

# Java 大作业“聊天系统”实验报告

秦文涛 1500012882

指导老师：唐大仕

# 全文目录

全文目录	1
第一章 功能描述	2
第二章 功能实现	3
1. 用户界面	3
2. 数据库的设计	7
3. 服务器与客户端的传输	10
第三章 不足与展望	12

# 第一章 功能描述

本次大作业，使用 Java 编写了一个类似于 QQ、微信等社交软件的程序软件。该软件支持 QQ 软件的功能子集，叙述如下：

- 账号的注册与登录
- 按账号搜索用户，添加好友，与好友进行聊天
- 创建群聊房间，添加指定群号的群聊房间，支持群聊
- 与好友分享文件
- 用户头像的更改

## 第二章 功能实现

这一章，先在第1节通过一些截图，一方面展示软件的界面效果，另一方面，具体地展现一下软件的功能。然后在第2节，介绍了数据库的设计。之后在3节，描述了服务器与客户机的传输流程。

### 1. 用户界面

当客户端启动的时候，首先呈现给用户的是登录界面，如图1所示。当用户



图 1 登录界面

拥有账号的时候，直接输入账户名（ID）和密码（PASSWORD），然后点击**登录**按钮，即可登录。如果没有账号，则点击**注册**按钮，在注册界面进行账号的注册。注册界面如图2所示。用户可以输入用户名、账号并设定密码，然后点击**确定**按钮即可注册。如果账号与已有用户的账号重复，则提示注册失败。

用户登录成功之后，会显示程序的主界面，主界面如图3中左边的图所示。主界面里面包含诸多元素，描述如下：

- 头像，用户名。点击头像可以修改头像



图 2 注册界面

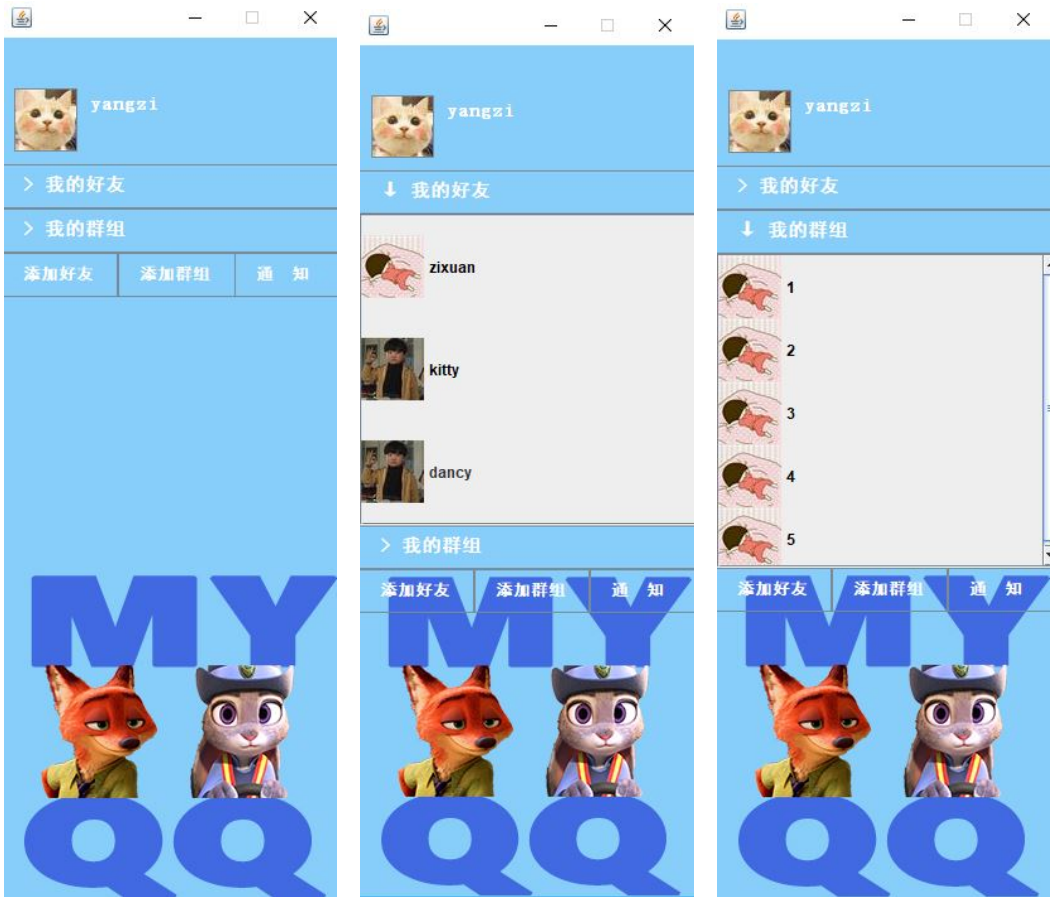
- 我的好友。里面展示了我的好友列表
- 我的群组。里面展示了我的群组列表
- 添加好友。可以搜索用户账号，进行好友的添加
- 添加群组。可以搜索群号，加入新的群聊。也可以创建群聊
- 通知。显示好友添加的请求

