

# Java 多线程聊天室

## [软件设计说明书]

小组名：Java 冲冲团

成员	季晓东
成员	邓浩
成员	于添

## 目录

1 项目简介.....	3
1.1 项目概述.....	3
1.2 项目目的.....	3
1.3 命名规范.....	3
1.4 使用到的第三方资源 .....	3
1.5 文件以及网络解析规则.....	3
1.6 基本功能.....	4
1.7 配置要求.....	4
1.8 类图.....	4
2 总体设计.....	7
2.1 主要结构设计 .....	7
2.2 软件运行环境设计 .....	8
2.3 包清单 .....	8
2.4 功能清单 .....	9
3 功能模块设计 .....	11
3.1 登录/注册 .....	11
3.2 主界面 .....	13
3.3 群聊.....	15
3.4 私聊.....	16
3.5 创建群聊 .....	18
3.6 添加好友 .....	20
4 数据库设计 .....	21
4.1 数据库表名列表 .....	21
4.2 数据库表之间关系说明.....	22
4.3 数据库表的详细清单 .....	23
5 存储过程设计 .....	25
5.1 个人信息管理系统.....	25
5.2 聊天管理系统 .....	26
5.3 消息系统 .....	26
6 系统出错处理设计 .....	28
6.1 出错信息 .....	28
6.2 系统维护设计 .....	28

# 1 项目简介

## 1.1 项目概述

基于 Java 多线程和网络编程的聊天室。JDK 版本 1.8，UI 界面使用 JavaFx 设计，并通过 Maven 添加组件包 jfoenix 美化界面。

## 1.2 项目目的

实现多人多群组聊天，并能够实现添加好友、私聊群聊功能。

## 1.3 命名规范

变量命名规则：函数名、自定义变量名使用小驼峰命名法；组件名用大驼峰命名法；组件引用、Css 样式、事件名采用 kebab-case 命名法；保证命名至少由两个有意义的单词构成。

数据库对象命名规则：

## 1.4 使用到的第三方资源

添加依赖：jfoenix:8.0.8

界面图标资源来源：阿里矢量图标库

用户头像资源来源：b 站 up QuAn\_\_

## 1.5 文件以及网络解析规则

文件读写规则：

获取字节流，通过 InputStreamReader 桥接为字符流，并通过缓冲流提高读入效率

输出使用 PrintWriter 向字节流中写入数据

网络协议：通过套接字进行 TCP 协议通信

## 1.6 基本功能

- 1.登录:
- 2.注册:
- 3.加好友:
- 4.创建群聊:
- 5.加入群聊:
- 6.发消息:
- 7.历史记录:

## 1.7 配置要求

### 服务器运行要求:

1. JDK 1.8
2. SDK 1.8

### 服务器运行方法:

1. 导入数据库文件
2. 修改数据库账号
3. 运行 Server Start.bat

具体说明请查看 git 仓库中 README 文件

## 1.8 类图

通过 IDEA 导出生成:





## 2 总体设计

### 2.1 主要结构设计

几个主要结构关系如图 2-1 所示

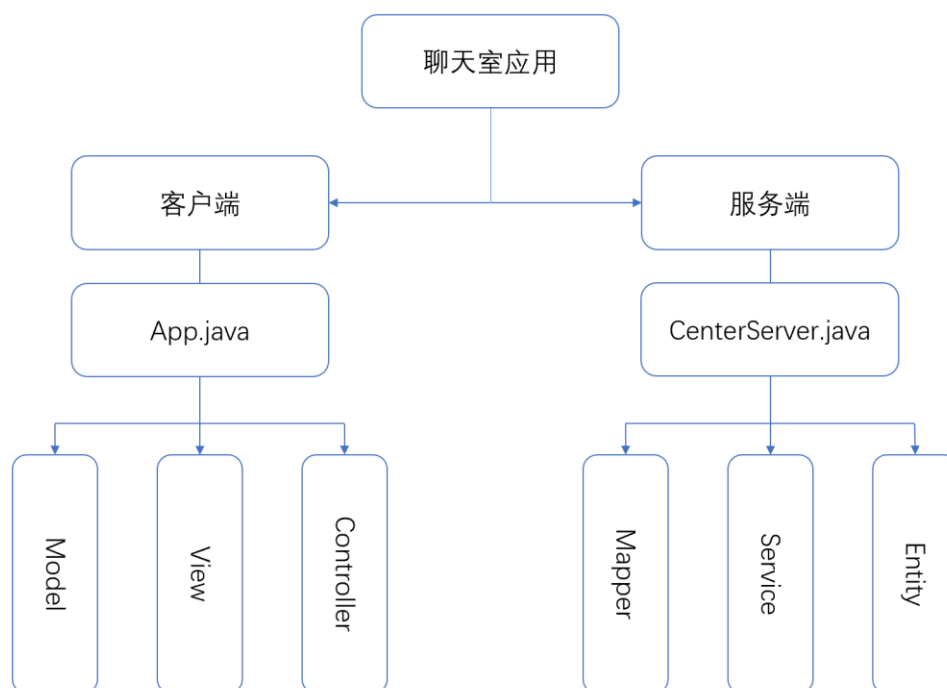


图 2-1 主要结构关系图

客户端：

App.java:客户端入口，加载'login'窗口

Model: 一些实体类

View: 登录注册等界面

Controller: 核心类，初始化界面，获取数据并渲染到界面上，实时监控

服务端：

CenterServer: 主类

Entity: 实体类

Service: 服务类

Mapper: 数据库操作相关

## 2.2 软件运行环境设计

软件平台：

- (1) 服务器操作系统：Windows 10
- (2) 数据库管理系统：msSQL
- (3) 项目运行环境：JDK1.8
- (4) 客户机的操作系统：Windows 10

硬件平台：

- (1) 运行设备 1：Windows 10
- (2) 运行设备 2：msSQL
- (3) 运行设备 3：JDK1.8

## 2.3 包清单

客户端包清单如表 2-1 所示

**表 2-1 客户端包清单**

包名称	功能
Data 包	存放 Model 类
Controller 包	存放 controller 类
Component 包	存放一些已封装控件
Util 包	存放工具类 DataUtil DataManager

服务端包清单如表 2-1 所示

**表 2-2 服务端包清单**

包名称	功能
center 包	服务端主程序、自定义套接字和服务端 Controller
entity 包	存放实体类，包括好友群聊信息等
exception 包	存放自定义异常类
mapper 包	数据库操作映射层
Service 包	服务层



## 2.4 功能清单

功能清单如表 2-2 所示

表 2-2 功能清单

功能名称	客户端请求格式	服务端返回格式	说明
登录	login#username#password	成功返回: login#200+start#{uid}&{username}&{avatar}&{sex}&{introduce}#{uid}&{username}&{avator}@@... (好友消息)...#{gid}&{groupname}&{avatar}@@... (群聊消息)...# 密码错误返回: login#401 用户不存在返回: login#402 用户已登录返回: login#403	成功返回 id、用户名等信息
注册	register#username#password#sex#introduce	注册成功返回: register#200#uid 注册失败返回: register#404	注册成功分配新 id
加好友	add#uid	成功返回: add#200 失败返回: add#405	通过 id 加好友并返回状态码
创建群聊	cgroup#uid#uid...	成功返回: cgroup#gid 失败返回: cgroup#406	成功创建为新群聊分配 id
加入群聊	jgroup#gid	成功返回: jgroup#200 失败返回: jgroup#407	通过 id 加群并返回状态码
发消息	群聊: G#gid#uid#message	G#gid#uid#message#timestamp、	gid 指定群聊, uid 指定发信人, 并传送时间

