwd;  
 **private** JLabel title\_lb;  
 **private** JLabel javalogo\_lb;  
 **private** JPanel buttonBar;  
 **private** JButton loginButton;  
 **private** JButton cancelButton;  
 *// JFormDesigner - End of variables declaration //GEN-END:variables*}

#### 2.4.6 ClientAfterLogin.java

*/\*  
 \* Created by JFormDesigner on Fri Dec 13 18:47:00 CST 2019  
 \*/***package** view;  
  
**import** java.awt.\*;  
**import** java.awt.event.\*;  
**import** java.io.\*;  
**import** java.net.Socket;  
**import** java.nio.file.FileSystem;  
**import** java.text.SimpleDateFormat;  
**import** java.util.Date;  
**import** javax.swing.\*;  
**import** javax.swing.GroupLayout;  
**import** javax.swing.filechooser.FileSystemView;  
  
*/\*\*  
 \** ***@author*** *jiengup  
 \*/***public class** ClientAfterLogin **extends** JFrame{  
 **private** Socket **socket**;  
 **private** String **userName**;  
 **private** PrintWriter **outStream**;  
 **private** BufferedReader **inStream**;  
 **private** String **filePath**;  
 **private** String **lessonTimes**;  
 **private boolean keepgoing**;  
 **public** ClientAfterLogin(Socket skt, String name, String classTimes) {  
 **socket** = skt;  
 **lessonTimes** = classTimes;  
 System.***out***.println(classTimes);  
 initComponents();  
 **try**{  
 **outStream** = **new** PrintWriter(**socket**.getOutputStream());  
 **inStream** = **new** BufferedReader(**new** InputStreamReader(**socket**.getInputStream()));  
 } **catch** (IOException e) {  
 JOptionPane.*showMessageDialog*(**this**, **"CONNECT FAILED!\nPLEASE CHECK SERVER"**, **"Alert"**, JOptionPane.***ERROR\_MESSAGE***);  
 **keepgoing** = **false**;  
 e.printStackTrace();  
 System.*exit*(0);  
 }  
 **userName** = name;  
 **welcome\_lb2**.setText(**userName**);  
 **if**(**lessonTimes**.equalsIgnoreCase(**"1"**)) **lessonTimes** += **"st"**;  
 **else if**(**lessonTimes**.equalsIgnoreCase(**"2"**)) **lessonTimes** += **"ed"**;  
 **else if**(**lessonTimes**.equalsIgnoreCase(**"3"**)) **lessonTimes** += **"rd"**;  
 **else lessonTimes** += **"th"**;  
 **content\_lb4**.setText(**lessonTimes**);  
 }  
  
 **private void** thisWindowClosing(WindowEvent e) {  
 **try**{  
 **outStream**.println(**"exit"**);  
 **outStream**.flush();  
 **inStream**.close();  
 **outStream**.close();  
 } **catch** (IOException ex) {  
 ex.printStackTrace();  
 }  
 **keepgoing** = **false**;  
 System.*exit*(0);  
 }  
  
 **private void** submitButtonMouseClicked(MouseEvent e) {  
 *//Get Desktop;* FileSystemView fsv = FileSystemView.*getFileSystemView*();  
 **filePath** = **""**;  
 JFileChooser selectFile = **new** JFileChooser(fsv.getHomeDirectory());  
 **if**(selectFile.showOpenDialog(**this**) == selectFile.***APPROVE\_OPTION***){  
 **filePath** = selectFile.getSelectedFile().getPath();  
 }  
 **if**(!**filePath**.equalsIgnoreCase(**""**)){  
 **outStream**.println(**"sendfile"**);  
 **outStream**.flush();  
 File src = **new** File(**filePath**);  
 String fileName = src.getName();  
 **long** size = src.length();  
 **try**{  
 String suffix = fileName.substring(fileName.lastIndexOf(**"."**) + 1);  
 DataOutputStream dos =**new** DataOutputStream(**socket**.getOutputStream());  
 dos.writeLong(size);  
 dos.writeUTF(suffix);  
 BufferedInputStream bis = **new** BufferedInputStream(**new** FileInputStream(src));  
 BufferedOutputStream bos = **new** BufferedOutputStream(**socket**.getOutputStream());  
 **long** writeSize = 0;  
 **while**(**true**){  
 bos.write(bis.read());  
 writeSize++;  
 **if**(writeSize > size){  
 **break**;  
 }  
 }  
 bos.flush();  
 **fileSendMessage**.setText(**"File Submitted: "** + **filePath**);  
 **fileSendStatu**.setIcon(**new** ImageIcon(getClass().getResource(**"/resource/sendpass.png"**)));  
 } **catch** (IOException ex) {  
 JOptionPane.*showMessageDialog*(**this**, **"CONNECT FAILED!\nPLEASE CHECK SERVER"**, **"Alert"**, JOptionPane.***ERROR\_MESSAGE***);  
 **keepgoing** = **false**;  
 ex.printStackTrace();  
 System.*exit*(0);  
 ex.printStackTrace();  
 }  
 }  
 }  
  