在主界面中点击**我的好友**，会显示出我的好友列表，如图3中中间的图所示。在主界面中点击**我的群聊**，会显示出我的群聊列表，如图3中右边的图所示。

主界面的左上方显示了我的头像和我的用户名，点击我的头像可以选择已有的其他图片作为新的头像。选择头像的界面如图4所示。选择一张图片，点击**submit** 按钮，头像会随之改变。右键主界面还可以更换背景图。

在主界面，点击**添加好友**按钮，会弹出添加好友界面，如图5的上边所示。在输入框中输入已存在用户的账号，再点击旁边的搜索图标，会弹出相应的用户供添加好友使用。

与添加好友类似，在主界面，点击**添加群组**按钮，会弹出添加群组界面，如图6的上边所示。在输入框中输入已存在的群组号，再点击旁边的搜索图标，会弹出相应的群组供添加好友使用。



(a) 主界面                      (b) 我的好友列表                      (c) 我的群聊列表

图 3 左图是刚刚登录时的主界面的样子，点击**我的好友**可以显示我的好友列表，如中间的图所示。点击**我的群组**可以显示我加入的群聊，如右图所示

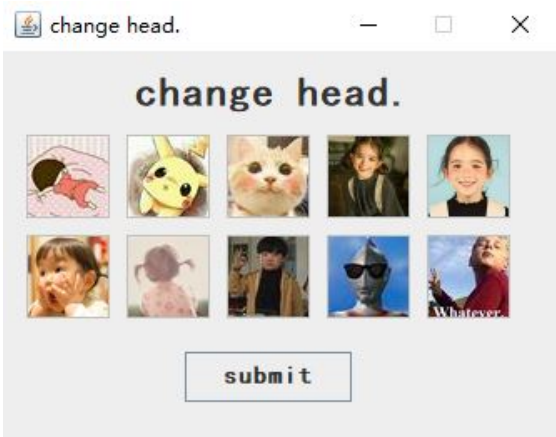


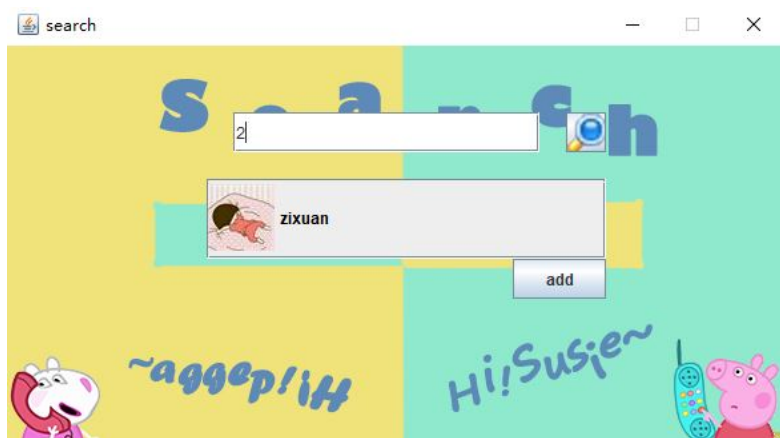
图 4 选择头像界面

在添加群组界面，还存在一个**创建群聊**按钮，点击它会弹出创建群聊的界面，如图7所示。输入新的群号，点击确定按钮，即可创建新的群聊。

点击主界面的**通知**按钮，会弹出新的好友通知界面，如图8所示。可以接受



(a) 添加好友初始界面



(b) 搜索好友并添加

图 5 上图是点击主界面**添加好友**后弹出的初始页面，搜索存在的用户之后界面如下图所示，可点击其中的 **add** 按钮进行好友的添加

或者拒绝其他用户的添加好友请求。