			戳
	私聊: P#uid_from#uid_to #message	P#uid#message#timestamp	客户端标 记收发 id, 服务端只 传送发信 人
历史 记录	群聊: gh#group_id	gm#{gid}#{uid}@@{username}@@{avatar}#{uid}@ @{username}@@{avatar}@@... gh#{gid}#{uid}@@{time}@@{content}#{uid}@@{ti me}@@{content}#..	先传达群 聊信息, 在 传达群聊 历史消息
	私聊: ph#user_id	ph#{uid}#{sender_name}@@{time}@@{content}#{ sender_name}@@{time}@@{content}#..	传达私聊 消息包括 发信人 id 和消息等

## 3 功能模块设计

### 3.1 登录/注册

**包含功能模块：**个人信息管理

**页面参数：**userid，为主键，一个用户只有一个唯一的userid。username 和 avatar 可以重复

**调用背景：**当程序运行时，会跳转到登录注册页面。当注册完毕或者已注册用户点击“登录”时，可直接登录，未注册用户可以点击注册进行注册。

**页面组成：**头像、用户名栏、密码栏

**页面制作示意图：**



图 3-1 注册界面



图 3-2 注册成功界面

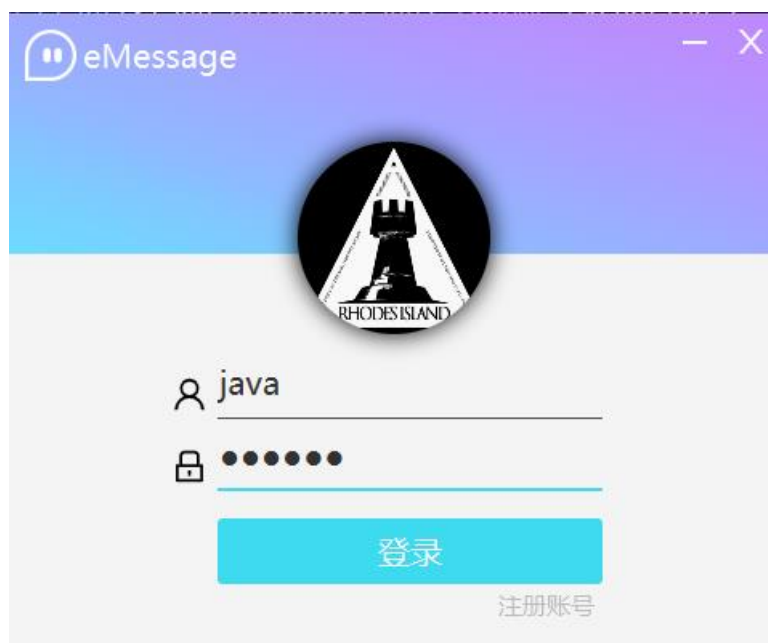


图 3-3 登录界面

## 页面使用简介：

当程序启动时，会跳转到登录页面。可以点击注册账号按钮跳转到用户注册，用户需要填写用户名、密码、确认密码、个性签名和性别。注册成功后会返回系统分配的 id。登录时可以使用用户名进行登录。

