



本科毕业设计

论文题目	用户访问行为感知的 IT 社区设计与实现
作者姓名	翟广福
所学专业名称	物联网工程
指导教师	马丽生

2020 年 4 月 23 日

学 生： (签字)

学 号： 2016212079

答 辩 日 期： 2020 年 5 月 19 日

指 导 教 师： (签字)

目录

摘要.....	1
Abstract	1
1 绪论.....	2
1.1 设计背景.....	2
1.2 设计目标	2
1.3 论文组织结构.....	2
2 可行性分析.....	2
2.1 系统架构概述.....	2
2.2 技术可行性.....	3
2.3 操作可行性.....	4
2.4 发展可行性.....	4
3 需求分析.....	4
3.1 系统功能概述	5
3.2 系统用例图描述.....	5
3.3 系统设计规则.....	8
4 系统设计.....	8
4.1 系统结构设计	8
4.2 数据库逻辑设计	9
4.3 数据库表设计	10
5 系统实现.....	16
5.1 系统前台实现	16
5.2 系统后台实现	28
5.3 文章推荐实现	33
6 系统测试与维护.....	35
6.1 系统测试概述	35
6.2 系统测试	35
结束语.....	38
参考文献.....	38
致 谢.....	39

用户访问行为感知的 IT 社区设计与实现

摘要：随着互联网、物联网等技术的发展，对于计算机类专业的学生，由于技术的更新、迭代非常快，需要一个具有学院特色的网络学习与知识分享的平台，来辅助专业学习和技术交流。为此，针对于本院学生的实际需求，本毕业设计利用 Java 相关技术设计，并实现了一个用户访问行为感知的 IT 社区系统。本文首先对该系统的设计背景等内容进行了描述；其次给出了系统的总体架构，并对系统可行性进行了分析；然后使用用例图对系统进行了需求分析；在此基础上，基于 Spring Boot、Vue.js 和 Python 等技术实现了该系统；最后对系统进行相关测试。本系统为学生信息交流和知识分享提供了一个平台，促进学生能力的提升。

关键词：IT 社区；系统；设计；行为感知

Design and Implementation of IT Community Based on User Access Behavior Perception

Abstract: With the development of the Internet, Internet of Things and other technologies, for the computer students, a network learning exchange and knowledge sharing platform with college characteristics is necessary to assist professional learning and technical exchange due to the rapid update and iteration of technologies. To this end, for the actual requirements of the students of our college, this graduation design uses Java related technologies to design and implement an IT community system based on user access behavior perception. This paper first describes design background of the system. Secondly, overall architecture and feasibility analysis of the system are given. Then system requirements are analyzed by using use case diagram. After that the system is implemented based on Spring Boot, Vue.js and Python. Finally, the system is tested. This system provides a platform for students' information exchange and knowledge sharing to promote the improvement of students' abilities.

Key words: IT community; system; design; behavior perception.

1 绪论

1.1 设计背景

随着互联网、物联网等技术的发展，大学生学习和交流的空间得以扩展，能够实现随时、随地学习和交流，能够利用网络资源学到所需要的知识，如在课程学习方面，有国家、省级精品课程、MOOC 课程，建有中国大学 MOOC、学堂在线等学习平台，有职业技能学习的网易云课堂、腾讯云课堂等。对于计算机类专业的学生，由于技术的更新、迭代非常快，更加需要一个专业的平台来辅助专业学习和技术交流。现有针对于计算机类专业的学习交流平台有 CSDN^[1]、博客园、牛客网和简书等，这些平台内容涉及计算机相关技术的各个方面且内容很丰富，对于常规的问题几乎都可以在这些平台上找到满意的答案。但是，由于这些平台都是通用平台，面向所有的计算机相关专业学生，对于一些特定的、针对性的技术问题，学生就很难在这些平台上找问题的解决办法。对于还在校学习的大学生，需要一个更有针对性的专业学习交流与知识分享社区辅助专业学习，从而更好地学好专业知识和技术，提升专业能力。为此，本毕业设计设计并实现了一个用户访问行为感知的 IT 社区系统，系统可以为计算机类专业学生提供一个分享、展现以及提高自我的平台。

1.2 设计目标

针对知识交流、技术分享、学生自我展示，特色专业知识解答等需求，所开发的用户访问行为感知的 IT 社区，实现了以文章形式为主体的知识交流和技术分享功能，以展示自己的作品以及共同协作的案例功能，以用户提交需求寻求他人帮助的需求交易功能。通过用户访问行为感知，实现个性化推荐，从而帮助专业学生更高效地获取问题解决方案以及实现知识共享。

1.3 论文组织结构

本论文分为六大部分。第一部分绪论，主要叙述了系统的设计背景，设计目标以及论文组织结构；第二部分可行性分析，从系统架构、技术可行性、操作可行性以及发展可行性四个方面分析本项目；第三部分需求分析，通过用例图对系统的功能进行了需求分析；第四部分系统设计，介绍了系统的结构和数据库设计；第五部分系统实现，主要叙述了系统前台各个模块，以及系统后台管理功能的实现；第六部分系统测试，主要介绍了对该系统的测试方法、测试过程以及测试结果。

2 可行性分析

在本项目开始设计之前，首先应该明确系统的开发架构，然后基于此架构进行系统可行性分析。一个软件项目能否成功实现，平稳的上线运行以及长久使用，是通过对项目进行可行性分析得出的结果。基于项目可行性分析，可以对项目进行需求分析以及合适的规划，可以最大程度的节省人力、物力以及时间方面的输出^[2]。针对所开发的系统，主要从系统架构、技术可行性、操作可行性以及发展可行性等方面进行分析。

2.1 系统架构概述

系统架构设计是软件开发过程中必不可少的部分，优秀的系统架构可以使整个系统开发的逻辑变得更加清晰，利于项目的开发、维护和迭代^[3]。本系统采用的系统架构如图 2-1 所示，系统整体部署在 Linux 操作系统上；系统存储层使用 MySQL 作为系统数据库，Redis^[4]作为系统缓存，

Elasticsearch^[5]作为搜索服务器，FastDFS^[6]作为文件存储服务器；系统业务层是通过 Spring Boot^[7]搭建并整合其他框架，其中 Controller 提供业务层与前端界面的交互，Service 提供对数据的处理、封装以及事务等相关业务操作，Dao 提供系统和存储层数据的交互操作；业务层为系统提供内容检索、内容推荐、内容管理、案例管理、用户管理等功能；交互层分为前台和后台，整体使用 Vue.js 开发，前台为本系统用户对象提供的社区服务，后台为本系统社区管理，交互层与业务层通过 RESTful 风格接口实现用户与具体业务的数据交互。

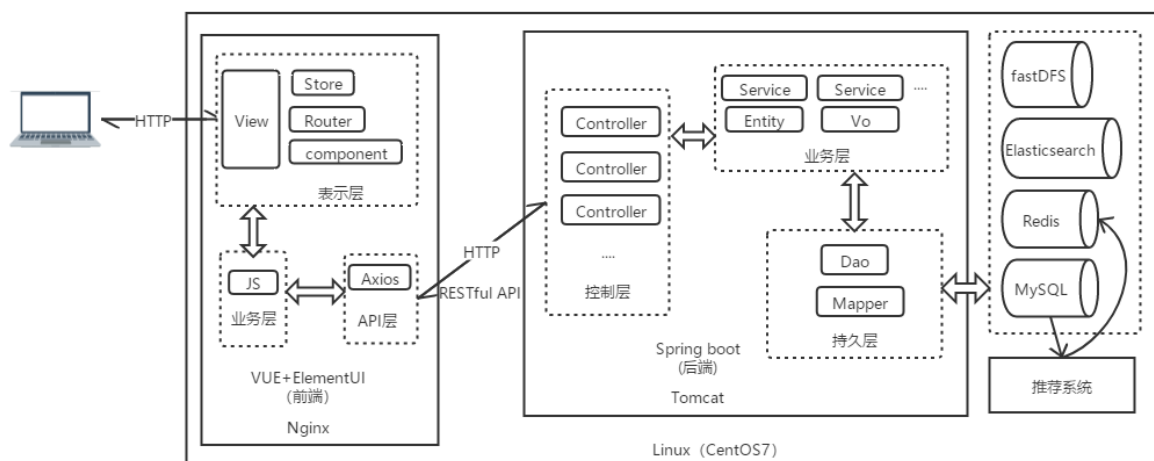


图 2-1 系统架构图

2.2 技术可行性

本系统主要使用 Spring Boot、Vue.js 和 Python 等技术进行开发，这些技术都是目前主流的开发技术且本人对这些技术都有过较为深入学习可以实现相关功能，并且这些主流技术在网上的社区，以及相关生态系统已经非常庞大，遇到问题可以较为容易的找到解决方案。

数据库设计是软件系统开发中必不可少的一环，数据库的选择和明确系统所需要的表，以及表与表之间的逻辑关系是软件项目开发的前提。本系统后台使用 Java 开发，又需要数据持久存储和保证数据不会丢失，系统需要使用关系型数据库。本系统采用 MySQL 数据库作为系统数据库，该数据库也是目前进行系统开发的主流数据库。和 SQL Server 数据库相比，MySQL 数据库对 Java 语言的兼容性非常好，MySQL 所属公司也为其提供非常完善的数据库操作接口实现类；和 Oracle 数据库相比，MySQL 数据库更加的轻量级且免费，虽然在并发操作方面要差于 Oracle 数据库，但是本系统因是社区论坛类系统，并发操作量并不会很大，若在系统以后的发展中出现大量并发操作，也可以通过搭建 MySQL 数据库集群^[8]解决此问题，因此在并发操作上 MySQL 数据库完全能适用。

系统开发模式是软件系统开发前需要明确的另一个重要问题。本系统属于前后端分离式开发，较于传统的 JSP 技术开发，前后端分离式开发已经成为主流，前后端各司其职，通过 AJAX 异步数据接口进行交互。分离式开发较于传统 JSP 开发，因项目前后端部署在不同的服务器，或者同一服务器的不同端口会产生 Session 只能持续一次接口访问的问题，即用户登录行为不能保存。当用户在浏览器登录系统后，需要保存登录信息，但又不需要持久化的存储，一种解决方案是使用浏览器自带的缓存保存用户登录信息；另一种是后端使用缓存保存登录信息，返回给前端 token 令牌，前台请

求数据需要携带令牌验证。显然第二种解决办法更加安全。本系统采用后端缓存，Redis 非关系型数据库在系统缓存方面的应用也非常的广泛，本系统使用 Redis 实现缓存用户登录信息以及个性化推荐信息的功能。对于用户撰写文章中产生的图片以及用户头像等文件资料，不能保存到数据库中，需要保存到服务器的磁盘中存储。本系统采用 FastDFS 实现，FastDFS 是轻量级的分布式文件系统，使用 FastDFS 可以很好的对文件进行管理，实现文件上传、文件下载等操作。本系统使用 Docker^[9] 作为镜像容器，管理 MySQL、Redis 等系统实现所需要的服务器。

因本系统为 Web 应用，所以系统前后端都需要部署到 Web 应用服务器上。后端采用 Java 开发，Tomcat 服务器本身就是用 Java 编写的，对于 Java 的兼容性非常好，因此，系统后端采用 Tomcat 作为后端服务器。但是 Tomcat 服务器对于 HTML 类型的静态界面并不太兼容，因此系统前端采用 Nginx 作为静态页面服务器。