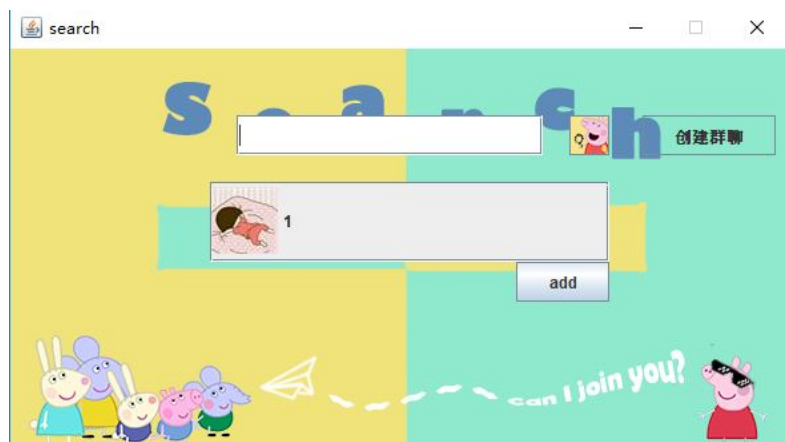
在我的好友列表中，双击好友的头像，会弹出和好友的聊天界面，如图9所示。在界面下方的文本框中，可以输入聊天文字，再点击**发送**按钮，即可向好友发送消息。点击界面左上角的图标，可以选择文件进行发送。在本机同时运行两个客户端，模拟用户聊天和文件发送，演示如图10所示。

在我的群组列表中，双击群组头像，会弹出群聊的聊天界面，如图11所示。在界面下方的文本框中，可以输入聊天文字，再点击**发送**按钮，即可向群聊中发送消息。群聊界面的右边显示了群成员，在线的群成员的会将其背景置为绿色。同样，为了模拟群聊，在本机同时运行两个客户端，进入同一个群聊房间，演示如图12所示。

通过上述的截图与描述，软件的大部分界面和功能都已经得到了展示。在之



(a) 添加群组初始界面



(b) 搜索群组并添加

图 6 上图是点击主界面**添加群组**后弹出的初始页面，搜索存在的群组之后界面如下图所示，可点击其中的 **add** 按钮进行群组的添加

后的两节，将从数据库设计和服务器和客户端的传输层面，来叙述一下软件的技术细节。

## 2. 数据库的设计

程序使用了 MySQL 操作数据库。针对程序的功能，设计了六张表用来存储数据，数据库的设计如表1所示。

下面详细地解释一下各个表的含义和作用。**admin** 表存储量注册用户的信息，包括其账号、用户名、密码，以及头像文件与主界面背景图片，其中账号为主码。用户在登录或者注册时，需要查看或修改 **admin** 表。**friend** 表记录了好友的关系，其包含两个属性，**id** 与 **friend**，为好友双方的账号，二者在一起构成了 **friend** 的主码。**group\_head** 存储了群聊的信息，包括群号 **id** 和群头像文件 **head**，



