
华南农业大学

课程论文(设计)

课程名称 程序设计与算法综合实习III

论文题目 局域网聊天程序

指导老师 彭红星

所在学院 数学与信息学院软件学院

专业年级 14计算机科学与技术<1>班

提交日期 2016. 5. 10

小组成员表

姓名	学号	承担任务
张涛	201430320127	框架设计分析, 网络通信协议设计分析, 主要功能实现
祝春雨	201430320132	功能分析, 测试, 部分功能实现
郑紫月	201430320129	需求分析, 测试, 界面美化设计

华南农业大学数学与信息学院

《程序设计与算法综合实习Ⅲ》评分表

题 目	局域网聊天程序			
专业班级	14 计算机科学与技术<1>班	提交日期	2016. 5. 9	
小组成员	学 号	姓 名	工作量%	成绩 (指导老师填写)
组 长	201430320127	张 涛	60	
组 员	201430320132	祝春雨	20	
组 员	201430320129	郑紫月	20	
小组评分表				
评分项目				分数
1. 实现情况（15）：完成课程设计要求的情况				
2. 系统分析（10）：对系统的理解及分析的深入程度。				
3. 系统设计（20）：总体设计、具体设计、数据结构设计等。				
4. 程序设计（30）：面向对象程序设计方法与编程工具掌握水平。				
5. 界面设计（15）：界面美观程度、操作方式友好程度等。				
6. 论文撰写（10）：论文撰写的规范性、合理性等等。				
小组总成绩：				
教师签名		日 期		

