|  |  |  |
| --- | --- | --- |
| 南农大  **网络工程课程设计**  说明: 南农校徽 | | |
|  | 题目： | 基于UDP协议的QQ聊天室系统 |
|  | 班级： | 计科162 |
|  | 学号： | 19216228 |
|  | 专业： | 计算机科学与技术 |
|  | 姓名： | . 黎远波 |
|  | 学院： | 信息科学与技术 |
|  | 指导教师： | 叶锡君 |
|  |  |  |
| 2019年10月28日 | | |
| 南京农业大学教务处制 | | |

目录

[1 目的及意义 3](#_Toc529891078)

[2 设计任务及要求 4](#_Toc529891079)

[3 系统设计 5](#_Toc529891080)

[3.1 功能设计 5](#_Toc529891081)

[3.2通用数据结构及方法设计 6](#_Toc529891082)

[3.2.1 消息类 6](#_Toc529891083)

[3.2.2 用户类 7](#_Toc529891084)

[3.2.3 文件类 7](#_Toc529891085)

[3.2.4 消息序列化和反序列化 7](#_Toc529891086)

[3.3用户注册登录退出 8](#_Toc529891087)

[3.3.1 用户注册 8](#_Toc529891088)

[3.3.2 用户登录 9](#_Toc529891089)

[3.3.3 用户退出 10](#_Toc529891090)

[3.4 会话消息功能 10](#_Toc529891091)

[3.4.1 群聊消息 10](#_Toc529891092)

[3.4.2 私聊消息 11](#_Toc529891093)

[3.5文件操作 12](#_Toc529891094)

[3.5.1 文件上传 12](#_Toc529891095)

[3.5.2 文件下载 12](#_Toc529891096)

[3.6 服务器端线程设计 13](#_Toc529891097)

[3.7 客户端线程设计 15](#_Toc529891098)

[4 系统测试 16](#_Toc529891099)

[4.1 用户注册登录 16](#_Toc529891100)

[4.1.1 用户注册 16](#_Toc529891101)

[4.1.2 用户登录 17](#_Toc529891102)

[4.2 用户聊天 19](#_Toc529891103)

[4.2.1 聊天室群聊 19](#_Toc529891104)

[4.2.2 私聊消息 19](#_Toc529891105)

[4.3 文件操作 20](#_Toc529891106)

[4.3.1 上传文件 20](#_Toc529891107)

[4.3.2 下载文件 21](#_Toc529891108)

[4.4 用户退出 23](#_Toc529891109)

[5 实验总结 24](#_Toc529891110)

**基于UDP协议的QQ聊天室系统**

计算机专业学生 黎远波

指导教师 叶锡君

**摘要：**为了方便多人聊天沟通，承载多用户即时聊天功能，QQ聊天室为大家提供了一种方便的网上交流模式，使得在线多用户可同时多人聊天，分享信息，促进交流沟通。本系统基于UDP协议，利用客户端及服务器端的消息传送回应转发，实现多人聊天室。用户可实现注册、登录、退出客户端，并且在登录和退出时均伴有音效；用户可在聊天室内自由发言进行群聊，也可点击用户列表与某一用户私聊；同时，用户还可以在聊天室内上传文件，其他用户可根据需要点击文件列表下载文件。本系统以促进多人沟通为目的，以UDP协议为依托，通过Java网络编程技术，实现客户端与服务器端、客户端与客户端之间的联系与交流。

**关键字：**QQ聊天室；Socket网络编程；UDP协议；客户端与服务器；文件上传与下载

# 1 目的及意义

计算机技术的快速发展，特别是计算机网络的发展，正在越来越深刻的改变人们生活的各方各面。各种在线服务系统，更是深刻的影响了人们的联系和交流方式，使得人们可以在千里之遥随时通讯。过去的种种陈旧的联系方式，已经不能满足现代生活的需要。其中网上聊天系统为人们进行及时的交流联系提供了一个便捷的平台，作为一种方便人们之间联系的实用系统得到了广泛的传播和大量的使用。通过完善的系统管理和贴合使用者需求的系统设计，可以实现增进人与人之间交流和联系的目的，也可以很大程度上推动社会其他领域的发展。

UDP（User Datagram Protocol,用户数据报协议）属于传输层协议。它是一个面向非连接的协议，有很多优良的特性。UDP协议传输数据时源端和终端不建立连接，而是直接把客户端发的数据报发给服务器端，因此不需要维护复杂地连接状态，这使它适用于一次传输数据量很少、对可靠性要求不高的或对实时性要求高的应用场景。UDP协议网络通信框架如图1所示。UDP协议地常见服务有DNS、SNMP、QQ等[1]。



