编号

潍 坊 学 院

毕 业 设 计 技 术 报 告

课题名称：开放社区论坛系统的设计与实现

学生姓名：付立威

学 号：17026240040

专业班级：网络工程（移动嵌入式开发）

指导教师：薛莹

2021年5月

**摘要：**自中国改革开放以来，随着互联网的发展， 人与人之间的交流方式愈来愈多。网上聊天，网上视频早已不是什么稀奇的事，后来，为了人们在某一专业领域进行深入的的探讨与发言，出现了在线论坛(BBS)，大家可以就自己的学识经验进行提出问题和针对已有的问题发表自己的见解。由此可见BBS论坛系统已经深入人民的生活中，在人们的生活中担当着一个不可获取或缺的角色，是很多人展示才能、畅所欲言的重要场所。

在当今社会的很多居民社区中，大多数物业通知小区用户的方式还都是贴纸质告示在单元楼下，这样需要耗费很大人力而且用户容易错过一些重要的通知，这时候如果有一个社区专用的论坛，将会事半功倍，所以开发一个社区论坛系统还是很有必要的。开放社区论坛系统，该论坛使用多元化的社区服务模式，其主要目的就是为了丰富社区居民的生活方式。业主和物业等可以自行注册账号，在上面自由发布一些需求和通知，用户间可以互相交流，兴趣多样，话题讨论丰富，更好的方便大家的生活。

本毕设系统是根据现在流行的互联网技术进行开发的一个小型论坛，使用SpringBoot框架整合SSM框架和MySQL数据库进行开发的，通过这些技术，可以实现一个相对完整的论坛系统。

**关键词:** 社区论坛、SpringBoot框架、SSM框架、MySQL数据库

**Abstract:** Since China's reform and opening up, with the development of the Internet, there have been more and more ways of communication between people. Online chats and online videos are no longer uncommon. Later, in order for people to conduct in-depth discussions and speeches in a certain professional field, an online forum (BBS) appeared, where you can ask questions and address your own knowledge and experience. Some issues express their opinions. It can be seen that the BBS forum system has penetrated into people's lives and plays an indispensable role in people's lives. It is an important place for many people to show their talents and speak freely.

In many residential communities in today’s society, most of the ways for properties to notify users in the community are stickers posted downstairs in the unit, which requires a lot of manpower and users easily miss some important notifications. At this time, if there is a community A dedicated forum will get twice the result with half the effort, so it is necessary to develop a community forum system. Open the community forum system, the forum uses a diversified community service model, the main purpose of which is to enrich the lifestyle of community residents. Owners and property owners can register their own accounts and post some requirements and notices freely on them. Users can communicate with each other, have diverse interests, and discuss rich topics, which will better facilitate everyone's life.

This system is a small forum developed based on the current popular Internet technology. It uses the SpringBoot framework to integrate the SSM framework and MySQL database for development. It also uses many middleware technologies such as Redis, ElasticSearch, Kafka, etc. Through these technologies, A relatively complete forum system can be realized.

**Keywords:** community forum、SpringBoot framework、 SSM framework、 MySQL database

目录

[1 前言 1](#_Toc71736670)

[2 系统总体分析 2](#_Toc71736671)

[2.1 系统介绍 2](#_Toc71736672)

[2.2 需求分析 2](#_Toc71736673)

[2.3 可行性分析 2](#_Toc71736674)

[2.4 系统开发环境及简介 3](#_Toc71736675)

[3 系统总体设计 6](#_Toc71736676)

[3.1 系统目标 6](#_Toc71736677)

[3.2 系统功能结构 6](#_Toc71736678)

[3.3 系统流程 7](#_Toc71736679)

[3.4 数据库设计 9](#_Toc71736680)

[4 系统主要功能实现 12](#_Toc71736681)

[4.1 开发社区首页 12](#_Toc71736682)

[4.2 开发注册登录模块 13](#_Toc71736683)

[4.3 开发核心功能 20](#_Toc71736684)

[结束语 24](#_Toc71736685)

[参考文献 25](#_Toc71736686)

[附录 26](#_Toc71736687)

[致谢 29](#_Toc71736688)

1 前言

BBS的英语都叫Bultin Board System。中文的意思是“电子公告牌”。 随着互联网的普及，计算机价格和外部设备在中国急剧下降，家家户户都有无线网，家家户户都有计算机，互联网的概念在21世纪已经深入人心，中国网民数量呈几何级数增长，BBS的普及程度也接近电子邮件，几乎每个网站都有自己的BBS和BBS连接，你可以通过电脑手机的浏览器就可以访问BBS，非常的方便。此外，还可以实现各种私信评论等多种功能。网络用户将依次接收这种利用方便、日益强大的系统，用系统网民将各种信息粘贴发表，可以讨论各种各样的话题。

论坛系统服务也是Internet网站上非常常见的交互式通信服务。 论坛为用户提供了一个开放的，分类的主题讨论区服务。 为了在这里表达您的一些想法，您可以交流一些技能，经验，甚至生活见识和乐趣。 在这里，您还可以回答用户的问题以及发布的一些消息。 通过论坛可以使个人，团体，团体之间以及个人与团体之间的交流和沟通更为紧密。 通过互联网实现信息传输，提高服务效率。 因此，论坛在互联网应用中的地位是明确的，已经成为现代通信和信息获取的重要组成部分，并受到人们的高度重视。

Java是现在主流的编程语言，SSM框架是现在企业开发用的很多框架的组合，它们得到了广泛的认可和支持。SMM框架与许多Java技术相结合，构建了更加稳定、高效、安全的运行环境，实现了非常复杂的企业级应用系统。所以，我用Java语言开发了这个社区论坛系统，作为毕业设计。

对于社区而言，该论坛可以从房地产开发的开始就吸引用户。 此外，所有者社区的独特“小圈子”特征使页面上的活动用户非常准确。比如在阿里巴巴旗下的产品“闲鱼”中，就有以小区为单位的“鱼塘”，所谓的鱼塘就是一个聚集地，其目的就是为了方便大家，在这种情况下，您可以参与多个鱼塘，并快速了解周围的人在卖什么，例如在1公里和2公里之内的朋友在卖什么。非常直观，更贴近生活。但是这种仅仅局限于在二手交易上，我觉得一个社区可以有很多丰富的多元化的东西，不仅仅只是如此。

本文主要利用所学知识，进行了一个社区论坛的项目开发。主要功能是在客户端和服务器之间建立动态交互，论坛最重要的功能是投稿和回复，另外，要注册主题编辑和责任主题，系统必须提供用户注册和注册功能。