《程序设计与算法综合实习Ⅲ》正文

一、系统分析

1.1 问题描述

客户端：

实现简易版的局域网聊天器，实现富文本内容聊天，智能聊天机器人，群发消息，传送文件等功能

服务器端：

实现群发通知，管理聊天线程

1.2 系统功能分析

客户端功能：

登陆注册
发送表情消息
发送文本消息
截取图片
图片处理
震动效果
发送文件
群发消息
设置聊天文本样式

服务器端：

广播通知

1.3 开发平台及工具介绍

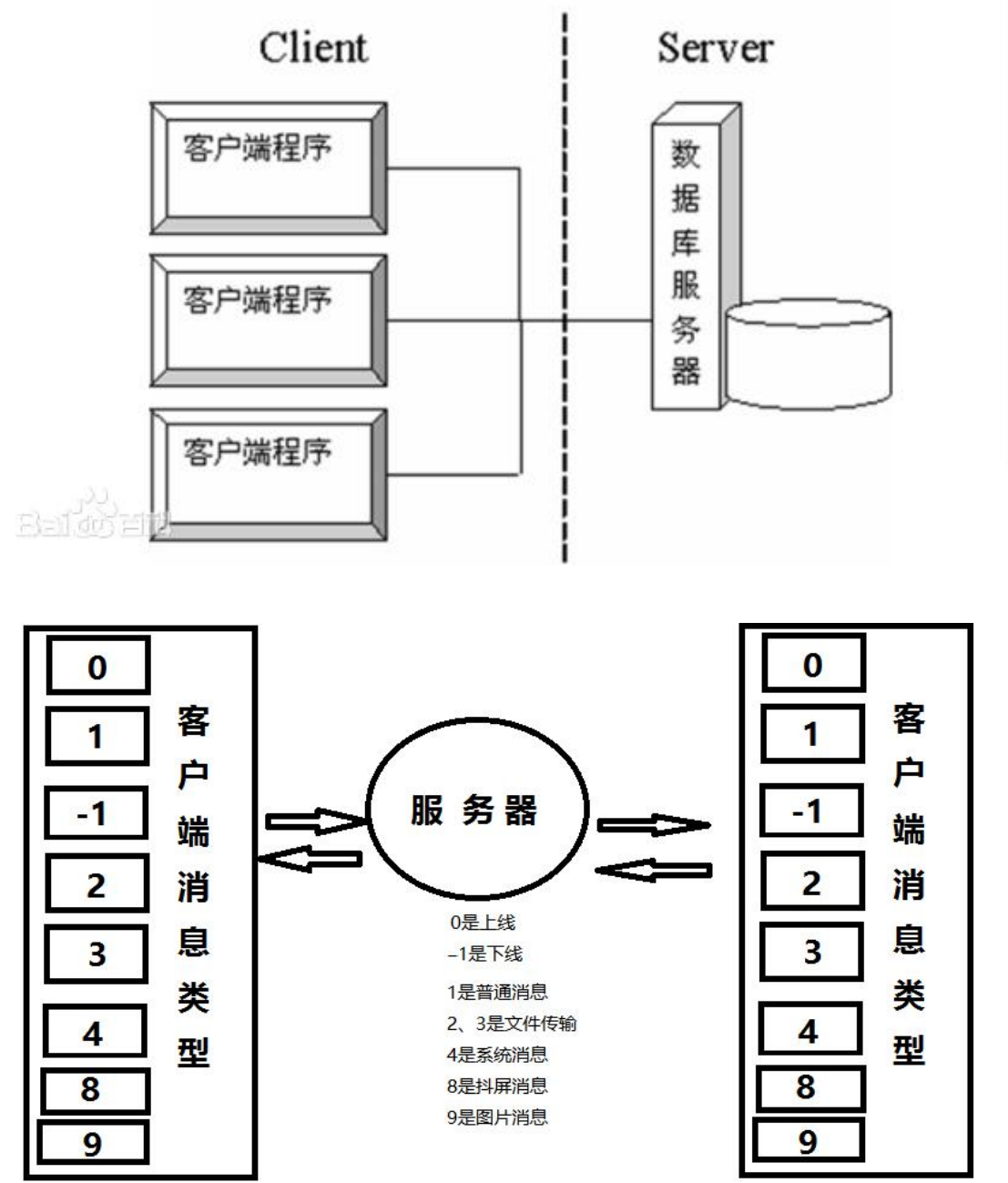
Eclipse Mars2

二、系统设计

1.1 系统总体结构设计

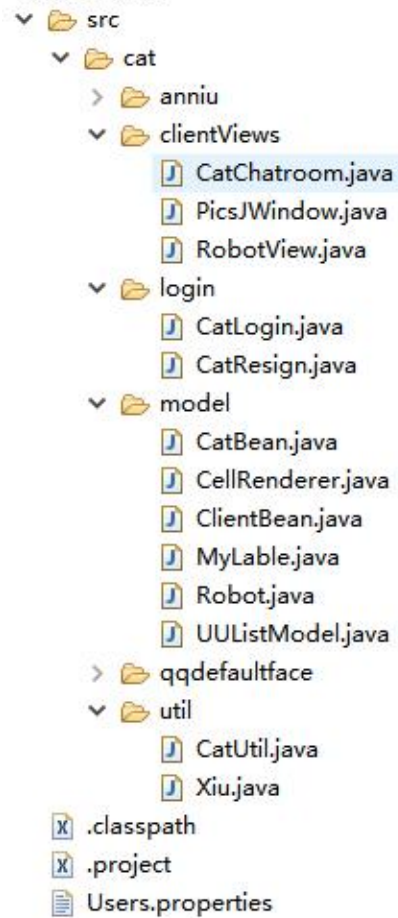
系统采用自己设计的网络消息传输协议，系统采用 CS 架构模式实现数据传送。

C/S系统架构

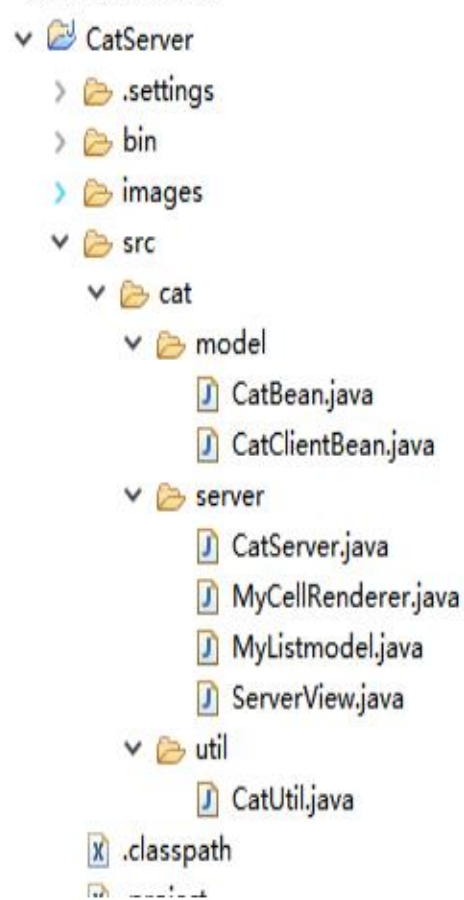


1.2 系统各个类及类之间关系设计

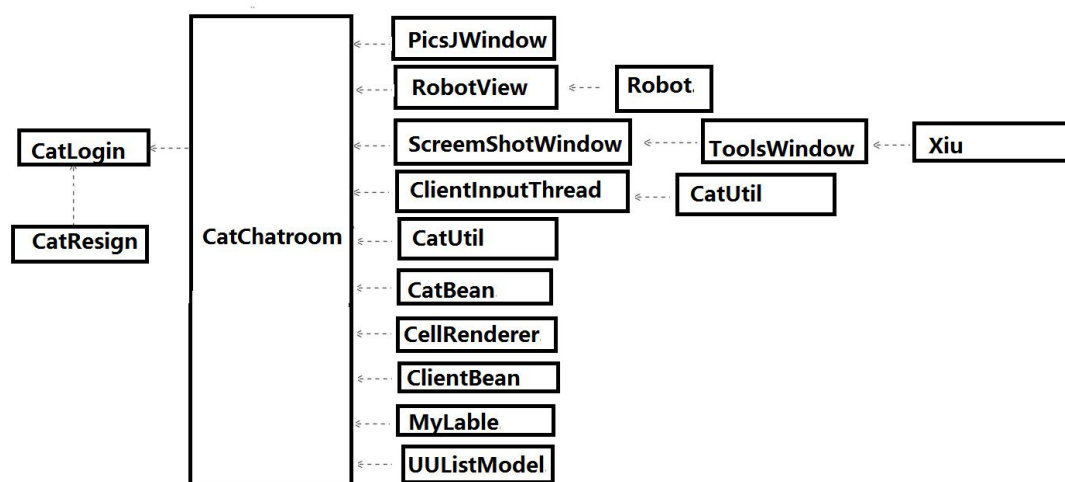
客户端包结构:



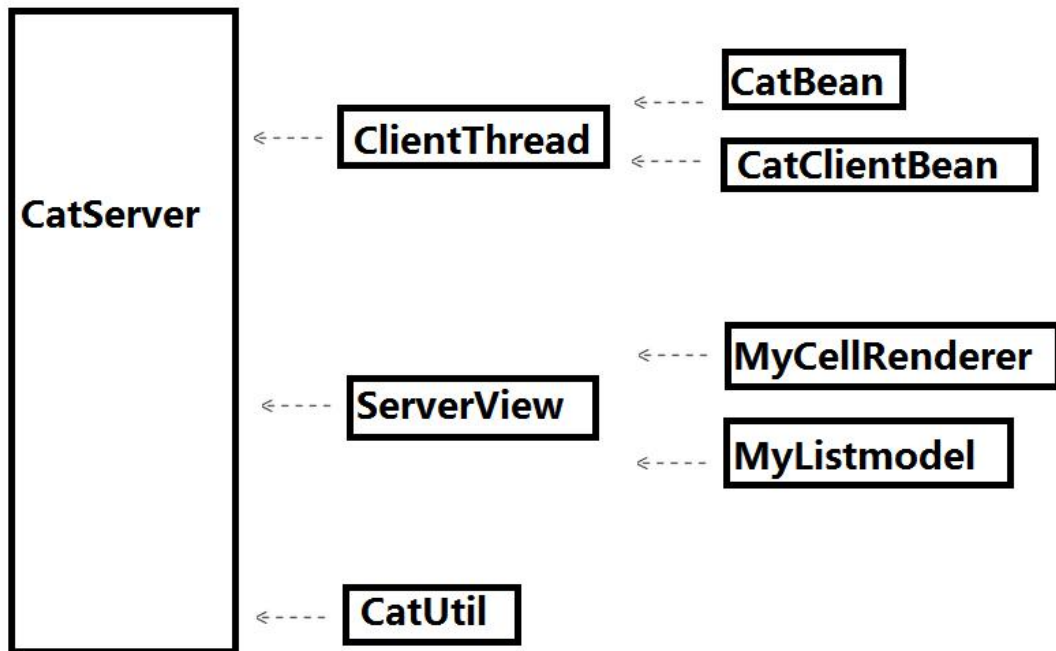
服务器端包结构:



