



本科毕业设计（论文）

**经验分享web应用开发**

**学 院 计算机学院**

**专 业 计算机科学与技术**

**年级班别 2018级（1）班**

**学 号 3118004667**

**学生姓名 陈更成**

**指导教师 朱鉴**

**2022年 4 月**

**经验分享web应用开发 陈更成 计算机学院**

**摘要**

在互联网技术高度发达的今天，从网络获取知识与经验已经成为家常便饭，无论从事什么行业，当遇到技术难题或者需要寻求经验支持时，自然而然的会想到从网络上获取相关信息。当前从网络获取信息的方式主要有：查找专业论文和书籍等高质量内容、查找相关问答和博客等由其他内容创作者提供的信息。对于查找专业论文和书籍这种方式，主要是从论文的研究结果中获取需要的信息，这种方式主要是针对所需知识专业性比较强的情况，对于大多数场景而言，主要采用第二种方式。

当前主流的经验分享平台多为综合性的知识分享平台，不区分用户的需求，不对内容进行分类，我们在查找所需要的内容过程中，很多情况下都会被热搜信息、推荐内容所吸引、绝大多数内容分享平台在商业化之后都存在严重的广告等问题，同时当前专业性的论坛系统并不多，主要集中在计算机技术相关领域，其他专业领域几乎没有集中的知识分享平台。

针对当前市场环境中经验分享平台的现状，通过SSM框架、Bootstrap、Redis、Kafka、ElasticSearch等技术，实现了一个简洁美观、操作简便的专业性经验分享平台。该平台旨在为有专业性知识需求的用户提供一个分享经验、总结交流学习心得的平台。

**关键词：**论坛系统 ，SSM框架，Bootstrap框架，B/S架构

**Abstract**

In today's highly developed Internet technology, it has become commonplace to acquire knowledge and experience from the Internet. No matter what industry you are engaged in, when you encounter technical problems or need to seek experience support, you will naturally think of obtaining relevant information from the Internet. The main ways to obtain information from the Internet are to find high-quality content, such as professional papers and books, relevant questions and answers, and information provided by other content creators, such as blogs. For searching professional papers and books, the method is mainly to obtain the required information from the research results of the paper. This method is mainly aimed at the situation where the required knowledge is relatively professional. For most scenes, the second method is mainly adopted.

The current mainstream of experience sharing platform for more comprehensive knowledge sharing platform, do not distinguish between the needs of users, not content to classify, we needed in finding content in the process, in many cases will be attracted by hot search information, recommendations, the vast majority of content sharing platform in commercialization after there are serious problems such as advertising, At the same time, there are not many professional forum systems at present, mainly focusing on computer technology-related fields, while there are almost no centralized knowledge sharing platforms in other professional fields.

In view of the current situation of experience sharing platform in the current market environment, a simple and beautiful professional experience sharing platform is realized through SSM framework, Bootstrap, Redis, Kafka, ElasticSearch and other technologies. The platform aims to provide a platform for users who need professional knowledge to share experience, summarize and exchange learning experience.

**Key words:** Forum system ，SSM，Bootstrap，B/S framework

目录

[1 绪论 1](#_Toc101995753)

[1.1 研究背景与意义 1](#_Toc101995754)

[1.2 研究内容 2](#_Toc101995755)

[2 需求分析 3](#_Toc101995756)

[2.1 可行性分析 3](#_Toc101995757)

[2.2 功能需求分析 4](#_Toc101995758)

[2.3 非功能性需求分析 8](#_Toc101995762)

[2.4 本章小结 9](#_Toc101995763)

[3 概要设计 10](#_Toc101995766)

[3.1 系统架构设计 10](#_Toc101995767)

[3.2 功能模块设计 11](#_Toc101995768)

[3.3 数据库设计 15](#_Toc101995769)

[3.4 本章小结 20](#_Toc101995770)

[4 详细设计 21](#_Toc101995771)

[4.1 注册登录模块详细设计 21](#_Toc101995772)

[4.2 帖子模块详细设计 23](#_Toc101995773)

[4.3 消息模块详细设计 28](#_Toc101995774)

[4.4 系统设置模块详细设计 31](#_Toc101995775)

[4.5 搜索模块和后台管理模块详细设计 33](#_Toc101995776)

[4.6 本章小结 35](#_Toc101995777)

[5 编码实现 36](#_Toc101995778)

[5.1 开发环境 36](#_Toc101995779)

[5.2 各功能模块实现 36](#_Toc101995780)

[5.3 本章小结 45](#_Toc101995781)

[6 测试 46](#_Toc101995782)

[6.1 测试环境 46](#_Toc101995783)

[6.2 功能测试 46](#_Toc101995784)

[6.3 非功能测试 49](#_Toc101995785)

[6.4 本章小结 50](#_Toc101995786)

[7 总结与展望 51](#_Toc101995787)

[7.1 总结 51](#_Toc101995788)

[7.2 展望 51](#_Toc101995789)

[参考文献 53](#_Toc101995790)

[致谢 54](#_Toc101995791)

# 绪论

该部分主要介绍Eshare经验分享平台的课题相关背景以及实现该系统的现实意义，同时对比国内外相关的主流经验分享平台的具体功能，结合实际的项目需求，提出当前课题的主要研究内容。

## 研究背景与意义

第48次《中国互联网络发展状况统计报告》显示，截至2021年6月，中国网民规模达10.11亿，较2020年12月增长2175万，互联网普及率达71.6%。如此庞大的网民数量之下，互联网上的信息总量出现了爆炸式的增长。丰富的信息为我们提供了许多前辈的经验，无论是考研、就业、还是考公，我们都会习惯在网络上寻找各种经验贴，从过来人的经历中汲取经验教训，确立自己的学习方向和路线。但是现存的论坛、博客系统体量大多都很庞大，其中充斥着各种信息，很容易在寻找自己所需要的信息过程中被一些娱乐八卦，新闻或者热门话题所干扰，极大的降低了学习效率。现对当前主流的论坛系统等经验分享平台存在的不足进行详细的分析：

（1）与百度知道、知乎类似的问答型平台，这些平台经过长时间的发展，用户规模已经非常庞大，几乎达到了无人不知的程度，任何人都可以通过注册账号发表自己的问题和回答，但也正因发布信息的低门槛，导致了整体内容质量逐年下滑，内容杂乱无章，对于专业性知识相关问题无法提供有效的帮助，或者寻找信息的时间成本太高。

（2）内容分享平台并非专用知识分享，一些娱乐八卦，新闻等内容也被发布到其中，这些无关信息，尤其是热搜等标题具有强烈吸引力的信息，对于查找专业内容的使用者而言无疑产生了巨大的干扰，极大的影响了寻找目标信息的效率。

（3）与CSDN、博客园等类似的计算机专业知识博客平台，这些平台提供专业的知识问答，但仅限于计算机技术相关的内容，对于计算机相关行业的人来讲是一个优秀的内容分享网站，但对于其他专业的从业者而言，CSDN等计算机行业论坛或者博客系统并不能满足他们的需求。

## 研究内容

利用SSM框架，完整的实现一个专门用于交流分享各种专业知识的平台。论坛应该以简约，小巧，高质量为基本目标，实现帮助内容创作者方便快捷的编写经验贴，为内容的使用者创建一个专注于学习交流，没有各种不必要的干扰信息的经验交流平台。现将Eshare经验分享平台的具体研究内容总结如下：

（1）针对当前主流的博客平台内容杂乱、信息数量繁杂、存在大量冗余功能、无法及时寻找到专业性知识相关内容的问题，设计一个界面简洁、专门用于专业性知识分享、具备内容检索功能的论坛系统。

（2）针对当前问答平台的使用者零门槛，导致平台信息多而质量低，甚至出现言语攻击等行为的问题，Eshare经验分享平台应该对不同使用身份的用户进行权限控制，对于专业领域的人才有发布内容的资格，其他用户只能浏览或者评论通过专业认证的用户所发布的内容。

# 需求分析

需求分析对于任何一个软件系统的开发过程都有着决定性的导向作用。软件系统开发的初衷和存在的意义就是解决现实中的问题，例如简化繁琐的业务流程，减少重复劳动，集中整合资源等。在进行实际开发之前，进行全面、准确的需求分析，搞清楚软件要完成什么工作，需要具备什么功能，存在的意义是什么，才能最大限度的避免最终产品达不到想要的效果，不得不多次大规模调整项目结构，甚至重新进行开发。一旦出现这种情况，毫无疑问会使整个项目的开发周期延长进而错失合适的上线时机，项目的成本激增等问题。因此，在项目开发之前，必须从系统的可行性，功能需求等各个方面进行调查分析，在对软件的需求有了清晰的定义和认识之后，才能开始具体的开发工作。

## 可行性分析

可行性分析的实质就是在确定软件存在的实际意义和基本功能的基础上，设想软件开发阶段和发布之后的运行阶段可能出现什么问题，能否应对这些问题。例如在开发阶段主要考虑软件开发的关键技术是否成熟，根据开发周期考虑开发的经济成本；项目发布运行后，可能面临的同类产品的市场竞争，如何吸引用户等因素。这个过程显然不是一蹴而就的，而是需要与整个需求分析的过程一起，不断的调整改进。可行性分析的目的，不是肯定或否定一份需求分析是否合理，而是要辅助制定一份合理的需求分析。

综合考虑产品的实际需要，本项目主要进行以下几点可行性分析：

1. 市场可行性

Eshare经验分享平台是一个论坛系统，当前比较热门的论坛系统中娱乐性的偏多，技术性的论坛多集中在计算机技术，但是论坛作为一个交流经验，分享资源的平台，各个专业其实都需要，论坛的核心吸引力其实并不是论坛平台本身，而是其中的高质量内容。虽然软件系统的开发只是一个成功的论坛系统的第一步，但不可否认的是，当前网络上的信息数量激增，但信息质量层次不齐，需要高质量的信息平台，尤其是用于高质量专业性知识交流的信息平台。Eshare经验分享平台不是要做一个多专业综合的问答平台，而是作为一个基础模板，每个专业领域都在此基础上结合该专业的特点，添加合适的个性化功能，构建独立的该专业领域内的知识交流平台。

1. 技术可行性

Eshare经验分享平台的核心模块将采用当前广泛应用的SSM框架，SSM框架的MVC分层设计模式对于简化开发流程，设计清晰的代码逻辑结构，提升开发效率的作用非常显著。同时，对于点赞、关注等特殊业务需求，将使用Redis存储相关数据，以提升响应速度，节省存储空间。对于私信、系统通知等消息模块相关的功能，将采用当前最流行的分布式消息队列Kafka。对于搜索模块，采用主流的ElasticSearch。这些技术都是被许多知名公司采用的技术，都经过了长时间的迭代，不仅可靠性高，同时具有高度抽象的API接口，使用方便。

1. 经济可行性

Eshare经验分享平台的核心功能比较简单，属于轻量级的web应用，项目整体代码量较少，完全可以独立开发，同时，项目中使用的技术多为开源技术，项目上线后可以采用云服务器，根据用户数量动态调整购买的服务器资源，以此减少服务器成本。

## 功能需求分析

功能性需求分析是为了确定系统应该具备哪些功能模块而进行的，为了做好功能性需求分析，首先需要明确系统的使用者分为哪几类，根据用户类型确定每种用户需要的系统功能。

### 用户角色分类

Eshare经验分享平台的用户主要分为三种类型：普通用户、普通管理员、后台管理员。普通用户就是正常的论坛系统使用者，是最主要的用户，普通管理员也是论坛的使用者，是数量有限的、经过评选的普通用户，主要工作是确保论坛系统中的内容质量，维护论坛环境和氛围，后台管理员是论坛系统的运营者和维护者，拥有论坛系统的最高权限。

1. 普通用户的功能需求：

 登录（登录已有账号、注册账号、激活账号）

 帖子（查看帖子、发布帖子、评论帖子、点赞帖子、搜索帖子）

 评论（发布评论、点赞评论、回复评论）

 消息（发送私信、查看私信、查看系统通知）

 个人主页（关注列表、粉丝列表、获赞列表）

 账号设置（修改头像、修改密码）

 退出登录

1. 普通管理员的功能需求：

 帖子管理（帖子置顶、设置帖子为精华帖）

1. 后台管理员的功能需求：

 帖子管理（删除帖子）

 网站数据统计（独立访客数UV、日活跃用户DAU）

### 系统功能分类

基于上一小节中的不同用户角色的功能需求，确定论坛系统的基本功能模块应该包括：注册登录模块、帖子模块、消息模块、系统设置模块、搜索模块、后台管理模块。各模块的详细描述如下：

1. 注册登录模块

账号登录模块包含两个主要功能：账号注册、账号登录。用户在注册时，每个邮箱只能绑定一个账号。为了防止恶意注册其他人的邮箱，账号在注册后，并不能立即使用，而是先由系统发送给注册用户一个激活邮件，起到确认身份的作用，用户通过激活邮件中的激活链接激活账号之后方可正常登录。用户在登录时，需要输入提示的图片验证码，以保证账号安全。

1. 帖子模块

帖子模块的核心功能有：展示帖子列表、发布帖子、查看帖子作者主页、帖子详情页功能（为帖子点赞、评论帖子、回复评论、对评论点赞）、需要管理员权限的帖子管理模块（设置帖子为精华帖、指定帖子、删除帖子），帖子模块的用例图如下：

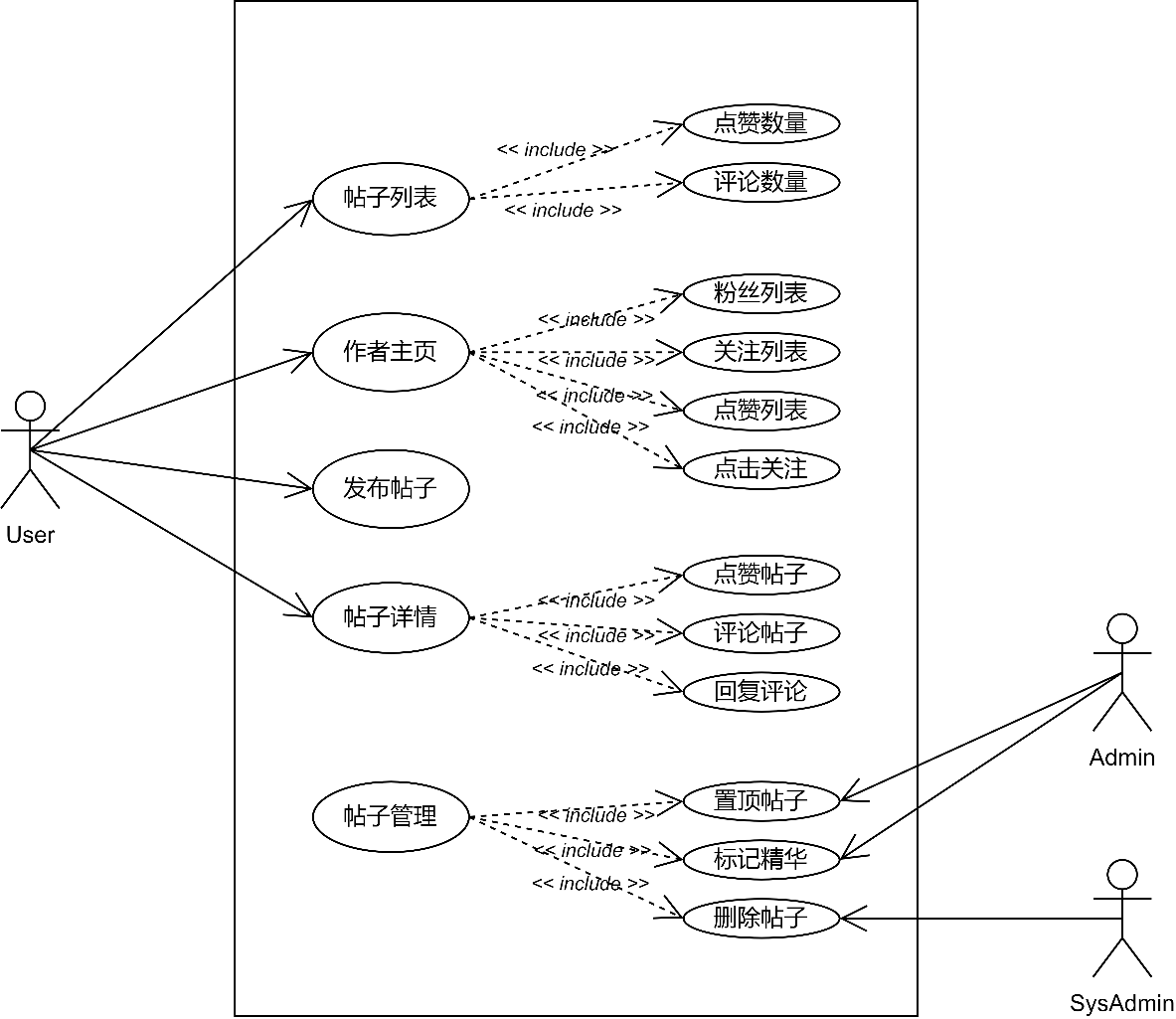


图 2.1 帖子模块用例图

1. 消息模块

消息模块的核心功能有：朋友私信功能（在私信详情页面可以进行以下操作：显示历史消息列表、回复私信、删除历史消息）、发送私信功能、查看系统通知功能（系统通知主要有三类，包括被评论通知、被点赞通知、被关注通知）。消息模块的用例图如下：