## 3.2 主界面

**包含功能模块：**个人信息管理、聊天管理

### 页面参数：

自身个人信息，包括聊天、个人资料等。

**调用背景：**当用户登录后，首先就会该页面

### 页面组成：

1. 头像以及个性签名
2. 好友栏
3. 群聊栏

## 页面制作示意图：

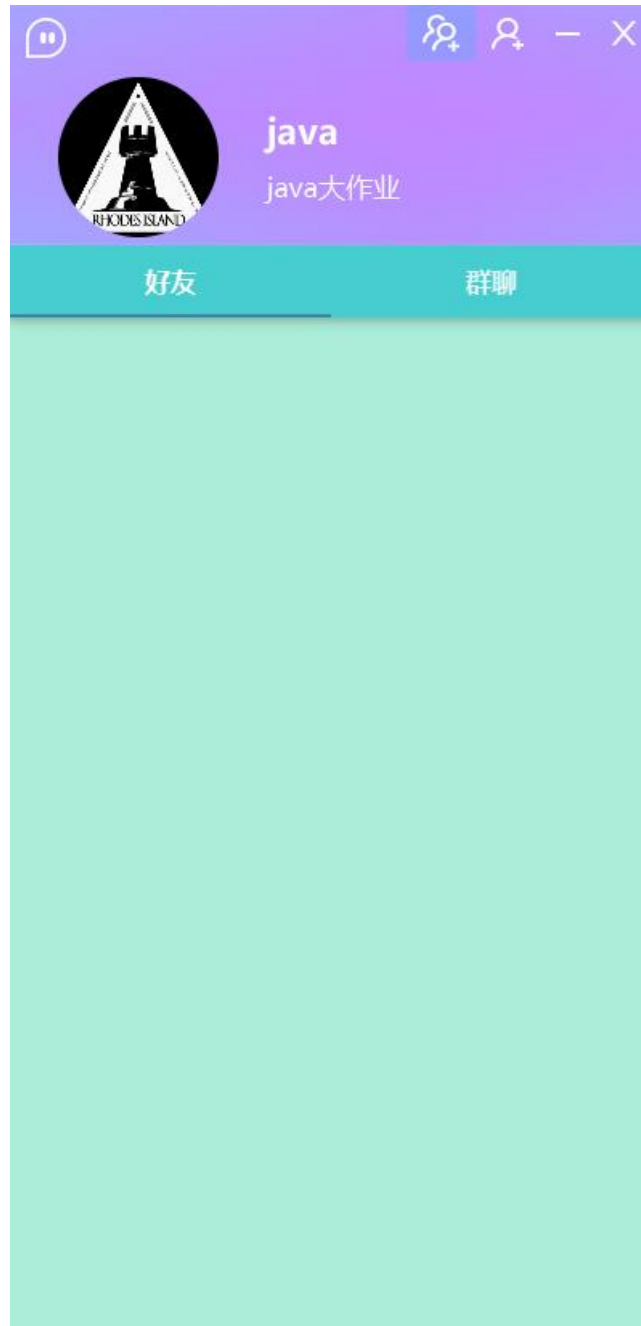


图 3-4 主界面

## 页面使用简介：

1. 当用户进入页面，会在页面上方显示个人头像、签名和用户名
2. 在个人信息下分别是好友栏和群聊栏，点击可分别索引好友和群聊
3. 点击好友可以同步私聊历史记录到本地