客户端主要的依赖关系



服务器端主要依赖关系



注：界面一般是继承了JFrame或JWindow

JList、JTable、所用的model和渲染器renderer都是自己重写的
其余的一些比较琐碎的关系理起来比较复杂杂乱
程序中的控件大多是自己自定义改写的

1.3 数据存储的设计（文件等）

采用键值对的方式存储账号密码

截图默认保存在ScreenCut/+ 文件名.jpg

用到了多种流操作，以及网络传输最关键的套接字操作

1.4 界面设计

自定义渲染器和数据模型，实现JList、JTable的外观改变
最长用的是在渲染器中继承JLabel实现Renderer接口

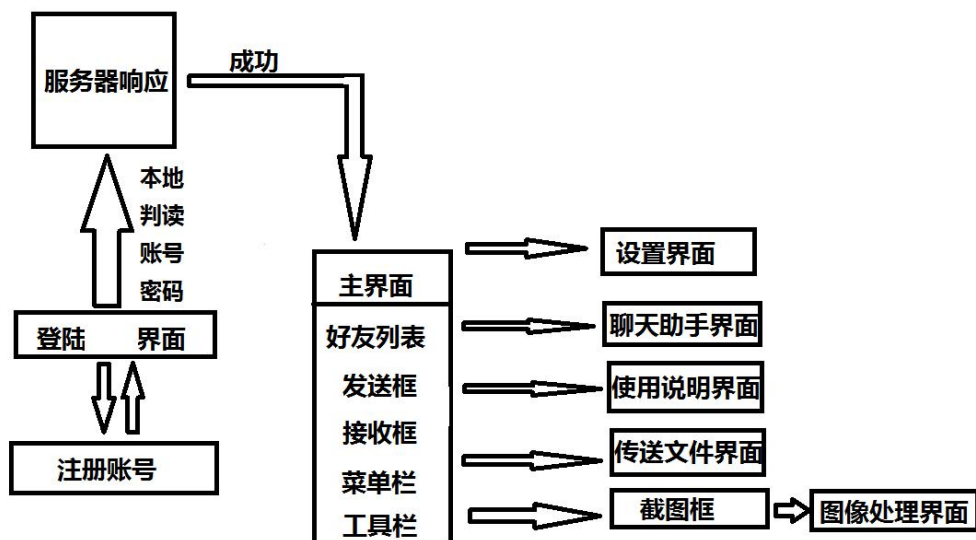
```

class MyCellRenderer extends JLabel implements ListCellRenderer{
    public MyCellRenderer() {
        this.setOpaque(true);
    }
    @Override
    public Component getListCellRendererComponent(JList list, Object value, int index, boolean isSelected,
        boolean cellHasFocus) {
        if(value!=null){
            setText(value.toString());
            ImageIcon img=new ImageIcon("images//touxiong.jpg");
            img.setImage(img.getImage().getScaledInstance(50,50,Image.SCALE_DEFAULT));
            setIcon(img);
        }
        if(isSelected){
            setBackground(list.getSelectionBackground());
            setForeground(list.getSelectionForeground());
        }else{
            setBackground(list.getBackground());
            setForeground(list.getForeground());
        }
        return this;
    }
}

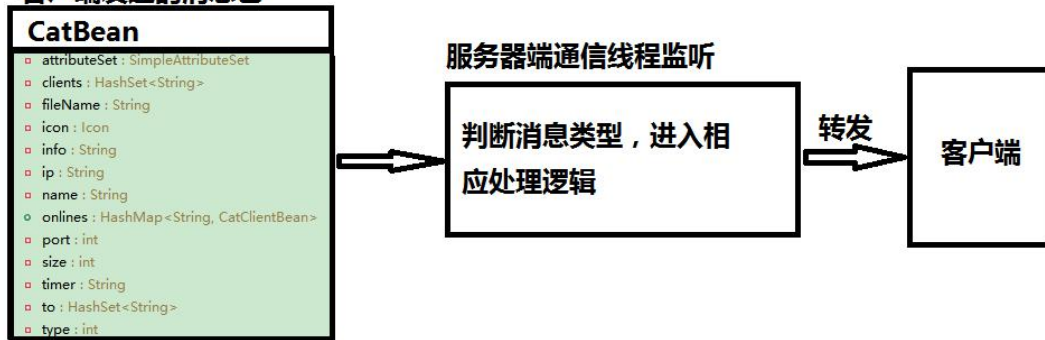
```

更多类的继承重写请参考代码

三、系统实现



客户端发送的消息包



服务器监听线程代码，客户端类似

```
class ClientThread extends Thread {  
  
    private Socket client;  
    private CatBean bean;  
    private ObjectInputStream ois;  
    private ObjectOutputStream oos;  
  
    public ClientThread(Socket client) {  
        this.client = client;  
    }  
  
    @Override  
    public void run() {  
        try {  
            // 不停的从客户端接收信息  
            while (true) {  
                // 读取从客户端接收到的catbean信息  
            }  
        }  
    }  
}
```

```
ois = new ObjectInputStream(client.getInputStream());
bean = (CatBean)ois.readObject();

// 分析catbean中, type是那样一种类型
switch (bean.getType()) {
// 上下线更新
case 0: { // 上线
    // 记录上线客户的用户名和端口在clientbean中
    CatClientBean cbean = new CatClientBean();
    cbean.setName(bean.getName());
    cbean.setSocket(client);
    // 添加在线用户
    onlines.put(bean.getName(), cbean);

    new Thread(){
        public void run() {
            myListModel.addElement(bean.getName());
            Object[] data = new Object[3];

            data[0]=(bean.getName());
            data[1]=(onlines.get(bean.getName()).getSocket().getLocalSocketAddress().toString());
            data[2]=(CatUtil.getTimer());

            defaultTableModel.addRow(data);
            serverView.list.setModel(myListModel);
            serverView.table.setModel(defaultTableModel);
        }
    }.start();

    // 创建服务器的catbean, 并发送给客户端
    CatBean serverBean = new CatBean();
    serverBean.setType(0);
    serverBean.setInfo(bean.getTimer() + " "
        + bean.getName() + "上线了");
```

```
// 通知所有客户有人上线
HashSet<String> set = new HashSet<String>();
// 客户昵称
set.addAll(onlines.keySet());
serverBean.setClients(set);
sendAll(serverBean);
break;
}
case -1: { // 下线
// 创建服务器的catbean , 并发送给客户端
CatBean serverBean = new CatBean();
serverBean.setType(-1);

try {
    oos = new ObjectOutputStream(
        client.getOutputStream());
    oos.writeObject(serverBean);
    oos.flush();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

onlines.remove(bean.getName());
new Thread(){
    public void run() {
        for(int i =0;i<defaultTableModel.getRowCount();i++){
            if(defaultTableModel.getValueAt(i,0).equals(bean.getName())){
                defaultTableModel.removeRow(i);
            }
        }
        myListmodel.removeElement(bean.getName());
        serverView.list.setModel(myListmodel);
    }
}
```

```
        };
    }.start();
    // 向剩下的在线用户发送有人离开的通知
    CatBean serverBean2 = new CatBean();
    serverBean2.setInfo("\r\n"+bean.getTimer() + " "
        + bean.getName() + " " + "下线了");
    serverBean2.setType(0);
    HashSet<String> set = new HashSet<String>();
    set.addAll(online.keySet());
    serverBean2.setClients(set);

    sendAll(serverBean2);
    return;
}

case 1: { // 聊天

    // 创建服务器的catbean，并发送给客户端
    CatBean serverBean = new CatBean();
    serverBean.setType(1);
    serverBean.setClients(bean.getClients());
    serverBean.setInfo(bean.getInfo());
    serverBean.setName(bean.getName());
    if(bean.getAttributeSet()!=null){
        serverBean.setAttributeSet(bean.getAttributeSet());
    }
    serverBean.setTimer(bean.getTimer());
    // 向选中的客户发送数据
    sendMessage(serverBean);
    break;
}

case 2: { // 请求接受文件
    // 创建服务器的catbean，并发送给客户端
    CatBean serverBean = new CatBean();
```

```
String info = bean.getTimer() + " " + bean.getName()
+ "向你传送文件,是否需要接受";

serverBean.setType(2);
serverBean.setClients(bean.getClients()); // 这是发送的目的地
serverBean.setFileName(bean.getFileName()); // 文件名称
serverBean.setSize(bean.getSize()); // 文件大小
serverBean.setInfo(info);
serverBean.setName(bean.getName()); // 来源
serverBean.setTimer(bean.getTimer());
// 向选中的客户发送数据
sendMessage(serverBean);

break;
}
case 3: { // 确定接收文件
    CatBean serverBean = new CatBean();

    serverBean.setType(3);

    serverBean.setClients(bean.getClients()); // 文件来源
    serverBean.setTo(bean.getTo()); // 文件目的地
    serverBean.setFileName(bean.getFileName()); // 文件名称
    serverBean.setIp(bean.getIp());
    serverBean.setPort(bean.getPort());
    serverBean.setName(bean.getName()); // 接收的客户名称
    serverBean.setTimer(bean.getTimer());
    // 通知文件来源的客户, 对方确定接收文件
    sendMessage(serverBean);

    break;
}
case 4: {
    CatBean serverBean = new CatBean();
    serverBean.setType(4);
    serverBean.setClients(bean.getClients());
    serverBean.setTo(bean.getTo());
    serverBean.setFileName(bean.getFileName());
    serverBean.setInfo(bean.getInfo());
    serverBean.setName(bean.getName());
```

```
        serverBean.setTimer(bean.getTimer());
        sendMessage(serverBean);
        break;
    }
    case 9: {
        CatBean serverBean = new CatBean();

        serverBean.setType(9);
        serverBean.setClients(bean.getClients());
        serverBean.setTo(bean.getTo());

        serverBean.setName(bean.getName());
        serverBean.setTimer(bean.getTimer());
        sendMessage(serverBean);

        break;
    }
    case 8:{
        CatBean serverBean = new CatBean();

        serverBean.setType(8);
        serverBean.setIcon(bean.getIcon());
        serverBean.setClients(bean.getClients());
        serverBean.setTo(bean.getTo());

        serverBean.setName(bean.getName());
        serverBean.setTimer(bean.getTimer());
        sendMessage(serverBean);

        break;
    }
    default: {
        break;
    }
}
```



```

    }
}
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (ClassNotFoundException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} finally {
    close();
}
}
}