图1 UDP通信框架

Socket，也称作套接字，是一个通信链的句柄，用于描述IP地址和端口，可以用来实现不同虚拟机或不同计算机之间的通信，应用程序通常通过Socket向网络发出请求或者应答网络请求[2]。Socket结构会指明套接字的连接状态、可对套接字进行的操作以及套接字的通信类型等信息。一个套接字的连接状态可能有几种情况：套接字是空闲的，还没进行相应的端口和地址绑定；还没连接；正在连接中；已经连接；正在解除连接[3]。实际应用中，在Internet上的主机一般运行了多个服务软件，同时提供几种服务。每种服务都打开一个Socket，并绑定到一个端口上，不同的端口对应于不同的服务。Socket是连接运行在网络上的两个程序间的双向通信的端点，网络通讯其实指的就是Socket间的通讯，通讯的两端都有Socket，数据在两个Socket之间通过IO来进行传输[4]。Socket网络通信模式如图2所示。



图2 Socket网络通信模式

服务器程序将一个套接字绑定到一个特定的端口，并通过此套接字等待和监听客户的连接请求。客户程序根据服务器程序所在的主机和端口号发出连接请求。 如果一切正常，服务器接受连接请求。并获得一个新的绑定到不同端口地址的套接字。客户和服务器通过读、写套接字进行通讯。

Java语言是一种面向对象的可以编写跨平台应用软件的高级程序设计语言。它为网络编程提供了强大的网络编程接口。针对网络通信的不同层次，Java有四大类网络功能：InetAddress、URL、Socket、Datagram [5]。Java.net包中提供Socket类，隐藏了Socket的实现细节，不需要开发者编写接口程序，而可以快速的实现网络的通信[6]。

在Java中可以使用两种Socket方式，即流式Socket和数据报Socket。流式Socket提供了双向的、有序的、无重复的数据流服务，采用的是TCP服务。数据报Socket支持双向的数据流，但不保证是可靠传输，使用的是UDP服务[7]。为了提高数据传输效率和网络利用率，本文采用的是数据报Socket方式。

此次课程设计一方面可以帮助我们进一步熟悉掌握网络通信技术，客户端与服务器端的交互模式等方面的知识，巩固自身所学；另一方面，可以通过实现聊天室系统，提高小组成员的系统设计能力以及Java编程能力，为之后进一步应用打下坚实的基础。

# 2 设计任务及要求

本次课程设计拟完成一个基于UDP协议的QQ聊天室系统，通过QQ客户端及服务器端的消息传送回应转发，实现多人聊天室系统。

用户可以通过聊天室系统注册、登录客户端，用户上线时会伴有上线音效，其他用户会收到该用户上线提示音，也会将上线用户加入在线用户列表。用户登录客户端之后，可在聊天室内进行自由发言，通过服务器的消息转发，实现多用户聊天功能。用户也可以通过点击在线用户列表与某一用户进行私聊，被私聊用户会收到其他用户的私聊消息，可以通过点击用户列表的该用户进行私聊消息回应。同时，用户可以在聊天室聊天过程中选择文件进行上传，其他用户可以在文件列表中点击选择进行文件下载。当用户聊天结束退出客户端时，其他用户也会收到该用户下线提示音，并将该用户在在线用户内删除。

本次课程设计完成的内容有：用户注册、登录客户端功能，用户群聊功能，用户私聊功能，用户上传及下载文件功能，用户退出客户端功能，用户上线下线提示音等功能。

小组分工：

（1）宋玉红：聊天界面、群聊与私聊功能实现、文件上传与下载操作实现、服务器与客户端线程实现、报告撰写；

（2）孙云晓：注册界面、私聊界面、注册功能实现、退出功能实现、服务器监听功能实现、报告撰写；

（3）延悦帆：登录界面、服务器界面、登录功能实现、报告撰写。

# 3 系统设计

## 3.1 功能设计

本次课程设计完成的功能有：

（1）注册、登录系统：当用户打开登录界面时，在用户列表已有账号的前提下，可以自行登录账号；若用户为新用户，可以点击注册按钮进行新用户的注册，注册成功后即可进行登录QQ账户。

（2）用户群聊：多个用户在聊天室内可以自由发言，当用户发送聊天内容时，服务器会判断用户消息类型，将用户发送的信息进行转发，其他用户即可接收到某用户发送的消息，从而实现多人聊天沟通。

（3）用户私聊：用户可以在聊天室用户列表处，单击选择某一用户进行消息私发聊天，只有被私聊用户会收到用户的私聊消息，并可通过点击用户列表进行回复，从而实现双人单独私密聊天。