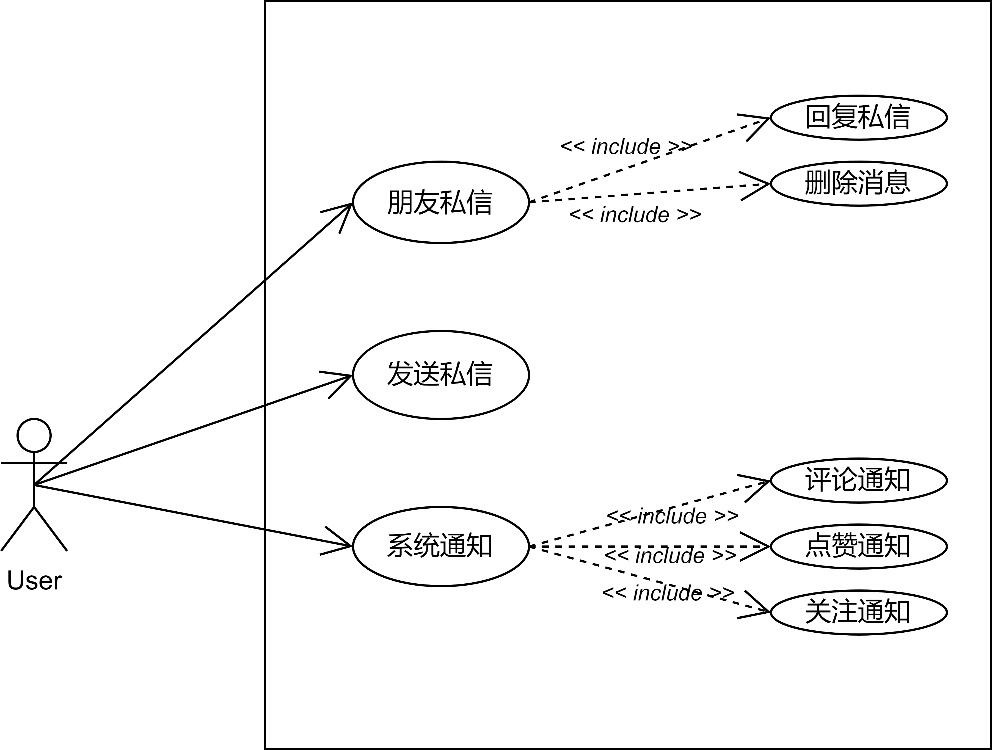


图 2.2 消息模块用例图

1. 系统设置模块

系统设置模块的核心功能有：个人主页（个人主页中又包含三个子模块：个人信息、我的帖子、我的回复）、账号设置功能（包括更改头像、修改密码）、退出登录。其中个人信息页面需要包括以下信息：用户的关注列表、用户的粉丝数量（点击可以查看用户的粉丝列表）、用户获赞数量、关注该用户的的关注按钮。系统设置模块的用例图如图 2.3所示。

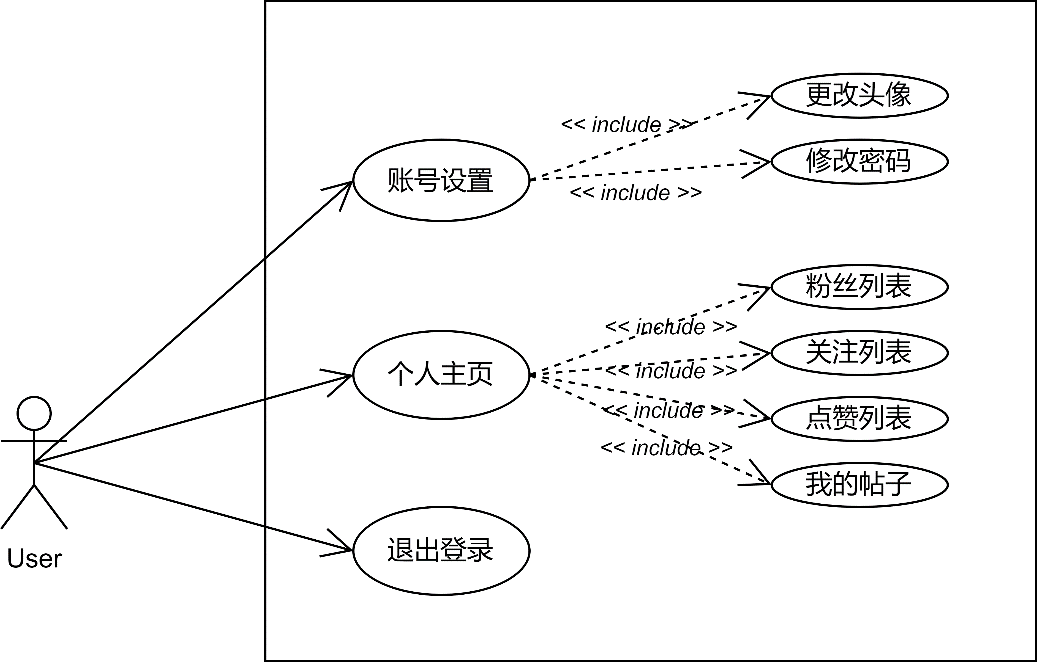


图 2.3 系统设置模块用例图

1. 搜索模块

搜索模块需要在首页显示搜索框，搜索的范围包括论坛中的帖子以及帖子的标题。搜索结果以列表的形式显示，每一项都显示完整的帖子内容、作者信息等基本信息。

1. 后台管理模块

后台管理模块主要是针对系统管理员的模块，在前端页面不需要设置入口，系统管理员需要访问时，直接通过管理员路径访问。后台管理模块包括两个主要功能：统计论坛系统一定时间范围内的独立访客数、统计一定时间范围内的活跃用户数。后台管理模块的用例图如下：

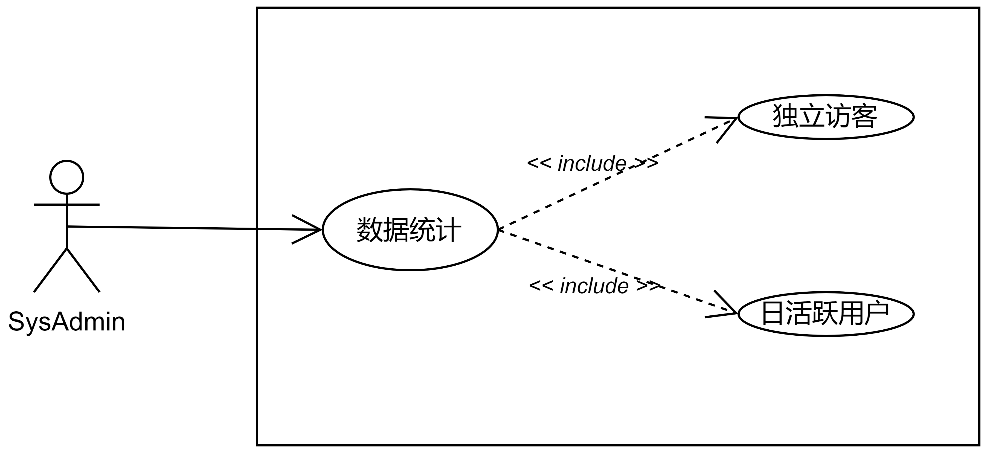


图 2.4 后台管理模块用例图



## 非功能性需求分析

对于大型软件系统而言，非功能性需求分析与功能性需求分析同样重要，尤其是对于目标是商业化的，需要有大量用户访问的系统，非功能性需求分析是必不可少的。非功能需求分析中最关注的点就是系统性能和系统安全，面对高并发的现实场景，系统如何设计才能承载，面对常见的网络攻击，采用何种策略才能保证数据安全。结合现实需求，Eshare论坛系统的非功能性需求分析主要考虑以下几点：

1. 平台性能

论坛系统并不像电商系统一样，需要经常举办活动，因此经常面临高并发的场景。论坛系统的用户访问变化趋势一般比较平稳，不需要过多的考虑高并发的场景。但是，这并不意味着论坛系统不需要考虑性能。论坛系统的性能主要是用户日常访问响应的及时性，包括帖子数据加载、评论回复、点赞等操作的及时响应。系统设计以及选择实现技术时，要着重考虑这些问题。

1. 平台操作友好性

几乎所有的软件系统都把简化操作流程，方便用户使用作为设计的重要考量因素。论坛系统本身涉及的操作并不多，但是涉及的页面却非常多，在设计时需要重点考虑多页面之间跳转的逻辑关系，通过参考现有的优秀论坛的用户交互设计，在保证界面美观的基础上，尽最大可能减少用户的操作次数。

1. 平台数据安全

论坛系统中的数据大部分情况下并不敏感，论坛系统的数据安全主要是考虑到以下两点：第一，随着数据分析技术的发展，通过系统的、连续的数据分析用户行为、性格，已经变得越来越简单，论坛作为用户直接表达想法，交流学习的平台，应该注意防止恶意爬取同一用户所有帖子、评论等信息用于分析用户行为。第二，当前软件系统账号安全技术虽然已经比较成熟，但是盗用他人账号的行为依然层出不穷，所以如何设计有效的账号安全保护机制，防止恶意利用他人账号发布错误言论，也是论坛系统应该考虑的问题。

## 本章小结

本章在查阅了大量相关资料的基础上，从论坛系统的可行性，功能和性能等多个方面对Eshare经验分享平台的用户类型、系统应该具备的主要模块进行了系统、全面的分析，通过多次调整修改，比较好的确定了论坛系统的基本功能。



# 概要设计

概要设计和详细设计是在需求分析之后进行的设计阶段的两个主要内容。其中概要设计可以理解为详细设计做的准备工作，需求分析中得到的软件所应该具备的功能有很多，如何组织这些功能，在程序中如何存储这些功能相关的数据，是概要设计的主要任务。概要设计的主要关注点有三个：软件系统的功能模块、数据结构、数据库表的结构，本章主要就这三个方面展开介绍。

## 系统架构设计

参考目前热门的技术论坛实现方案，Eshare经验分享平台的后端核心模块也将采用流行的SSM框架开发，系统中的消息功能使用kafka消息队列，点赞、关注功能使用Redis缓存，搜索模块采用ElasticSearch。同时为了实现系统界面的一致性，美观性，前端使用Bootstrap框架，前后端交互中需要用到异步请求时采用Ajax，论坛系统的核心数据存储在mysql数据库。

1. 数据存储

当前主流的数据存储方式主要有两大类，关系型数据库（例如Mysql）、非关系型数据库（例如Redis）。根据实际需要将数据存入不同类型的数据库中有利于系统整体性能的提高。Eshare论坛系统中的数据主要分三类：帖子、评论等需要存到关系型数据库中的信息，点赞、关注等需要存到Redis缓存中的数据，用于支持搜索的ElasticSearch中的帖子数据。

1. 持久层

持久层主要定义用于访问数据库等系统中数据的方法，Mybatis中提供了一套用于定义数据库访问方法的规范，使用这些规范可以减少数据访问方法中出现错误的可能性，提高代码的可维护性，所以针对数据库的访问采用Mybatis规范。对于Redis中存储的数据，Spring官方提供了一套定义好的用于操作Redis中数据的类，我们只需要调用API方法，即可完成Redis相关的增删改查操作。对于ElasticSearch中的数据，Spring官方同样提供了一套访问ElasticSearch中数据的接口，针对当前项目的需求，我们不需要自定义访问ElasticSearch中数据的方法。

1. 业务层

业务层是整个系统中最核心的一层，整个论坛系统核心功能的实现是在业务层中完成的。业务层是对数据访问层中方法的进一步抽象，针对于控制层的需要，定义合适的业务层方法，有利于减少代码量，提高代码整体的逻辑性。Eshare经验分享平台中，业务层主要包括以下几类方法：评论、帖子、消息、搜索等核心业务相关，点赞、关注、用户管理相关。

1. 控制层

控制层用于处理前端请求，控制层的本质是Servlet，用于定义请求路径与后端方法的映射，在Spring项目中，这个过程通过注解就可以完成。控制层方法一般与前端页面相关联。

1. 表现层

表现层主要用于直接与用户进行交互，在Eshare经验分享平台中，表现层主要是前端页面，为了保证前端所有页面之间的风格一致、美观，所有页面都是用Bootstrap框架开发。同时是用Thymeleaf模板引擎简化前后端之间的数据交互。

## 功能模块设计

该小节结合需求分析中初步确定的系统功能模块，进一步描述Eshare经验分享平台中各个功能模块的详细信息。Eshare经验分享平台的总体系统模块图示如下：

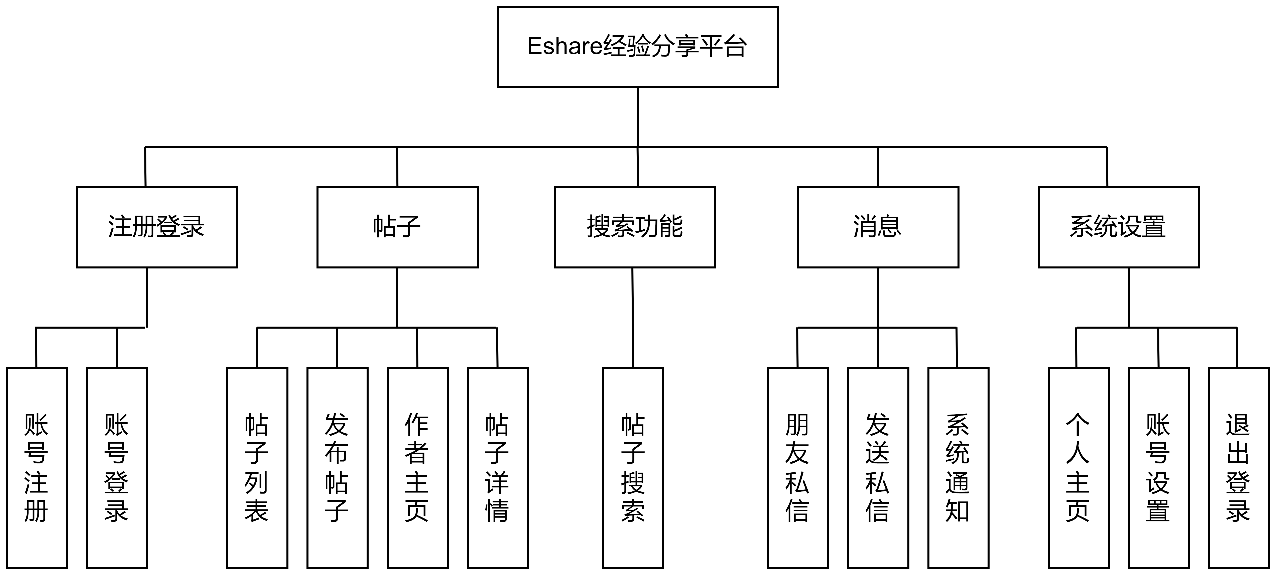


图 3.1 系统功能模块设计图

### 注册登录模块

注册登录模块主要的功能有：账号注册、账号激活、账号登录。各个功能的详细描述如下：

（1）账号注册

账号注册应该要求用户提供想要注册的用户名、密码、用于验证和绑定的邮箱等信息。账号的用户名不能与其他用户重复，账号的绑定邮箱只能绑定一个账号。当满足以上条件后，账号注册成功，向用户提供的邮箱中发送激活链接。

（2）账号激活

用户注册成功后，并不能立即使用注册的账号登录使用，因为至此还未验证用户身份，不排除使用他人邮箱注册的可能。所以要求用户收到激活邮件后，点击邮件中提供的激活链接，激活账号后，才能正常登录。

（3）账号登录

登录模块除了验证账号密码之外，还要求用户输入登录页面提示的图片验证码，将用户输入与后台生成是的记录比较通过，才能登录成功。

### 帖子模块

帖子模块的主要功能有：帖子列表、发布帖子、作者主页、帖子详情等功能。各个功能的具体描述如下。

（1） 帖子列表

帖子列表应该在首页显示，无论用户是否登录，都可查看帖子列表。帖子列表的每一项应该包括的信息有帖子标题、帖子内容的简略信息、帖子作者的头像、帖子作者的用户名、帖子获赞数量、帖子发布时间、帖子评论数量。

（2）发布帖子

发布帖子需要首先有编辑待发布帖子内容的界面，这个界面设计成弹出式的编辑窗口，点击发布后，无论是否发布成功，需要提示用户发布结果，帖子发布后返回首页。

（3）作者主页

作者主页应该包括作者头像、作者用户名、作者粉丝数量、作者关注数量、作者获赞数量、点击关注该作者的关注按钮、作者的注册时间。其中点击作者的关注数量可以查看该用户的关注列表，点击作者的粉丝数量可以获取该用户的粉丝列表。