2 系统总体分析

2.1 系统介绍

本系统是B/S结构的小型论坛，主要基于Windows操作系统 ，通过这个论坛，社区居民之间可以方便地在线上互相交流沟通。

2.2 需求分析

2.2.1 功能性需求

为什么要开发BBS论坛系统呢？是为了给社区使用者提供一个可以交流的平台，并为更多用户讨论交流经验和问题提供一个社区。 因此，本毕设系统的最基本功能是发表主题，然后其他人根据再根据版主发表的主题发表意见。 另外，系统必须提供用户登录和登录功能，以记录主题发布者和主题追加者信息。 只有注册的社区区民用户才能发布和回复主题。 查看者只能查看主题信息。 BBS论坛需要根据用户需求和上述分析提供前端和后端功能。

1. 用户权限：显示各论坛的种类和篇幅，查阅纸面下的所有线程，查阅精华的帖子，查阅自己发表的帖子，查看私信，发表帖的内容、用户登录。

（2）管理者权限：进入系统后台、管理用户、管理帖子、用户注册。

2.2.2 使用需求

用户只需使用可以连接网络的Windows系统的浏览器进行操作即可，无需安装其他应用软件。

2.3 可行性分析

2.3.1 技术可行性分析

论坛系统采用IDEA作为开发工具，控制层框架采用SpringMvc，业务逻辑层框架采用Spring，持久层框架采用Mybatis，使用SpringBoot整合了当下最热门的三大框架，将它们的优点效率最大化。模块划分方面分为了视图层，业务层和持久层，使各个模块之间互相协作，共享数据，易于维护。

因此在技术上完全可行。

2.3.2 经济可行性分析

主要对当前项目前期投入和预计回报进行系统经济分析，系统估计系统开发过程成本和后续运行维护成本，系统评价系统正式使用后获得的经济效益和社会效益。本系统的开发目的是用户能够更快地接收消息通知，由于花费很少，所以在经济上也是可能的。

2.3.3 维护可行性分析

社区论坛系统由几个功能子模块构成，在前端和后端的相互作用中，采用Thymleaf引擎的方式，易于维护，开发使用的工具IDEA在市场上经常被使用，所以本系统易于维护并更新。

2.4 系统开发环境及简介

• 构建工具：Apache Maven

• 集成开发环境：IDEA

• 快速开发框架：SpringBoot整合Sring+SpringMVC+Mybatis

• 页面模板引擎：Thymleaf

• 数据库：MySQL

• 应用服务器：Apache Tomcat

Spring框架：

Spring是一个开放源码的轻量级Java框架，它的出现是公司的福音，它大大降低了企业级应用开发的复杂性。核心思想是“高内聚、低耦合”，主要优点是采用了分层式体系结构，Spring的实现并不局限于服务器端的开发。自spring以来，任何Java应用程序都将得到升级和改进，它是一个分层的体系结构，主要由Core Container，AOP，WEB，Data Access，Test五个模块组成。

Spring MVC框架：

SpringMVC是一个视图层框架，主要用于处理用户所接受到所看到的页面，早期市面上用的MVC框架都是Struts2，但是因为Struts2框架暴露出来一些安全问题，用Struts2的越来越少，用SpringMVC的越来越多。SpringMVC上手容易配置简单，所以它已经成为广大企业优先选择的MVC框架，这也是我在这个项目中为什么选择使用它来作为视图层框架的原因。SpringMVC共有六大组件：DisPatcherServlet、HandlerMapping、HandLer、HandlerAdapter、ViewResolver、View。

MyBatis框架：

MyBatis是独特的开源数据的耐久层框架。他的内部完成了JDBC的建立，使用JDBC访问了数据库的一系列操作，支持常规SQL查询、高级映射和存储过程，并且几乎用手册把JDBC连接代码的操作抹掉了。作为长框架的MyBatis，为了使SQL语句的大部分脱离程序，如果预先写在设定文件中，则能够灵活地配置SQL语句，不需要每次变更代码。

SSM框架：

SSM框架就是Spring 框架+ Spring MVC框架 + MyBatis框架的一个整合。SSM框架是继SSH框架之后，现在很多公司使用的JavaEE企业的开发框架，适用于各种大型企业，正在开发复杂的企业级应用系统。

SpringBoot框架：

SpringBoot框架其被设计出来的目的就是简化开发，秉承“约定优于配置”的设计理念，它内置许多默认配置，使得开发者近乎0配置，就可以让项目快速的跑起来。此外它还有一个重要的特点，它集成了非常多的第三方库的配置信息，近乎所有的常用框架，它都有相对应的组件的支持，它使得这些第三方库可以开箱即用，使得开发者不必花很多心思在配置组件上，而更专注于业务逻辑代码的开发。所以它的优点就显而易见了，这也是它这么流行的原因，本设计中，采用SpringBoot整合SSM框架和Redis、Kafka等中间件，非常方便高校。

Maven：

Maven是一个开源的项目的管理工具和构建工具，它的主要组成组件是项目对象模型、依赖管理系统、项目生命周期和各种插件。有了它，我们不需要自己管理导入依赖jar包，直接告诉Maven项目需要那些依赖jar包，它会帮助我们下载所有依赖项。

POM文件是Maven项目的核心文件，是XML文件，包含工程的构筑、插件依存关系、项目的基本信息等。

开发人员只需要配置pom.xml这个文件，就可以很方便的管理Maven项目，使得开发事半功倍。

Thymeleaf：

Thymeleaf是模板引擎，为了变更引擎采用Java开发。独立环境下的服务器端Java模板引擎。面向Web，可以处理多种格式的文件文本。其优点是能够高度维持，并且基于自然模板的概念，其逻辑被插入模板文件中，并且不影响被用作设计的原型。这样可以改善设计沟通，缩小设计团队和开发团队的差距。所以可以完全代替JSP。

3 系统总体设计

3.1 系统目标

1. 实现帖子浏览分页查看详情功能
2. 实现用户的注册登录及权限控制功能
3. 实现帖子的置顶拉黑功能
4. 实现用户点赞评论和私信功能
5. 实现用户在线发布帖子功能
6. 实现管理员管理普通用户功能
7. 用户可以查看自己发布的帖子列表
8. 保证系统化的易操作性和易维护性
9. 保证系统运行安全稳定

3.2 系统功能结构

未登录的游客可以浏览帖子，但不能点赞评论和私信帖子主人，游客注册登录之后，成为正式用户，可以解锁发布帖子、点赞、评论、私信等功能，用户可以查看编辑自己发布的帖子。社区管理员用户可以管理用户和所有帖子。