（4）上传文件：用户在交流过程中，不仅仅是文字的交流，也可以进行文件内容的分享，用户可以选择上传文件，点击上传按钮进行文件的分享。

（5）下载文件：当某用户上传文件后，该文件会被添加进聊天室文件列表，其他用户可以在文件列表中查看，选择需要的文件进行点击下载保存到本地即可。

（6）用户下线：用户在完成线上交流之后，可以选择退出客户端，进行用户下线，退出多人聊天室。

（7）音效：在用户登录系统时，会伴有上线音效，同时其他用户会收到此用户上线的提示音；在用户收到群聊消息、私聊消息、文件消息时，会伴有接收消息提示音；用户退出客户端下线时，其他用户收到此用户下线的提示音。

系统H图如图3所示：



图3 系统H图

## 3.2通用数据结构及方法设计

### 3.2.1 消息类

//定义消息结构，规定消息协议

public class Message implements Serializable{//序列化消息必须实现Serializable接口

private String userId=null;//用户ID，唯一标识

private String password=null;//用户密码

private String filename=null;//文件名

private String fileId=null;//文件Id

/\*消息类型

\* M\_LOGIN:用户登录消息

\* M\_REGISTER:用户注册消息

\* M\_SUCCESS:登录成功消息

\* M\_RSUCCESS:注册成功消息

\* M\_FAILUER:登录失败消息

\* M\_RFAILURE:注册失败消息

\* M\_ACK:服务器对客户端的回应消息

\* M\_MSG:会话消息

\* M\_PRIVATE:用户私聊消息

\* M\_QUIT:用户退出消息

\* M\_FILE：上传文件

\* M\_REQFILE:请求下载文件，服务器回应

\* \*/

private String type=null;//消息类型

private String text=null;//消息内容

private InetAddress toAddr=null;//目标用户地址

private int toPort;//目标用户端口

private String targetId=null;//目标用户ID

}

### 3.2.2 用户类

//用户对象，用于记录用户ID和用户报文

public class User {

private String UserId;//用户ID

private String Passwd;//用户密码

private DatagramPacket packet=null;//用户报文

}

### 3.2.3 文件类

//文件类，用于服务器文件存储，文件记录

public class Fileup {

private String userId;//用户ID

private String filename;//文件路径

private String filecontent;//文件内容

private String fileId;//文件Id

private DatagramPacket packet=null;//用户报文

}

### 3.2.4 消息序列化和反序列化

//用于消息序列化传输，反序列化将传送的消息进行反序列化提取

public class Translate {

//序列化：对象转化为字节数组形式obj->oo->bo->buffer

public static byte[] ObjectToByte(Object obj)

{

byte[] buffer=null;

try{

ByteArrayOutputStream bo=new ByteArrayOutputStream();//字节数组输出流

ObjectOutputStream oo=new ObjectOutputStream(bo);//对象输出流

oo.writeObject(obj);//obj写入输出流

buffer=bo.toByteArray();//对象序列化为字节数组

}catch(IOException ex){}

return buffer;

}

//反序列化：字节数组转换为对象形式buffer->bi->oi->obj

public static Object ByteToObject(byte[] buffer)

{

Object obj=null;

try{

ByteArrayInputStream bi=new ByteArrayInputStream(buffer);//字节数组输入流

ObjectInputStream oi=new ObjectInputStream(bi);//对象输入流

obj=oi.readObject();//从对象输入流中读出对象

}catch(IOException | ClassNotFoundException ex){}

return obj;

