《数据库系统原理》课程设计

系统实现报告

题目名称： **综合聊天系统**

学号及姓名：13061210，何涛

13061192，赵东方

2015年 12 月 27 日

**一．系统功能需求分析**

**1. 用户信息管理**

综合聊天系统对用户信息的管理，包括登陆，注册和修改用户信息。

登录时，需要提供用户名和密码，若请用户存在，且密码正确，则登陆成功，否则会告诉用户登录失败。

注册时，需要填写昵称，邮箱，密码，以及确认密码，各项不能为空，且密码和确认密码必须一致。注册成功时，会返回一个用户号，这个号是独一无二的，用于之后用户登录名使用，所以需要牢记自己的用户号。

本系统还提供了修改用户信息功能，用户可以修改自己的昵称，邮箱以及密码。修改时需要提供旧密码。若不想修改密码，则新密码那一项为空即可。

**2. 好友关系管理**

综合聊天系统对好友关系的管理，包括查找好友，加好友，删除好友，查询好友列表和查询好友的聊天记录四个功能。

查找好友时，先给用户反馈自己的好友，根据前缀予以显示，若需要进一步查询不是好友的用户，则可点击查找好友，来显示所有用户的列表。

加好友，在查找好友的基础上，选择想要加的用户，即可成为好友。

同时还可以删除好友，删除好友时，两人的聊天记录也会一并删除。

查询好友列表，即显示当前用户的好友列表。

查询好友的聊天记录功能，可以使用户查找聊天记录，便于回忆和记录。

**3. 群组信息管理**

综合聊天系统对群租信息的管理，包括创建群组，查找群组，加入群组，删除群组，查找所在群组，查询群组成员和查询群组聊天记录这些功能。

创建群组时，用户需要提供群组名和群组说明，以便其他用户寻找并加入。创建群组的同时，创建者就相当于加入了群组。

查找群组时，先给用户反馈自己所在的群组，根据前缀予以显示，若需要进一步查询其它群组，则可点击查找更多，来显示群组列表。

加入群组，用户可以加入自己想加入的群组。在查找群组的基础上，选择用户想加入的群组，加入就可成为群组成员。

同时本系统还提供了删除群组的功能。不是此群组创建者若删除群组，只是退出群组；但若是此群创建者选择删除群组，则会解散群组，并删除与之相关的群组成员信息和群组聊天记录。

查找所在群组，即显示当前用户的群组列表，分为自己创建的群组和加入的群组两类。

查询群组成员，即显示当前群组的成员列表。

查询群组的聊天记录功能，可以使用户查找群组聊天记录，避免在用户不在线的时候，错过重要的消息。

**4. 聊天室信息管理**

综合聊天系统对聊天室的管理，包括查询聊天室和查看聊天室成员这两个功能。用户不能创建聊天室，由系统设定。用户可随意进出任何聊天室，但同时只能在一个聊天室进行聊天。

查询聊天室，即显示所有聊天室的列表，便于用户加入聊天室进行聊天。

查看聊天室成员，可以使用户看到当前聊天室有谁在线。

**5. 留言管理**

综合聊天系统对留言的管理，包括发布留言和查看留言两个功能。留言会发布在留言板上，每个聊天室有且只有一个留言板，用户可以在相应留言板上进行留言。因为聊天室是不保存聊天记录的，所以本系统提供留言板，来方便用户发布一些信息和通知给关注相应聊天室的用户。

发布留言，用户可以在对应聊天室的留言板上进行留言，留言将会被任何人看到，且不可删除。

查看留言，即查看对应聊天室的留言板上的留言。

**6. 消息处理**

本综合聊天系统的核心，就是聊天。用户有三种方式进行聊天，一是好友与好友之间的聊天，二是在群组中聊天，三是在聊天室中聊天。其中好友之间和群组的聊天记录会被保存下来，并且能够查看，而聊天室的不会被保存。

聊天，即发送消息。用户发送消息时，消息的内容，时间，以及发送者这些信息会被保存在数据库中。

## 二．系统功能结构设计

**三、数据库基本表的定义**

**1. 用户表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 是否可为空值 | 说明及约束 |
| userid | int(11) | 否 | 用户号，主键，auto\_increment |
| nickname | varchar(30) | 否 | 昵称 |
| passwd | varchar(32) | 否 | 密码 |
| email | varchar(20) | 否 | 邮箱 |

**2. 群组表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 是否可为空值 | 说明及约束 |
| groupid | int(11) | 否 | 群组号，主键，auto\_increment |
| groupname | varchar(30) | 否 | 群组名 |
| admin | varchar(10) | 否 | 管理员 |
| decription | varchar(512) | 是 | 群组说明 |

**3. 消息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 是否可为空值 | 说明及约束 |
| messageid | int(11) | 否 | 消息号，主键，auto\_increment |
| ctime | timestamp | 否 | 发送时间，自动生成 |
| content | varchar(512) | 否 | 消息内容 |
| userid | int(11) | 否 | 发送者用户号 |
| state | tinyint(1) | 否 | 消息状态 |

**4. 留言表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 是否可为空值 | 说明及约束 |
| noteid | int(11) | 否 | 留言号，主键，auto\_increment |
| ctime | timestamp | 否 | 发送时间，自动生成 |
| content | varchar(512) | 否 | 留言内容 |
| userid | int(11) | 否 | 发送者用户号 |

