廣東工業大學

软件工程课程设计

学生学院 计算机学院

专业班级 19 级软件工程（1）班

学　　号 3119005028

学生姓名 魏耀辉

指导教师 欧毓毅

2021 年 06 月 18 日

摘 要

2021 年当今，微信、QQ、钉钉等社交聊天软件的使用人数愈发壮大，由此可见，社交类应用软件在当前移动互联网的应用市场占有非常重要的份额地位。但在科技高速发展的今天，很多老人对界面复杂的 QQ 和微信等软件感到难以上手，当老人或者儿童想要快速地给亲人朋友发信息时，总会被许许多多无关的弹窗影响使用体验，甚至是难以使用最基本的聊天功能。

基于此问题，GDUTYouChat 是一款基于局域网的简洁聊天软件，旨在为局域网当中的组员提供聊天服务，而避免了过多功能的互联网的聊天软件带来的烦恼。在本文所涉及的局域网聊天软件中，用户可以自定义用户名，设置 IP 地址和相同的服务器端口号来进行局域网下的多人聊天室功能，简单快捷。服务器管理也可以通过全体广播的方式来对每一个用户进行广播通讯。

关键词：局域网，通讯

目录

1 可行性研究.....5
1.1 技术可行性.....5
1.1.1 现有系统分析.....5
1.1.2 开发方案.....5
1.2 经济可行性.....5
1.3 社会可行性.....5
1.3.1 操作可行性.....5
1.3.2 法律可行性.....5
2 需求分析.....6
2.1 功能需求.....6
2.1.1 客户端模块.....6
2.1.2 服务端模块.....6
2.2 性能需求.....6
2.2.1 客户端性能需求.....6
2.2.2 服务端性能需求.....6
2.3 其他需求.....7
2.3.1 独立性需求.....7
2.3.2 界面需求.....7
2.3.3 先进性需求.....7
2.4 功能模型.....7
2.4.1 用例图.....7
2.4.2 用例描述.....8
2.5 静态模型.....9
3 概要设计.....9
3.1 系统体系架构设计.....9
3.2 模块设计.....9
4 详细设计.....10
4.1 系统层级结构.....10
4.2 客户端算法 IPO 表.....11
4.3 服务端算法 IPO 表.....12
5 系统实现.....12
5.1 聊天功能实现.....12
5.2 登录功能实现.....18
6 测试计划.....21
6.1 白盒测试.....21
6.1.1 登录功能测试.....21
6.1.2 聊天功能测试.....24
6.2 黑盒测试.....26
6.2.1 登录功能.....26
6.2.2 聊天功能.....27
7 总结与期望.....28
7.1 总结.....28
7.2 展望.....28

参考文献..28
致谢..29

1 可行性研究

1.1 技术可行性

1.1.1 现有系统分析

目前较为广泛的通讯软件有微信和 QQ，微信和 QQ 的基本功能有单对单通讯，群聊通讯，收发文件以及发送图片等，且发送和接受方均能通过服务器记录通讯记录在云端，可以方便快捷的下载并查看。微信的朋友圈功能和 QQ 的空间功能是社交通讯的延伸。

1.1.2 开发方案

初步计划使用 Java 语言进行开发操作，由于是简易的局域网聊天通讯，故暂时不使用数据库记录用户数据和聊天数据。使用客户端和服务端模式，使用并发的 Thread 类和网络编程 Socket 来进行开发。

1.2 经济可行性

由于是简易的局域网通讯软件开发，初步计划为仅聊天功能的实现，且是通过不同端口来进行不同聊天室的划分，功能实现较为简单，独立开发，所以经济成本较低。

1.3 社会可行性

1.3.1 操作可行性

本软件功能简单且易于上手，操作均有提示且逻辑合理，适配于低到高配置的 x86 架构的 PC 主机，符合用户使用习惯。

1.3.2 法律可行性

符合中华人民共和国现有法律法规。

2 需求分析

2.1 功能需求

2.1.1 客户端模块

在客户端模块中，提供给用户一些操作的功能，如表 1.1 所示

表 2.1 客户端模块功能需求

名称	详细描述
用户登录	用户输入用户名，IP 地址（可省略）和端口号（必须）后加入服务器
用户信息发送	用户可以发送信息给同一 IP 同一端口下的服务器中的其他用户

2.1.2 服务端模块

表 2.2 服务端模块功能需求

名称	详细描述
服务端启动	输入服务端端口号，启动服务器
服务端广播信息	服务端可以全体广播在同一 IP 同一端口下的服务器的其他用户

2.2 性能需求

2.2.1 客户端性能需求

局域网聊天室软件是基于局域网中的用户使用的一款软件，局域网中的用户量不会过大，至少需要保证在 10 人局域网下的通讯畅通和收发信息的及时性。考虑到不少老人的电脑是 Windows XP 系统或以下，要做好向下兼容的准备

2.2.2 服务端性能需求

局域网通讯软件必须要确保通讯双方收发信息的及时性，时延不能过大，否则会给通信双方带来不可靠的通信，同时要做好局域网内多人通信的并发情况，服务端接受客户端的请求处理时间要在 1s 之内。

2.3 其他需求

2.3.1 独立性需求

不同聊天室的划分是通过不同的端口号来实现的，每个端口号的设置都是对应一个服务端，而同一用户名的用户可以通过连接不同的端口号来进入不同的聊天室来进行通讯且互不干扰。

