JAVA 课程设计报告
聊天室系统：熊猫聊天室

姓　名	詹少雄
班　级	计算机 2003 班
学　号	20193082102O2

2021 年 12 月

JAVA 课程设计报告：聊天室系统

计科 2003 詹少雄 2019308210202

目录

一、 课程设计要求

JAVA 课程设计程序报告，利用 JAVA 网络编程完成一个聊天室系统。基本要求是：1、实现基本聊天通讯功能；2、使用到 JAVA 的多线程编程知识； 3、聊天记录能够保存；4、使用到 JAVA 的 GUI 图形界面编程知识；提高版本的要求：1、利用 MySQL 数据库实现用户的登录与注册；2、能够发送文件；3、界面美化

二、 本程序概述

本程序名称：熊猫聊天室

本程序作者：计科 2003 班 詹少雄

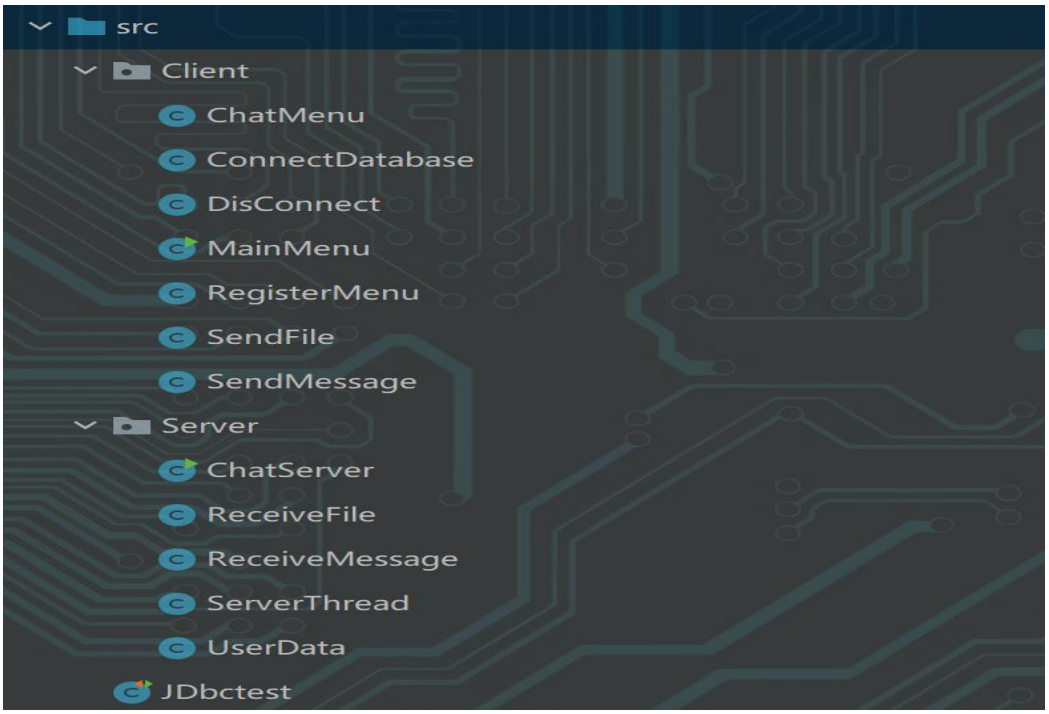
本程序源码编译环境：JAVA version 8_221

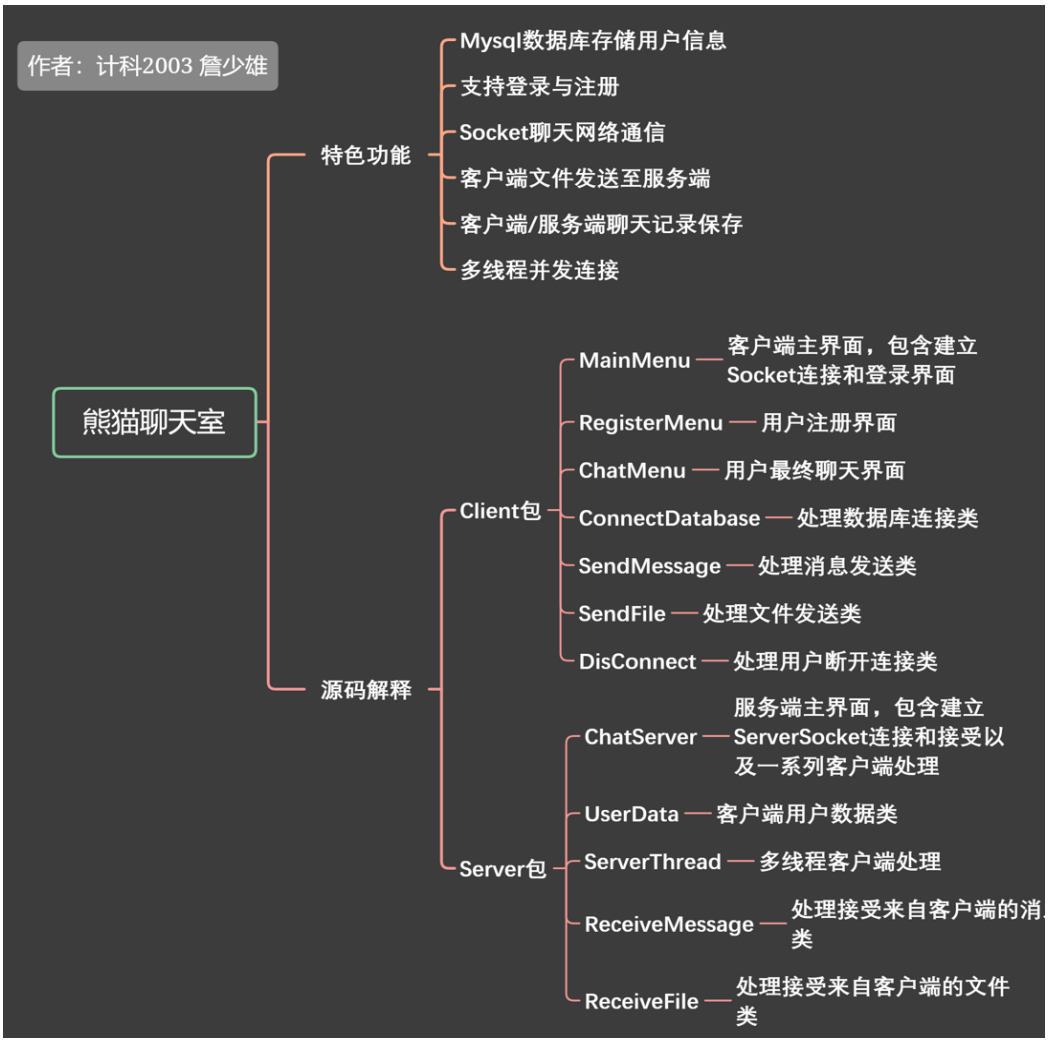
编写本程序所用的编辑器：IntelliJ IDEA Community Edition 版本 2021.2.2

本程序实现了一个多用户聊天室的功能，并且**采用了 MySQL 数据库来存储用户的信息**，且用户可以通过客户端 Client **进行登录/注册等操作**，并且制作了**多个界面**（如登录界面、注册界面、客户端聊天界面、服务端信息界面），服务端支持多线程访问，可以支持**同一时间多个客户端进行并发聊天访问**，并且本程序实现了**文件传输的功能**，用户可以上传文件到熊猫聊天云端存储空间，上传成功后，服务端将会返回该文件的 FTP 下载地址，用户可以通过 FTP 下载到该文件。

本程序所用到的 JAVA 知识：1、GUI 图形界面；2、Socket 网络通信；3、字节/字符/文件输入/输出流；4、JDBC Mysql 数据库连接；5、Thread 多线程架构。

