

浙江大学大学计算机科学与技术学院

Java 程序设计课程报告

2020—2021 学年秋冬学期

题目	仿制 QQ 群的多人聊天系统
学号	3180106071
学生姓名	刘轩铭
所在专业	软件工程
所在班级	软工 1801

目录

- 1 引言 1
 - 1.1 设计目的 1
 - 1.2 设计说明 2
- 2 总体设计 3
 - 2.1 功能模块设计 3
 - 2.2 流程图设计 4
- 3 详细设计 5
 - 3.1 服务端设计 5
 - 3.2 客户端设计 7
 - 3.3 数据库设计 10
- 4 测试与运行 11
 - 4.1 程序测试 11
 - 4.2 程序运行 11
- 5 心得与总结 16

1 引言

本次作业是使用 Java 的 Swing 框架开发具有 GUI 的网络应用，支持多用户使用，同时也要使用数据库，记录用户的信息。我选择使用 Swing 开发了一个仿制 QQ2021 的 QQ 群聊系统，包含注册、登录、聊天等功能。通过此次编程，加深了自己对 Java 的理解，提高了自己的编程水平。

1.1 设计目的

实现的功能有：

1. 使用 GUI 编程。用户界面用 GUI 实现；
2. 使用网络编程。支持多用户同时使用；
3. 使用数据库(记录用户信息，发送的消息)；
4. 使用并发编程，每个用户采取多线程调度，也支持多个客户端在多个进程下使用

此次的 QQ 群聊应用，使用到了 socket 编程，多线程，数据库，以及用户的注册登陆等功能，满足老师布置的实验要求，并且编程过程中，也对照老师的代码要求，对代码进行了规整并且给予了较为充足的注释。

1.2 设计说明

本程序采用 Java 程序设计语言，使用 Maven 包管理工具，在 IntelliJ IDEA 平台下编辑、编译与调试。具体程序由我个人开发而成。工作时间轴如表 1 所示：

表 1 工作时间轴表

时间	完成的主要工作
12 月 31 日	整个程序前期的需求分析和整体功能的架构 阅读并学习 Swing, Java 线程库, Java 网络编程 等相关知识, 熟悉用法
1 月 3 日	开始代码的编写工作, 并对代码进行测试, 查找 bug。
1 月 7 日	完成报告的撰写工作