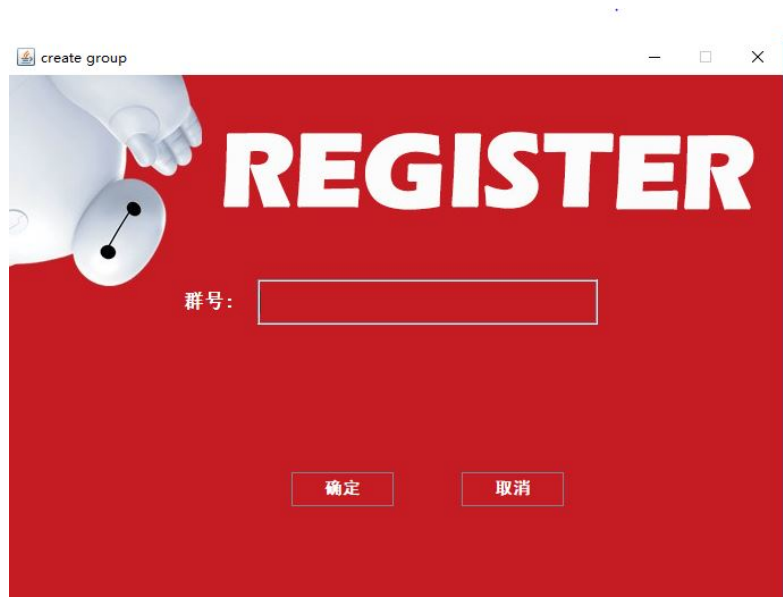


图 7 创建群聊界面

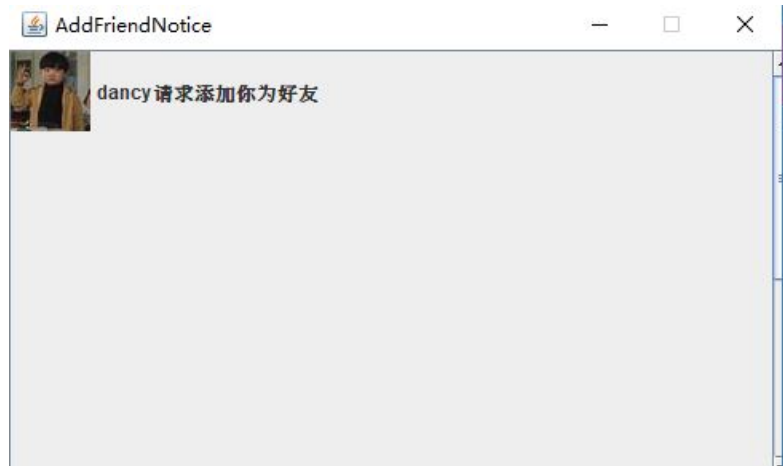


图 8 好友通知界面

表名	属性
admin	<u>id</u> (账号) , name (用户名) , pwd (密码) , head (头像) ,back (背景图片)
friend	<u>id</u> (好友一方) , <u>friend</u> (好友另一方)
group_head	<u>id</u> (群号) , head (群头像)
user_group	<u>id</u> (用户账号) , <u>group_id</u> (群号)
notice	<u>admin_id</u> (被添加好友的用户) , <u>friend_id</u> (请求添加好友的用户)

表 1 数据库中的表

其中 id 为主码。user\_group 记录了用户和群聊的关系，其中的 id 为用户账号，group\_id 为群号，二者共同组成 user\_group 的主码。当用户 a 在群聊 B 中时，则 user\_group 中会多出由用户 a 的账号和群聊 B 的群号组成的项。表 notice 记录了