4. 点击群聊可以同步群聊历史记录到本地
5. 界面右上角有创建群聊和添加好友功能按钮

### 3.3 群聊

**包含功能模块：**聊天模块

**页面参数：**

1. 输入的文本框
2. 发送人 id

**调用背景：**

在主页面中点击对应的群聊即可进入

**页面组成：**

1. 群聊名称
2. 群聊成员
3. 群聊历史记录
4. 输入文本框以及发送按钮

## 页面制作示意图:



图 3-5 群聊界面

## 页面使用简介:

1. 用户可以在文本框输入文本并发送
2. 用户可以查看历史记录
3. 用户可以点击群聊成员进行私聊

## 3.4 私聊

### 包含功能模块：聊天模块

## 页面参数:

1. 输入的文本框
2. 发送人 id



### 3. 对方 id

## 调用背景:

在主页面中点击对应的好友即可进入

## 页面组成:

- 1 好友名称
- 2 历史记录
- 3 输入文本框以及发送按钮

## 页面制作示意图:

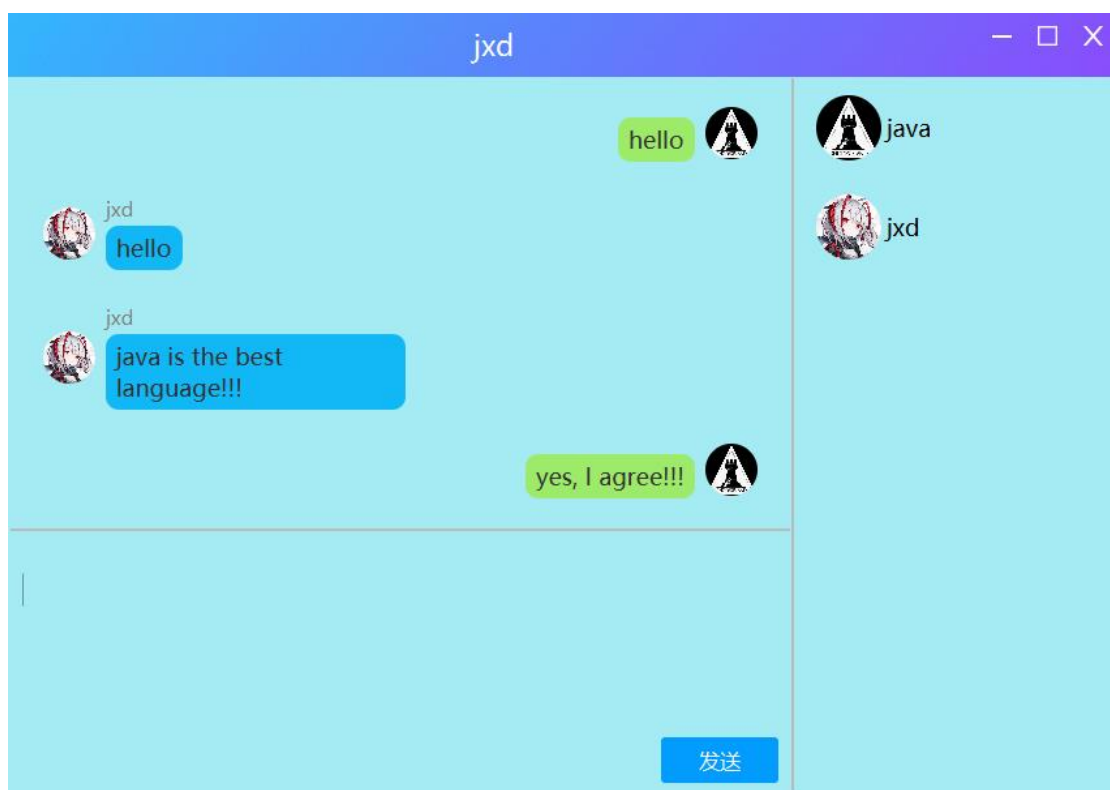


图 3-6 私聊界面

## 页面使用简介:

1. 用户可以在文本框输入文本并发送
2. 用户可以查看历史记录
3. 上方显示对方用户名

## 3.5 创建群聊

### 包含功能模块：群聊功能

#### 页面参数:

1. 群聊名
2. 头像
3. 邀请成员列表
4. 确认创建

**调用背景:** 点击主页的创建群聊按钮

#### 页面组成:

1. 上方内容区: 自定义群聊名、自定义群聊头像