为了实现基于用户访问行为感知的个性化推荐，本系统设计需要解决用户行为感知，以及个性化推荐问题。在本系统中用户行为感知定义为用户对文章操作行为的感知，因此，系统需要感知用户对文章操作的行为，获取用户行为数据。用户在文章方面的行为有撰写文章、浏览文章、评论文章、点赞文章、以及收藏文章，本系统通过将这些行为数据量化，计算用户对他正在操作文章的兴趣程度，即估算用户对该文章的评分。将系统感知的用户行为数据保存，通过合适的推荐算法实现个性化文章推荐。

综上所述，系统开发所使用的相关方法和技术能够满足系统需求，系统开发技术上是可行的。

2.3 操作可行性

本系统操作非常容易，本系统的整体设计类似于已有的 CSDN 等社区，学生可以非常方便的对文章进行浏览、阅读、收藏、评论等一些操作。本系统还提供举报功能，对于违规的文章或者评论，用户可以通过举报通知管理员处理该类信息。在文章编辑方面对于不会使用 Markdown 编辑器的用户，本系统也提供了操作示例，用户可以在很短的时间内学会使用本系统编辑器撰写文章。对于用户的个人信息采集并不涉及到隐私信息，前端获取用户信息是也并不会将用户密码传递到前台，同时本系统为防止数据库被恶意入侵盗取用户资料，系统对用户密码进行加密保存。对于系统后台的操作，也是非常方便，系统后台以顶级管理功能为节点构建功能菜单树，管理员可以很直接的找到自己所需要的系统功能。

2.4 发展可行性

本系统所使用的相关技术如 Spring Boot,Vue.js,MySQL, Redis 等等，都是目前市面上所流行的技术，系统相当长时间都不会面对技术过时的问题。目前信息学院还没有这样一个为学生提供信息交流、知识分享、技能展现的综合社区。因此，本系统的发展潜力和前景也非常大。

3 需求分析

软件开发的过程中最重要的环节就是需求分析，通过对分析用户的需求，根据用户的实际需求开发软件系统，从而解决用户的业务问题。本系统开发，首先需要整理清楚本学院学生的实际需求，将用户实际需求转化为本系统的业务逻辑是本系统开发中决定成败的一环。

3.1 系统功能概述

根据需求，本系统的使用者分为四种不同类型：游客、普通学生用户、普通管理员以及超级管理员。功能模块包含三个大模块，即交流模块、案例模块以及需求模块。学生用户可以通过 PC 端访问系统前台，学生用户在未登录状态时，即游客状态，可以在交流模块浏览文章、搜索文章，可以在案例和需求模块，查看用户提供的案例以及需求。学生用户在登录完成后，在交流模块，系统会为其生成个性化推荐的文章列表，具有对文章收藏、点赞、评论等基础操作，以及写文章的功能；在案例模块和需求模块具有上传案例和需求，以及管理自己的案例和需求的功能；用户可在个人中心修改个人基础信息；超级管理员，在未登录时，是游客身份，具有最基础的查看和搜索文章的功能，在登录系统后，具有后台管理的内容管理、案例管理、系统管理等所有功能；普通管理员，是超级管理员通过系统权限分配的，具有特定管理功能的管理人员，具有系统部分功能管理权限。

该系统的主要工作流程是：在未登录状态时，任何身份进入本系统社区都为游客状态，具备基础的浏览和搜索功能权限，若进行超越权限的操作，系统会提醒用户登录。在已登录状态时，系统会根据当前登录用户的身份查询当前用户的权限，若没有管理权限，即为普通学生用户，可以获取系统前台提供的所有功能的使用，如：写文章、收藏文章、点赞、评论等等。若有管理权限，系统会自动提供进入后台的接口。管理员进入后台后，系统会根据当前管理员的管理权限为其生成管理菜单，管理员通过操作管理菜单实现对本系统的管理。用户在浏览文章的时候，系统会采集用户行为数据，当数据量足够大时，会为其生成内容推荐。

3.2 系统用例图描述

根据对整个系统流程的分析，系统的主要使用者有游客，普通学生用户，普通管理员，超级管理员四种，四种用户拥有不同的权限。下面对整个系统以用例图来表达需求：

(1) 游客：游客状态的用户可以进行登录，查看文章详细信息、搜索文章、查看案例和查看需求的操作。如图 3-1 所示。

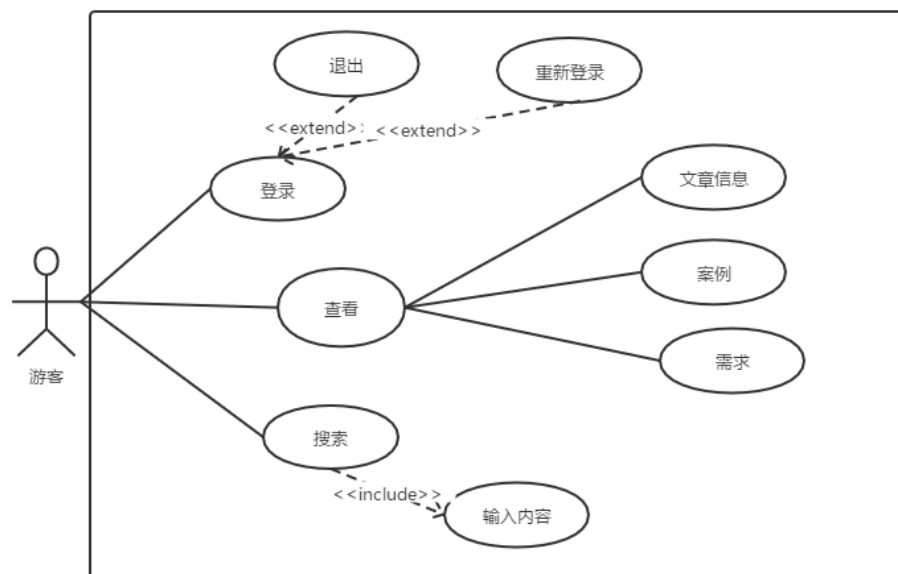


图 3-1 游客模块用例图

对于游客模块用例图的描述：

① 登录：游客可以进行用户登录，当游客登录成功后，游客身份会根据用户实际身份转变成普通用户或管理员。

② 查看：游客身份可以查看系统所有交流模块下的文章、文章下的评论，案例模块下的所有案例以及需求模块下的所有需求。

③ 搜索：游客身份可以点击搜索框输入内容在系统中检索文章。

(2) 普通学生用户：普通学生用户可以进行登陆，查看文章、案例以及需求，上传自己的文章案例以及需求，管理自己的文章案例、需求以及个人资料等操作。如图 3-2 所示。

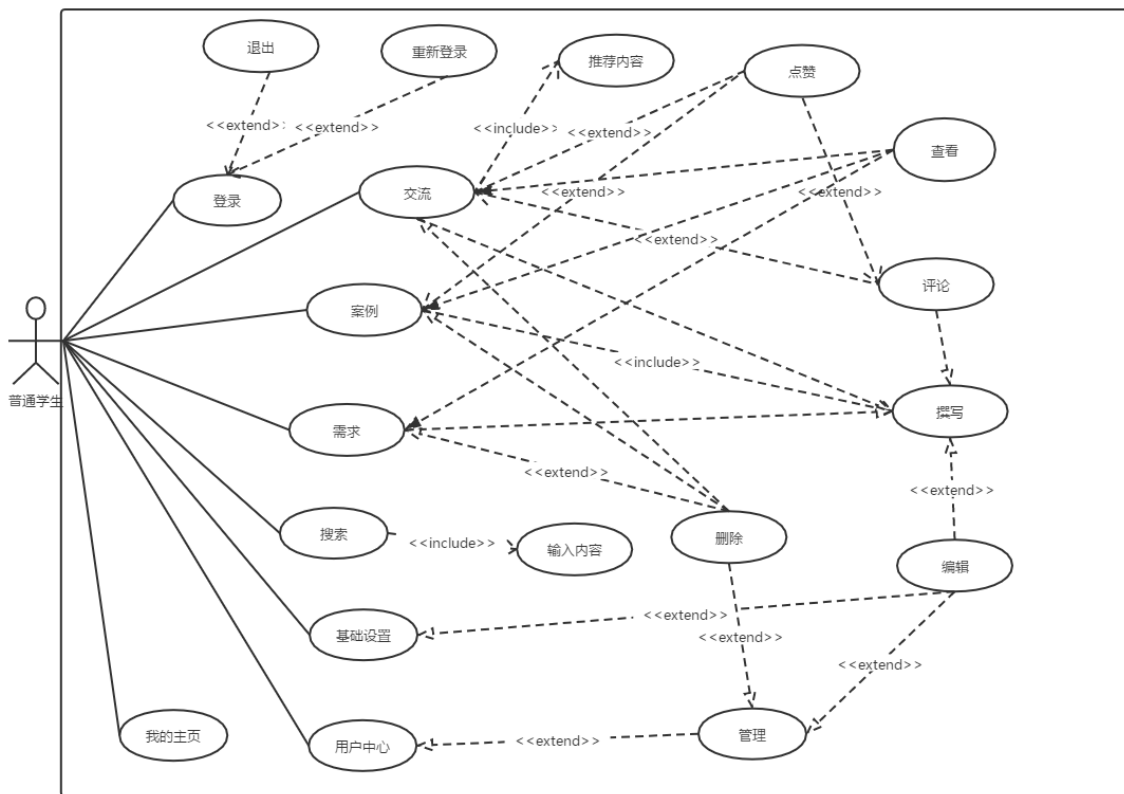


图 3-2 普通用户模块用例图

对普通用户模块用例图的描述：

① 系统登陆：通过用户名，密码，验证码实现用户的登录，当用户执行退出操作后，可进行重新登录。

② 内容搜索：用户可以通过关键字或者包含关键字的语句，检索到包含相关内容的文章列表返回给用户。

③ 基础设置：用户在基础设置中，可以查看个人信息并对个人信息进行编辑修改。

④ 用户中心：用户可在用户中心管理自己的文章以及自己收藏的文章。

⑤ 我的主页：用于展示用户个人信息，包括最近的发布的文章、最近的评论。

⑥ 交流：用户交流是以文章为媒介，用户可以发布自己的文章，可以查看文章，可对文章进行点赞、收藏、评论以及删除，同样可以对文章的评论进行点赞和评论。

⑦ 案例：用户可提交自己的案例，案例提交后，需要等待系统后台管理员审核，在此状态，所属用户可以执行查看或者删除操作。审核通过后，才能能够展示，并其他用户可以查看以及点赞。

⑧ 需求：用户可以提交自己的需求，需求提交后同样需求等待管理员审核，在此状态，所属用户可以执行查看或者删除操作。审核通过后，才能能够展示，其他用户可通过需求上的联系方式，联系需求提供方共同解决需求。

