

**毕 业 设 计**

**题 目 实验室设备常见问题管理网站的设计与实现**

**英文题目 Design and Implementation of a Laboratory Equipment Common Problem Management Website**

**学生姓名**： **郭欢军** **申请学位门类**： **工学**

**学 号**：  **2019213227**

**专 业：** **软件工程**

**学 院**： **软件学院**

**指导教师：** **余小军** **职称：**

**二〇二三年四月十二日**

**作 者 声 明**

本人以信誉郑重声明：所呈交的学位毕业设计（论文），是本人在指导教师指导下由本人独立撰写完成的，没有剽窃、抄袭、造假等违反道德、学术规范和其他侵权行为。文中引用他人的文献、数据、图件、资料均已明确标注出，不包含他人成果及为获得东华理工大学或其他教育机构的学位或证书而使用过的材料。对本设计（论文）的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本毕业设计（论文）引起的法律结果完全由本人承担。

本毕业设计（论文）成果归东华理工大学所有。

特此声明。

毕业设计（论文）作者（签字）：郭欢军

签字日期： 2023年 4 月 19 日

本人声明：该学位论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

学位论文指导教师签名：

年 月 日

**基于SpringBoot的实验室设备常见问题管理网站的设计与实现**

**\*\*\*\*\*\*（名字）**

**Design and Implementation of a Laboratory Equipment Common Problem Management Website**

\*\*\*\*\*\*\*（英文名字）

**2023年4月12日**

摘要

随着近年来计算机实验室研究人员越来越多，因而出现的问题也就越来越多，为了避免人们在一样的错误上浪费自己宝贵的时间，所以设计了这个实验室常见设备问题管理网站。现在各个高校教学设备众多，随之的问题也会越来越多， 问题出现后如果再次复现，没有解决人在场，我们又需要浪费大量的时间去找解决途径。而实验室设备常见问题管理网站可以记录每个人遇到的任何问题， 以及解决方法，下次再出现此类的问题，只需要在网站上面找到对应的方法，按照方法进行就可以，如果找不见也可以发帖子求救， 有遇到此类问题的研究人员也可以通过评论的方式帮助提问人解决问题。网站使用MySQL作为数据库，前台页面主要采用了HTML、CSS、JavaScript、Layui、Axious、jQuery等技术，后端使用Spring Boot、Mybatis、Mybatis-Plus等技术，实现web页面的人机交互。

**关键词：**实验室；常见问题；Spring Boot；JavaScript

**ABSTRACT**

With the increasing number of computer lab researchers in recent years, there will be more and more problems. In order to avoid people wasting their precious time on the same mistake, so the design of this laboratory common equipment problem management website. Now there are so many teaching equipment in various colleges and universities that there will be more and more problems. If the problems occur again and there is no solver present, we need to waste a lot of time to find solutions. The laboratory equipment management website can record any problems encountered by everyone, and the solution, the next time there are such problems, just need to find the corresponding method on the website, according to the method can be carried out, if you can not find can also send a post for help, Researchers who have encountered this problem can also help the questioner solve it by commenting on it. The website uses MySQL as the database, the front page mainly uses HTML, CSS, JavaScript, Layui, Axious, jQuery and other technologies, and the back end uses Spring Boot, Mybatis, Mybatis-Plus and other technologies to achieve human-computer interaction of web pages.

**Key Word：** laboratory；frequently asked question；Spring Boot；JavaScript

目 录

[第1章 绪论 1](#_Toc101458295)

[1.1 开发背景、目的和意义 1](#_Toc101458296)

[1.2 国内外研究现状 1](#_Toc101458297)

[1.2.1 国外研究现状 1](#_Toc101458298)

[1.2.2 国内研究现状 2](#_Toc101458299)

[1.3 主要研究内容 2](#_Toc101458300)

[第2章 相关技术介绍 3](#_Toc101458301)

[2.1 SPRINGBOOT框架 3](#_Toc101458302)

2.2 MYBATIS 3

[2.3 MYBATIS-PLUS 3](#_Toc101458304)

[2.4 THYMELEAF 4](#_Toc101458305)

[2.5 LAYUI 4](#_Toc101458306)

[2.6 AJAX 4](#_Toc101458307)

[2.7 MYSQL 4](#_Toc101458308)

2.8 CHART.JS

2.9 WANGEDITOR

2.10 NGINX

[第3章 系统需求分析 5](#_Toc101458309)

[3.1 可行性分析 5](#_Toc101458310)

[3.1.1 技术可行性 5](#_Toc101458311)

[3.1.2 操作可行性 5](#_Toc101458312)

[3.1.3 经济可行性 5](#_Toc101458313)

