序号\_\_\_\_\_

**华南理工大学广州学院**

**2020-2021学年度第1学期**

**课程名称： Java Web应用开发项目实践**

**题　　目： 简易论坛**

**专业班级： 2018级**

**姓　　名： 黄泽辉**

**学　　号：** 201810089286

|  |  |  |  |
| --- | --- | --- | --- |
| 一级指标 | 二级指标 | 分值 | 评分及成绩 |
| 得分 |
| 作品完成度  （占50%） | 1.工作量达标，功能点完整，独立完成 | 50 |  |
| 2.设计合理性，逻辑条理性 |
| 3.功能测试 |
| 4.数据处理水平 |
| 5.特色鲜明，有设计亮点 |
| 文档撰写（30%） | 1.文字描述规范，语言通顺，结构完整 | 30 |  |
| 2.思路描述清晰，功能测试结果完整 |
| 3.结果分析合理 |
| 答辩成绩 （占20%） | 1.能够对设计进行合理说明 | 20 |  |
| 2.能较好的回答答辩所提问题，解释合理清晰 |
| 合计（百分制） | | |  |
| 总 评 | □优 □良 □中 □及格 □不及格 | | 签名 |

1. 项目概述
2. 开发环境及安装配置

操作系统: windows 10 64位

服务器：tomcat 9

IDE环境:NETBEANS 12.0

项目构建工具：maven

所用技术:javaBean+servlet+JSP

项目部署方式：netbeans自动部署

1. 项目的需求设计和分析
2. 需求分析

作为一个论坛，最基本要有发帖和回帖的功能，然后是管理功能，根据不同的权限，控制用户和管理者对论坛的操作权限，用户信息管理和消息收发机制也是基本需求之一，在实现基本功能的前提下，可以增加各种的功能。