(3) 超级管理员：超级管理员具有系统后台的所有权限，可以进行内容管理、案例管理、需求管理、用户管理以及系统管理等操作。如图 3-3 所示。

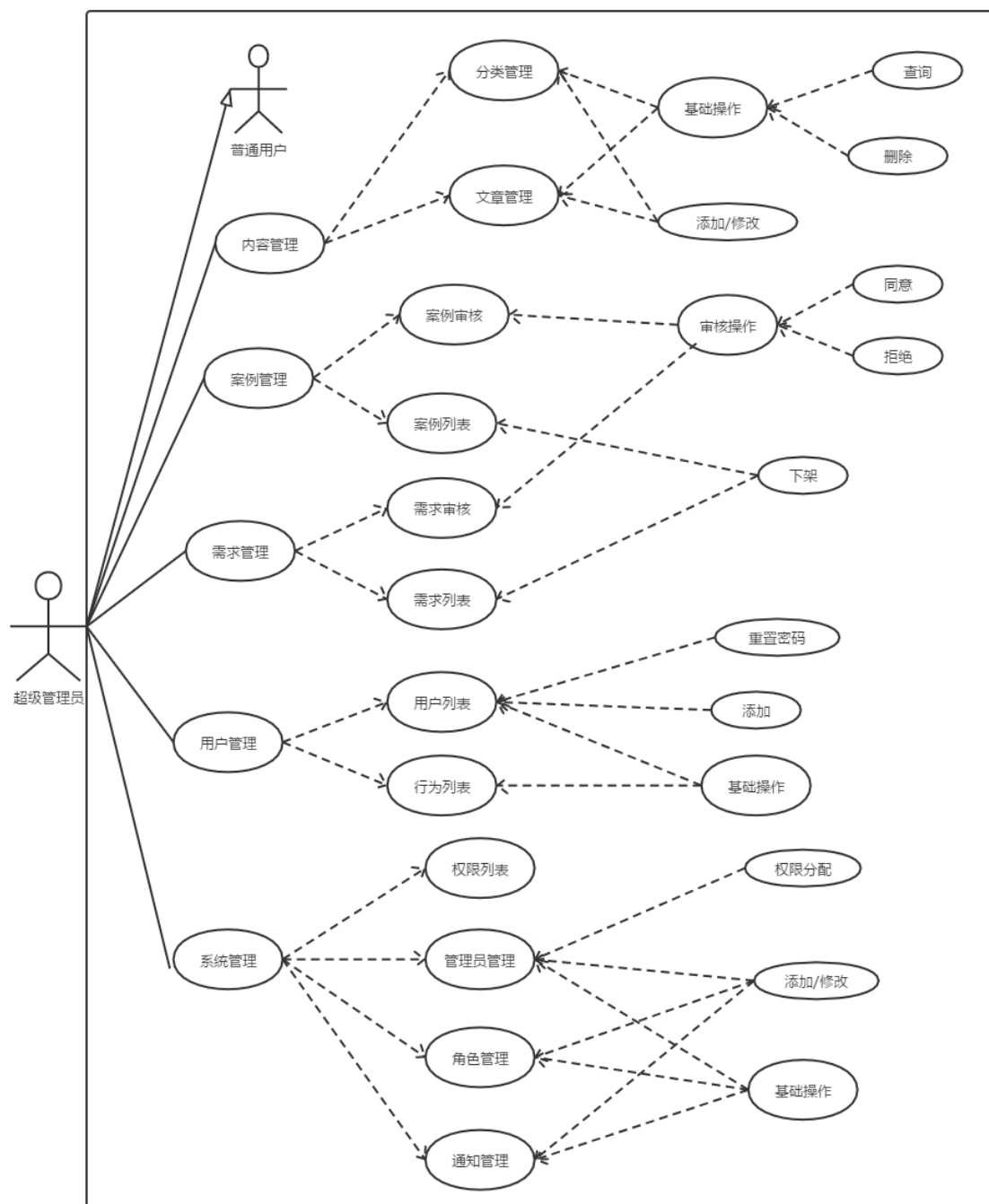


图 3-3 超级管理员模块用例图

对系统管理员模块用例图的描述：

- ① 普通用户：超级管理员继承自普通用户，其具有普通用户的所有权限。
 - ② 内容管理：管理员可通过内容管理对系统内所有文章进行管理，可以对文章分类进行管理。
 - ③ 案例管理：管理员可通过案例管理对普通用户提交的案例进行审核，以及对通过审核的案例进行管理。
 - ④ 需求管理：管理员可通过需求管理，对普通用户提交的需求进行审核，以及对通过审核的需求进行管理。
 - ⑤ 用户管理：管理员通过可以对用户进行管理，并可以查看用户行为数据。
 - ⑥ 系统管理：管理员可以管理系统通知信息，系统权限、角色以及系统管理员。
- (4) 普通管理员：普通管理员的具体权限，是通过超级管理员分配的角色控制，超级管理员通过创建角色，并为角色分配权限菜单，从而使得普通管理员具备超级管理员的某些特定功能。

3.3 系统设计规则

(1) 前端设计：社区前台和系统后台，都采用组件化结构设计，将通用的结构进行抽取为单独的组件，系统组件间通过依赖引入或者路由跳转进行管理，形成系统组件树，构建单页面应用系统。前台设计延续主流的社区论坛设计风格，没有广告，界面更加简洁，美观，操作更加容易，对于操作有些难度的文章撰写部分，本系统也提供了操作说明，力求保证初次使用用户没有抗拒感；系统后台也是用传统的后台布局模式，以系统管理功能为基础，在系统后台左侧生成功能菜单树形结构，方便管理人员查找并正确使用功能。

(2) 后端设计：后端 API 接口采用 RESTful 设计风格，通过 POST、GET、DELETE、PUT 请求限定，更加明确每个数据接口的含义。后端整体采用 MVC(Model View Controller)设计模式，Controller 为前后台交互的入口，Service 层采用接口设计模式，在 Service 对数据进行具体操作，为 Controller 层提供服务，Mapper 层为系统后台与数据库的交互层。将各个层级的代码分开编写，各个层级只管负责本层级功能，同层级之间尽量做到高内聚低耦合，从而提高代码的可维护性。严格遵循语义化命名，类名采用大驼峰命名规则，即所有单词首字母大写，方法及变量名采用小驼峰命名规则，即第一个单词首字母小写，其它单词首字母大写，并编写详细注释提高代码的可读性。

4 系统设计

4.1 系统结构设计

系统结构设计是对项目需求分析的结果进行整理，划分为不同的模块以及不同的功能。优秀的系统结构设计可以提高项目开发速度、减少项目开发成本。本系统在系统结构上分为系统前台结构和系统后台结构。

4.1.1 系统前台结构设计

系统前台结构包含交流模块、案例模块、需求模块以及基础模块，如图 4-1 所示。在交流模块模块下又分为文章列表、文章推荐、文章搜索、文章撰写、文章操作(如：文章点赞、收藏等操作)、周用户评论榜以及周热议文章榜；在案例模块下又分为案例列表、提交案例、我的案例以及案例操作；在需求模块又分为需求列表、需求提交以及我的需求；在基础模块下右包含基础设置、用户中心、

我的主页、我的消息、用户登录以及系统后台入口。

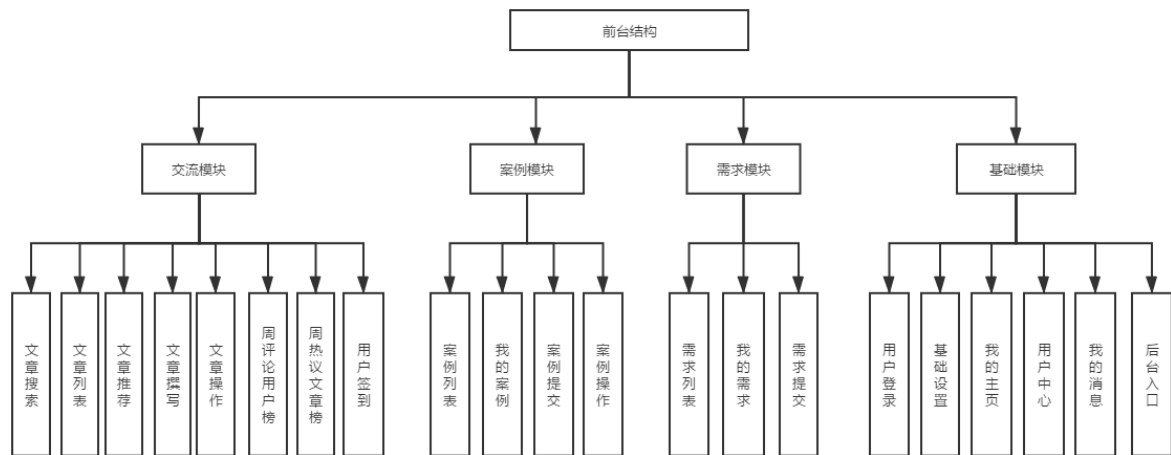


图 4-1 系统前台结构图

4.1.2 系统后台结构设计

系统后台由管理员使用，主要实现系统管理的功能，因此系统后台结构设计采用管理功能划分，通过前台结构可得，系统后台包含内容管理、案例管理、需求管理、用户管理以及系统管理，如图 4-2 所示。内容管理中又包含文章分类管理以及文章管理；案例管理中又包含案例申请列表和案例管理列表；需求管理中也包含需求申请列表和需求管理列表；系统管理中包含权限管理、管理员管理、角色管理以及通知管理。

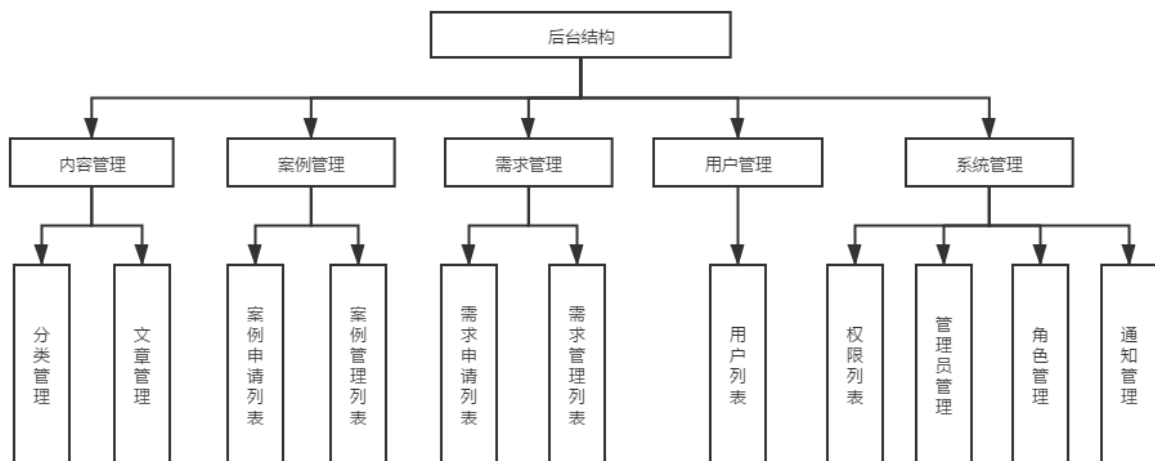


图 4-2 系统后台结构图

4.2 数据库逻辑设计

通过整理本系统的业务需求可知，本系统是以用户为主体，用户之间的联系体现在文章、案例以及需求上，本系统需要用户表、文章表、案例表以及需求表；用户浏览文章产生了用户行为数据以及可能的点赞、评论、举报、收藏以及收藏所需要的收藏文件夹，本系统在该模块还需要用户行为数据表、文章点赞表、文章收藏表、收藏文件夹表、评论表、评论点赞表以及举报表，用户在添加

2	name	varchar	20	用户真实姓名	非空
3	nickname	varchar	20	用户昵称	非空
4	account	varchar	10	账号	非空
5	password	varchar	36	密码	非空
7	class_grade	varchar	30	班级	非空
8	phone	int	11	电话	非空
9	email	varchar	30	邮箱	空
10	avatar	varchar	100	头像	非空
11	gender	int	1	性别(1: 男, 0: 女)	非空
12	signature	varchar	100	签名	空
13	learn_coin	int	4	学币	非空
14	city	varchar	30	城市	空
15	create_time	datetime		创建时间	非空
16	state	int	1	邮箱激活态(1: 激活, 0: 未激活)	非空
17	vip	int	1	VIP 级别	非空
18	role_id	int	3	外键、角色 ID	空

