同济大学

《Java 开发技术与应用》

实验报告

C/S 聊天室

专　　业　软件工程

年　　级　2017 级

学　　号

姓　　名　王亮

2018 年 6 月

一．系统概述

1.1 系统简介

这是一款使用 Java 语言开发的聊天室软件，该系统由客户端（Client）与服务端（Server）、数据库（MySQL）组成。通过客户端、网络端、数据库之间的通信，可以实现在客户端进行用户注册、登录、显示在线用户、清空聊天记录、群聊、私聊、发送文件等操作，在服务端进行显示在线用户、强制用户下线等操作。

1.2 开发与运行环境

1.2.1 Java 开发环境：

Windows 10

jdk1.8.0_171

eclipse-java-mars-2-win32-x86_64

1.2.2 数据库开发环境

MySQL Server 8.0

1.2.3 数据库的使用

1.2.3.1 安装 MySQL 数据库

1.2.3.2 数据库配置

root 用户密码设置为”password”（若不一致，应在 ChatServer/src/server/backstage/DatabaseManager.java 第 27 行对应位置进行修改），并在 MySQL 命令行中输入 set global time_zone='+8:00';（最新版本的用于连接 mysql 数据库的 jar 包中要求数据库必须设置时区，否则报错）。其他设置保持 mysql 默认设置即可。

```
mysql> set global time_zone='+8:00';
Query OK, 0 rows affected (0.00 sec)
```

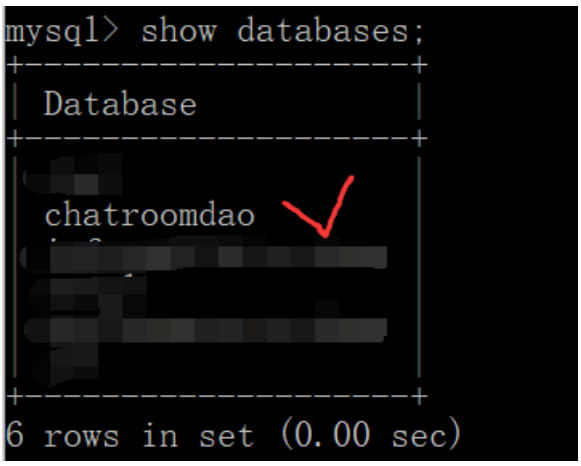
1.2.3.3 数据库的导入

在 MySQL 命令行中输入’source (项目中 ChatRoomDao.sql 文件的路径名);’导入数据库。

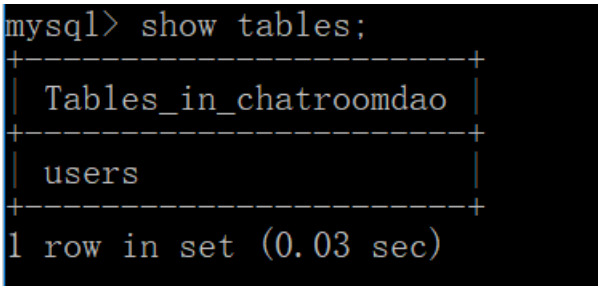
```
mysql> source F:\Java\ChattingRoom\ChatServer\database\ChatRoomDao.sql
Query OK, 1 row affected (0.09 sec)

ERROR 1046 (3D000): No database selected
Database changed
Query OK, 0 rows affected, 2 warnings (0.06 sec)
```

输入’show databases;’查看数据库 chatroomdao 是否成功导入。



输入'show tables;'查看数据表 users 是否成功导入。



1.2.3.4 在 Eclipse 中导入工具包 mysql-connector-java-8.0.11.jar

二. 系统设计和实现

2.1 系统功能

2.1.1 客户端

注册、登录、群聊、单人聊天、群发文件、向单人发送文件、显示在线用户、清空聊天记录

2.1.2 服务端

开启/关闭服务器、显示在线用户、发送系统消息、强制用户下线、维护数据库（增加、查询）

2.1.3 数据库

存放用户信息（用户名、密码、当前是否在线）

2.2 类与接口的设计

2.2.1 Client:

2.2.1.1 common 文件夹：客户端和服务端通用的类。

Message：通过网络传送的消息类，包含消息类型、消息内容、发送时间、发送方、接收方等成员变量，构造函数，成员变量的访问器和构造器函数。

MessageCode：消息类型接口。

User：用户类。包含用户名、密码、用户类型及其访问器和构造器函数。

2.2.1.2 client.backstage 文件夹：客户端后台运行的类。

ConnectServer：实现 Runnable 接口，客户端运行过程中与服务端保持通信的线程类。负责不停接收服务端的消息，并根据消息类型调用后台处理 Manager 类的对象进行相应的处理。