[3.2 功能需求分析 5](#_Toc101458314)

[3.3 数据流程分析 7](#_Toc101458316)

[第4章 系统详细设计 9](#_Toc101458317)

[4.1 系统架构 9](#_Toc101458318)

[4.2 系统功能模块设计 9](#_Toc101458319)

[4.3 系统用户用例图 13](#_Toc101458320)

[4.4 数据库设计 14](#_Toc101458321)

[4.4.1 概念模型设计 14](#_Toc101458322)

[4.4.2 数据库表设计 15](#_Toc101458323)

[第5章 系统实现 19](#_Toc101458324)

[5.1 登录界面 19](#_Toc101458325)

[5.2 注册界面 19](#_Toc101458325)

[5.3 管理界面 19](#_Toc101458326)

[5.4 用户管理 20](#_Toc101458327)

[5.5 问题管理 22](#_Toc101458330)

[5.6 22](#_Toc101458331)查看问题

[5.7 23](#_Toc101458332)个性发布

[5.8 23](#_Toc101458333)收藏评论

[5.9 25](#_Toc101458334)个人信息

[5.10 25](#_Toc101458334)学霸用户息

[第6章 系统测试 29](#_Toc101458337)

[6.1 测试的目的 29](#_Toc101458338)

[6.2 测试方法 29](#_Toc101458339)

[6.3 测试用例 29](#_Toc101458340)

[6.4 测试结论 29](#_Toc101458341)

[结 论 30](#_Toc101458342)

[致 谢 31](#_Toc101458343)

[参考文献 32](#_Toc101458344)

# 第1章 绪论

## 1.1 开发背景、目的和意义

实验室设备管理工作中，设备管理由于数量过多容易出错，如果使用计算机技术进行问题记录，则能够大幅度减轻管理人员的工作强度，提高工作效率。现在各个高校教学设备众多，随之的问题也会越来越多， 问题出现后如果再次复现，没有解决人在场，我们又需要浪费大量的时间去找解决途径。而实验室设备常见问题管理网站可以记录每个人遇到的任何问题， 以及解决方法，下次再出现此类的问题，只需要在网站上面找到对应的方法，按照方法进行就可以，如果找不见也可以发帖子求救， 有遇到此类问题的研究人员也可以通过评论的方式帮助提问人解决问题。

## 1.2 国内外研究现状

### 1.2.1 国外研究现状

国外的实验室设备问题管理网站相对较多，其中一些比较具有代表性的网站如下：

1. Labguru：该网站提供实验室管理解决方案，包括设备管理、实验数据管理、项目管理等。其中设备管理模块可以帮助实验室管理人员跟踪设备维护记录、预定设备使用时间等。
2. Quartzy：该网站提供实验室管理平台，包括设备管理、试剂管理、订单管理等。其中设备管理模块可以帮助实验室管理人员跟踪设备状态、维护记录等。
3. MyLabBook：该网站提供实验室管理解决方案，包括设备管理、实验数据管理、项目管理等。其中设备管理模块可以帮助实验室管理人员跟踪设备使用情况、维护记录等。
4. LabArchives：该网站提供实验室数据管理平台，包括实验数据管理、文档管理、项目管理等。其中设备管理模块可以帮助实验室管理人员跟踪设备使用情况、保养记录等。 总的来说，国外的实验室设备问题管理网站注重实验室管理的全面性和系统性，提供了多种实验室管理功能，为实验室的设备问题管理提供了多种解决方案。

### 1.2.2 国内研究现状

近年来，随着实验室设备的不断增多和复杂化，实验室设备问题管理的重要性日益凸显。国内一些高校和科研机构开始关注和研究实验室常见设备问题管理网站的设计和实现，取得了一些进展。

一些高校和科研机构建立了自己的设备问题管理系统，例如华中科技大学的实验室设备管理系统、中山大学的实验室设备管理系统等。这些系统主要包括设备信息管理、设备维护管理、设备维修管理、设备借用管理等功能模块，通过实现设备信息的集中管理、设备维护计划的制定和执行、设备维修申请和记录、设备借用申请和归还等功能，提高了设备的使用效率和可靠性，方便了设备管理人员和用户的操作。 此外，一些研究机构也对实验室设备问题管理网站进行了研究和探索。例如，中国农业科学院农业信息研究所的研究人员提出了一种基于Web的实验室设备管理系统，实现了实验室设备信息的集中管理、设备维护计划的制定和执行、设备维修申请和记录、设备借用申请和归还等功能，并且通过实验验证了系统的可行性和实用性。 总之，国内一些高校和科研机构已经开始关注和研究实验室常见设备问题管理网站的设计和实现，取得了一些进展。未来，随着实验室设备的不断增多和复杂化，实验室设备问题管理网站的研究和应用将会得到更广泛的关注和应用。