项目**源文件结构**以及**程序概述大纲导图**：





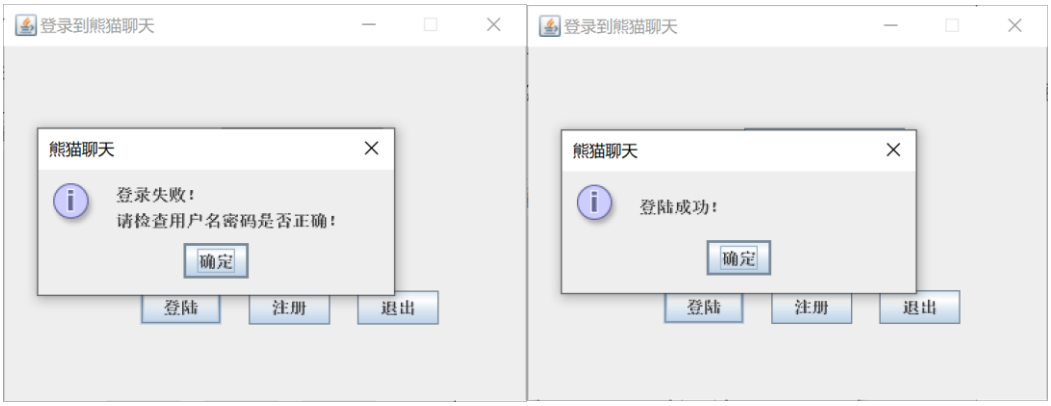
三、 本程序功能介绍与程序演示

1. 登录功能

用户首先打开的主界面便是登录界面

如下图所示：

输入用户名和密码便可进行登录，若无用户名和密码可以点击注册按钮进行新用户注册。



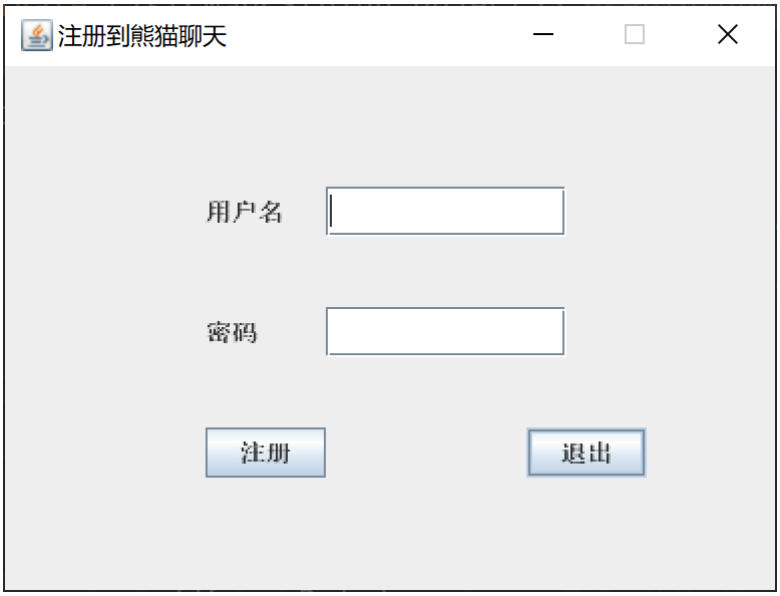
（分别是登录失败与登录成功的提示）

2. 注册功能

若你希望登录到熊猫聊天系统，但无奈没有账户，你可以通过点击主界面的“注册”按钮，进行到注册界面。

本程序采用 MySQL 数据库存放用户的数据，并且支持数据库处理等操作。

熊猫聊天注册界面如下图所示：

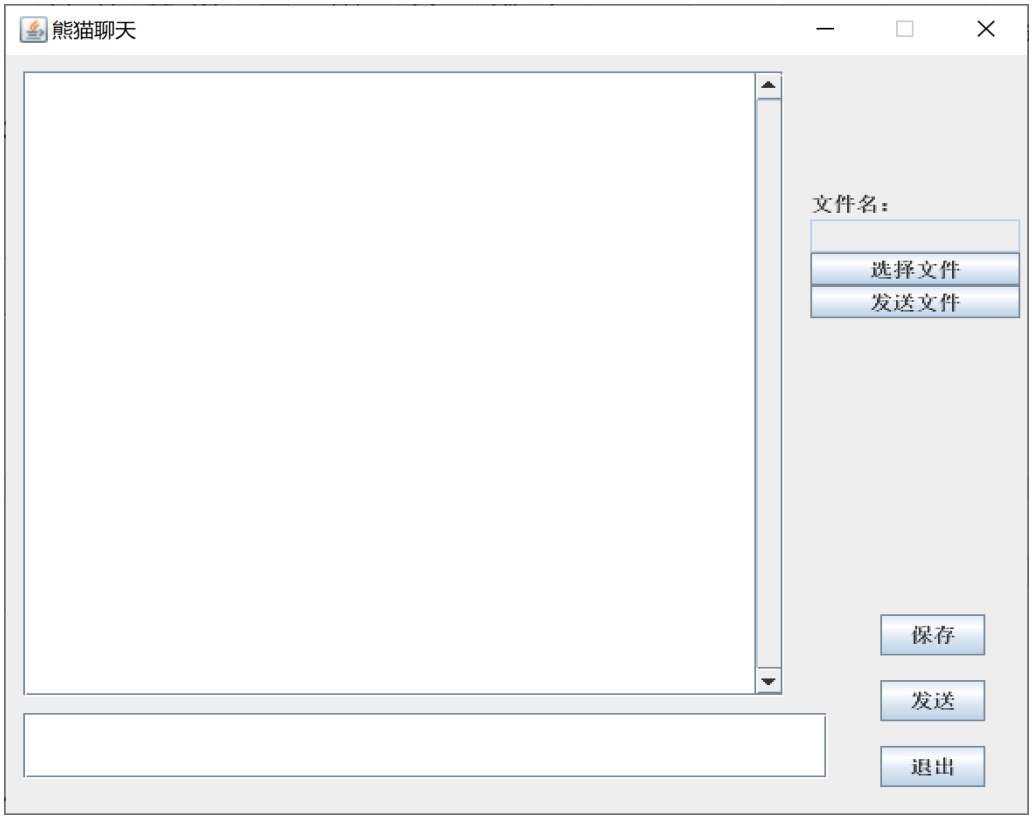


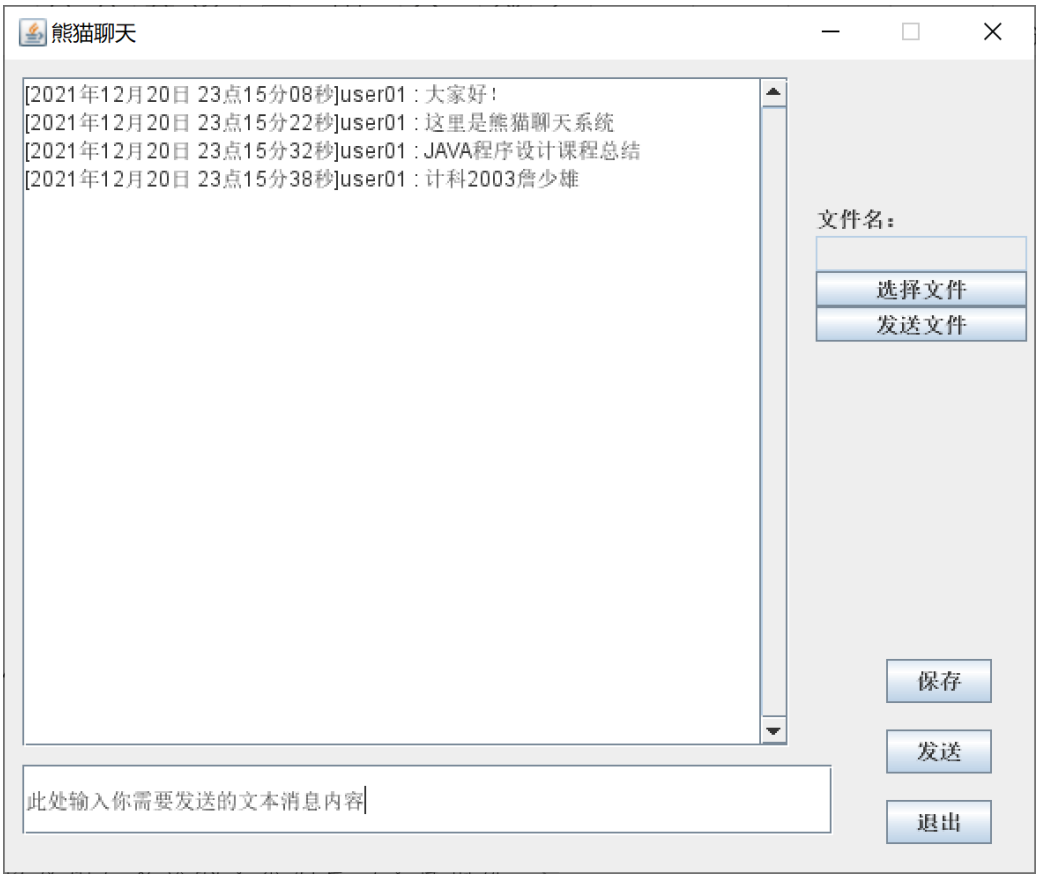
此处细节：用户名要求大于 3 个字符，密码要求大于 6 个字符