Manager：实际处理各种操作的类。提供注册、登录、发送消息、接收消息、发送文件、接收文件等各种操作的方法，被 ConnectServer 线程对象调用。

2.2.1.3 client.util 文件夹：运行过程中的工具类。

ClientThreadCollection：维护通信线程的 HashMap，提供添加、查询方法。

ManageClientCollection：维护群聊界面的 HashMap，提供添加、查询方法。

ManageClientPersonCollection：维护私聊界面的 HashMap，提供添加、查询方法。

Tools:提供窗体居中的方法。

2.2.1.4 client.ui 文件夹：四个继承自 JFrame 的界面类。

Register_Frame：注册界面

Login_Frame：登录界面

Client_Frame：群聊界面

ClientFrame：私聊界面

2.2.1.5 client.run 文件夹：客户端启动的接口（main 方法）

2.2.2 Server:

服务端的设计与客户端非常类似，在此仅叙述特别之处。

2.2.2.1 server.backstage：

ConnectDataBase：进行服务端与数据库的通信。

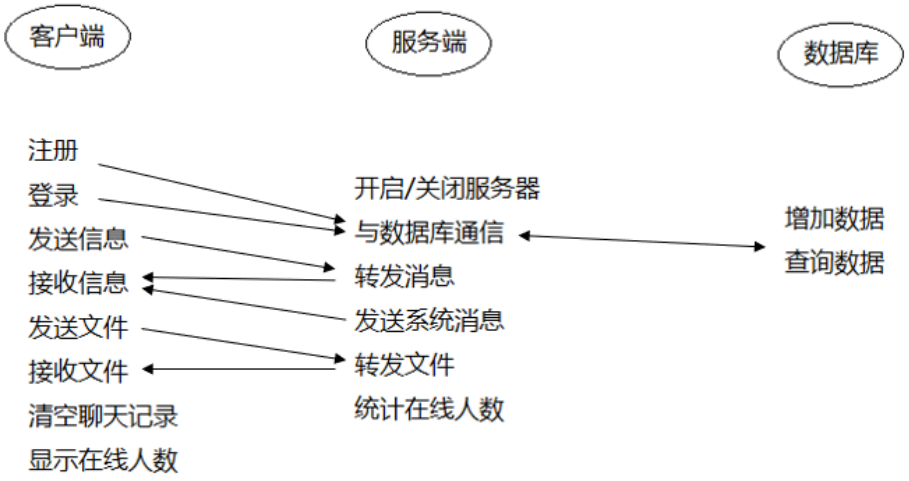
DatabaseManager：处理对数据库的操作，包括连接、增加、修改、访问数据。

2.3 数据库设计

```
mysql> SELECT * FROM users;
+--------+----------+---------+
| Name   | Password | IsLogin |
+--------+----------+---------+
| java   | 123      |         |
| tongji | 123      |         |
+--------+----------+---------+
2 rows in set (0.00 sec)
```

该数据库中仅有一个表格 users。表格存放用户信息：用户名、密码、当前是否在线。Server 程序中通过 DatabaseManager 对象对数据库进行维护。

2.4 系统架构设计



2.5 详细设计

2.5.1 GUI 界面设计

各个界面均继承自 JFrame 类，通过添加组件和事件监听处理实现相应的功能。

2.5.2 网络通信及 I/O

通过使用网络相关类和多线程技术实现服务端与客户端之间的异步通信，通过使用输入输出流（主要是对象输入输出流）对象实现数据的传递。

ChatMessage 实现 Serializable 接口，实现对象的串行化，便于把对象通过网络进行传播。

User 实现 Serializable 接口，便于把通项通过网络进行传播和存入数据库。

通过文件输入输出流对象（FileInputStream/FileOutStream）实现文件的发送。

2.5.3 通过 JDBC 使用数据库

向项目中导入驱动包 mysql-connector-java-8.0.11.jar 以使用 MySQL 数据库，在 server 中 DatabaseManager 类中加载驱动并对数据库进行维护。

2.5.4 功能实现

2.5.4.1 注册功能

服务端启动服务器，开启一个线程，监听 9999 端口，不断的接受客户端发送过来的信息。

客户端注册用户，填写用户名，密码，确定密码（判断输入信息不能为空，且对密码和确认密码进行校对）点击注册后，客户端将用户名，密码和用户操作的类型（登录，注册）封装为一个 User 对象。通过对象流，将这个 User 对象传送给服务器端。