## 1.3 主要研究内容

1. 系统需求分析：对实验室设备问题管理网站的使用者、管理者和开发者进行需求调研和分析，确定系统功能和性能的需求，为后续系统设计和实现提供基础。
2. 系统架构设计：根据需求分析结果，设计实验室设备问题管理网站的系统架构和模块，并确定各个模块的功能和接口，为后续具体功能的实现提供基础。
3. 数据库设计：设计实验室设备问题管理网站的数据库结构，包括设备信息、设备维护计划、设备维修记录、设备借用记录等数据表，并确定数据表之间的关系和约束。
4. 系统开发和实现：根据需求分析和系统设计，使用相应的开发语言和开发工具进行系统开发和实现，包括前端界面设计和实现、后端业务逻辑设计和实现、数据库设计和实现等。
5. 系统测试和验收：对实验室设备问题管理网站进行系统测试和验收，包括功能测试、性能测试、安全测试等，确保系统的可靠性和稳定性。
6. 系统维护和优化：对实验室设备问题管理网站进行系统维护和优化，包括系统升级、数据备份、安全防护等，保证系统的正常运行和稳定性。 总之，实验室常见设备问题管理网站国内的主要研究内容涵盖了系统需求分析、系统架构设计、数据库设计、系统开发和实现、系统测试和验收、系统维护和优化等多个方面，这些研究内容为实验室设备问题管理网站的设计和实现提供了理论基础和实践指导。

# 

# 第2章 相关技术介绍

## 2.1 SPRINGBOOT框架

Spring Boot 自动配置了大部分常见的配置，使得开发者可以更加专注于业务逻辑的实现，避免了繁琐的配置过程。Spring Boot 提供了丰富的开箱即用的功能模块，例如：数据库访问、模板引擎、安全认证等，可以快速地构建出一个可用的 Web 应用程序。Spring Boot 对微服务提供了良好的支持，可以方便地构建和部署微服务应用程序。Spring Boot 提供了丰富的开发工具和插件，可以提高开发效率。框架采用模块化和自动化配置的方式，可以使得代码结构更加清晰，易于维护和升级。Spring Boot 框架采用松耦合的设计，可以方便地扩展和定制自己的功能模块，同时也能够与其他框架和组件进行集成。 Spring Boot 框架帮助开发者快速地构建出高效、可靠、可扩展的 JavaWeb 应用程序，从而提高开发效率和生产力。

## 2.2 MYBATIS框架

MyBatis 是一种基于 Java 的持久层框架，它可以帮助开发者简化数据库访问层的开发，提高开发效率和代码质量。采用 SQL 映射文件的方式，使得开发者可以灵活地编写自己的 SQL 语句，同时也可以方便地与其他框架进行集成。提供了简单易用的 API，可以帮助开发者快速地实现数据库访问层的功能，避免了繁琐的 JDBC 编程。支持多种数据库，同时也支持多种 SQL 映射方式，例如：XML 文件、注解方式等，可以满足不同项目的需求。MyBatis 的 SQL 映射文件可以使代码结构更加清晰，易于维护和升级，同时也可以方便地进行单元测试和调试。采用了预编译 SQL 语句、缓存机制等优化方式，可以提高数据库访问效率，从而提高系统的性能。可以帮助开发者快速、简单、高效地实现数据库访问层的功能，在 JavaWeb 项目中具有重要的作用。

## 2.3 MYBATIS-PLUS框架

MyBatis-Plus 是在 MyBatis 基础之上的一个增强工具包，它提供了很多便捷的功能，可以进一步简化数据库访问层的开发。提供了很多便捷的 API，可以帮助开发者减少重复代码的编写，例如：分页查询、条件构造器等。MyBatis-Plus 提供了很多基于代码生成器的快速开发功能，可以快速生成实体类、Mapper 接口和 XML 映射文件等，从而提高开发效率。代码结构清晰，易于维护和升级，同时也方便进行单元测试和调试。MyBatis-Plus 支持多种数据库，例如：MySQL、Oracle、SQL Server 等，可以满足不同项目的需求。提供了多种主键策略，例如：自增、UUID、雪花算法等，可以方便地满足不同项目的需求。支持 Lambda 表达式，可以使查询条件更加简洁、易读，同时还能有效避免 SQL 注入攻击。