表 4-2 文章表(ic_article)

序号	字段	数据类型	长度	字段说明	备注
1	art_id	int	11	主键 ID	非空
2	u_id	int	11	外键、户 ID	非空
3	title	varchar	50	文章标题	非空
4	content	longtext		文章内容	非空
5	create_time	datetime		创建时间	非空
6	c_id	int	2	外键、类 ID	非空
7	tag	varchar	20	文章标签	非空
8	is_att	int	1	是否有附件	非空
9	learn_coin	int	4	悬赏学币	非空
10	is_over	int	1	是否完结	非空
11	status	int	1	状态 1 精华	非空
12	top	int	1	是否置顶	非空
13	t_expiry_time	datetime		置顶失效时间	空
14	hits_zan	int	4	点赞数	非空
15	hits_comment	int	4	评论数	非空
16	watch	int	4	阅览人数	非空

表 4-3 评论表 ic_article_comment)

序号	字段	类型	长度	字段说明	备注
----	----	----	----	------	----

1	art_c_id	int	11	主键 ID	非空
2	u_id	int	11	外键、用户 ID	非空
3	art_id	int	11	外键、文章 ID	非空
4	p_id	int	11	外键、父评论 ID	空
5	create_time	datetime		创建时间	非空
6	content	text		评论内容	非空
7	is_take	int	1	是否采纳	非空
8	hits_zan	int	4	评论点赞数	空

表 4-4 文章分类表(ic_category)

序号	字段	数据类型	长度	字段说明	备注
1	c_id	int	11	主键 ID	非空
2	name	varchar	10	分类名称	非空
3	pinyin	varchar	20	分类名称的拼音	非空
4	available	int	1	是否可用	非空
5	state	int	1	普通用户是否可用	非空

表 4-5 文章点赞表(ic_article_zan)

序号	字段	数据类型	长度	字段说明	备注
1	art_hz_id	int	11	主键 ID	非空
2	art_id	int	11	外键、文章 ID	非空
3	u_id	int	11	外键、用户 ID	非空
4	create_time	datetime		点赞时间	非空

表 4-6 文章阅览表(ic_article_watch)

序号	字段	数据类型	长度	字段说明	备注
1	art_w_id	int	11	主键 ID	非空
2	art_id	int	11	外键、文章 ID	非空
3	u_id	int	11	外键、用户 ID	非空
4	create_time	datetime		阅览时间	非空

表 4-7 评论点赞表(ic_article_comment_zan)

序号	字段	数据类型	长度	字段说明	备注
1	art_c_hz_id	int	11	主键 ID	非空
2	art_c_id	int	11	外键、评论 ID	非空
3	u_id	int	11	外键、用户 ID	非空
4	create_time	datetime		点赞时间	非空

表 4-8 收藏文件夹表(ic_favorite_folder)

序号	字段	数据类型	长度	字段说明	备注
1	d_id	int	11	主键 ID	非空
2	dirname	varchar	20	文件夹名称	非空
3	u_id	int	11	外键、用户 ID	非空
4	create_time	datetime		创建时间	非空
5	state	int	1	是否可以被修改	非空

表 4-9 文章收藏表(ic_article_favorite)

序号	字段	数据类型	长度	字段说明	备注
1	f_id	int	11	主键 ID	非空
2	dir_id	int	11	外键、文件夹 ID	非空
3	u_id	int	11	外键、用户 ID	非空
4	art_id	int	11	外键、文章 ID	非空
5	create_time	datetime		创建时间	非空

表 4-10 文章附件表(ic_attachment)

序号	字段	数据类型	长度	字段说明	备注
1	att_id	int	11	主键 ID	非空
2	art_id	int	11	外键、文章 ID	非空
3	url	varchar	100	附件地址	非空

表 4-11 用户行为表 ic_access_behavior)

序号	字段	类型	长度	字段说明	备注
1	id	bigint	20	主键 ID	非空
2	u_id	in	11	外键、用户 ID	非空
3	art_id	int	11	外键、文章 ID	非空
4	rating	int	2	评分	非空
5	create_time	datetime		创建时间	非空

表 4-12 案例表(ic_case)

序号	字段	数据类型	长度	字段说明	备注
1	id	int	11	主键 ID	非空
2	u_id	int	11	外键、用户 ID	非空
3	title	varchar	20	案例标题	非空
4	introduce	varchar	100	案例介绍	非空
5	case_img	varchar	100	案例封面地址	非空

6	url	varchar	100	案例链接地址	非空
7	create_time	datetime		创建时间	非空
8	zan	int	4	点赞人数	非空
9	sysu_id	int	11	外键、审核管理员 ID	空
10	state	int	1	审核状态	非空

表 4-13 案例点赞表(ic_case_zan)

序号	字段	数据类型	长度	字段说明	备注
1	id	int	11	主键 ID	非空
2	c_id	int	11	外键、案例 ID	非空
3	u_id	int	11	外键、用户 ID	非空
4	create_time	datetime		点赞时间	非空

表 4-14 需求表(ic_exchange)

序号	字段	数据类型	长度	字段说明	备注
1	id	int	11	主键 ID	非空
2	u_id	int	11	外键、用户 ID	非空
3	name	varchar	20	发布人	非空
4	title	varchar	50	需求标题	非空
5	claim	varchar	200	具体要求	非空
6	qq	varchar	12	联系 QQ	非空
7	create_time	datetime		创建时间	非空
8	expiry_time	datetime		失效时间	空
9	sysu_id	int	11	外键、审核管理员 ID	空
10	state	int	1	审核状态	非空

表 4-15 通知表(ic_news)

序号	字段	数据类型	长度	字段说明	备注
1	news_id	int	11	主键 ID	非空
2	sysu_id	int	11	外键、管理员 ID	非空
3	title	varchar	11	标题	非空
4	content	varchar	100	具体内容	非空
5	create_time	datetime		创建时间	非空
6	expiry_time	datetime		失效时间	非空
7	level	int	1	重要程度	非空

表 4-16 消息表(ic_message)

序号	字段	数据类型	长度	字段说明	备注
----	----	------	----	------	----

1	msg_id	int	11	主键 ID	非空
2	send_id	int	11	外键、发送用户 ID	非空
3	recv_id	int	11	外键、接收用户 ID	非空
4	content	varchar	50	消息内容	非空
5	create_time	datetime		创建时间	非空

表 4-17 举报表(ic_report)

序号	字段	数据类型	长度	字段说明	备注
1	id	int	11	主键 ID	非空
2	type	int	1	举报类型	非空
3	tid	int	11	举报类对应表的外键 ID	非空
4	content	varchar	50	举报内容	非空
5	create_time	datetime		创建时间	非空

表 4-18 签到表(ic_signIn)

序号	字段	数据类型	长度	字段说明	备注
1	s_id	int	11	主键 ID	非空
2	u_id	int	11	外键、用户 ID	非空
3	create_time	datetime		创建时间	非空
4	days	int	4	连续签到天数	非空

表 4-19 权限表(ic_rights)

序号	字段	数据类型	长度	字段说明	备注
1	id	int	3	主键 ID	非空
2	auth_name	varchar	20	权限名称	非空
3	pid	int	3	父权限 ID	非空
4	path	varchar	50	权限路径	非空
5	available	int	1	是否可用	非空
6	level	int	1	权限级别	非空
7	icon	varchar	20	权限图标	非空

表 4-20 角色表(ic_role)

序号	字段	数据类型	长度	字段说明	备注
1	id	int	11	主键 ID	非空
2	role_name	varchar	20	角色名称	非空
	role_desc	varchar	50	角色描述	非空
3	create_time	datetime		创建时间	非空
4	available	int	1	是否可用	非空

表 4-21 角色权限表(ic_rights_role)

序号	字段	数据类型	长度	字段说明	备注
1	role_id	int	11	角色 ID	非空
2	right_id	int	11	权限 ID	非空

5 系统实现

本系统的前台和后台都是使用 Vue.js+ElementUI 构建，用户通过页面操作，系统前端通过 Axios 异步的 RESTful 风格的数据接口与系统后端进行数据交互。后端在业务层处理业务逻辑，在数据层和数据库进行数据交互。

5.1 系统前台实现

5.1.1 交流模块

前台交流模块主要用于查看、搜索、添加用户文章等操作，该模块包含了本系统的大部分功能，亦是本系统的主体模块。

(1) 交流首页：将项目部署到 Web 服务器上之后，在浏览器中访问系统首页。系统首先从数据库中获取文章分类列表，前台渲染文章分类数据，系统根据默认获取推荐的条件到后端请求文章推荐数据以及置顶文章数据加载到页面渲染显示。当前台请求推荐数据时，系统后端首先判断当前请求数据的用户是否登录，系统会先从请求中获取 token 值，若请求头中没有 token 值，则用户未登录；若有 token 值，则在 Redis 缓存中获取当前 token 对应的用户，若无用户，则判定 token 为虚假值；若有用户，则判定已登录。若当前用户未登录，则推荐点赞数最多的 N 条文章给用户；若用户已登录，则从 Redis 中获取当前用户的推荐列表，若有推荐数据，则根据推荐数据到数据库中获取具体文章列表返回；若无推荐数据，则查询数据库中用户行为数据，获取当前用户浏览过的所有文章的标签，通过标签到 Elasticsearch 中查询符合的数据，若数据不足规定的浏览条数，则补足热议的且不包含在查询数据中的文章，根据数据到数据库中获取具体的文章列表返回；若当前用户没有行为数据，则推荐点赞数最多的且当前用户未浏览的 N 条文章给用户。交流模块首页如图 5-1 所示，首先右侧，开始为搜索框和发布文章按钮，接着为系统通知栏，然后是用户签到区，回帖周榜以及本周热议文章，右侧最下方为发布文章快捷按钮和处于隐藏状态的回到顶部按钮。用户通过点击顶部的文章分类列表可以查看不同分类下的文章，也可通过点击当前文章分类下的子分类，如：未结、已结、按最新、热议等操作对当前分类下的文章进行筛选。

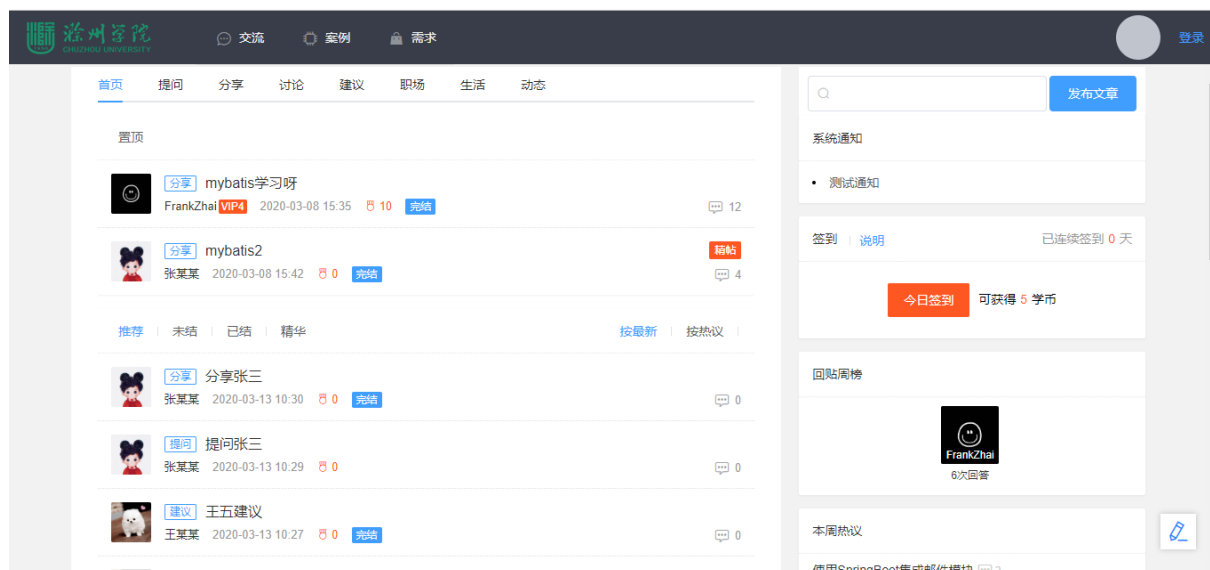


图 5-1 系统前台首页

(2) 搜索：在首页右侧搜索框中输入要查询的文章标题，键入回车，前台首先判断用户是否输入具体查询内容，若未输入查询内容，则提示用户且不执行此次查询操作。若正确输入查询内容，便将输入的内容发送到后端，后端业务层根据用户输入的内容到 Elasticsearch 服务器中，查询符合相关词条的文章返回相关文章数据，然后对数据进行封装返回给用户。若数据返回为空，页面则显示未查询到文章提示用户。若数据不为空，则前台渲染数据，展示用户查询的结果。搜索结果如图 5-2 所示。