2.3.2 界面需求

本次的局域网聊天室通讯软件为 x86 架构下的 PC 应用软件，界面设计需要做到简洁大方、方便操作。且功能按钮明确有提示。

2.3.3 先进性需求

需要采用先进、成熟的计算机软件开发技术进行开发，保障所开发的系统能够最大限度的适应今后技术的发展。软件结构应实现模块化开发，并统一开发规范，保证软件质量。

2.4 功能模型

2.4.1 用例图

本系统主要分为登录用例和发送信息用例。如图 2.1 所示

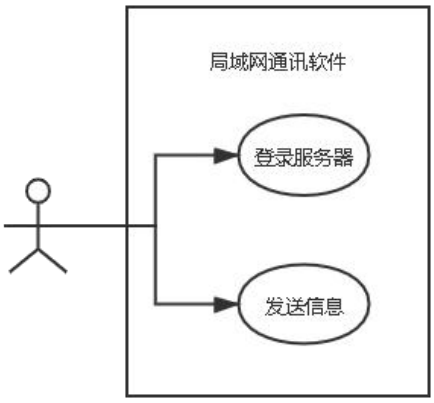

图 2.1 用例图

2.4.2 用例描述

表 2.3-表 2.4 为本系统各用例的详细描述，每个用例描述表中都详细描述了该用例的编号、名称、用例简要描述、参与者、前置条件、主事件流以及备选事件流。

表 2.3 用户登录服务器用例

名称	内容
用例编号	uc-01
用例名	登录服务器
用例描述	用户输入信息后登录到服务器
参与者	用户
前置条件	用户输入正确的端口号
主事件流	1. 输入用户名（可选） 2. 输入 IP 地址（可选） 3. 输入端口号 4. 登录成功
备选事件流	1. 服务器关闭，无法登录 2. 端口号错误

表 2.4 发送信息用例

名称	内容
用例编号	uc-02
用例名	发送信息
用例描述	用户输入文本内容后发送
参与者	用户
前置条件	用户输入信息不为空
主事件流	1. 输入信息 2. 发送信息
备选事件流	1.服务器关闭，无法发送

2.内容为空

2.5 静态模型

本系统的领域类图由 Server 和 User 两个数据类构成，如图 2.2 所示。

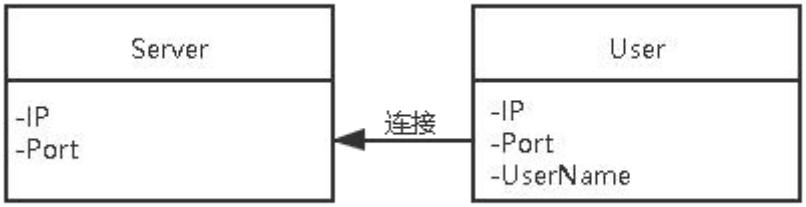

图 2.2 领域类图

3 概要设计

3.1 系统体系架构设计

3.1.1 系统构件图

本次设计是基于 C/S 模型的局域网聊天软件，由服务端 Server 和客户端 Client 构成，多个客户端 Client 连接到一个服务端 Server 进行通讯。

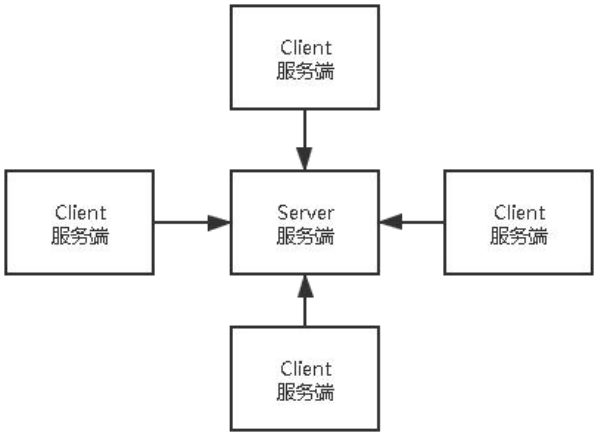

图 3.1 系统构件图

3.2 模块设计

GDUTYouChat 分为用户模块、信息模块一共两个模块。其中用户模块负责用户的登录，而信息模块负责用户之间的发送信息和接收信息，如图 3.2 所示。

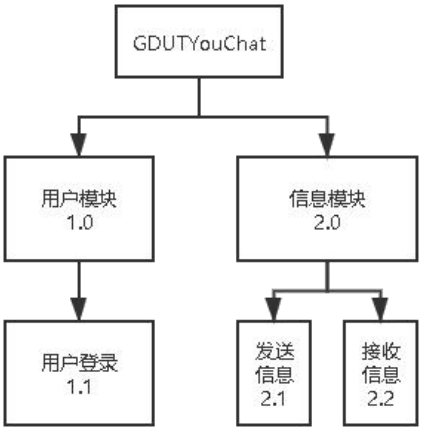

图 3.2 系统模块图

4 详细设计

4.1 系统层级结构

GDUTYouChat 客户端分为用户模块和信息模块共两个模块，用户模块负责用户登录到服务端。而信息模块负责让用户发送信息和接收信息。如图 4.1 所示。

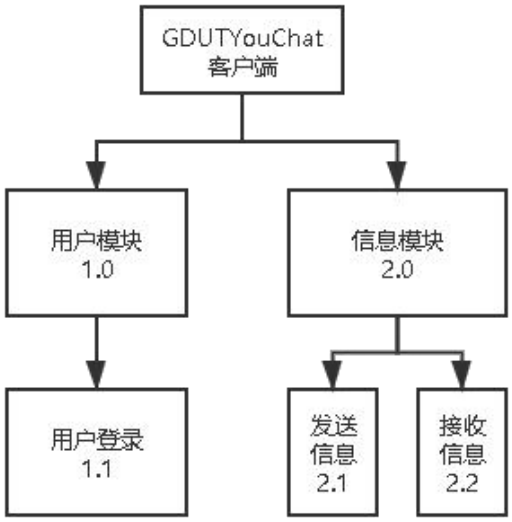

图 4.1 GDUTYouChat 客户端系统结构图

GDUTYouChat 服务端分为启动模块和信息模块共两个模块，其中启动模块负责设置端口号并启动服务器。而信息模块负责让服务器广播信息和接收用户发送的信息。如图 4.2 所示