2 总体设计

2.1 功能模块设计

本程序需实现的主要功能有：

- 用户登陆
- 用户注册
- 用户退出
- 多人聊天并共享消息，并由多线程进行调度
- 数据库存储历史消息

程序的总体功能如图 1 所示：

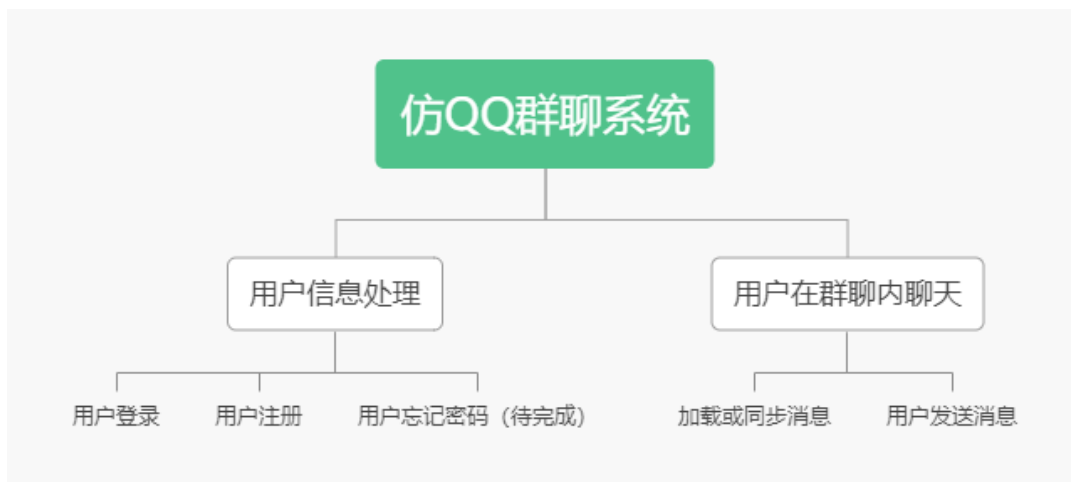


图 1 总体功能图

2.2 流程图设计

程序总体流程如图 2 所示：

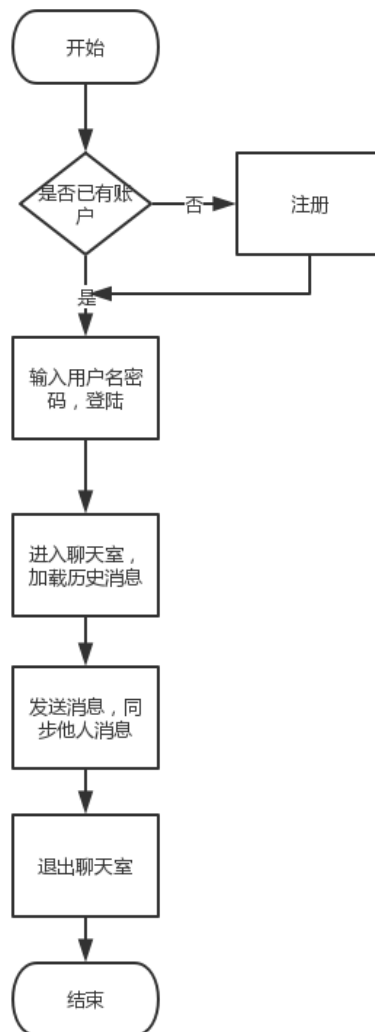


图 2 总体流程图

3 详细设计

在设计过程中，主要分为三部分，客户端设计，服务器设计，数据库设计和管理。

以下将逐一进行说明。

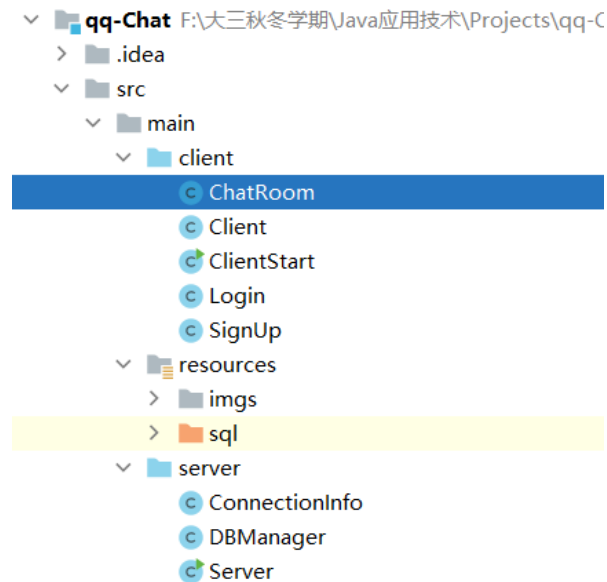


图 3 项目目录结构

3.1 服务端设计

服务端主要提供，监听端口并接受客户端的建立、与数据库建立连接、收发客户端的信息这三个功能。主要设计如下：

Server 类：

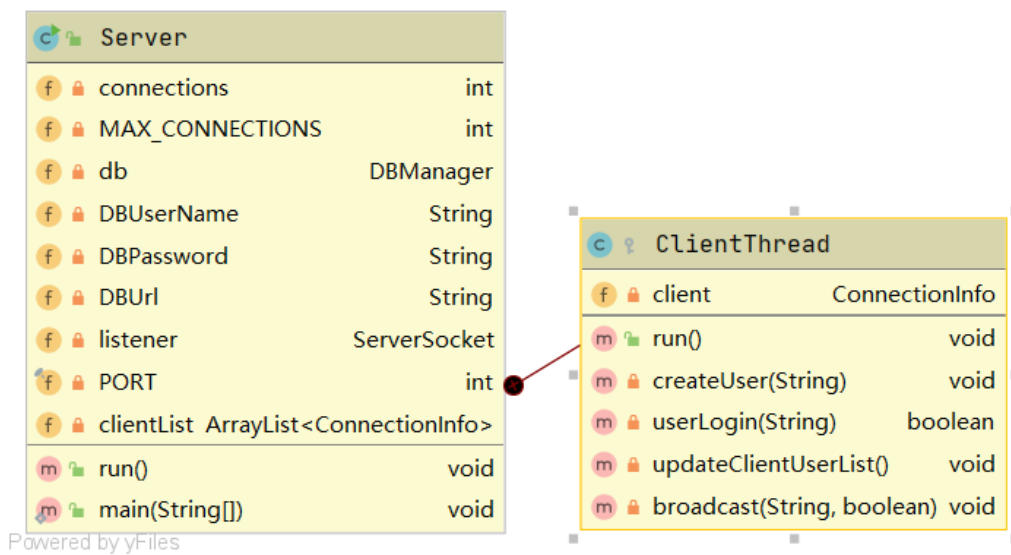


图 4 Server 类 Class Diagram