**5. 聊天室表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 是否可为空值 | 说明及约束 |
| roomid | int(11) | 否 | 聊天室号，主键，auto\_increment |
| roomname | varchar(30) | 否 | 聊天室名 |
| description | varchar(512) | 是 | 聊天室说明 |

**6. 好友关系表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 是否可为空值 | 说明及约束 |
| useraid | int(11) | 否 | 用户A，主键 |
| userbid | int(11) | 否 | 用户B，主键 |
| state | tinyint(1) | 否 | 好友验证状态 |

**7. 聊天室状态表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 是否可为空值 | 说明及约束 |
| roomid | int(11) | 否 | 聊天室号，主键 |
| userid | int(11) | 否 | 用户号，主键 |

**8. 聊天记录表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 是否可为空值 | 说明及约束 |
| useraid | int(11) | 否 | 用户A，主键 |
| userbid | int(11) | 否 | 用户B，主键 |
| messageid | int(11) | 否 | 消息号，主键 |

**9. 群组归属表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 是否可为空值 | 说明及约束 |
| groupid | int(11) | 否 | 群组号，主键 |
| userid | int(11) | 否 | 用户号，主键 |

**10. 群组聊天记录表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 是否可为空值 | 说明及约束 |
| groupid | int(11) | 否 | 群组号，主键 |
| messageid | int(11) | 否 | 消息号，主键 |

**11. 留言归属表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 是否可为空值 | 说明及约束 |
| boardid | int(11) | 否 | 留言板号，主键 |
| noteid | int(11) | 否 | 留言号，主键 |

**12. 留言板归属表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 是否可为空值 | 说明及约束 |
| boardid | int(11) | 否 | 留言板号，主键 |
| roomid | int(11) | 否 | 聊天室号，主键 |

**四、触发器的定义与实现**

**1. add\_group**

设置触发器，当创建群组操作发生时，会将创建者加入到group\_belong中，表示将创建者加入群组中。

CREATE TRIGGER add\_group

AFTER INSERT ON group\_info

FOR EACH ROW

insert into group\_belong (groupid, userid) values (new.groupid, new.admin);

**2. add\_noteboard**

设置触发器，当聊天室创建时，会随之建立一个留言板，并将留言榜与此聊天室绑定。

CREATE TRIGGER add\_noteboard

AFTER INSERT ON chatroom

FOR EACH ROW

insert into board\_belong (roomid) values (new.roomid);

**3. del\_friend**

设置触发器，当删除好友操作发生时，同时也会将二人的聊天记录一并删除。

CREATE TRIGGER del\_friend

AFTER DELETE ON friend

FOR EACH ROW

DELETE FROM chat\_record WHERE chat\_record.useraid = old.useraid AND chat\_record.userbid = old.userbid;

**4. del\_group**

设置触发器，当创建群组者，即管理员，删除群组时，同时会将此群组的聊天记录和此群租的成员信息一并删除。

CREATE TRIGGER del\_group

AFTER DELETE ON group\_info

FOR EACH ROW

BEGIN

DELETE FROM group\_belong WHERE group\_belong.groupid = old.groupid;

DELETE FROM group\_record WHERE group\_record.groupid = old.groupid;

END;

**五、存储过程的定义与实现**

**1. add\_user**

当创建新用户时，先将信息插入到数据库中，然后返回用户ID

CREATE PROCEDURE add\_user(IN nickname\_user varchar(30),IN passwd\_user varchar(32), IN email\_user varchar(20))

BEGIN

INSERT INTO user\_info(nickname, passwd, email) VALUES (nickname\_user, passwd\_user, email\_user);

SELECT userid FROM user\_info ORDER BY userid DESC LIMIT 1;

END;

**2. add\_message**

当创建新消息时，先将消息插入到数据库中，然后返回消息ID

CREATE PROCEDURE add\_message(IN content\_message varchar(512), IN userid\_message int)

BEGIN

INSERT INTO message(content, userid, state) VALUES (content\_message, userid\_message, false);

SELECT messageid, ctime FROM message ORDER BY messageid DESC LIMIT 1;

END;

**3. add\_note**

当创建新留言时，先将留言插入到数据库中，然后返回留言ID

CREATE PROCEDURE add\_note(IN content\_note varchar(512), IN userid\_note int)

BEGIN

INSERT INTO note(content, userid) VALUES (content\_note, userid\_note);

SELECT noteid FROM note ORDER BY noteid DESC LIMIT 1;

END;