图 9 和好友的聊天界面

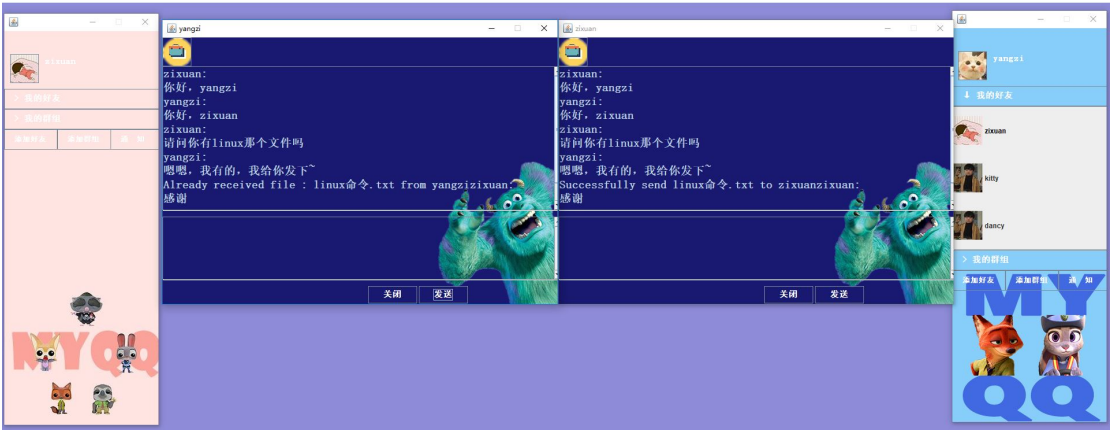


图 10 运行两个客户端进行聊天的演示

未验证的好友添加请求，admin\_id 为被添加的好友账号，friend\_id 为请求添加好友的用户账号。

在服务器和客户端程序运行之前，需要初始化数据库中的表。提前使用 MySQL 语句建立上述提到的表，MySQL 语句如图13所示。

用户在与软件交互的过程中，在许多地方会触发对数据库的访问。如在登录、注册的时候，在添加好友、添加群组、创建群组的时候，等等。

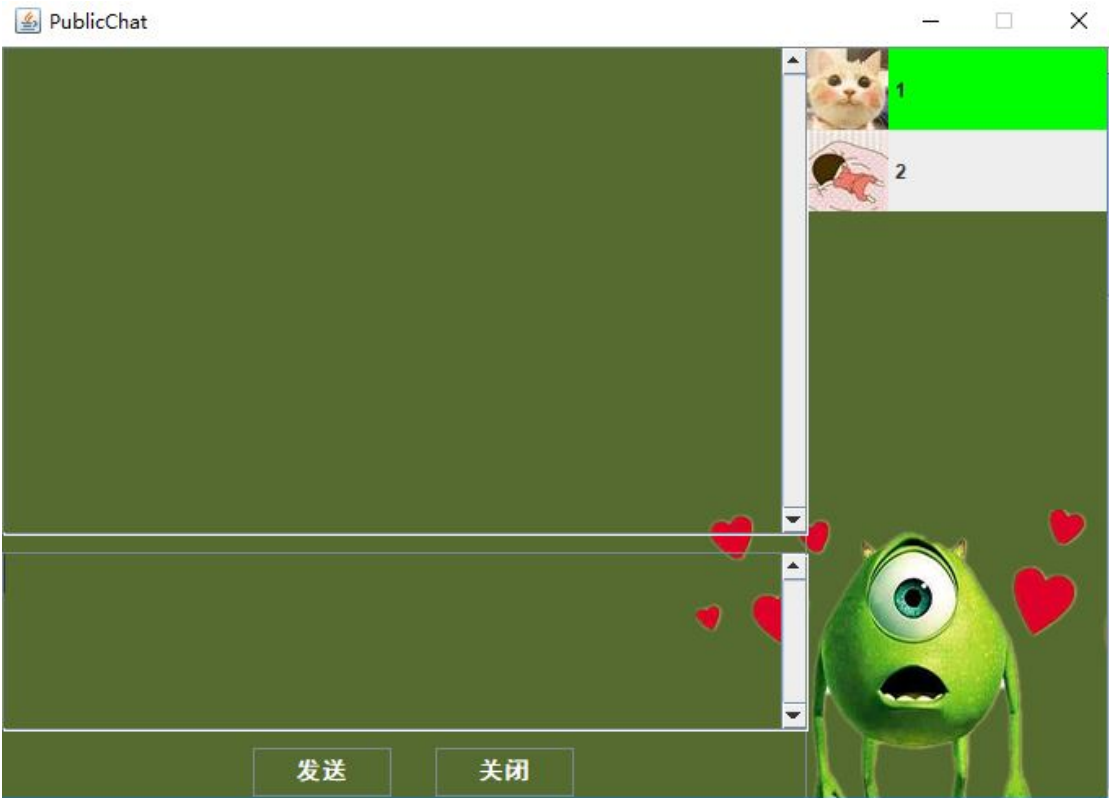


图 11 群聊的聊天界面

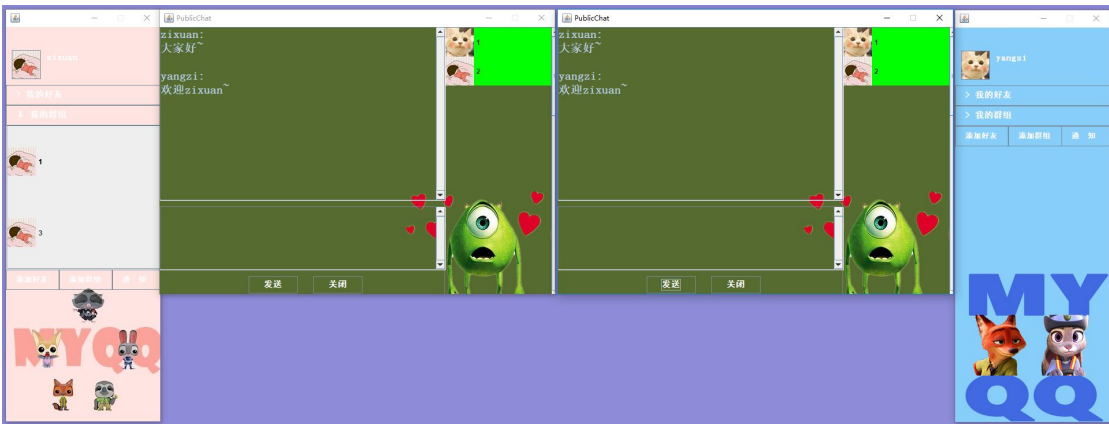


图 12 运行两个客户端进行群聊的演示

### 3. 服务器与客户端的传输

对于用户间的聊天和文件传输，程序采用了客户-服务器模式，即用户所在的客户端只能和服务器进行通信，因此用户之间的信息传输需要通过服务器的中转。以与好友的聊天为例，当用户 a 准备向用户 b 发送消息 M 时，a 所在的客户端首先向服务器发送包含 a、b、M 的一段信息，之后服务器再向 b 发送包含 a、M 的一段信息。对于文件的传输也是一样，只不过需要传输更多的内容，包

```
create table admin(  
    id int primary key,  
    name varchar(20),  
    pwd varchar(20),  
    head varchar(20),  
    back varchar(20)  
);  