 **private void** checkBox1ActionPerformed(ActionEvent e) {  
 **if**(**checkBox1**.isSelected()){  
 **logoutButton**.setEnabled(**true**);  
 }  
 **else**{  
 **logoutButton**.setEnabled(**false**);  
 }  
 }  
  
 **private void** logoutButtonMouseClicked(MouseEvent e) {  
 **if**(!**logoutButton**.isEnabled()) **return**;  
 **else**{  
 **try**{  
 **outStream**.println(**"logout"**);  
 **outStream**.flush();  
 **inStream**.close();  
 **outStream**.close();  
 } **catch** (IOException ex) {  
 ex.printStackTrace();  
 }  
 **keepgoing** = **false**;  
 System.*exit*(0);  
 }  
 }  
  
 **private void** initComponents() {  
 *// JFormDesigner - Component initialization - DO NOT MODIFY //GEN-BEGIN:initComponents* **haad\_img** = **new** JLabel();  
 **welcome\_lb1** = **new** JLabel();  
 **welcome\_lb2** = **new** JLabel();  
 **submitButton** = **new** JButton();  
 **content\_lb1** = **new** JLabel();  
 **content\_lb2** = **new** JLabel();  
 **content\_lb3** = **new** JLabel();  
 **content\_lb4** = **new** JLabel();  
 **content\_lb5** = **new** JLabel();  
 **content\_lb8** = **new** JLabel();  
 **content\_lb6** = **new** JLabel();  
 **content\_lb7** = **new** JLabel();  
 **label2** = **new** JLabel();  
 **logoutButton** = **new** JButton();  
 **label3** = **new** JLabel();  
 **checkBox1** = **new** JCheckBox();  
 **scrollPane1** = **new** JScrollPane();  
 **textPane1** = **new** JTextPane();  
 **content\_lb9** = **new** JLabel();  
 **fileSendMessage** = **new** JLabel();  
 **fileSendStatu** = **new** JLabel();  
  
 *//======== this ========* setTitle(**"Client"**);  
 setDefaultCloseOperation(WindowConstants.***EXIT\_ON\_CLOSE***);  
 setResizable(**false**);  
 addWindowListener(**new** WindowAdapter() {  
 @Override  
 **public void** windowClosing(WindowEvent e) {  
 thisWindowClosing(e);  
 }  
 });  
 Container contentPane = getContentPane();  
  
 *//---- haad\_img ----* **haad\_img**.setIcon(**new** ImageIcon(getClass().getResource(**"/resource/head.png"**)));  
  
 *//---- welcome\_lb1 ----* **welcome\_lb1**.setText(**"Welcome, "**);  
 **welcome\_lb1**.setFont(**new** Font(**"MathJax\_Typewriter"**, Font.***PLAIN***, 20));  
  
 *//---- welcome\_lb2 ----* **welcome\_lb2**.setFont(**new** Font(**"MathJax\_Typewriter"**, Font.***PLAIN***, 20));  
 **welcome\_lb2**.setText(**"Jengup"**);  
  
 *//---- submitButton ----* **submitButton**.setText(**"Submit"**);  
 **submitButton**.setToolTipText(**"select a file"**);  
 **submitButton**.addMouseListener(**new** MouseAdapter() {  
 @Override  
 **public void** mouseClicked(MouseEvent e) {  
 submitButtonMouseClicked(e);  
 }  
 });  
  