**六、系统实现结果**

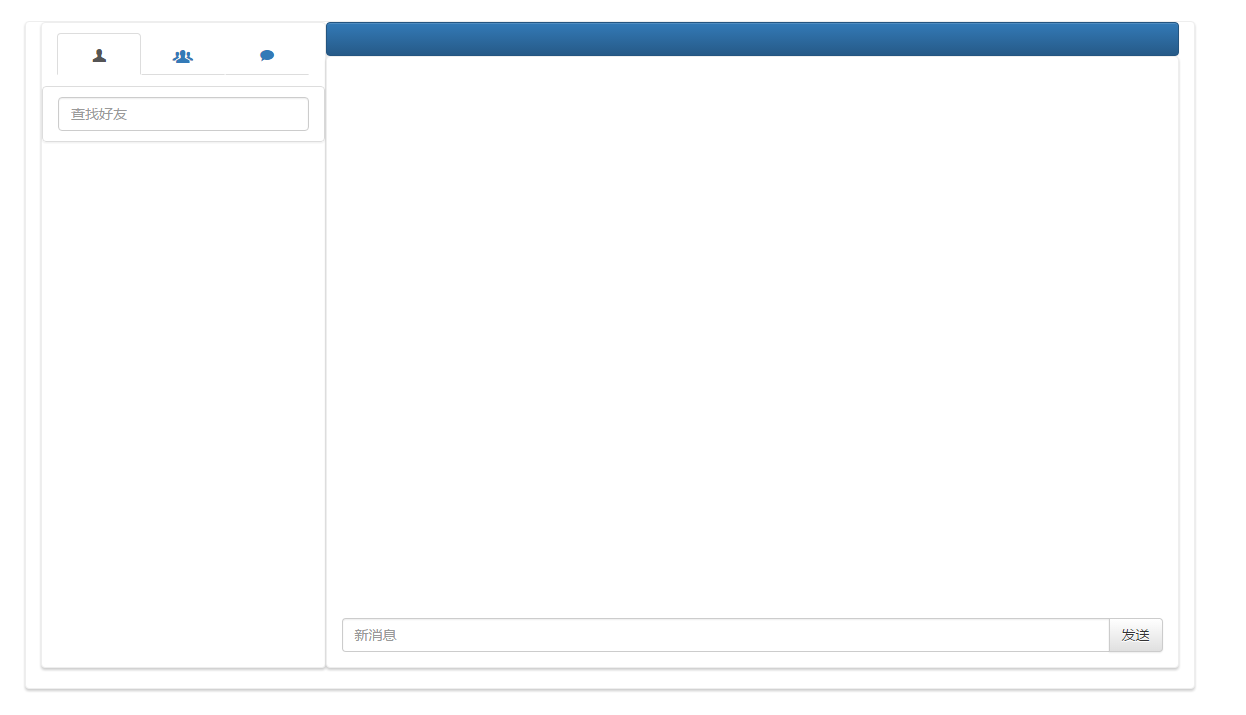
**1. 登录界面**



**2. 注册界面**

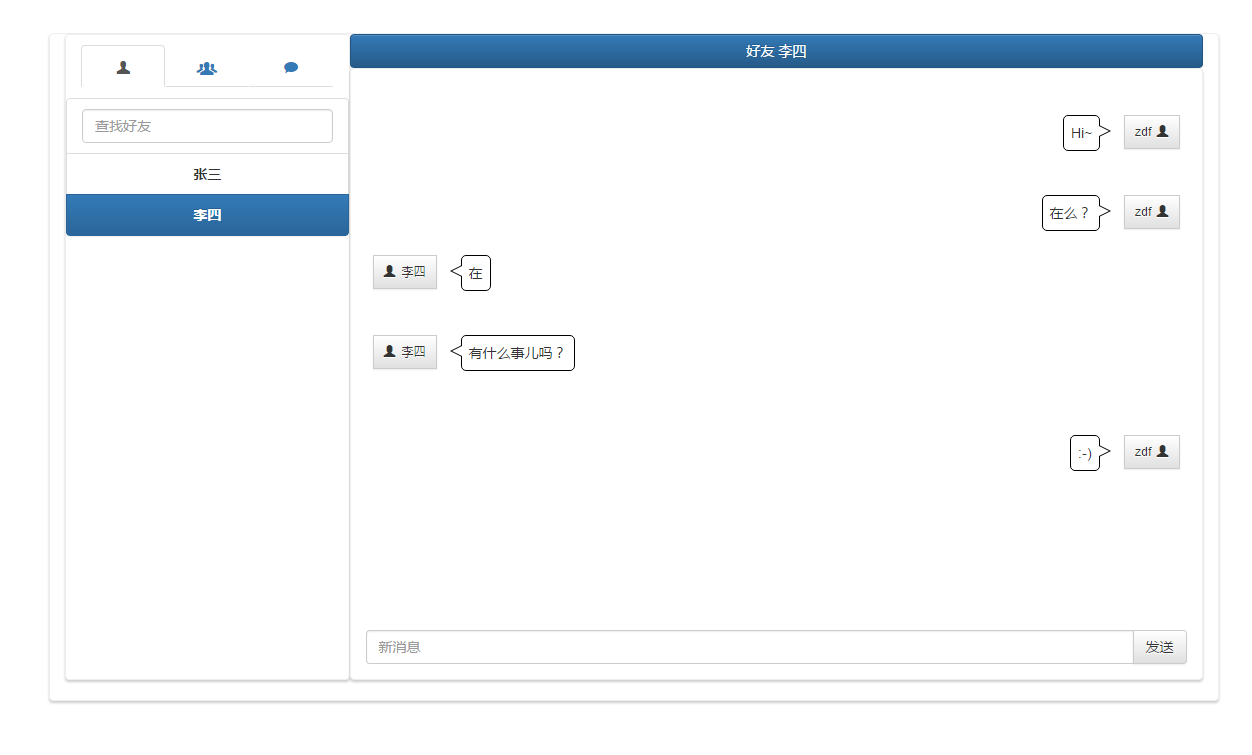


**3. 主界面**

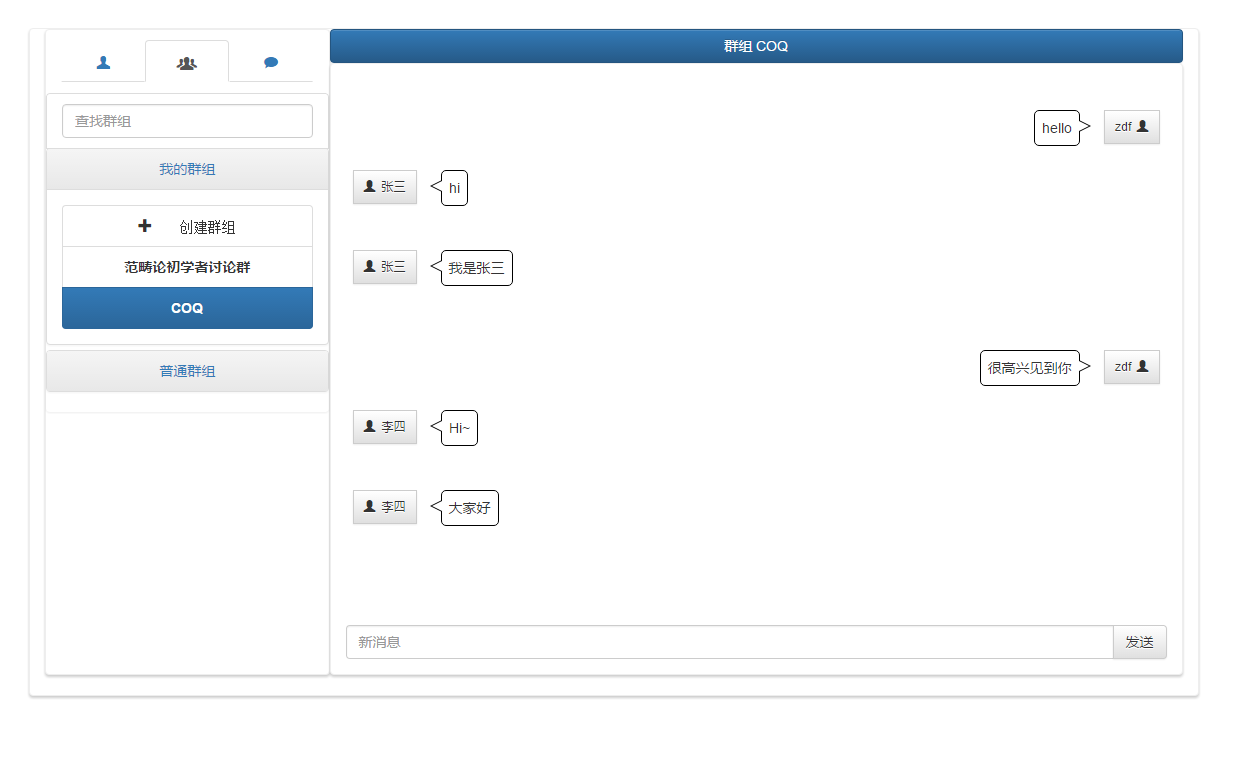


