

**毕业论文**

**题 目： 学术论坛系统的设计与实现**

**学生姓名：**  韩天宇 **申请学位门类：** 工学

**学 号：**  201420180924

**学 院：**  软件学院

**指导教师：**  贾惠珍 **职称：** 讲师

**二 零 一 八 年 五 月 一 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）的研究做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）：

签字日期： 年 月 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

**学术论坛系统的设计与实现**

**韩天宇**

Design and Implementation of Academic forum system

HanTianyu

**2018年5月1日**

**摘 要**

在这个知识爆炸的时代，各个学科，各个行业每天都会有很多新的知识和技术诞生；而新诞生的知识和技术若不能及时广播，很有可能会在短时间内埋没，其中的发光点不能被及时挖掘，不能及时的使有用的知识和技术被运用到其需要的场景上。

知识是强国富民的有效途径，知识是唯一生产力，知识可以改变命运。在进行学术研究和知识学习的时候，可参考的图书和别人写的论文或者文献都很零散，如果自行整理，会浪费很多宝贵的时间。

由于以上的问题和需求，提出新建一个学术交流系统，对信息、学术知识、学术论文进行统一整理，统一存储，对于论文和文献的查找相当便利，还可以存储即时的想法和灵感。并能根据用户使用习惯进行个性化定制，比如对用户进行个性化文章推荐、根据用户的使用习惯进行调整用户界面。减少了新知识、新技术、新想法、新论文不能被其需要的人发现和使用的可能。

本文简单叙述了使用到的某些JavaScript框架，例如基于Node.js的后端框架Koa.js和前端框架Vue.js，数据库使用的是Sun公司的MySql数据库。本系统对MarkDown语法进行了支持。

**关键词：**批处理； 学术； 交流； JavaScript； 服务

**Abstract**

In this era of knowledge explosion, various disciplines and industries will have many new knowledge and technologies born every day. If new born knowledge and technologies cannot be broadcast in time, it is very likely that they will be buried in a short period of time. Being excavated in time can not make timely use of useful knowledge and technology to their desired scenes.

Knowledge is an effective way to strengthen the country and enrich people. Knowledge is the only productive force, and knowledge can change its destiny. When conducting academic research and knowledge learning, the books that can be referenced and the papers or articles written by others are scattered. If you organize yourself, you will waste a lot of valuable time.

Due to the above problems and demands, a new academic exchange system was proposed to organize information, academic knowledge, and academic papers in a unified way and store them in a unified manner. The search for papers and documents was very convenient, and instant ideas and inspiration could also be stored. And can be personalized according to user habits, such as personalized articles for users to recommend, adjust the user interface according to the user's habits. Reduces the possibility that new knowledge, new technologies, new ideas, and new papers cannot be discovered and used by the people they need.

This article briefly describes some of the JavaScript frameworks used, such as the Node.js-based backend framework Koa.js and the front-end framework Vue.js. The database uses Sun's MySql database. This system supports MarkDown syntax.

**Keywords:** batch processing; academic; communication; JavaScript; services

# 绪论

## 需求的背景和意义

随着时代的发展，互联网技术发展的非常迅猛，作为互联网产物之一的网页受到很多中小型企业的欢迎，用浏览器展示信息的方式，基于万维网的技术，不用过多的投资即可很快见到很好的效果。

学术论文和文献资源的整合、处理以及维护的工作是一项工作量大、工作内容单调和工作时长繁重的任务，如果通过人为整理会有很大的误操作风险。人为整理各个系统之间是独立的，无法从一个全局角度去查找、更改、发行，上线和下载都很艰难。在此前提下建立一个学术论坛交流系统，在对学术文献和论文的整理的基础上，还支持用户与用户之间交流讨论的功能，对系统内的数据实现系统化的统一管理和监控。

## 系统主要功能

用户信息：存储用户的信息，包括私密的不公开信息，如密码等，和公开信息，如：用户名等。

用户文章：存储用户的文章，包括发表和未发表的文章，用户为每篇文章上传的附件，如论文、图片、视频等。还可以进行文章资源的搜索，对用户部分公开的文章进行发布，使其他用户可以搜索到自己需要的文章或论文资源。用户还可以对文章以评论的形式进行交流。

统一管理：对于非法的用户或文章进行管理，保障用户良好的用户体验，对于非法的用户或文章进行删除，维护良好的网络环境。生成系统的数据日志，在需要时可以提供全面的分析数据和管理数据。将现有系统的各个资源进行整合，使新系统拥有较为全面的数据存储。