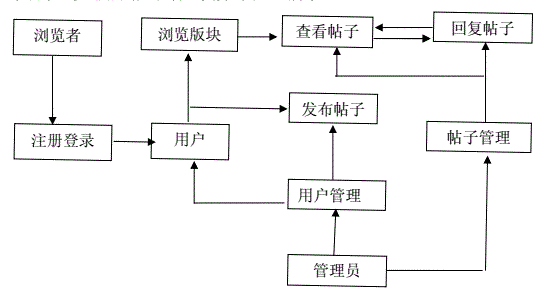


图3.1 系统运行流程总览

3.3 系统流程

3.3.1 注册流程

1. 用户打开注册页面。
2. 填写注册信息。
3. 点击注册按钮，把注册的信息发送给服务器。
4. 登录注册填写的邮箱中找到激活邮件。
5. 打开邮件中的激活链接打开网页，完成注册。

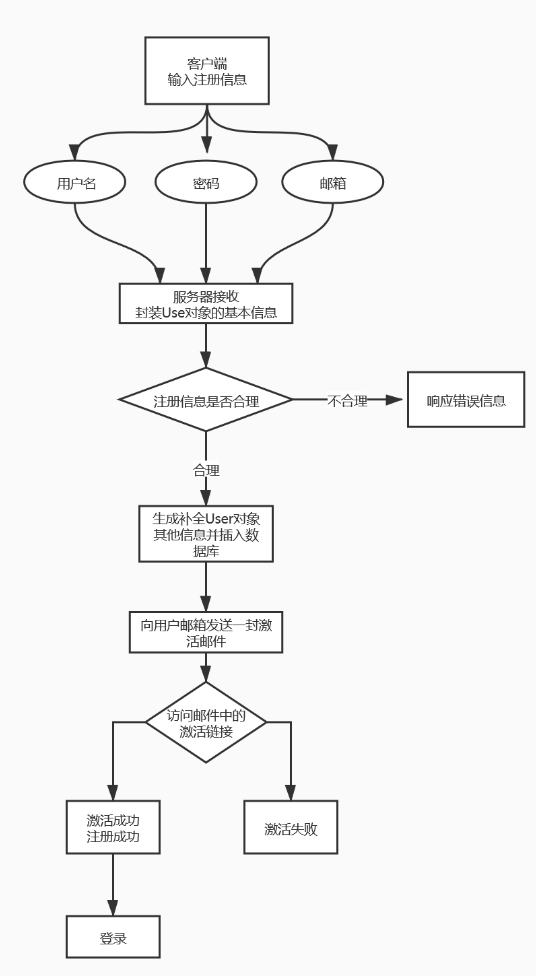


图3.2 用户注册流程总览

3.3.2 登录流程

1. 客户端输入登录信息。
2. 点击登录按钮将信息提交到服务器。
3. 服务器收到登录信息进行有效性验证。
4. 验证通过后生成一个登录凭证到Cookie中。
5. 服务器将Cookie响应给客户端，之后用户再访问就会携带凭证Cookie，服务器就会将其判定为登录状态。

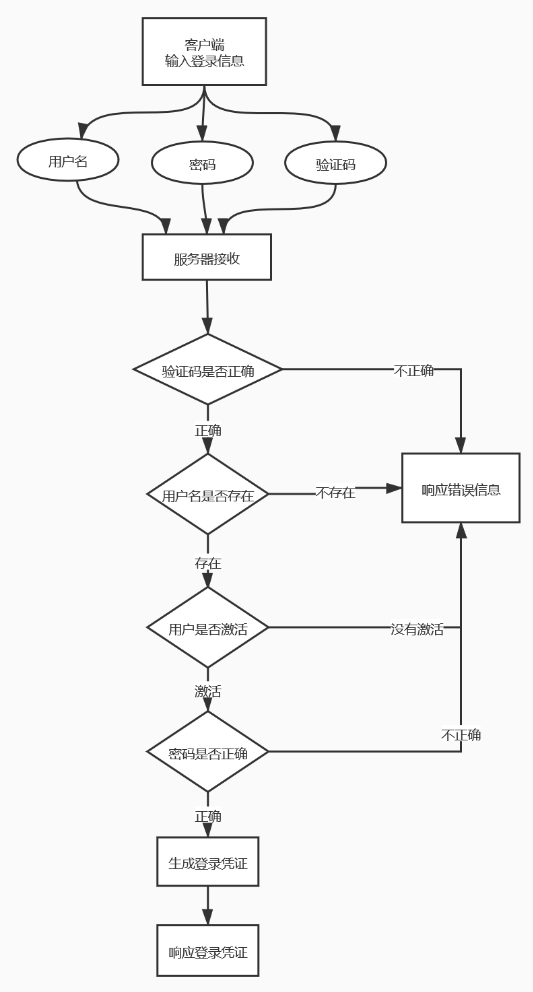


图3.3 用户登录流程总览

3.4 数据库设计

**用户表 user：**

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 备注 |
| id | int | 主键、自增 |
| username | varchar | 用户名 |
| password | varchar | 用户密码 |
| salt | varchar | 加密盐值 |
| email | varchar | 用户邮箱 |
| type | int | 用户类型 |
| status | int | 用户状态 |
| activation\_code | varchar | 激活码 |
| header\_url | varchar | 用户头像地址 |
| create\_time | timestamp | 注册时间 |

**评论表 comment：**

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 备注 |
| id | int | 主键、自增 |
| user\_id | int | 评论的用户 id |
| entity\_id | int | 评论实体 id |
| entity\_type | int | 评论实体类型 |
| target\_id | int | 评论目标 id |
| content | text | 评论内容 |
| status | int | 评论状态 |
| create\_time | timestamp | 评论发表时间 |

**帖子表 discuss\_post：**

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 备注 |
| id | int | 主键、自增 |
| user\_id | int | 发帖的用户 id |
| title | varchar | 帖子表标题 |
| content | text | 帖子内容 |
| type | int | 帖子类型 |
| comment\_count | int | 评论数量 |
| status | int | 帖子状态 |
| create\_time | timestamp | 评论发表时间 |

**用户登录凭证表 login\_ticket：**

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 备注 |
| id | int | 主键、自增 |
| user\_id | int | 登录用户 id |
| ticket | varchar | 登录凭证，随机字符串 |
| status | int | 登录状态 |
| expired | timestamp | 过期时间 |

