|  |  |  |
| --- | --- | --- |
| 计算机网络课程设计 | | |
|  | 题目: | 基于TCP的QQ消息与文件传输 |
|  | 姓名: | 陆家诚、陆润风 |
|  | 学院: | 人工智能学院 |
|  | 专业: | 计算机科学与技术系 |
|  | 班级: | 网工191 |
|  | 学号: | 19319118、19319117 |
|  | 指导教师: | 叶锡君 |

2022年4月17日

目录

[1. 选题意义 1](#_Toc103166388)

[2. 选题目标 1](#_Toc103166389)

[3. 课程设计内容 1](#_Toc103166390)

[3.1 需求分析和系统结构 1](#_Toc103166391)

[3.2 关键技术介绍 1](#_Toc103166392)

[3.3 详细设计和实现 2](#_Toc103166393)

[3.3.1 界面设计 2](#_Toc103166394)

[3.3.2 聊天消息类型数据结构设计 2](#_Toc103166395)

[3.3.3 端与端通信消息类型数据结构设计 3](#_Toc103166396)

[3.3.4 Socket通信线程设计 3](#_Toc103166397)

[3.3.5 消息接收与处理方法设计 3](#_Toc103166398)

[3.4 遇到并解决的主要问题 4](#_Toc103166399)

[4. 功能模块和程序测试 4](#_Toc103166400)

[4.1 注册 4](#_Toc103166401)

[4.2 登录 4](#_Toc103166402)

[4.3 找回密码 4](#_Toc103166403)

[4.4 消息传输 4](#_Toc103166404)

[4.5 文件传输 4](#_Toc103166405)

[4.6 其他用户的登录登出 4](#_Toc103166406)

[4.7 登出 4](#_Toc103166407)

[5. 分工说明和课程设计心得与体会 5](#_Toc103166408)

[5.1 小组分工和任务完成说明 5](#_Toc103166409)

[5.2 心得体会 5](#_Toc103166410)

[参考文献 5](#_Toc103166411)

基于TCP的QQ消息与文件传输

1. 选题意义

我组选题为“基于TCP的QQ消息与文件传输”，服务器与客户端之间的通信通过TCP协议实现。程序能够实现用户之间的简单的的聊天通信和文件传输功能。

1. 选题目标

（1）实现客户端与服务器端之间的通信；

（2）实现多用户之间的实时消息传输和文件传输；

（3）实现服务器端与数据库之间的通信；

1. 课程设计内容

3.1 需求分析和系统结构

（1）客户端需要与服务器端联动实现用户的注册、登录、找回密码、信息传输和文件传输，以及登出的功能。

注册：客户端提交注册请求给服务器，服务器查询数据库，若没有重名用户，则向数据库添加一条记录，发送反馈消息给客户端。

登录：客户端提出登录请求，服务器比对数据库中的信息后，向客户端发送成功或者失败的反馈。如果成功，客户端为用户开启聊天界面，并获取服务器发送的“当前在线用户列表”。

找回密码：客户端向服务器发送找回密码请求以及自己设置的新密码，服务器比对数据库中的信息，如果匹配，则将数据库中的密码修改为新的密码，返回密码修改成功的消息给客户端，若失败，返回失败的消息给客户端。

信息传输：信息传输分为两种——私聊和广播。以广播形式发送的消息，服务器接收到之后，会向所有在线的用户的客户端发送该消息；以私聊形式发送的消息，服务器仅会向选定的接受者发送消息。客户端接收到消息后，输出在消息框中。

文件传输：文件传输分为两种——私聊和广播。以广播形式发送的文件，服务器接收到之后，会向所有在线的用户的客户端发送该文件；以私聊形式发送的文件，服务器仅会向选定的接受者发送文件。客户端接收到文件后，会弹出窗口用于接收文件。

登出：用户关闭客户时，会向服务器发出登出请求，服务器收到请求后，向所有在线的客户端发出用户登出的消息，客户端接收到之后，更新在线用户列表。

其他用户的登录和登出：其他用户登录和登出时，会向服务器发送相应消息，服务器收到相应消息后，更新在线用户列表，并向所有客户端发送相应消息，客户端接收到消息后，更新在线用户列表。