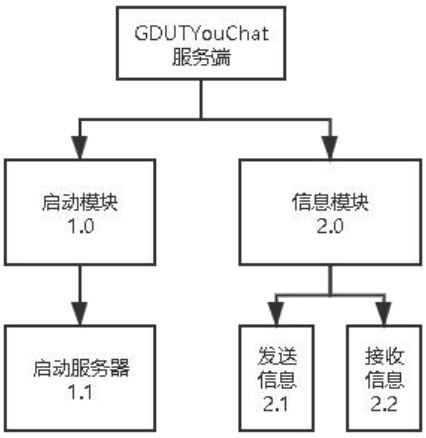

图 4.2 GDUTYouChat 服务端系统结构图

4.2 客户端算法 IPO 表

表 4.1-4.2 展示了客户端各个功能模块的算法 IPO 表，介绍了各功能模块的编号、模块、日期、作者、被调用模块、输入输出、数据处理与相关数据。

表 4.1 客户端用户登录功能 IPO 表

IPO 表			
系统：	GDUTYouChat		
编号：	1.1	日期：	2021/06/18
模块：	用户登录功能	作者：	魏耀辉
被调用：	1.0 用户模块		
输入：	用户输入信息，点击登录	输出：	登录成功提示信息
处理：	用户点击登录后，系统发出欢迎用户的提示信息		
数据：	用户身份信息，验证端口号		

表 4.2 客户端信息发送功能 IPO 表

IPO 表			
系统：	GDUTYouChat		
编号：	2.1	日期：	2021/06/18
模块：	信息发送	作者：	魏耀辉
被调用：	2.0 信息模块		
输入：	用户输入信息，点击发送	输出：	聊天室中显示信息
处理：	用户输入文本信息后，点击发送，文本内容显示在聊天室共享区中		

数据：	用户身份信息，验证端口号

4.3 服务端算法 IPO 表

表 4.3-4.4 展示了客户端各个功能模块的算法 IPO 表，介绍了各功能模块的编号、模块、日期、作者、被调用模块、输入输出、数据处理与相关数据。

表 4.3 服务端启动服务器功能 IPO 表

IPO 表			
系统：	GDUTYouChat		
编号：	1.1	日期：	2021/06/18
模块：	用户登录功能	作者：	魏耀辉
被调用：	1.0 用户模块		
输入：	输入端口号，点启动服务器	输出：	启动成功提示信息
处理：	启动服务器后显示启动成功提示，并等待客户端连接		
数据：	端口号		

表 4.4 客户端信息发送功能 IPO 表

IPO 表			
系统：	GDUTYouChat		
编号：	1.2	日期：	2021/06/18
模块：	信息发送	作者：	魏耀辉
被调用：	2.0 信息模块		
输入：	输入信息，点击全体发送	输出：	聊天室中显示信息
处理：	用户输入文本信息后，点击发送，文本内容显示在聊天室共享区中		
数据：	端口号		

5 系统实现

5.1 聊天功能实现

聊天功能的实现逻辑是：已登录的用户才能发送信息，且服务端已启动才会发送信息，当用户名不为空时显示用户名，而用户名为空会显示为匿名用户。随后用户输入文本信息，若文本不为空，则点击发送按钮后可以发送，否则不发送。整体流程图如图 5.1 所示

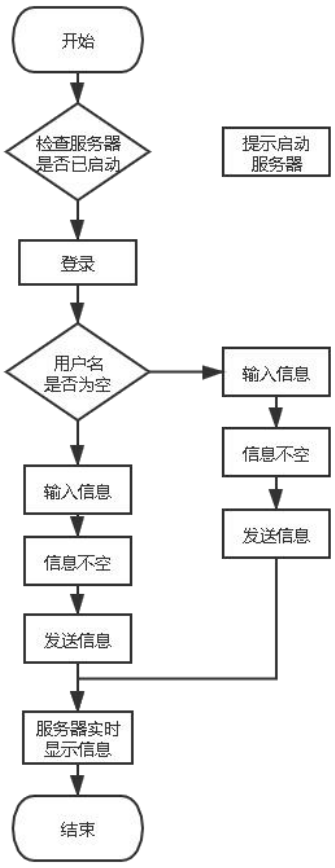

图 5.1 聊天功能流程图

5.1.1 界面设计

客户端界面由用户名输入栏，IP 地址输入栏，端口号输入栏在顶部，文本输入栏在底部构成，可操作按钮为登录至服务器和发送信息。服务端界面由端口号输入栏在顶部，文本输入栏在底部构成，可操作按钮为启动服务器和发送全体信息。界面详细设计，如图 5.2-5.4 所示