**消息表 message：**

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 备注 |
| id | int | 主键、自增 |
| from\_id | int | 发消息的 id |
| to\_id | int | 收消息的 id |
| conversation\_id | varchar | 会话 id，由通信双方 id 拼接 |
| content | text | 消息内容 |
| status | int | 消息状态 |
| create\_time | timestamp | 消息发送时间 |

4 系统主要功能实现

4.1 开发社区首页

在用户访问首页的时候，要展示帖子列表，不能一次性全显示出来，要使用分页功能，创建一个分页JavaBean对象Page，包含当前页码，每页显示条目数，总条目数，查询路径这四个属性。

查询的数据库表是discuss\_post，排序规则是置顶精华类型优先展示，剩余帖子按热度进行排序。

由于还要展示用户的信息，所以还要查询user表。

**开发数据访问层：**

首先创建Bean DisscussPost 实体类，作为查询数据库返回的接收对象

创建访问持久层数据接口文件并并标注注解，使Spring扫描并加载到Bean容器中去。

创建与接口对应的mapper文件：disscusspost-mapper.xml，编写查询的SQL语句。

注意：排除状态为拉黑的帖子，如果传入用户id字段，说明查询的是某用户帖子列表，需要添加条件。

**开发业务层：**

创建DiscussPostService类，在类上声明@Service注解，新增查询帖子详情方法，查询帖子数量方法。

还需要创建UserService类，同样声明@Service注解，在其中新增根据用户id查询用户方法，因为展示帖子的时候不能展示用户id，要展示用户名。

**开发视图控制层：**

创建static目录， css、html、img、js 静态资源等静态资源。

创建template目录，存放mail、site、index.html 等文件。

创建HomeController视图层对象，在类上方标注@Controller注解，表示此类用来处理接收url请求的类，同时让Spring加载到Bean容器中，新增getIndexPage方法，将用户帖子信息封装到Model对象中，方法返回值为字符串，返回index主页路径。

修改文件：index.html，将里面需要加载到页面上的值用th标签动态替换。

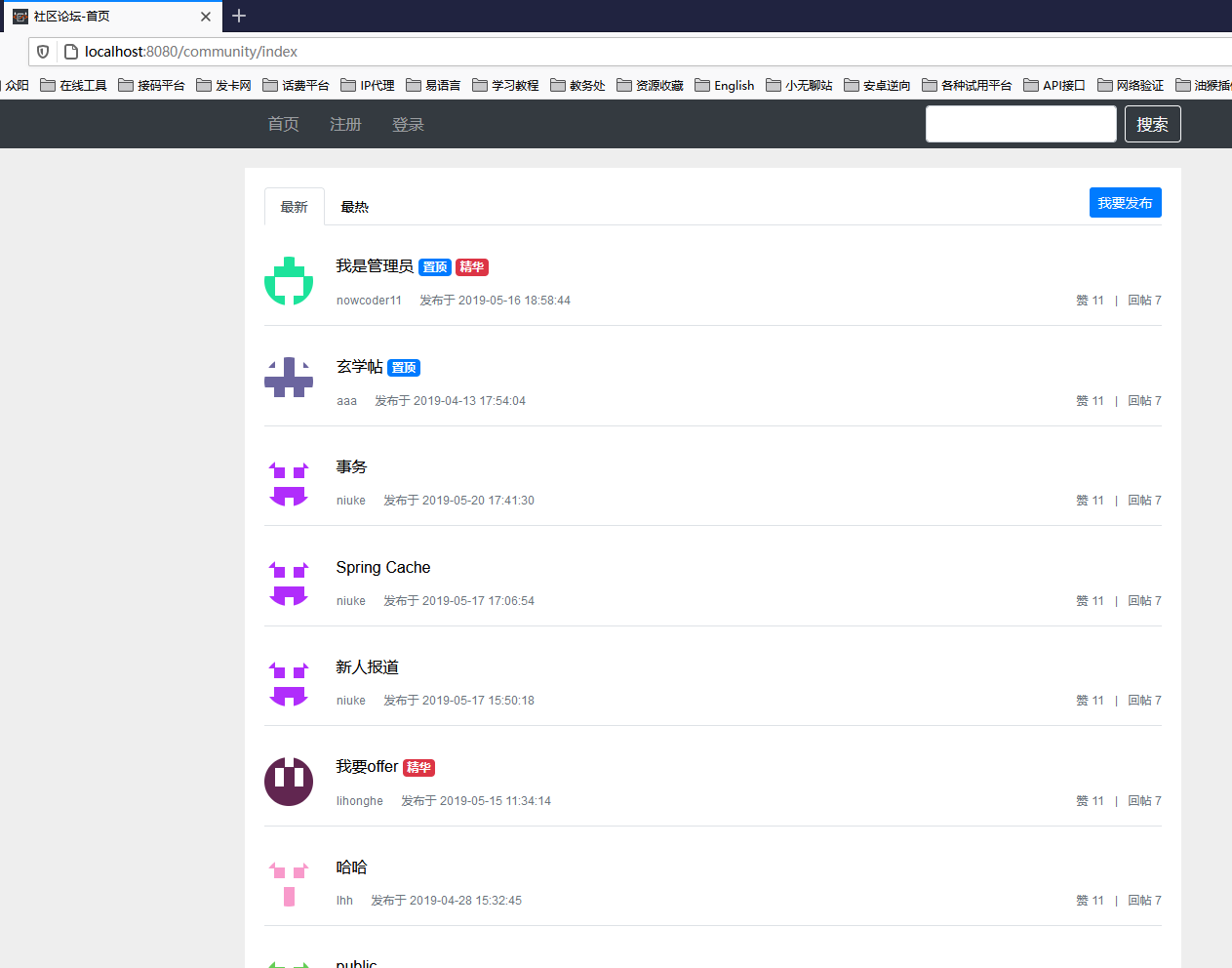


图4.1论坛首页



图4.2分页组件

4.2 开发注册登录模块

**发送邮件：**

发送邮件的功能是必要的，因为需要给用户的邮箱发送激活邮件，在自己常用的邮箱设置中心开启SMTP服务，无论是哪种邮箱都会有的，记下来用户名和和访问SMTP服务的密码。

在pom文件中需要中引入邮件服务的起步依赖：spring-boot-starter-mail 。

在Spring主配置文件application.properties中配置邮件服务器主机地址、邮件服务器主机端口号、发件人邮箱、发件人授权码等信息。

新建工具类：MailClient，封装使用JavaMailSender 组件来完成发送邮件功能，这样封装好，后面直接调用就可以了。

写一个邮件用的HTML模板，使用thymeleaf 加载模板，并替换对应的需要动态显示的信息，再调用TemplateEngine把注册用户的信息动态加载到 HTML 模板中，完成激活邮件要发送的内容。