 *//---- content\_lb1 ----* **content\_lb1**.setFont(**new** Font(**"MathJax\_Typewriter"**, Font.***PLAIN***, 20));  
 **content\_lb1**.setText(**"You hava signup successfully!"**);  
  
 *//---- content\_lb2 ----* **content\_lb2**.setFont(**new** Font(**"MathJax\_Typewriter"**, Font.***PLAIN***, 20));  
 **content\_lb2**.setText(**"This is your"**);  
  
 *//---- content\_lb3 ----* **content\_lb3**.setFont(**new** Font(**"MathJax\_Typewriter"**, Font.***PLAIN***, 20));  
 **content\_lb3**.setText(**"From now on, hand on your task!"**);  
  
 *//---- content\_lb4 ----* **content\_lb4**.setFont(**new** Font(**"MathJax\_Typewriter"**, Font.***BOLD*** | Font.***ITALIC***, 20));  
 **content\_lb4**.setText(**"3rd "**);  
  
 *//---- content\_lb5 ----* **content\_lb5**.setFont(**new** Font(**"MathJax\_Typewriter"**, Font.***PLAIN***, 20));  
 **content\_lb5**.setText(**"Java Class"**);  
  
 *//---- content\_lb8 ----* **content\_lb8**.setIcon(**new** ImageIcon(getClass().getResource(**"/resource/arrow.png"**)));  
  
 *//---- content\_lb6 ----* **content\_lb6**.setFont(**new** Font(**"MathJax\_Typewriter"**, Font.***PLAIN***, 20));  
 **content\_lb6**.setText(**"If you compete:"**);  
  
 *//---- content\_lb7 ----* **content\_lb7**.setFont(**new** Font(**"MathJax\_Typewriter"**, Font.***PLAIN***, 23));  
 **content\_lb7**.setText(**"If you have competed"**);  
  
 *//---- label2 ----* **label2**.setText(**"(Please select sigle file, if you have muiti-files, compress them.)"**);  
 **label2**.setForeground(Color.***darkGray***);  
 **label2**.setFont(**new** Font(**"DejaVu Sans"**, Font.***ITALIC***, 12));  
  
 *//---- logoutButton ----* **logoutButton**.setText(**"Log Out"**);  
 **logoutButton**.setBackground(Color.***red***);  
 **logoutButton**.setForeground(Color.***red***);  
 **logoutButton**.setEnabled(**false**);  
 **logoutButton**.addMouseListener(**new** MouseAdapter() {  
 @Override  
 **public void** mouseClicked(MouseEvent e) {  
 logoutButtonMouseClicked(e);  
 }  
 });  
  
 *//---- label3 ----* **label3**.setText(**"Powered By javax.Swing"**);  
 **label3**.setForeground(Color.***gray***);  
 **label3**.setFont(**new** Font(**"DejaVu Sans"**, Font.***ITALIC***, 10));  
  
 *//---- checkBox1 ----* **checkBox1**.setText(**"I'm sure I have submitted all the tasks"**);  
 **checkBox1**.addActionListener(**new** ActionListener() {  
 @Override  
 **public void** actionPerformed(ActionEvent e) {  
 checkBox1ActionPerformed(e);  
 }  
 });  
  
 *//======== scrollPane1 ========* {  
  
 *//---- textPane1 ----* **textPane1**.setEditable(**false**);  
 **textPane1**.setText(**"\u8bf7\u5c3d\u5feb\u5b8c\u6210\u5b9e\u9a8c"**);  
 **textPane1**.setFont(**textPane1**.getFont().deriveFont(**textPane1**.getFont().getSize() + 3f));  
 **scrollPane1**.setViewportView(**textPane1**);  
 }  
  
 *//---- content\_lb9 ----* **content\_lb9**.setFont(**new** Font(**"MathJax\_Typewriter"**, Font.***PLAIN***, 20));  
 **content\_lb9**.setText(**"Message from teacher"**);  
  
 *//---- fileSendMessage ----* **fileSendMessage**.setFont(**fileSendMessage**.getFont().deriveFont(**fileSendMessage**.getFont().getSize() - 2f));  
  