## 2.4 THYMELEAF模板

Thymeleaf 是一种模板引擎，可以将模板和数据进行结合，生成最终的 HTML 页面。Thymeleaf 提供了很多便捷的语法，可以帮助开发者减少重复代码的编写，例如：条件判断、循环等。可以直接在 HTML 页面中使用，不需要额外的学习成本，可以快速开发出美观、清晰的页面。Thymeleaf 支持多种模板，例如：HTML、XML、文本、JavaScript 等，可以满足不同项目的需求。提供了国际化支持，可以方便地实现多语言切换等功能。Thymeleaf 支持表单绑定，可以将表单数据绑定到实体类中，简化表单提交的操作。兼容 HTML、XHTML、HTML5 等多种版本的 HTML，可以方便地与其他框架进行集成。可以帮助开发者快速、简单、高效地实现页面的渲染和数据绑定。

## 2.5 LAYUI

Layui 是一款基于 jQuery 的前端 UI 框架，它提供了很多简单易用的组件和样式，可以方便地实现页面的美化和交互效果。Layui 提供了很多简单易用的组件和样式，可以帮助开发者减少页面代码的编写，例如：表格、表单、图表等。Layui 的组件和样式可以快速实现页面的美化和交互效果，可以提高开发效率。Layui 兼容多种浏览器，例如：IE、Chrome、Firefox 等，可以满足不同用户的需求。支持响应式布局，可以根据不同设备的屏幕大小自适应显示页面内容，提高用户体验。提供了丰富的定制化选项，可以根据项目需求进行个性化设置。有庞大的社区支持，提供了很多示例代码和解决方案，可以方便地解决开发过程中的问题。

## 2.6 AJAX

AJAX（Asynchronous JavaScript and XML）是一种前端技术，可以实现异步加载数据和更新页面内容，同时不会导致页面的刷新或重载。可以实现页面的无刷新更新，可以提高用户体验，减少页面刷新的时间和流量消耗。AJAX 可以实现局部刷新，可以使页面更加动态和实时。AJAX 可以减少不必要的请求和响应，可以降低服务器的压力，提高页面的性能。AJAX 可以支持多种数据格式，例如：JSON、XML、HTML 等，可以满足不同项目的需求。AJAX 可以与其他技术集成，例如：jQuery、Vue.js 等，可以提高开发效率和功能实现。AJAX 提供了良好的错误处理机制，可以有效避免因为请求错误导致的程序异常。

## 2.7 MYSQL

MySQL 是一种开源的关系型数据库管理系统，可以存储和管理大量数据。MySQL 可以存储和管理大量数据，可以满足 JavaWeb 项目对数据存储和管理的需求。支持可扩展性，可以随着数据量的增大进行水平和垂直扩展，可以满足不同项目的需求。MySQL 的查询速度快，可以快速响应用户请求，提高系统的性能。  具有较高的可靠性，可以实现数据备份和恢复，可以有效避免数据丢失或损坏的情况。支持数据加密和用户权限管理等安全机制，可以保护数据的安全性和隐私性。MySQL 可以与 JavaWeb 项目的其他技术进行集成，例如：JDBC、Hibernate 等，可以方便地实现数据的读写操作。

## 2.8 CHART.JS

Chart.js是一个功能强大、灵活且易于使用的JavaScript图表库，它可以在Web项目中实现各种类型的图表和数据可视化。Chart.js支持动态和交互式图表，用户可以通过鼠标悬浮、点击等方式与图表进行交互，从而更加直观地理解数据。 Chart.js具有丰富的图表类型和样式，可以实现各种复杂的数据可视化效果，帮助用户更好地理解数据。Chart.js是一个基于JavaScript的库，可以与JavaWeb项目无缝集成，可以轻松地在JSP页面中嵌入图表，也可以通过Ajax等方式动态生成图表。Chart.js具有简单易用的API和丰富的文档，开发成本相对较低，而且可以快速地实现各种复杂的数据可视化需求。可以在各种设备和浏览器上运行，无需安装任何插件或扩展，具有良好的跨平台性。

## 2.9 WANGEDITOR

WangEditor是一款基于JavaScript和jQuery开发的富文本编辑器，它提供了丰富的文本编辑功能和易用的操作方式，可以轻松地集成到JavaWeb项目中。WangEditor提供了丰富的文本编辑功能，包括格式化、插入图片、插入视频等，可以大大提升用户在JavaWeb项目中的文本编辑体验。基于JavaScript和jQuery开发，可以轻松地集成到JavaWeb项目中，无需安装任何插件或扩展，而且具有良好的跨平台性。WangEditor提供了丰富的配置选项，可以根据需要对编辑器的样式和功能进行定制，满足JavaWeb项目的不同需求。WangEditor可以在各种设备和浏览器上运行，无需安装任何插件或扩展，具有良好的兼容性。WangEditor具有良好的安全性，可以有效地防止XSS攻击等安全漏洞。