（4）帖子详情

帖子详情页面应该包括两个模块：帖子内容模块、帖子评论模块。帖子的内容模块显示的信息包括帖子作者的基本信息、帖子标题、帖子内容、帖子发布时间、帖子获赞数量、帖子评论数量。帖子的评论模块显示的信息包括帖子的评论列表、评论输入框、评论的点赞按钮、评论的回复按钮。

当点击帖子的点赞按钮或者评论的点赞按钮时，为帖子或评论添加一个赞，显示为已赞，此时再次点击，取消赞。当对帖子进行评论时，点击提交评论按钮，采用异步请求发送给服务器，提交数据后自动刷新页面。

### 消息模块

消息模块的主要功能包括朋友私信、系统通知、发送私信。消息模块除了基本的内容展示外，需要提示用户当前未读消息数量，每一个消息类别上都应该显示当前消息类别有多少条未读消息，用户每查看一条消息，未读消息数减一。下面对消息模块的各个功能进行详细描述：

1. 朋友私信

朋友私信模块主页面显示所有已存在会话列表，列表的每一项显示会话目标的头像、用户名、当前会话的最新一条消息、最新一条消息的时间。点击会话项，进入会话详细信息页面，该页面显示会话的历史消息列表，列表的每一项显示消息的内容，消息时间。

1. 发送私信

在朋友私信页面应设置发送私信功能入口，用户点击发送私信按钮，弹出编辑私信窗口，点击发送私信，采用异步方式提交私信，自动刷新页面。

1. 系统通知

系统通知功能主要维护三个会话，包括被点赞通知、被评论通知、被关注通知。每个通知会话前显示当前会话的未读通知数量。

### 系统设置模块

系统设置模块主要功能包括：个人主页、账号设置、退出登录。系统设置模块默认是隐藏的，当点击折叠菜单时才显示。系统设置模块的主要功能描述如下：

1. 个人主页

个人主页中的内容与上面提到的作者主页内容基本一致，依然是用户的基本信息和用户的关注列表、粉丝列表、获赞数量等内容。唯一不同的是，由于自己不能关注自己，所以当查看自己的个人主页时，要隐藏关注按钮。

1. 账号设置

账号设置功能又分为两个部分：更换头像、修改密码。其中当用户点击更改头像按钮时，弹出资源管理器窗口，要求用户上传头像，将用户上传的头像存入服务器磁盘中，更新数据库中的用户头像信息，返回首页，自动刷新页面。需要注意的是，为了避免多个用户上传的头像有相同的文件名导致上传失败，每次上传时，由系统随机生成文件名。

1. 退出登录

用户点击退出登录时，退出当前账号，返回首页。

### 搜索模块

搜索模块是基于ElasticSearch服务器的，因为ElasticSearch服务器独特的信息存储方式可以加快信息检索速度。当用户在搜索输入框中输入关键词，点击搜索按钮，调用搜索相关业务方法。返回结果列表，结果列表的每一项显示与当前关键词相匹配的帖子标题和内容、帖子作者的基本信息、帖子发布时间等信息。

### 后台管理模块

后台管理模块主要包括两个部分：独立访客数据统计、日活跃用户数据统计，两种数据都支持选定时间范围。其中独立访客数据每个用户id只计数一次，日活跃用户只统计访问次数，不区分是否是同一个用户的多次访问。后台管理模块的功能模块图如下：

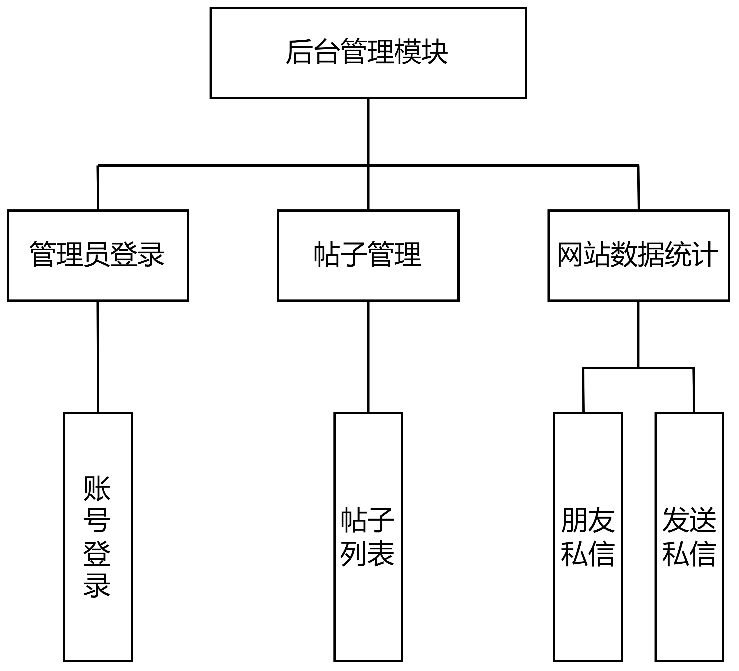


图 3.2 后台管理功能模块图

## 数据库设计

Eshare经验分享平台的核心数据是存放到关系型数据库中的，因此数据库表的设计至关重要，直接决定了后续编码阶段系统的各个功能否顺利实现。要保证数据库表设计的合理性，必须按照数据库表的设计规范。数据库表的设计流程一般是通过需求分析中的功能确定实体，找到实体之间的关系，画出ER图，根据ER图设计数据库表。

### 概念模型设计

结合需求分析和系统的功能模块分析，确定Eshare经验分享平台的实体类型包括：用户、消息、登录凭证、帖子、评论。

其中登录凭证实体需要用userid属性标识是哪个用户的登录凭证，因此需要引用用户实体的id属性；消息实体中需要fromid和toid属性标记发消息者和接收消息者，因此发消息者的fromid属性和接受者的toid属性需要引用user实体的id属性，帖子实体需要标记发帖人的属性userid需要引用用户实体的id属性，评论实体需要标记评论作者的userid属性需要引用用户实体的id属性，同时，评论依附于帖子而存在，需要标记所依附帖子的entityid属性，需要引用帖子实体的id属性。综上，确定实体之间关系的ER图如下所示：

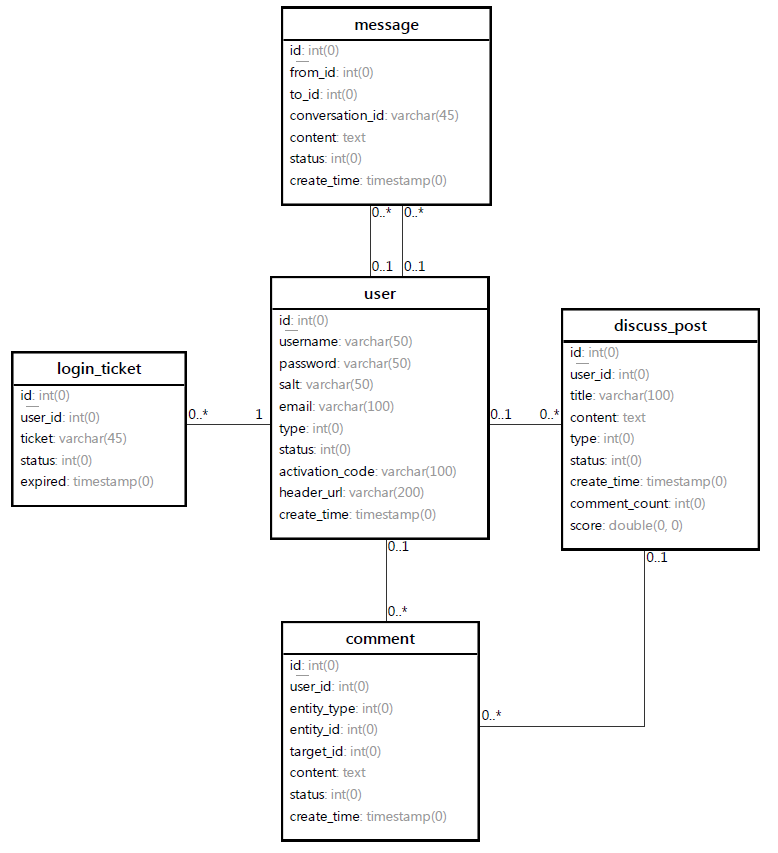


图 3.3 系统ER图

### 逻辑结构设计

本节对各个数据库表进行详细设计，根据上一小节确定的实体类型，以及每种实体相关联的属性信息，为每种实体类型建立对应的数据库表。现确定实体与数据库表的对应关系如下：

**表 3.1 逻辑结构设计表**

|  |  |
| --- | --- |
| 实体名称 | 数据库表名 |
| 用户 | user |
| 消息 | message |
| 登录凭证 | login\_ticket |
| 评论 | comment |
| 帖子 | discuss\_post |

各个数据库表的详细设计如下：

1. user表

用户表是整个系统最关键的表，几乎所有的操作都与用户表相关，用户表中需要特别说明的字段是salt，salt字段没有实际意义，主要是为了数据安全，我们存储在user表中的密码是通过md5加密算法加密后的密码，并不是明文，因此即使user表数据被泄露了，获取到数据的人依然无法直接登录用户的账号。md5算法在加密时需要两个数据，一个是用户的实际密码，另一个是在用户注册时随机生成的一串随机数字，就是salt字段。由于md5算法只能加密，没有解密算法，所以用户登录时，需要把用户输入与salt使用md5算法加密后的结果与数据库中的密码比对，相同则登录成功。

**表 3.2 数据库表user描述表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 约束 | 注释 |
| id | int | 非空、主键 | 用户id |
| username | varchar | 非空 | 用户名 |
| password | varchar | 非空 | 密码 |
| salt | varchar | 非空 | md5加密后缀 |
| email | varchar | 非空 | 邮箱 |
| type | int | 非空 | 用户类型 |
| status | int | 非空 | 账号状态 |
| activation\_code | varchar | 非空 | 激活码 |
| header\_url | varchar | 非空 | 头像地址 |
| create\_time | timestamp | 非空 | 账号创建时间 |

1. message表

与message表相关的功能有私信和系统通知，向message表存入数据的情况有两种：用户之间发送私信、用户触发系统通知。message表中需要特殊说明的字段有conversation\_id，这个字段表示一个会话的id，因为两个人之间通信，对应的会话只有一个，所以这个字段是根据消息发送者的id和接收者的id生成的，默认生成的规则是会话双方的id较小的放在前面，使用下划线连接。

**表 3.3 数据库表message描述表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 约束 | 注释 |
| id | int | 主键、非空 | 消息id |
| from\_id | int | 非空、外键 | 发送者id |
| to\_id | int | 非空、外键 | 接收者id |
| conversation\_id | varchar | 非空 | 会话id |
| content | text | 非空 | 消息内容 |
| status | int | 非空 | 消息状态 |
| create\_time | timestamp | 非空 | 创建时间 |

1. login\_ticket表

登录凭证表，这个表主要是为了实现登录页面下方的记住我选项，当用户勾选记住我选项时，服务器生成一个登录凭证存入登录凭证表，同时作为cookie发送给浏览器，存储在浏览器端，当用户下次访问服务器时，请求头中携带的cookie信息包含登录凭证，服务器从登录凭证表中查找到该凭证，则无需登录。

**表 3.4 数据库表login\_ticket描述表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 约束 | 注释 |
| id | int | 主键、非空 | 凭证id |
| user\_id | int | 非空、外键 | 用户id |
| ticket | varchar | 非空 | 凭证内容 |
| status | int | 非空 | 状态 |
| expired | timestamp | 非空 | 过期时间 |

1. comment表

评论表相关的功能主要是评论和回复，需要特殊说明的字段有entity\_type和target\_id，因为comment表中存储的不止有评论，还有对评论的回复，最主要的区别就是前端页面对评论和评论的回复加载策略不同，评论直接显示，而评论的回复需要依附于评论而存在。因此entity\_type字段用于指定评论的类型（对帖子的评论、回复），target\_id字段主要对回复有效，当记录为回复时，target\_id字段标识回复给哪条评论。

**表 3.5 数据库表comment描述表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 约束 | 注释 |
| id | int | 主键、非空 | 评论id |
| user\_id | int | 非空 | 评论作者id |
| entity\_type | int | 非空 | 1表示评论，2表示回复 |
| entity\_id | int | 非空 | 依附的帖子id |
| target\_id | int | 非空 | 被回复评论的id |
| content | text | 非空 | 评论内容 |
| status | int | 非空 | 评论状态 |
| create\_time | timestamp | 非空 | 评论创建时间 |

（5）discuss\_post表

帖子表用于存放帖子信息，是数据量最大的表。帖子表中需要特殊说明的字段是commetn\_count，该字段表示当前帖子下的评论数量，也就是说该字段的数据依赖于comment表中相关记录的条数。所以当删除comment表中的数据时，务必要更新该字段的数据。

**表 3.6 数据库表discuss\_post描述表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 约束 | 注释 |
| id | int | 主键、非空 | 帖子id |
| user\_id | int | 非空、外键 | 作者id |
| title | varchar | 非空 | 帖子标题 |
| content | text | 非空 | 内容 |
| type | int | 非空 | 帖子类型 |
| status | int | 非空 | 帖子状态 |
| create\_time | timestamp | 非空 | 帖子创建时间 |
| comment\_count | int | 非空 | 帖子评论数量 |
| score | double | 非空 | 帖子评分 |

## 本章小结

本章确定了Eshare经验分享平台的主要功能模块，并对每个模块应该包括什么功能，采用什么技术实现这些功能进行了详细的描述。同时，通过分析系统中可能用到的实体类型，对系统的数据存储策略进行了设计，把系统中可能用到的实体映射到关系型数据库的存储表，每个实体的属性映射到存储表的字段。

# 详细设计

详细设计阶段更加注重代码的实现方案，对于大型项目而言，整个系统的功能比较多，最终项目的代码量也非常庞大，直接根据概要设计进行编码，可能会出现由于类的设计不合理、调用关系太复杂导致的大量冗余代码，甚至出现错误。因此在概要设计阶段设计的各种功能，通过代码实现之前，需要先进行详细设计，这个阶段主要根据功能的执行流程，对编码实现时需要的类、每个类中应该设置的具体方法、方法之间的调用关系和方法的调用顺序进行设计。

结合我们当前项目所采用的MVC设计模式，由于Controller层主要负责调用设计好的业务层方法去处理前端请求，Controller层本身并不涉及复杂的处理逻辑，因此本章详细设计的重点应该是Service层和Dao层。

## 注册登录模块详细设计

注册登录模块有两个核心功能：注册和登录，这两个功能的本质其实就是向用户表中插入一条用户数据或者查询一条用户数据是否存在。所以程序中应该有以下几部分内容：与用户表相对应的存储用户数据的实体类（User）；Service层的登录和注册方法，同时根据概要设计中的需要激活账户的需求，业务层应该提供用于激活账号的激活方法； Dao层对用户表执行插入和查询操作的方法，修改激活状态的方法。本小节根据以上分析对注册登录模块进行详细设计。

### 注册登录模块程序流程

用户向服务器发送访问请求时，服务器会首先检查用户的请求头中是否携带登录凭证信息，如果有并且有效，则直接登录成功，返回登录成功后的首页。如果没有，则要求用户输入登录信息，服务器验证登录信息合法性，登录失败时要返回给前端错误信息。对于注册，用户输入注册信息，服务器验证合法性，不合法则返回错误原因，注册成功后向用户的注册邮箱中发送激活邮件。