**4. 好友聊天界面**

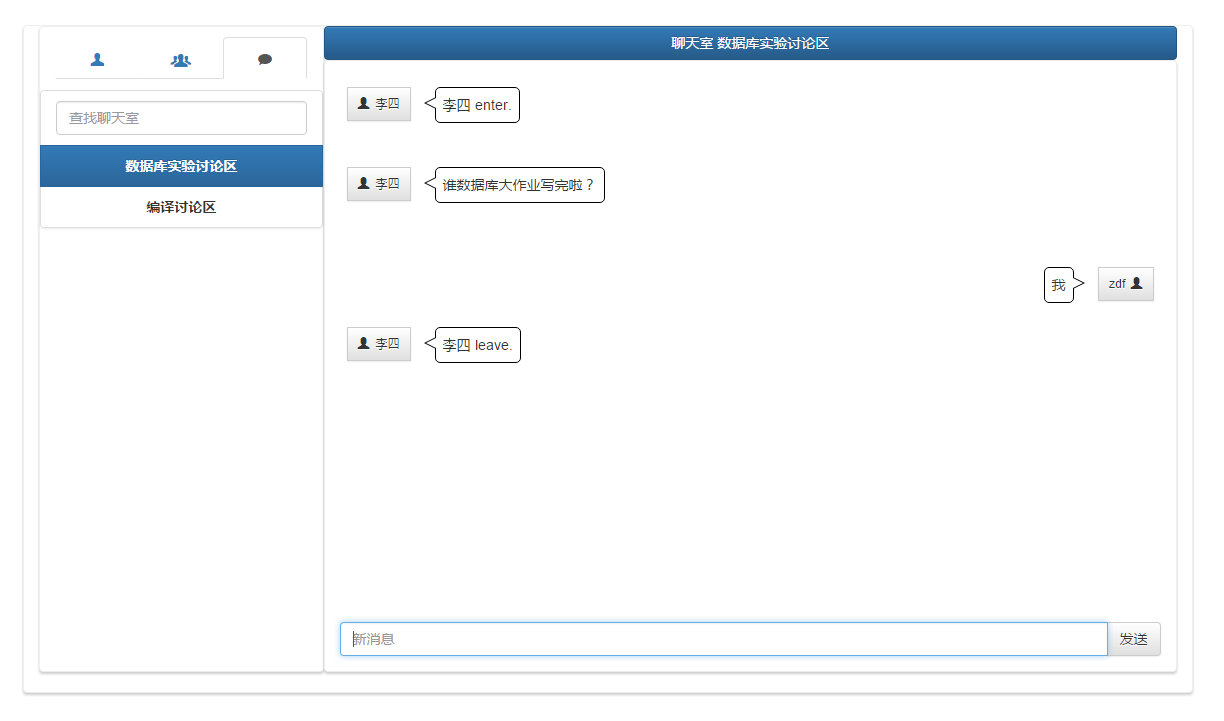




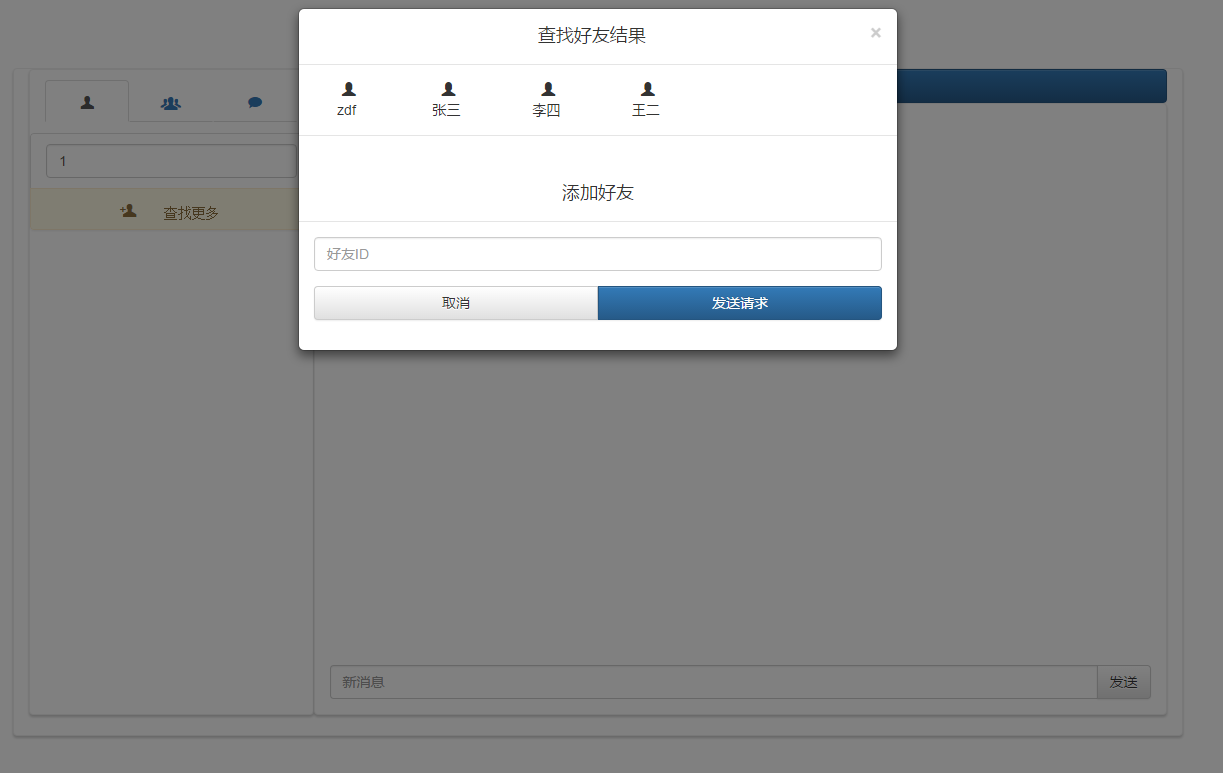
**5. 群组聊天界面**



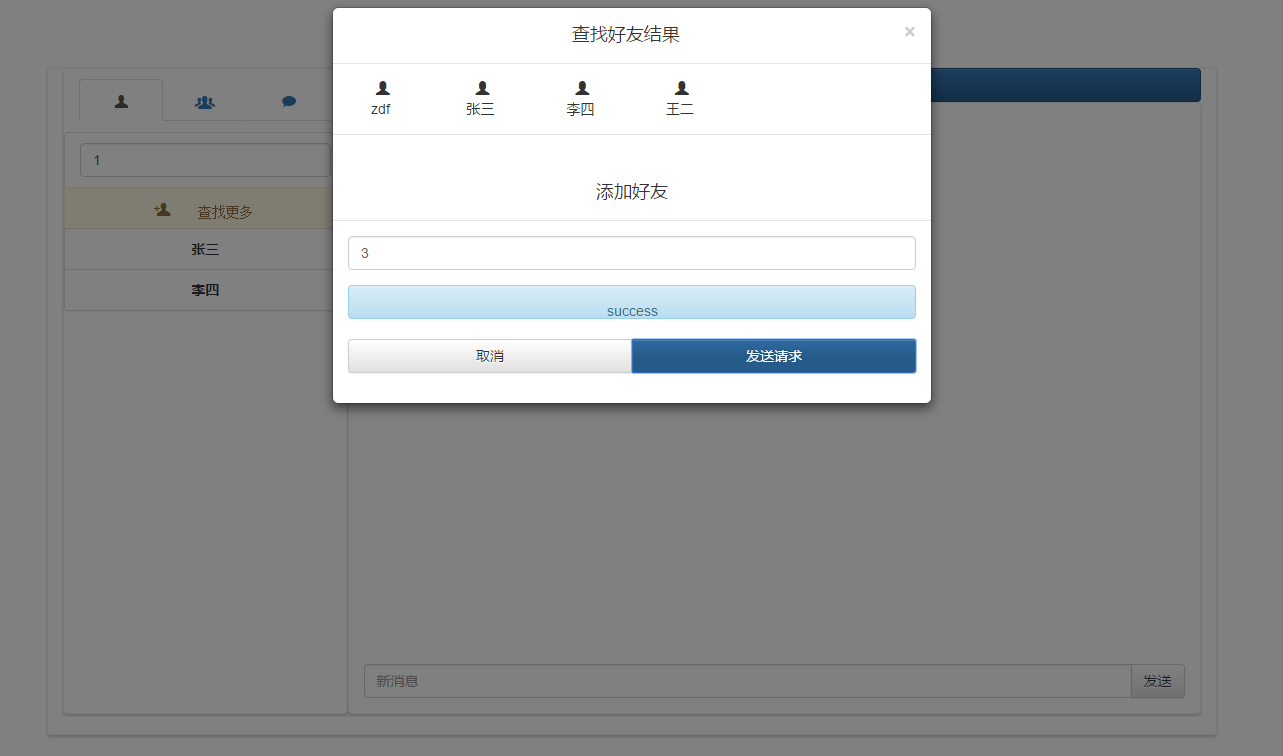