 GroupLayout contentPaneLayout = **new** GroupLayout(contentPane);  
 contentPane.setLayout(contentPaneLayout);  
 contentPaneLayout.setHorizontalGroup(  
 contentPaneLayout.createParallelGroup()  
 .addGroup(GroupLayout.Alignment.***TRAILING***, contentPaneLayout.createSequentialGroup()  
 .addContainerGap(241, Short.***MAX\_VALUE***)  
 .addComponent(**haad\_img**, GroupLayout.***PREFERRED\_SIZE***, 359, GroupLayout.***PREFERRED\_SIZE***)  
 .addGap(233, 233, 233))  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addGap(353, 353, 353)  
 .addComponent(**label3**)  
 .addContainerGap(363, Short.***MAX\_VALUE***))  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addGroup(contentPaneLayout.createParallelGroup()  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addGap(33, 33, 33)  
 .addGroup(contentPaneLayout.createParallelGroup()  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addComponent(**checkBox1**)  
 .addGap(18, 18, 18)  
 .addComponent(**logoutButton**))  
 .addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.***TRAILING***)  
 .addComponent(**label2**)  
 .addGroup(contentPaneLayout.createParallelGroup()  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addGap(3, 3, 3)  
 .addComponent(**welcome\_lb1**)  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***RELATED***)  
 .addComponent(**welcome\_lb2**))  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addComponent(**content\_lb7**)  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***RELATED***)  
 .addComponent(**content\_lb8**)  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***RELATED***)  
 .addComponent(**submitButton**))  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addComponent(**content\_lb2**)  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***RELATED***)  
 .addComponent(**content\_lb4**)  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***RELATED***)  
 .addComponent(**content\_lb5**))  
 .addComponent(**content\_lb1**)  
 .addComponent(**content\_lb3**))))  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***RELATED***, 87, Short.***MAX\_VALUE***))  
 .addGroup(GroupLayout.Alignment.***TRAILING***, contentPaneLayout.createSequentialGroup()  
 .addGap(0, 41, Short.***MAX\_VALUE***)  
 .addComponent(**fileSendStatu**, GroupLayout.***PREFERRED\_SIZE***, 33, GroupLayout.***PREFERRED\_SIZE***)  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***UNRELATED***)  
 .addComponent(**fileSendMessage**, GroupLayout.***PREFERRED\_SIZE***, 307, GroupLayout.***PREFERRED\_SIZE***)  
 .addGap(79, 79, 79)))  
 .addGroup(contentPaneLayout.createParallelGroup()  
 .addComponent(**scrollPane1**, GroupLayout.***PREFERRED\_SIZE***, 329, GroupLayout.***PREFERRED\_SIZE***)  
 .addComponent(**content\_lb9**))  
 .addGap(32, 32, 32))  
 );  
 contentPaneLayout.setVerticalGroup(  
 contentPaneLayout.createParallelGroup()  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addComponent(**haad\_img**)  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***RELATED***, 12, Short.***MAX\_VALUE***)  
 .addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.***TRAILING***, **false**)  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.***BASELINE***)  
 .addComponent(**welcome\_lb1**)  
 .addComponent(**welcome\_lb2**))  
 .addGap(24, 24, 24)  
 .addComponent(**content\_lb1**)  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***UNRELATED***)  
 .addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.***BASELINE***)  
 .addComponent(**content\_lb2**)  
 .addComponent(**content\_lb4**)  
 .addComponent(**content\_lb5**))  
 .addGap(18, 18, 18)  
 .addComponent(**content\_lb3**)  
 .addGap(32, 32, 32)  
 .addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.***TRAILING***)  
 .addComponent(**content\_lb7**)  
 .addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.***LEADING***, **false**)  
 .addComponent(**submitButton**, GroupLayout.***DEFAULT\_SIZE***, GroupLayout.***DEFAULT\_SIZE***, Short.***MAX\_VALUE***)  
 .addComponent(**content\_lb8**, GroupLayout.***PREFERRED\_SIZE***, 38, GroupLayout.***PREFERRED\_SIZE***)))  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***RELATED***)  
 .addComponent(**label2**)  
 .addGap(32, 32, 32)  
 .addGroup(contentPaneLayout.createParallelGroup()  
 .addComponent(**fileSendStatu**, GroupLayout.***DEFAULT\_SIZE***, GroupLayout.***DEFAULT\_SIZE***, Short.***MAX\_VALUE***)  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addComponent(**fileSendMessage**, GroupLayout.***PREFERRED\_SIZE***, 25, GroupLayout.***PREFERRED\_SIZE***)  
 .addGap(0, 0, Short.***MAX\_VALUE***)))  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***RELATED***, 48, Short.***MAX\_VALUE***)  
 .addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.***BASELINE***)  
 .addComponent(**checkBox1**)  
 .addComponent(**logoutButton**, GroupLayout.***PREFERRED\_SIZE***, 48, GroupLayout.***PREFERRED\_SIZE***)))  
 .addGroup(contentPaneLayout.createSequentialGroup()  
 .addComponent(**content\_lb9**)  
 .addPreferredGap(LayoutStyle.ComponentPlacement.***RELATED***)  
 .addComponent(**scrollPane1**, GroupLayout.***PREFERRED\_SIZE***, 344, GroupLayout.***PREFERRED\_SIZE***)))  
 .addGap(48, 48, 48)  
 .addComponent(**label3**)  
 .addContainerGap())  
 );  
 setSize(835, 610);  
 setLocationRelativeTo(getOwner());  
 *// JFormDesigner - End of component initialization //GEN-END:initComponents* }  
  