1. 大概设计：

登录注册、用户信息的增删改查功能；

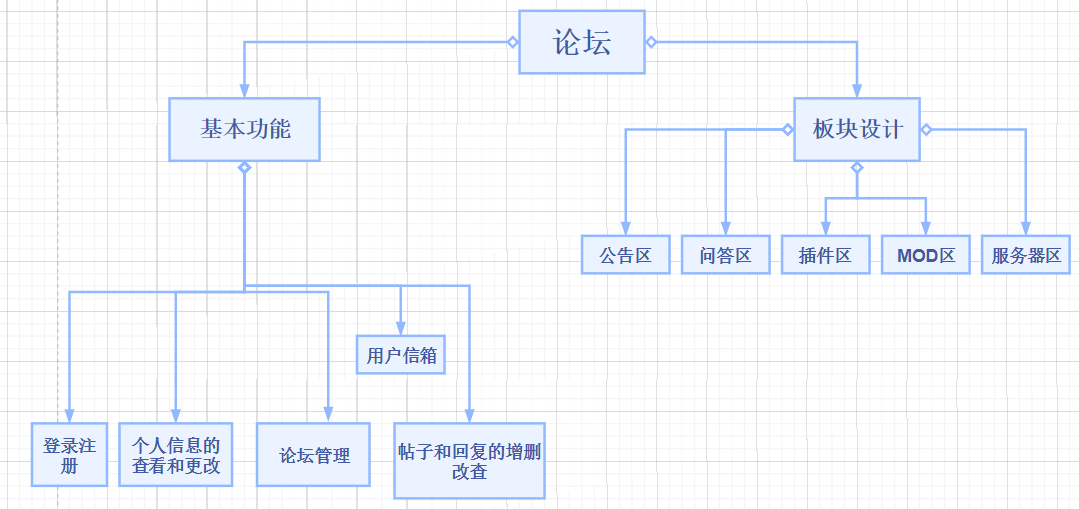
版主和超管的权限限制和对论坛的管理功能；

对帖子和回复的基本增删改查功能；

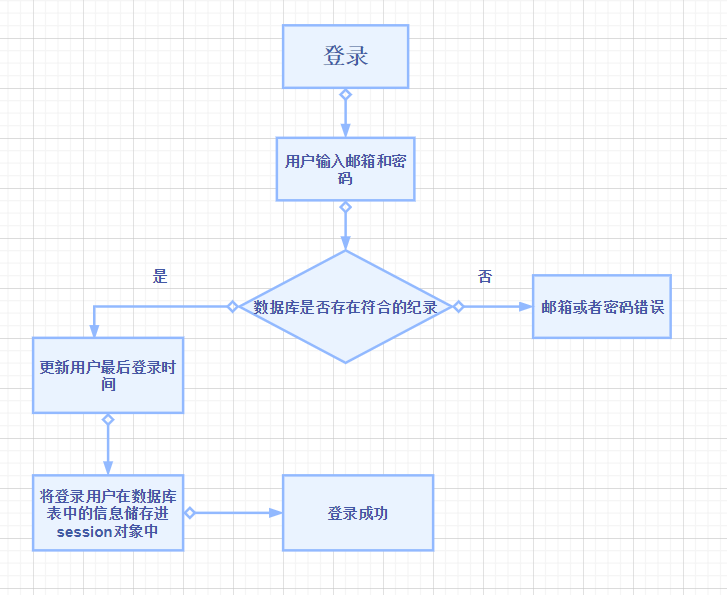
简易消息收发机制。

1. 程序概要设计
2. 程序的功能模块

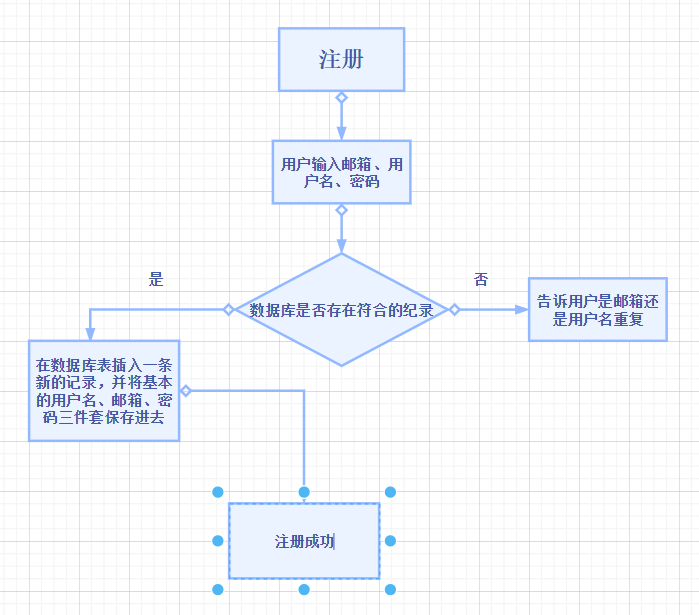
程序的功能模块如下图所示：



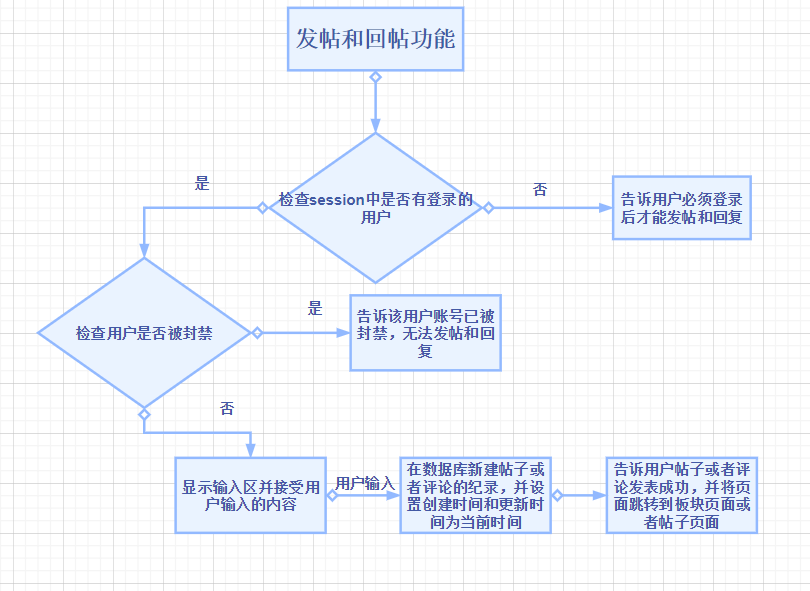
1. 程序的流程
2. **登录功能**



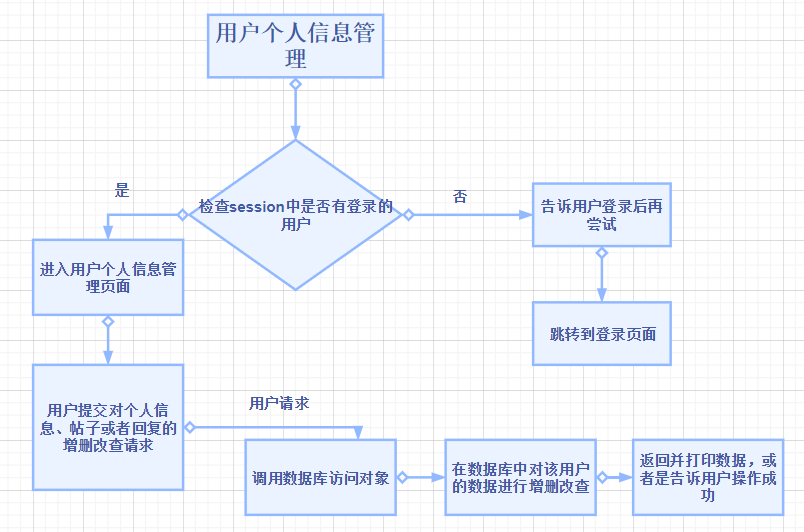
1. **注册功能**



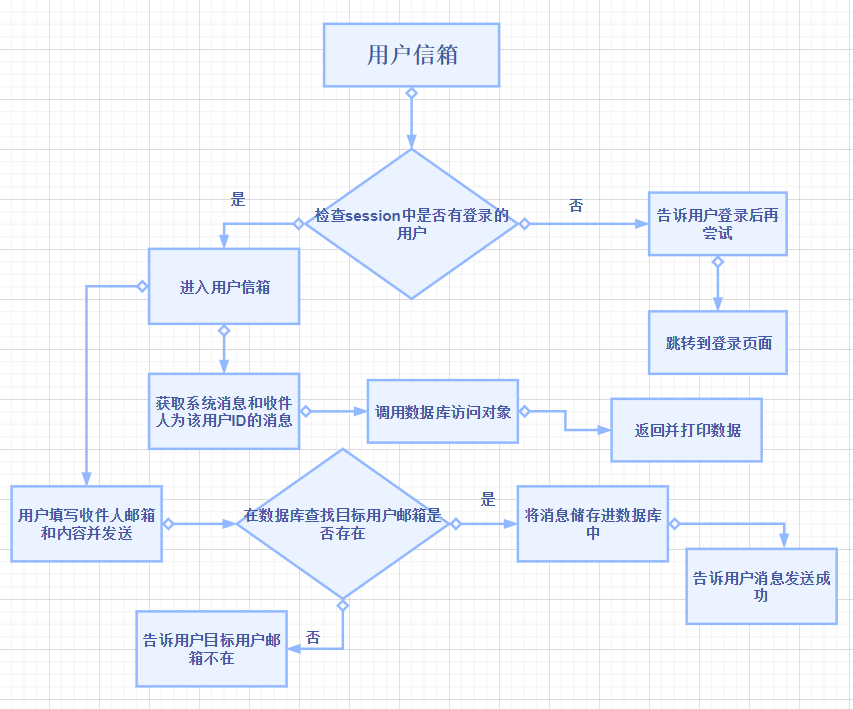
1. **发帖和回帖功能**



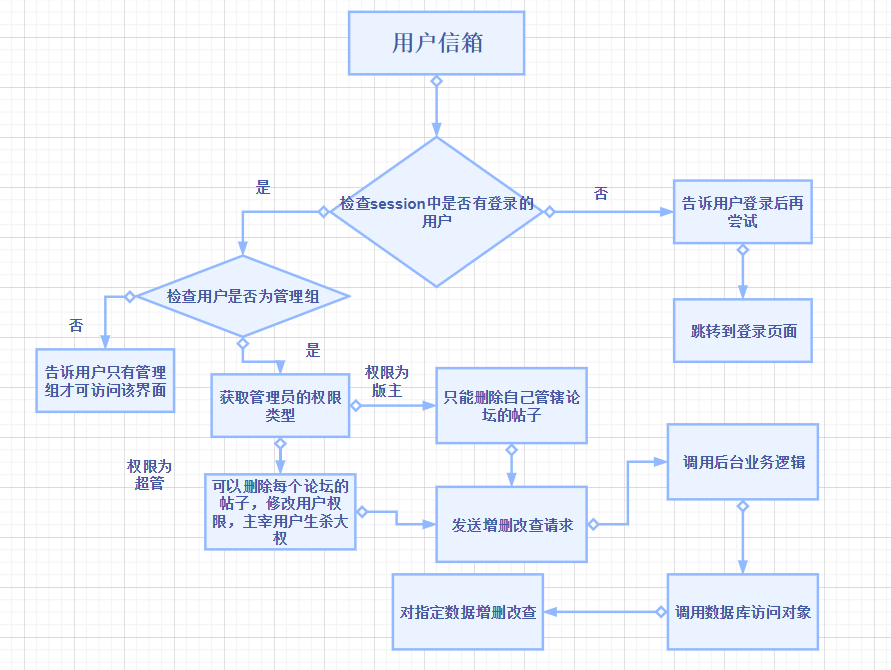
1. **用户个人信息管理**



1. **用户信箱**



1. **论坛权限管理**



1. 程序的文件结构分析

大文件夹列表：WebPages、Source Packages、Test Packages、Dependencies、

Java Depencies、Project Files

1. WebPages:

主要功能：储存项目的配置信息、静态资源、网页文件

子文件夹：META-INF、WEB-INF、static

META-INF：主要文件为context.xml，保存项目的路径和环境变量配置

WEB-INF：主要文件为web.xml，保存项目的基本信息、servlet的配置、session 和cookie等设置

Static：保存项目的静态资源包括图片资源、样式资源、脚本资源等

1. Source Packages：

主要功能：存放项目的逻辑代码，包括bean、dao、servlet

子文件夹：JDBC\_connector、bean、dao、servlet

JDBC\_connector：封装了JDBC连接数据库的类的代码

Bean：存放数据库表的实体对象类

Dao：存放数据库访问对象类

Servlet:存放项目业务逻辑的servlet文件类

1. Test Packages

主要功能：存放项目的测试类文件

1. Dependencies

主要功能：存放项目需要调用的jar包，maven构建项目会根据pom.xml文件 自动下载jar包到此文件夹，省去用户到处找jar包的时间

1. Java Depencies

主要功能：是JDK 包中的jar文件夹

1. Project Files

主要功能：存放maven的配置文件pom.xml，用户需要下载什么jar包，直接在该文件里面添加jar包的描述即可，然后Maven就会自动下载对应的jar包

（4）数据库设计

# 数据库名: bbs\_db

## 表格: manager

备注: 管理组

### 索引:

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 其他 | 备注 |
| **主索引** | Id | unique |  |
| uid | uid | unique |  |

### 字段:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 空 | 默认值 | 其他 | 备注 |
| **Id** | int(11) | 否 | <auto\_increment> |  |  |
| uid | int(11) | 否 | 0 |  |  |
| job | int(11) | 是 | <空> |  | 权限组，0是超管，大于0是管理的板块的id |

## 表格: message

备注: 消息

### 索引:

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 其他 | 备注 |
| **主索引** | Id | unique |  |

### 字段:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 空 | 默认值 | 其他 | 备注 |
| **Id** | bigint(20) | 否 | <auto\_increment> |  | 消息id，无意义 |
| sender\_id | int(11) | 是 | <空> |  | 发送者id,0为系统，大于0为用户id |
| receiver\_id | int(11) | 是 | <空> |  | 收件人id，0为所有人，大于0为用户id |
| type | int(2) | 是 | <空> |  | 消息类型，0是系统消息，1是帖子回复通知，2是点赞通知,3为私信 |
| content | text | 是 |  |  | 消息内容 |
| create\_time | datetime | 是 | <空> |  | 消息发送的时间 |

## 表格: reply

备注: 回复

### 索引:

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 其他 | 备注 |
| **主索引** | Id | unique |  |

### 字段:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 空 | 默认值 | 其他 | 备注 |
| **Id** | bigint(20) | 否 | <auto\_increment> |  | 无意义 |
| topic\_id | int(11) | 是 | <空> |  | 帖子id |
| reply\_uid | int(11) | 是 | <空> |  | 回复人的用户id |
| content | text | 是 |  |  | 回复内容 |
| create\_time | datetime | 是 | <空> |  | 回帖时间 |

## 表格: tab

备注: 板块

### 索引:

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 其他 | 备注 |
| **主索引** | Id | unique |  |

### 字段:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 空 | 默认值 | 其他 | 备注 |
| **Id** | int(11) | 否 | <auto\_increment> |  |  |
| tab\_name | varchar(10) | 是 | <空> |  | 板块名 |
| tab\_name\_en | varchar(20) | 是 | <空> |  | 板块的英文名 |