## 系统的技术栈以及系统框架概览

### 前端框架Vue.js

Vue.js是一套构架用户界面的渐进式的前端框架，在HTML、CSS和JavaScript的基础上产生的，只关注视图层的渲染，采用自底向上的增量式开发的设计模式。通过尽可能简单的API实现响应式的数据绑定和组合的视图组件。不仅有易于理解的开发文档，还便于其使用在现有项目上，与第三方库既有项目进行整合，当现代化的工具链以及各种支持类库结合使用时，Vue.js页完全能够为复杂的单页面提供驱动。

### 后端框架Koa.js

Koa.js 是由 Express.js 原班人马打造的，致力于成为一个更小、更富有表现力、更健壮的 Web 框架。 使用 Koa.js编写 web 应用，通过组合不同的 generator，可以免除重复繁琐的回调函数嵌套， 并极大地提升错误处理的效率。koa 不在内核方法中绑定任何中间件， 它仅仅提供了一个轻量优雅的函数库，使得编写 Web 应用变得得心应手，优雅、简洁、表达力强、自由度高。本身代码只有1000多行，所有功能都通过插件实现，很符合 Unix 哲学。

### 前端UI框架element-ui

Element UI 是一套采用 Vue 2.0 作为基础框架实现的组件库，它面向企业级的后台应用，能够帮助你快速地搭建网站，极大地减少研发的人力与时间成本。与现实生活的流程、逻辑保持一致，遵循用户习惯的语言和概念，所有的元素和结构需保持一致，比如：设计样式、图标和文本、元素的位置等；通过界面样式和交互动效让用户可以清晰的感知自己的操作；操作后，通过页面元素的变化清晰地展现当前状态；设计简洁直观的操作流程；语言表达清晰且表意明确，让用户快速理解进而作出决策；界面简单直白，让用户快速识别而非回忆，减少用户记忆负担；根据场景可给予用户操作建议或安全提示，但不能代替用户进行决策；用户可以自由的进行操作，包括撤销、回退和终止当前操作等。

### MySQL数据库

Mysql是最流行的关系型数据库管理系统，在WEB应用方面MySQL是最好的RDBMS(Relational Database Management System：关系数据库管理系统)应用软件之一。MySQL是一个关系型数据库管理系统，由瑞典MySQL AB公司开发，目前属于Oracle公司。MySQL是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。Mysql是开源的，支持创建大型的数据库。是一个可以处理拥有成千上万条记录的大型数据库。使用标准的SQL数据库语言形式。MySQL对多个系统上进行了支持，并且可以用于多种编程语言。这些语言包括C、C++、Python、Java、Perl、PHP、Eiffel、Ruby和Tcl等。MySQL是可以定制的，采用了GPL协议，你可以修改源码来开发自己的Mysql系统。

### Visual Studio Code开发工具

Visual Studio Code是微软公司出品的一款轻量级的代码编辑器，有很多功能和支持并且是免费开源的。对JavaScript和NodeJS的支持非常好，自带很多功能，例如代码格式化，代码智能提示补全、Emmet插件等。VSCode推荐一个项目以文件夹的方式打开。相对于其他主流JavaScript开发工具，VSCode更加轻量，它本身的安装包只有40多MB；用户界面更加简洁明了，还有强大的快捷键加持，对现在比较流行的代码托管工具Git进行了很好的集成。

### XMind简介

XMind使用非常方便，在众多思维导图工具软件中脱颖而出，不论是在商业还是个人设计使用都显示出非常大的优势，它所使用的Eclipse RCP是一个全球顶尖的软件架构，这款架构是基于很多基础组件，并把这些基础组件标准化，例如常用的菜单、工具栏等等。XMind创建思想是构建一个具有漂亮的可视化的图形工具，使其简单易用并且操作简单高效的思维导图工具。它努力优化软件的扩展性，使其在绘画思维导图领域有着专业的设计而不止于此，在概念图、笔记、架构图的绘画上都能很好的驾驭。

### 1.3.7 系统框架简述

本系统分为三大模块，第一个模块是用户模块，主要是用户信息的存储和用户信息修改以及删除等功能；第二个模块是用户评论模块，主要记录用户对某一篇论文或文章的评论，还有用户对评论的回复。第三个模块是论文模块，主要功能是存储用户的论文和文章，用户可以进行修改，删除，发布等操作。