图 5-2 搜索结果

(3) 系统通知：在首页加载的同时会获取系统所有未到失效时间的系统通知。系统后端通过前端请求，在数据层判断失效时间是否大于当前时间，若大于，则获取该条通知，反之，则过滤该条通知。然后根据通知紧急程度以及创建时间排序，将紧急程度高，创建时间最近的通知排在前列返回给前端。前端获取数据，渲染显示在前台首页右侧，用户通过点击通知标题，便可以查看通知详情，如图 5-3 所示。

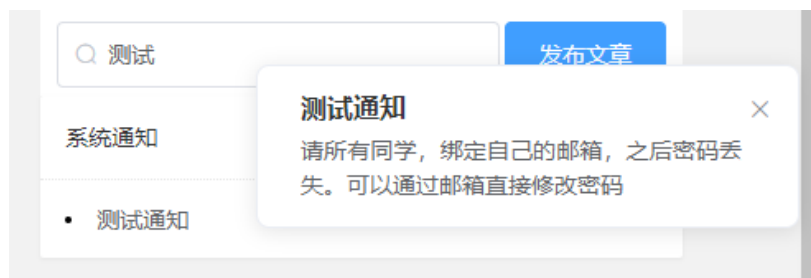


图 5-3 通知详情图界面

(4) 签到：用户通过点击说明标签，便可以查看签到规则，用户点击今日签到按钮，系统首先判断用户是否登录，若该用户未登录，则跳转到登录页面。若已登录，系统前台向后端请求用户签到信息，判断用户连续签到天数，根据下表 5-1 签到规则表，判断当天用户签到应获取的学币数，显示在签到按钮右侧提示用户。用户点击签到按钮便可以完成签到，系统将用户签到数据发送给后端，后端在业务层处理用户签到数据，让连续签到天数加一，然后传送到数据层保存到系统数据库中。签到结果如图 5-4 所示。

表 5-1 系统签到规则表

连续签到天数	可获取学币数
<5	5
≥5	10
≥15	15
≥30	20
≥100	30
≥365	50



图 5-4 通签到结果界面

(5) 回帖周榜：系统在加载前台首页的同时，会向后端请求回帖周榜数据，后端接收到数据请求，将请求条件传到业务层做数据处理，再通过数据层向数据库中获取回帖数多的且数据产生时间为本周的用户数据，将查询到的用户数据进行封装处理，通过控制层返回给前台，前台渲染返回的数据在首页左侧生成回帖周榜排名界面。

(6) 本周热议：系统在加载前台首页的同时，会向后端请求本周热议数据。后端接收到数据请求，将请求条件数据传到业务层做数据处理，再通过数据层向数据库中，根据文章评论数排序将评论数多的，且文章发布于本周的文章数据获取，在业务层进行封装，并返回给前台。前台渲染返回的数

据在首页左侧生成本周热议列表展示界面。

(7) 写文章：用户通过点击首页的发布文章按钮，或者其他模块中的快捷按钮进入撰写文章页面。系统首先判断当前用户是否登录，若未登录，则跳转到登录页；若已登录，则正常跳转到写文章页。用户可使用 Markdown 语法规则撰写文章，如图 5-5 所示对于不知道规则的用户，可以点击左侧的帮助按钮即可获取语法提示，同时，可以再左侧边写边预览文章，如图 5-6 所示。当用户写完文章点击该界面顶部功能选项区域的保存按钮，或者键入 Ctrl+S 快捷键。当文章内容为空时，系统会提示用户；当内容不为空时，系统会弹出输入框，需要在此完善文章标题、文章标签、文章分类以及悬赏学币等操作，点击保存，便可将文章数据传送到后端进行保存。后端获取文章数据在业务层对数据进行完善，首先传输到存储层进行保存，并添加用户行为数据，再将文章的标题、标签以及文章在数据库中的主键 ID，保存到 Elasticsearch 服务器中，完成所有业务操作后，后端通过控制层返回保存结果给前台。前台获取保存结果，若失败则提示用户；若成功，则跳转到用户中心展示用户所有文章列表。文章保存如图 5-7 所示。

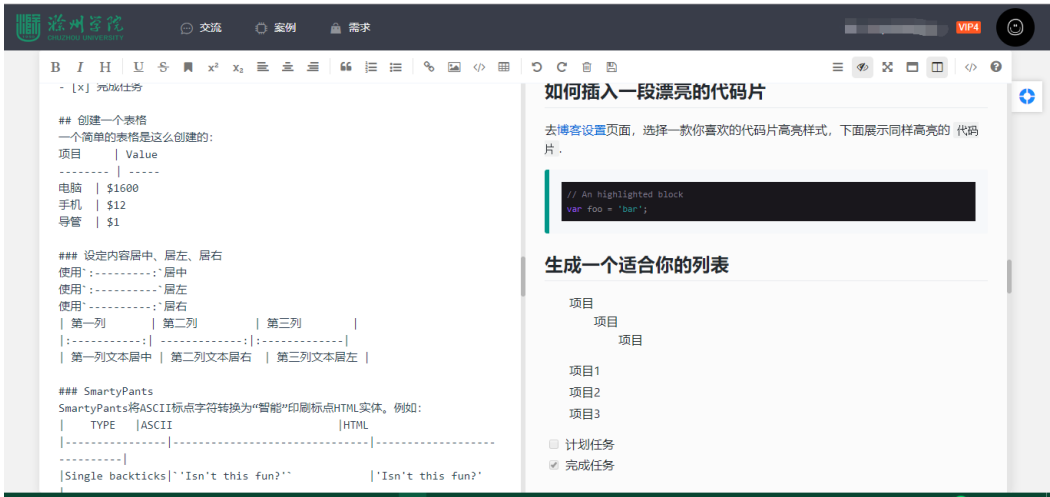


图 5-5 撰写文章界面

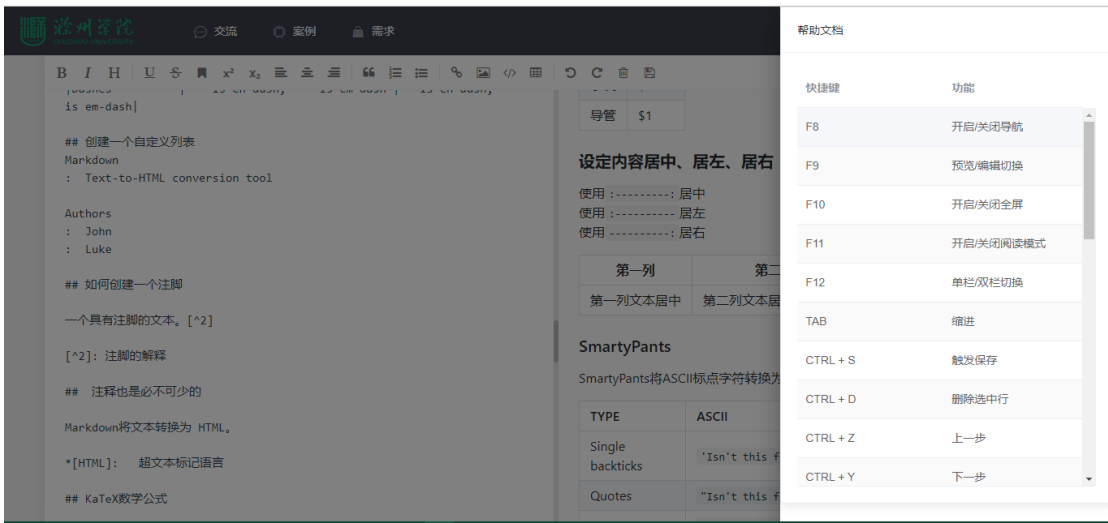


图 5-6 文档帮助界面



图 5-7 上传文章界面

(8) 浏览文章：用户在首页文章列表点击文章标题，便可跳转到文章详情页，当用户未登录时，只能浏览文章，当用户点击点赞等操作时系统会自动跳转到登录页；当用户已登录时便可对文章进行点赞、评论、收藏以及举报操作。当浏览的文章属于当前用户时，用户还能进行编辑和删除的操作。当用户为超级管理员或者具有文章管理权限的普通管理员，可以删除当前文章或者文章下的评论，删除前，管理员必须填写删除理由，系统会将管理员删除行为，以及删除理由以系统消息的形式，通知文章所属用户或者评论所属用户。系统后端业务层在删除或者修改文章的同时，也会根据文章在数据库中的主键 ID，删除或修改 Elasticsearch 中对应的文章信息。当用户退出当前文章时，系统会记录用户的行为数据，具体体现在系统计算用户浏览文章的时间长度、以及点赞、评论、收藏的操作，为该用户对当前文章预估一个评分。系统前台将感知的用户行为数据传送给后端，保存到系统数据库中。文章浏览页面如图 5-8 所示。



图 5-8 文章浏览界面

(9) 文章收藏：当文章不属于当前用户时，便可以在文章详情页的文章标题右侧，显示文章收藏

按钮。当用户点击收藏按钮时，系统会向后端发送请求，获取当前用户的收藏文件夹，系统后端业务层首先判断是否有收藏文件夹。当用户没有收藏文件夹时，系统便会创建一个默认文件夹，封装数据，设置此文件夹为不可修改状态，保存到数据库中并返回给前台显示；当用户有收藏文件夹时，便会返回属于该用户的收藏文件夹列表。前台渲染数据显示给用户，用户可在该界面右下角点击新建文件夹按钮，创建新的收藏文件夹，系统后端业务层在保存用户收藏文件夹时，会设置文件夹为可修改状态。当用户点击对应文件夹右侧的收藏标签时，前台便会发送保存数据到后端，后端在业务层处理用户收藏数据后，将数据通过数据层保存到数据库中，并将保存结果返回给用户显示，将收藏的星号按钮设置为选中的状态。添加收藏如图 5-9 所示。