3. 客户端聊天功能

登录成功或者注册成功后将会自动跳转到聊天系统主界面，在聊天主界面，可以进行消息文本的发送或文件发送，也可以保存目前客户端的聊天记录到文件。

聊天主界面如下图所示：



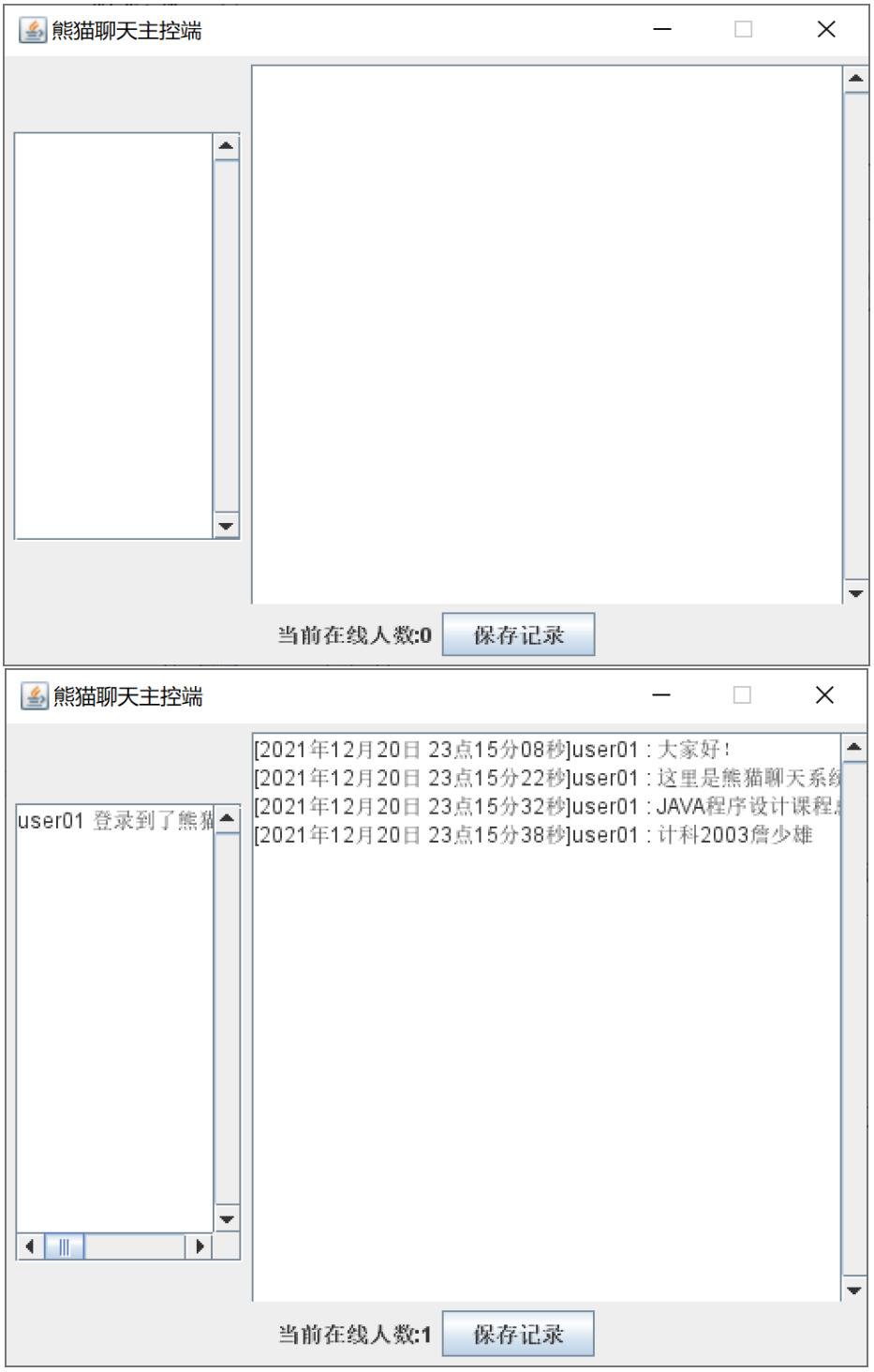


（客户端聊天界面演示）

4. 服务端信息显示

服务端可显示一系列信息，如**目前有多少人在线，用户在何时登录到了系统以及何时离开，用户发送的消息与文件情况。**

服务端信息显示界面如下图所示：

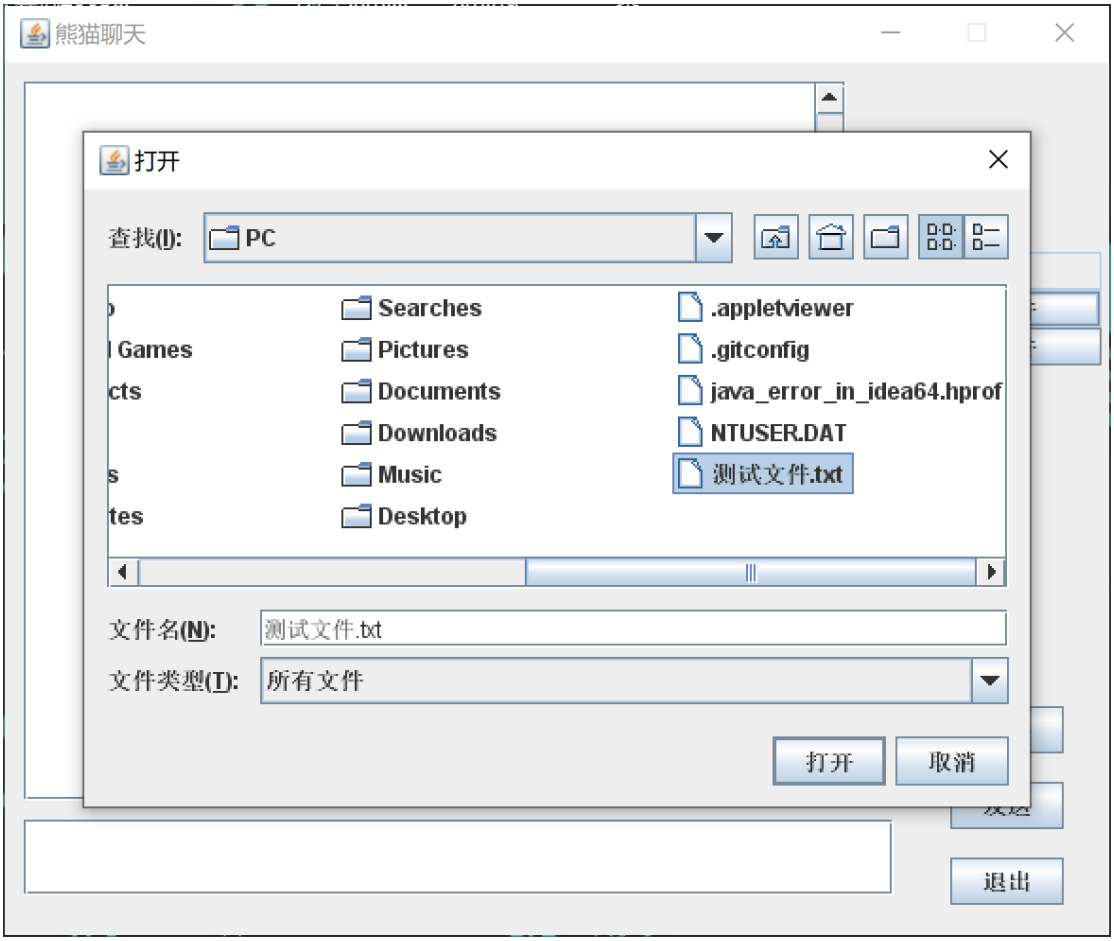


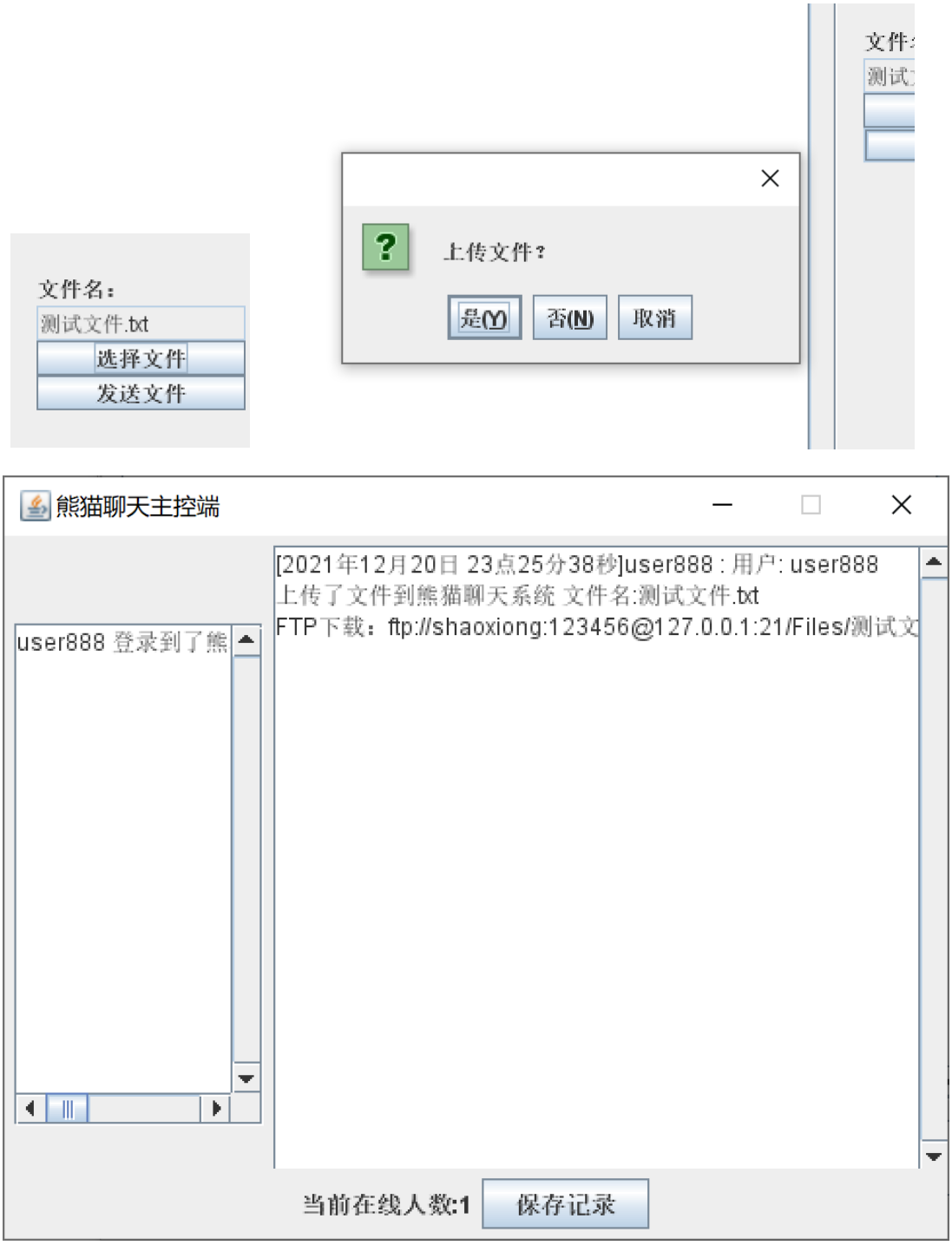
(服务端信息显示功能演示)

左侧区域用于显示用户的登录/登出情况，可以显示用户在 X 年 X 月 X 日 X 时 X 分 X 秒登录/登出到了熊猫聊天系统，右侧区域则是显示从客户端发过来的用户的消息或文件信息。