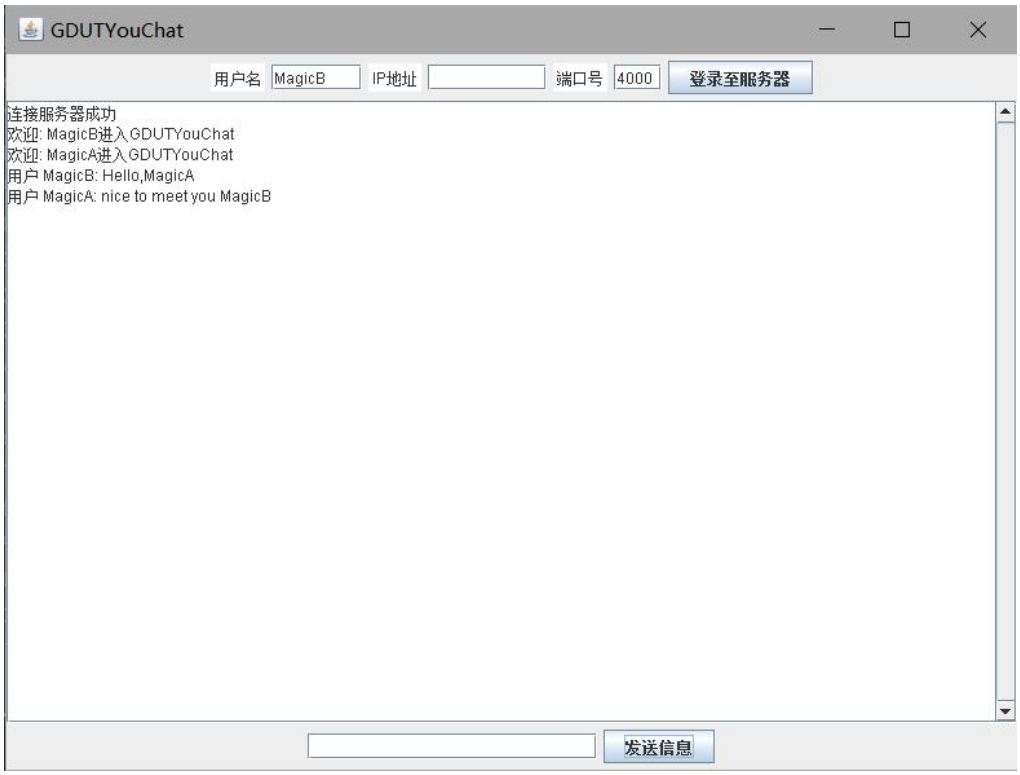

图 5.2 客户端 1 界面

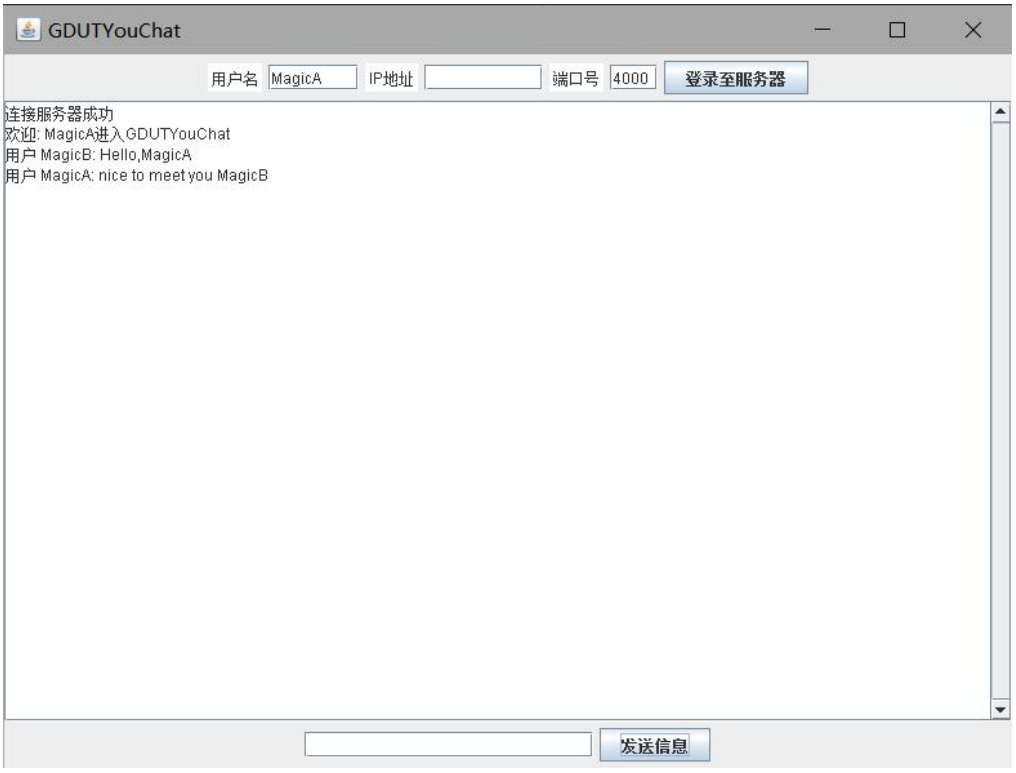

图 5.2 客户端 2 界面

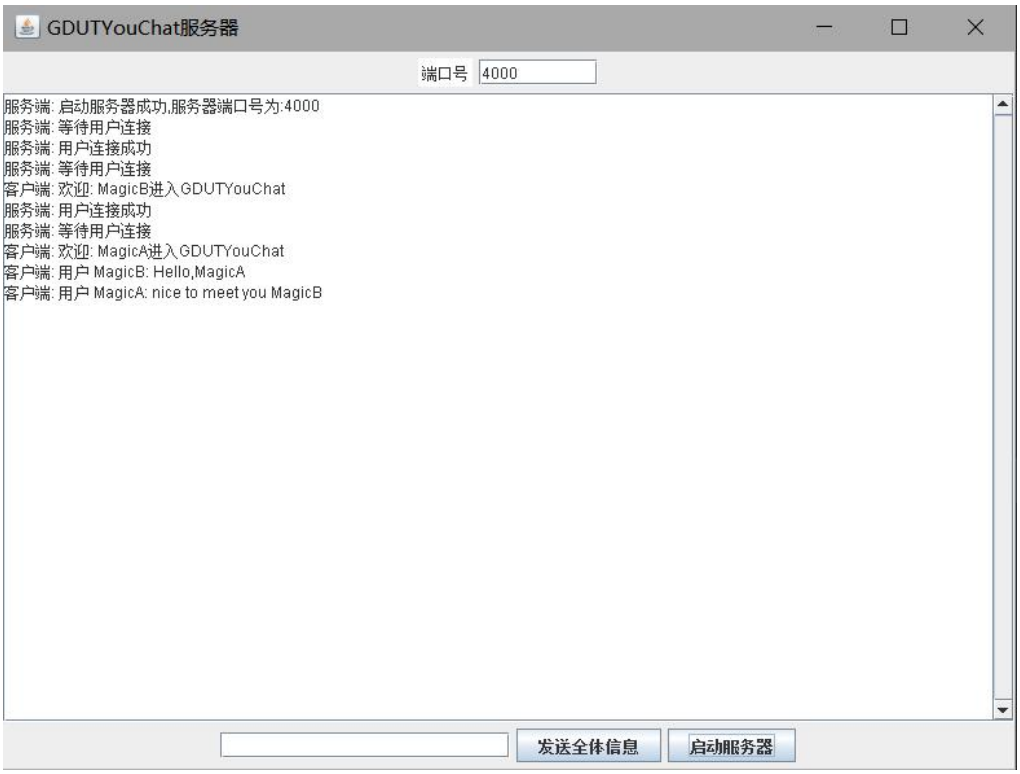

图 5.4　服务端界面

5.1.2 聊天功能实现代码

客户端：

```
public void run() {
        String message = "";
        while (true) {
            try {
                message = bufferedReader.readLine();
            }
            catch (IOException e) {//检测服务器关闭
                println("服务器断开连接,请联系服务器管理");
                break;
            }
            if (message != null && message.trim() != "") {
                println(message);
            }
        }
```

```
    }
    public void sendMessage(String msg) {//发送信息方法
        try {
            printWriter.println(msg);
        }
        catch (Exception e) {
            println(e.toString());
        }
    }
    public void println(String s) {//打印发送信息
        if (s != null) {
            this.UI.textAreaShow.setText(this.UI.textAreaShow.getText()  +  s  +
"\n");
            System.out.println(s + "\n");
        }
    }
```

服务端：

```
public synchronized void sendMsg(String msg) {
        try {
            for (int i = 0; i < UI.clients.size(); i++) {
                Socket client = UI.clients.get(i);
                printWriter = new PrintWriter(client.getOutputStream(), true);
                printWriter.println(msg);
            }
        }
        catch (Exception e) {
            println(e.toString());
        }
    }
```

```
    public void println(String s) {
        if (s != null) {
            s = "服务端: " + s;
            this.UI.textAreaShow.setText(this.UI.textAreaShow.getText()  +  s  +
"\n");
            System.out.println(s + "\n");
        }
    }
```

监听类：

```
public ConnectClient(ServerUI UI, Socket client) {
        this.UI = UI;
        this.client=client;
        this.start();
    }
    @Override
    public void run() {//为每一个客户端创建线程等待接收信息，然后把信息广播出去
        String messsage = "";
        while (true) {
            try {
                bufferedReader          =          new          BufferedReader(new
InputStreamReader(client.getInputStream()));