（2）服务器用于接收、处理服务器发送的消息和发送给服务器消息。在服务器端，会显示程序日志信息。

（3）服务器与客户端分别统一接收消息，并对消息进行解析，随后根据不同的消息类型做出相应的动作，并在服务器端界面上显示相应的日志信息。

3.2 关键技术介绍

3.2.1 Java AWT和Swing图形界面编程

3.2.2 JDBC数据库通信技术

3.3.3 Socket嵌套字编程

3.3 详细设计和实现

3.3.1 界面设计

图形界面使用java.awt和javax.swing架构进行编写，界面的布局代码相对复杂。

客户端共分为4个界面：登录（Login）、注册（Register）、找回密码（Findpasswd）和聊天（Chat）。其中登录界面主要处理登录认证事件，注册界面处理注册认证事件，找回密码主要处理找回密码认证事件，聊天界面处理消息和文件传输事件以及实时刷新在线用户列表。

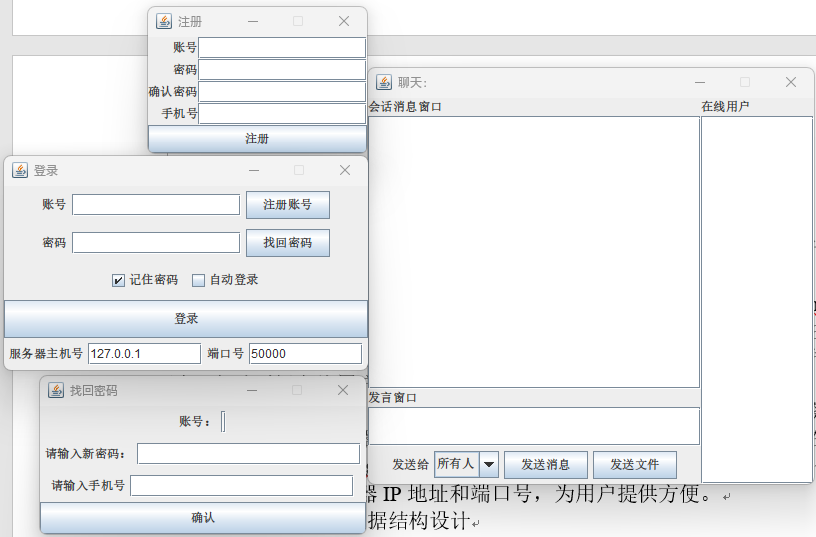


图3-1 客户端各图形界面

服务器端

3.3.2 聊天消息类型数据结构设计

（1）字符串消息的类型定义如下：

public String message;// 消息字符串

public LocalDateTime time;// 时间

public String sender;// 发送方

public int type;// 0：广播，1：私聊

public String receiver;// 接收方

（2）文件消息的类型定义如下：

private String Filename;// 文件名

private byte str[] = new byte[8096];// 部分文件内容

private boolean EndOfFile;// 文件开始标志

private boolean StartOfFile;// 文件结束标志

private LocalDateTime time;// 时间

private String sender;// 发送方

private int type;// 0：广播，1：私聊

private String receiver;// 接收方

3.3.3 端与端通信消息类型数据结构设计

public int type;// 消息类型

public String Username;// 用户账号

public String Passwd;// 密码

public String Phone;// 手机号

public ArrayList<String> nowUsers;// 当前在线用户列表

public boolean result;// 数据库操作结果

public com.Message message;// 消息

public com.FileMessage filemessage;// 文件

public LocalDateTime time;// 时间

public String receiverip;// 接收方ip

public int receiverport;// 接收方端口号

3.3.4 Socket通信设计

在客户端，拥有一个用于端与端之间通信的线程ClientThread。线程开启后，会实时监听由服务器端发送而来的消息，并进行处理，处理方法详见3.3.5节。

在客户端的类ClientThread中，有一个用于向服务器端发送消息的方法SendToServer(Message message)，这个方法会被调用，用于包装消息，并向服务器端发送。本课程设计中，客户端与服务器端之间的通信通过对象流（ObjectInputStream、ObjectOutputStream）实现。每次需要发送消息时，就会将需要发送的消息包装成类util.Message，写入到对象流中，并刷新流。