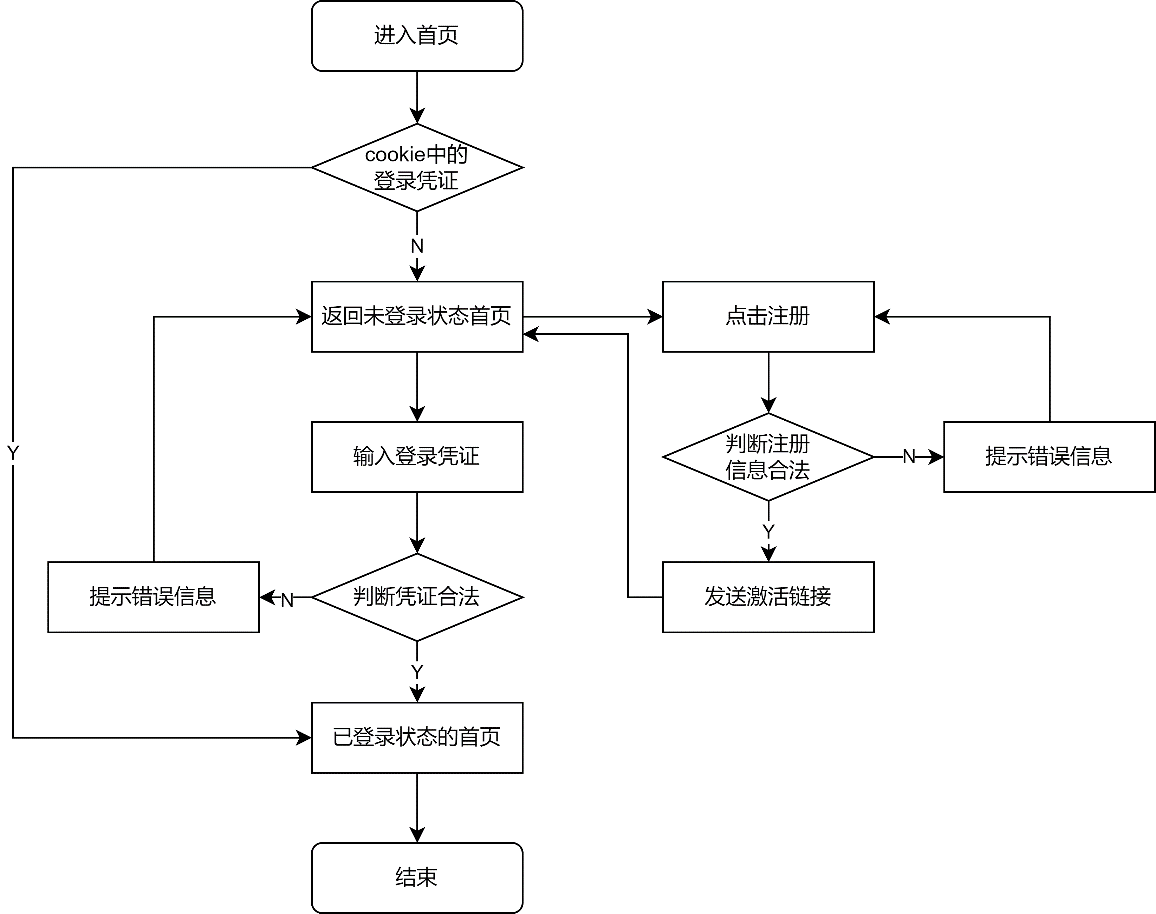


图 4.1 注册登录模块流程图

### 注册登录模块类设计

1. 实体类

User类：与数据库表user表相关联的实体类

LoginTicket类：与数据库表login\_ticket相关联的实体类

1. Service层、Dao层

UserService类：该类中应该封装用于注册（register）、登录（login）、查找用户（findUserById、findUserByName、findUserByEmail）、激活（activation）、查找登录凭证（findLoginTicket）等方法。

UserMapper类：该类中封装用于直接对user表进行操作的查找用户（selectUserById、selectUserByName、selectUserByEmail）、新增用户（insertUser）、修改用户状态（updateStatus）等方法。需要说明的是，查询用户的方法之所以设置三个，是需求中要求用户名和邮箱不能重复，注册时需要验证邮箱和用户名是否已经存在。

1. 其他类

LoginInterceptor类、Hostholder类：验证cookie信息使用拦截器LoginInterceptor完成，拦截器在controller层方法执行之前执行，尝试从请求中获取登录凭证信息，如果获取并验证成功，则将凭证相关联的用户对象存入请求域对象Hostholder中供后续处理使用（Hostholder使用ThreadLocal实现数据与线程之间的关联存储，所以服务器会为每个请求维护一个Hostholder对象）。

CommunityUtil类：这个类定义了整个项目中常用的通用方法，例如登录模块中要使用的生成随机字符串（getUUID）方法、用于加密用户密码的方法（md5）。需要说明的是，md5算法是公开的，所以直接对用户的密码执行md5加密是没有意义的，所以我们计划为每个用户生成一个随机字符串，与用户的明文密码拼接后在使用md5算法加密。

## 帖子模块详细设计

帖子模块的功能很多，但总体分为两大类：直接与帖子相关的操作、帖子的附加功能相关操作（例如点赞和评论）。与帖子直接相关的操作实现比较加单，主要是对数据表的添加操作，需要特别说明的是点赞和评论操作。虽然前端的点赞和评论操作很简单，但是点赞和评论都是独立的处理方法，且点赞和评论的结果需要发送系统通知，放在同一个类中太过庞大，因此为每个功能设置独立的业务模块。

### 帖子模块类设计

1. 实体类

Discusspost类：数据库表discuss\_post对应实体类，存放帖子相关信息。

Comment类：数据库表comment对应实体类，存放帖子评论相关信息。

Page类：用于支持分页功能，系统正式上线之后，数据量很大，不可能一次把数据全部加载到一个页面，因此分页是许多内容网站经常采用的策略。既然分页的最终目的是在前端实现分页的显示效果，我们就应该从前端的角度去设计Page类。通过观察常用网站的分页设计策略，我们了解到，分页主要涉及的信息有总页数、当前页码、跳转到首尾页，所以Page类中应该设置获取这些值的方法。除此之外，分页不只有静态的显示，当我们点击分页中的某一页时，应该获取并显示这一页对应的信息，此时前端需要知道要向服务器发送的请求路径，因此分页类中应该提供当前分页的访问地址信息，方便前端分页模块直接获取。

1. service层

UserService接口：这个接口是许多功能共用的，由于帖子列表的每一项信息中包括用户头像，用户名等信息，所以需要调用UserService接口的findUserById方法。

DiscussPostService接口：这个接口中定义帖子模块核心功能相关的方法，包括添加帖子（addDiscussPost）、查询帖子（findDiscussPosts、findDiscussPostById）、查询帖子数量（findDiscussPostRows）、更新帖子评论数量（updateCommentCount）。需要说明的是更新帖子评论数量方法，因为我们在设计数据库表时，为了方便加载帖子数据时显示帖子的评论数量，所以在帖子表discuss\_post中添加了一个记录评论数量的字段，因此当帖子数量变化时，要更新这个字段，保持数据一致性。这样做虽然违背了数据库设计第三范式，但是用一定的存储空间换取了系统效率的提升。

CommentService接口：这个接口中定义了评论功能相关的方法，包括查询评论（findCommentsById、findCommentById）、添加评论（addComment）、查询评论数量（findCommentCount）。需要说明的是查询评论方法，查询评论有两种类型，一种是对帖子的评论，一种是回复，我们在设计数据库表的时候为了区分这两种类型的评论，在评论表中设计了一个type字段区分评论类型，所以设计方法时也要设计一个用于区分评论类型的参数。另一个查询方法findCommentById是专门用于查询评论的回复，id是被回复评论作者的userid，提供这个方法主要是为了在前端显示时实现“A回复了B：”的显示效果。

LikeService接口：这个接口中定义了点赞相关的方法，包括点赞动作（like），查询帖子或评论的获赞数量（findEntityLikeCount）、查询获赞状态（findEntityLikeStatus）。需要说明的是查询获赞状态，因为需求中提到已经点赞时，在执行点赞操作就取消赞。所以每次执行点赞操作前要判断点赞状态，已点赞就把点赞数量减一，已点赞就把点赞数量加一。

1. dao层

UserMapper接口：该接口中与帖子模块相关的方法包括selectUserById，主要用于显示帖子作者的用户基本信息。

DiscussPostMapper类：该接口中定义了直接对帖子表discuss\_post相关的操作，包括查询用户帖子（selectDiscussPosts）、查询用户帖子数量（selectDiscussPostRows）、新增帖子（insertDiscusspost）、更新帖子评论数量（updateDiscussPost）。

CommentMapper类：该类中定义了对评论表comment的操作，包括查询评论（selectCommentsById、selectCommentById）、查询评论数量（selectCommentCount）、添加评论（insertComment）。

### 帖子模块时序设计

上一小节中提到的帖子模块相关类和接口有很多，对于实现功能时类之间的调用关系、执行顺序比较复杂的功能，本小节将采用时序图的方式，展示系统的各层如何协调配合实现这些功能。

1. 添加帖子功能

前端通过异步请求的方式发送添加帖子请求，controller接收到请求后，调用DiscussPostService的addDiscussPost方法，在addDiscussPost方法中调用过滤敏感词方法，将过滤后的内容调用DiscussPostMapper接口的insertDiscussPost方法，向discuss\_post表中插入一条数据，同时为了支持后面的帖子搜索功能，调用ElasticSearch官方提供的添加方法，将帖子数据同步添加到ElasticSearch服务器。帖子添加功能的时序图如下所示：

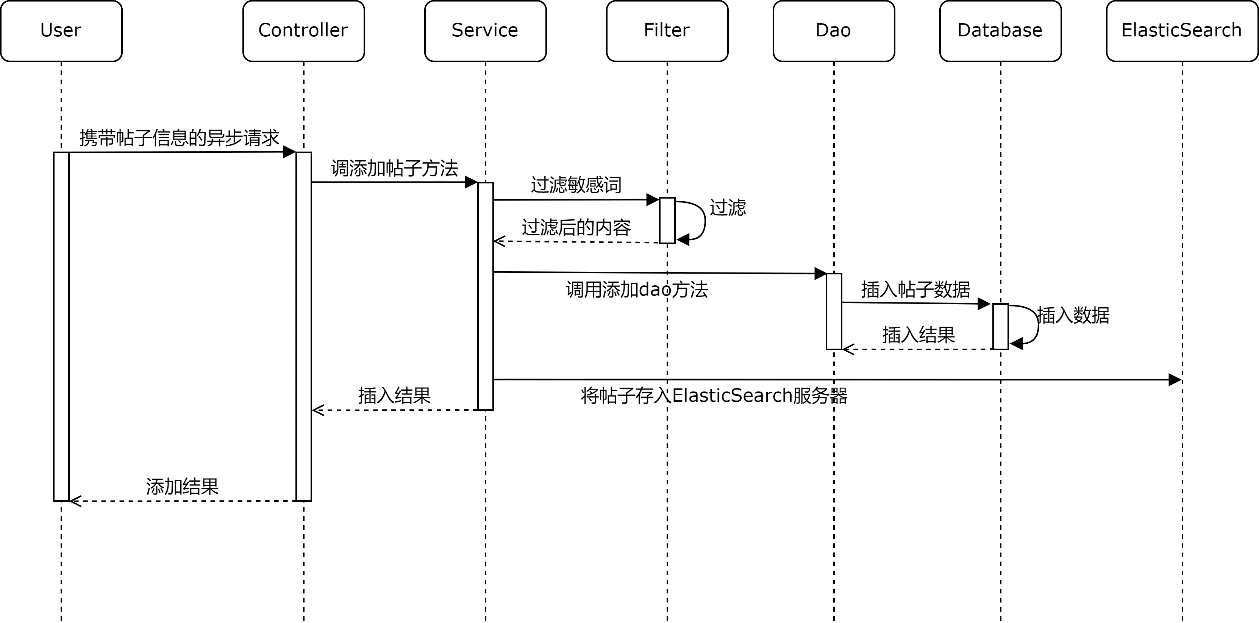


图 4.2 添加帖子功能时序图

1. 添加评论功能

用户编辑评论后点击发表，前端发送添加评论请求，Controller层收到请求调用CommentService的addComment方法，addComment方法再调用CommentMapper接口的insertComment方法，向comment表中添加数据。由于需求分析中要求被评论时需要有系统通知，所以此时还需要调用UserService的findUserById方法查询被评论用户的用户信息，调用Kafka的生产者类中发布消息的方法，消费者会接收消息，并将被评论事件存入通知表。

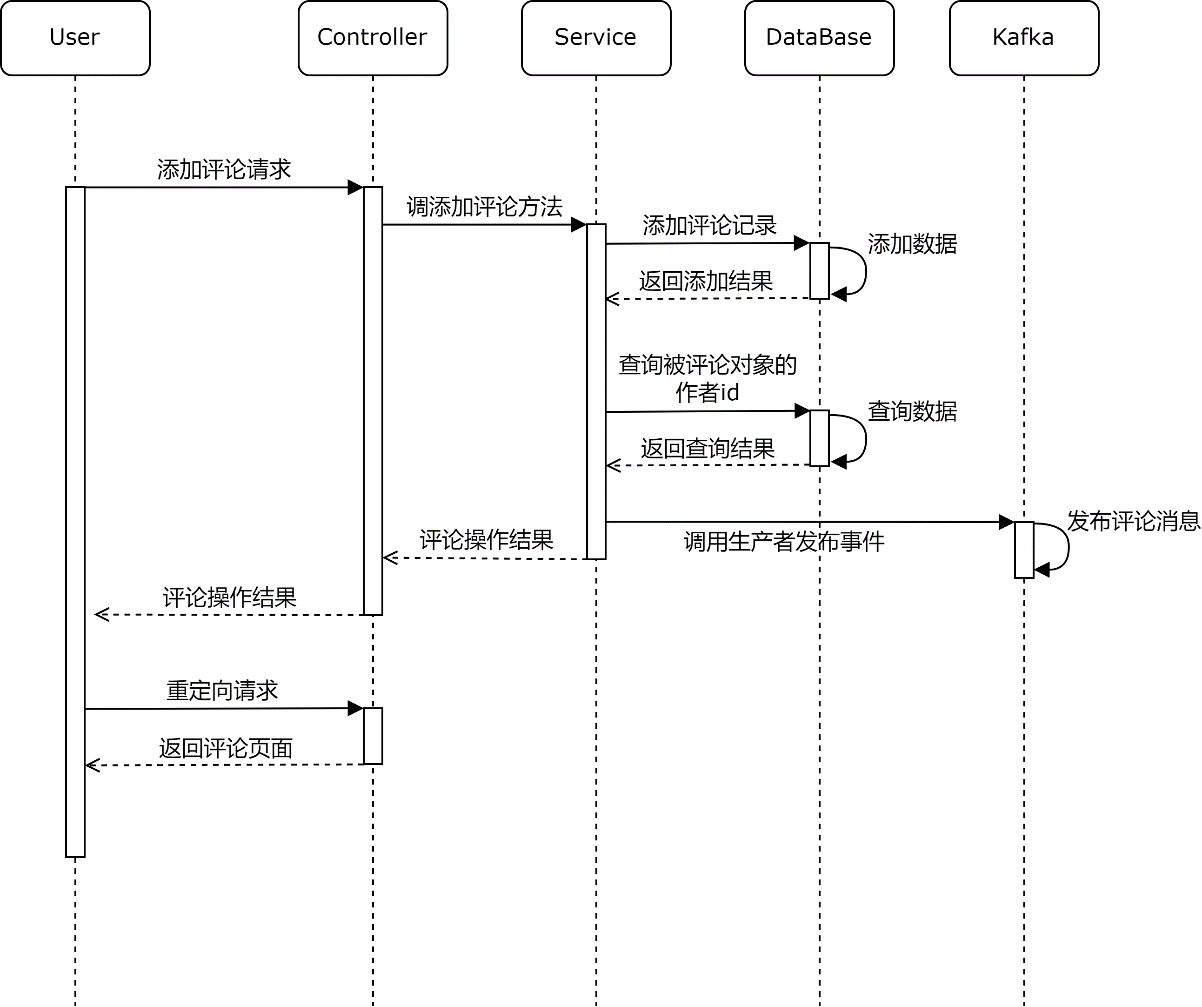


图 4.3 添加评论功能时序图

1. 点赞功能

用户点击点赞按钮，浏览器发送异步请求，Controller层接收到点赞请求后，调用LikeService接口的like方法，like方法中先调用findEntityLikeStatus查看当前点赞状态，在根据当前点赞状态，调用Spring官方提供的Redis相关的API方法，修改点赞数量。同样根据需求分析，需要调用Kafka相关方法将事件发布到点赞主题。

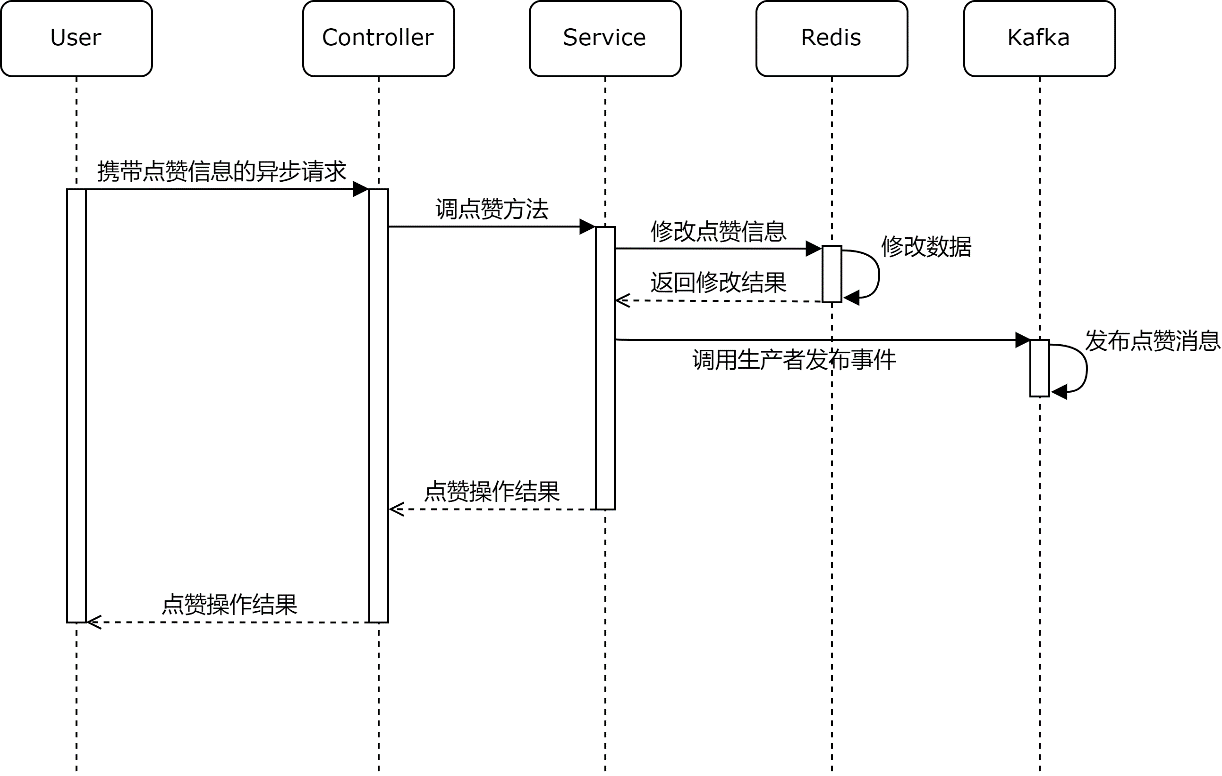


图 4.4 点赞功能时序图

## 消息模块详细设计

消息模块有两个主要功能：私信功能和系统通知功能。两者之间的区别就是私信功能是用户与用户之间的消息往来，系统通知是系统向用户发送的单向消息，这两个子功能共用一个数据库表message。发送私信、接收私信与发送系统通知、接收系统通知的过程本质都是向数据库表message中插入数据和读取数据。本节将详细介绍两种功能相关的类和类之间的调用关系应该如何设计。