### 数据:

|  |  |  |
| --- | --- | --- |
| **Id** | tab\_name | tab\_name\_en |
| **1** | 公告区 | notice |
| **2** | 问答区 | FAQs |
| **3** | MOD区 | MOD |
| **4** | 插件区 | plugin |
| **5** | 服务器区 | server |

## 表格: topic

备注: 帖子

### 索引:

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 其他 | 备注 |
| **主索引** | Id | unique |  |

### 字段:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 空 | 默认值 | 其他 | 备注 |
| **Id** | int(11) | 否 | <auto\_increment> |  |  |
| user\_id | int(11) | 是 | <空> |  | 发帖用户id |
| create\_time | datetime | 是 | <空> |  | 发帖时间 |
| update\_time | datetime | 是 | <空> |  | 更新时间 |
| title | varchar(30) | 是 | <空> |  | 帖子标题 |
| tab\_id | int(2) | 是 | <空> |  | 板块id |
| content | text | 是 |  |  | 帖子内容 |
| clicks | int(11) | 否 | 0 |  | 点击量 |
| like | int(11) | 否 | 0 |  | 点赞数 |
| reply\_num | int(11) | 否 | 0 |  | 回复数量 |

## 表格: users

备注: 用户表

### 索引:

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 其他 | 备注 |
| **主索引** | Id | unique |  |
| email | email | unique |  |

### 字段:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 空 | 默认值 | 其他 | 备注 |
| **Id** | int(11) | 否 | <auto\_increment> |  |  |
| username | varchar(30) | 否 | UnkownUser |  | 用户名 |
| password | varchar(20) | 是 | <空> |  | 密码 |
| group | int(2) | 是 | 0 |  | 权限，0普通用户，1管理组 |
| email | varchar(30) | 是 | <空> |  | 邮箱 |
| phone | varchar(12) | 是 | <空> |  | 电话 |
| qq | varchar(20) | 是 | <空> |  | QQ号码 |
| sex | int(1) | 是 | 0 |  | 性别，0秘密，1为男，2为女 |
| birth | datetime | 是 | <空> |  | 生日 |
| avatar | varchar(100) | 是 | static/avatar/default/default.png |  | 用户头像的路径 |
| create\_time | datetime | 是 | <空> |  | 账号创建时间 |
| update\_time | datetime | 是 | <空> |  | 上次登录时间 |
| is\_banned | int(1) | 是 | 0 |  | 用户是否被封禁，0为否，1为被封禁 |

1. 程序详细设计
2. 关键代码分析

**登录模块：**

users user = new users();

user.setEmail(request.getParameter("email"));

user.setPassword(request.getParameter("password"));

login\_dao loginDAO = new login\_dao();

isTrue = login\_dao.isTrue(user);

if (isTrue == true) {

/\*获取用户个人信息\*/

user\_dao userDAO = new user\_dao();

user = userDAO.GetUserInfoByEmail(user.getEmail());

/\*将登录的用户信息保存进session\*/

request.getSession().setAttribute("users",user);

response.sendRedirect("index.jsp");

}

else{

response.sendRedirect("login.jsp?loginstatus=fail");

}

代码说明:从表单获取用户输入的邮箱和密码，并通过数据对象访问数据库获取登录结果，如果登录成功，将用户信息储存进session对象中，并跳转页面

**注册模块：**

users user = new users();

user.setUsername(request.getParameter("username"));

user.setEmail(request.getParameter("email"));

user.setPassword(request.getParameter("password"));

int status = -1;

int emailExist = 0;

int usernameExist = -1;

user\_dao userDAO = new user\_dao();

try {

status = userDAO.NewUser(user);

} catch (SQLException sQLException) {

}

if (status == usernameExist) {

response.sendRedirect("register.jsp?status=usernameExist");

} else if (status == emailExist) {

response.sendRedirect("register.jsp?status=emailExist");

}

out.println("注册成功！3秒后跳转到首页");

response.setHeader("refresh", "3;URL =index.jsp");