## 2.10 NGINX

Nginx是一款轻量级的高性能Web服务器和反向代理服务器，采用事件驱动的异步非阻塞架构，能够处理更多的并发请求，提高JavaWeb项目的性能和响应速度。可以作为JavaWeb项目的反向代理服务器，实现负载均衡和高可用性，可以有效地分摊服务器的负载，提高项目的可靠性和稳定性。Nginx支持静态资源的缓存，可以缓存JavaWeb项目中的静态文件，减轻后端服务器的负载，提高JavaWeb项目的访问速度。 具有良好的安全性，可以通过配置SSL证书、限制IP访问等方式保障JavaWeb项目的安全性。Nginx具有良好的可扩展性和定制性，可以通过配置文件进行灵活的定制和扩展，满足JavaWeb项目不同的需求。

# 第3章 系统需求分析

## 3.1 可行性分析

### 3.1.1 技术可行性

前端技术：采用HTML、CSS、JavaScript、Layui、JQuery等前端技术实现网站的页面设计和交互效果。

后端技术：采用Java后端语言实现网站的业务逻辑和数据处理，使用后端框架SpringBoot提高开发效率和代码质量。

数据库技术：可以采用关系型数据库如MySQL存储网站的数据。

服务器技术：可以采用Nginx服务器软件实现网站的部署和运行。

### 3.1.2 操作可行性

基本用户：如果有账号可以直接登录网站，如果没有那需要使用真实邮箱进行注册，进入到首页可以查看全部文章，也可以按照分类查看，点击某一个文章可以进入详情页面，在详情页面可以进行点赞、收藏以及评论操作，如果是本人发布的文章则可以进行编辑操作；点击用户头像或者姓名可以直接跳转到用户详情界面，查看用户近期发布的帖子以及评论。在首页，用户还可以发布文章，也可以看到当前网站的学霸用户，学霸用户的定义是本网站文章浏览量前三的用户。在首页头部栏位右侧，可以点击用户头像，进入到用户中心，在用户中心可以看到用户的文章，收藏的文章，发布的评论；在基本设置中用户可以更改自己的一些属性。

管理员用户：用户基本用户的所有功能。此外，在首页可以通过按钮进入到管理界面，对本网站所有的非管理员用户进行管理，封号或者解封的操作。也可以对用户发布的文章进行管理，对文章进行封禁以及解封的操作。

### 3.1.3 经济可行性

购买一台2核2G的服务器一年使用权，花费100人民币；购买域名[www.ghjlj.cn](http://www.ghjlj.cn)一年使用权花费39人民币。合计花费139人民币。

## 3.2 功能需求分析

登录注册功能：用户通过访问网站登录界面，拥有账号者输入验证码后可以直接登录网站，无账号者需要使用邮箱获取邮箱验证码从而注册账号。

首页个性化展示：用户可以通过文章的分类进行展示对应的内容，也可以按照时间和文章评论量对文章排序，也可以搜索文章标题，模糊匹配文章展示；用户还可以通过图表的形式查看本网站文章访问量前三的用户；同时还可以通过友情连接访问至别的网站。

文章功能：用户通过点击首页文章标题，可以进入文章详情界面，查看文章内容也可以对文章进行评论和点赞收藏功能；如果访问的是自己的文章，则会有编辑按钮，通过编辑按钮可以进入文章的编辑界面；在首页，用户可以通过发布文章按钮进行文章发布功能。

用户信息功能：通过点击用户的名称或者头像，即可访问用户的主页，用户主页会展示该用户最近发布的文章和近期的评论，通过点击文章标题也可访问文章详情界面；点击首页头像，可以进入当前用户的主页，以及个人信息界面，当前用户可以通过此界面对该用户信息的修改；在用户中心，可以看到本人发布的全部文章和评论。

管理员功能：管理员拥有普通用户的所有功能。此外，管理员用户可以通过首页的管理界面按钮进入到管理员界面，对本网站的所有非管理员用户进行封号或者解封的操作，被封号的用户，在管理员设置的封禁时间之前不能登录网站，用户被解封后可以进入网站；同时也可对用户发布的文章进行管理，封禁文章和解封文章功能，被封禁的文章普通用户不能查看到该文章，解封后则可以被查看。

## 3.3 数据流程分析

登录注册后，用户信息将被保存至数据库；进入首页后，会将数据库中的文章和用户信息查询出来并展示在界面；用户对文章的增删改操作，同时也会将数据同步至数据库。用户对自己个人信息修改时，也会将信息保存。管理员对用户和文章进行管理操作的同时，会将信息更新至数据库。