### 消息模块类设计

根据消息模块需要实现的具体功能：发送私信、显示私信、接收系统通知，分析每个功能的实现需要调用的方法。

其中，发送私信的过程涉及以下方法：查询目标用户用户名是否存在的UserService相关方法、MessageService类中应该新增的向message表中添加消息的相关方法、CommunityUtil类中应该新增的响应异步请求时所需要的JSON字符串生成方法。

显示私信包括显示私信列表和显示私信详情：显示私信列表的过程涉及UserService接口中与查询用户信息相关的方法、MessageService中应该新增的查询某个用户会话中最新消息的方法、查询某个会话的未读消息数量的方法，显示某个私信会话详情的过程涉及UserService中查询用户信息的方法（用于在详情页面显示当前会话是与哪个用户的会话）、MessageService中应该新增的根据用户Id查询私信列表的方法。

接收系统通知功能涉及MessageService中应该添加的根据当前用户id查询相关的通知列表的方法、查询未读通知数量的方法。需要说明的是，我们这里设计的接受通知相关方法其实只是显示系统通知相关方法，真正从kafka中接受系统通知消息并存入数据库的是kafka的消费者方法。

将以上分析过程中涉及到的类以及需要用到的其他类的详细实现总结如下：

1. 实体类

Message类：与数据库表message相关联的实体类

Page类：用于支持消息列表和会话中私信列表的分页，原理与上面提到的帖子列表的分页原理相似，这里不再赘述。

1. service层

MessageService接口：查询最新消息列表（findLatestMessages）、查询当前用户会话数量（findConversationCount）、查询某个会话的消息列表（findMessages）、查询某个会话的消息数量（findMessagesCount）、查询未读消息数量（findUnreadMessages-Count）、新增消息（addMessage）、读取消息（readMessage）。

UserService接口：根据用户名查询用户的方法（findUserByName）、查询用户方法（findUserById）。

1. dao层

MessageMapper接口：查询当前用户的会话列表（selectLatestMessages）、查询当前会话数量（selectConversationCount）、查询某个会话的所有消息（selectAllMessages-ById）、查询某个会话的所有消息数量（selectMessagesCount）、查询未读消息数量（selectUnreadMessageCount）、修改消息状态（updateStatusOfMessage）、新增消息（insertMessage）、查询通知列表（selectNotices）、查询未读通知数量（selectUnReadNoticeCount）、查询最新通知内容（selectLatestNotice）。

1. 其他类

CommunityUtil类：json字符串生成方法（genJson），这个方法使用阿里巴巴提供的fastjson工具，把ajax请求中用于返回给前端的数据(ajax请求状态码，提示信息，数据）封装成json，再转化成字符串。

### 消息模块时序设计

在类设计的基础上，对于比较复杂的功能通过时序图的方式进一步分析类之间的调用关系和执行顺序。消息模块中需要特别说明的是发送私信功能，发送私信时，前端通过异步请求的方式将要发送的私信数据发送给Controller处理，Controller层调用MessageService中添加私信的方法addMessage，addMessage方法调用MessageMapper中的方法insertMessage，向message表中插入数据。插入成功后，MessageService调用CommunityUtil的genJson方法生成发送成功的json数据，返回给前端提示用户发送成功。

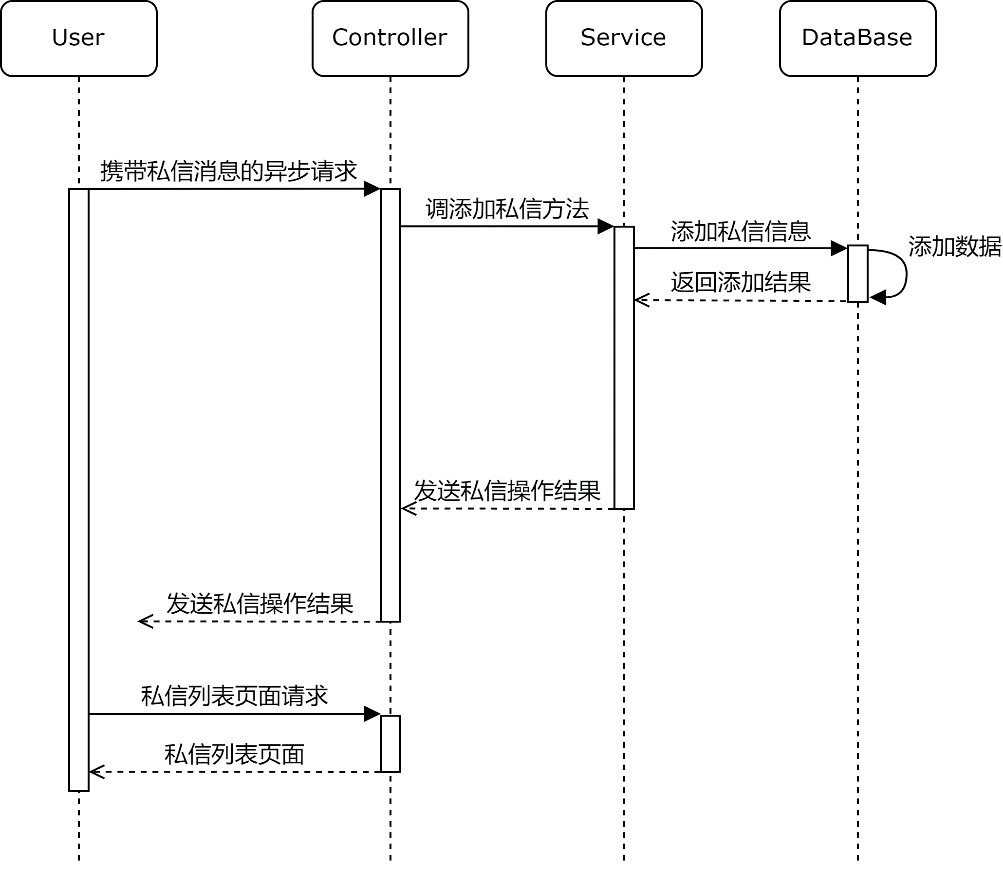


图 4.5 消息模块时序图

## 系统设置模块详细设计

系统设置模块主要有三个功能：个人主页、账号设置、退出登录。这个模块的设计重点和难点是个人主页中的关注功能和账号设置中对于上传头像功能的实现。大致的思路是，关注功能采用Redis实现，上传头像后将头像文件存到服务器磁盘，生成磁盘文件的访问地址，更新用户表中的用户头像地址信息。

### 系统设置模块类设计

整个系统设置模块中除了关注功能和上传头像功能之外，都是基本的数据增删改查操作，没有复杂的逻辑，与前面的模块设计中的只涉及增删改查操作的功能实现类似。需要特别说明的关注功能和头像上传功能相关的类设计如下：

1. 实体类

User类：整个系统设置模块主要就是对user表数据的相关操作。

1. Controller层

UserController类：用户上传用户头像的uploadHeader方法，用户上传头像请求由UserController直接通过Java的文件IO操作存入服务器磁盘，需要特别说明的是UserController接收图片文件的方式，这里我们使用Spring提供的MultiPartFile类作为参数接收前端请求中的图片文件。

1. service层

UserService接口：更新用户头像地址方法（updateHeaderUrl）。

### 系统设计模块时序设计

虽然整个系统设置模块的功能实现并不复杂，但为了方便后续开发时参考，还是给出各个功能的时序图如下：

1. 查看个人信息功能时序图

个人信息的加载过程中需要注意头像的加载是独立的请求，浏览器第一次向服务器发起个人信息查询请求时，查询到了头像地址，返回给浏览器，浏览器会再次请求这个地址获取用户的头像文件。

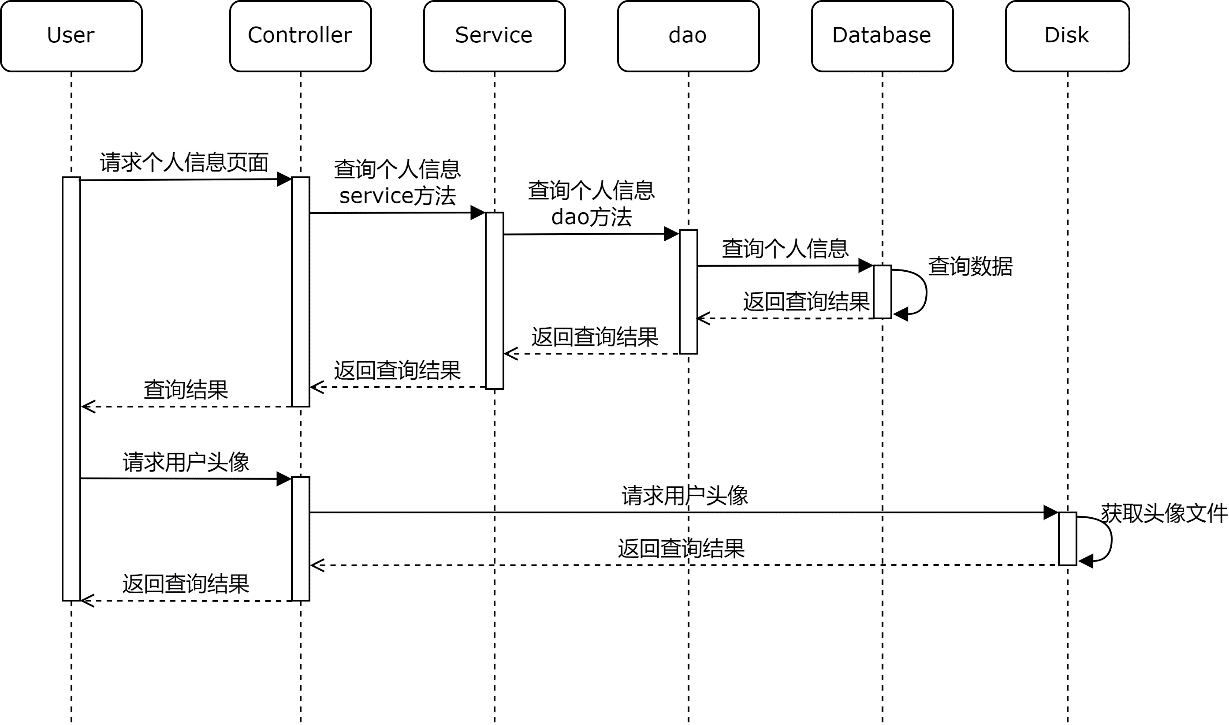


图 4.6 查看个人信息功能时序图

1. 关注功能时序图

关注功能和前面提到的点赞功能十分相似，关注功能中也要求用户点击一次则关注该用户，再次点击则取消关注，因此每次执行关注操作之前需要先判断当前的关注状态。执行关注操作后按照需求，需要发送系统通知提醒被关注用户。

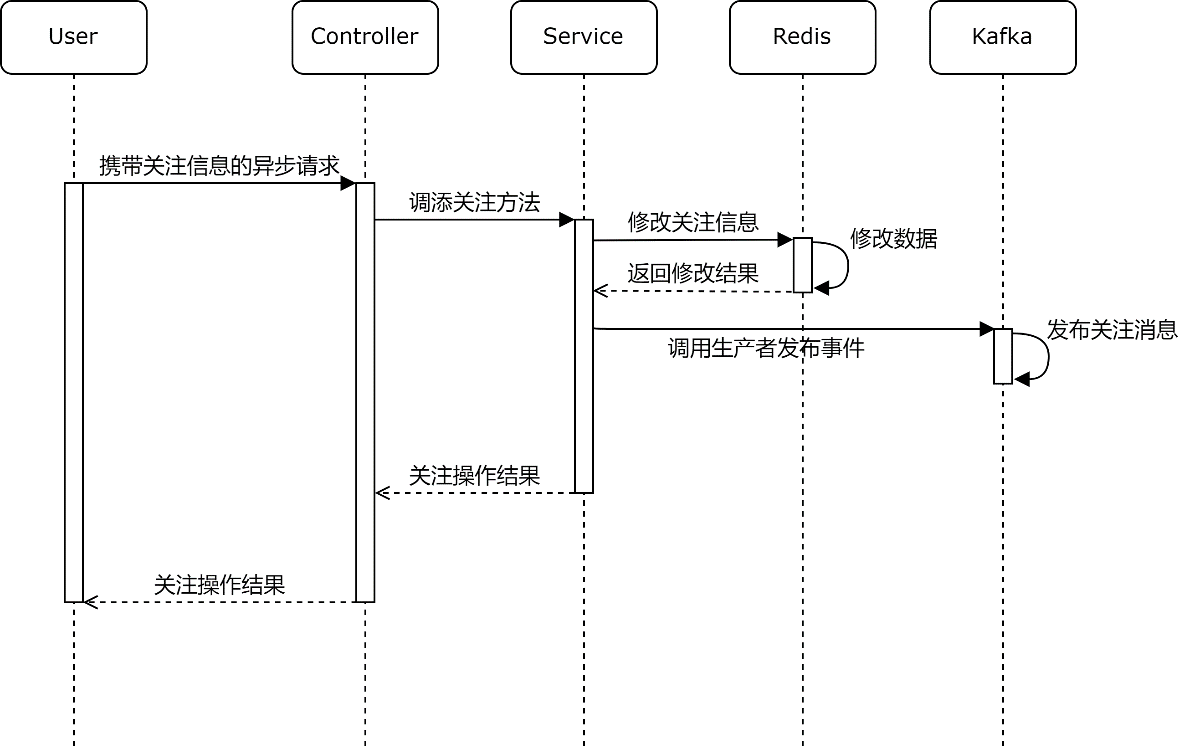


图 4.7 关注功能时序图

1. 上传头像功能时序图

上传头像时，前端发送上传文件请求，Controller层接收到图片文件，调用Java的文件IO方法存入服务器的磁盘，并生成文件的URL地址，调用UserService的updateHeaderUrl方法修改用户头像地址，前端再次发起请求刷新页面，显示上传的头像。