代码说明:从表单获取用户输入的邮箱、用户名、密码，并且在用户表中查找用户的邮箱和用户名的纪录，如果存在纪录，说明邮箱或者是用户名已经被人注册，并把状态通过URL传值的方式传到注册界面。

**回复模块（发帖模块差不多和功能一样不过少了自动消息发送功能）：**

/\*获取表单值\*/

String topicId = request.getParameter("topicId");

String content = request.getParameter("content");

/\*获取用户信息\*/

users user = (users) request.getSession().getAttribute("users");

if (user == null) {

out.println("无法获取您的用户信息，请登录。3秒后页面将跳转");

response.setHeader("refresh", "3;URL =index.jsp");

}

/\*获取当前登录的用户ID\*/

int uid = user.getId();

/\*创建新的回复\*/

reply new\_reply = new reply();

new\_reply.setReply\_uid(uid);

new\_reply.setContent(content);

new\_reply.setTopic\_id(topicId);

reply\_dao replyDAO = new reply\_dao();

try {

replyDAO.NewReply(new\_reply);

} catch (SQLException sqle) {

}

/\*自动发送消息告诉楼主有人回帖,但是如果回帖的人是楼主本人，就不要发送消息\*/

/\*获取帖子信息\*/

topic this\_topic = new topic();

topic\_dao topicDAO = new topic\_dao();

try {

this\_topic = topicDAO.getTopicInfoById(topicId);

} catch (SQLException sqle) {

}

/\*判断回复的人是不是楼主本人\*/

if (user.getId() != this\_topic.getUser\_id()) {

/\*消息类型\*/

int m\_reply = 1;/\*回复通知\*/

/\*消息发送者:系统\*/

int m\_sender = 0;

/\*消息内容\*/

String message\_content = "用户<a href=\"userInfo.jsp?uid=" + user.getId() + "\">";

message\_content = message\_content + "" + user.getUsername() + "(uid:" + user.getId() + ")";

message\_content = message\_content + "</a>评论了你的帖子:<br/>";

message\_content = message\_content + "标题：<a href=\"topic.jsp?topicId=" + this\_topic.getId() + "\">";

message\_content = message\_content + "" + this\_topic.getTitle() + "</a>";

/\*新建消息\*/

message new\_message = new message();

new\_message.setContent(message\_content);

new\_message.setType(m\_reply);

new\_message.setSender\_id(m\_sender);

new\_message.setReceiver\_id(this\_topic.getUser\_id());/\*收件人为楼主\*/

message\_dao messageDAO = new message\_dao();

try {

messageDAO.NewMessage(new\_message);

} catch (SQLException sqle) {

}

}

out.println("回复发送成功，3秒后页面将跳转");

response.setHeader("refresh", "3;URL =topic.jsp?topicId=" + topicId);

代码说明：从服务器获取session中的用户对象，并储存在一个user对象中，如果user对象为空，告诉用户需要登录并跳转。新建reply对象并将表单传来的值和user对象中的用户id储存进该reply对象中，然后通过把reply对象传入数据库访问对象中，数据库对象再执行插入语句在数据库新建一条回复纪录。

紧接着判断回复的帖子的楼主是不是当前登录用户，如果不是的话就发送一条回帖通知给楼主(数据库新建消息纪录)。

**其他模块基本都是获取表单值，通过dao对象进行数据库操作，区别基本知识操作和字段的不同**

**数据库连接模块**

private static Connection conn = null;

public static Connection getCon()

{

try{

Class.forName("com.mysql.jdbc.Driver");

String user = "root";

String pwd = "a1585093722\*";

String url = "jdbc:mysql://localhost:3306/bbs\_db";

conn = DriverManager.getConnection(url+"?useUnicode=true&characterEncoding=utf-8", user, pwd);

}catch(Exception e){

e.printStackTrace();

}

return conn;

代码说明：封装了数据库连接的操作，减少在dao对象的代码量

**DAO类的其中一个方法：**

/\*根据帖子的id获取回帖信息\*/

public ArrayList getReplyByTabId(String topic\_id) throws SQLException {

ArrayList reply\_list = new ArrayList();

Connection conn = DBConn.getCon();

ResultSet rs = null;

PreparedStatement pstmt = null;

String sql = "";

sql = "select `Id`,`topic\_id`,`reply\_uid`,`content`,`create\_time` from reply ";

sql = sql + "where topic\_id = '" + topic\_id + "'";

pstmt = conn.prepareStatement(sql);

rs = pstmt.executeQuery();

while (rs.next()) {

reply one\_reply = new reply();

one\_reply.setId(rs.getInt("Id"));

one\_reply.setTopic\_id(rs.getString("topic\_id"));

one\_reply.setReply\_uid(rs.getInt("reply\_uid"));

one\_reply.setContent(rs.getString("content"));

one\_reply.setCreate\_time(rs.getString("create\_time"));

reply\_list.add(one\_reply);

}

rs.close();

pstmt.close();

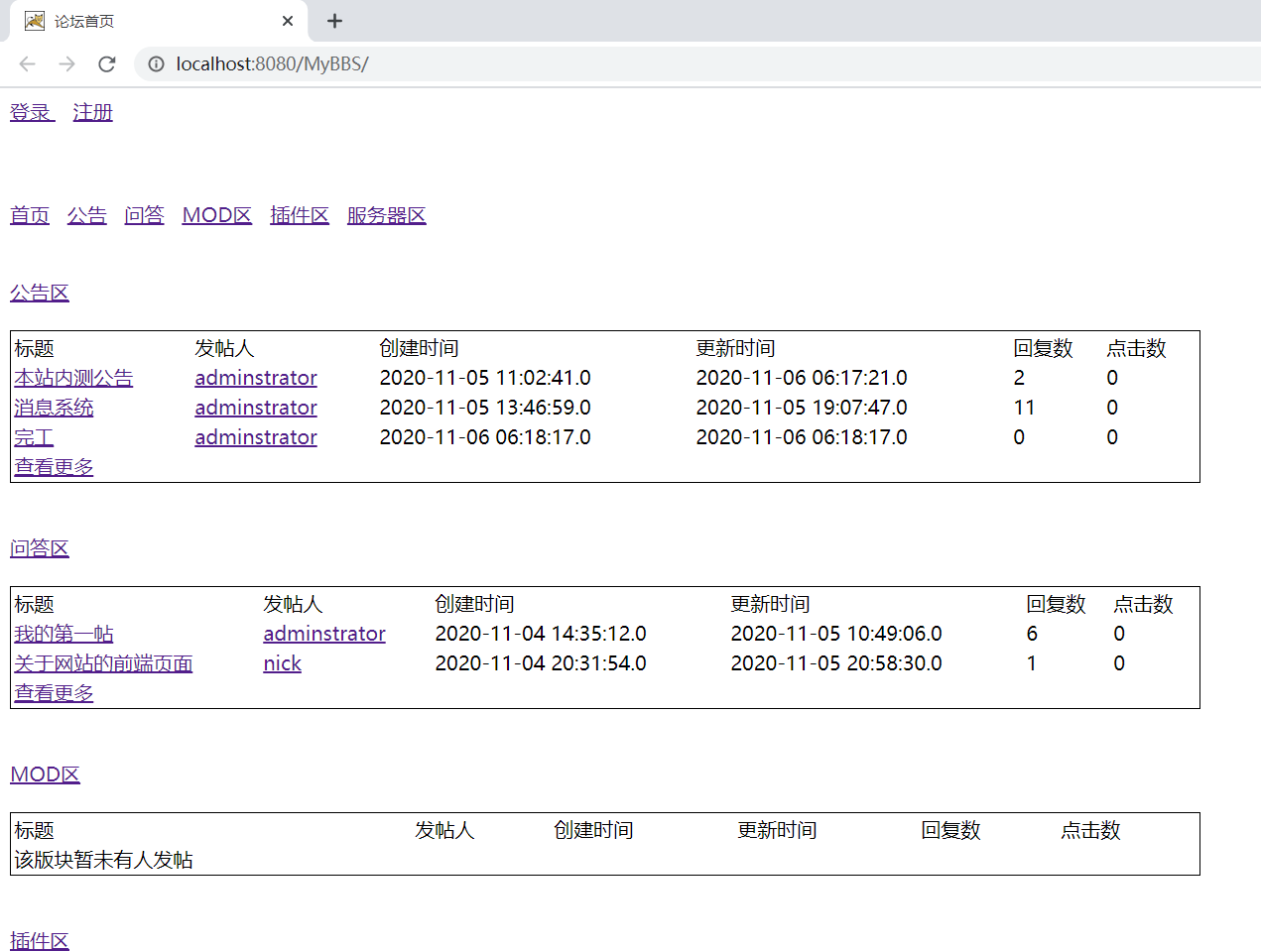
conn.close();

return reply\_list;

}

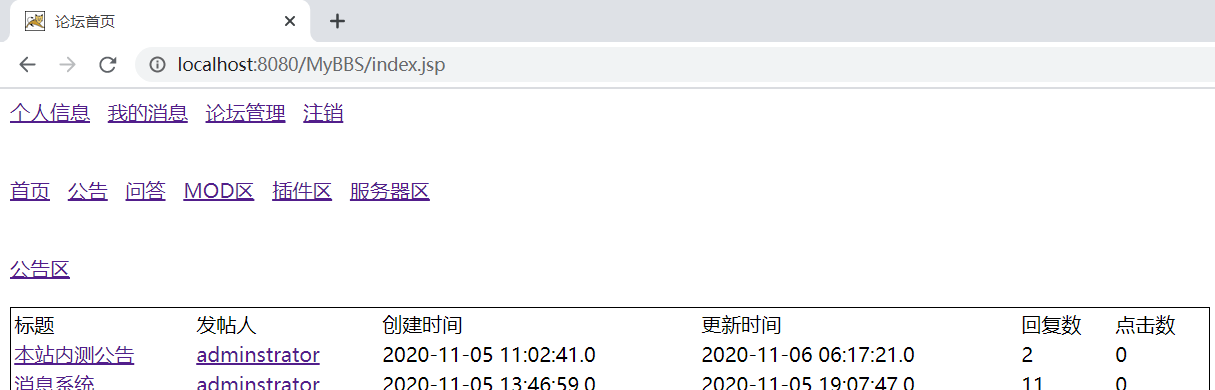
代码说明：关于JAVABEAN类，javabean类必须包含对应数据表中的所有字段，但是可以多出几个其他表的字段，因为关联查询要获取其他表的字段，不用关联查询的话，就会白白增加数据库连接次数和访问次数，造成不必要的运行效能损耗。这段代码是DAO类的其中一个方法，用来获取数据库表中符合条件的所有回帖记录，并将一条记录的值储存进一个对象中，并将对象添加进ArrayList列表中，并作为返回值。