# 第4章 系统详细设计

## 4.1 系统架构

用户模块：包括用户注册、登录、信息修改等功能。

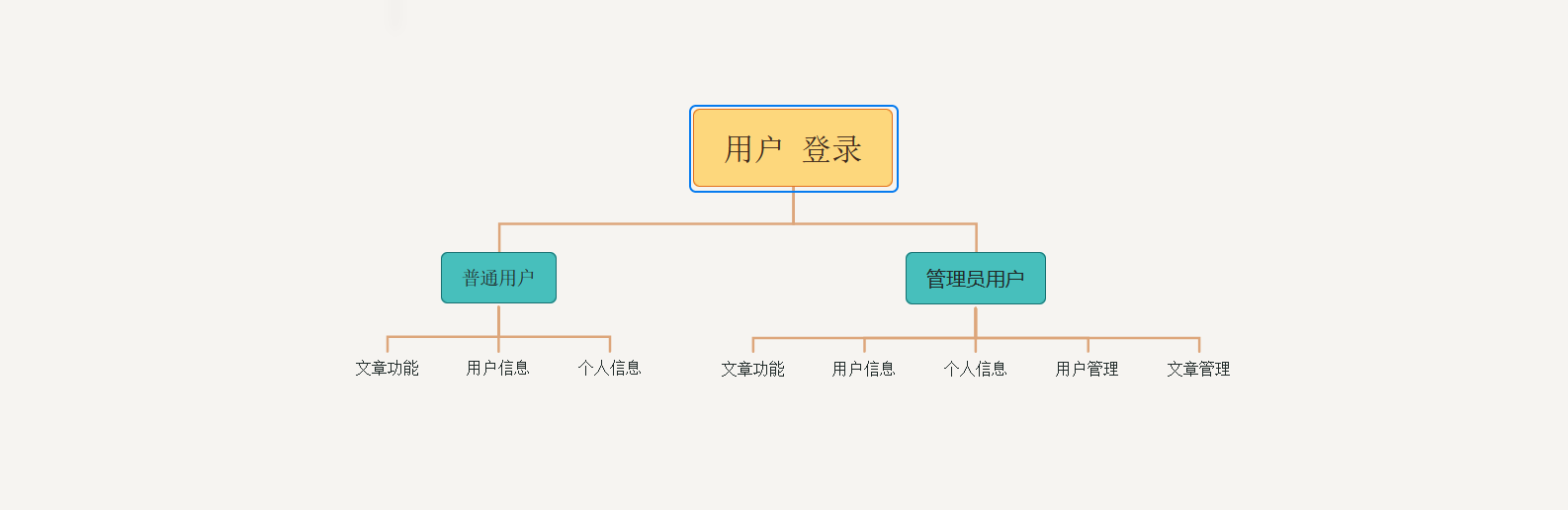
文章模块：包括文章发布、回复、点赞、收藏等功能。

后台管理模块：包括管理员登录、用户管理、文章管理等功能。

数据库模块：包括存储用户信息、文章信息、分类信息等数据。

服务器模块：包括网站的部署和运行，处理用户请求并返回响应。

前端模块：包括网站的页面设计和交互效果，实现用户与网站的交互。 该系统架构图采用了简单的三层架构模式，将系统分为表示层、应用层和数据层。表示层负责向用户展示网站的页面和交互效果，应用层负责处理用户请求和业务逻辑，数据层负责存储和管理数据。同时，该系统架构图也考虑了安全、可扩展性和可维护性等问题，为论坛网站的设计和实现提供了参考。