create table friend(  
    id int primary key,  
    friend int primary key  
);  
create table group_head(  
    id int primary key,  
    head varchar(20)  
);  
create table user_group(  
    id int primary key,  
    group_id int primary key  
);  
create table notice(  
    admin_id int primary key,  
    friend_id int primary key  
);
```

图 13 建表的 MySQL 语句

括文件名、文件大小和文件内容。对于群聊，如用户 a 准备向群 B 发送消息 M，则首先由 a 所在的客户端向服务器发送包含 a、B、M 的一段信息，之后服务器先得到群 B 中的所有群成员，向每个群成员发送包含 a、M 的信息。

## 第三章 不足与展望

本次大作业实现的软件有以下几点不足之处：

1. 数据库未能存储聊天的信息，导致重新打开聊天界面（关闭再打开，或者下次登录时再打开），不能看到之前发过的消息
2. 客户端和服务端一样，都可以直接操控数据库

针对第 1 个问题，可以在数据库中存储聊天的信息，然后在用户重新打开某个聊天界面时，从数据库中找到与该聊天相关的信息，然后在界面显示。对于第 2 个问题，解决方式是只允许在服务器端访问操控数据库，而当客户端需要访问时，则由服务器作为访问数据库的中转站。

完成时间：2019 年 6 月