图 5-9 文章收藏界面

(10) 文章评论：用户通过点击顶部的评论标签，便会弹出文章评论的富文本编辑器，点击文章详情页底部的评论中的评论标签，便会弹出评论评论的富文本编辑器。用户可在编辑器中输入评论内容或者上传图片，点击确定，前台首先判断该条评论是评论文章还是评论评论，若评论的对象为评论，前台还会封装该条评论父评论信息，然后提交评论数据到系统后端。后端接收到数据，在业务层对数据进行处理，然后保存到数据库中，将保存结果返回给前台展示给用户，若保存成功，则关闭评论编辑器，并重新获取评论信息；若保存失败，则提示用户，并显示错误信息。评论编辑如图 5-10 所示当用户将文章详情滑倒底部的时候，便可看见文章的所有评论。所有对文章的评论默认是显示状态，对评论的评论默认是隐藏状态，当用户点击对应文章评论的查看全部回复标签时，便可查看所有对该评论的评论。用户可对评论进行点赞，或者评论，当用户鼠标放置在具体的评论上，还会显示举报标签。文章评论详情如图 5-11 所示。

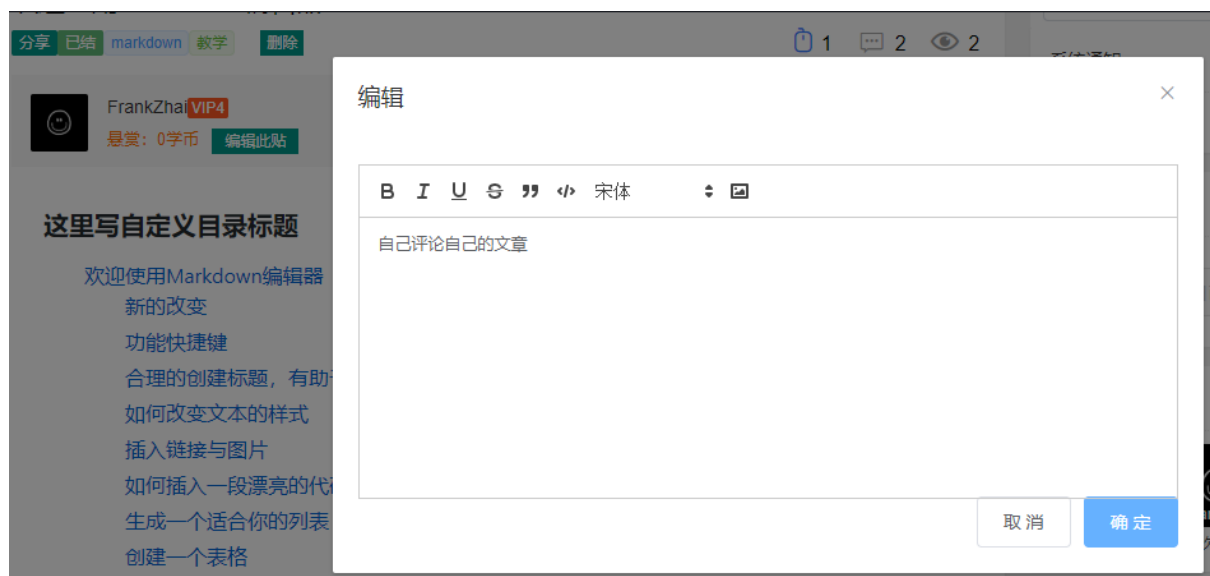


图 5-10 评论编辑器界面



图 5-11 文章评论界面

5.1.2 案例模块

案例模块是用于用户展示自己的成果，交流模块中，用户只能通过文章的形式展现自己。在当前模块，用户可以添加自己的项目地址，上传项目封面封面和项目介绍展示给所有用户。项目地址可以为你在 GitHub 或者 Gitee 上的项目地址，也可以为部署在可远程访问的服务器上的项目地址。

(1) 案例首页：进入系统前台后，点击顶部的案例标签，便可进入系统案例模块。首页默认展示本年度，按提交时间降序排序的所有案例。当显示案例数超过 12 条时，则分页显示。用户通过点击底部的下一页、上一页或者具体页码进行翻页。用户可选择查看本年度、获取上一年度的案例数据。在当前年度下，用户又可通过按时间或者按点赞进行排序。案例模块首页具体展示界面如下图 5-11 所示。用户可以对案例进行点赞，也可以通过点击案例封面根据案例地址具体跳转到案例具体位置。



图 5-12 案例首页界面

(2) 案例提交：当用户点击页面上的案例提交按钮，当用户未登录时，会自动跳转到系统登录界面；当用户已登录，则系统会弹出案例添加界面。用户需要输入项目名称、项目介绍、项目地址以及上传项目封面图片，点击确定，系统便会将数据发送到后端；后端获取数据，在业务层对数据进行处理，设置状态为待审核，通过存储层将数据保存到数据库中，然后将保存结果返回给前台，前台将结果展示给用户。用户项目案例添加界面如图 5-13 所示。



图 5-13 案例添加界面

(3) 我的案例：用户可在案例模块通过点击我的案例进入，当用户未登录时，会自动跳转到系统登录界面；当用户已登录，会在下方显示用户所有已提交的案例，包括未审核和已下架，用户可在我的案例下查看案例状态或者修改、删除已提交的案例。我的案例界面如图 5-14 所示。



图 5-14 案例添加界面

5.1.3 需求模块

需求模块是给用户提交需求使用，用户在遇到自己不可解决的问题时，且该问题在交流模块的提问分类中不能解决，如：完成一个项目或者项目中的某些模块，可在该模块下添加自己的需求以及联系方式，其他用户若有能够解决可以通过联系方式联系需求提交方。

(1) 需求首页：用户在进入系统前台后，通过点击顶部的需求标签即可进入需求首页，系统向后端获取需求数据加载到前台显示给用户。用户可查看具体需求，若能够解决可以通过 QQ 的方式联系需求提交方。需求首页界面如图 5-15 所示。



图 5-15 需求模块首页

(2) 提交需求：当用户点击提交需求按钮时，系统先从 Vuex 中获取登录用户对象，若该用户对象为空，则用户未登录，跳转到用户登录界面；若已登录，则弹出需求添加界面。用户可在需求添加界面输入需求标题、具体要求以及自己的 QQ 联系方式，点击确定前台会将数据传送到后端，业

务层对数据进行处理，添加状态为待审核，将数据存储到数据库中，返回给用户存储结果。需求添加界面如图 5-16 所示。

图 5-16 需求添加界面

(3) 我的需求：用户通过点击我的需求按钮进入我的需求界面，系统前台首先向后端请求用户个人需求信息，后端从数据库中获取用户个人需求信息，按时间正序排序返回给用户。若无数据则显示暂无用户个人需求提示用户，若有数据则前台渲染显示如下图 5-17 所示。在此界面，用户可以对自已添加的需求进行查看，编辑以及删除操作。

图 5-17 我的需求界面

5.1.4 基础模块

本系统基础模块包括用户登录，以及用户登录后的个人信息管理，用户中心、我的主页、我的消息以及退出，当登录的用户具有管理权限时，会显示进入后台的入口。当用户登录系统后将鼠标移动置头像或者用户名处，便可显示基础模块中的功能。

(1) 用户登录：用户进入本系统后可点击系统导航栏右上角的登录按钮，或者使用超越游客的权限会自动进入登陆界面。用户在登录页面输入用户名密码以及验证码点击登录按钮进行登录，前台会将用户登录数据发送给后端，后端从 Redis 缓存中获取验证码元数据对比用户输入的验证码，若

不正确，则直接返回错误信息提示用户；若验证码正确，则将用户登录账号和密码传送给业务层，业务层数据进行处理，将用户密码进行 MD5 加密，再将账号和加密后的密码作为查询条件在数据库中搜索。若有该用户，则生成 token 值，以 token 值为键以用户 ID 为值，将用户存储到 Redis 中，并返回登录成功信息以及 token 值给前台。前台展示用户登录成功信息，并将 token 数据以及用户数据保存到浏览器缓存以及 Vuex 中做权限操作验证；若没有查询到，则返回用户名或者密码错。用户登录界面如图 5-18 所示。

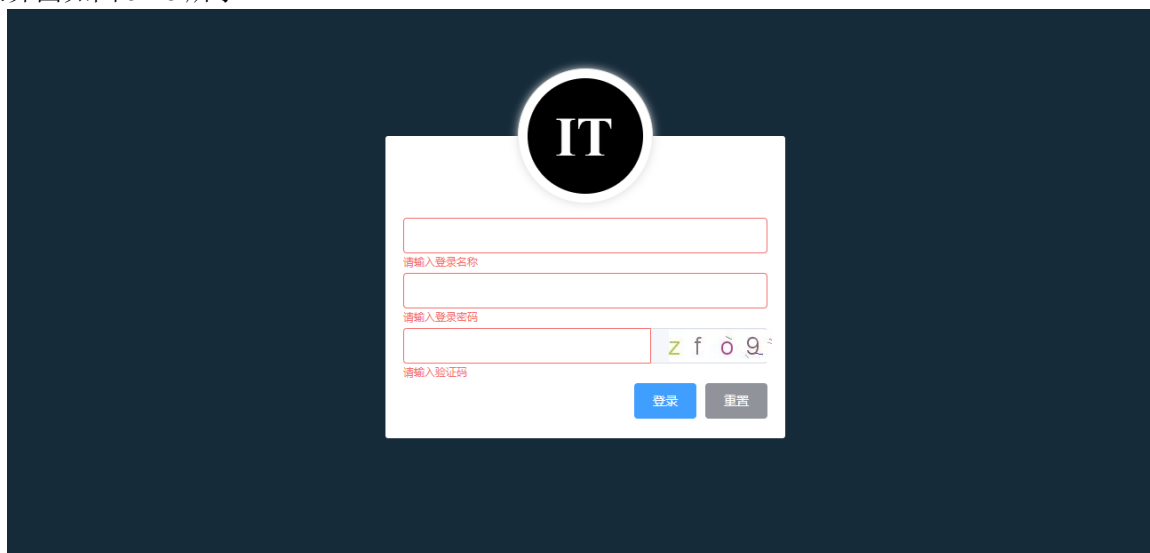


图 5-18 用户登录界面

(2) 基础设置：当用户点击基础设置按钮便可进入基础设置界面，用户可在此见面完善或修改自己的个人信息，包括基础资料、头像以及密码。在用户进行密码修改时，需要先输入原始密码，以防止恶意修改操作，用户输入原始密码、新密码以及确认密码数据，若确认密码和新密码不相等，则提示用户，且不执行修改操作；若相等，再将用户输入的原始密码和新密码发送到系统后端。后端接收到数据，在业务层先判断用户原始密码是否正确，若不正确，则后端不执行修改操作，并将错误信息返回；若正常，则将新密码进行加密，更新用户数据。用户基础设置如下图 5-19 所示。



图 5-19 基础信息界面

(3) 我的主页：用户点击我的主页，便可查看我的主页信息，我的主页是用于展示给其他用户浏

览的。当其他用户在首页文章列表点击用户昵称或者用户头像，或者回帖周榜中的头像，或者文章详情中的用户头像和昵称都可进入用户主页，用户主页展示用户的基本信息，最近发布的文章以及最近的评论回答等信息。用户主页如图 5-20 所示。



图 5-20 用户主页界面

(4) 用户中心：用户点击用户中心按钮便可进入用户中心界面，用户可在用户中心管理自己的文章以及收藏的其他文章。系统前台首先发送请求，获取当前用户的所有文章数据、以及收藏文章数据。系统后端在业务层对用户个人文章数据封装返回；对用户收藏文章，首先获取当前用户的收藏文件夹，再根据文件夹获取文件夹下所属的收藏文章，再将数据封装返回。前台获取到数据，首先展示用户文章数量以及收藏文章数量，再渲染用户个人文章列表，用户收藏文件夹列表以及每个文件夹下的收藏文章列表。用户中心界面如图 5-20 所示。