1. 程序的发布和测试
2. 程序的测试过程
3. **首页展示**

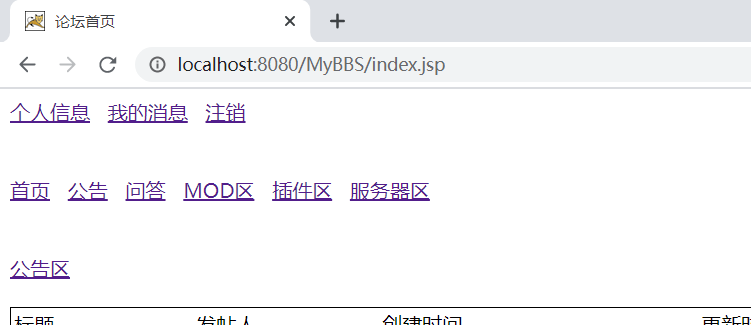


1. **登录之后的首页头部**

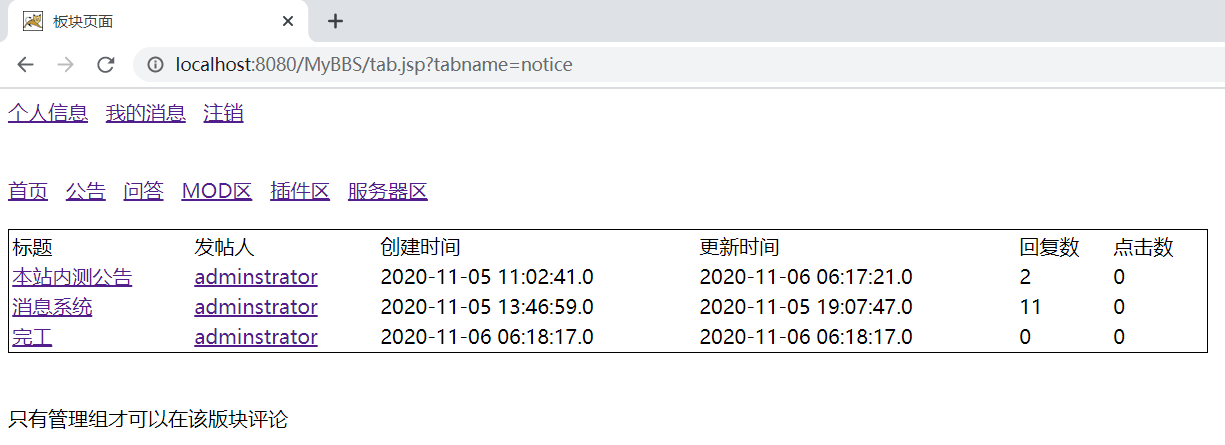
**管理员界面**



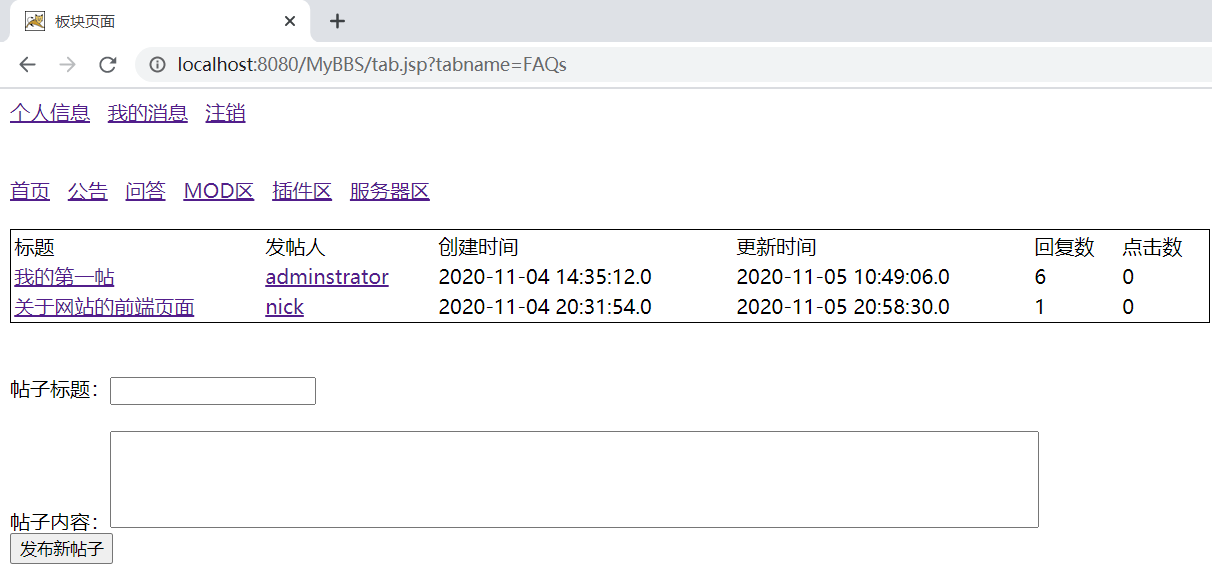
**普通用户界面**



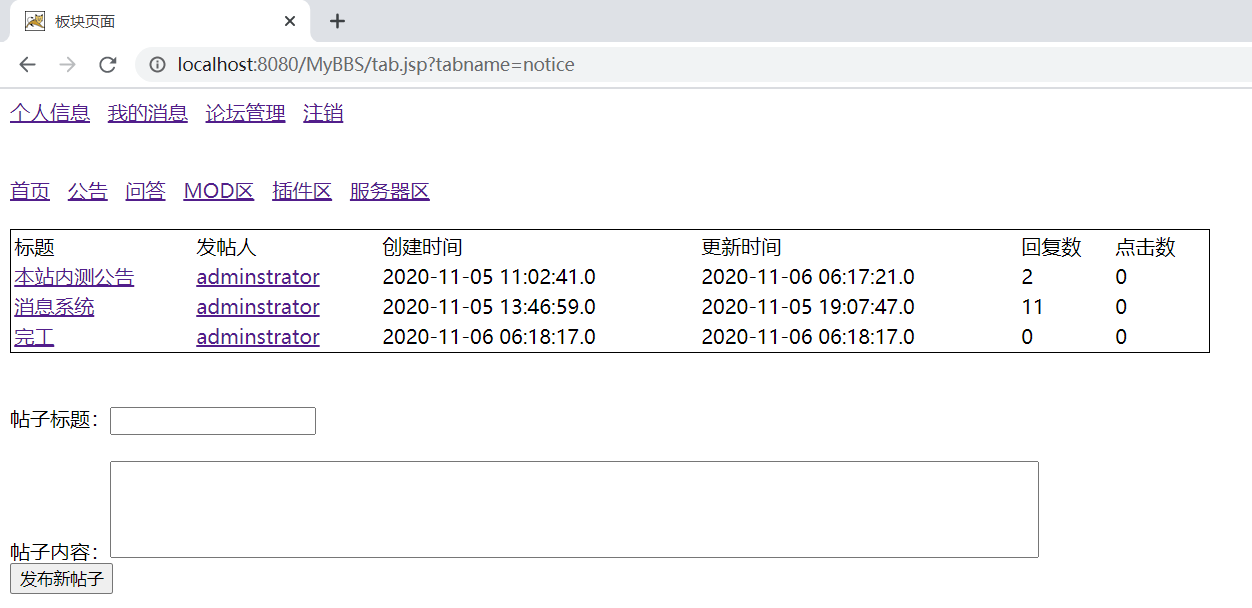
1. **板块区，普通用户不能再公告区发帖**



**但是可以其他板块发帖**



**管理员可以在公告区发帖**



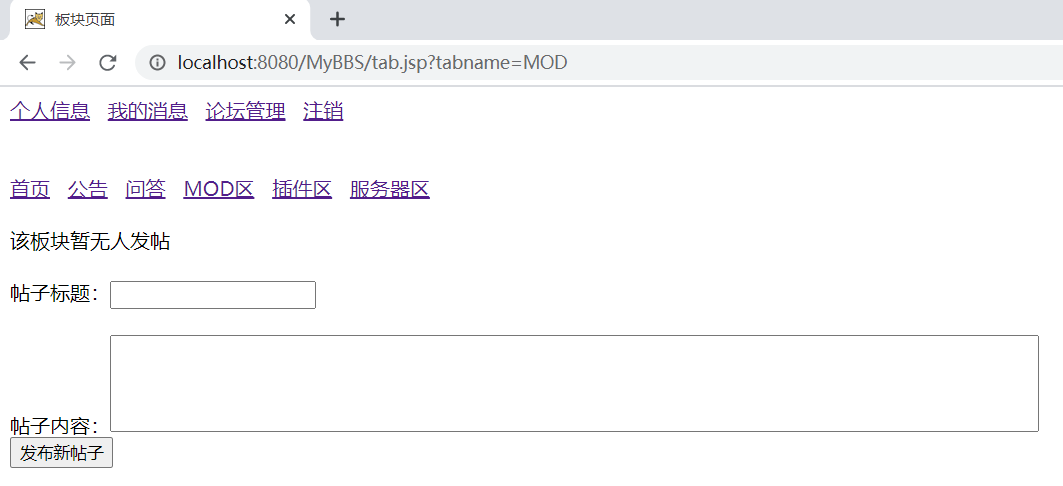
1. **打开一个帖子**