```

四、系统测试

模块测试与系统测试：

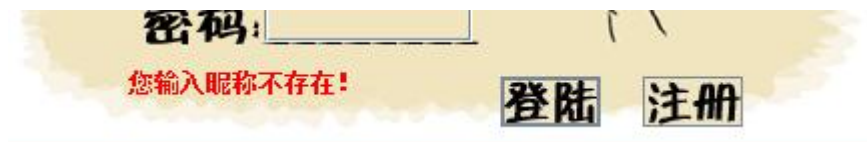
<1>登陆测试：输入已注册和未注册账号密码、未启用服务器

登陆



<2>实际运行程序得到：均能正确提示和正确登陆





<3>测试结论：功能正常实现

<1>注册测试：输入不一样的密码，已存在的账号



<2>实际运行程序得到：均能正确提示和正确写入文件



```
1 #Copyright (c) Boxcode Studio
2 #Tue May 10 18:05:04 CST 2016
3 \u5C0F\u950B=xiaofeng
4 4=4
5 3=3
6 \u5C0F\u767D=xiaobai
7 999=1
8 2=2
9 1=1
10
```

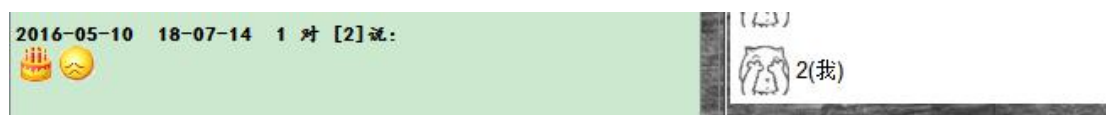
<3>测试结论：功能正常实现

<1>表情测试：

表情框



<2>实际运行程序得到：发送与显示表情



<3>测试结论：功能基本正常实现，显示部分效果不如人意

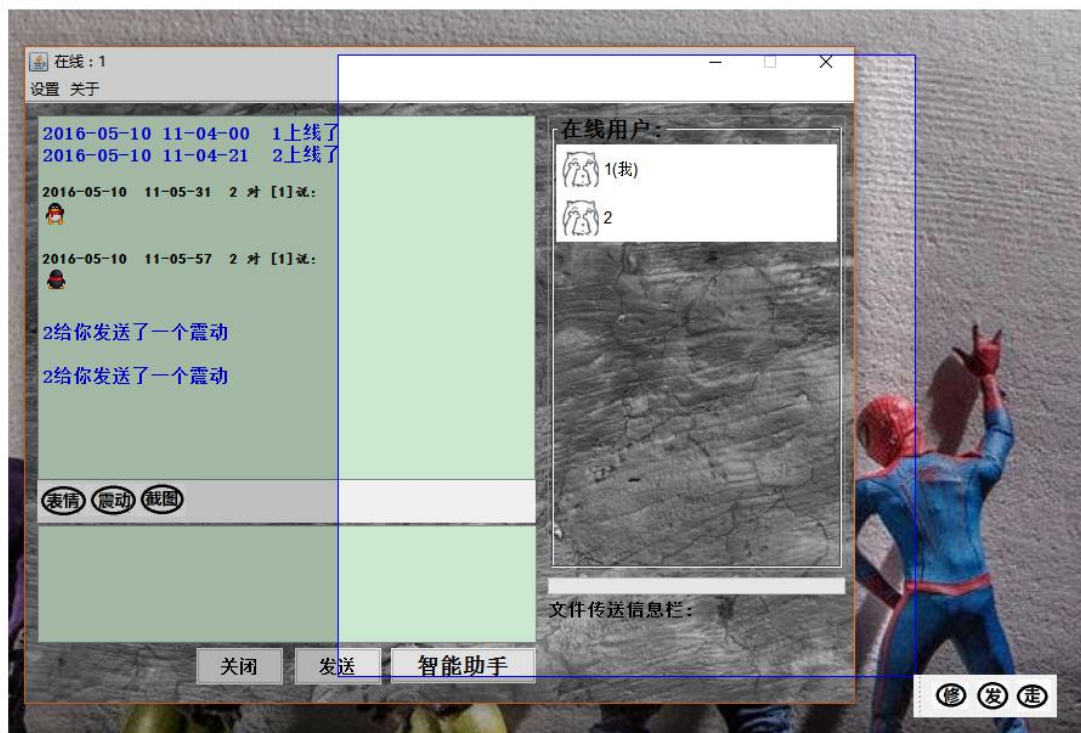
<1>震动测试，实际运行程序得到：

震动效果无法截图

<2>测试结论：功能正常实现

<1>截图测试：截图与发送

截图框



<2>实际运行程序得到：

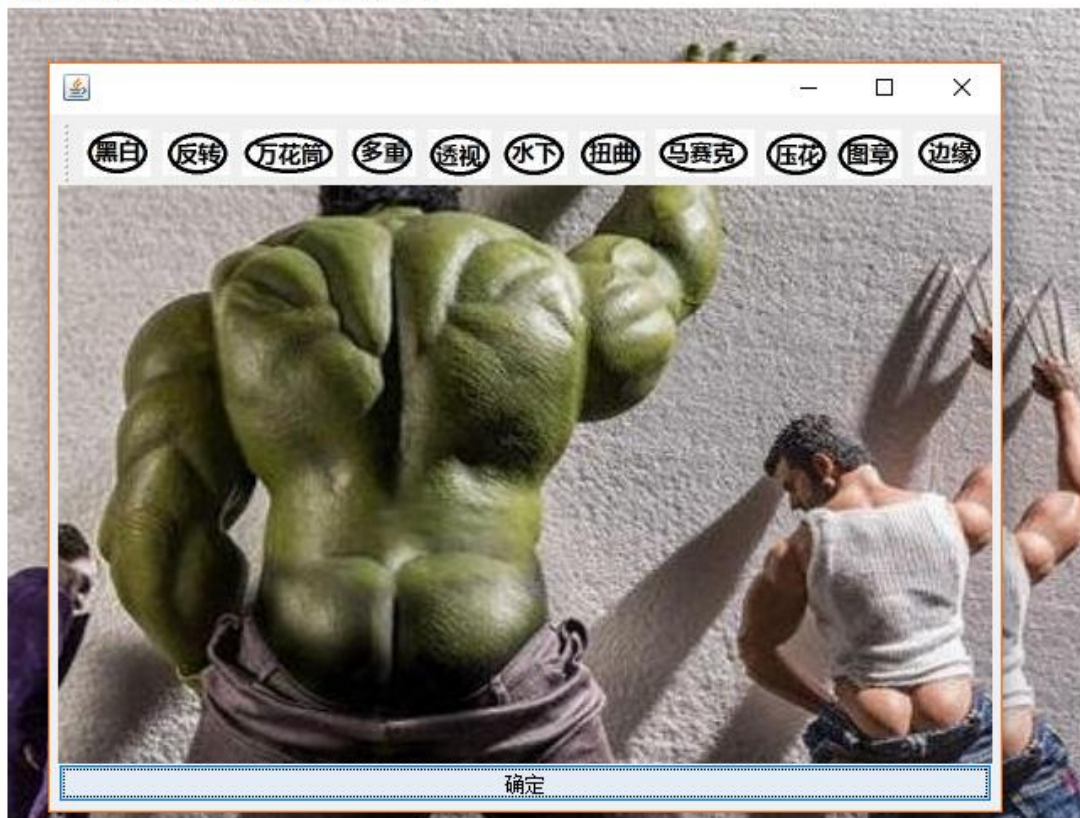
发送截图



<3>测试结论：功能正常实现