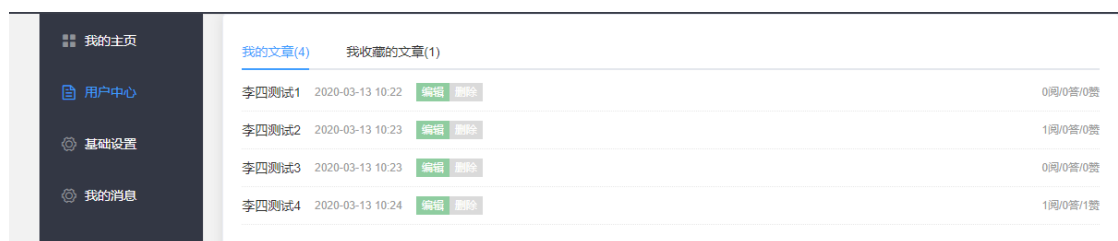


图 5-20 用户中心界面

(5) 我的消息：用户点击我的消息按钮便可进入我的消息界面，该页面会获取所有发送给当前用户的消息，消息是由系统产生的，如：管理员通过了当前用户的案例申请，其他用户点赞了该用户的文章，系统在业务层处理这些数据时，便会自动生成系统消息，发送给当前用户。如图 5-21 所示，用户通过点击具体消息右下角的删除按钮删除对应消息，也可以通过点击顶部的清空全部消息，删除所有属于当前用户的消息。



图 5-21 我的消息界面

5.2 系统后台实现

当用户登录系统后，若用户拥有后台管理权限，便可通过点击前台的基础模块中的后台管理按钮进入系统后台。后台首页如图 5-22 所示。



图 5-22 后台首页界面

5.2.1 内容管理

(1) 分类管理：用户点击内容管理下的分类管理按钮便可进入分类管理界面，在当前界面中，管理员可对文章分类进行增、删、改的操作。通过修改普通用户是否可见，可以区分，哪些分类是只有管理员可用的；通过修改是否可用，可以控制分类是否会显示在前台交流模块首页导航，以及文章添加分类选择中。分类管理页面如图 5-23 所示。



图 5-23 分类管理页面

(2) 文章管理：用户点击内容管理下的文章管理按钮，便可以进入文章管理界面。系统前台首先通过默认的查询条件以及分页条件，获取前 15 条文章数据，渲染界面，若文章总条数大于 15 条，则显示分页，否则隐藏分页。管理员可通过数据表头的查询工具进行文章搜索排序等操作。当双击文章列表具体行时，可以进入文章详情页查看文章详情，以及可以为文章添加置顶等管理员具有的操作，文章管理页面如图 5-24 所示。



图 5-24 文章管理界面

5.2.2 案例管理

(1) 案例审核：管理员通过点击案例管理下的案例审核按钮，便可进入案例审核界面。系统前台首先获取所有用户提交的，且处于待审核状态的案例，渲染到此界面显示。管理员可以点击数据表格左侧的详情按钮查看案例详情，也可通过点击项目地址查看具体项目。审核完成后，管理员可以点击对应待审核案例右侧操作列的通过按钮，审核失败管理员可以点击操作列的拒绝按钮。案例审核界面如图 5-25 所示。

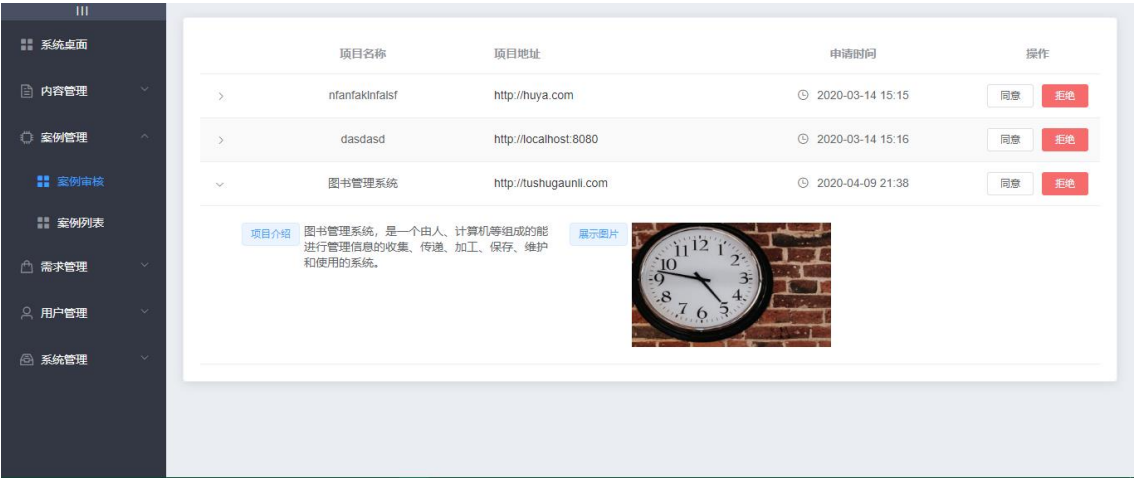


图 5-25 案例审核界面

(2) 案例列表：管理员可以通过点击案例管理下的案例列表进入案例列表界面，管理员可以在该界面搜索栏输入条件搜索案例，点击数据表格左侧详情按钮查看案例，点击数据表格右侧的下架按钮，使得不在前台展示，点击用户名可进入用户主页。案例列表界面如图 5-26 所示。

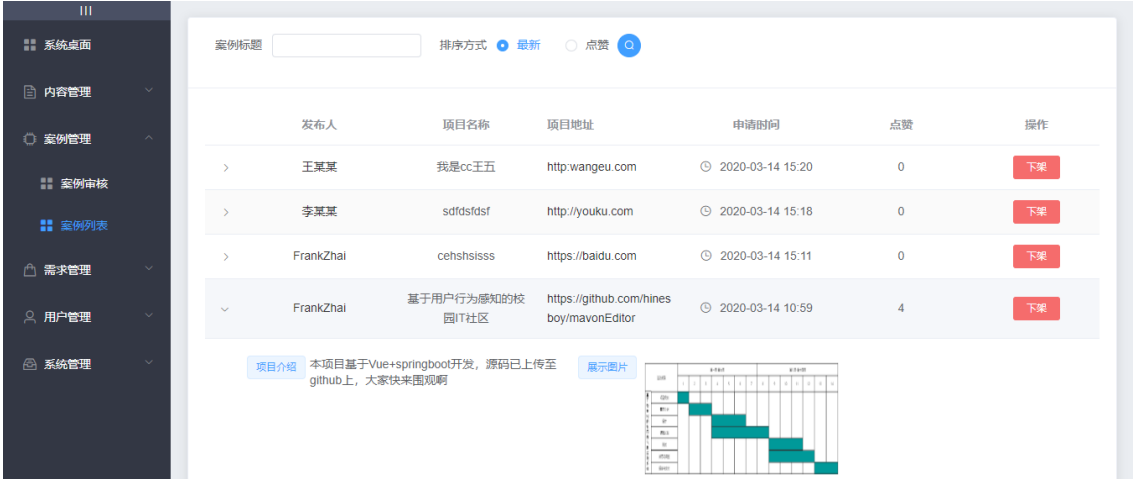


图 5-26 案例列表界面

5.2.3 需求管理

(1) 需求审核：管理员通过点击需求管理下的需求审核按钮，便可进入需求审核界面。管理员可以点击数据表格左侧的详情按钮查看具体要求，审核通过管理员可以点击通过按钮，系统后端会将该条需求自动添加为期一个月的展示时间；审核失败管理员可以点击拒绝按钮。需求审核界面如图 5-27 所示。



图 5-27 需求审核界面

(2) 需求列表：管理员可以通过点击需求管理下的需求列表进入需求列表界面。管理员可以在该界面查看基本信息，点击数据表格左侧详情按钮查看需求具体要求，点击数据表格右侧的下架按钮，使得不在前台展示，点击用户名可进入用户主页。需求列表界面如图 5-28 所示。



图 5-28 需求列表界面

5.2.4 用户管理

管理员可在用户管理下点击用户列表进入用户列表界面，管理员可以通过表头搜索工具栏通过条件搜索用户，可以通过表头的导入按钮导入多个用户，通过添加按钮添加单个用户，通过删除按钮删除多个用户，通过数据表格右侧的操作按钮删除多个不是管理员的用户，通过双击具体的用户所在行进入该用户的用户主页。用户列表如图 5-29 所示。



图 5-29 用户列表界面

5.2.5 系统管理

(1) 权限列表：该列表只展示给系统超级管理员，超级管理员通过点击系统管理下的权限列表，便可以查看系统的所有权限，超级管理员可以操控是否可用开关，从而决定该权限是否可以被使用。系统权限列表如图 5-30 所示。

	权限名称	路径	权限等级	是否可用	图标
1	内容管理		一级		
2	分类管理	/back/categoryManager	二级		
3	文章管理	/back/articleManager	二级		
4	案例管理		一级		
5	案例审核	/back/caseReview	二级		
6	案例列表	/back/caseManager	二级		
7	需求管理		一级		
8	需求审核	/back/exchangeReview	一级		

图 5-30 系统后台权限界面

(2) 角色管理：角色管理是系统管理中最重要的一环，超级管理员通过点击系统管理下的角色管理，即可进入角色管理列表界面。系统后台向后端请求数据，系统后端在业务层获取所有的角色信息，再将每一个角色所对应的权限获取，处理并封装数据通过控制层返回给后台。后台获取数据并渲染成如下图 5-31 所示界面。超级管理员可在查看所有角色，以及点击每个角色信息左侧的详情按钮，查看对应角色所分配的权限信息。超级管理员操作是否可用开关可以启用或禁止使用该角色，通过编辑按钮修改该角色名称以及角色描述，通过分配权限按钮可以为角色分配权限，通过创建角色按钮可以创建一个新的角色，通过删除按钮可以删除一个角色。

添加角色				
	角色名称	角色描述	是否可用	操作
1	内容管理员	测试角色		编辑 删除 分配权限
<div><div>案例管理 × ▶</div><div>需求管理 × ▶</div><div>用户管理 × ▶</div><div>案例列表 × ▶</div><div>需求审核 × ▶</div><div>需求列表 × ▶</div><div>用户列表 × ▶</div></div>				
2	案例管理员	具有案例管理的所有权限		编辑 删除 分配权限
3	通知管理员	具有通知管理的所有权限		编辑 删除 分配权限

图 5-31 系统角色管理界面

(3) 管理员管理：超级管理员通过系统管理下的管理员管理，可以进入管理员管理界面。超级管理员可以在该页面，查看系统所有管理员以及管理员的角色。可以通过控制是否可用开关控制该管

理员权限是否可用，通过角色分配按钮为该管理员重新分配角色，通过删除按钮删除管理员；通过添加按钮，打开添加页面，从用户列表中搜索用户，并为其添加角色生成新的管理员。系统管理员管理界面如图 5-32 所示。



图 5-32 系统角色管理界面

(4) 通知管理：管理员可以通过系统管理下的通知管理，进入系统通知列表界面。后台发送数据请求，系统后端业务层处理业务逻辑，按紧急程度以及创建时间排序，从数据库中获取所有的系统通知信息，后台获取数据渲染成如下图 5-33 所示界面。管理员可以查看系统所有通知，可以对通知进行增、删、改的操作。管理员在添加通知时需要选择通知失效时间，默认失效时间为一周，对于已过失效时间的通知将不会在系统前台显示。