}

}

## 3.3用户注册登录退出

### 3.3.1 用户注册

点击注册按钮，打开用户注册界面，用户账号为系统设定，用户不可自行填写，用户填写密码及确认密码，向服务器请求注册。

用户注册流程图如图4所示：



图4 用户注册流程图

### 3.3.2 用户登录

用户在注册之后，进入登录界面，用户输入自己的账号密码请求登录客户端，服务器处理后对用户登录消息进行回应。

用户登录流程图如图5所示：



图5 用户登录流程图

### 3.3.3 用户退出

用户可以点击退出按钮退出群聊。

用户退出流程图如图6所示：



图6 用户退出流程图

## 3.4 会话消息功能

### 3.4.1 群聊消息

用户可通过客户端对话框进行聊天室消息群发，与众多用户一起交流沟通。

用户群聊消息流程图如图7所示：



图7 群聊消息流程图

### 3.4.2 私聊消息

用户也可点击用户列表中的某用户，打开私聊界面，对目标用户进行私密消息的发送。同时目标用户也可以同样的方式回应用户的私聊消息。

用户私聊消息流程图如图8所示：



图8 私聊消息流程图

## 3.5文件操作

### 3.5.1 文件上传

用户可以点击选择文件进行上传，上传文件将加入文件列表内。

用户上传文件流程图如图9所示：



图9 上传文件流程图

### 3.5.2 文件下载

用户可以根据需要点击文件列表请求下载文件，选择路径对文件进行保存。

用户请求下载文件流程图如图10所示：



图10 下载文件流程图

## 3.6 服务器端线程设计

服务器端在启动套接字后进行监听，等待客户端传送数据报文进行处理，针对不同的消息类型，客户端给予不同的回应报文。

服务器接收发送报文流程图如图11所示：



图11 服务器线程

## 3.7 客户端线程设计

客户端在提出不同的请求后，服务器会根据不同的请求给予回应，客户端再根据服务器给予的回应做出不同的处理，包括界面、音效的处理。

客户端接收服务器回应报文流程图如图12所示：



图12 客户端线程

# 4 系统测试

首先启动服务器，开始监听，如图13。

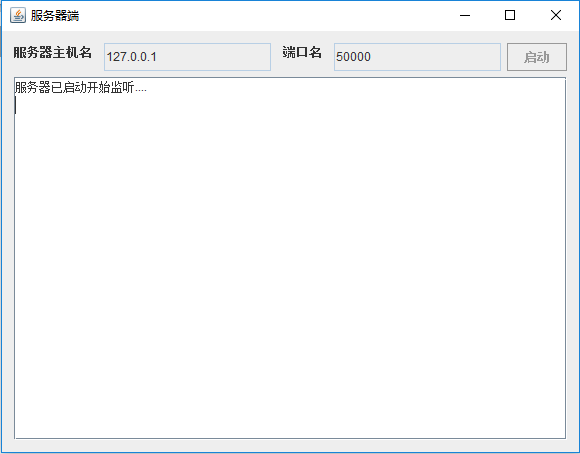


图13 启动服务器

## 4.1 用户注册登录

### 4.1.1 用户注册

输入密码及确认密码。当输入密码为空时，会弹出错误提示框，提示输入密码，如图14；当确认密码为空时，会弹出错误提示框，提示再次输入密码，如图15；当输入密码与确认密码不匹配时，会弹出错误提示框，提示密码不匹配，如图16；当输入密码与确认密码匹配时，才会提示信息，可以登录聊天室进行聊天，如图17。我们现在注册三个用户分别为1111、1112、1113，密码均为111。

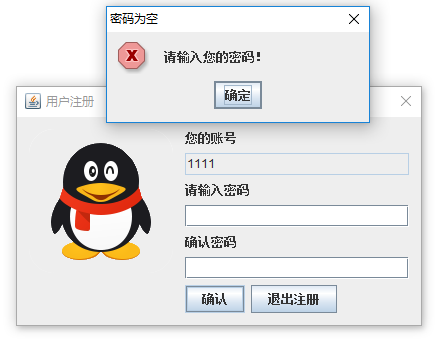


图14 密码输入提示



图15 密码再次输入提示

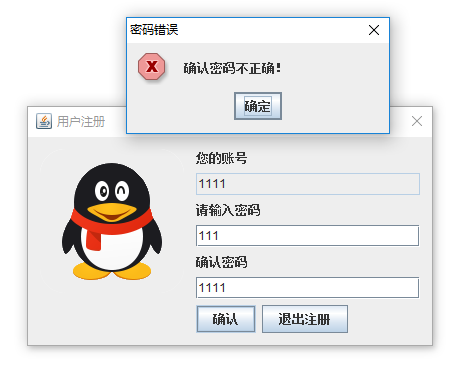


图16 密码错误提示



图17 注册成功提示

### 4.1.2 用户登录

输入账号及密码。当用户输入账号与密码不匹配时，弹出错误提示框，账号或密码输入错误，如图18；当输入账号未注册时，仍会弹出错误提示信息，如图19；当用户已注册并且账号与密码匹配时，会弹出信息提示，登录成功，可以进入聊天室进行聊天，如图20；当登录过的账号再次登录时，会弹出错误提示框，账号已在线，如图21。我们分别登录刚刚注册好的三个用户，这时你会听到用户上线提示音，以及其他用户上线提示音，随后进入聊天室客户端界面。



图18 账号密码不匹配



图19 未注册用户登录



图20 登录成功提示



图21 用户已在线

## 4.2 用户聊天

### 4.2.1 聊天室群聊

用户可以在发送消息文本框内键入消息，点击发送按钮进行发送，其他用户会收到该用户的消息，并会将消息显示在会话消息文本框中，并伴有消息提示音；同时我们也可以发现客户端右侧在线用户列表显示了当前在线用户1111、1112、1113，如图22。



图22 客户端展示

### 4.2.2 私聊消息

用户可以点击右侧在线用户列表对某一用户私发消息，系统弹出私聊窗口，用户可以在窗口内键入想要私聊的消息，点击发送按钮，就可以私发给目标用户，目标用户客户端会收到私发消息，并伴有消息提示音，如图23。