# 需求分析和系统设计

## 2.1 需求分析

### 2.1.1 需求分析概述

需求分析也叫做软件需求分析、系统需求分析或需求分析工程等，是负责开发的相关人员经过深入细致的群众调研和对所有调研的专业分析，准确理解用户和项目的功能、性能、可靠性等具体要求，将用户非形式的需求表述转化为完整的需求定义，从而最终确定目标系统必须满足某些功能和做什么的过程。需求分析是一项对应用环境以及面对的问题进行深入理解和详细分析的工作，对现有问题涉及的已有信息、目标功能以及系统行为建立系统模型，精确用户的需求。

### 2.1.1 任务概述

1.任务目标

在这个信息化的时代，快速的传播学术信息是非常需要的，学术信息不能再全局流通会拖慢时代科技发展的步伐，学术交流系统为人们集中交流、学习、倾吐心得提供了一个良好的平台和环境，使世界各地的用户进行专业的学术信息的交流成为可能。

## 2.2系统结构

学术论坛系统包含有板块、主题和评论。板块是对所有论文和文章的分类，评论是在主题、文章或论文下发表的，是仅针对已有账号的评论和回复。首页会显示论坛的所有论文以及作者、发布时间、类别等，进入文章和论文后会显示论文的内容和作者，论文的详细页面还会显示论文作者的积分、等级和注册时间，页面下面会显示所有的评论和用户针对评论的回复以及回复的时间和回复者。

### 2.1.用户的基础操作和功能

学术交流论坛系统的用户的基础操作就是上网进行对系统的使用，若想发表有样式的有更好的阅读体验的文章，就要学会使用基本的markdown语法进行写作。

1. 普通用户

作者管理员。作者可以进行查看论文、评论、写文章等操作，还可以进行登录和注销。

1. 系统管理员

管理员也是用户，也有用户的基本操作和功能，还要进行对用户的文章、评论、回复和用户账户进行管理，维护系统的正常运行和对变动进行相应的操作。最好具备计算机专业知识，以便于在系统故障时对系统及时进行维护。

### 2.2学术交流论坛系统功能

1.用户中心：

用户可以进行注册、登录操作来申请账号，可以用注销操作删除账号，用写文章、发表文章来发表自己的论文或文章，可以评论别人的文章和论文并可以进行对回复的评论。按关键字搜索文章和论文，查看自己发表的文章和论文。

2.作者中心

发表文章、编辑文章、回复评论、搜索文章、删除文章、文章置顶等。

3.管理员中心：

删除文章、文章置顶、设置热门文章、按关键字搜索文章、对普通用户账户删除、授权，对文章展示板块的权限管

4.注册用户

当用户进入网站首页时，用户可以登录，没有登录的用户可以查看论文，但是不能进行评论和进入个人中心等操作。登录之后用户就可以进行评论和对评论进行回复操作，还可以进入个人中心个性化自己的信息，如：修改头像、修改用户名、修改手机号和邮箱等。

5.管理员

管理员可以添加或减少用户权限，管理注册用户，管理所有文章和与文章关联的评论或回复。

### 2.3系统数据流图

系统的数据流图（Data Flow Diagram，简称DFD）是从数据的传递和数据的加工角度出发，以有联系的图形的方式来表达系统的逻辑功能、系统内部的数据流向、逻辑流向、和逻辑的变换过程，是用于表示软件模型的结构化系统分析方法的主要表达工具。如图2-1。

管理员

管理注册用户

管理板块

管理论文信息

系统

注册

查看文章

写文章

发表文章

登录

修改文章

搜索文章

参与评论

用户

图2-1

用户注册：用户输入信息，信息经过前端校验后传给后端，后端确认信息无误后存入数据库中，存入成功或失败都会返回消息提示用户。

用户登录：用户在登录页输入注册的用户信息，前端进行信息安全性校验，确认信息安全可靠后传入后端，后端校验用户信息是否真实，返回登录结果。

写文章：用户注册账号并登录后才可以进行写文章的操作，点击写文章后，系统会直接创建一篇新的文章并对新的文章进行修改，新的文章会保存到用户中心-我的管理里。得益于vue.js的响应式框架，修改文章时系统会即时保存用户修改，并不需要点击保存。

读文章：用户在首页可以浏览所有文章，可以进行关键字搜索查询自己想要的文章，或根据不同分类查询文章，点击文章标题会进入文章详细信息页，展示文章的内容和所有评论、回复。