注册功能

新建工具类： CommunityUtil，在工具类新建产生随机字符串UUID的方法，和 传入字符串获取唯一MD5加密值的方法。

在LoginController视图层类中，新建方法：getRegisterPage，返回register.html路径使其跳转到注册页面。

在UserService业务层类中新建方法： register，这样是为了在方便视图层调用，用于判断传入的User对象包含的注册信息是否合理，如果合理生成密码盐和激活码插入数据库，并向用户邮箱发送激活邮件。

激活电子邮件成功发送后，用户在其邮箱中注册，单击电子邮件中的链接，然后访问logincontroller激活方法。如果激活链接中的激活码与数据库中的设置相同，则相应的用户状态将更改为1，表示已激活。

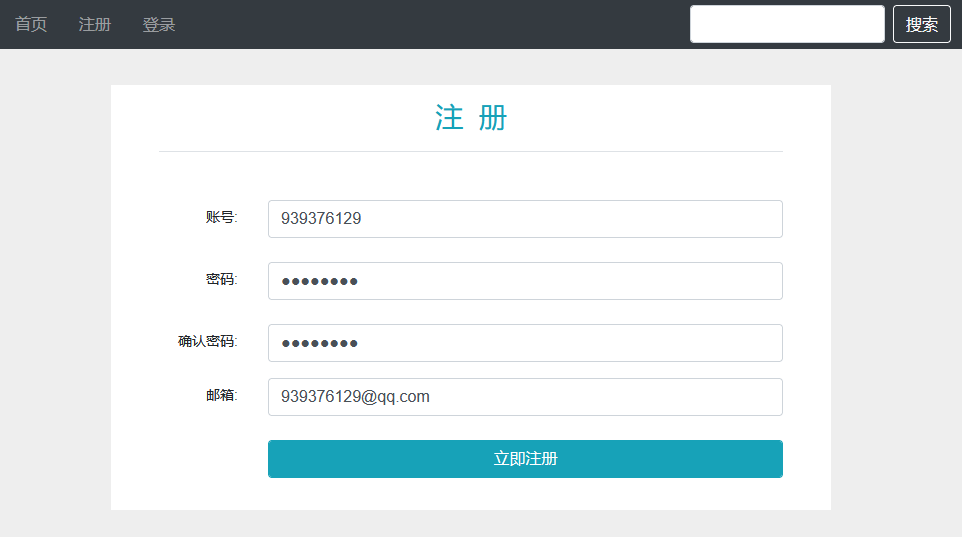


图4.3注册页面



图4.4发送激活邮件



图4.5接收激活邮件



图4.6激活账号成功

**生成验证码：**

在用户登录的时候，在登录页面展示验证码，要求用户必须输入验证码且输入正确验证码，才可以登录成功，这样做是为了防止机器程序恶意操作。

首先在 pom.xml配置文件中导入 kaptcha 工具类jar包的坐标，让SpringBoot加载此工具类。

新建验证码配置类：KaptchaConfig，用于配置验证码的各个属性，如：范围、长度、大小等。

在登录的视图层类中新建方法： getKaptcha ，用来获取生成的验证码图片。

在文件：login.html 中，将验证码img标签的src属性替换成LoginController中的getKaptcha的RequestMapping路径，更新校验码的超链接绑定。请点击事件。在kaptcha js方法中，通过Jquery的id选择器取得认证代码组件，并点击该组件来变更刷新认证代码，从而以访问Src属性的背景getKaptcha的方法生成新的验证代码图片。



图4.7验证码效果

登录及退出登录功能

系统判定用户是否登录成功是在用户登录信息正确无误后响应给用户一个带着登录凭证的Cookie，登录状态就是用户本次访问携带此Cookie，并且里面的凭证是有效的。要实现退出登录功能，就是将此Cookie设置为无效即可，比如让此Cookie的时间立即过期，此时当客户端再次访问，就不会携带此Cookie，服务器自然就判定此用户没有登录。

存放登录凭证的数据库表是login\_ticket，其中expire字段表示过期时间，status表示用户登录是否有效。

创建对应与表对应的Bean实体类： LoginTicket，用于封装查询数据库的信息。

创建访问持久层接口文件：LoginTicketMapper，并定义关于用户凭证的增、查、改三个方法，因为这三个的SQL语句并不复杂，所以分别为其标注@Insert、@Select、@Update ，并在注解中写出对应的SQL语句，免去了创建xml文件的步骤。

在 UserServce中，新建login方法，改方法首先验证客户端传入的用户名密码是否正确，如果正确则将合规的用户的凭证信息系插入到数据库login\_ticket表。新建logout调用LoginTicketMapper更新凭证方法，将凭证状态置为失效状态。

在控制层中，接收登录请求，并访问UserServce的login方法，获得一个map的返回值对象，在视图层来判断map对象中是否包含名为“ticket”的键值，如果包含，则代表登录成功，此时重定向到首页。若不包含则表示登录失败，此时回显错误信息到登录页面。



图4.8登录页面

**显示登录信息：**

新建工具类Cookie.Util，新建方法getValue通过name查询. cookie对应的value值的方法。

新建类HostHolder来模仿session功能，利用 ThreadLocal实现不同用户的信息存储在不同的线程中。.

新建登录拦截器类，使其实现SpringMVC提供的接口文件：HandlerInterceptor，并重写里面的方法。

在preHandle 方法中，首先查询客户端请求的Cookie中是否包含凭证信息，如果查询到有效凭证，那么调用UserService的方法 findloginTicket查询用户的 ID，再通过用户的ID查出对应的User对象，最后将User对象放入hostHolder对象中。

在postHandle方法中，去访问hostHolder对象，获取对应的惟一的User对象，并将User对象存入到视图中。

在afterCompletion方法中，调用hostHolder对象，清除存放的User信息。



图4.9登录信息展示

**上传头像**

在用户的业务层中新增方法updateHeader，修改数据库用户表中表示用户头像的字段。

新建UserController类

新建方法getSettingPage设置返回值路径指向setting.html，同时在index.html页面的账号设置处关联此链接。

新建方法uploadHeader，来上传用户头像到服务器中来，利用SpringMVC提供的MultipartFile类来加载上传图片，如果头像图片在上传过程中出现了异常，那我们就记录一下异常日志，并将异常的错误信息存储在Model对象中，最后响应到用户的个人信息设置页面上，用户就可以直观的观察到错误。如果没有出现错误，则利用 MultipartFile类的transferTo方法将头像文件上传到服务端，而且要从hostHolder对象中取出对应User对象，更新User对象的头像路径。