 **public void** run(){  
 **new** Thread(){  
 **public void** run(){  
 **while**(**true**){  
 **try**{  
 String messageFromSever = **inStream**.readLine();  
 System.***out***.println(**"Messgae from Server : "** + messageFromSever);  
 **if**(messageFromSever.equalsIgnoreCase(**"broadcast"**)){  
 SimpleDateFormat df = **new** SimpleDateFormat(**"yyyy-MM-dd HH:mm:ss"**);  
 String broadcastMessage = **"<"** + df.format(**new** Date()) + **">"**;  
 broadcastMessage += **inStream**.readLine();  
 **textPane1**.setText(**textPane1**.getText() + **"\n"** + broadcastMessage);  
 }  
 } **catch** (IOException e) {  
 JOptionPane.*showMessageDialog*(**null**, **"LOST CONNECT"**, **"Alert"**, JOptionPane.***ERROR\_MESSAGE***);  
 System.*exit*(0);  
 e.printStackTrace();  
 }  
 }  
 }  
 }.start();  
 }  
 *// JFormDesigner - Variables declaration - DO NOT MODIFY //GEN-BEGIN:variables* **private** JLabel **haad\_img**;  
 **private** JLabel **welcome\_lb1**;  
 **private** JLabel **welcome\_lb2**;  
 **private** JButton **submitButton**;  
 **private** JLabel **content\_lb1**;  
 **private** JLabel **content\_lb2**;  
 **private** JLabel **content\_lb3**;  
 **private** JLabel **content\_lb4**;  
 **private** JLabel **content\_lb5**;  
 **private** JLabel **content\_lb8**;  
 **private** JLabel **content\_lb6**;  
 **private** JLabel **content\_lb7**;  
 **private** JLabel **label2**;  
 **private** JButton **logoutButton**;  
 **private** JLabel **label3**;  
 **private** JCheckBox **checkBox1**;  
 **private** JScrollPane **scrollPane1**;  
 **private** JTextPane **textPane1**;  
 **private** JLabel **content\_lb9**;  
 **private** JLabel **fileSendMessage**;  
 **private** JLabel **fileSendStatu**;  
 *// JFormDesigner - End of variables declaration //GEN-END:variables*}

#### 2.4.7 MySqlConnect.java

**package** connection;  
  
**import** javax.swing.\*;  
**import** java.sql.Connection;  
**import** java.sql.DriverManager;  
**import** java.sql.SQLException;  
  
**public class** MySqlConnect {  
 Connection **conn** = **null**;  
 **public static** Connection connectDB(){  
 **try**{  
 Class.*forName*(**"com.mysql.jdbc.Driver"**);  
 Connection conn = DriverManager.*getConnection*(**"jdbc:mysql://cdb-7b4hqwgw.cd.tencentcdb.com:10082/sys"**, **"root"**, **"imagoodboy456"**);  
 System.***out***.println(**"Connected to database"**);  
 **return** conn;  
 } **catch** (ClassNotFoundException | SQLException e) {  
 JOptionPane.*showMessageDialog*(**null**, e);  
 **return null**;  
 }  
 }  
}

## 运行与测试（运行效果截图在此）

### 3.1 服务端功能测试

#### 3.1.1登录界面