登录界面启动时，会为该客户端创建线程ClientThread，开启与服务器的连接，开始监听由服务器发送而来的消息或者向服务器发送消息。线程会在用户退出系统时关闭。同时，在项目根目录存有文件“Settings.json”，保存登录界面的相关配置，包括是否记住密码、界面下方配置等服务器IP地址和端口号，为用户提供方便。

在服务器端，

3.3.5 消息接收与处理方法设计

客户端与服务器端的通信线程开启后，实时监听由服务器发来的消息。每当接收到消息时，由线程将对象流中的消息提取后，根据消息类型进行对应的处理。

本设计中，消息类型一共分为11类，用类成员type区别。这11类消息的描述如下：

|  |  |  |  |
| --- | --- | --- | --- |
| type | 对应事件 | 流向 | 需要传输的消息 |
| 01 | 注册 | 客户端→服务器 | 用户名、密码、手机号 |
| 02 | 服务器→客户端 | 结果 |
| 11 | 登录 | 客户端→服务器 | 用户名、密码 |
| 12 | 服务器→客户端 | 结果 |
| 13 | 退出 | 客户端→服务器 | 用户名 |
| 14 | 其他用户登录 | 服务器→客户端 | 用户名 |
| 15 | 其他用户退出 | 服务器→客户端 | 用户名 |
| 31 | 找回密码 | 客户端→服务器 | 用户名、新密码、手机号 |
| 32 | 服务器→客户端 | 结果 |
| 41 | 文字消息 | 服务器→客户端 | 用户名、接收方、文字消息 |
| 42 | 文件消息 | 服务器→客户端 | 用户名、接收方、文件消息 |

消息解析后，根据其类型，调用不同的方法进行相应的处理。

3.4 遇到并解决的主要问题

1. 功能模块和程序测试

4.1 注册

用户点击登录界面上的“注册”按钮，即可进入注册界面。再此界面中，用户需要在文本框中输入自己的账号、密码和自己的手机号。其中，自己的手机号可以用于找回密码。点击“注册”按钮，客户端向服务器发送注册请求，服务器先查询数据库，查找是否有同名账号，如果没有，则会向数据库写入一条新的用户信息，随后发送注册成功的反馈信息给客户端。否则，返回注册失败的信息给客户端。

4.2 登录

在登录窗口输入账号和密码，点击下方的“登录”按钮，客户端向服务器发出请求，服务器接收到消息后，通过数据库查找账号与密码是否匹配，若匹配成功，则会发送反馈消息给客户端，客户端接收到消息后，则成功登录，为账号创建一个聊天窗口，并读取反馈消息中的当前在线用户列表，显示在窗口上。如若不匹配，服务器则会向客户端发送登陆失败的反馈消息。

4.3 找回密码

在登录界面点击“找回密码”，即可跳转到找回密码界面。在界面输入账号、手机号和新密码，点击“找回密码”，客户端向服务器发出找回密码请求。服务器接收到请求之后，查询数据库，对提供的用户名和手机号进行比对，若成功，则会把新密码写入数据库，并发送修改密码成功的消息给客户端；若失败，则发送失败的反馈消息。

4.4 消息传输

消息传输分为两种模式：广播和私聊。在界面下方下拉框可以选择“发送给”的对象。客户端发出消息后，由服务器接收，并根据其类型进行处理。若为广播消息，则发送给所有客户端；如果是私聊消息，则发送给特定的客户端。客户端接收到消息后，显示在聊天窗口中。

4.5 文件传输

文件传输分为两种模式：广播和私聊。在界面下方下拉框可以选择“发送给”的对象。客户端发出文件后，由服务器接收，并根据其类型进行处理。若为广播文件，则发送给所有客户端；如果是私聊文件，则发送给特定的客户端。客户端接收到文件后，弹出接收文件对话框，选择需要保存文件的路径，将文件保存在相应路径下。

4.6 其他用户的登录登出

当有其他用户登录或者登出时，其客户端会向服务器发送消息，服务器接收到该消息并进行处理之后，更新在线用户列表，并会向所有客户端发送“用户登录或登出”的消息。客户端接收到这条消息后，会更新在线用户列表，并刷新界面显示。

4.7 登出

用户关闭聊天窗口时，会向服务器发出登出请求信息，服务器接收信息后，会向所有的客户端发送用户登出的消息。

1. 分工说明和课程设计心得与体会

5.1 小组分工和任务完成说明

5.2 心得体会

参考文献