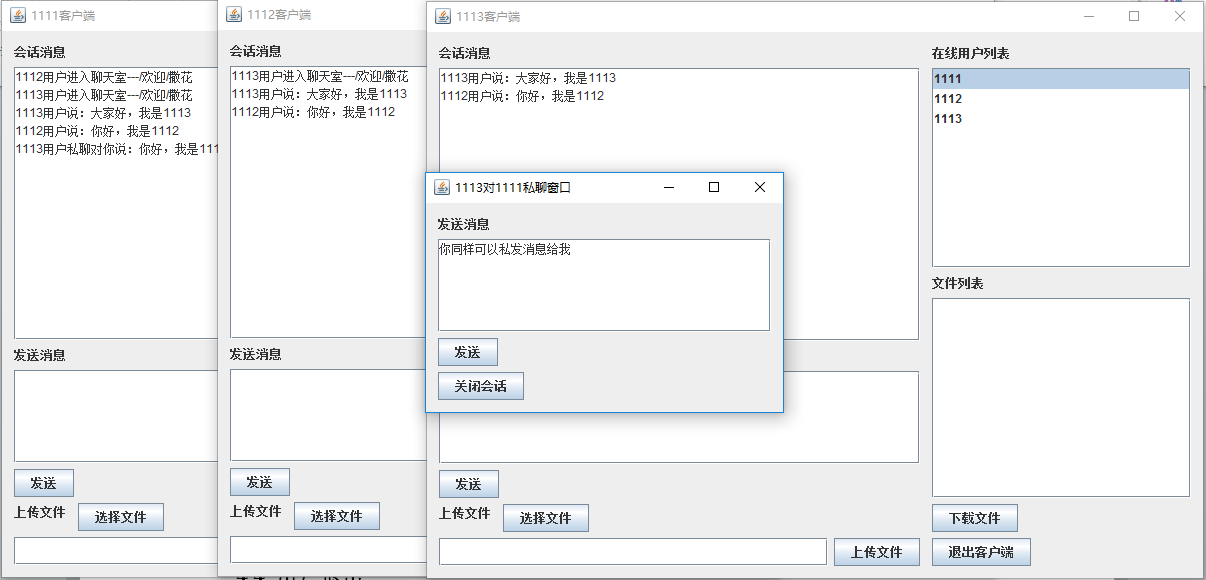


图23 消息私聊展示

## 4.3 文件操作

### 4.3.1 上传文件

用户可以点击选择文件按钮，系统会弹出选择文件对话框，选择路径下的文件，点击文件，点击“请选择要上传的文件”按钮进行确认，如图24，这时你会看到选择文件文本框内置入了选择的文件的绝对路径；点击上传文件，其他用户均会收到该用户的上传文件消息，并会将该文件简要信息加入文件列表内，如图25。