在该类中运行 `main` 方法，启动服务端。类中定义了服务端运行的端口，连接的数据库及用户信息。Server 类中最主要的方法是 `run`，`run` 中对 6666 的端口进行监听。此外，Server 中还含有一个内部类 `ClientThread`，用来让新的线程执行。当有客户端连接到端口上时，服务端会新开一个线程用来管理该用户的行为，比如用户申请创建用户，登录等行为等。

此外，服务端还有一个 `ConnectionInfo` 类，用来存储用户的信息，并处理客户端的发送和接收消息实现。其中的 `outputWriter` 和 `inputWriter` 实现了与客户端 socket 的信息传输功能。

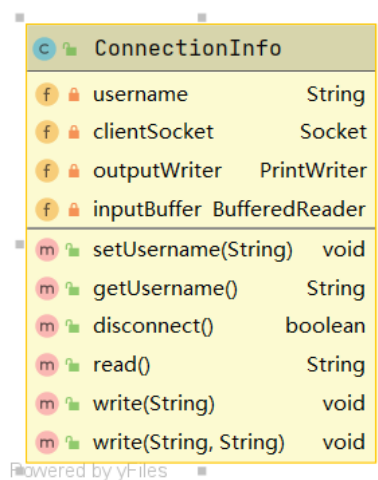


图 5 ConnectionInfo 类 Class Diagram

3.2 客户端设计

客户端在 ClientStart 中调用 main 方法启动客户端。然后先建立 Login 页面，提供用户登录功能。用户（注册）登录后，会显示 Chatroom 页面，即进入聊天室，并完成聊天等功能。