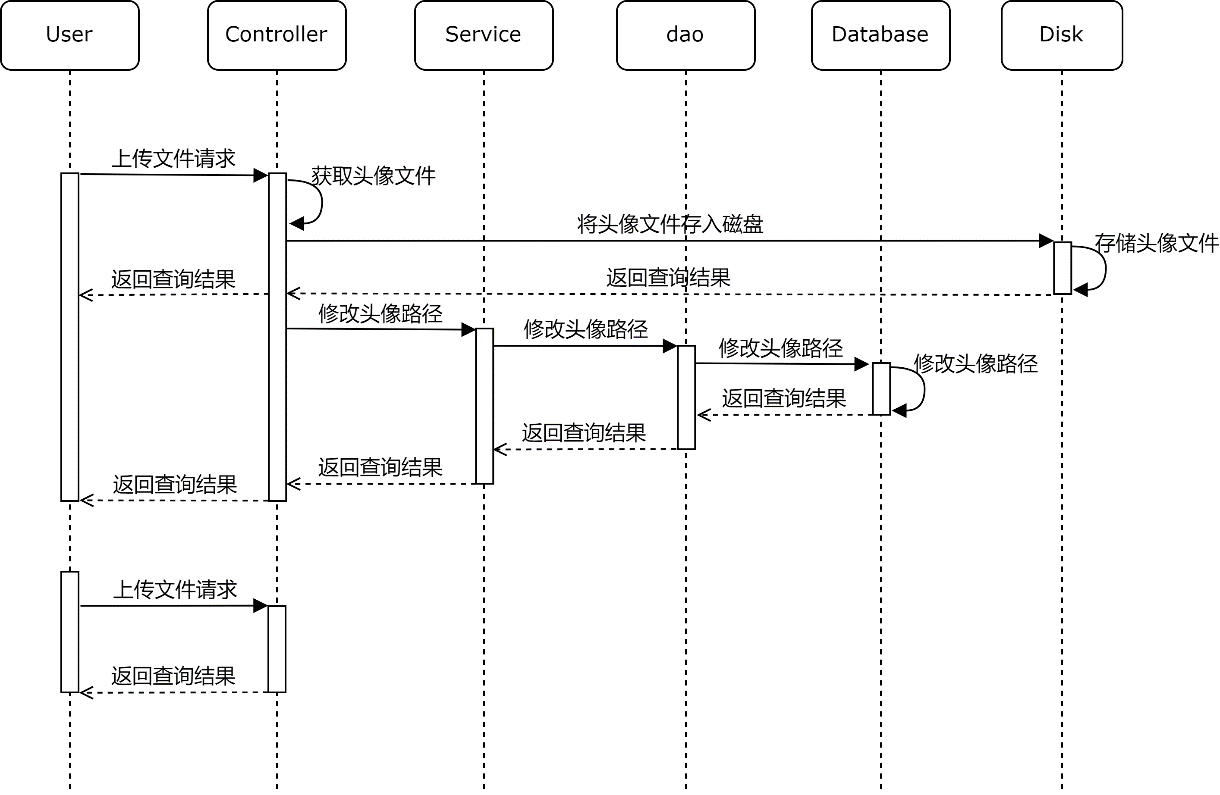


图 4.8 上传头像功能时序图

## 搜索模块和后台管理模块详细设计

搜索模块和后台管理模块的内容比较少，其中需要特别说明的是搜索功能的实现，搜索功能采用ElasticSearch作为搜索服务器，经过多个版本的迭代，现在对ElasticSearch的操作基本与对数据库的操作相同，都可以直接调用API方法实现数据的添加和查询。

### 搜索模块与后台管理模块类设计

搜索业务的的核心实现在ElasticSearchService的search方法中，search方法实现通过关键字搜索的原理是先通过NativeSearchQuery类的对象指定查询条件，再通过调用ElasticSearchRestTemplate类的search方法执行查询，最新的ElasticSearchRestTemplate类的search方法返回值是SearchHits<T>类型，SearchHit可以看做是包含了查询相关信息的实体类集合，可以通过getContent方法获取其中的实体类对象。

1. 实体类

DiscussPost类：查询结果集合的元素类型是DiscussPost类型。

1. service层

ElasticSearchService接口：向ElasticSearch中添加帖子的方法（saveDiscussPost）、通过id删除ElasticSearch中的帖子的方法（deleteDiscussPost）、通过关键字查询帖子的方法（search）。

UserService接口：显示查询结果时，需要展示每一个帖子的作者信息。

LikeService接口：查询结果的帖子中需要展示每一个帖子的获赞数量

### 搜索模块与后台管理模块时序设计

可以理解为搜索功能使用了ElasticSearch作为“数据库”，每次搜索并不是从数据库中搜索目标数据，而是从ElasticSearch服务器中查询关键词。下面展示搜索功能相关类的时序图：

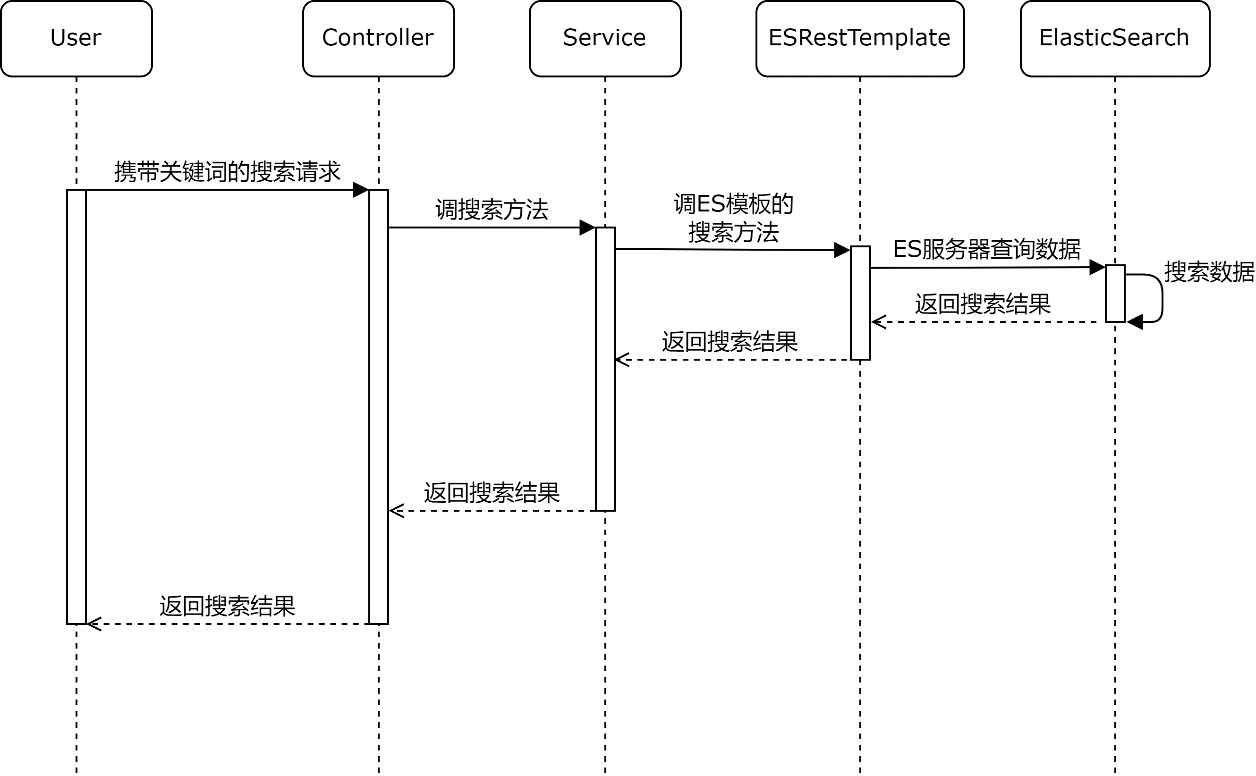


图 4.9 后台管理模块时序图

## 本章小结

本章通过分析每个功能模块的具体功能的执行流程和实现方案，给出了每个功能模块所需要类的详细设计，对于调用关系比较复杂的类，给出了该功能对应的时序图。通过本章的设计，Eshare经验分享平台代码实现的基本框架和部分重点技术细节编码实现方案已经初步确定，接下来就是通过具体的代码，实现系统的各个功能。

# 编码实现

编码实现部分是项目真正的开发阶段，本章主要根据上一章详细设计的内容，对Eshare经验分享平台的各个功能模块进行编码实现，整个项目后端将采用SSM框架，前端采用Bootstrap框架，前后端数据交互采用Thymeleaf模板引擎。下面将详细介绍项目开发环境、项目各模块功能的实现效果等内容。

## 开发环境

在项目开发过程中，使用的各种工具之间并非相互独立，彼此之间都有紧密的依赖关系，因此项目的开发环境，尤其是各软件和第三方库的版本，在个人开发过程中，对整个项目开发过程影响非常大。下面列出Eshare经验分享平台的开发环境信息：

**表 5.1 开发环境表**

|  |  |
| --- | --- |
| 名称 | 版本信息 |
| JDK | 1.8 |
| SpringBoot | 2.6.4 |
| MySQL | 8.0.21 |
| Redis | 3.0.504 |
| ElasticSearch | 7.15.2 |
| Kafka | 2.13-3.10 |
| Tomcat | 9.0.58 |
| Maven | 3.8.4 |
| Jquery | 3.1.0 |
| Bootstrap | 4.3.1 |

## 各功能模块实现

现在的的项目实际开发过程中通常都使用模板引擎，项目前后端的代码基本没有关联，只有数据的交互，编码阶段大体遵循从前端到后端的实现顺序，根据前端页面中的功能，逐个编写每个功能对应的后端代码。这样边写边查看实现效果，有助于理清开发思路，及时调整各个功能的实现方案，提升开发效率。

### 注册登录模块实现

1. 注册功能

用户在未登录状态点击导航栏的注册，跳转到注册页面，提示用户输入注册信息，点击立即注册时，服务器主要会验证账号名是否重复，邮箱是否已经被注册。如果验证通过，向用户发送激活邮件，账号激活后才可登录。



图 5.1 注册功能实现效果图

1. 登录功能

对于第一次登录的用户，输入登录信息和图片验证码中的提示信息，服务器主要验证两个内容：登录信息、验证码信息，如果验证通过，进入首页。对于已登录过并勾选了记住我选项的用户，在登陆凭证有效期内，直接进入已登录状态的首页。



图 5.2 登录功能实现效果图

### 帖子模块实现

1. 帖子列表

不管用户是否已经登录，都可以查看帖子列表，帖子列表中的每一项展示了帖子标题，帖子的发布事件，帖子的作者基本信息，帖子的评论数量，获赞数量信息。



图 5.3 帖子列表功能效果图

1. 发布帖子

在帖子列表界面，点击右上角的我要发布按钮，系统弹出编辑帖子窗口，在编辑界面可以输入帖子的标题，帖子的内容信息。编辑完成之后点击发布按钮，帖子发布成功，刷新页面即可看到新发布的帖子。

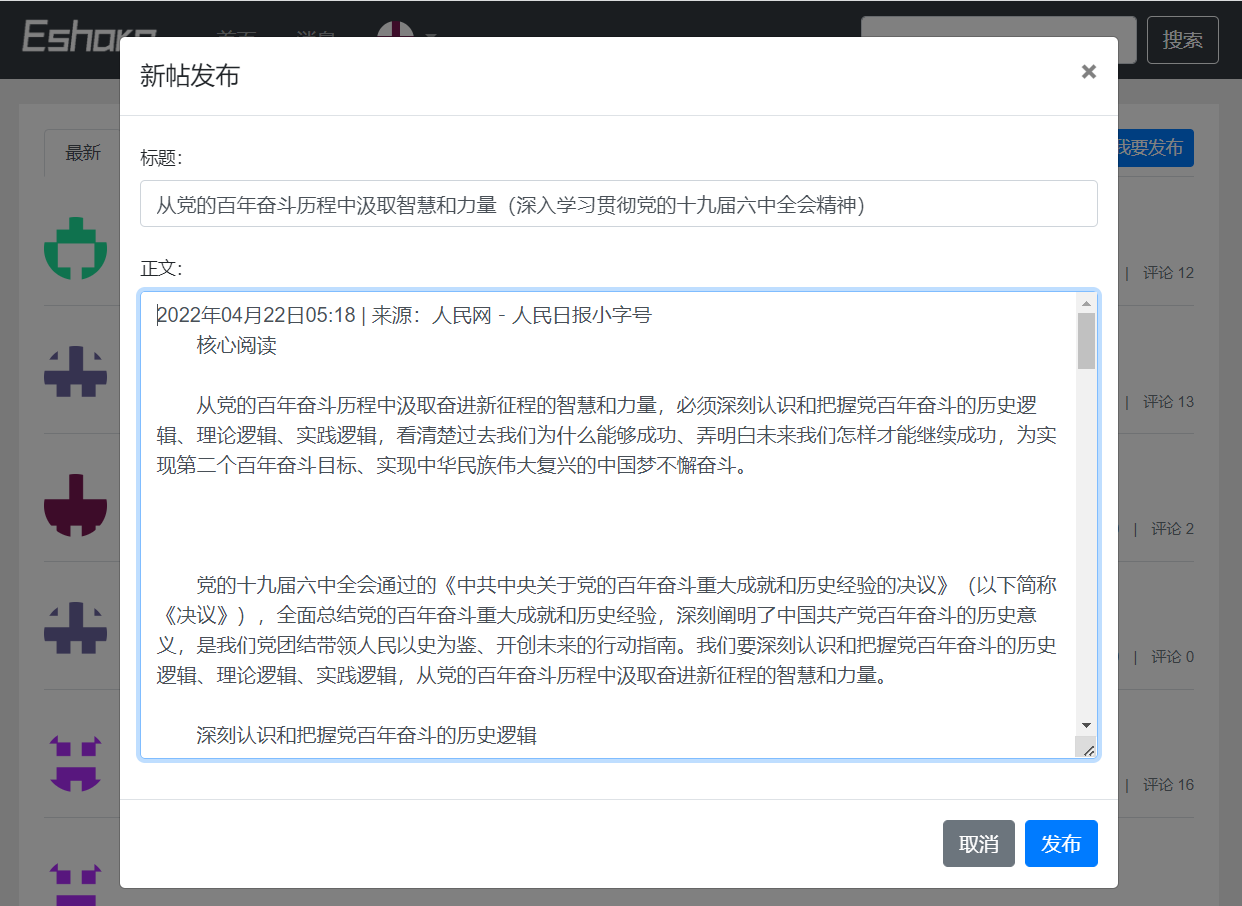


图 5.4 发布帖子功能实现效果图

1. 帖子详情

帖子详情页除了展示帖子和作者的基本信息之外，还可以对帖子进行点赞、评论操作。另外，通过点击详情页面的作者头像，可以进入到作者的个人主页，查看作者相关的更加详细的信息。

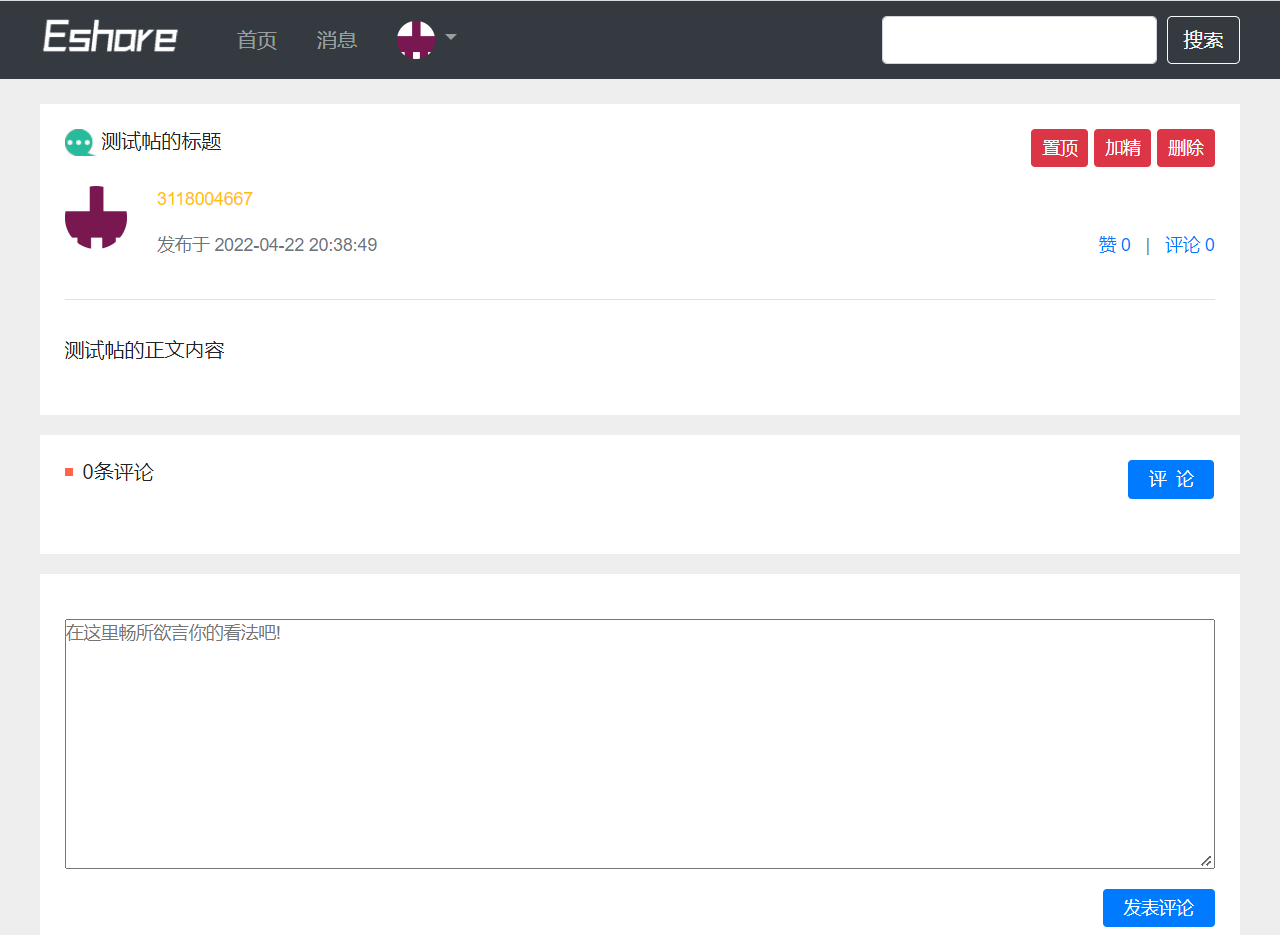


图 5.5 帖子详情实现效果图

### 消息模块实现

1. 朋友私信

点击导航栏的消息可以进入到私信列表页面，点击任何一个会话，进入到朋友私信的详情页，该页面中显示了当前会话的历史消息，可以通过点击有上角的给他私信按钮快速回复私信消息。