5. 文件发送

用户可以在聊天界面点击“选择文件”选择所要发送的文件然后进行发送。可以发送至服务端并保存在熊猫聊天系统内，用户可以通过返回的 FTP 地址进行下载文件。





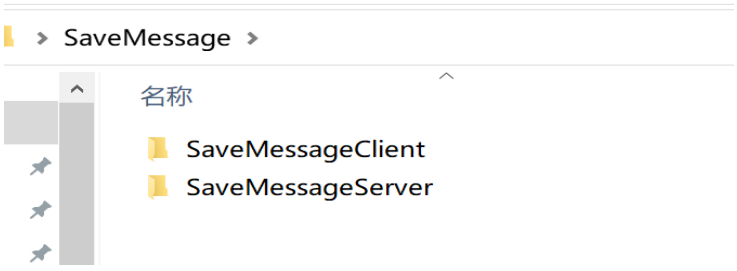
（客户端上传文件到服务端效果演示）

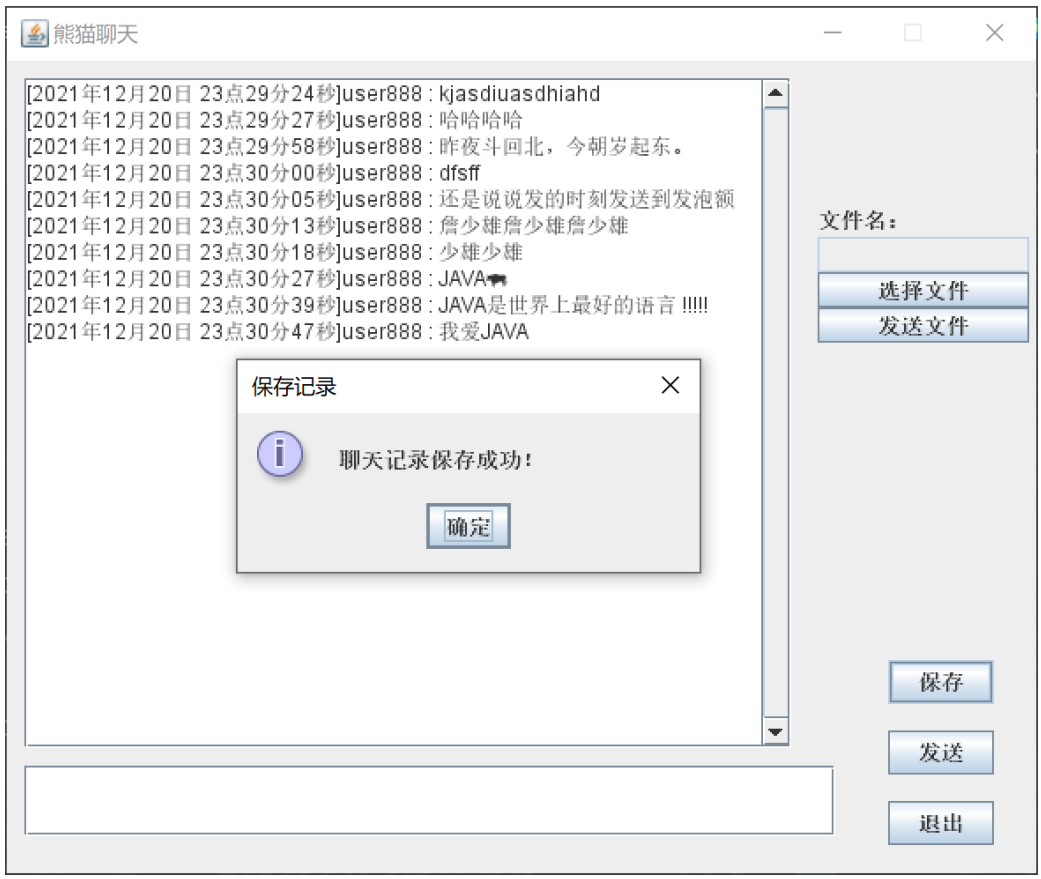
6. 聊天记录保存

可以将界面所产生的聊天文本保存到文本文件中，并且服务端和客户端都能分别进行保存。

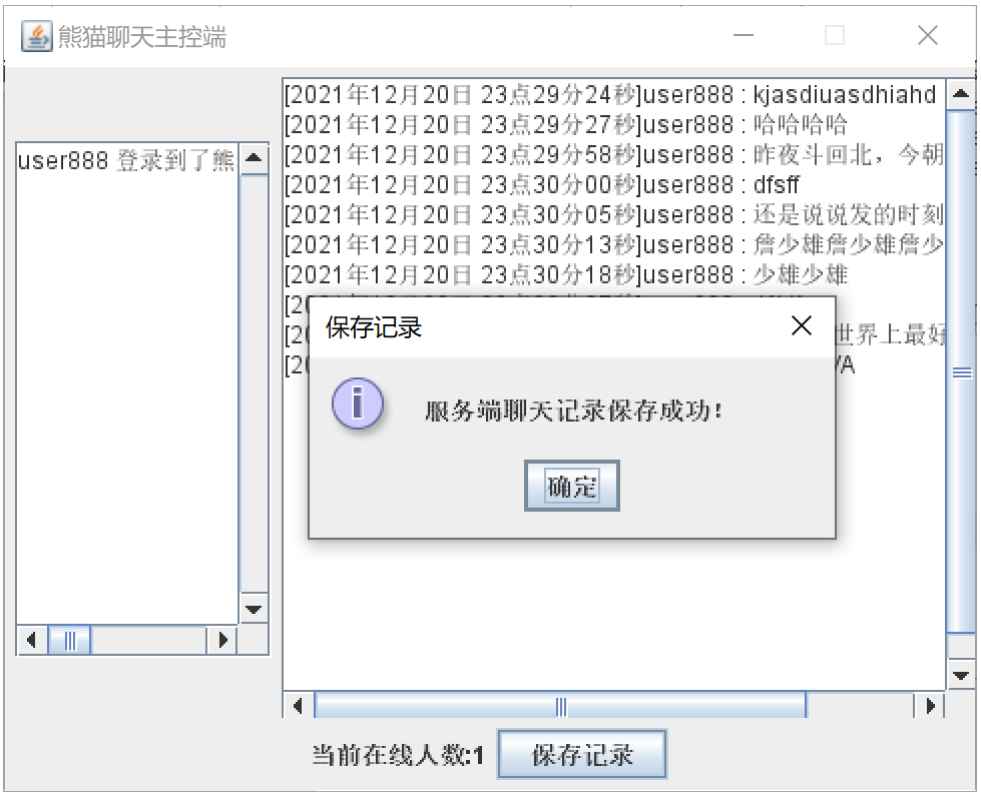
聊天记录保存效果如下图所示：

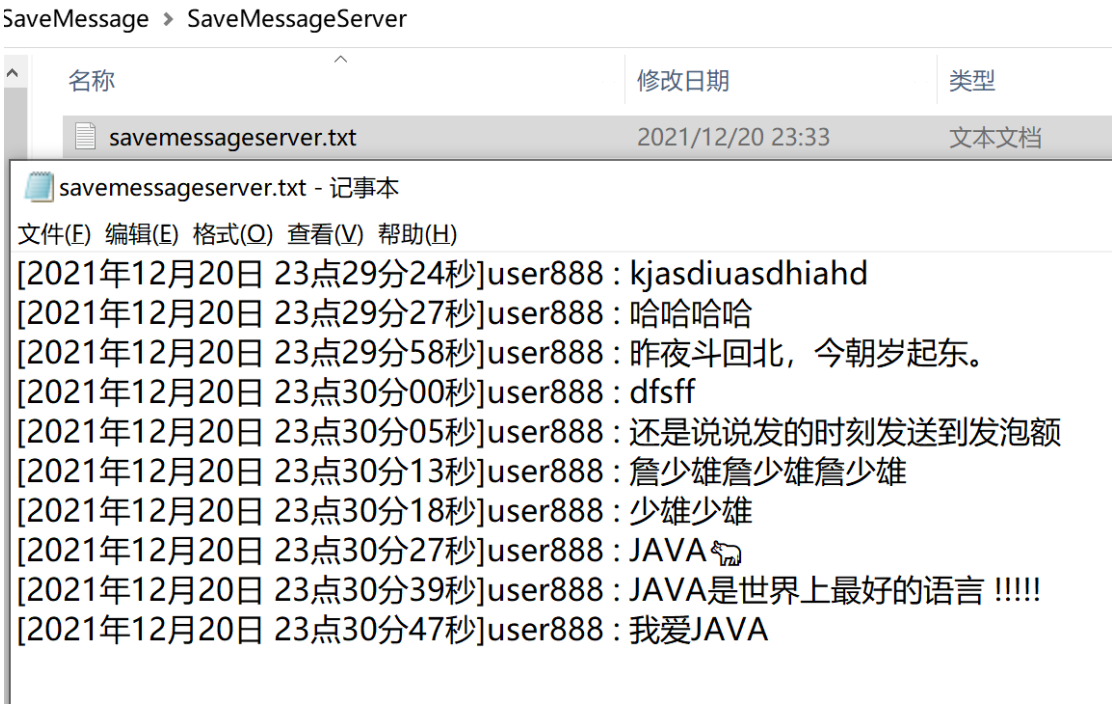
聊天记录保存文件路径：





（客户端聊天记录保存演示）

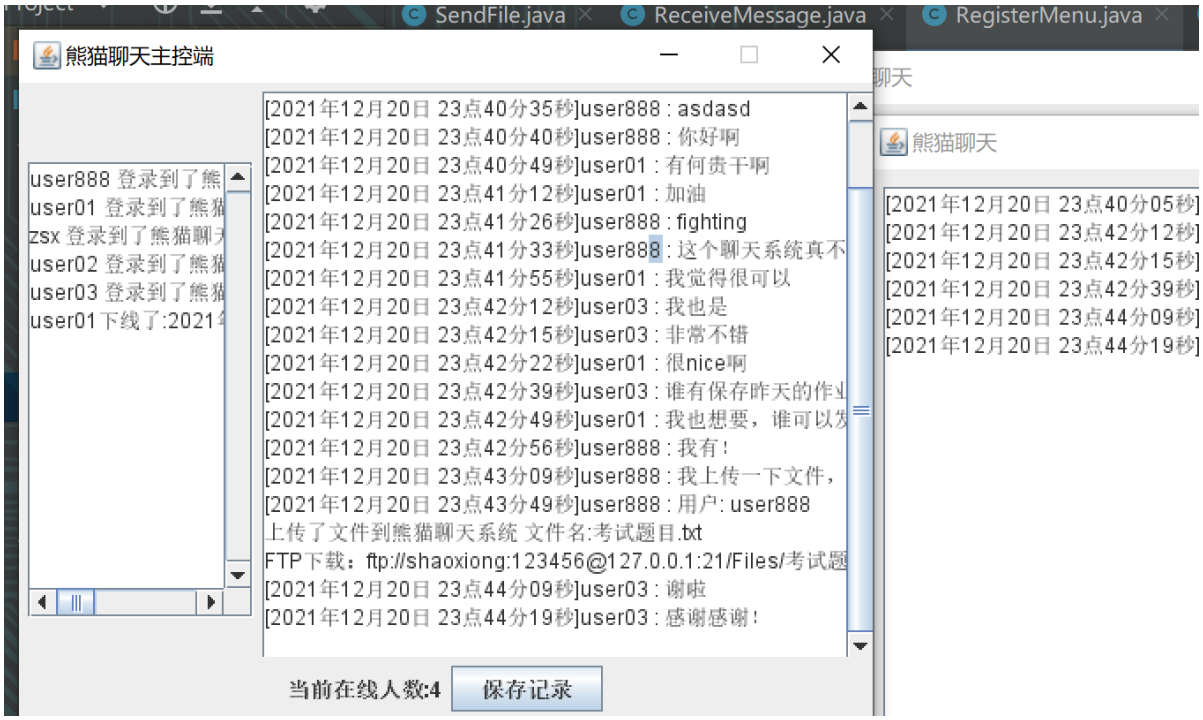


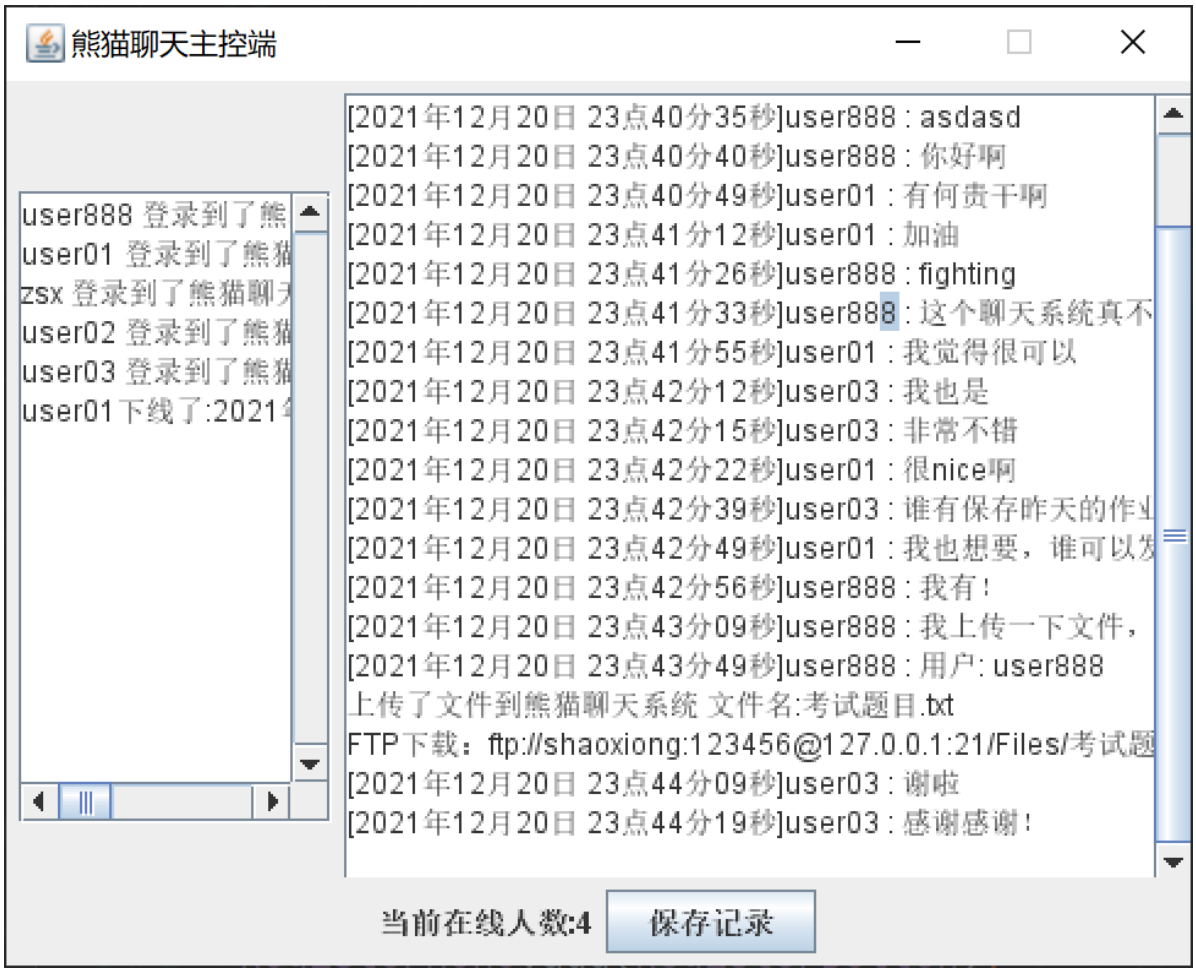


(服务端聊天记录保存演示)

7. 程序实际应用演示

综合模拟一个多人聊天室，进行实际应用的展示。

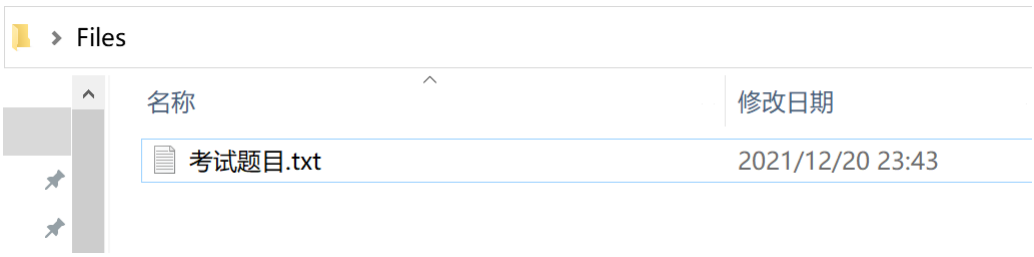

SendFile.java
ReceiveMessage.java
RegisterMenu.java
熊猫聊天主控端
聊天
熊猫聊天
user888 登录到了熊
user01 登录到了熊猫
zsx 登录到了熊猫聊天
user02 登录到了熊猫
user03 登录到了熊猫
user01下线了:2021
[2021年12月20日 23点40分35秒]user888 : asdasd
[2021年12月20日 23点40分40秒]user888 : 你好啊
[2021年12月20日 23点40分49秒]user01 : 有何贵干啊
[2021年12月20日 23点41分12秒]user01 : 加油
[2021年12月20日 23点41分26秒]user888 : fighting
[2021年12月20日 23点41分33秒]user888 : 这个聊天系统真不
[2021年12月20日 23点41分55秒]user01 : 我觉得很可以
[2021年12月20日 23点42分12秒]user03 : 我也是
[2021年12月20日 23点42分15秒]user03 : 非常不错
[2021年12月20日 23点42分22秒]user01 : 很nice啊
[2021年12月20日 23点42分39秒]user03 : 谁有保存昨天的作业
[2021年12月20日 23点42分49秒]user01 : 我也想要，谁可以发