图4-1 系统架构图

## 4.2 系统功能模块设计

登录模块：使用已有账号登录，或者通过邮箱注册登录。

文章功能模块：支持文章的添加、修改、删除和查询等功能，接入富文本编辑器。

用户信息功能：支持用户信息的查看、修改功能。

管理员模块功能：支持对用户管理，文章管理。

## 4.3 系统用户用例图

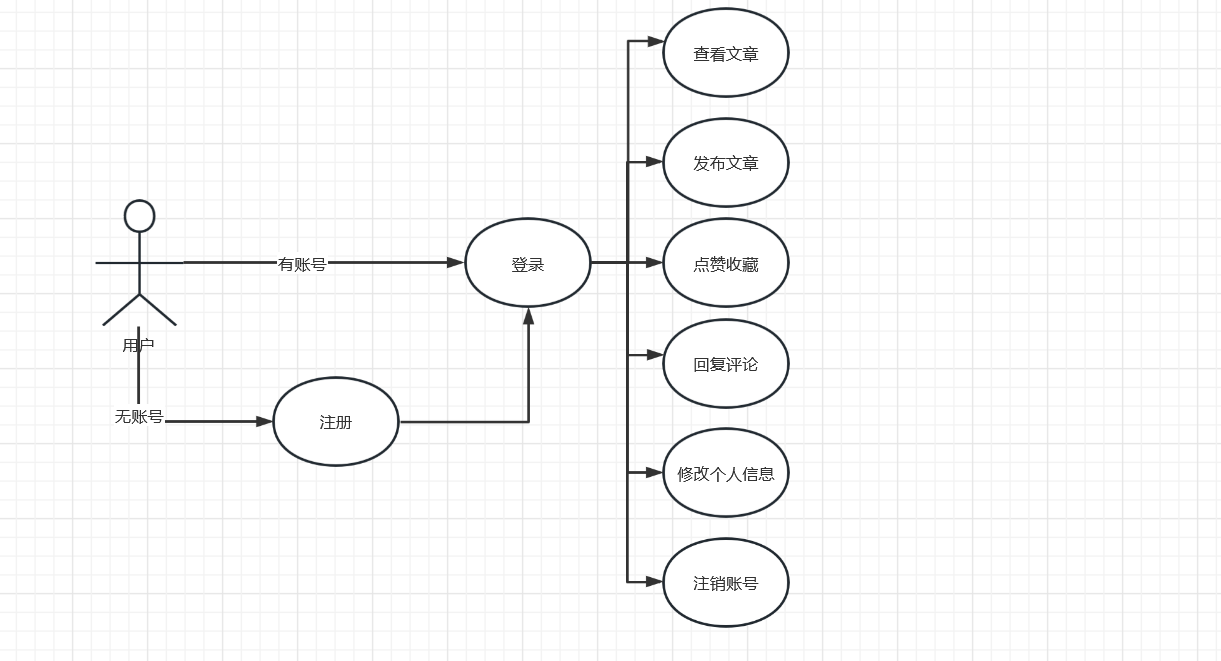


图4-2 系统用户用例图

## 4.4 数据库设计

### 4.4.1 概念模型设计

 用户实体：包括用户ID、用户名、密码、邮箱、头像、注册时间等属性。

 文章实体：包括文章ID、标题、内容、发布时间、更新时间、点赞数收藏数等属性。

 分类实体：包括分类ID、分类名等属性。

 评论实体：包括评论ID、评论内容、评论时间等属性。

 点赞收藏实体：记录用户对文章的点赞情况，包括点赞ID、用户ID、文章ID等属性。

 分类-文章关系：表示分类和文章之间的关系，一个分类可以包含多个文章，一个文章只能属于一个分类。

 用户-文章关系：表示用户和文章之间的关系，一个用户可以发布多个文章，一个文章只能属于一个用户。

 用户-评论关系：表示用户和评论之间的关系，一个用户可以发表多个评论，一个评论只能属于一个用户。

 文章-评论关系：表示文章和评论之间的关系，一个文章可以有多个评论，一个评论只能属于一个文章。

 用户-点赞收藏关系：表示用户和点赞收藏之间的关系，一个用户可以对多个文章点赞，一个文章可以被多个用户点赞。

### 4.4.2 数据库表设计

 用户表：用于存储用户信息，包括用户ID、用户名、密码、邮箱、头像、注册时间等字段。

CREATE TABLE `user` (

`user\_id` bigint NOT NULL AUTO\_INCREMENT COMMENT '用户主键id',

`login\_name` varchar(32) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL DEFAULT '' COMMENT '登陆名称(默认为邮箱号码)',

`password\_md5` varchar(32) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL DEFAULT '' COMMENT 'MD5加密后的密码',

`nick\_name` varchar(8) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL DEFAULT '' COMMENT '昵称',

`head\_img\_url` varchar(128) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL DEFAULT '' COMMENT '头像',

`gender` varchar(4) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL DEFAULT '' COMMENT '性别',

`location` varchar(4) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL DEFAULT '' COMMENT '居住地',

`introduce` varchar(32) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL DEFAULT '' COMMENT '个人简介',

`user\_status` tinyint NOT NULL DEFAULT '0' COMMENT '用户状态 0=正常 1=禁言 2=封号',

`last\_login\_time` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '最新登录时间',

`create\_time` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '注册时间',

`maturity\_time` datetime DEFAULT NULL COMMENT '禁言或者封号到期时间',

`role\_id` bigint DEFAULT NULL COMMENT '角色id',

PRIMARY KEY (`user\_id`) USING BTREE

) ENGINE=InnoDB AUTO\_INCREMENT=18 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci ROW\_FORMAT=DYNAMIC;