**6. 聊天室聊天界面**



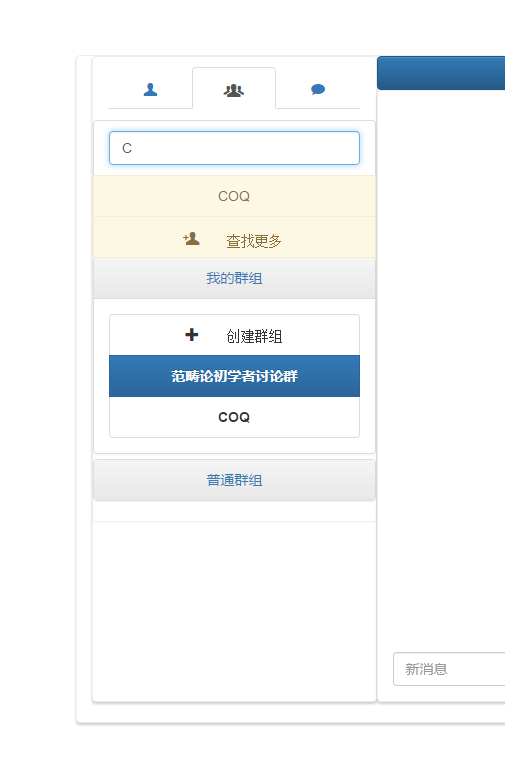
**7. 查找好友**



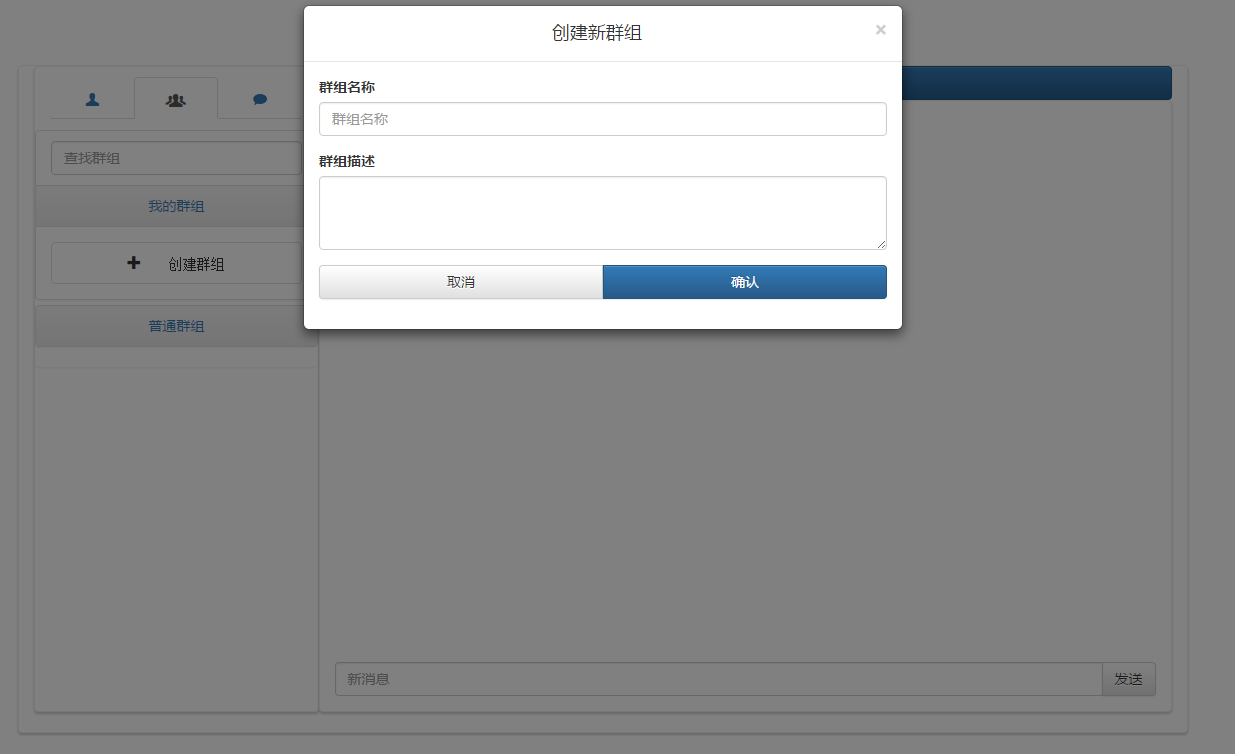
**8. 添加好友**

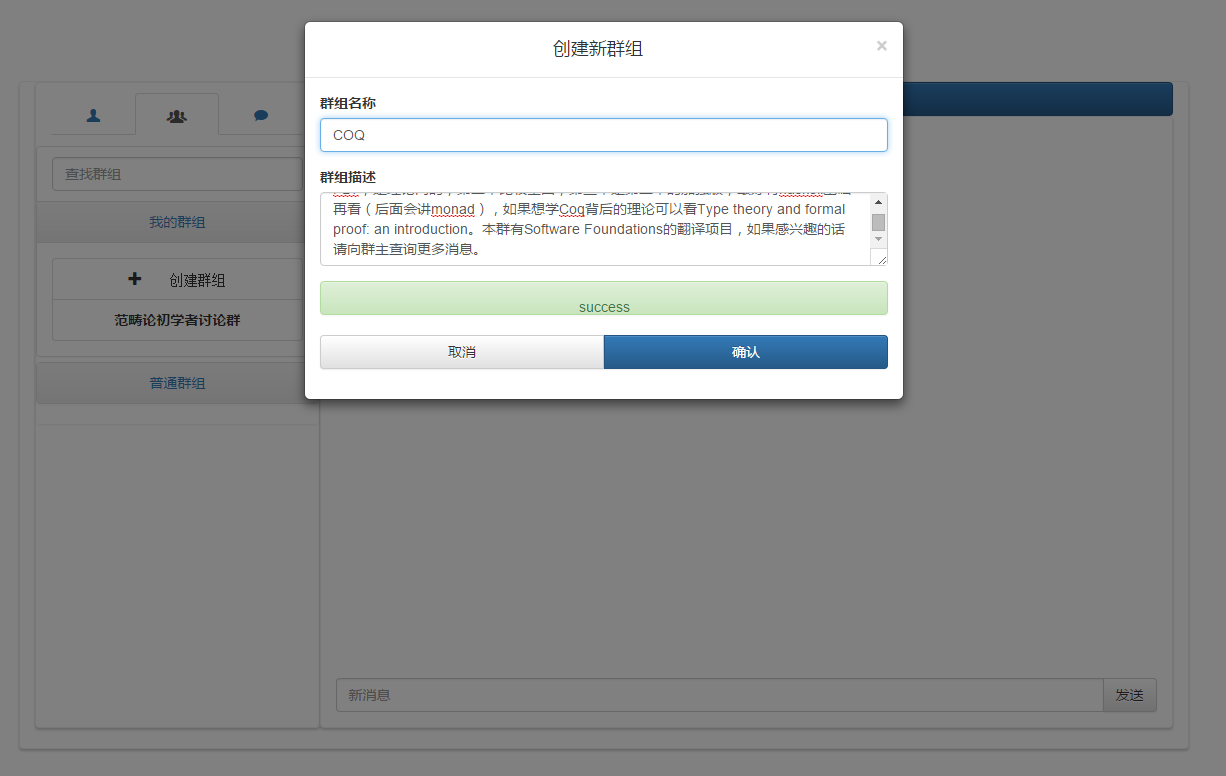