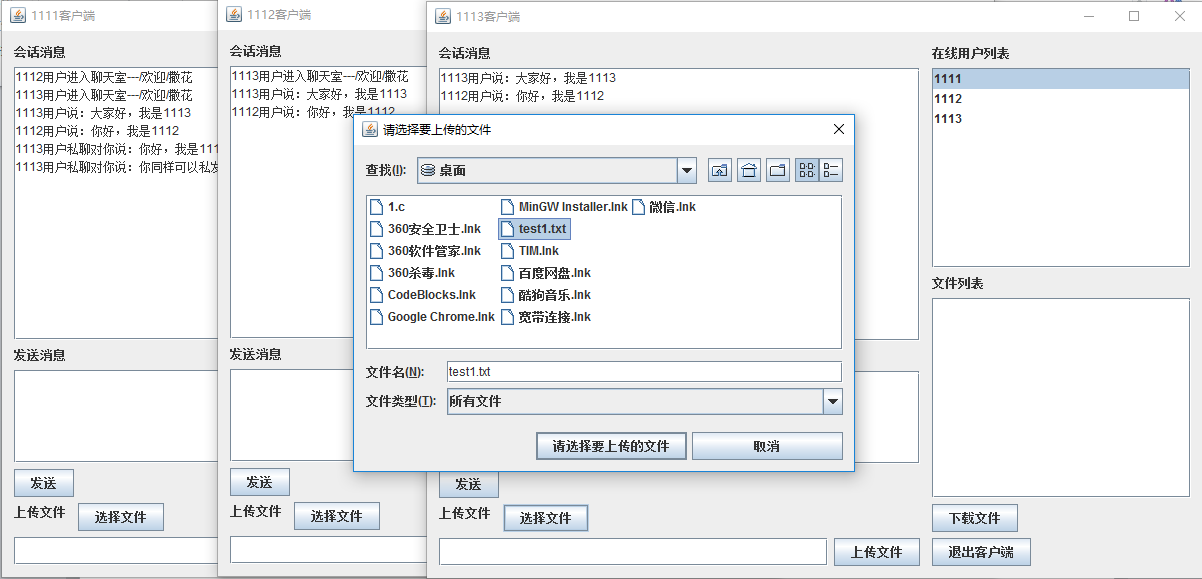


图24 上传文件展示



图25 上传文件成功展示

### 4.3.2 下载文件

用户可以选择点击文件列表，点击下载文件按钮，系统会弹出保存文件提示框，选择想要保存的文件路径，输入文件名及扩展名，点击保存按钮，即可将文件保存到指定路径下，如图26；我们选择将文件保存到桌面上，可以看出文件test2.txt已经被保存到桌面上，当保存文件时不写扩展名，保存的文件会没有类型，如图27；打开文件内容查看就是我们所上传的文件中的内容，如图28、29。