伴随着客户端启动，客户端会实例化一个 Client 对象。其内容如下：

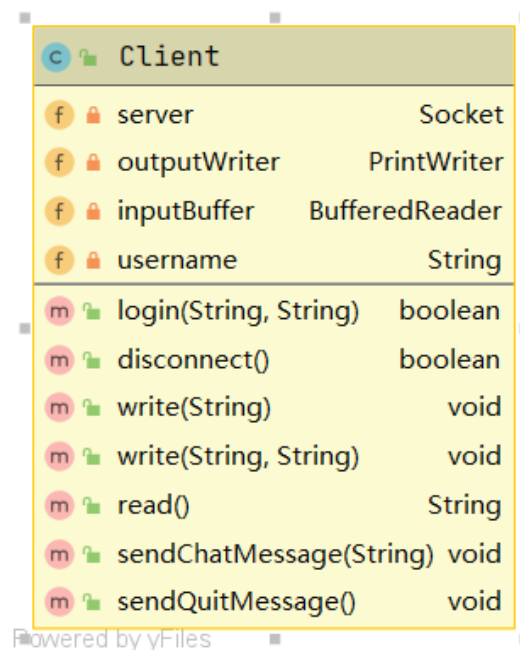


图 6 Client 类 Class Diagram

该对象申请一个 socket 套接字然后连接到对应的 IP 和端口。然后伴随着用户的一系列点击行为，执行不同的方法，来完成和服务端的交互。例如，当用户输入了用户名和密码，会调用 login 方法：

```
1. public boolean login(String user, String pass) {
2.     // 登陆是否成功
3.     boolean accepted = false;
4.
5.     // 传送用户名和密码
6.     outputWriter.println("LOGIN: " + user + "," + pass);
7.     outputWriter.flush();
8.
9.     //服务器返回消息
```

```

10.    String response;
11.    try {
12.        response = inputBuffer.readLine();
13.        System.out.println("SERVER: " + response);
14.        if(response.equals("ACCEPTED")) {
15.            accepted = true;
16.            username = user;
17.        }
18.    } catch(IOException e) {
19.        e.printStackTrace();
20.    }
21.
22.    return accepted;
23. }

```

客户端通过特定的标记告诉服务端用户正在登陆，客户端验证后将结果返回，完成一次交互。其他的功能还包括发送聊天消息，断开链接等。

客户端还完成了对于界面的绘制功能。实现的三个界面分别为聊天界面，登陆界面和注册界面。其中登陆界面完成了 UI 的设计，仿照 QQ2021 进行设计完成。其他两个页面由于时间关系，暂时没有设计精美的 UI，后期可以进行优化。

下面以登陆界面为例说明设计：



图 7 登陆界面（仿照 QQ2021）

该页面由 Login 类进行实现。

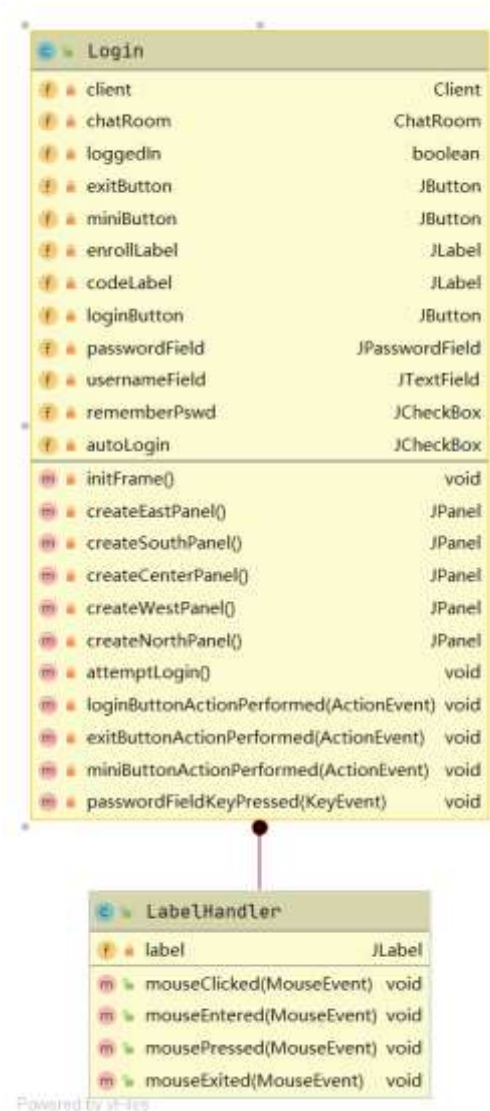


图 8 Login 类 Class Diagram

该页面采用 BorderLayout 布局方式，通过五个 create 方法创建了东南西北中五个 panel 并添加到 JFrame 中。在每个 panel 中都镶嵌了一定的内容。用户点击登录，会调用 attemptLogin 方法，尝试和服务端沟通并进行登录；用户点击注册，会切换到注册界面；用户点击最小化，会让窗口最小化……这些功能都进行了实现。记住密码和忘记密码这两个功能，由于时间关系暂时没有进行实现，可以在之后的版本中进行更新。