2. 下方内容区: 选取成员列表

#### 页面制作示意图:

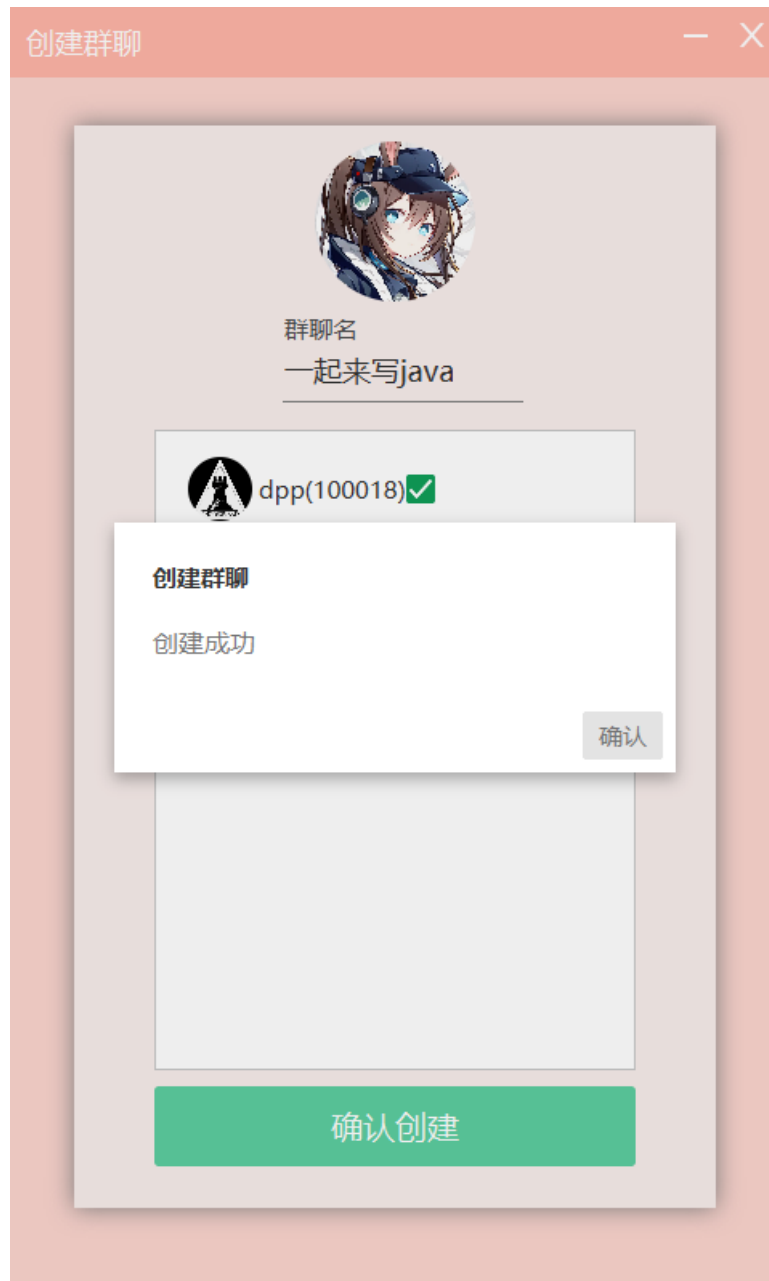


图 3-7 创建群聊界面

### 页面使用简介:

在该页面可以创建群聊，编辑群聊属性并选择成员即可创建。

## 3.6 添加好友

**包含功能模块：** 好友功能

**页面参数：**

1. 搜索 id
2. 用户名

**调用背景：** 点击主页的添加好友按钮

**页面组成：**

1. 搜索栏
2. 查找结果

**页面制作示意图：**



图 3-8 搜索好友界面



图 3-9 好友添加成功界面

页面使用简介：

在该页面可以搜索并添加好友。

4 数据库设计

4.1 数据库表名列表

数据库表名列表格式如表 5-1 所示

表 4-1 数据库表名列表

序号	中文表名	英文表名	表功能说明
----	------	------	-------

1	用户信息	userinfo	用于储存用户 id、用户名、密码和
2	群聊用户	userbelong2group	用于储存用户 id 和所属群聊 id 的关系保存
3	私聊消息	private_message	存储发信人收信人消息内容和发送时间
4	好友关系	froendsrelation	存储好友消息
5	群聊信息	groupinfo	存储群聊 id 和
6	群聊历史消息	groupMessageHistory	存储群聊 id、消息内容、发信人 id、发信人用户名、发送时间