新建getHeader方法来获取用户的头像，先读取图片的字节流数据，再使用Response对象的字节输出流向页面输出回显。

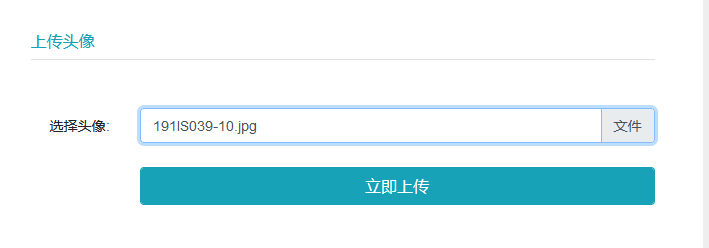


图4.10图片上传页面



图4.11图片成功效果展示

**修改密码：**

控制层调用业务层修改密码方法，在业务层中新建方法changePassword，判断是否满足修改密码条件并在满足条件情况下去访问数据库进行修改，返回给控制层一个状态码：1表示修改成功， 0表示修改失败，控制层接收并判断状态码并显示对应的成功/失败信息给用户页面。



图4.12修改密码页面

**检查登录状态：**

在项目实际应用中，像个人设置个人主页这样的页面，要确保当前用户已经登录了才能访问，使用拦截器，在用户访问需要登录的页面时候，进行登录状态的判断，只有用户登录状态为真的时候才可以顺利访问，否则使其重定向到登录页面。

新建一个自定义注解，取名为@LoginRequired，为其标注原注解，使其在运行时生效，只能标注在方法上。

在控制层一些登录后才有权限访问的方法上，例如修改密码、头像，添加声明注解@LoginRequired。

新建登录拦截器，在方法执行开始前进行拦截，通过反射，判断拦截的方法上是否被标注注解：@LoginRequired，如果查询到此注解，则去hostHolder对象中本请求对应的User对象，如果能获取的到说明用户已经登录，如果没有获取到，则说明用户没有登录，应当拒绝本次请求，并重定向到登录页面。

4.3 开发核心功能

**发布帖子：**

客户端发送帖子首先要向服务端发送一条POST请求，请求体采用JSON的数据格式，方便后台读取并解析，后台使用的解析工具是阿里巴巴的fastJson，首先要在pom.xml文件中引入此工具类的坐标，其次为了方便我们的调用，在工具类：CommunityUtil中定义方法，完成对此工具类的封装，方便后续调用获取JSON信息。

在控制层中，新增方法addDiscussPost接收用户发送过来的请求数据，封装对象，并调用Service中的 addDiscussPost实现发帖流程。

从控制层获取到用户发表的帖子JSON后，要将其插入到数据库中去，持久层新增方法insertDiscussPost，在mapper.xml文件中编写对应的insertSQL语句。



图4.13发布帖子页面

**显示帖子内容：**

显示帖子内容首先要查询出来帖子内容，如果是显示单个用户发布的帖子，则需要带上用户id去查询数据库。

在控制层新增方法getDiscussPost用于接收处理客户端浏览器发来的请求，在调用业务层findDiscussPostById方法，业务层中处理逻辑并调用数据库返回给控制层，控制层接收到帖子详情信息和用户信息后，存储在model中，由Thymleaf模板引擎解析并回显到用户的页面中。

**显示评论：**

首先新建评论类JavaBean对象Comment。

再创建持久层接口CommentMapper，新增根据Comment对象查询一页的评论数据、根据Comment对象查询评论的数量。

在控制层中新增方法：getDiscussPost查询帖子的评论详细和回复详细，将最后取得的对象存在Model对象中，，由Thymleaf模板引擎解析并回显到用户的页面中。

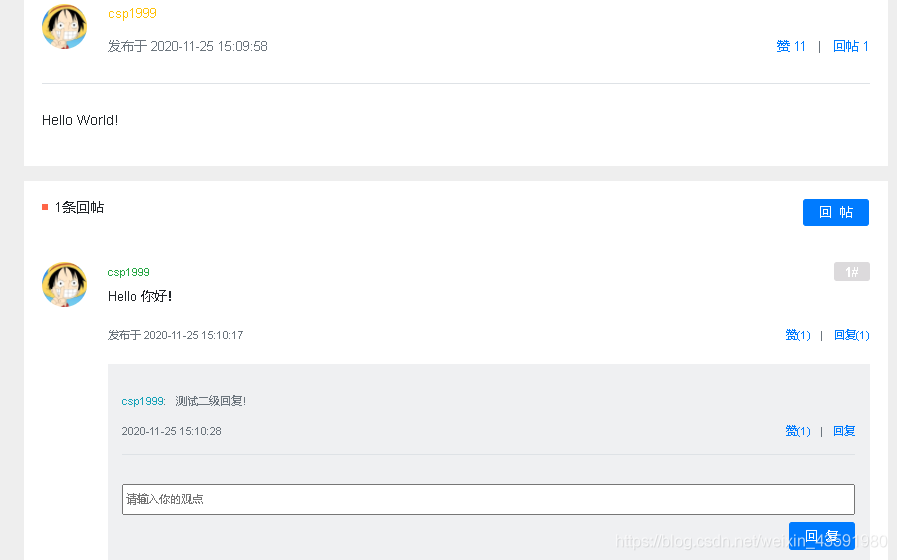


图4.14显示评论效果展示

**添加评论：**

添加评论就是将客户端信息封装成Comment对象，再由用系统后台插入到数据库中，与查询很是相似，就是讲select语句换成了insert语句。

在控制层，新增方法：addComment，从hostHolder类中获取当前登录用户的个人信息，再调用业务层方法addComment，根据用户信息添加评论。

**显示私信列表：**

首先创建存储私信的数据库表对应的实体类Message，

创建持久层接口文件，增加一系列查询方法，包括会话列表方法，查询会话数量方法，查询私信列表方法，查询私信数量方法和查询未读私信数量方法等，并在mapper.xml中配置与之对应的 select SQL语句。

在控制层新增方法：getLetterList接收用户请求，访问业务层查询，返回查询出的Message对象，将最后取得的对象存在Model对象中，由Thymleaf模板引擎解析并回显到用户的页面中。