[2021年12月20日 23点42分56秒]user888 : 我有！
[2021年12月20日 23点43分09秒]user888 : 我上传一下文件，
[2021年12月20日 23点43分49秒]user888 : 用户: user888
上传了文件到熊猫聊天系统 文件名:考试题目.txt
FTP下载：ftp://shaoxiong:123456@127.0.0.1:21/Files/考试题
[2021年12月20日 23点44分09秒]user03 : 谢啦
[2021年12月20日 23点44分19秒]user03 : 感谢感谢！
[2021年12月20日 23点42分05秒]
[2021年12月20日 23点42分12秒]
[2021年12月20日 23点42分15秒]
[2021年12月20日 23点42分39秒]
[2021年12月20日 23点44分09秒]
[2021年12月20日 23点44分19秒]
当前在线人数:4
保存记录

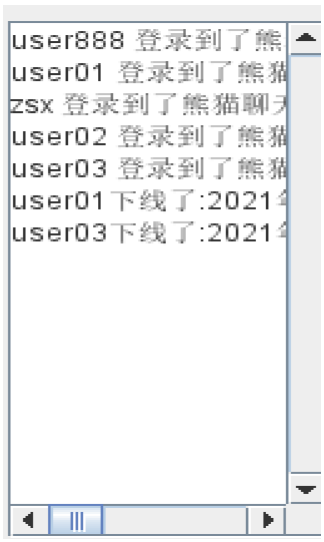

熊猫聊天主控端
user888 登录到了熊
user01 登录到了熊猫
zsx 登录到了熊猫聊天
user02 登录到了熊猫
user03 登录到了熊猫
user01下线了:2021
[2021年12月20日 23点40分35秒]user888 : asdasd
[2021年12月20日 23点40分40秒]user888 : 你好啊
[2021年12月20日 23点40分49秒]user01 : 有何贵干啊
[2021年12月20日 23点41分12秒]user01 : 加油
[2021年12月20日 23点41分26秒]user888 : fighting
[2021年12月20日 23点41分33秒]user888 : 这个聊天系统真不
[2021年12月20日 23点41分55秒]user01 : 我觉得很可以
[2021年12月20日 23点42分12秒]user03 : 我也是
[2021年12月20日 23点42分15秒]user03 : 非常不错
[2021年12月20日 23点42分22秒]user01 : 很nice啊
[2021年12月20日 23点42分39秒]user03 : 谁有保存昨天的作业
[2021年12月20日 23点42分49秒]user01 : 我也想要，谁可以发
[2021年12月20日 23点42分56秒]user888 : 我有！
[2021年12月20日 23点43分09秒]user888 : 我上传一下文件，
[2021年12月20日 23点43分49秒]user888 : 用户: user888