 分类表：用于存储帖子分类信息，包括分类ID、分类名等字段

CREATE TABLE `post\_category` (

`category\_id` int NOT NULL AUTO\_INCREMENT COMMENT '分类表主键',

`category\_name` varchar(16) COLLATE utf8mb4\_unicode\_ci NOT NULL COMMENT '分类的名称',

`category\_rank` int NOT NULL DEFAULT '1' COMMENT '分类的排序值 被使用的越多数值越大',

`dr` tinyint NOT NULL DEFAULT '0' COMMENT '是否删除 0=否 1=是',

`create\_time` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '创建时间',

PRIMARY KEY (`category\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci ROW\_FORMAT=DYNAMIC;

 文章表：用于存储帖子信息，包括帖子ID、标题、内容、分类ID、用户ID、发布时间、更新时间、点赞数、收藏数等字段。

CREATE TABLE `post` (

`post\_id` bigint NOT NULL AUTO\_INCREMENT COMMENT '帖子主键id',

`publish\_user\_id` bigint NOT NULL COMMENT '发布者id',

`post\_title` varchar(64) COLLATE utf8mb4\_unicode\_ci NOT NULL DEFAULT '' COMMENT '帖子标题',

`post\_content` mediumtext COLLATE utf8mb4\_unicode\_ci NOT NULL COMMENT '帖子内容',

`post\_category\_id` int NOT NULL COMMENT '帖子分类id',

`post\_category\_name` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL COMMENT '帖子分类(冗余字段)',

`post\_status` tinyint NOT NULL DEFAULT '1' COMMENT '0-用户被封 1-审核通过 2-审核失败 (默认审核通过)',

`post\_views` bigint NOT NULL DEFAULT '0' COMMENT '阅读量',

`post\_comments` bigint NOT NULL DEFAULT '0' COMMENT '评论量',

`post\_collects` bigint NOT NULL DEFAULT '0' COMMENT '收藏量',

`last\_update\_time` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '最新修改时间',

`create\_time` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '添加时间',

`dr` tinyint DEFAULT '0' COMMENT '逻辑删除',

PRIMARY KEY (`post\_id`) USING BTREE

) ENGINE=InnoDB AUTO\_INCREMENT=36 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci ROW\_FORMAT=DYNAMIC;

* 评论表：用于存储评论信息，包括评论ID、帖子ID、用户ID、评论内容、评论时间等字段。

CREATE TABLE `post\_comment` (

`comment\_id` bigint NOT NULL AUTO\_INCREMENT COMMENT '主键id',

`post\_id` bigint NOT NULL DEFAULT '0' COMMENT '关联的帖子主键',

`comment\_user\_id` bigint NOT NULL DEFAULT '0' COMMENT '评论者id',

`comment\_body` text CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL COMMENT '评论内容',

`parent\_comment\_user\_id` bigint NOT NULL DEFAULT '0' COMMENT '所回复的上一级评论的userId',

`comment\_create\_time` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '评论时间',

`dr` tinyint DEFAULT '0' COMMENT '是否删除 0-未删除 1-已删除',

PRIMARY KEY (`comment\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=25 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci ROW\_FORMAT=DYNAMIC;

* 点赞收藏表：用于存储用户对帖子的点赞收藏情况，包括点赞收藏ID、用户ID、文章ID等字段。

CREATE TABLE `post\_collect\_record` (

`record\_id` bigint NOT NULL AUTO\_INCREMENT COMMENT '主键id',

`post\_id` bigint NOT NULL DEFAULT '0' COMMENT '收藏帖子主键',

`user\_id` bigint NOT NULL DEFAULT '0' COMMENT '收藏者id',

`create\_time` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '收藏时间',

PRIMARY KEY (`record\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=24 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci ROW\_FORMAT=DYNAMIC;

 角色表：用于存放用户角色ID、角色名称等。

CREATE TABLE `role` (

`id` bigint NOT NULL AUTO\_INCREMENT COMMENT 'id',

`name` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_general\_ci NOT NULL COMMENT '角色名称',

`code` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_general\_ci NOT NULL COMMENT '角色编码',

`create\_time` datetime DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP COMMENT '创建时间',

`dr` tinyint NOT NULL DEFAULT '0' COMMENT '逻辑删除 1 删除 0 正常',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

# 第5章 系统实现

## 5.1 登录界面

## 5.2 注册界面

## 5.3 管理界面

## 5.4 用户管理

## 5.5 问题管理

## 5.6 查看问题

## 5.7 个性发布

## 5.8 收藏评论

## 5.9 个人信息

## 5.10 学霸用户

# 第6章 系统测试

## 6.1 测试的目的

## 6.2 测试方法

## 6.3 测试用例

## 6.4 测试结论

# 

# 结 论

# 

# 致 谢

# 

# 参考文献