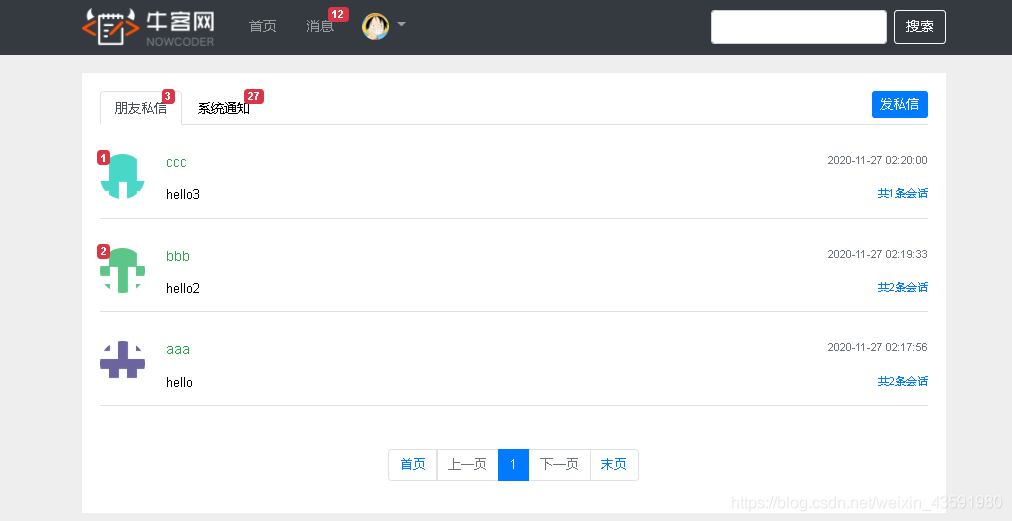


图4.15私信列表页面

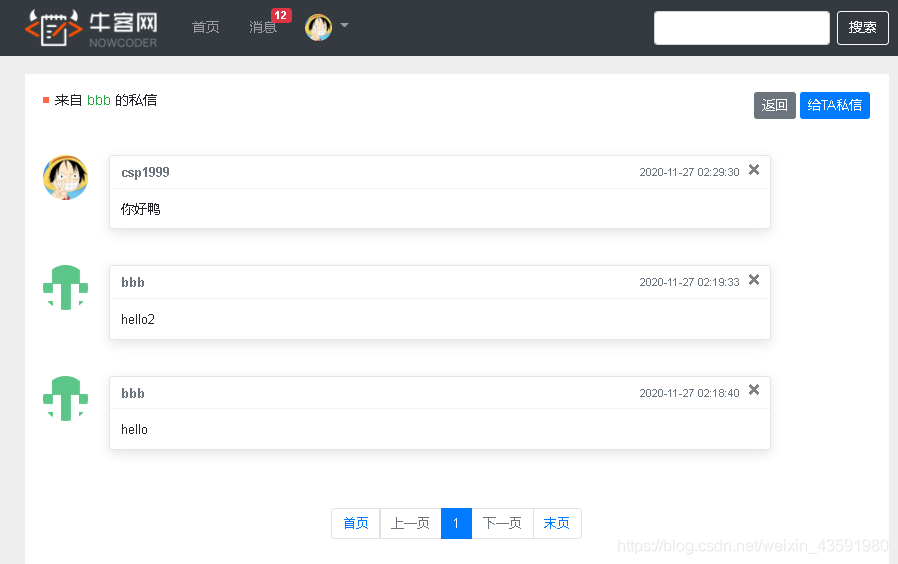


图4.16详细私信页面

**发送私信：**

发送私信就是将客户端信息封装成Message对象，再由用系统后台插入到数据库中，与查询很是相似，就是将select语句换成了insert语句。

在控制层中，新增方法：getLetterIds，其作用是将一些未读状态的私信，整合到一个列表中，并返回。新增方法：sendLetter，其作用是处理用户发送私信的请求，调用业务层方法addMessage实现私信的发送流程。

结束语

通过这次的毕业设计，完成了系统整体的需求分析和结构。完成以上需求分析后，根据系统计划的需要，根据之前制作的详细设计代码，对系统进行了完全的测试。我测试了注册、登录、发帖、评论等功能。进行了反复测试。最后，社区论坛系统各功能正常运行。

在开发系统前要做足系统分析，晚点动手做也不迟，系统分析的越简单越抽象，那么在之后的编码工作中遇到的困难就会越多，所以做足分析并不是在浪费时间，而是为了之后更快更效率的开发。此外严格按照开发规范编写代码很重要，这样可以在后面修改或者新增需求的时候改动最少的文件，就可以达成需求。

对于本系统来说，实现了大部分论坛必要的功能，但也有很多地方要改进，就UI界面来说，不是很美观，没有很多交互动画，因为对于前端方面专业知识和相关技术的不足，没能做到这些。此外系统的高合并性、系统的安全性等还是有很多不足的方面，其他的功能也等待后续实现。

回顾设计开发毕业设计系统的过程，还是蛮辛苦的，不过投入的心血越多，得到的充实快乐也就越多。

参考文献

[1]钟玉琢,沈洪,吕小星等.多媒体技术及其应用[M].北京:机械工业出版社.2013

[2]周向东,施伯乐等.基于长期学习的多媒体数据库相似性检索[J].软件学报.2014,15（1）：32-39

[3]沈琦,张虹.多媒体数据库模型与存储模型研究[J].计算机工程.2007,23（6）：23-30

[4]刘玉照,黄蕾.多媒体数据库模型及实现途径之比较研究[J].情报科学.2011,19（8）：853-857

[5]郑庆华,李人厚,胡俊.多媒体数据库的主要问题、研究内容与实现方法[J].计算机工程与应用.2015,32（1）：23-30

[6]黄蕾.多媒体数据基于内容检索综述[J].情报探索.2000,4（76）：8-13

[7]黄丽娟.基于内容的多媒体信息检索[J],情报杂志.2010,19（5）：47-51

[8]李国辉,曹莉华,柳伟.基于内容的多媒体数据查询和检索[J].小型微型计算机系统.1998,19（4）：23-30

[9]郝小花.多媒体信息中基于内容的检索[J].情报理论与实践.1999,22（3）：23-30

[10]李国辉,胡小峰.基于内容的检索[J]．计算机世界．1998，D1-D3：23-30

[11]徐建华.一种新型的多媒体检索技术----基于内容的检索[J].情报学报.2000,19（4）：53-58

[12]王涛等.融合型多媒体数据库模型的实现技术[J].焦作工学院学报.2003,22（6）：13-20

[13]李国辉,曹莉华等.多媒体信息查询和检索系统MIRC[J].小型微型计算机系统．2009，20（9）：27-40

[14]何清法,李国杰.综合分块主色和相关反馈技术的图象检索方法[D].北京:中国中科学院.2000

[15]王玉波.多媒体信息检索技术概论[J].情报科学.2008,17（2）：56-60

附录

业务层注册方法代码

*/\*\*  
 \*   
 \** ***@Title*** *register  
 \** ***@Author*** *付立威  
 \** ***@Description*** *注册逻辑   
 \** ***@Date*** *17:41 2021-05-12  
 \** ***@Param*** *User user  
 \** ***@return*** *Map*<*String*,*Object*>  
 *\*\*/***public** Map***<***String,Object***>*** register***(***User user***){*** Map***<***String,Object***>*** map = **new** HashMap***<>()***;  
  