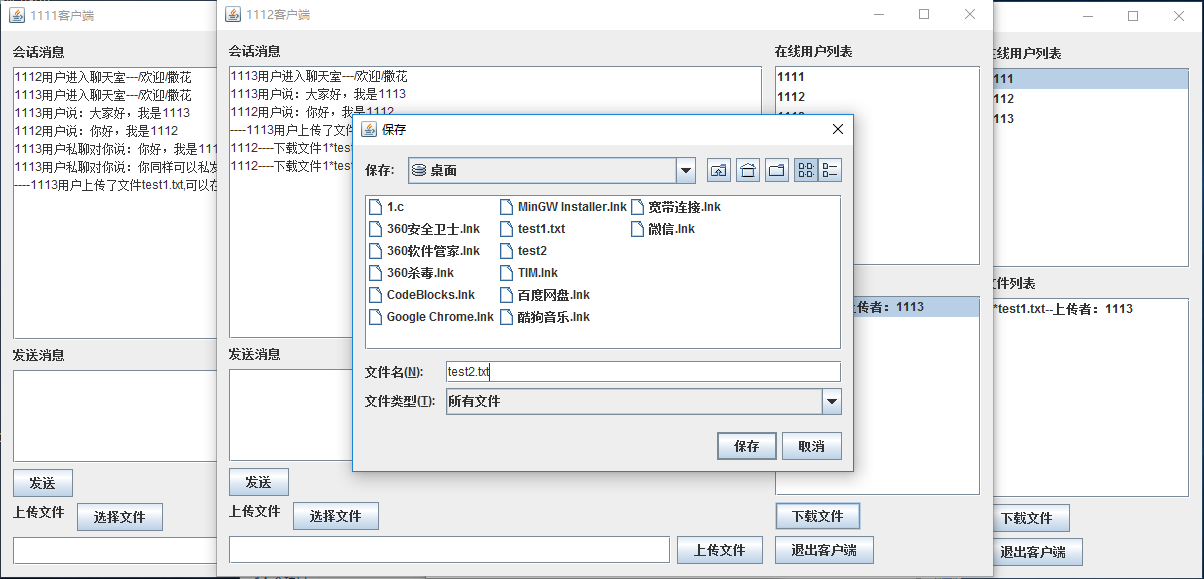


图26 下载文件展示

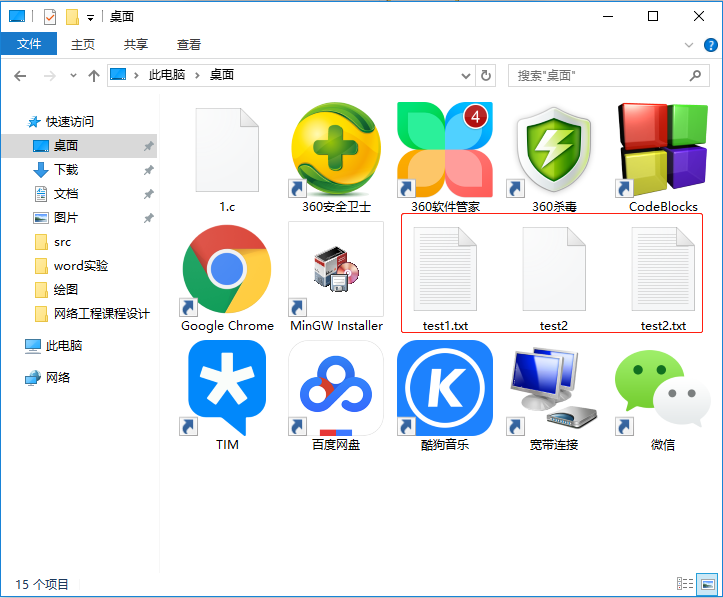


图27 下载文件成功展示

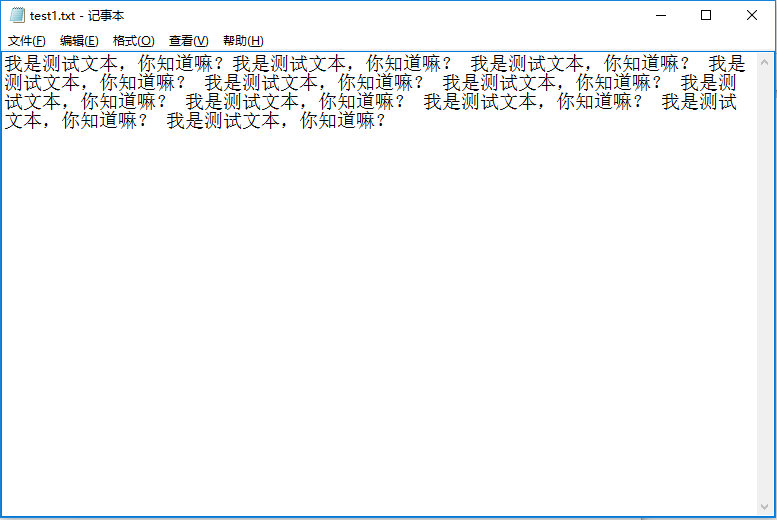


图28 上传测试文本

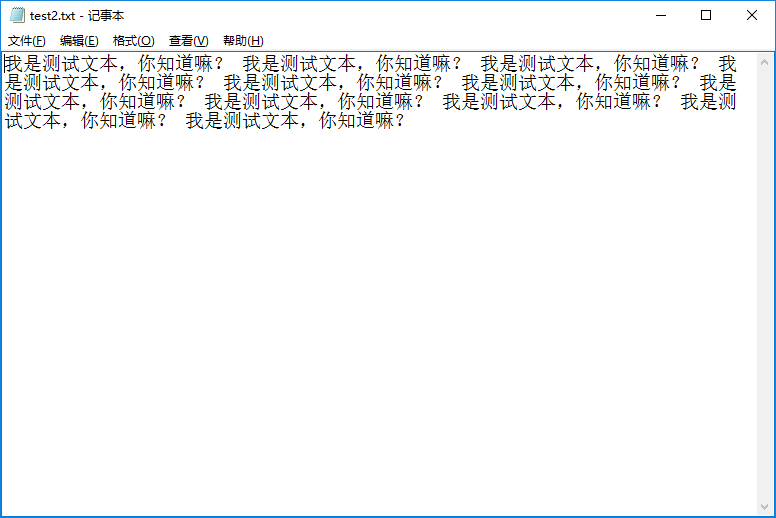


图29 下载测试文本

## 4.4 用户退出

点击1113用户客户端的退出客户端按钮，1113客户端退出，并从在线用户列表移除，其他用户会收到该用户下线的通知，并伴有其他用户下线提示音，如图30；点击1112用户的退出客户端按钮，1111用户会收到1112用户的下线通知，如图31；当关闭了所有的客户端之后，我们查看服务器端会发现，所有的用户操作信息及通讯信息均被服务器记录下来，如图32。