                printWriter = new PrintWriter(client.getOutputStream(), true);
                messsage = bufferedReader.readLine();
                sendMessage(messsage);
            } catch (IOException e) {
                println(e.toString());
                break;
            }
            if (messsage != null && messsage.trim() != "") {
```

```
                println("客户端: " + messsage);
            }
        }
    }
    public synchronized void sendMessage(String msg) {//把信息广播到所有用户
        try {
            for (int i = 0; i < UI.clients.size(); i++) {
                Socket client = UI.clients.get(i);
                printWriter = new PrintWriter(client.getOutputStream(), true);
                printWriter.println(msg);
            }

        } catch (Exception e) {
            println(e.toString());
        }
    }
    public void println(String s) {
        if (s != null) {
            this.UI.textAreaShow.setText(this.UI.textAreaShow.getText() + s +
"\n");
            System.out.println(s + "\n");
        }
    }
```

5.2 登录功能实现

5.2.1 登录功能实现代码

客户端：

```
public Client(ClientUI UI) {
        this.UI = UI;
        try {
            String ip = UI.textFieldIP.getText(); //获取输入的 Ip 地址
            int port = Integer.parseInt(UI.textFieldPort.getText()); //获取输入的端口号
            client = new Socket(ip, port);//设置客户端连接服务端的 Ip 和端口
            println("连接服务器成功");
            bufferedReader = new BufferedReader(new InputStreamReader(client.getInputStream()));
            printWriter = new PrintWriter(client.getOutputStream(), true);
            String name = UI.textFieldName.getText();//用户名信息
            if (name == null || "".equals(name)) {//若用户名为空，则标记为匿名用户
                name = "匿名用户";
            }
            sendMessage("欢迎: " + name + "进入 GDUTYouChat");
        }
        catch (NumberFormatException nu) {
            println("连接服务器失败，请输入正确的端口号");
            nu.printStackTrace();
        }
        catch (IOException e) {
            println("连接服务器失败，请输入正确的 IP 地址与端口号格式");
            println(e.toString());
            e.printStackTrace();
        }
        this.start();//启动客户端
    }
```