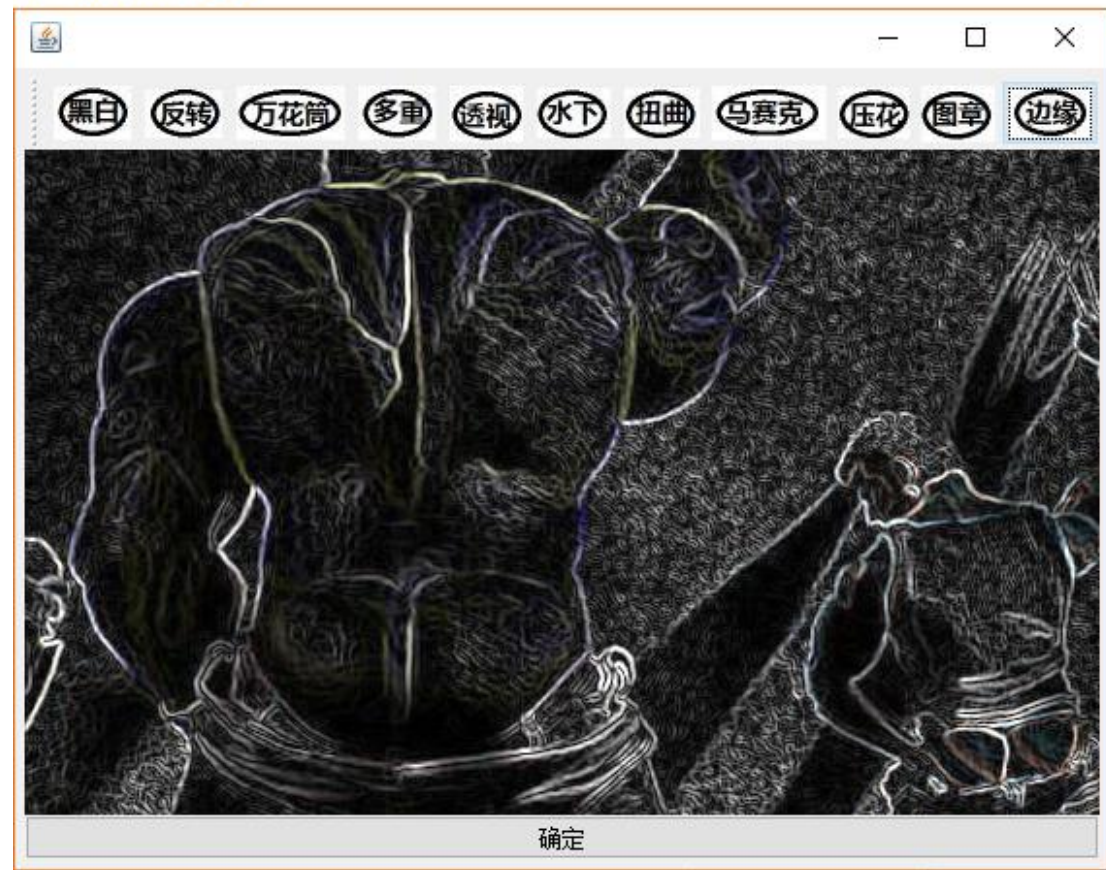
<1>滤镜效果测试：

截图图片简单滤镜处理界面



<2>实际运行程序得到：能实现并发送

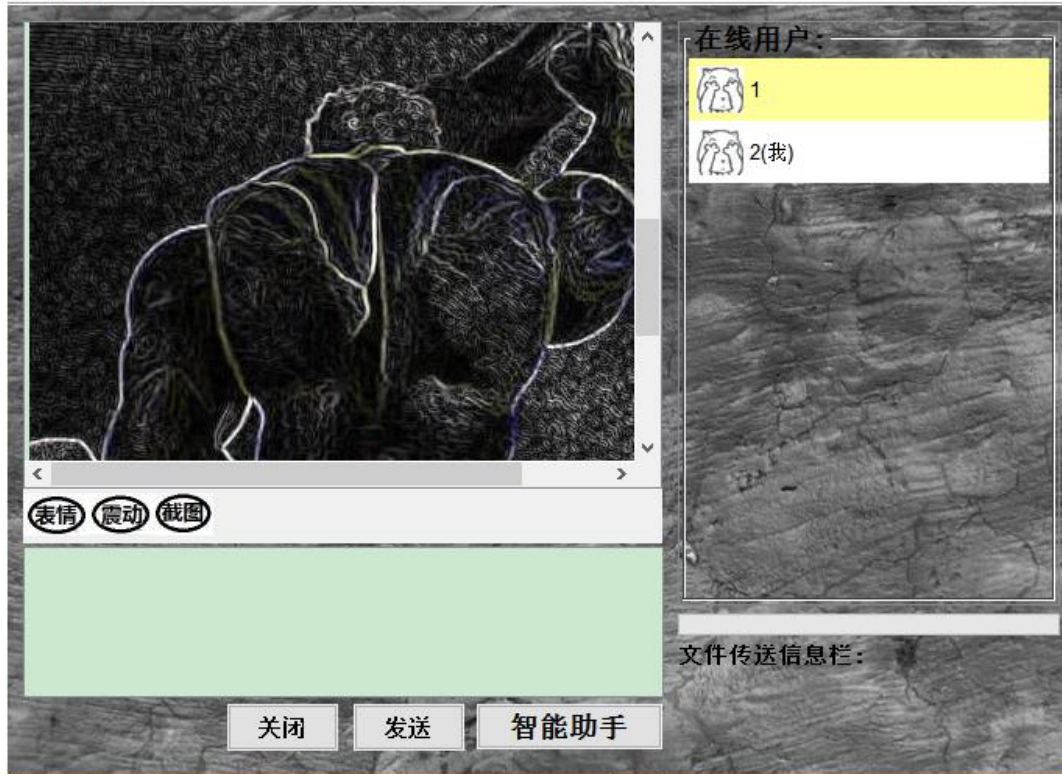
边缘效果展示



发送截图

在线: 2

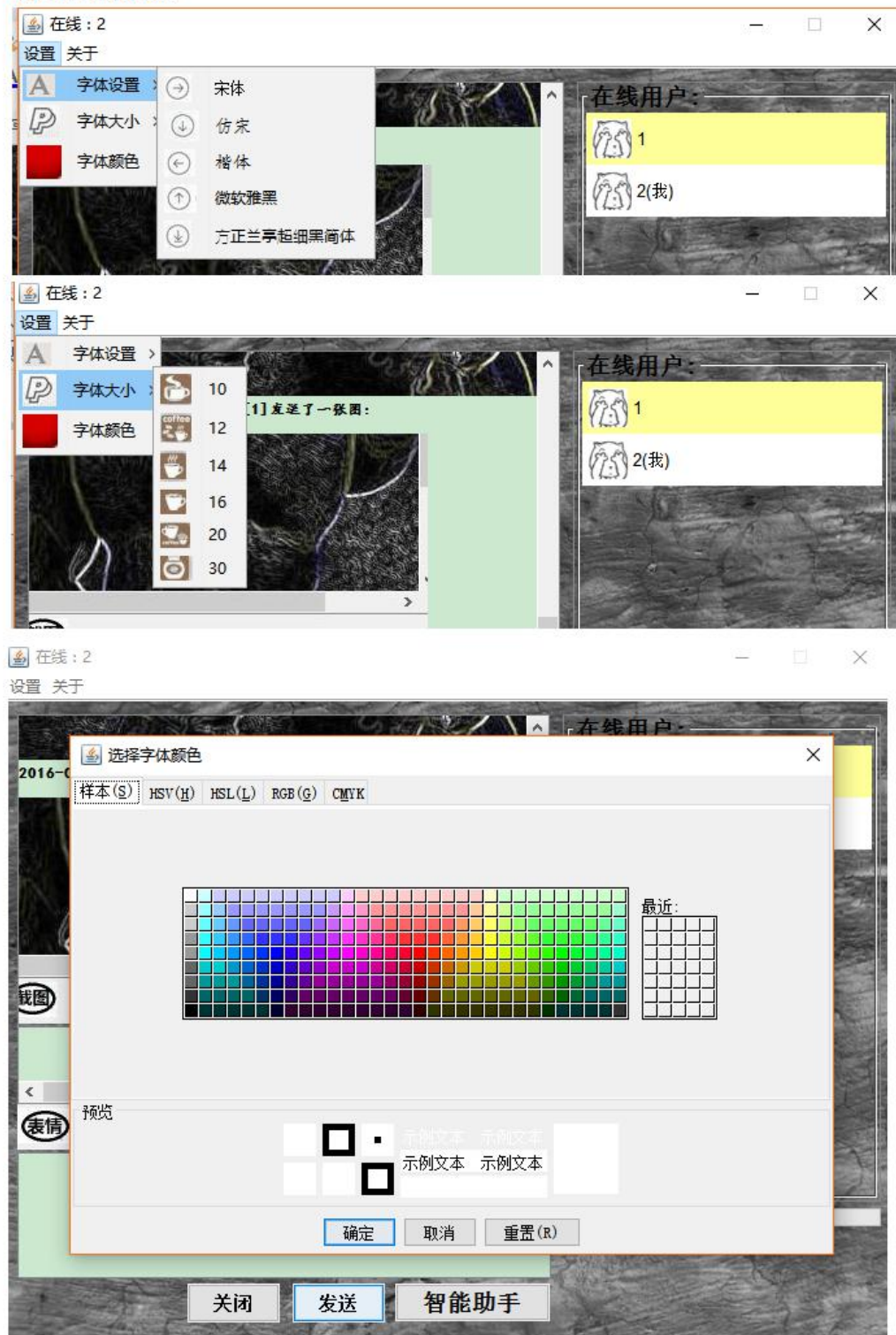
设置 关于



<3>测试结论: 功能正常实现

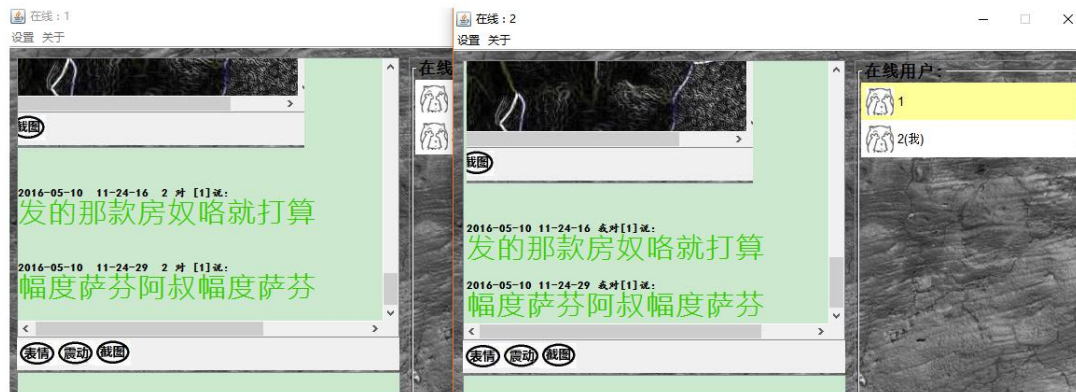
<1>设置测试: 富文本文字属性设置与传送

设置界面展示



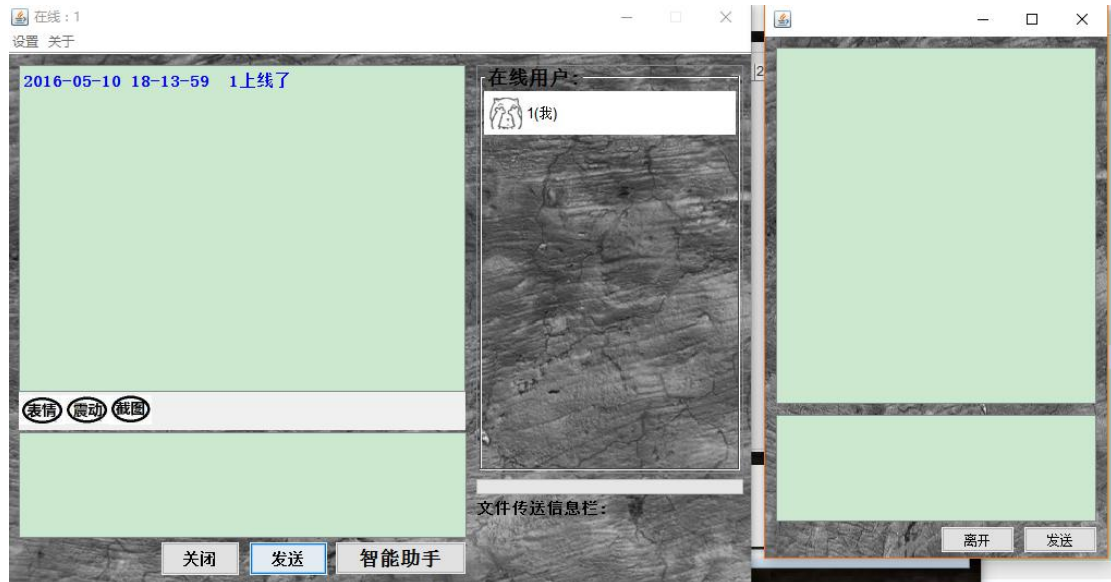
<2>实际运行程序得到:

富文本效果传送展示



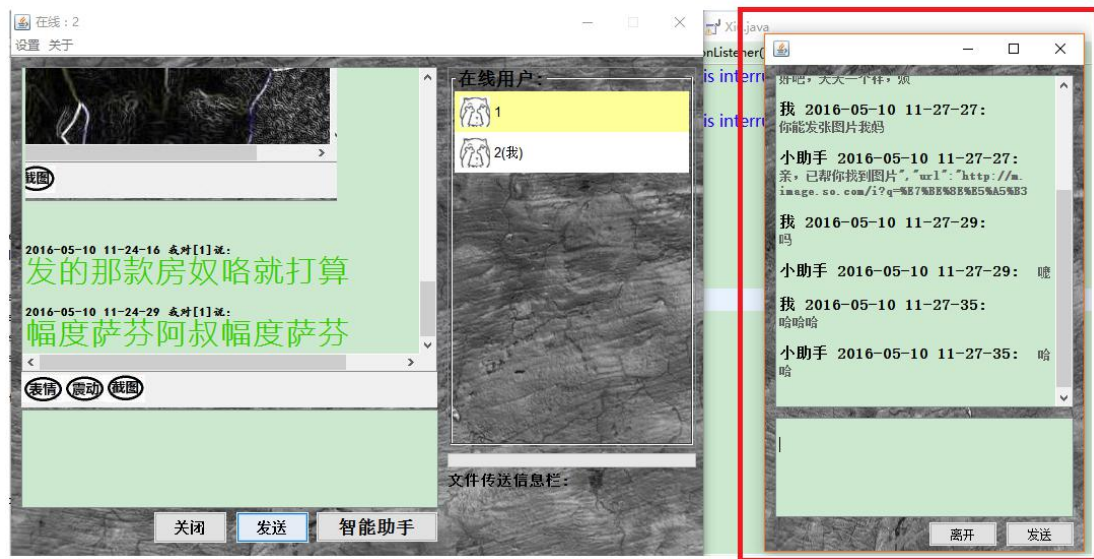
<3>测试结论：功能正常实现

<1>机器人聊天测试：



<2>实际运行程序得到：

聊天机器人界面

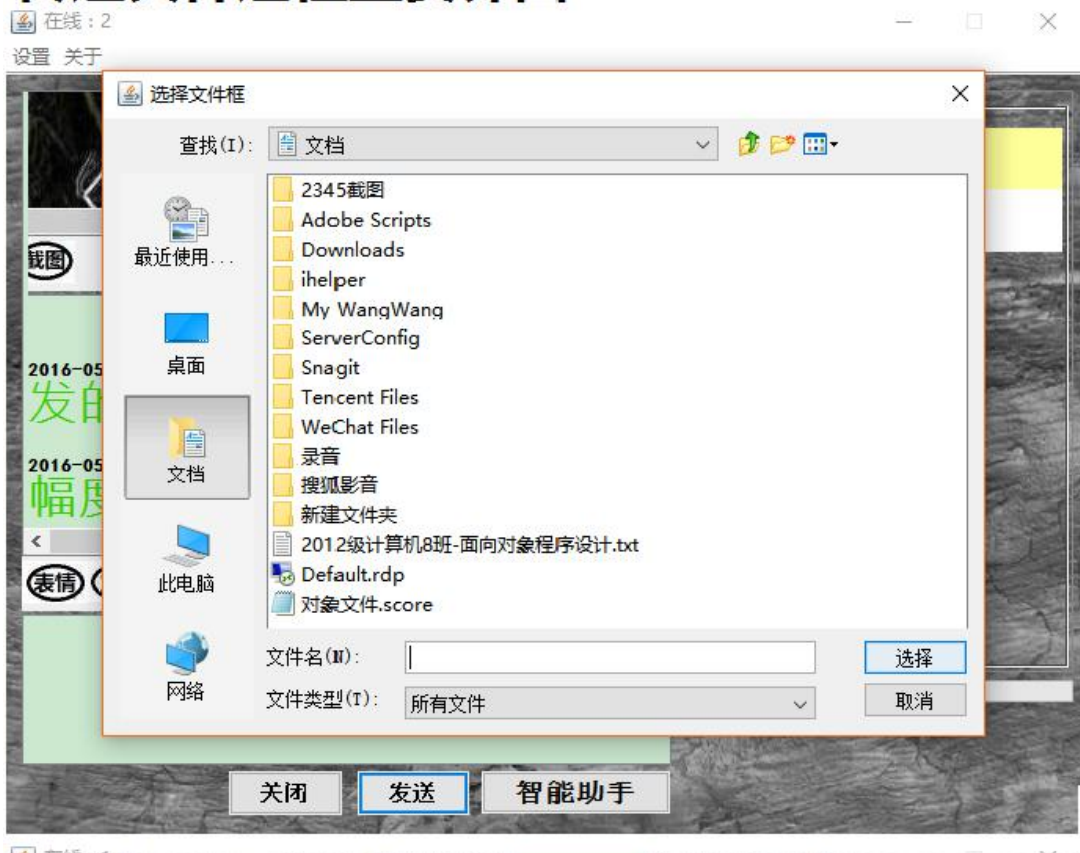


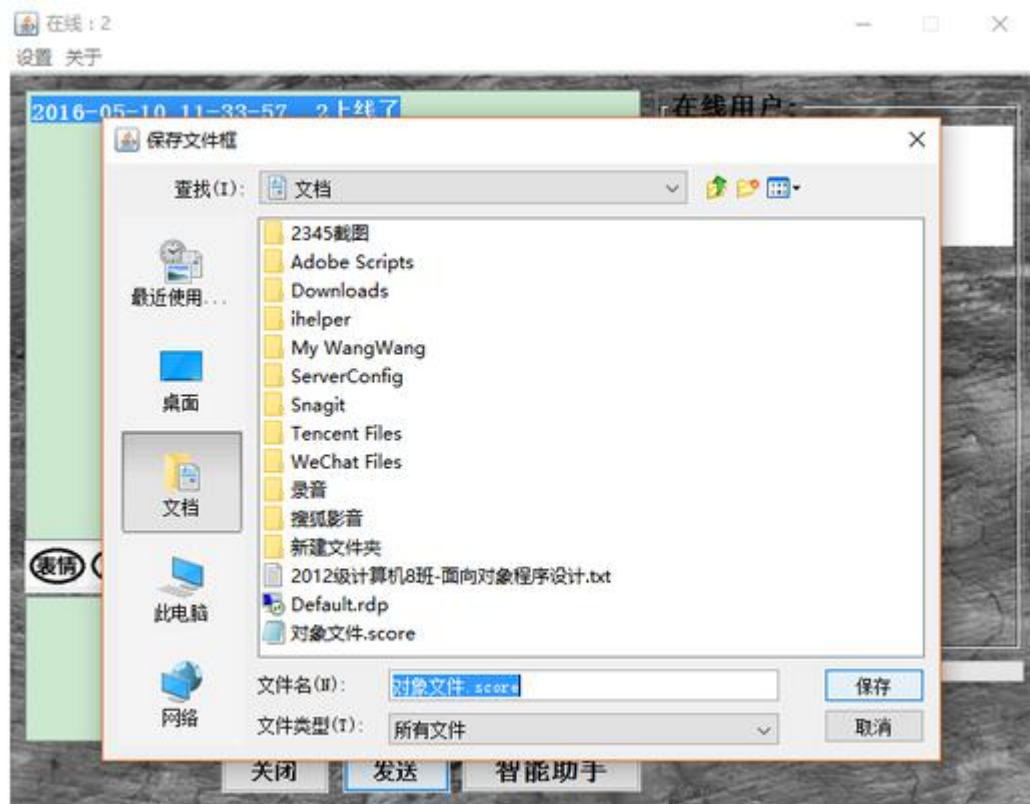
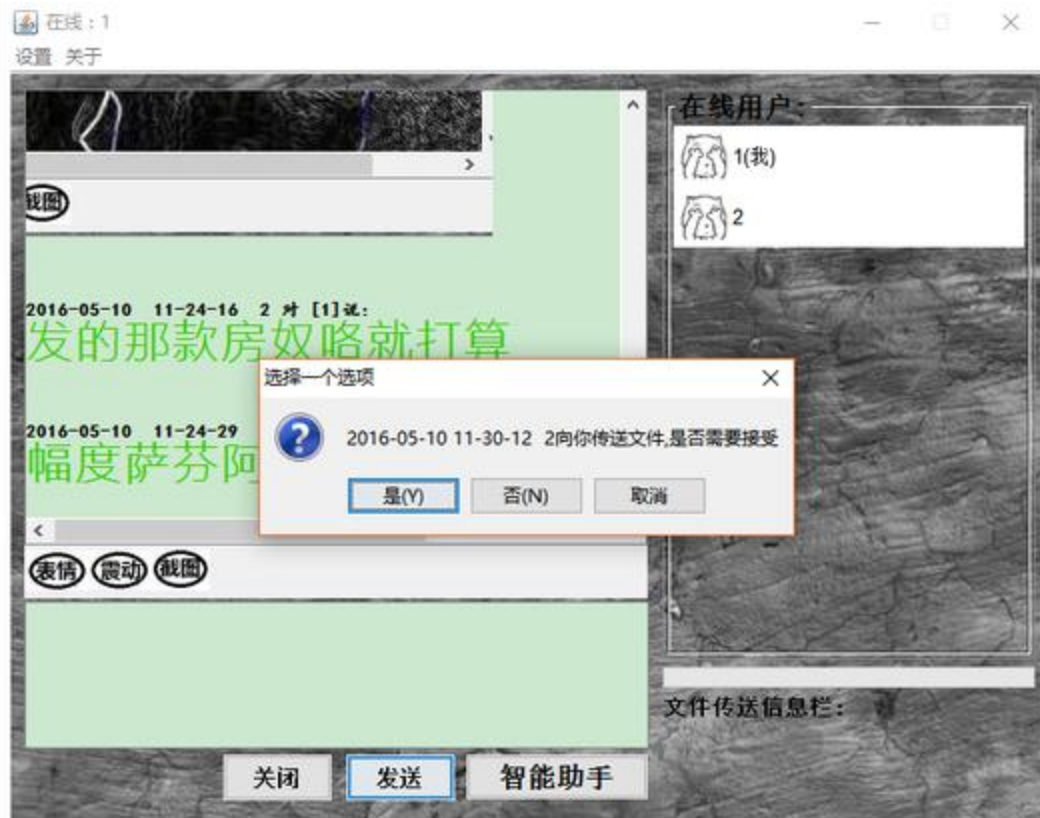
<3>测试结论：功能正常实现