上传了文件到熊猫聊天系统 文件名:考试题目.txt
FTP下载：ftp://shaoxiong:123456@127.0.0.1:21/Files/考试题
[2021年12月20日 23点44分09秒]user03 : 谢啦
[2021年12月20日 23点44分19秒]user03 : 感谢感谢！
当前在线人数:4
保存记录

savemessageserver.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

聊天记录:
[2021年12月20日 23点40分05秒]user03 : 大家好
[2021年12月20日 23点40分22秒]user01 : sss
[2021年12月20日 23点40分25秒]user02 : @text
[2021年12月20日 23点40分35秒]user888 : asdasd
[2021年12月20日 23点40分40秒]user888 : 你好啊
[2021年12月20日 23点40分49秒]user01 : 有何贵干啊
[2021年12月20日 23点41分12秒]user01 : 加油
[2021年12月20日 23点41分26秒]user888 : fighting
[2021年12月20日 23点41分33秒]user888 : 这个聊天系统真不错欸
[2021年12月20日 23点41分55秒]user01 : 我觉得很可以
[2021年12月20日 23点42分12秒]user03 : 我也是
[2021年12月20日 23点42分15秒]user03 : 非常不错
[2021年12月20日 23点42分22秒]user01 : 很nice啊
[2021年12月20日 23点42分39秒]user03 : 谁有保存昨天的作业啊，可以发给我吗?
[2021年12月20日 23点42分49秒]user01 : 我也想要，谁可以发我一份？？？
[2021年12月20日 23点42分56秒]user888 : 我有!
[2021年12月20日 23点43分09秒]user888 : 我上传一下文件，大家可以下载!
[2021年12月20日 23点43分49秒]user888 : 用户: user888
上传了文件到熊猫聊天系统 文件名:考试题目.txt
FTP下载：ftp://shaoxiong:123456@127.0.0.1:21/Files/考试题目.txt
[2021年12月20日 23点43分58秒]user01 : 谢谢!
[2021年12月20日 23点44分09秒]user03 : 谢啦
[2021年12月20日 23点44分19秒]user03 : 感谢感谢!