服务端：

```
public Server(ServerUI UI) {
        this.UI = UI;
        this.start();
        Port=Integer.parseInt(UI.textFieldPort.getText());
    }
    @Override
    public void run() {
        try {
            serverSocket = new ServerSocket(Port);
            UI.clients=new ArrayList<>();
            println("启动服务器成功,服务器端口号为:"+Port);
            while (true) {
                println("等待用户连接");
                Socket client = serverSocket.accept();
                UI.clients.add(client);
                println("用户连接成功");
                new ConnectClient(UI, client);
            }
        }
        catch (IOException e) {
            println("启动服务器失败");
            println(e.toString());
            e.printStackTrace();
        }
    }
```

6 测试计划

6.1 白盒测试

6.1.1 登录功能测试

测试详细如图 6.1-6.4 所示。

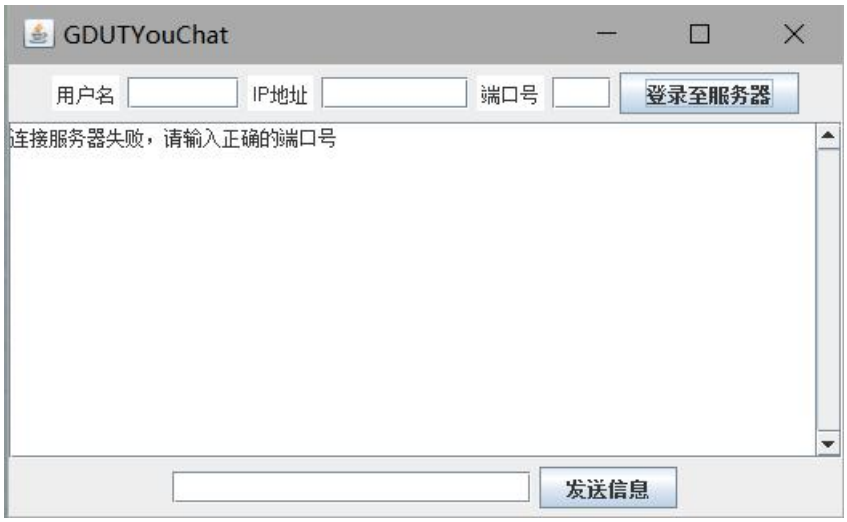

图 6.1 用户输入信息全空测试

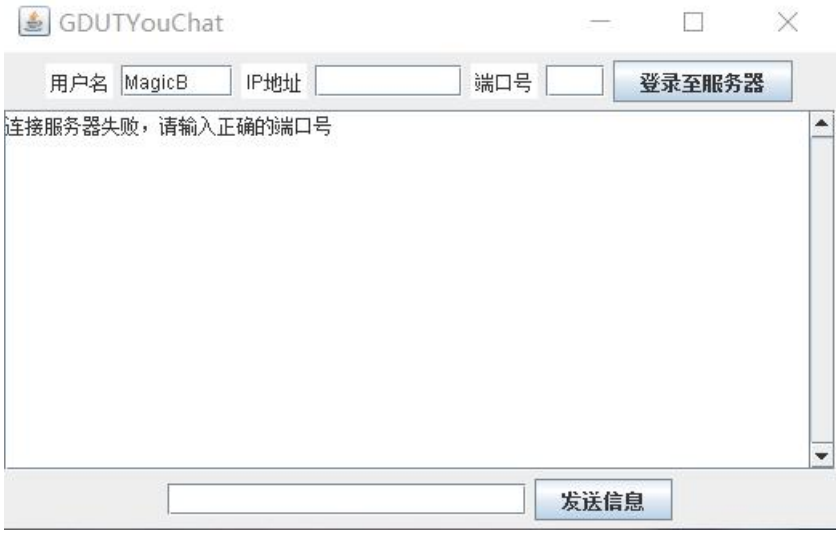

图 6.2 端口号输入信息为空测试

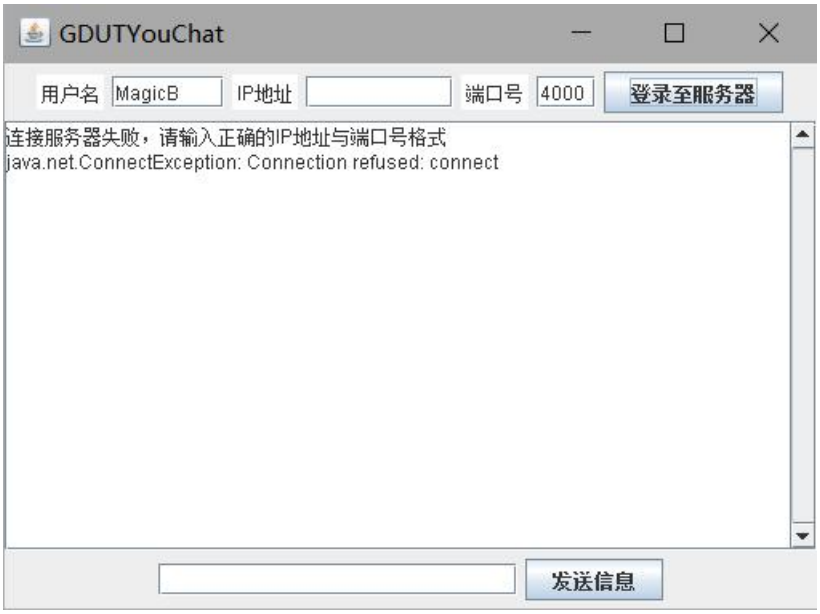

图 6.3 端口号未开放测试

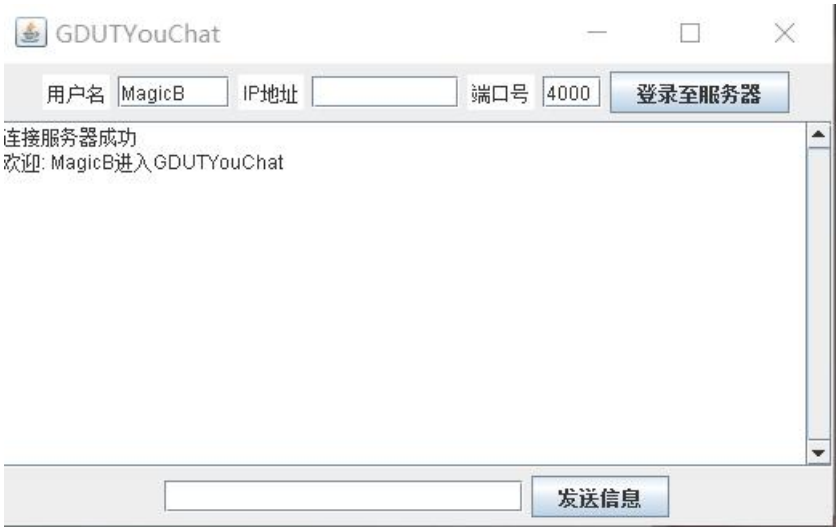

图 6.4 IP 地址为空测试

图 6.4 信息全写测试

6.1.2 聊天功能测试

测试详细如图 6.5-6.6 所示。

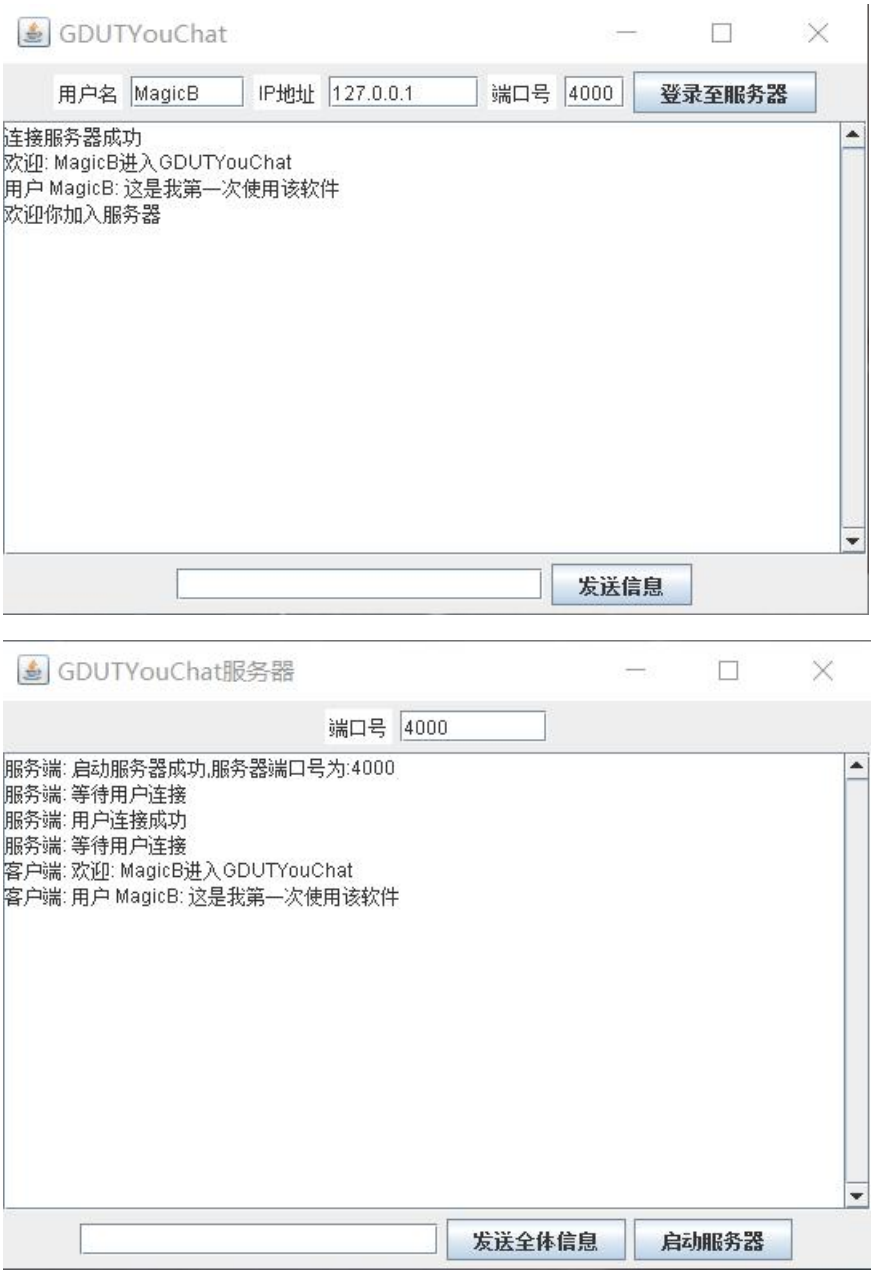

图 6.5 单人聊天与服务器广播测试

图 6.6 多人聊天与服务器广播测试

6.2 黑盒测试

6.2.1 登录功能

测试详细如图 6.7 所示。