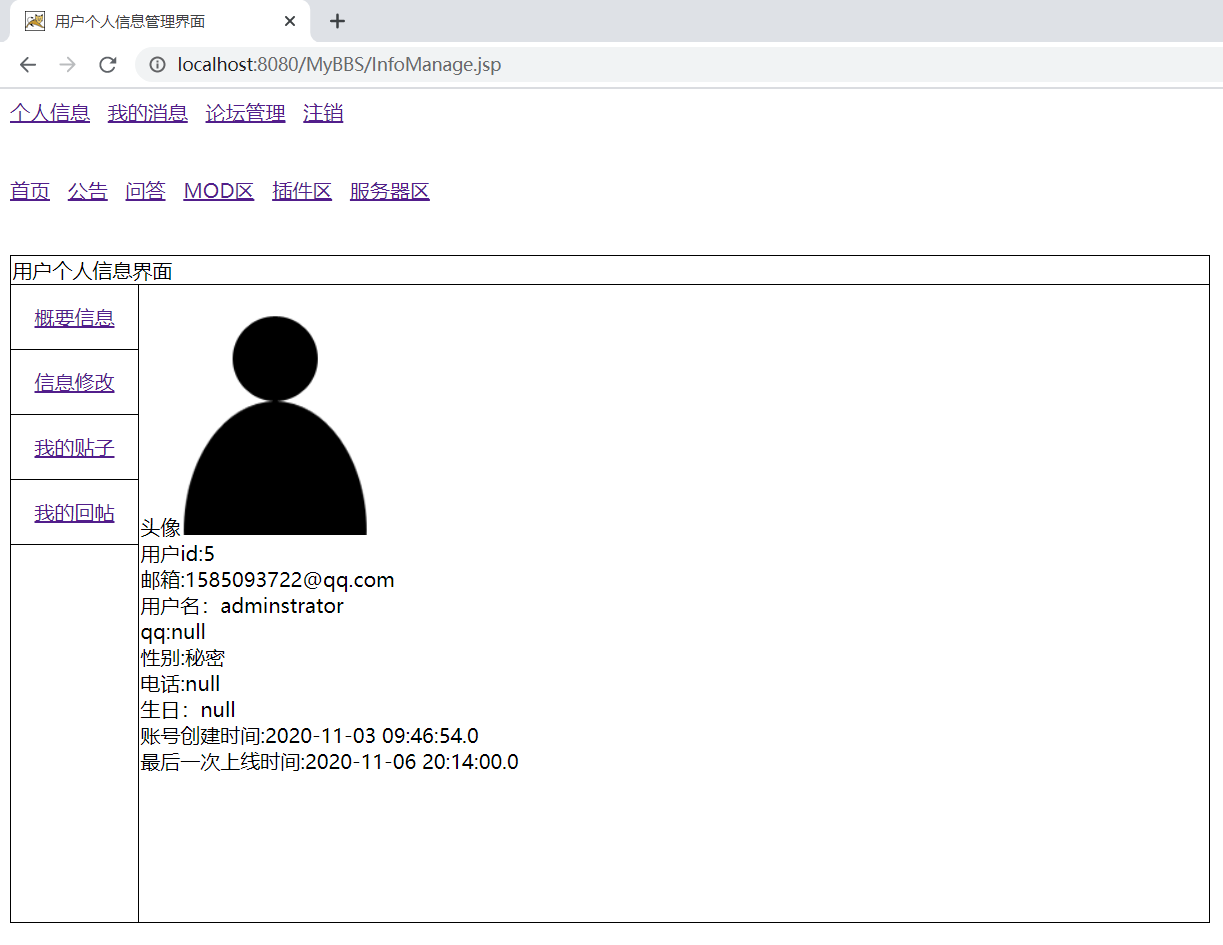
**点击用户头像或者名字可以查看该用户信息**



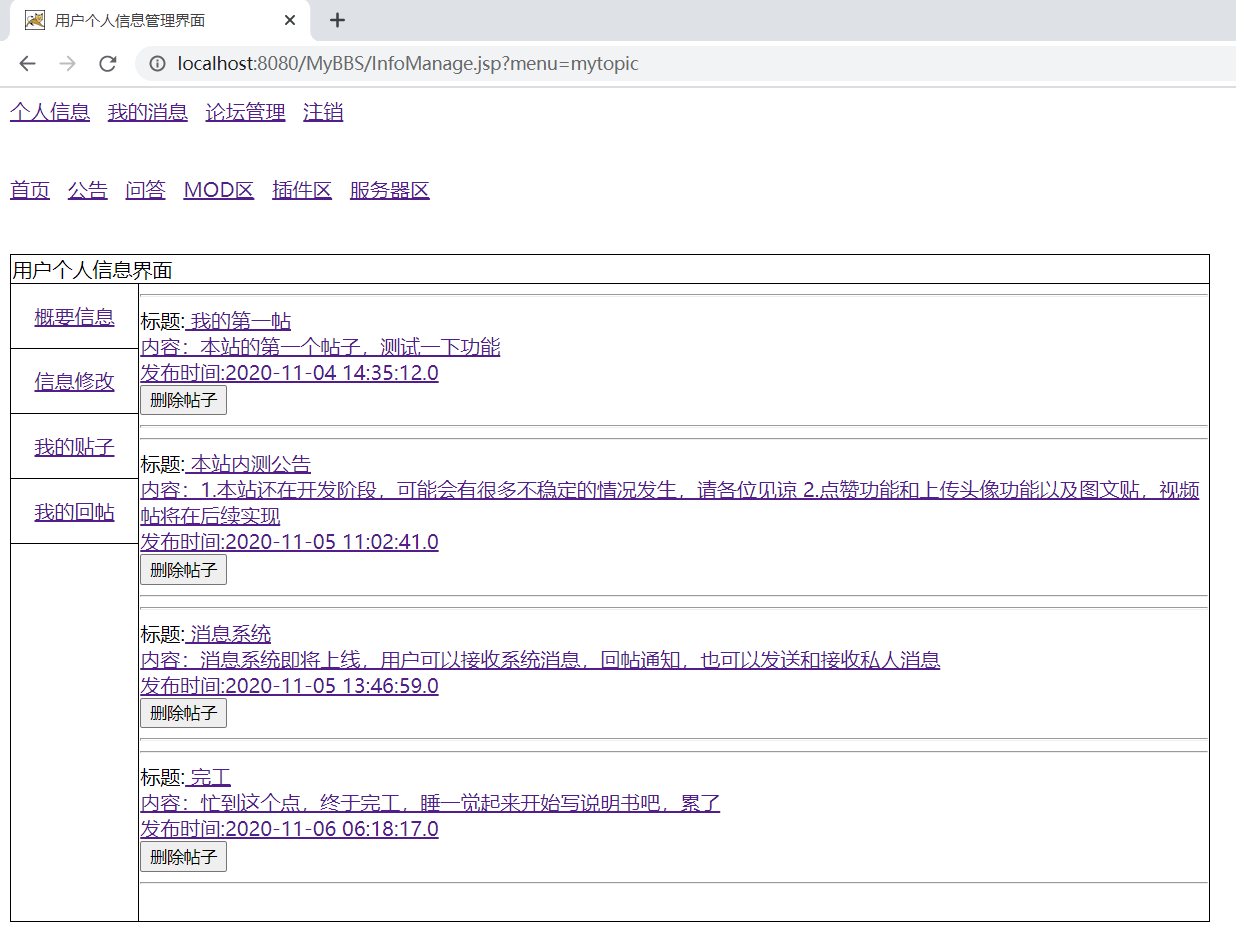
**如果一个版块没人发帖**



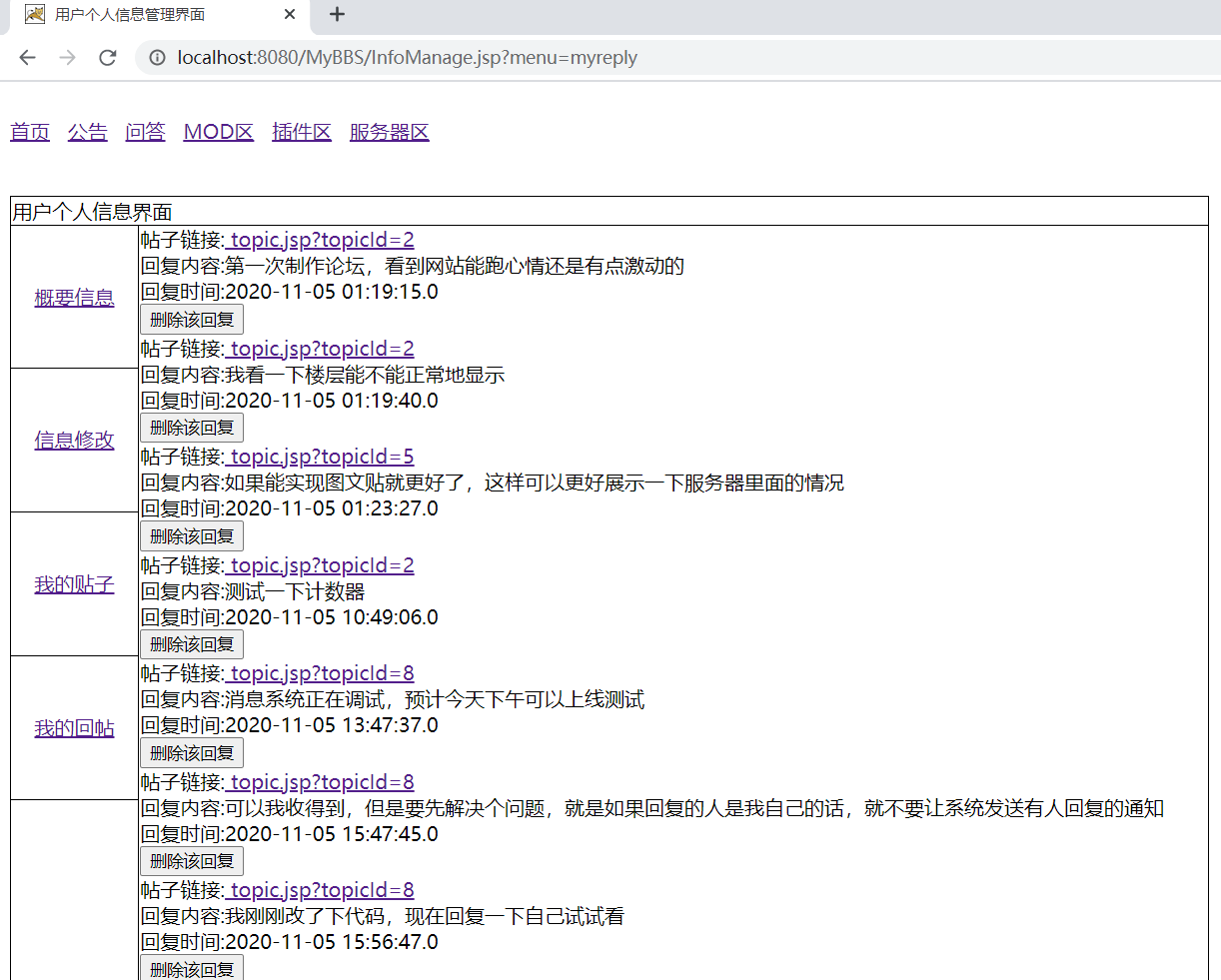
1. **用户个人信息界面**



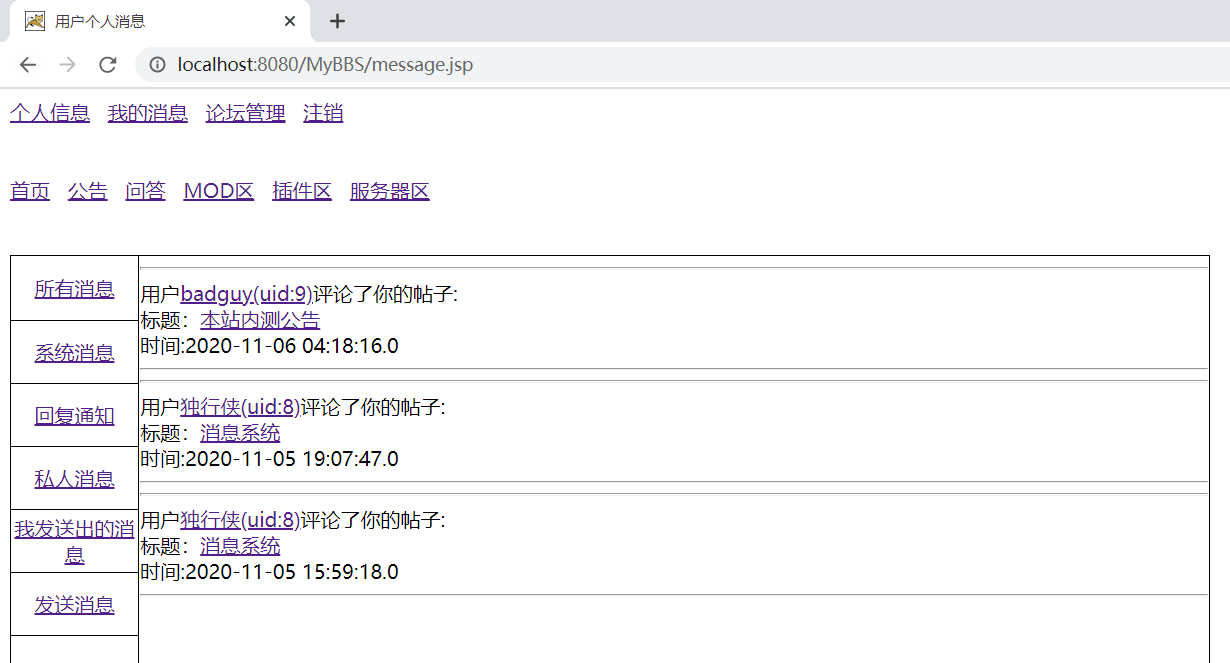
**可以删除自己的帖子，点击去还可以查看帖子**



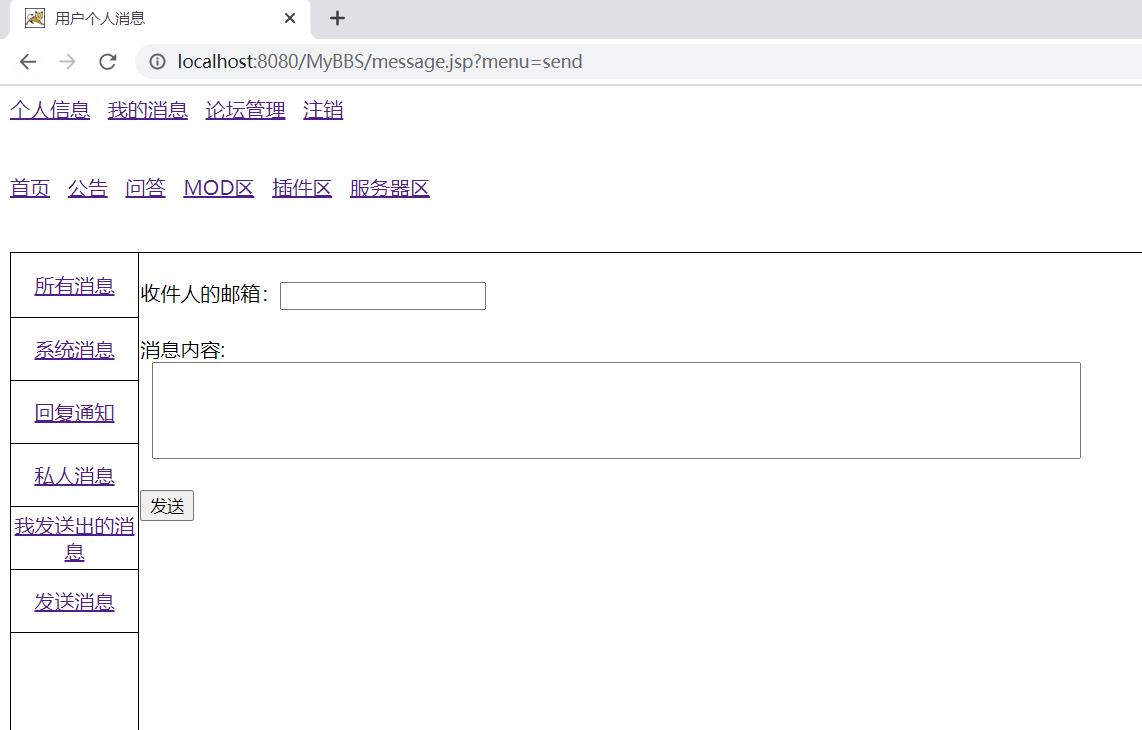
**查看和删除自己的回复，因为工期紧，没有加入页码功能，但是实现起来不难，核心就是通过url传页码的值来控制arrayList对象的输出**