用户个人中心：用户可以进入个人中心对自己的个性信息进行修改，如用户名、密码、头像、联系方式等信息；还可以管理自己的文章，在个人中心的‘我的管理’里对用户的文章进行编辑、删除等操作；在个人中心查看自己的使用信息，如登录时间、注销时间、浏览过的文章、参与的评论等。

管理员用户：管理员用户有修改、编辑、删除普通用户的权限；管理所有文章信息和所有用户评论信息，管理文章页面有编辑文章和删除文章的功能；‘管理文章类别’页面有删除某一种类别下的所有文章的功能。

### 2.4系统用例图

用例图是指由参与者（Actor）、用例（Use Case），边界以及它们之间的关系构成的用于描述系统功能的视图。用例图（User Case）是外部用户（被称为参与者）所能观察到的系统功能的模型图。用例图是系统的蓝图。用例图呈现了一些参与者，一些用例，以及它们之间的关系，主要用于对系统、子系统或类的功能行为进行建模。本系统的用例图如图2-2所示。

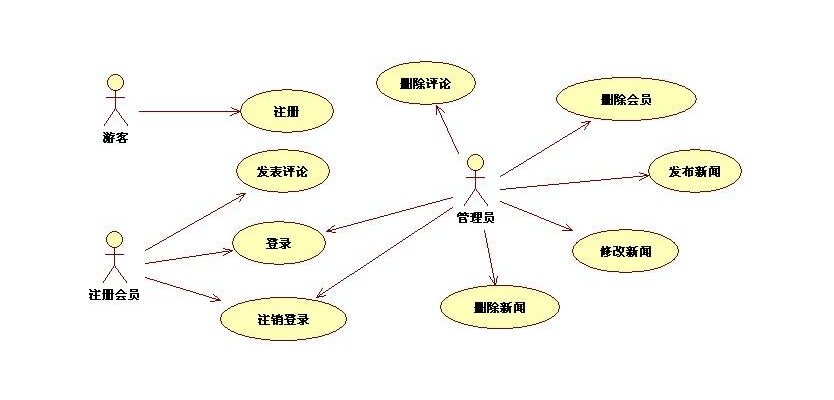


图2-2

系统用例都包含用户注册、用户登录、发表文章、修改文章、删除文章、搜索文章、参与评论、参与回复，管理员管理所有文章、管理用户、管理评论。

## 2.3 系统数据库设计

### 2.3.1 数据库结构分析

系统有用户模块、文章模块、评论模块、用户使用情况模块和管理员管理模块。需要‘用户信息表’存储用户信息；需要‘用户登录信息表’存储用户的使用信息；需要‘用户个性化信息表’存储用户的个性化设置；需要‘用户评论信息表’存储用户评论和回复的信息；需要‘用户使用情况表’来记录系统被使用的情况，需要‘论文信息表’存储用户书写的文章信息；需要‘论文附件表’来存储论文的附件存储位置。

### 2.3.2数据库表详细信息

（1）用户信息表：

用户信息表主要存储用户在注册时填写的个人信息，用户名和密码用来在登录时校验用户的唯一标识，其他信息用户可以选择性公开。表结构如表2-3所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为主键 | 允许空 | 字段描述 |
| userID | INT | YES | NO | 用户唯一ID |
| username | VARCHAR(50) | NO | NO | 用户名 |
| password | VARCHAR(50) | NO | NO | 密码 |
| sex | VARCHAR(10) | NO | YES | 性别 |
| age | INT | NO | YES | 年龄 |
| telephone | VARCHAR(50) | NO | YES | 用户电话号码 |
| QQ | VARCHAR(50) | NO | YES | 用户QQ号码 |
| WeChat | VARCHAR(50) | NO | YES | 用户微信号码 |
| email | VARCHAR(50) | NO | YES | 用户邮箱 |
| headImgUrl | VARCHAR(50) | NO | YES | 用户头像地址 |

表2-3

（2）用户个性化信息表

用户个性化信息表是用来存储用户对系统的个性化的使用习惯，使用户可以以自己最喜欢的、最习惯的方式使用系统。如表2-4所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为主键 | 允许空 | 字段描述 |
| userID | INT | YES | NO | 用户唯一ID |
| autoLoadContent | Int | NO | YES | 当浏览到底时是否自动加载内容，0为不加载，1为加载 |
|  |  |  |  |  |