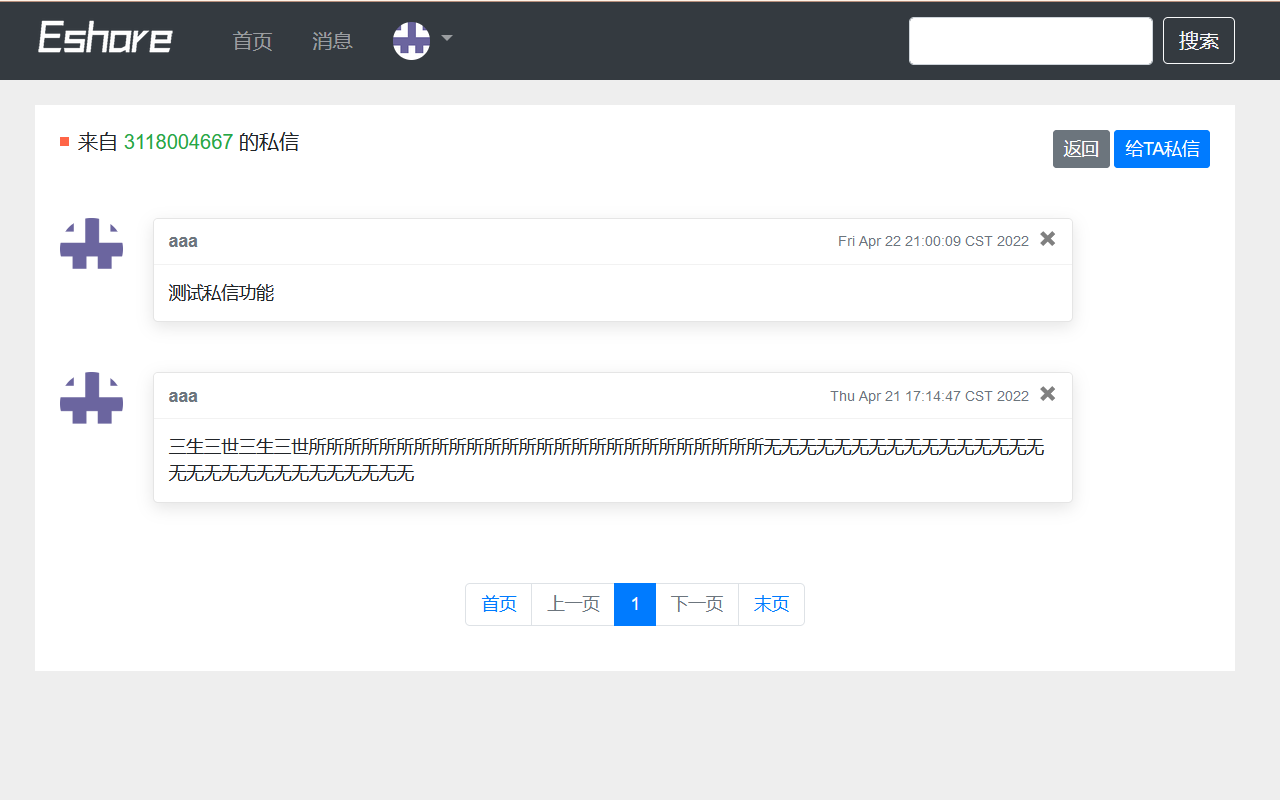


图 5.6 朋友私信实现效果图

1. 系统通知

系统通知界面目前显示的会话数量最多只有三个：被关注消息、被点赞消息、被评论消息。每个会话上通过红点显示当前会话的未读消息数量，导航栏的系统通知项显示所有未读通知数量，点击会话查看当前未读通知的详细信息。

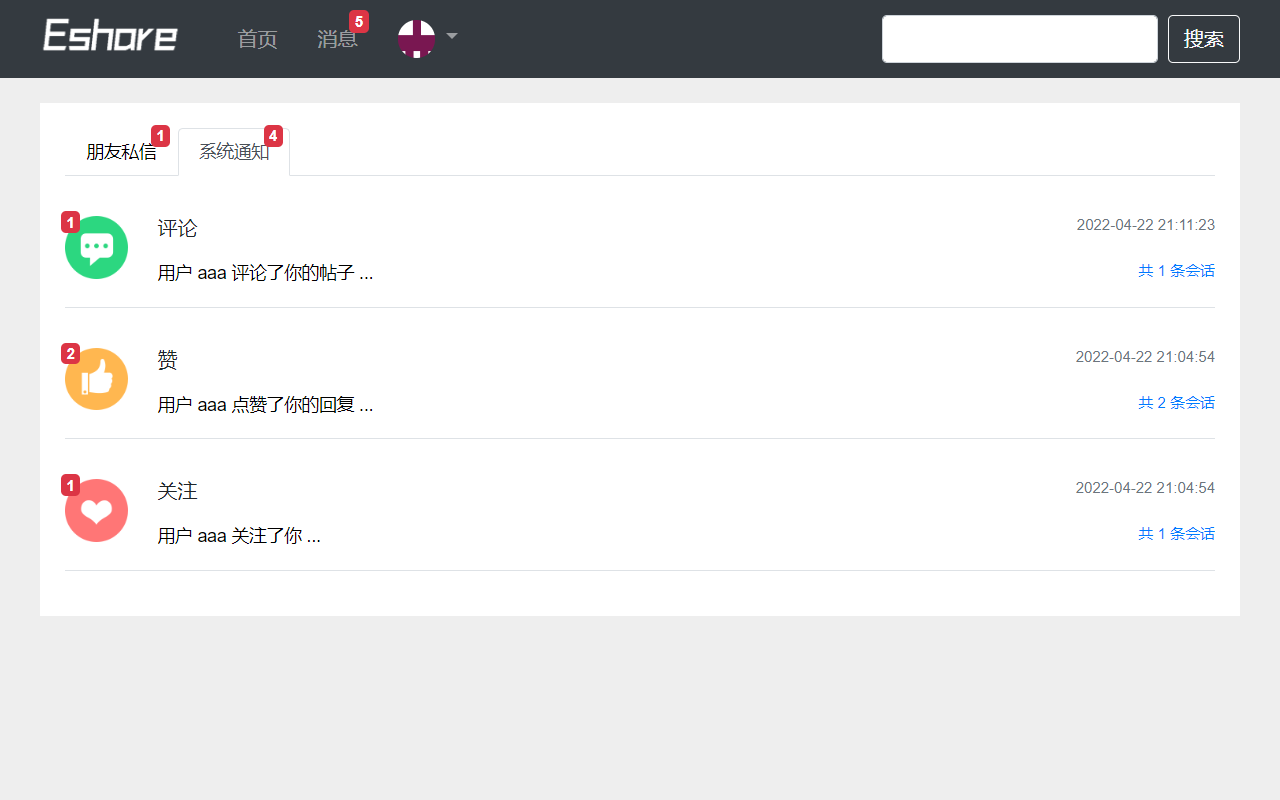


图 5.7 系统通知实现效果图

### 系统设置模块实现

* 1. 个人主页

个人主页显示了用户的详细信息，包括注册时间、关注数量、粉丝数量、获赞情况。通过点击关注数量和粉丝数量的数字，可以显示该用户的关注列表和粉丝列表。

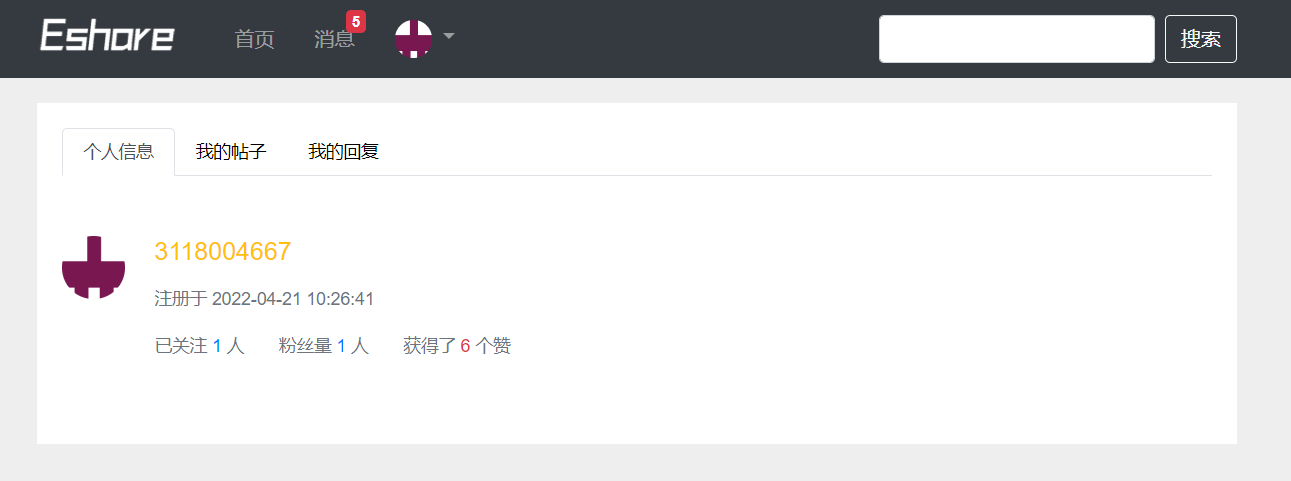


图 5.8 个人主页实现效果图

* 1. 关注功能

当用户查看其他人的主页时，会显示关注按钮，用户点击关注按钮，关注成功，当前显示已关注字样，再次点击已关注按钮则取消关注。

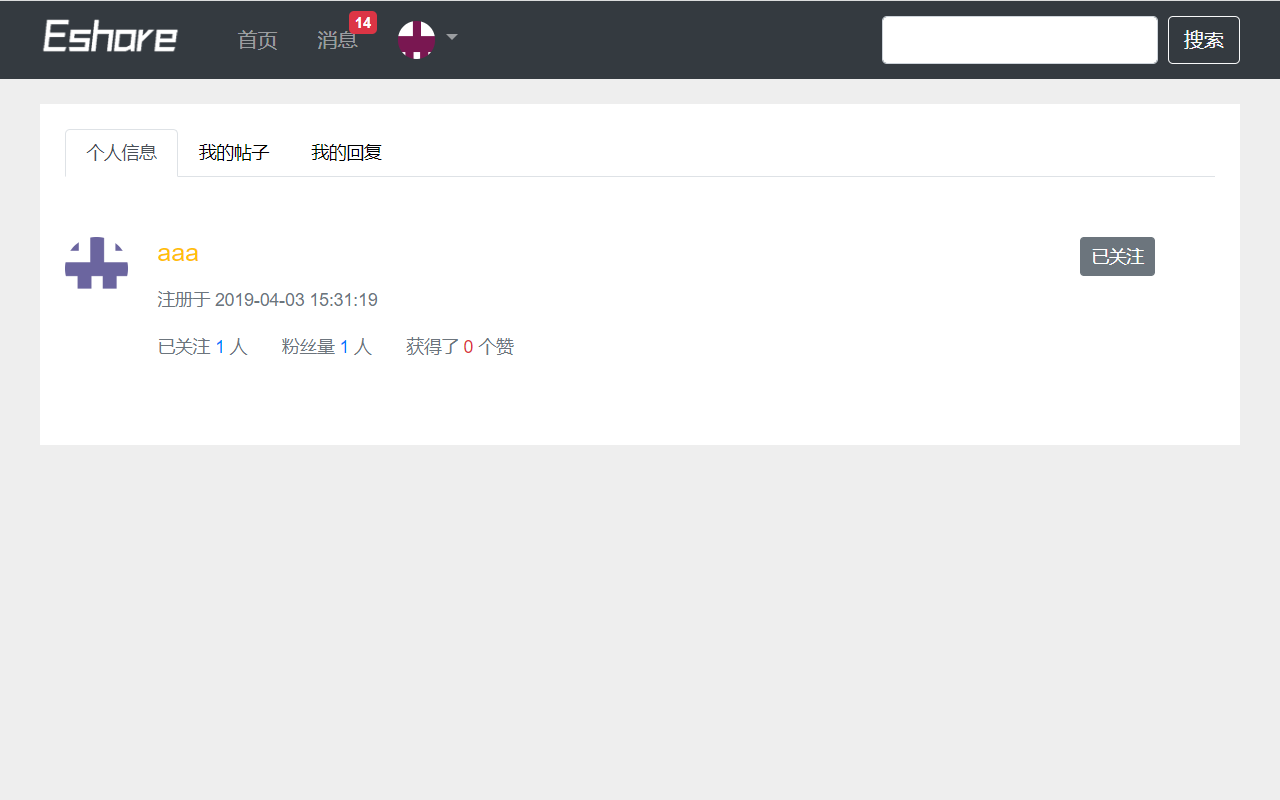


图 5.9 关注功能实现效果图

* 1. 粉丝列表

粉丝列表中显示了关注当前用户的所有用户，在该界面可以直接通过每一项右侧的关注按钮关注这些用户。另外，还可以通过切换主页面上方的标签，快捷的实现关注列表和粉丝列表之间的跳转。

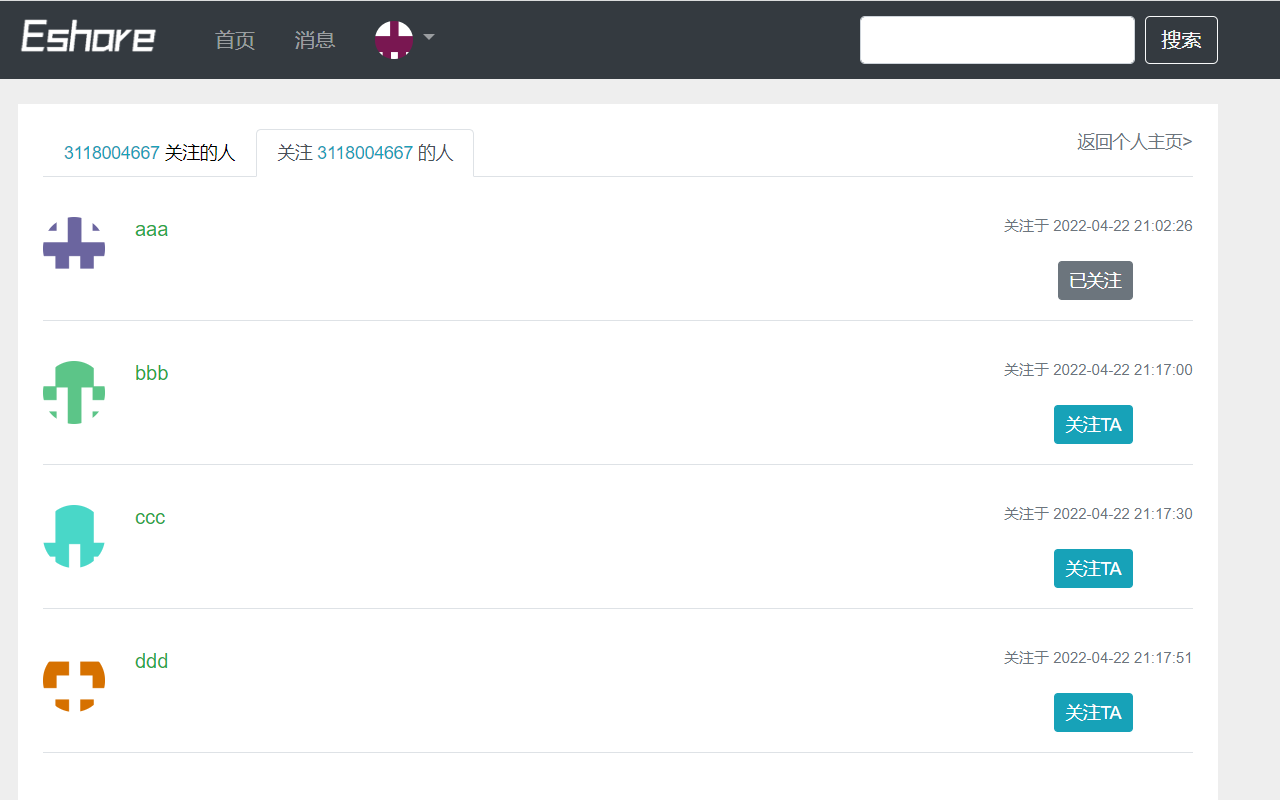


图 5.10 粉丝列表实现效果图

* 1. 账号设置

账号设置页面目前可以进行修改头像和修改密码操作，当用户点击文件按钮时，系统弹出资源管理器，让用户选择头像文件，系统为用户选择的文件生成一个随机的文件名，用户点击立即上传，服务器将头像文件存入磁盘，返回主界面，更新头像。