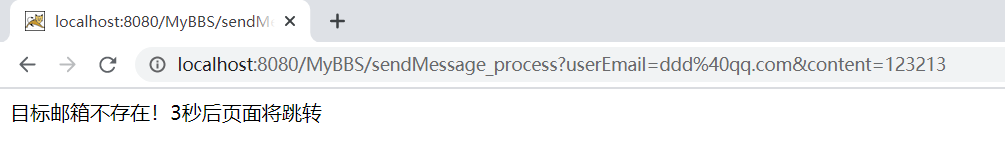
1. **用户消息界面**



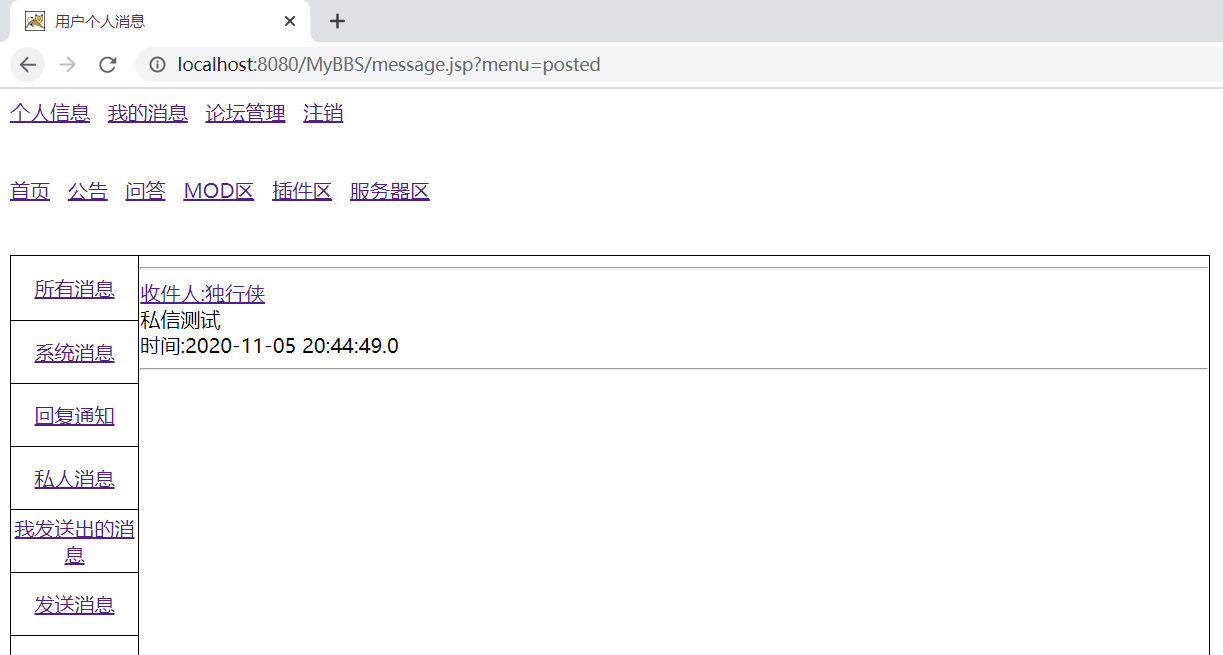
**私人消息发送界面**



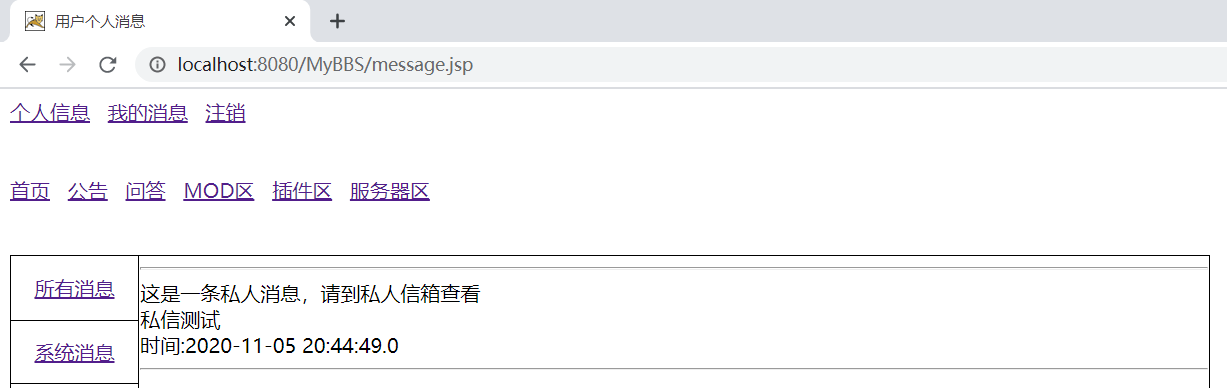
**如果目标用户的邮箱不存在**



**发送的消息可以在“我发送出的消息”选项卡查看，并且会显示收件人和标题**



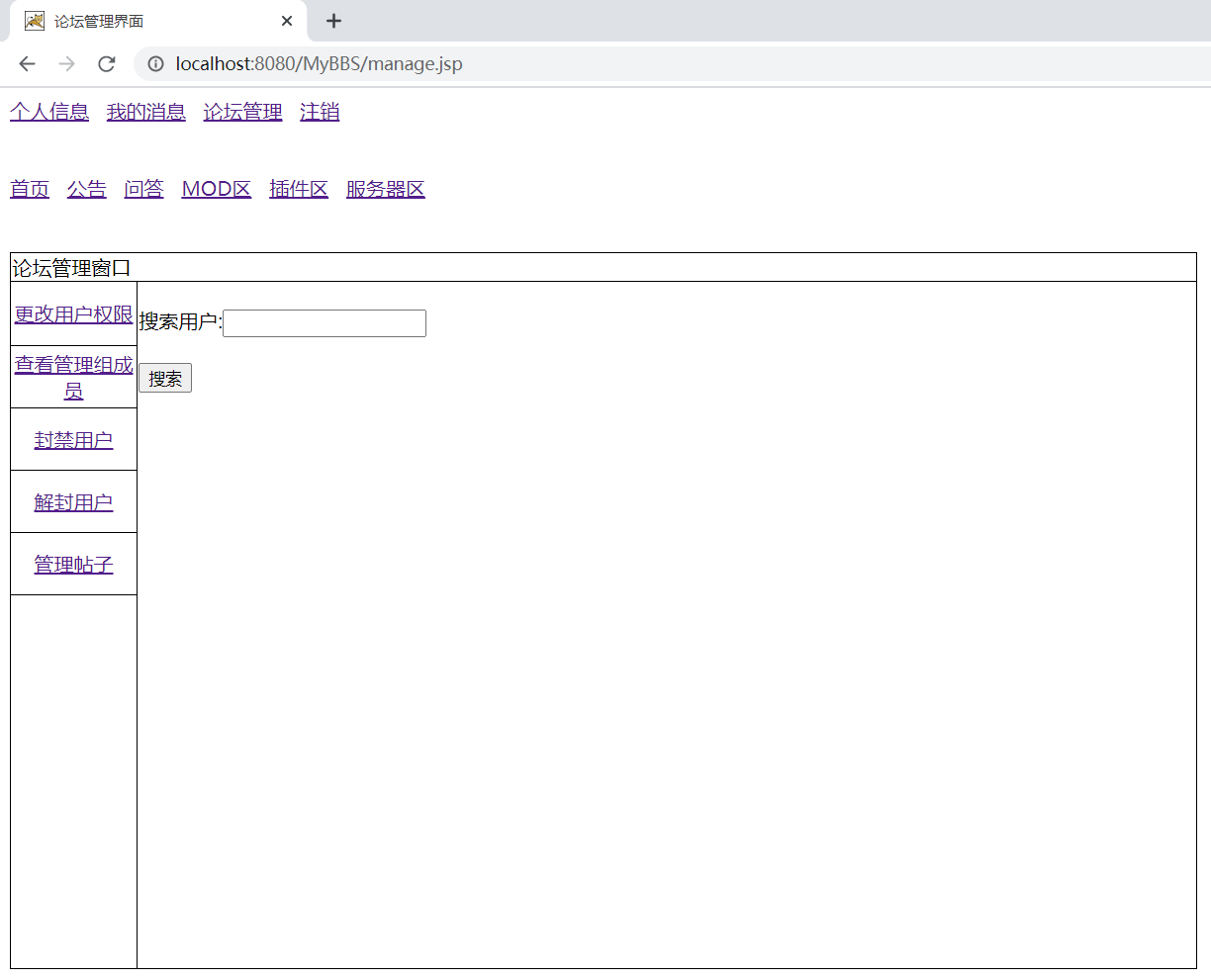
**查看一下收件人的信息**



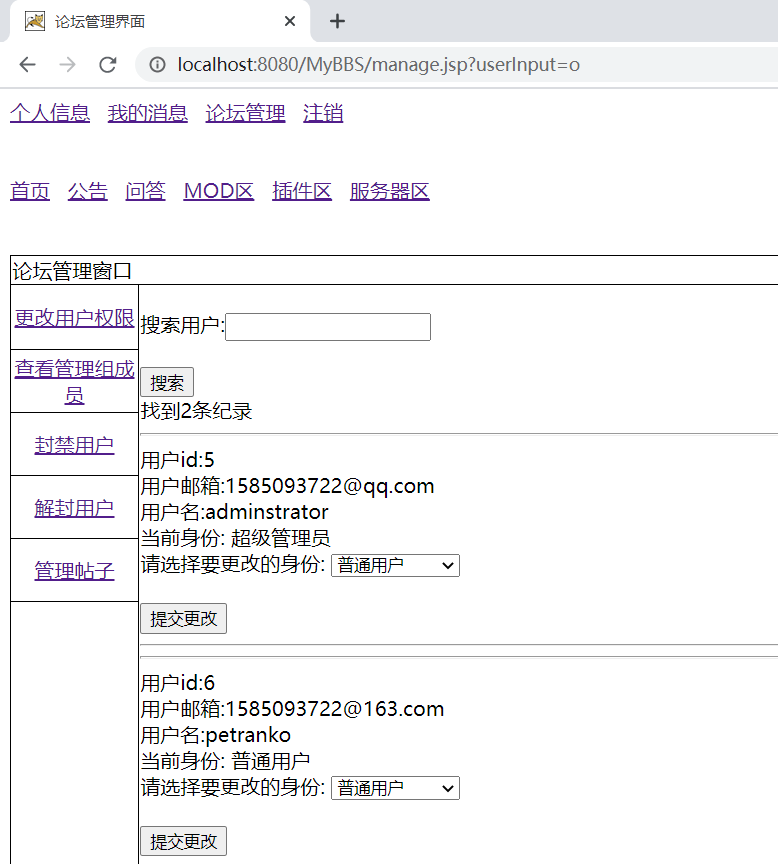


1. **论坛管理界面**

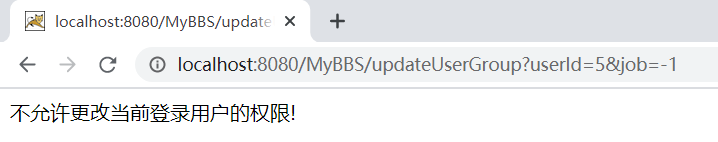
**超级管理员看到的界面，内置了一个用户搜索**



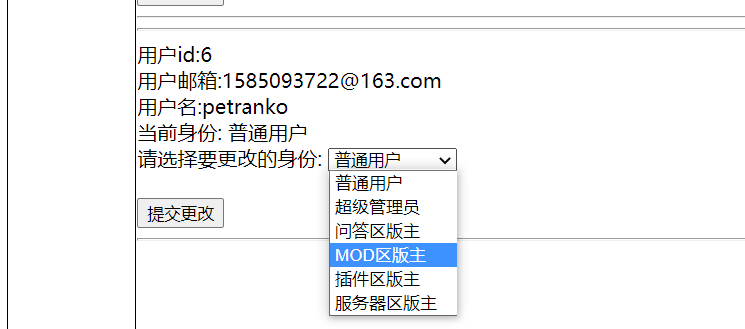
**用一下搜索功能**



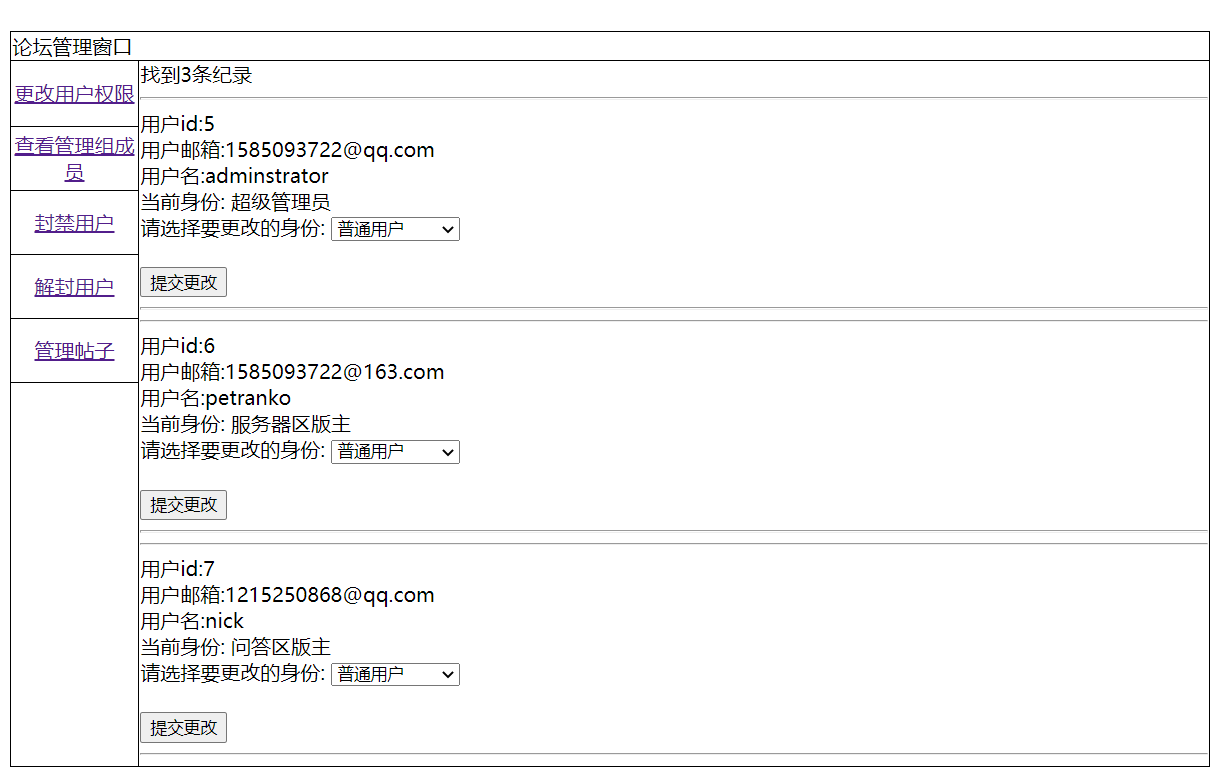
**不能改自己的权限**



**修改他人的权限**



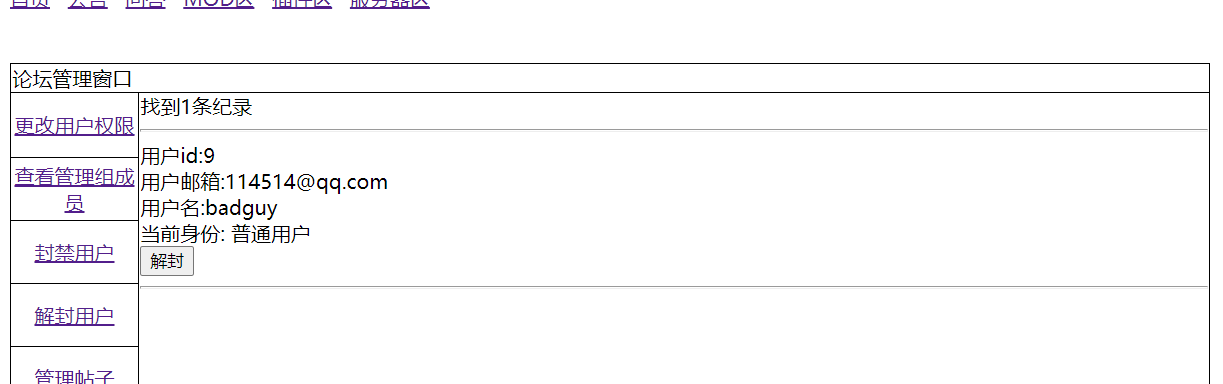
**修改之后，点击查看管理组成员查看一下，多了一个管理组的成员**



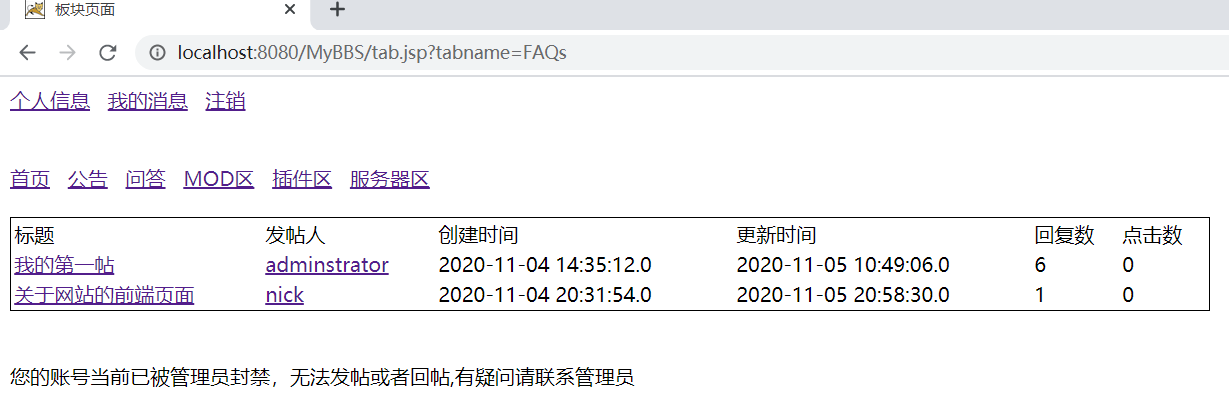
**封禁用户，找一个倒霉蛋做试验**



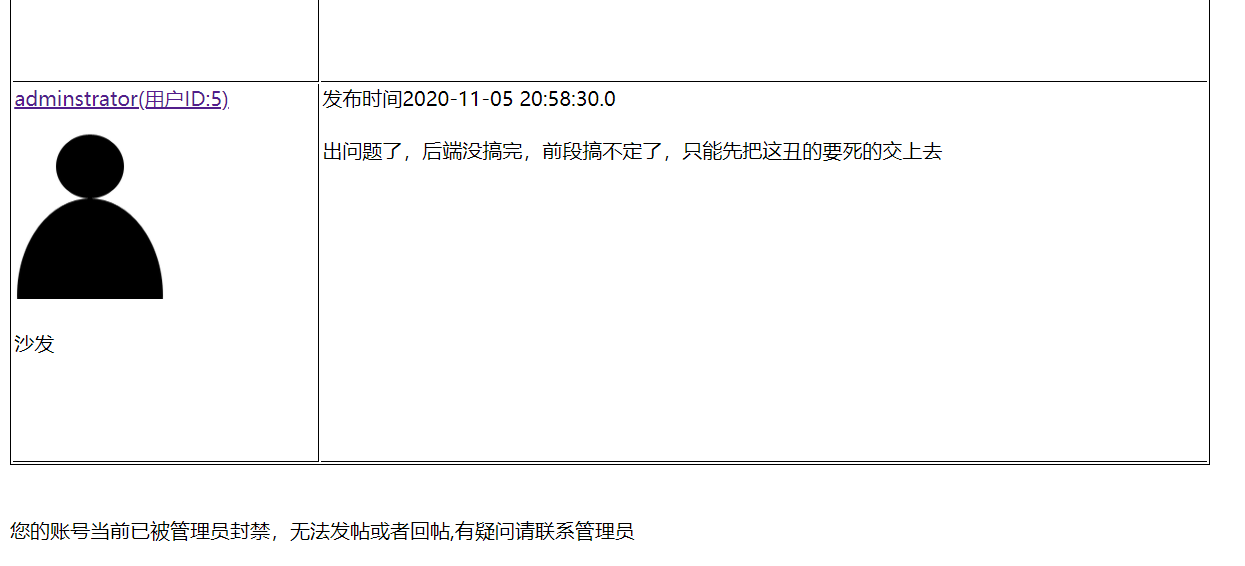