3.3 数据库设计

数据库采用了 MySQL 8.0.16 版本，设计了如下的模式：

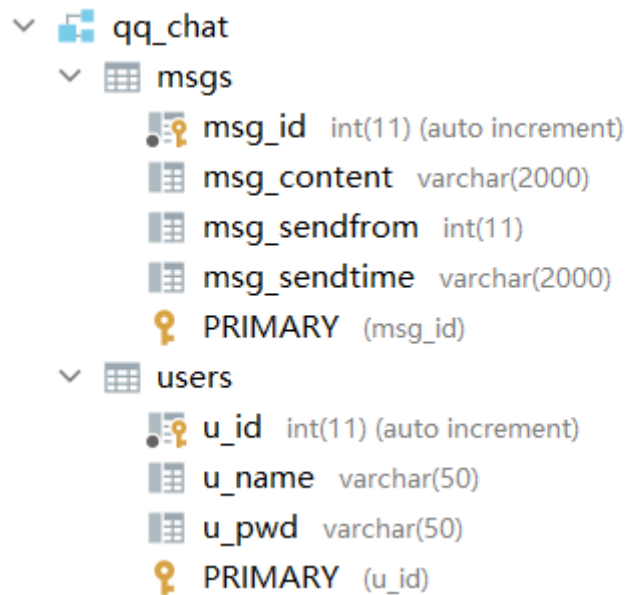


图 9 数据库模式图

msgs 表中存放发送消息的用户名，消息内容和时间，users 存放用户名、密码。由 DBManager 类进行管理：

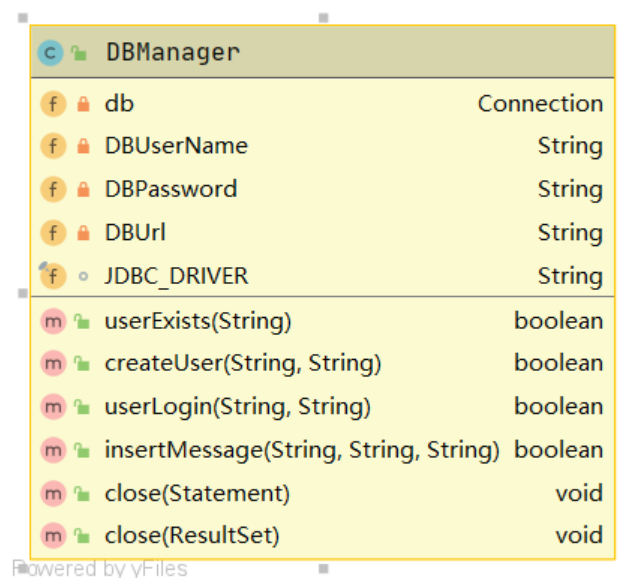


图 10 DBManager 类 Class Diagram

数据库实现了对于两个表的管理，以及四个操作方法：判断用户是否存在，创建新的用户，用户登录检验，收发信息的储存。

4 测试与运行

4.1 程序测试

在程序代码基本完成后，经过不断的调试与修改，最后测试本次所设计的多人聊天可以正常运行，没有出现明显的错误和漏洞，但是在一些方面仍然需要完善，比如用户交互方面，可以进行优化 UI 设计，可以扩展私聊的功能，读取历史消息等功能，另外随着程序规模扩大，数据库最好也有更好的设计。

4.2 程序运行

```
G:\Java\bin\java.exe "-javaagent:G:\JetBrains\IntelliJ IDEA 2020.2.2\lib\
SERVER: Connect Database...
SERVER: Connection established.
SERVER: Listening for client...
```