## 4.2 数据库表之间关系说明

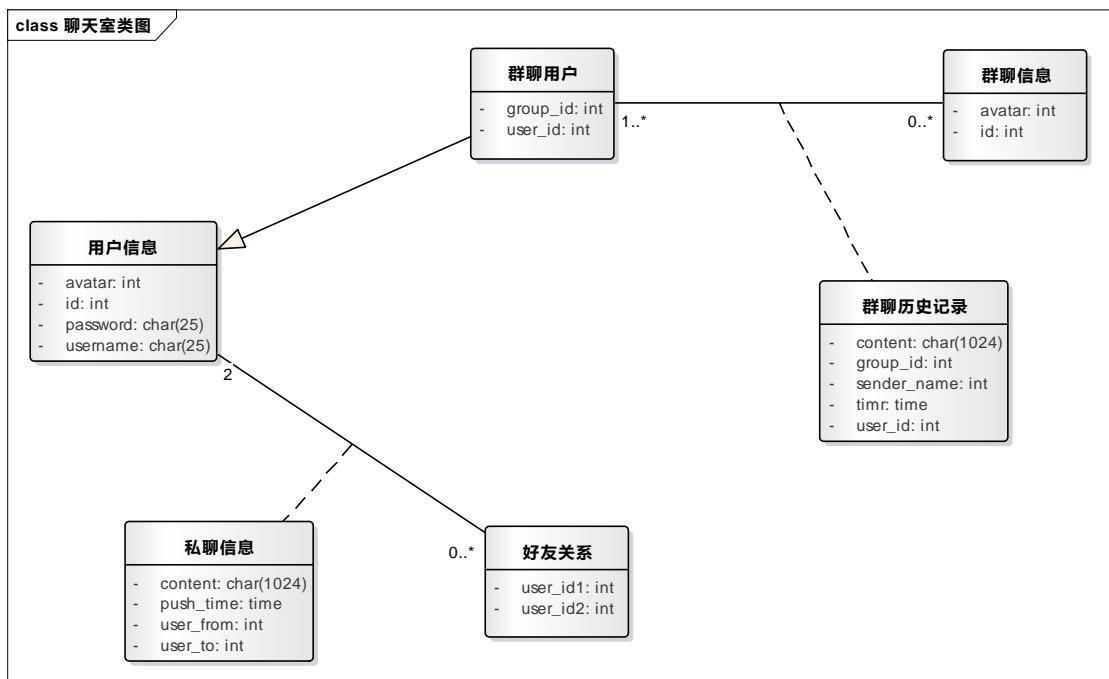


图 4-1 数据库表关系

## 4.3 数据库表的详细清单

表 4-2 用户信息

序号	字段名	数据类型	是否允许为空	说明	主键/外键
1	id	int	否	用户 ID	主键
2	username	varchar(25)	否	用户名	
3	password	varchar(25)	否	用户密码	
4	avatar	int	否	头像 id, 默认为 1	

表 4-3 群聊用户

序号	字段名	数据类型	是否允许为空	说明	主键/外键
1	group_id	int	否	所属群聊 id	外键
2	user_id	int	否	用户 id	外键

表 4-4 私聊信息

序号	字段名	数据类型	是否允许为空	说明	主键/外键
1	user_from	int	否	发信人 id	外键
2	user_to	int	否	收信人 id	外键
3	content	varchar(1024)	否	信息	
4	push_time	time	否	发送时间	

表 4-5 好友关系

序号	字段名	数据类型	是否允许为空	说明	主键/外键
1	user_id1	int	否	好友关系 1	外键
2	user_id2	int	否	好友关系 2	外键

表 4-6 群聊信息

序号	字段名	数据类型	是否允许为空	说明	主键/外键
1	id	int	否	群聊 id	主键
2	avatar	int	否	头像 id, 默认为 1	

**表 4-7 群聊历史记录**

序号	字段名	数据类型	是否允许为空	说明	主键/外键
1	group_id	int	否	群聊 id	外键
2	content	varchar(1024 )	否	消息	
3	user_id	int	否	发信人 id	外键
4	sender_name	varchar(25)	否		
5	time	time	否	发送时间	