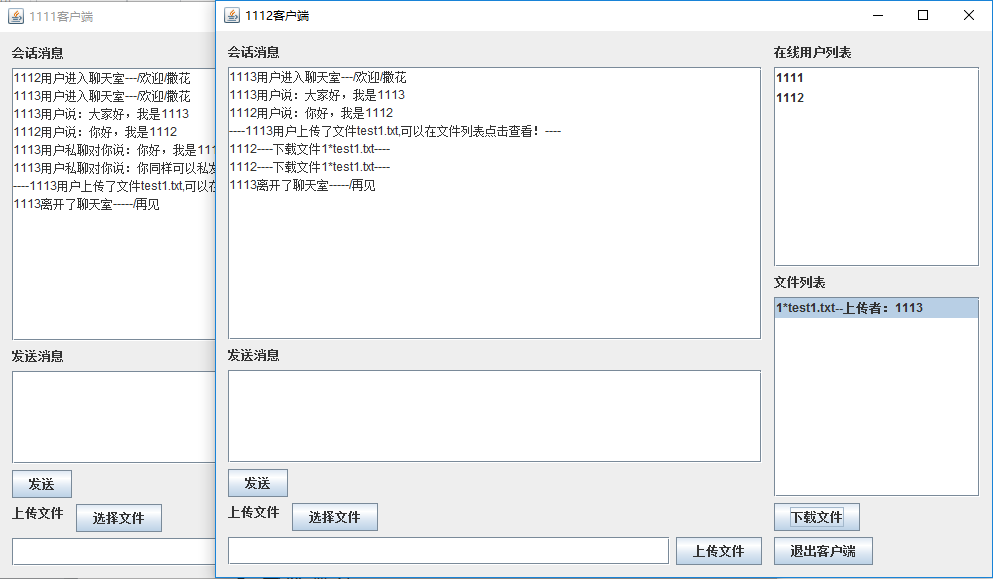


图30 其他用户下线通知



图31 多个用户下线

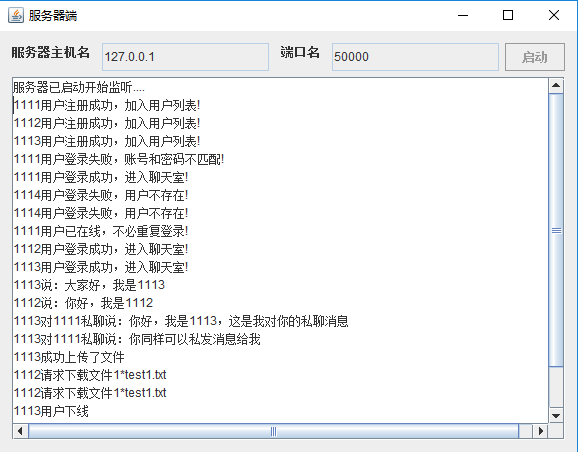


图32 服务器端显示

# 5 实验总结

**黎远波：**

我在本次课程设计中担任了小组组长的角色，我认为组长自然要承担更多的工作量并且学会合理分配任务，安排时间，进行系统设计。在本次课程设计中，每一部分的内容我都比较了解，能够做到统筹规划与设计。

本次课程设计实现的是基于UDP系统的QQ聊天室系统。本次设计依托于老师给的学习资料中的董相志老师的设计，在外在框架设计上给了我们一个先入的引导，使得我在本次设计中条理清晰，系统设计明确。

在看过董老师的基于UDP协议的QQ客户机与服务器项目设计之后，我立马就能够抓住里面的设计思路和设计路线，可以说好的设计就是这样条理清晰，系统化。但同时，我也发现了里面的几个问题：（1）打开用户登录界面，尝试多次不能登录系统，具体看过代码之后才能知道事先设定了三个用户，并且密码不设限，这不仅会使其他人进行测试时产生问题，同时这样的设计有点违背常理，我在我的系统中加以改进；（2）注册按钮功能未实现，在不知道登录账号密码时，很难登录系统进行测试；（3）此系统只能进行群发功能，而无法实现两个用户单独聊天私发消息的功能；（4）基于TCP协议的文件传输系统，我并没有看代码，但是在运行测试时，发现并不能实现文件的下载功能，只能上传显示进度。因此，我在我的系统中针对以上问题进行了改进：（1）增添注册功能，用户起初没有用户账号，注册即可，且账号由系统给定；（2）登录系统需要申请服务器进行账号和密码的匹配；（3）增加私聊功能，用户可以选择和另一用户进行私密聊天；（4）实现文件的上传及下载功能。

同时，本次课程设计中也遇到了许多问题，比如常见的空指针异常、未知源错误、报文传输错误、信息设置错误等，不过好在思路清晰，这些问题也一步步得到了解决。看老师的代码好像用到的有一些类库比较少见和高端，我没有尝试着去搞懂他的，只是大概知道每一步在干什么，然后根据自己的理解设计进行实现。

本系统同时也遗留了几个小问题：（1）在运行登录界面类时，点击注册按钮，系统会自动从1111开始分配账号，想要注册不同的账号需要连续注册，然后打开多个客户端登录即可，影响不大。（2）用户列表点击，应该是点击力度鼠标精度的问题，每次点击都会弹出两个对话框，也就是说点击了两次，关掉一个即可，可以说问题也不是很大。

这一次课程设计又一次学习了Socket网络编程，每次涉及Socket网络编程都会觉得比较有意思，也一次次的更加掌握了这种编程技术。这次课程设计仍然使我学习到了很多，使我重新掌握了Java编程，熟悉了Java编程语言和用法，同时也学习了别人的设计，受益匪浅。

**参考文献：**

[1]谢希仁. 计算机网络简明教程[M]. 电子工业出版社, 2007.

[2]李新良, 罗戈夕. 用Socket的编程机制实现网上交谈[J]. 电脑知识与技术, 2006(29):29-29.

[3]李勇. 进程间通信的分布式实现[D].吉林大学,2004.

[4]李新良, 罗戈夕. 用Socket的编程机制实现网上交谈[J]. 电脑知识与技术, 2006(29):29-29.

[5]江小燕. Java网络编程的研究[J]. 电脑编程技巧与维护, 2010(14):6-6.

[6]王静,曲凤娟.基于 Socket 的多用户并发通信的设计[J].福建电脑,2007(3):164.

[7]莫足琴,欧阳艳阶,马康.基于Java Socket多客户端并发通信聊天程序的设计与实现[J].电脑知识与技术,2008(20):253-255.