服务器端接受到客户端发送过来的 User 对象，通过 User 对象的 getType（）方法获取用户的操作类型。如果是注册操作，则通过后台的数据库操作类对 User 对象进行解析和对数据库的信息进行匹配，如果用户名已经存在，或者操作数据库失败，则返回 false，如果添加用户信息成功，则返回 true。服务器端通过返回的结果将数据封装入一个 ChatMessage 对象，ChatMessage 类指定信息的类型（通过接口 MessageCode 对类型进行定义），信息的内容和信息的发送时间、信息的发送者和信息的接受者。服务器端将 Message 对象通过对象流的形式又发送给客户端。

客户端接受到服务器端发送过来的 ChatMessage 对象，通过 getMessageType（）得知信息的类型，并进行相应的操作。得知发送过来的类型是注册成功类型，客户端会提

示注册成功（如果是注册失败的类型，则客户端会返回注册失败）

2.5.4.2 群聊功能

客户端通过文本域输入信息，然后将信息内容，信息发送的时间，信息的发送者（所有人）信息的发送者，信息的类型（群发消息）封装为一个 ChatMessage 对象，通过对象流传送给服务器。

服务器端通过信息的类型（群发消息）进行操作，然后取出专门管理线程的集合里的每一个元素，获得每一个与客户端连接的线程，然后把信息转发给每一个在线用户。

2.5.4.3 私聊功能

客户端选择用户点击单人聊天，弹出单人聊天界面。在文本域内输入信息，点击发送。

客户端将信息封装入 ChatMessage 对象，并指定类型为单人聊天，指定发送者和接受者。

服务器端收到客户端发送过来的 ChatMessage 对象，解析 ChatMessage 对象，得到接受者，通过接受者找到集合中对应的线程，取出来并转发消息。

2.5.4.4 群发文件功能

客户端点击发送文件，打开文件选择器，选择好文件，客户端将文件的名字和发送者和信息类型（群发文件）等信息封装入 ChatMessage 对象中，发送给服务器端。并且将文件发送给服务器。

服务器端解析发送过来的信息类型（群发文件），然后取出从集合中取出所有与客户端连接的线程，将文件转发给每个在线用户。并发送系统消息：（发送者）给所有人发送了（文件名）文件。

2.6 界面设计与运行结果

服务端界面：

发送系统消息：

强制用户下线：

客户端界面：

注册：

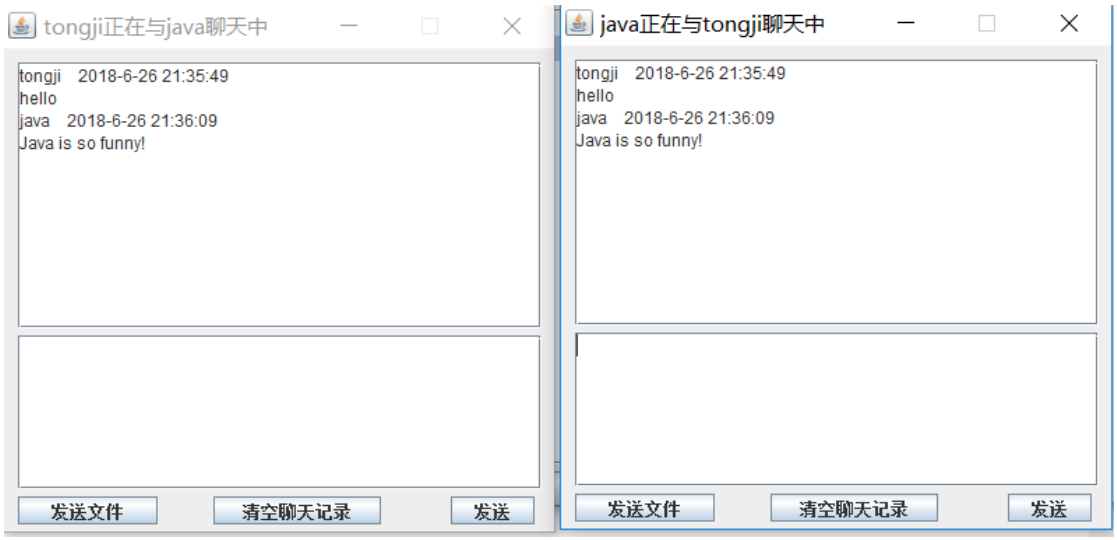
登录：


登录界面
用户名
tongji
密码
登录
重置
注册

群聊：


欢迎java进入聊天室
在线人数：2
java 2018-6-26 21:33:24
大家好
tongji
java
发送文件
清空聊天记录
发送
单人聊天

私聊：


tongji正在与java聊天中
tongji 2018-6-26 21:35:49
hello
java 2018-6-26 21:36:09
Java is so funny!
发送文件
清空聊天记录
发送
java正在与tongji聊天中
tongji 2018-6-26 21:35:49
hello
java 2018-6-26 21:36:09
Java is so funny!
发送文件
清空聊天记录
发送