图 5-33 系统通知管理界面

5.3 文章推荐实现

Python 在数值计算方面有着天然的优势，本系统推荐的具体实现使用 Python 编写。推荐算法^[11]通过 LFM(Latent factor model)个性化召回算法实现，该算法在推荐场景中适用于用户数大于物品数的情况。虽然在本系统中文章数随着时间的积累会大于用户数，但是此处并不是指具体的数量级，而是具体的类型。因为本系统针对于本校计算机类专业的学生，所文章类型不会大于用户数。LFM 是属于隐含语义模型的算法，该算法的在推荐中的实现是先找出用户和文章之间的 K 个联系，通过对应的梯度下降算法不断地训练模型，最终明确用户和文章之间的联系，从而为当前用户推荐可能与当前用户存在较深关系的 Top N 条文章。

本系统文章推荐算法的具体实现，首先从数据库中获取所有用户的访问行为数据，判断数据总条数是否满足计算。若不满足则不生成推荐，若满足则计算并获取推荐。接着对用户行为数据进行处理，将评分大于 6 分的作为正样本，其他作为负样本。将处理后的数据通过梯度下降算法，并且

指定隐特征纬度，正则化系数、学习率以及最大迭代次数，不断优化损失函数直至参数收敛，返回训练后的用户数据集和文章数据集，通过计算用户和文章之间的余弦相似度，将计算的结果逆序排序，获取前 15 条文章的 ID 值，并存储到 Redis 中。系统前台请求推荐数据，系统后端直接从 Redis 中获取文章 ID，最后在数据库中获取文章详细内容，在业务层做数据封装返回给前台显示。

主要代码具体实现：

```
def lfm_train(train_data, F, alpha, beta, max_iter):
    """
    Args:
        train_data: 训练集
        F: 用户和文章的隐特征纬度
        alpha: 正则化系数
        beta: 学习率
        max_iter: 迭代次数
    Return:
        dict: key art_id, value: np.ndarray
        dict: key user_id, value: np.ndarray
    """
    user_vec = {}
    article_vec = {}
    for iter_index in range(max_iter):
        for data_instance in train_data:
            user_id, art_id, label = data_instance
            if user_id not in user_vec:
                user_vec[user_id] = init_model(F)
            if art_id not in article_vec:
                article_vec[art_id] = init_model(F)
            delta = label - model_predict(user_vec[user_id], article_vec[art_id])
            for index in range(F):
                user_vec[user_id][index] += beta * (
                    delta * article_vec[art_id][index] - alpha * user_vec[user_id][index])
                article_vec[art_id][index] += beta * (
                    delta * user_vec[user_id][index] - alpha * article_vec[art_id][index])
            beta = beta * 0.9
        return user_vec, article_vec
def model_predict(user_vector, article_vector):
    """
    计算用户向量和文章向量的余弦距离
    Args: user_vector: 用户向量 article_vector: 文章项向量
    Return: 距离值
    """
    res = np.dot(user_vector, article_vector) / (np.linalg.norm(user_vector) *
    np.linalg.norm(article_vector))
    return res
```

6 系统测试与维护

6.1 系统测试概述

系统测试是实现一个系统上线应用的最终阶段也是最重要的阶段，系统测试是通过手动或者自动化的方式来运行或测定软件产品系统的过程，主要用于检测系统是否满足客户规定的需求。本系统需要通过系统测试不断解决和完善系统中的错误，最终达到实际应用的目标。

6.2 系统测试

计算机软件的测试方法有逻辑驱动法、集成测试法、黑测试法以及单元测试法，本系统测试采用黑盒测试法，即将确定的测试方案，通过等价划分的方式分成不同的子集，从每个不同的子集中选取具有代表性的元素构建测试用例^[12]。

6.2.1 系统功能测试

系统功能测试用例，包含系统对用户正常执行操作的反馈，即用户输入正确做出具体的操作；用户异常执行操作的反馈，即用户输入错误或用户执行无效指令，系统给予相关提示。因系统功能过多，此处描述系统部分功能测试用例。

正常操作测试用例如下表 6-1 所示。

表 6-1 正常操作测试用例

编号	测试项	前提条件	操作	输入	预期结果
U0001	用户登录	用户正常通过地址路径在浏览器进行系统首页	1.点击“登录”按钮 2.输入登录信息 3.点击登录页的“登录”按钮	正确填写登录信息 账号：2016212079 密码：test123 验证码：45DF	1.跳转到登录页 2.返回到进入登录的前一个界面 3.提示登录成功
U0002	修改密码	用户正常登录，进入基础信息界面	1.点击“密码”按钮 2.输入原始密码、新密码以及确认密码 3.点击“提交”按钮	正确输入密码信息 原始密码：test123 新密码：test456 确认密码：test456	1.切换到修改密码界面 2.提示修改成功
B0001	发布文章	用户正常登录	1.点击“发布文章”按钮或者对应快捷按钮 2.正确输入文章内容以及文章信息 3.点击“确定”按钮	正确填写文章内容以及文章信息 标题：title 内容：\w{20,} 分类：分享 标签：测试	1.进入撰写文章界面 2.进入我的文章界面 3.提示发布成功
B0002	修改	用户正常登录，且用户已	1.点击要修改文章的“编辑”按钮	正确填写文章内容以及文章信息	1.进入文章编辑界面

	文章	有文章	2.正确输入内容	标题: title edit 内容: \w{20,} 分类: 建议 标签: 测试	2. 返回上一个界面 3.提示修改成功
B0003	删除文章	用户正常登录, 用户已有文章	点击对应文章项的“删除”按钮		1.提示是否确定删除 2.提示删除成功
B0004	搜索文章	用户正常通过地址路径在浏览器进行系统首页	1.点击搜索框 2.正确输入内容	正确查询信息 query: title	1.跳转到查询结果页面, 并显示与“title”有关的文章列表

异常操作测试用例如下表 6-2 所示。

表 6-2 异常操作测试用例

编号	测试项	前提条件		操作	输入	预期结果
U0003	用户异常登录	用户进入登录页	仅账号无效	不输入账号		提示账号不能为空
				输入错误账号	6016222222	提示账号或密码错误
			仅密码无效	不输入密码		提示密码不能为空
				输入错误密码	test001	提示账号或密码错误
			仅验证码无效	不输入验证码		提示验证码不能为空
				输入错误验证码	4D5F	提示验证码错误
U0004	异常修改密码	用户进入修改密码界面	仅原始密码无效	不输入原始密码		提示原始密码不能为空
				输入错误的原始密码	test001	提示原始密码错误
			仅新密码无效	不输入新密码		提示新密码不能为空
				输入不合法的新密码	123	提示密码长度为 6-12 个字符
			仅确认密码无效	不输入确认密码		提示确认密码不能为空
				输入错误的确认密码	testtest	提示确认密码不正确

B0005	异常发布文章	用户进入文章撰写界面	无效的文章内容输入	1.不输入内容 2.输入内容为空		提示文章内容不能为空
				文章字数不符合	\w{1,20}	提示文章字数不能少于 20 字
		无效的文章信息输入		标题为空		提示文章标题不能为空
				标题长度不合法	\w	提示文章标题为 2-20 字
B0006	异常搜索文章	用户正常进行系统首页	搜索内容不合法	输入内容为空或空字符		提示搜索内容不能为空

6.2.2 系统安全测试

通过在浏览器地址栏直接输入越权访问路径，以及后端权限划分的部分用例进行测试，具体测试结果如下表 6-3 所示。

表 6-3 系统安全测试用例

编号	测试项	异常说明	操作	输入	预期结果
S0001	越权访问	直接通过 url 越权访问	未登录，直接通过 url 地址进入后台	http://192.168.56.110:8080/#/back	1.跳转到登录页 2.提示用户请进行登录
			已登录，并不具备管理权限，通过 url 进入后台		1.跳转到前台首页 2.提示用户越权访问
S0002	权限划分	管理员登录后查看权限是否正确	超级管理员进入后台		显示所有权限
			文章管理员进入后台		仅显示文章有关权限
			用户管理员进入后台		仅显示用户有关权限

结束语

本系统的开发是以大学期间所学的课程为基础,通过学习软件开发前沿的最新的实用技术所完成的。在本系统开发期间,本人通过互联网的云课堂以及电子阅览等等途径学习了很多关于 Web 应用前后端开发的主流技术。也使用了很多互联网上的知识分享社区,了解了它们的优缺点,结合自身所在的校园学习环境,明确了本系统应该包含的基础功能以及特色模块。用户可以在交流模块使用到知识分享社区包含的所有的基础功能,在案例模块展示自己所开发的项目,在需求模块提交自己不能实现的需求,让有能力解决的人帮助实现。通过本次毕业设计系统的开发,将我在大学期间所学的理论知识与实践技术得到充分的利用,对于所学的技术有了更深入的理解;对于项目开发的流程,数据库的设计、前后端的交互、前后台的分离以及后端业务的处理有了更深的认识。本系统基本功能都已经完成,但是还有很多不足的地方,比如有些功能的代码不够精简,在接下来的时间里,我会通过不断地优化代码、优化功能,完善本系统。

参考文献

- [1] 奚春雁.CSDN 带高校生走进企业实践——CSDN 创始人和董事长蒋涛先生专访[J].计算机教育,2017,10:1-4.
- [2] 李迎春. 论述软件开发可行性分析与需求分析[C]. 第二十一届中国(天津) 2007 IT、网络、信息技术、 电子、仪器仪表创新学术会议, 2007:131-134.
- [3] Roger Pressman,Bruce Maxim. Software Engineering: A Practitioner's Approach [M]. McGraw-Hill Education,2014.
- [4] 黄健宏. Redis 设计与实现[M]. 北京:机械工业出版社,2014.
- [5] 梁文楷. 基于 Elasticsearch 全文检索系统的实现[J]. 电脑编程技巧与维护, 2019(06):116-119.
- [6] 石珊. 云平台下基于 FastDFS 的文件管理系统的研究与实现[D].电子科技大学硕士学位论文,2019.
- [7] Phillip Webb, Dave Syer, Josh Long, Stéphane Nicoll.Spring Boot Reference Documentation[EB/OL].<https://docs.spring.io/spring-boot/docs/2.2.2.RELEASE/reference/html/>,2019-12-06 06:37:25 UTC.
- [8] 魏斌. 高性能 MySQL 集群部署[J]. 河南科技,2014(14):6-9.
- [9] 赵冠臣,王冬妮,刘至洋,孟振江. 浅谈 Docker 容器技术[J]. 有线电视技术, 2019(09):85-88.
- [10] 王珊,萨师煊. 数据库系统概论[M]. 第五版. 北京:高等教育出版社,2014.
- [11] 刘辉,郭梦梦,潘伟强. 个性化推荐系统综述[J]. 常州大学学报(自然科学版), 2017, 29(03):51-59.
- [12] 刘佳,王丽,杨俊叶. 计算机软件测试方法及应用分析[J]. 科技风,2020(09):119.

致 谢

写到这里，毕业设计将结，大学的时间也已经悄然进入到尾声，在此，我首先要感谢我的毕业设计指导老师马丽生老师，在本系统的功能分析，以及推荐算法理解上给了我很大的帮助，总是耐心的帮我们分析问题，并找出可行的解决方案。还要感谢我的大学辅导员给我提供了很多参加比赛的机会。感谢我在实验室学习期间帮助我的老师、学长、学姐，也感谢我的亲爱的室友们为宿舍学习提供了良好的氛围。感谢大学期间给我上课我的老师们、以及帮助我的同学们。最后还要感恩父母的无私与奉献。