## 5 存储过程设计

### 5.1 个人信息管理系统

#### 5.1.1 注册

功能描述：注册账户，如果没有该用户，则插入该用户。

存储过程名称：register

入口参数：username、password、sex、introduce

出口参数：状态码（200，404）+（成功时）注册得到的 uid

#### 5.1.2 登录

功能描述：在完成注册后，或已有注册账号时可以进行登录操作

存储过程名称：login

入口参数：username、password

出口参数：状态码（200,401,402,403）+（成功时）格式化返回好友信息及历史消息

以及群聊消息和历史记录：

start#{uid}&{username}&{avatar}&{sex}&{introduce}#{uid}&{username}&{avator}@@...(好友消息)...#{gid}&{groupname}&{avatar}@@...(群聊消息)...#

#### 5.1.3 加好友

功能描述：在完成登录后，可以进行添加好友操作

存储过程名称：add

入口参数：对方 uid

出口参数：状态码（200,405）

## 5.2 聊天管理系统

### 5.2.1 创建群聊

功能描述：完成登录后可以创建群聊

存储过程名称：cgroup

入口参数：群聊成员 uid 列表

出口参数：成功时：分配的群聊 id；失败时：状态码 406

### 5.2.2 加入群聊

功能描述：完成登录后可以加入已存在群聊

存储过程名称：jgroup

入口参数：群聊 id

出口参数：状态码 (200,407)

## 5.3 消息系统

备注：发送消息时请不要再发送空行。

### 5.3.1 发消息（群聊）

功能描述：已加入群聊的用户可在群聊中发送消息

存储过程名称：G

入口参数：群聊 id、发信人 id、消息

出口参数：将消息以指定格式发给指定群聊

### 5.3.2 发消息（私聊）

功能描述：已登录用户可以给好友私发消息

存储过程：P

入口参数：发信人 uid、收信人 uid、消息

出口参数：将消息以指定格式发给指定好友

### 5.3.3 历史记录（群聊）

功能描述：用户同步已入群聊的历史记录

存储过程：gh

入口参数：群聊 id

出口参数：群聊信息包括 id 和成员信息+历史记录包括群聊 id、发信人 id 和消息等。

### 5.3.4 历史记录（私聊）

功能描述：用户同步私聊的历史记录

存储过程：ph

入口参数：用户 id

出口参数：用户 id、昵称、发送时间和发送消息内容

## 6 系统出错处理设计

### 6.1 出错信息

对程序运行中可能产生的异常进行捕获，同时在收集错误信息后，储存在后端专门的数据库中，便于后期根据错误信息发下系统存在的问题。这可以通过 java 的异常捕获类实现。

### 6.2 系统维护设计

#### 6.2.1 编码规范化

编码实现时应采用模块化和分层的思想，提高模块内部的内聚，减少模块间的耦合。使系统逻辑结构清晰，从而增强可读性和可维护性。

在编码中时刻注意：编写的 UI 页面，应写清楚事件分发、需要的参数、路由等信息。服务端编写的接口，写清楚请求方法、返回数据类型和内容。以上信息作为文档，保持维护，并且上传到 github 和 gitee 仓库，供使用者阅读。

同时在编码过程中注意标识符命名的意义，添加适量注释。

#### 6.2.2 编码模块化

对于可重复部分均采用 class 进行样式的设定，依次增加代码的复用性。

同时将服务端针对不同请求的处理函数等封装在类中，减少代码冗余，使得代码可读性提高。

用界面展现子系统，作为系统的模块化；进一步在界面上划分组件区域，实现子系统的模块化。

而且面向数据与面向对象相结合，模块划分符合面向对象思想。

附注：

小组分工：

季晓东：负责结合数据库和通信信息解析

于添：负责主要的客户端 socket 通信框架

邓浩：负责客户端设计