<1>文件传输测试：传输文件

<2>实际运行程序得到：均能正确传输和保存文件

传送文件过程主要界面







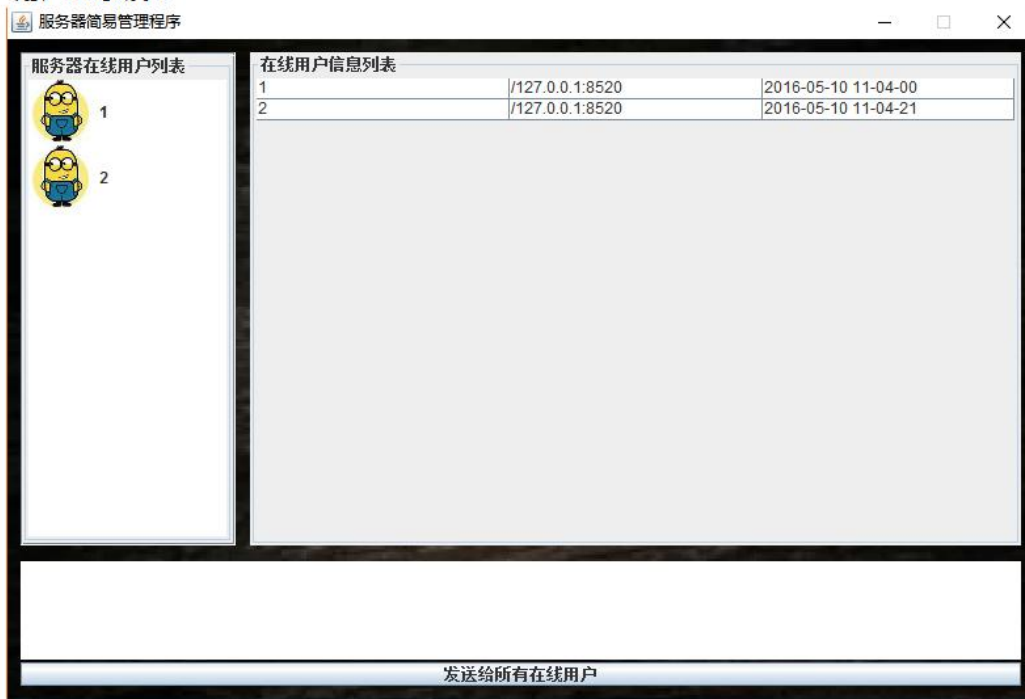
<3>测试结论: 功能正常实现

<1>服务器管理界面测试: 显示登陆用户与现实用户信息



<2>实际运行程序得到：均能正确显示

用户登录后：



<3>测试结论：功能正常实现

五、系统运行界面

子功能显示界面可从测试模块查看





六、总结

张涛：

通过这次的课程设计，可以说是开阔了自己的眼界，可以总结为以下几个方面：

- 1、是对JAVA体系的更深入的了解、对JAVA图像处理初步认识、对JAVA网络通信流的操作、流的套接更进一步深刻了解。
- 2、是对网络通信的理解，CS、BS架构模式的了解。
- 3、对设计模式的初步认识与使用，线程管理这块单例模式的功能。
- 4、对JAVA的MVC模式的更深刻了解，层与层的分工明确，效率高，易管理，让人惊叹。
- 5、对线程的理解更深刻了，深谙UI主线程与处理耗时操作的子线程的逻辑处理关系。
- 6、对网络协议的初步了解，关于协议的定义了解，以及各种传送协议的效率比较有了大体。印象，Socket通信UDP和TCP通信机制初步了解。
- 7、对系统的架构设计有了初步认识，缓存机制，分布式系统等，虽然代码中还未实现，程序也并不是面向抽象编程和面向接口编程，代码低耦高聚效果一般，但仍然受益。

郑紫月：

经过这次的 java 课程设计，我收获良多。

知识方面：我在课程设计的任务中主要负责 jlist jtable 覆盖重写、基础类的继承重写、接口实现、界面设计与美化等。玩了半个假期后很多学过的知识都记不清楚了，所以打码的过程中经常翻书，让我又重温了知识；一部分知识书上没有，还需要上网查询或者找同学帮忙解决。有很多瓶颈时候，但坚持过去，看到一起完成的作品会有很大的满足感；其他方面：经过这次课程设计任务，我又一次认识到了团队合作的力量和重要性。一起讨论问题：苦恼过，失落过，兴奋过，到最后的成就感，让我成长，也对自己有了很大的信心。在这个

祝春雨：

过程中，我也养成了沉静思考问题的习惯，这对我以后学习和工作都有很大的帮助

在这次课程设计，我们一起讨论要实现这个系统的哪些功能，把各自的想法说出来研究，我们还根据各自所学之长来分配工作，让我意识到在团队合作里每个人都能分享自己的想法、找到自己的位置发挥所长很重要，这样才能让我们更好地完成我们的工作。在完成我们的任务的过程中，我把每个功能逐步实现，比如在实现登录功能的时候，我会先把输入正确的账号和密码登录成功的功能实现，然后再实现判断输入的账号和密码是否一致，若不一致就返回重新输入账号密码这一功能，这让我在发现错误的时候更加容易找出并解决。在此过程虽然遇到许多困难，但是我都会去研究课本和课件里的例题或者上网去看教学视频，一步一步测试，自己实在无法解决了就去找同学帮助。

经过这次课程设计，我对 Java 有了更深的了解，但这还远远不够，为了未来的发展，我必须更加努力地去学习更广更深的知识。