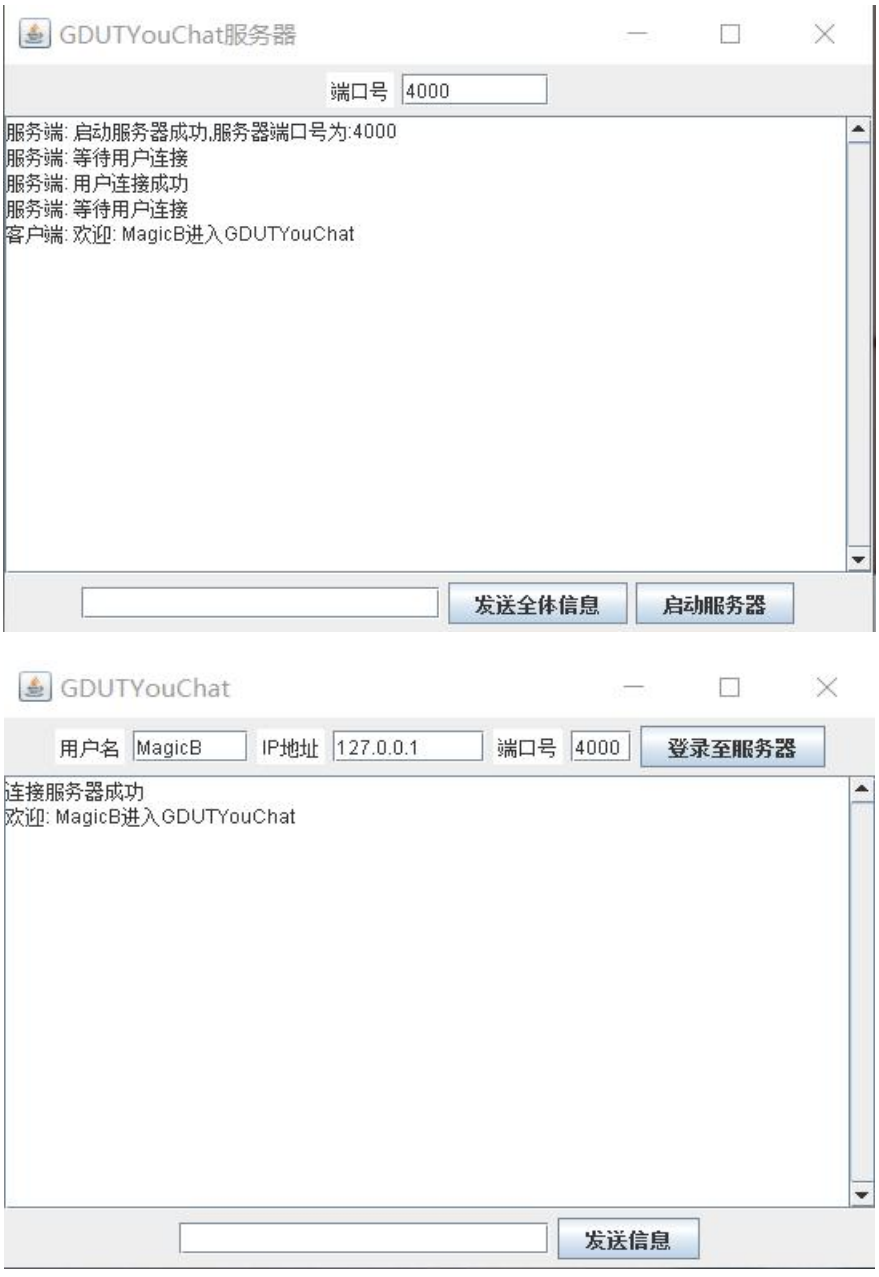

图 6.7 登录功能黑盒测试

6.2.2 聊天功能

测试详细如图 6.8 所示。

图 6.8 聊天功能黑盒测试

7 总结与期望

7.1 总结

在信息高速发展的今天，通讯软件越来越受到人们的依赖，各大互联网公司纷纷开发了自家的通讯软件，诸如腾讯的微信、QQ 和阿里的钉钉等。这些软件解决了人们日常交流通讯、语音通话、以及分享生活甚至是打卡的功能，但面对繁多复杂的界面，一些老人容易在复杂的操作中望而却步。为此，本文设计并实现了一款仅聊天用的局域网通讯软件，满足大部分人的聊天通讯需求。

本课题主要完成工作如下。

1. 分析了局域网通讯软件的需求，旨在为用户提供最简便的操作来获得聊天体验。

2. 设计了图形化界面，操作简便、易于上手。

7.2 展望

经过本文的研究与设计，设计并实现了 GDUTYouChat 的局域网通讯软件，但在功能以及界面等方面仍有改善和优化的空间。

1. 在软件功能方面，还未能实现图片传输和语音发送的功能，无法最大程度达到通讯的要求

2. 界面过于扁平化处理，希望后期能优化成拟物化，让界面显得生动形象，不会过于呆板

参考文献

[1] 张海藩，牟永敏. 软件工程导论（第 6 版）[M] . 北京：清华大学出版社, 2013.

[2] 赵锐，李卫华. Java 技术及应用（第 2 版）[M]. 北京：清华大学出版社，2017.

致谢