四、 功能实现核心代码展示

1. JDBC 数据库操作驱动连接 MySQL 数据库

利用 JAVA_JDBC 数据库方面的知识，我实现了与 MySQL 数据库的连接。

数据库连接测试代码：JDBCTest.class

```
import java.sql.DriverManager;
import java.sql.SQLException;
import org.junit.Test;
import java.sql.Connection;
public class JDbctest {
    @Test
    public void jdbcall() throws ClassNotFoundException,
SQLException{

        Class.forName("com.mysql.cj.jdbc.Driver");//加载驱动类
        String url="jdbc:mysql://localhost:3306/test";
        String username="shaoxiong";
        String password="Shaoxiong0405+";
        Connection
conn=DriverManager.getConnection(url,username,password);//用参
数得到连接对象
        System.out.println("连接成功！");
        System.out.println(conn);
    }
}
```

（以上代码用于测试数据库连接）

数据库连接操控实现：ConnectDatabase.class

代码过长，仅展示核心代码：

```java
public class ConnectDatabase {
    // MySQL 8.0 以下版本 - JDBC 驱动名及数据库 URL
    // DESPEARATE! static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/shaoxiong";
    // MySQL 8.0 以上版本 - JDBC 驱动名及数据库 URL
    //static final String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";

    static final String USER = "shaoxiong";
    static final String PASS = "Shaoxiong0405+";

    public boolean ConnectMySQl(String id, String pwd, int tag) {
        //tag标记，如果tag = 0 表示是注册检查
        //tag = 1表示是登录检查
        Connection conn = null;
        Statement stmt = null;
```

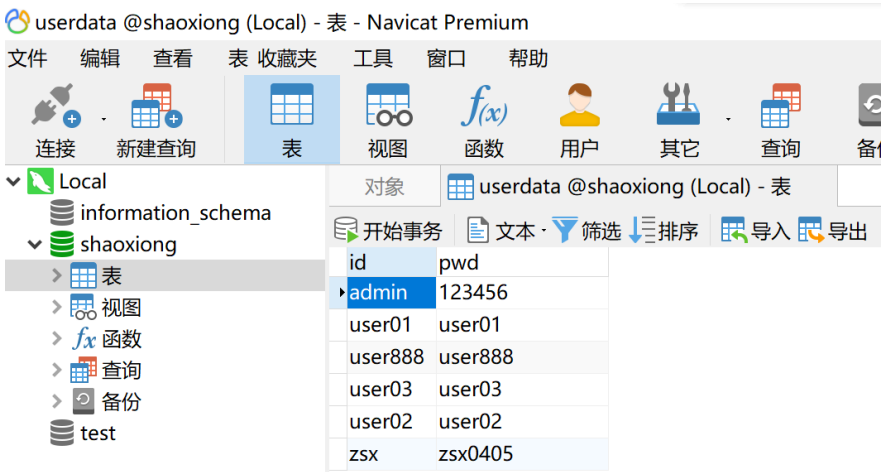
ConnectDatabase.class 用于处理客户端的数据库连接，主要是用于登录和注册，首先注册 JDBC 驱动`"com.mysql.cj.jdbc.Driver"`，然后连接到 JDBCMYSQL 数据库连接地址：

`"jdbc:mysql://localhost:3306/shaoxiong"`

ConnectMySQL(String id, String pwd, int tag)用于判断和处理来自客户端的数据库操作：

Tag = 0 表示注册操作；　Tag = 1 表示登录操作；

不同操作分别执行不同的数据库语句，最终返回一个布尔值表示是否登录/注册成功。



```java
if (tag == 0) {
    try {
        // 注册 JDBC 驱动
        Class.forName(JDBC_DRIVER);
        // 打开链接
        System.out.println("正在连接数据库...");
        conn = DriverManager.getConnection(DB_URL, USER, PASS);

        System.out.println(" 正在实例化Statement对象...");
        PreparedStatement ps = null;
        String sql;
        sql = "INSERT INTO userdata VALUES (?,?)";
        ps = conn.prepareStatement(sql);
        ps.setString( parameterIndex: 1,id);
        ps.setString( parameterIndex: 2,pwd);
        ps.execute();
        ps.close();
        conn.close();
        return true;
```

```java
if (tag == 1) {
    try {
        // 注册 JDBC 驱动
        Class.forName(JDBC_DRIVER);
        // 打开链接
        System.out.println("正在连接数据库...");
        conn = DriverManager.getConnection(DB_URL, USER, PASS);

        System.out.println(" 正在实例化Statement对象...");
        stmt = conn.createStatement();
        String sql;
        sql = "SELECT id, pwd FROM userdata";
        ResultSet rs = stmt.executeQuery(sql);

        // 展开结果集数据库
        while (rs.next()) {
            // 通过字段检索
            String cid = rs.getString( columnLabel: "id");
            String cpwd = rs.getString( columnLabel: "pwd");
```

(如上图所示分别是注册和登录相关的核心代码)

2. Client 方面

客户端方面包含如下几个类：MainMenu，RegisterMenu，ConnectDatabase,ChatMenu,SendFile,SendMessage,DisConnect

其中 MainMenu 是主类，包含建立 Socket 连接和登录界面

```
public static void main(String[] args) throws
IOException{
    Socket socket = new Socket("127.0.0.1",8666);
    MainMenu frame = new MainMenu(socket);
    frame.setVisible(true);
}
```

监听本地 IP:127.0.0.1，监听端口为 8666

界面 GUI 方面采用继承 JFrame 类采用 Swing 库以及 BordenLayout 和 GridLayout 进行界面布置

```
public class MainMenu extends JFrame{
    private JPanel LoginMenu;
    private  JButton LoginButton, RegisterButton, ExitButton;
    private  JLabel LoginText, PwdText;
    private JTextField NameBox;
    private JPasswordField PwdBox;

    public MainMenu(Socket s)
    {
        super( title: "登录到熊猫聊天");
        LoginMenu = new JPanel();
        LoginMenu.setLayout(null);
        LoginButton = new JButton( text: "登陆");
        RegisterButton = new JButton( text: "注册");
        ExitButton = new JButton( text: "退出");

        LoginButton.setBounds( x: 100,  y: 180,  width: 60,  height: 25);
        ExitButton.setBounds( x: 260,  y: 180,  width: 60,  height: 25);
        RegisterButton.setBounds( x: 180,  y: 180,  width: 60, height: 25);
        LoginMenu.add(LoginButton);
        LoginMenu.add(RegisterButton);
        LoginMenu.add(ExitButton);
```

SendMessage,SendFile 和 DisConnect 类分别表示客户端发送消息，发送文件和客户端断开连接的处理类，三个类均采用了多线程的处理方式。

通过使用输入输出流相关的知识，通过在 Socket 传输流的前面写入@text,@file 和@bye 告诉服务端客户端具体哪种操作。

```
@Override
public void run()
{
    super.run();
    OutputStream outputStream = null;
    try
    {
        outputStream = s.getOutputStream();
        BufferedWriter bufferedWriter = new BufferedWriter(new OutputStreamWriter(outputStream));
        bufferedWriter.write( str: "@text\n");
        bufferedWriter.flush();
        bufferedWriter.write(message);
        bufferedWriter.flush();
        System.out.println(message);
        System.out.println("发送消息！");
    }
    catch (IOException e)
    {
        e.printStackTrace();
```

```
@Override
public void run()
{
    try {
        PrintStream printStream = new PrintStream(socket.getOutputStream());
        FileInputStream fileInputStream = new FileInputStream(file);
        printStream.println("@file");
        printStream.println(file.getName());
        byte []bytes = new byte[10240];
        int len = 0;
        while ((len = fileInputStream.read(bytes)) != -1)
        {
            printStream.write(bytes);
        }
        fileInputStream.close();
        printStream.close();
```

```
public void run()
{
    super.run();
    OutputStream outputStream = null;
    try
    {
        outputStream = s.getOutputStream();
        BufferedWriter bufferedWriter = new BufferedWriter(new OutputStreamWriter(outputStream));
        bufferedWriter.write( str: "@bye\n");
        bufferedWriter.flush();
        Thread.sleep( millis: 100);
        System.out.println("用户断开了连接");
```