 *//判断空值* **if*(***user==**null*){* throw new** IllegalArgumentException***(*"参数不能为空！"*)***;  
 ***}* if*(***StringUtils.*isBlank****(***user.getUsername***())){*** map.put***(*"usernameMsg"**,**"账号不能为空！"*)***;  
 **return** map;  
 ***}* if*(***StringUtils.*isBlank****(***user.getPassword***())){*** map.put***(*"passwordMsg"**,**"密码不能为空！"*)***;  
 **return** map;  
 ***}* if*(***StringUtils.*isBlank****(***user.getEmail***())){*** map.put***(*"emailMsg"**,**"邮箱不能为空！"*)***;  
 **return** map;  
 ***}*** *//判断账号是否存在* User userFromDB = **userMapper**.selectByName***(***user.getUsername***())***;  
 **if*(***userFromDB!=**null*){*** map.put***(*"usernameMsg"**,**"该账号已存在！"*)***;  
 **return** map;  
 ***}*** *//判断邮箱是否存在* userFromDB = **userMapper**.selectByEmail***(***user.getEmail***())***;  
 **if*(***userFromDB!=**null*){*** map.put***(*"emailMsg"**,**"该邮箱已被注册！"*)***;  
 **return** map;  
 ***}*** *//数据没有问题，开始注册用户* user.setSalt***(***CommunityUtil.*generateUUID****()***.substring***(***0,5***))***;  
 user.setPassword***(***CommunityUtil.*getMD5****(***user.getPassword***()***+user.getSalt***()))***;  
 user.setActivationCode***(***CommunityUtil.*generateUUID****())***;  
 user.setHeaderUrl***(***String.*format****(*"http://images.nowcoder.com/head/%dt.png"**,**new** Random***()***.nextInt***(***1000***)))***;  
 **userMapper**.insertUser***(***user***)***;  
  
 *//激活邮件* Context context = **new** Context***()***;  
 context.setVariable***(*"email"**,user.getEmail***())***;  
 String activationUrl = **domain**+**contextPath**+**"/activation/"**+user.getId***()***+**"/"**+user.getActivationCode***()***;  
 context.setVariable***(*"activationUrl"**,activationUrl***)***;  
 String content = **templateEngine**.process***(*"/mail/activation"**,context***)***;  
 **mailClient**.sendMail***(***user.getEmail***()***,**"请激活您的论坛账号"**,content***)***;  
  
 **return** map;  
***}***

业务层登录方法代码

*/\*\*  
 \*  
 \** ***@Title*** *login  
 \** ***@Author*** *付立威  
 \** ***@Description*** *登录逻辑  
 \** ***@Date*** *18:31 2021-05-12  
 \** ***@Param*** *String username, String password, int expiredSeconds  
 \** ***@return*** *Map*<*String*,*Object*>  
 *\*\*/***public** Map***<***String, Object***>*** login***(***String username, String password, **int** expiredSeconds***) {*** Map***<***String, Object***>*** map = **new** HashMap***<>()***;  
  
 *// 空值处理* **if *(***StringUtils.*isBlank****(***username***)) {*** map.put***(*"usernameMsg"**, **"账号不能为空!"*)***;  
 **return** map;  
 ***}* if *(***StringUtils.*isBlank****(***password***)) {*** map.put***(*"passwordMsg"**, **"密码不能为空!"*)***;  
 **return** map;  
 ***}*** *// 验证账号* User user = **userMapper**.selectByName***(***username***)***;  
 **if *(***user == **null*) {*** map.put***(*"usernameMsg"**, **"该账号不存在!"*)***;  
 **return** map;  
 ***}*** *// 验证状态* **if *(***user.getStatus***()*** == 0***) {*** map.put***(*"usernameMsg"**, **"该账号未激活!"*)***;  
 **return** map;  
 ***}*** *// 验证密码* password = CommunityUtil.*getMD5****(***password + user.getSalt***())***;  
 **if *(***!user.getPassword***()***.equals***(***password***)) {*** map.put***(*"passwordMsg"**, **"密码不正确!"*)***;  
 **return** map;  
 ***}*** *// 生成登录凭证* LoginTicket loginTicket = **new** LoginTicket***()***;  
 loginTicket.setUserId***(***user.getId***())***;  
 loginTicket.setTicket***(***CommunityUtil.*generateUUID****())***;  
 loginTicket.setStatus***(***0***)***;  
 loginTicket.setExpired***(*new** Date***(***System.*currentTimeMillis****()*** + expiredSeconds \* 1000***))***;  
 **loginTicketMapper**.insertLoginTicket***(***loginTicket***)***;  
  
 map.put***(*"ticket"**, loginTicket.getTicket***())***;  
 **return** map;  
***}***

致谢

首先，经过几个月的研究和努力，系统功能已经基本实现，虽然在系统开发过程中遇到了很多困难，但在导师和好同学好朋友的帮助下，遇到的的坎坷将一个个迎刃而解。在解决问题的过程中，我们积累了丰富的经验，学到了很多实用技术和专业知识，收获了很多，在具体的系统实施过程中，查阅了大量的参考文献和技术博客，在此，我对相关文献的作者表示衷心的感谢。

其次，在系统开发过程中，在教师的指导下，对系统的需求进行了分析，在系统模型的构建和功能模块的划分上，教师提出了重点建议，按照低集成度、高内部集成度的基本原则，系统性能有保证，谢谢您的指导。

最后，我感谢学校给了我设计毕业舞会的机会。我再次衷心感谢老师和同学们的慷慨建议。因为你的帮助，我能顺利完成毕业设计。