图 11 服务端运行



图 12 客户端运行



图 13 输入不存在的账户信息



图 14 用户进行注册：成功



图 15 用户进行注册：失败



图 16 用户输入正确的信息进入聊天室

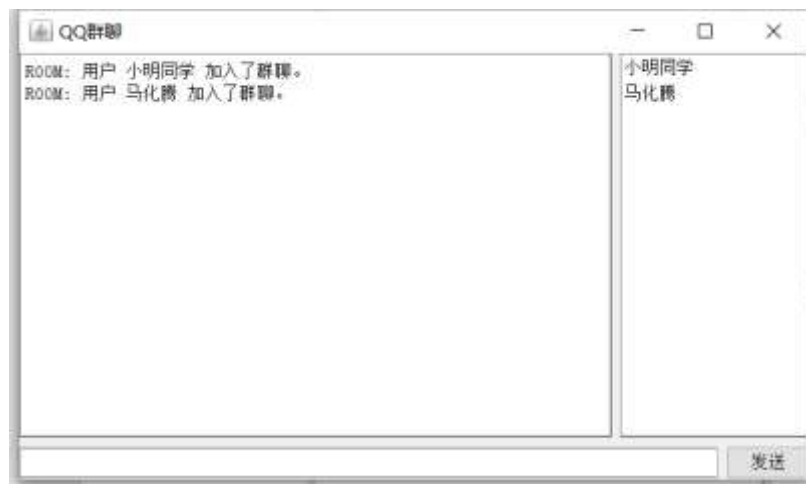


图 17 第二个用户进入聊天室（此时有二个进程）

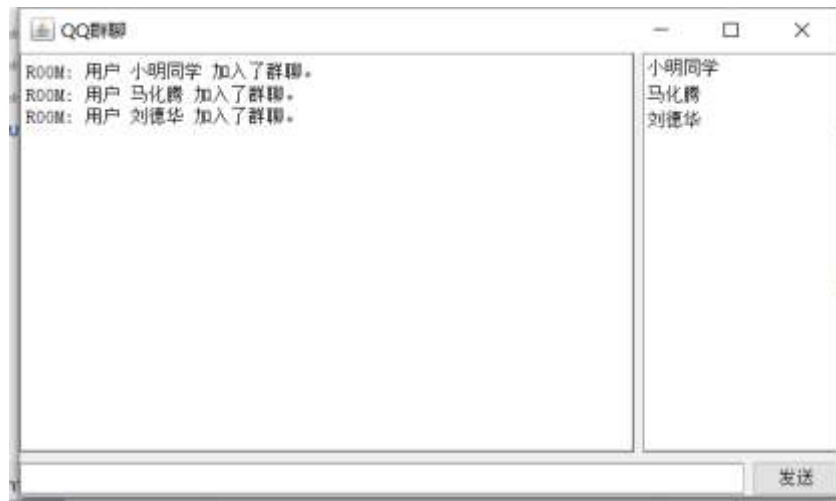


图 18 第三个用户进入聊天室（此时有三个进程）





图 19 三号用户发送消息



图 20 随机的对话内容

5 心得与总结

此次的实验，是比较有难度的，除了要学习 **Swing** 和 **Socket** 等库的使用方式，还需要综合的利用这些知识进行编程和实验。

通过本次实验，我对于 **Java** 的 **GUI** 编程有了一定的认识，明白了存在多种不同的布局方式可以进行利用。同时，我也明白了一定要善于阅读文档，来获取重要的知识。另外，如何设计好看的界面也是十分重要的。

在数据库的使用方面，我也学会了如何用 **Java** 连接数据库并进行操作。最大的收获在于，查询操作的时候，一定要加入引号，这个问题一开始困扰着我，导致有许多的 **bug**，后来查询相关网页才解决了这个问题。

```
ps = db.prepareStatement("SELECT * FROM users WHERE u_name = '" + username + "' and u_pwd = '" + password + "'");  
rs = ps.executeQuery();
```

最后还是要感叹，**Java** 的封装还是非常方便的，希望之后能尝试更多的项目。这次实验有不足在于，时间比较紧，课程任务比较多，导致 **Ui** 设计等还是有比较多的不足，希望之后能有时间继续改进，做出更加优秀的版本。