（如上图，分别是发送消息/发送文件/用户断开代码）

3. Server 方面

服务端主要包含以下几个类：

ChatServer,ServerThread,ReceiveFile,ReceiveMessage,UserData

其中 ChatServer.class 是主类，包含了服务端的界面 UI 和服务端 Socket 通信的建立。

```
public static void main(String[] args) throws IOException
{
    ChatServer chatserver = new ChatServer();
    ServerThread serverThread = new ServerThread();
    ServerSocket serverSocket = new ServerSocket(8666);
    //死循环等待多线程等待客户端连接
    while (true)
    {
        Socket socket = serverSocket.accept();
        System.out.println(socket.getLocalAddress());
        InputStream inputStream = socket.getInputStream();
        BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(inputStream));
        String message = bufferedReader.read();
        String []string = message.split("-");
        //JAVA 中 String 的处理 trim()方法是删除字符串前后的空格
        String username = string[0].replaceFirst("@text","
").trim();
        String password = string[1].trim();
        serverThread.addUserTextArea(ChatServer.getUserTextArea());
serverThread.addCommunicationTextArea(ChatServer.getSendTextArea(
));
serverThread.addCurrentNumberLabel(ChatServer.getLabelCurrentNumb
er());
        serverThread.addClient(username,password,socket);
        new Thread(serverThread).start();
    }
}
```

new ServerSocket(8666) 监听本地 8666 端口号

while(true)不断循环，接受客户端的 Socket 请求，

serverSocket.accept()

ServerThread 多线程处理来自客户端的请求，ReceiveMessage 和 ReceiveFile 分别处理接受来自客户端的消息和文件的功能。

UserData 通过数据库连接来短暂存储用户信息到内存，用于图形界面或聊天中用户信息的显示。

```
public class ChatServer extends Thread{
    private JFrame ServerFrame;
    private Panel wPanel,sPanel;
    private static JTextArea userTextArea,sendTextArea;
    private JScrollPane userScrollPane,textScroll;
    private static JLabel LabelCurrentNumber;
    private static JLabel LabelOnlineClient;
    private JButton saveMessageButton;
    public ChatServer()
    {
        ServerFrame = new JFrame( title: "熊猫聊天主控端");
        ServerFrame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        ServerFrame.setResizable(false);
        ServerFrame.setSize( width: 500, height: 400);
        ServerFrame.setLocation( x: 250, y: 250);
        ServerFrame.setLayout(new BorderLayout());

        wPanel = new Panel();
        wPanel.setLayout(new FlowLayout());
```

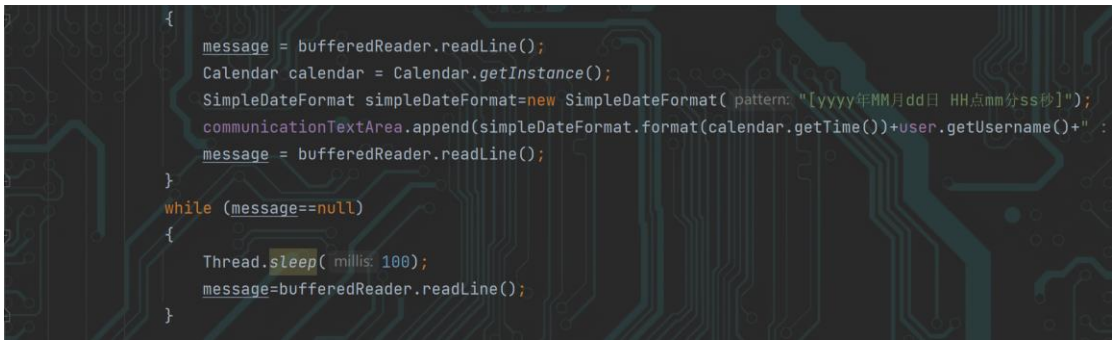
（服务端界面 UI）

ServerThread:多线程处理客户端请求，客户端登录/客户端发送消息文件，客户端退出等等请求。使用多线程并发处理。

```
public class ServerThread extends Thread {
    private List<UserData> users = new ArrayList<>();
    private int count = 0;
    private List<Socket> clientList = new ArrayList<>();
    private JTextArea userTextArea;
    private JTextArea communicationTextArea;
    private JLabel currentNumberLabel;
    private int aliveCount = 0;
    private List<Map<String,Object>> userInformation = new ArrayList<>();

    public void addClient(String username, String password, Socket socket)
    {
        Calendar calendar = Calendar.getInstance();
        SimpleDateFormat simpleDateFormat=new SimpleDateFormat( pattern: "[yyyy年MM月dd日 HH点mm分ss秒]");
        count = count + 1;
        aliveCount = aliveCount + 1;
        UserData userData = new UserData();
        userData.setUsername(username);
        userData.setPassword(password);
        users.add(userData);
        clientList.add(socket);
```

ReceiveMessage 接受消息：接受来自客户端发送的消息文本



```
{
    message = bufferedReader.readLine();
    Calendar calendar = Calendar.getInstance();
    SimpleDateFormat simpleDateFormat=new SimpleDateFormat( pattern: "[yyyy年MM月dd日 HH点mm分ss秒]");
    communicationTextArea.append(simpleDateFormat.format(calendar.getTime())+user.getUsername()+"
    message = bufferedReader.readLine();
}
while (message==null)
{
    Thread.sleep( millis: 100);
    message=bufferedReader.readLine();
}
```

ReceiveFile 接受文件：接受来自客户端发送的文件

```
try
{
    String fileName = dataInputStream.readUTF();
    System.out.println(fileName);
    File file=new File( pathname: "C:\\Users\\PC\\Desktop\\Files\\File"+fileName);
    System.out.println(file.getAbsoluteFile());
    if (!file.exists())
    {
        file.createNewFile();
    }
    fileOutputStream = new FileOutputStream(file);
    byte []bytes = new byte[10240];
    int length = 0;
    while ((length=dataInputStream.read(bytes, off: 0,bytes.length))!=-1)
    {
        fileOutputStream.write(bytes, off: 0,length);
        fileOutputStream.flush();
    }
    fileOutputStream.close();
```

4. 消息记录保存

通过记录一个 Swing 框架结构中的一个 JTextArea 部件，通过文件输出流将一个 JTextArea 部件中的内容逐行写入到一个文本文件当中。并将该操作绑定到“保存”按钮的鼠标点击监听当中。

```
SaveButton.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        super.mouseClicked(e);
        BufferedWriter bw = null;
        try{
```

```
            OutputStream os = new
FileOutputStream("C:\\Users\\PC\\Desktop\\SaveMessage\\SaveMessag
eClient\\savemessageserver.txt");
            bw = new BufferedWriter(new OutputStreamWriter(os));
            for (String value : textArea.getText().split("\n")) {
                bw.write(value);
                bw.newLine();//换行
            }
            JOptionPane.showMessageDialog(null, "聊天记录保存成功！",
"保存记录", JOptionPane.INFORMATION_MESSAGE);
        }catch (IOException e1) {
            e1.printStackTrace();
        } finally {
            if (bw != null) {
                try {
                    bw.close();
                } catch (IOException e1) {
                    e1.printStackTrace();
                }
            }
        }
    }
});
```

五、 总结

从学习、收集资料到写代码和调试探索花费了相当多的时间，但也在这过程中学到很多知识，如网络通信中的输入输出流，如多线程处理操作和 JAVA 程序与数据库的交互连接使用等等。过程中学到很多，同时也深刻了解到了 JAVA 面向对象程序设计的便捷性和优越性，通过这个网络聊天室程序的设计和开发，我断断续续经过了很长一段时间的设计思考，思索如何实现我希望实现的程序，同时在 DEBUG 调试过程中遇到了很多奇奇怪怪的问题，搜集资料和查阅 JAVA 文档逐一解决。解决 BUG 的过程也让我很愉悦，也很享受攻克难关的过程。通过这一次实验设计，也算是提前进行一次工程项目的开发与探索，这对我以后的编程开发与学习有很大的作用！

六、 参考文献

[1]菜鸟教程—JAVA 图形界面与网络编程教程

https://www.runoob.com/java/java-tutorial.html

[2] JAVA 怎么把从 TextArea 获得的字符串以文件的形式输出;

https://blog.csdn.net/tpian928/article/details/28388965

[3] Java 实现简易聊天室;

https://www.cnblogs.com/hzauxx/p/11145756.html

[4] JAVA-Socket 通信 打造属于自己的聊天室

https://www.cnblogs.com/csu-lmw/p/9709782.html