图 5.11 账号设置实现效果图

### 搜索模块实现

通过在导航栏右侧的输入框中输入需要查询的关键词，点击搜索按钮，系统返回搜索结果。搜索结果以列表的形式展示，每一项都显示完整的帖子信息和相关的作者信息。

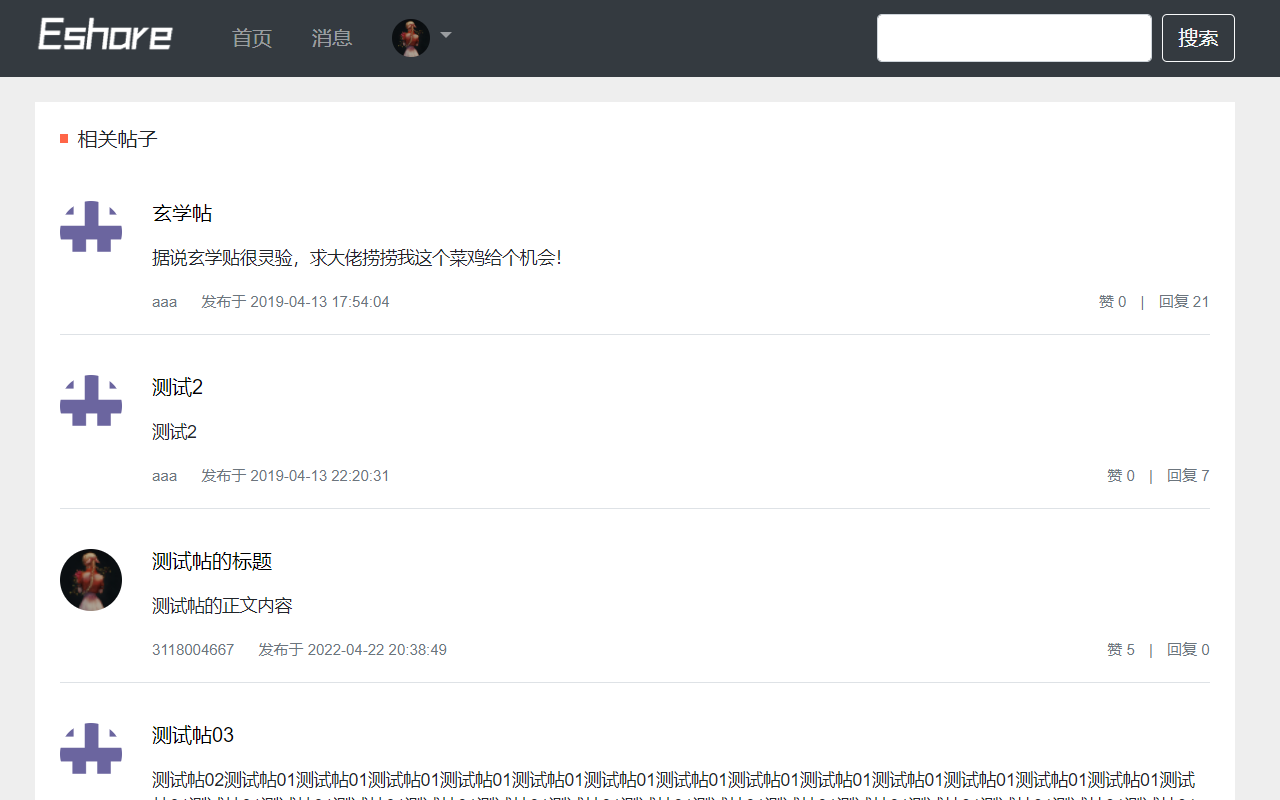


图 5.12 搜索功能实现效果图

## 本章小结

到现在为止，Eshare经验分享平台的基本工作已经完成，系统的功能已经基本实现，比较好的满足了需求分析中提到的个个功能要求，且各个模块在编码阶段各个模块完成是进行的单元测试都顺利通过，目前只需要对系统进行整体的系统的综合测试，即可发布上线。

# 测试

测试阶段是对整个系统各个功能模块的实现效果，系统的整体性能。安全等方面进行测评。本章主要对Eshare经验分享平台进行统一集中的综合测试，综合测试的内容包括：在实际服务器上，对Eshare经验分享平台的各个功能模块逐个进行测试，同时还要对整个系统进行包括响应时间，系统安全等方面的非功能性测试。

## 测试环境

实际测试时使用阿里云服务器作为项目部署环境，同时通过在不同客户端分别验证项目各个功能的实现效果验证整个系统的兼容性。

**表 6.1 测试环境表**

|  |  |
| --- | --- |
| 设备名 | 配置信息 |
| 服务器 | CPU：Intel Xeon Platinum 8269CY 2.5GHz 2核  操作系统：centos7.6  内存：8G |
| 客户端01 | 操作系统：Windows 10  软件版本：Google Chrome 100.0.4 |
| 客户端02 | 操作系统：Windows 10  软件版本：FireFox 99.0.1 |
| 客户端03 | 操作系统：Windows 10  软件版本：Yandex 22.3.2 |

## 功能测试

功能测试是对系统的各个功能在真实运行环境下可能面临的各种情况进行模拟，在正式上线之前进行整体的功能测试，对整个项目的功能做最后的验证工作。

### 注册登录模块功能测试

注册登录模块的测试主要围绕以下几点：首次登录用户登录操作、登录凭证有效期内的用是否免登录、注册账号后是否能正常收到激活邮件、点击激活邮件中的激活链接之后是否能正常使用账号登录。

**表 6.2 注册登录模块功能测试表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试项 | 操作步骤 | 预期结果 | 测试结果 |
| 01 | 登录 | 点击导航栏登录标签、  提交登录信息 | 登录成功 | 通过 |
| 02 | 免登录 | 直接访问项目URL | 无需登录 | 通过 |
| 03 | 注册 | 点击导航栏注册标签、  提交注册信息 | 收到激活邮件 | 通过 |
| 04 | 激活账号 | 点击激活邮件中的激活链接 | 激活成功 | 通过 |

### 帖子模块功能测试

帖子模块的功能测试主要包括：首页帖子列表是否正常显示、点击帖子列表中的任何一项是否能正常跳转至帖子的详情页、点击帖子列表的任何一项的作者头像是否能正常跳转至作者首页、在帖子详情页执行点赞、评论操作时响应是否正常。

**表 6.3 帖子模块功能测试表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试项 | 操作步骤 | 预期结果 | 测试结果 |
| 01 | 帖子列表 | 直接访问项目首页 | 显示帖子列表 | 通过 |
| 02 | 发布帖子 | 点首页我要发布按钮、  输入帖子信息后发布帖子 | 提示发布成功 | 通过 |
| 03 | 作者主页 | 点击帖子列表项中作者头像 | 跳转至作者主页 | 通过 |
| 04 | 评论 | 进入帖子详情页、  在页面底部输入评论内容  点击评论 | 激活成功 | 通过 |
| 05 | 点赞 | 帖子详情页标题右侧点赞按钮 | 显示已赞 | 通过 |

### 消息模块功能测试

消息模块的功能测试主要进行以下工作：点击导航栏的消息标签是否能正常显示消息列表、点击消息列表的任何一个会话是否能正常显示消息详情、在会话详情页面点击回复私信是否能成功向对方发送私信、在首页向任何人发送私信是否成功。

**表 6.4 消息模块功能测试表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试项 | 操作步骤 | 预期结果 | 测试结果 |
| 01 | 私信列表 | 点击导航栏消息标签 | 显示会话列表 | 通过 |
| 02 | 回复私信 | 私信列表页点击给他私信、  输入私信内容后发送 | 提示回复成功 | 通过 |
| 03 | 发送私信 | 消息主界面点击发私信、  输入私信信息、  发送私信 | 提示发送成功 | 通过 |
| 04 | 系统通知 | 私信列表界面切换至通知标签 | 展示通知列表 | 通过 |

### 系统设置模块功能测试

系统设置模块的功能测试主要进行以下内容：导航栏中下拉菜单中点击个人主页是否能正常显示、点击个人主页粉丝数量是否能正确显示粉丝列表、点击关注数量是否能正确显示关注列表、下拉菜单中点击账号设置后进行上传头像操作是否能成功、点击退出登录操作是否成功。

**表 6.5 系统设置模块功能测试表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试项 | 操作步骤 | 预期结果 | 测试结果 |
| 01 | 个人主页 | 点击导航栏下拉菜单、  点击个人主页标签 | 展示个人信息 | 通过 |
| 02 | 粉丝列表 | 进入个人主页、  点击粉丝数量 | 显示分析列表 | 通过 |
| 03 | 关注列表 | 进入个人主页、  点击关注数量 | 显示关注列表 | 通过 |
| 04 | 上传头像 | 点击导航栏下拉菜单账号设置、  账号设置页点击上传头像、  选择文件提交 | 头像更新 | 通过 |
| 05 | 退出登录 | 点击导航栏下拉菜单退出登录 | 进入未登录状态 | 通过 |

### 搜索与后台管理模块功能测试

搜索与后台管理模块主要进行以下测试：搜索框中输入关键词后点击搜索是否能正确显示搜索结果、使用管理员账号登录状态下后台管理地址能否正常访问。

**表 6.6 搜索与后台管理模块功能测试表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试项 | 操作步骤 | 预期结果 | 测试结果 |
| 01 | 搜索功能 | 在导航栏右侧搜索框输入关键词、  点击搜索按钮 | 搜索结果列表 | 通过 |
| 02 | 数据统计 | 使用管理员账号登录、  地址栏输入管理员访问路径、  选择日期范围 | 显示统计结果 | 通过 |

## 非功能测试

对Eshare经验分享平台的非功能测试主要包括两个内容：性能测试（主要是页面响应时间测试）、兼容性测试。通过这两项指标验证项目在实际上线之后能否应对各种使用场景，是否能够及时处理用户请求。

1. 性能测试

本次性能测试主要针对用户实际使用时常用的，比较影响用户使用体验的几个操作进行测试。测试结果如下：

**表 6.7 性能测试表**

|  |  |  |
| --- | --- | --- |
| 测试项目 | 平均响应时间(ms) | 最慢响应时间(ms) |
| 访问首页 | 653 | 1567 |
| 发布帖子 | 861 | 1865 |
| 发布评论 | 865 | 984 |
| 点赞 | 324 | 462 |
| 私信 | 1562 | 3287 |
| 接收系统通知 | 1867 | 2845 |

1. 兼容性测试

对于B/S架构的项目而言，由于用户可能使用不同的浏览器访问系统，因此，在正式上线前需要特别对主流的浏览器进行兼容性测试，以确保项目在正式发布之后尽可能的满足更多用户的使用需求。本次兼容性测试主要对三款使用最多的浏览器进行测试，包括谷歌浏览器、火狐浏览器、Yandex浏览器，以下是本次测试的结果：

**表 6.8 兼容性测试表**

|  |  |  |
| --- | --- | --- |
| 浏览器 | 测试方法 | 测试结果 |
| FireFox 99.0.1 | 对整个项目的各个功  能模块逐个测试 | 通过 |
| Google Chrome 100.0.4 | 通过 |
| Yandex 22.3.2 | 通过 |

## 本章小结

本章对Eshare经验分享平台的各个功能模块进行了逐个测试，测试结果显示项目整体实现符合预期，能较好的完成需求分析中的各种使用需要，同时对系统的响应时间，对各种浏览器的兼容性也进行了测试，非功能性测试的测试结果显示，Eshare经验分享平台具有良好的响应性能，同时与三款主流浏览器兼容性良好。

# 总结与展望

## 总结

本文主要记录了Eshare经验分享平台的整个开发过程，对每个开发阶段都进行了详细的介绍。包括通过前期对经验分享平台相关产品的市场现状、用户规模、开发周期等信息的深入调研，确定项目目标后，结合实际需要和个人技术水平制定了Eshare经验分享平台的需求分析相关内容。以需求分析为导向，通过概要设计确定了Eshare经验分享平台的各个功能模块具体应该实现的功能，在此基础上设计了整个系统的数据库表。在详细设计阶段，依据每个功能的流程特点，逐个设计每个功能具体实现时所需要的类，以及类之间的调用关系。项目进入编码阶段，按照设计好的类及其调用关系逐个实现系统所有功能。最后，通过系统的综合测试检验各个功能的实现效果，以及系统整体的性能和兼容性。

项目最终比较好的实现了需求分析所要求的系统设计目标，包括核心的帖子模块和消息模块、帖子的搜索功能等，基本实现了一个论坛系统所应该具备的所有功能。在项目实现过程中，尽可能的使用了当前最流行的技术组合，例如项目后端使用的SSM框架、前端使用的Bootstrap框架、前后端数据交互使用的Thymeleaf模板引擎、搜索功能所使用的ElasticSearch服务、系统通知功能使用的Kafka消息队列、点赞功能和关注功能使用的Redis缓存等，这些技术都是当前互联网相关产品的主流技术，采用这些流行的技术，有利于项目的后期维护和持续发展。

## 展望

通过本次设计已经完成了一个简洁美观的论坛系统，整个系统的功能已经可以满足日常使用，但是对整个论坛系统的长期发展而言，开发完成并不是终点，一个好的产品都是在不断迭代中逐步完善系统功能，对于Eshare经验分享平台而言，还要注意以下几个方面的问题：

（1）Eshare经验分享平台目前只针对少量用户进行了测试，实际应用场景中面对大量用户的访问是否能够只通过增加服务器的数量提升系统的并发访问性能，如何调整项目结构能够提升项目的并发性能。

（2）任何一个优秀的产品都是在听取大量用户反馈的基础上不断改进不足，最终形成比较稳定的界面和功能结构，由于当前用户数量较少，目前的Eshare经验分享平台并没有设计便捷的用户返回渠道，后续视平台的发展情况，需要及时添加用户反馈渠道。

（3）对于一个成功的论坛类产品而言，产品开发完成只是一小步，更加重要的是如何吸引高质量的内容，通过高质量内容吸引更多高质量的内容生产者加入到论坛系统中，才是项目的最终目标。

# 参考文献

[1] 贾佳,宋恩梅,苏环.社会化问答平台的答案质量评估——以“知乎”、“百度知道”为例[J].信息资源管理学报,2013,3(02):19-28.

[2] 贺勋.在线中文问答社区答案质量预测研究[D].齐鲁工业大学,2020.DOI:10.27278/d.c-nki.gsdqc.2020.000184.

[3] 沈洪洲,史俊鹏,马巧慧.社会化问答社区回答内容质量影响特征研究——以“知乎”为例[J].情报杂志,2020,39(10):169-175+202.

[4] 第48次《中国互联网络发展状况统计报告》发布 超十亿用户接入互联网 我国成全球最庞大数字社会[J].网络传播,2021(09):76-81.

[5] 陈颖慧. 基于SSM的英语学习网站的设计与实现[D]. 华中科技大学, 2019.

[6] 杨华. 基于SSM的宠物店信息系统的设计与实现[D]. 吉林大学, 2019.

[7] 孟祥雪. 基于SSM的外聘人员人事管理系统设计与实现[D]. 电子科技大学, 2020.

[8] 张泷元. 基于SSM的企业网站设计与实现[D]. 黑龙江大学, 2019.

[9] 王睿. 基于SSM的企业设备管理系统设计与实现[D]. 东南大学, 2020.

[10] 王冯涛. 基于SSM的地铁站智慧管理系统的设计与实现[D]. 北京交通大学, 2021.

[11] 明日科技.Java从入门到精通(第5版) [M].清华大学出版社,2019

[12] 王珊，萨师煊.数据库系统概论（第5版）[M].高等教育出版社,2014

[13] Lei Xiafei,Wang Zhe,He yuzhen. The Data Management and Real-time Search Based on Elasticsearch[C]//Proceedings of 2015 4th International Conference on Computer,Mechatronics,Control and Electronic Engineering(ICCMCEE 2015): Atlantis Press, 2015: 825-829.

[14] Lv Taizhi,Zhang Juan,Chen Yingying. Job Analysis System Based on Spark Platform-[C]//Proceedings of 2020 3rd International Conference on Financial Management,Education and Social Science(FMESS 2020): Clausius Scientific Press, 2020: 315-319.

# 致谢

从一开始选题时的一头雾水，不确定选题方向，到最终完成设计和整个系统的完整实现，这整个过程持续了近半年的时间，在项目开发的这半年时间里，经过了无数次的出错、修改、调试，在这个循环往复的过程中，我从一个十分缺乏完整项目经验的项目小白，逐步成长为了一个能够独立处理许多开发问题的入门级工程师。这个过程中离不开网上许多乐于分享开发经验的开发者，更离不开对我悉心指导的朱鉴老师。

整个大学期间，与老师的交流非常少，但在毕业设计过程中，遇到项目相关问题都可以向担任我们毕业设计指导工作的朱鉴老师提问，朱老师日常工作很忙，有许多次都发现朱老师深夜还在工作，但对于学生的提问，朱老师每次回复都很及时，能够感受到朱老师对学生的友善与耐心，在这里由衷的感谢朱老师在项目设计过程中给予的耐心指导。同时，每当遇到一些项目中的小问题时，都能从网上找到曾经遇到相关问题的人所分享的解决方案，这些分享对于个人开发者而言十分重要，在这里也由衷的感谢每一个乐于分享自己遇到的难题和解决方法的人。