**解封界面可以看到被封禁的用户**



**登录一下被封禁用户的账号看看他还能不能嚣张**



**应该是嚣张不起来了**



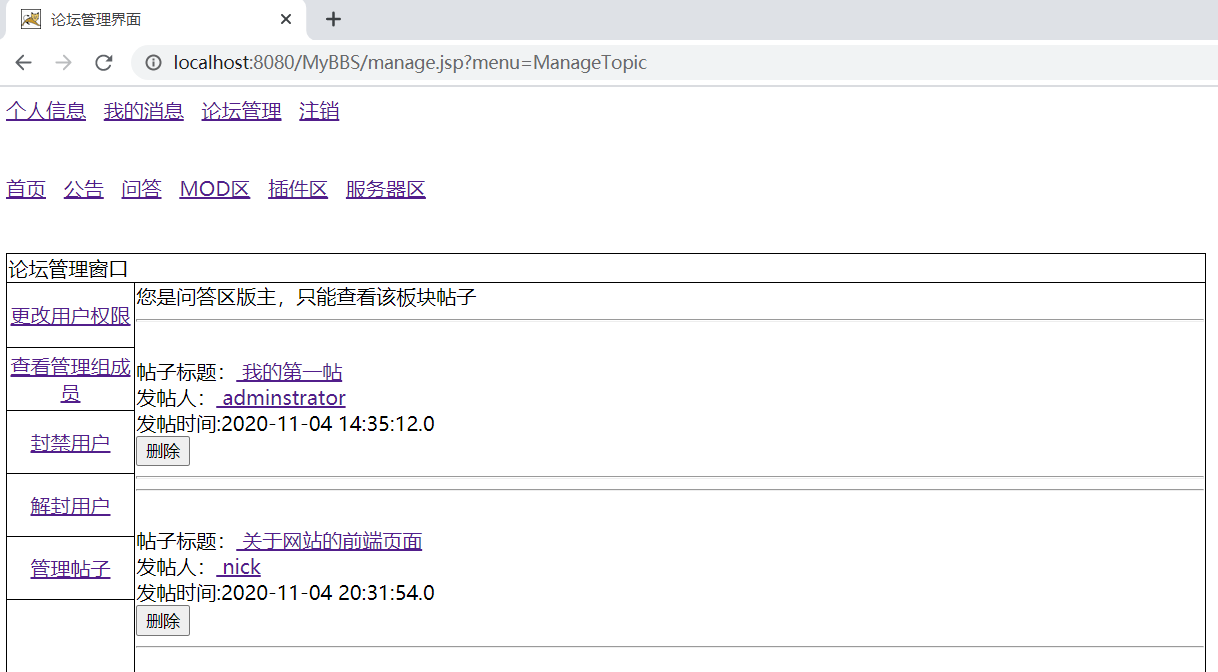
**还是放了他吧**



**管理帖子选项卡**



**如果是版主的话，界面显示如下**



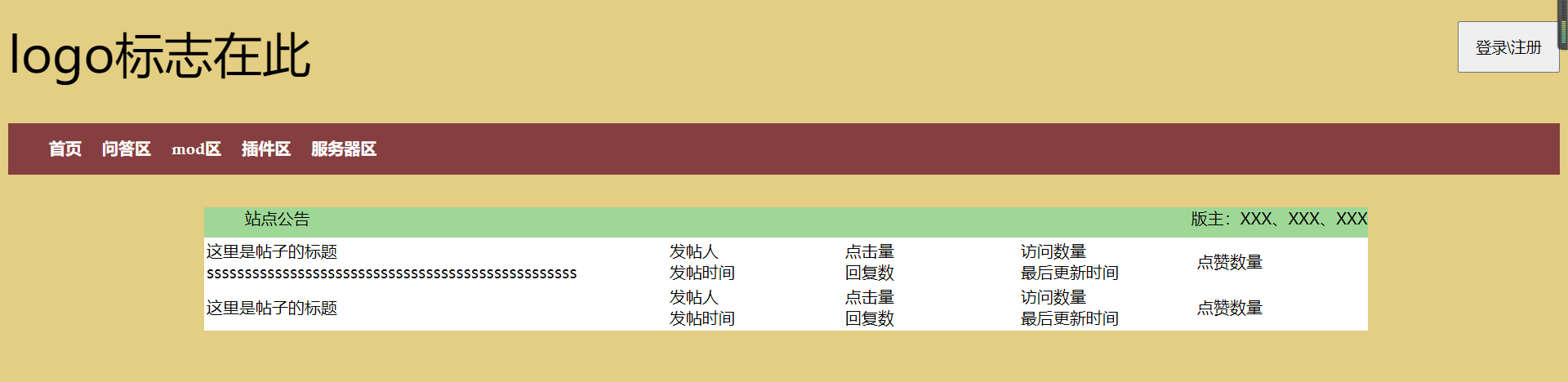
**版主也不能封禁和解封用户更改用户权限**







1. 程序的优点和不足
2. 优点
3. 使用了JAVABEAN和DAO和servlet，封装了对数据库的操作还有后台逻辑层
4. 基本实现了一个论坛的空壳
5. 不足
6. JSP+javaBean和servlet开发虽然实现了项目的一部分分层，但是并不能很好的区分开视图层和控制层，导致jsp页面上全都是java代码，排版和使用样式表困难，周围有人structs2框架，前端页面基本都只是样式代码，控制层和视图层还有服务层通过xml文件进行交互，实现了解耦合。总而言之，jsp、javabean、servlet的开发方式已经过时很久了。
7. 还是上面说的，因为jsp都是java代码，导致排版困难，索性让我放弃了排版，原本我是设计好页面了的



1. 因为拖延症的原因，工期太紧，很多功能没能实现，像是上传头像，图片，视频等功能没做出来
2. 参考目录
3. 设计参照：MCBBS论坛
4. 技术资料来源：百度知道、CSDN以及众多博客
5. 一部分代码参考了以前上javaweb时候留下来的代码