**9. 查找群组**

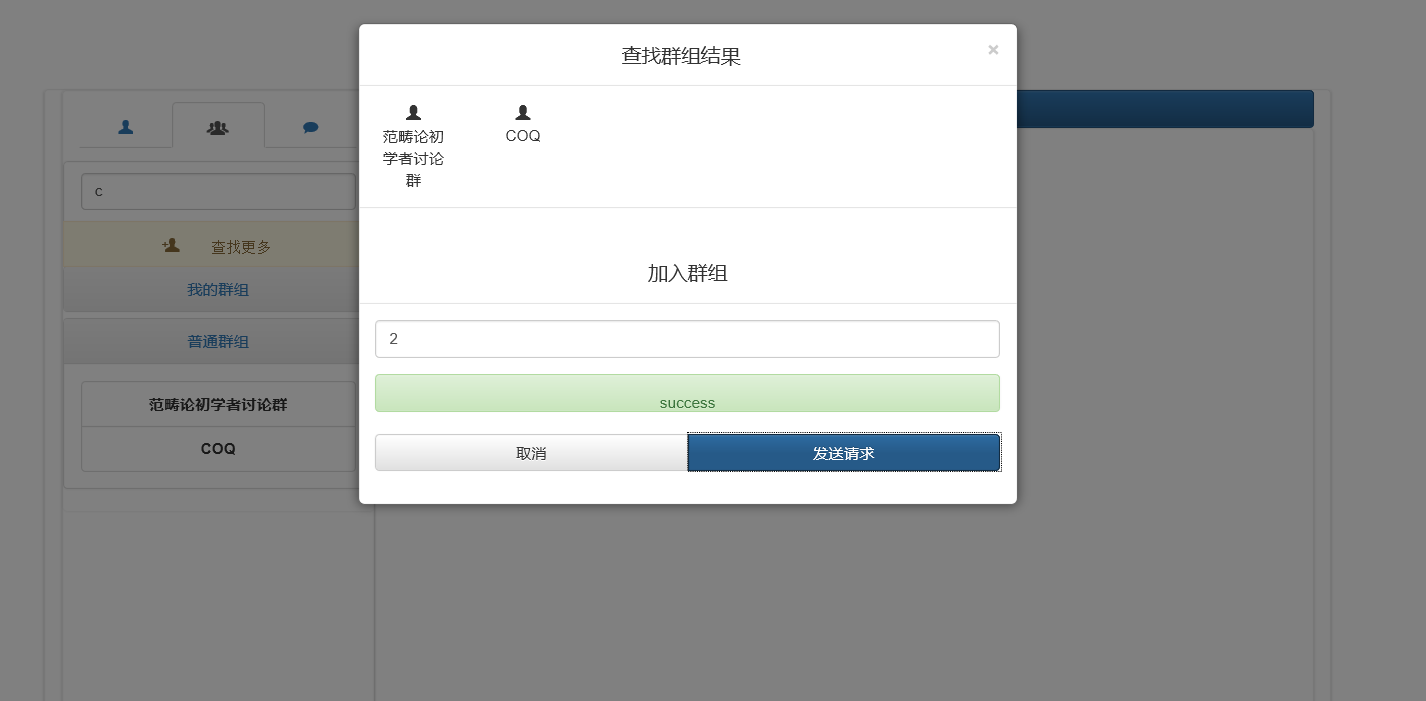


**10. 创建群组**

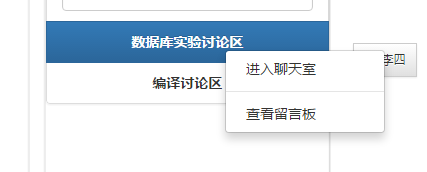


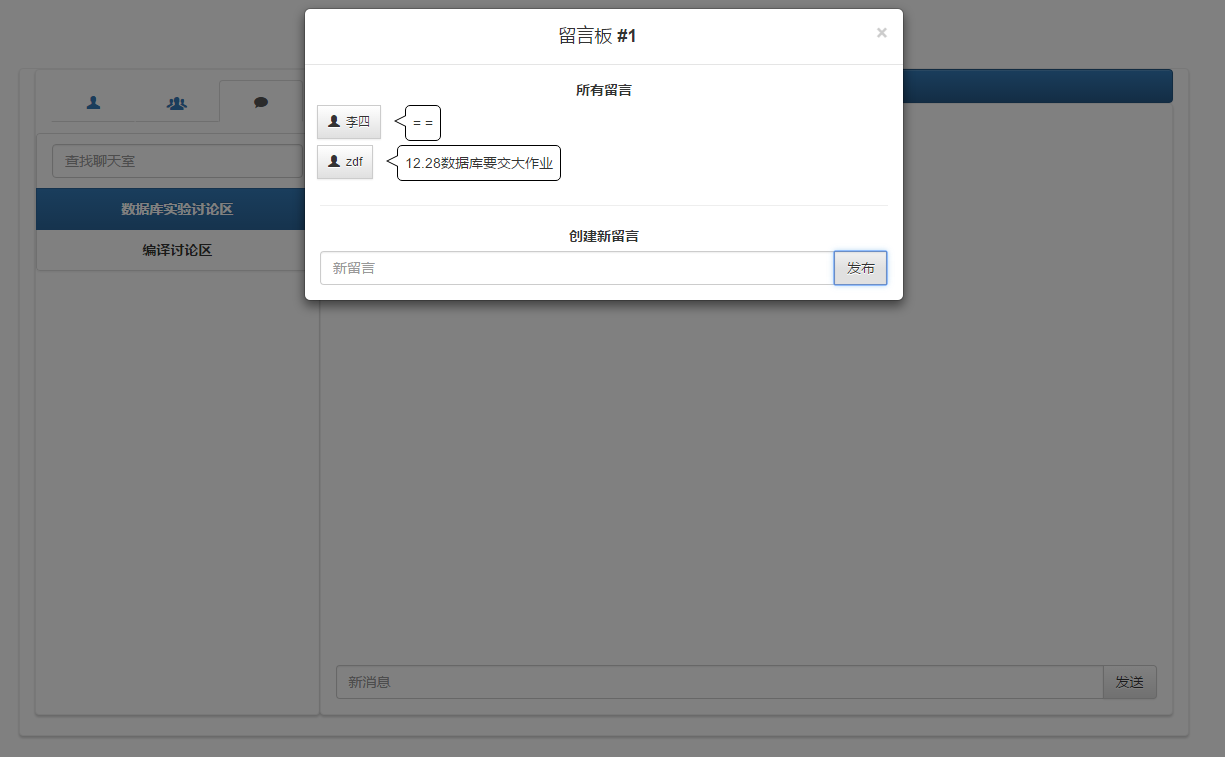


**11. 加入群组**



**12. 查看留言板+发布留言**





**七、总结**

综合聊天系统是一个实用的聊天系统，它除了可以进行好友与好友之间的聊天和群组聊天之外，还能够进行聊天室聊天。聊天室聊天的方式虽然并不新颖，但却是一个有趣的聊天方式。我们可以使用本综合聊天系统，建立自己的聊天圈子。

经过近两个星期的努力，我们最终实现这个综合聊天系统。实现的过程并不是一帆风顺的，但却在过程中充满了快乐，并且收获了知识与经验。

这个综合聊天系统是一个web应用，后台是由Java实现，前台是由angular.js搭建完成的。我们通过实现这个系统学习了mysql的操作，包括触发器和存储过程的建立；学习了java的servlet和web socket来实现网页与服务器之间的交互。同时还丰富了我们的团队合作经验和项目开发的经验。