表2-4

（3）用户登录信息表。

用户登录信息表主要存储用户的登录详细信息：登录时间、登出时间、登录地点、登录设备和用户在线时间，用于日后分析用户使用习惯，检测用户使用是否常规。如表2-5所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为主键 | 允许空 | 字段描述 |
| loginID | INT | YES | NO | 用户登录信息唯一ID |
| userID | VARCHAR(50) | NO | NO | 用户ID |
| loginTime | VARCHAR(50) | NO | NO | 登录时间 |
| logoutTime | VARCHAR(50) | NO | YES | 注销时间 |
| loginPosition | VARCHAR(50) | NO | YES | 登录地点 |
| loginDevice | VARCHAR(50) | NO | YES | 登录设备信息 |
| onlineTime | VARCHAR(50) | NO | YES | 在线时间 |
|  |  |  |  |  |

表2-5

（4）论文信息表

论文信息表主要存储论文的详细信息。如表2-6所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为主键 | 允许空 | 字段描述 |
| thesisID | INT | YES | NO | 论文唯一ID |
| userID | VARCHAR(50) | NO | NO | 用户ID |
| thesisTitle | VARCHAR(50) | NO | YES | 论文标题 |
| thesisContent | mediumtext | NO | YES | 论文内容 |
| thesisBeforeContent | mediumtext | NO | YES | 论文更改前内容 |
| thesisUUID | VARCHAR(50) | NO | NO | 论文唯一UUID |
| titleImgUrl | VARCHAR(50) | NO | YES | 论文展示图片URL |
| public | tinyint | NO | NO | 论文是否发布 |
| countNum | Int | NO | YES | 论文浏览统计 |
| time | VARCHAR(50) | NO | YES | 论文创建时间 |
| thesisKind | VARCHAR(50) | NO | YES | 论文类别 |

表2-6

（5）评论信息表

评论信息表是用来存储用户对某一篇论文的评论和对评论的回复的详细信息。discussID是唯一自增键，为每一条评论或回复添加唯一标识。表如图2-7所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为主键 | 允许空 | 字段描述 |
| discussID | INT | YES | NO | 评论唯一ID |
| userID | INT | NO | NO | 用户ID |
| thesisID | INT | NO | NO | 评论所对应的文章 |
| discuss | VARCHAR(50) | NO | YES | 用户评论内容 |
| otherDiscussID | INT | NO | YES | 用户回复评论所对应的ID |

表2-7

（6）论文附件信息表

论文附件信息表是主要用来存储论文的附件信息，在以书面形式下不能完全展示论文的内容或者作者的主题思想和详细步骤时，作者可以上传论文的附件，用以提供更加形象、具体、详细的信息。如表2-8所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为主键 | 允许空 | 字段描述 |
| thesisID | INT | YES | NO | 论文唯一ID |
| wordAppendix | VARCHAR(50) | NO | YES | Word格式的附件地址 |
| excelAppendix | VARCHAR(50) | NO | YES | Excel格式的附件格式地址 |
| pdfAppendix | VARCHAR(50) | NO | YES | PowerPoint格式的附件地址 |
| zipAppendix | VARCHAR(50) | NO | YES | Zip压缩包形式的附件地址 |
| audioAppendix | VARCHAR(50) | NO | YES | 音频格式的附件地址 |
| videoAppendix | VARCHAR(50) | NO | YES | 视频形式的附件地址 |
| otherAppendix | VARCHAR(50) | NO | YES | 其他形式的附件地址 |

表2-8

（7）系统推送信息表

系统推送信息表是用来记录和管理系统向用户推送的广告、通知等信息。如表2-9所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为主键 | 允许空 | 字段描述 |
| messageID | INT | YES | NO | 系统推送消息唯一ID |
| messageContent | VARCHAR(50) | NO | YES | 推送的消息的内容 |
| imgAppendix | VARCHAR(50) | NO | YES | 推送消息的附加图片图片的地址 |

表2-9

# 学术论坛系统详细设计

## 3.1 简介

学术论坛系统主要是用以存储和分发用户数据，在数据上具有及时性、智能存取、立体呈现、存储可靠等四大特性。其中，及时性是对数据的保存做到了及时修改及时保存的操作，以此方式避免了用户编辑修改后误操作或未能及时保存导致数据丢失的风险；数据的可靠性方面，系统在使用了MySQL数据库的基础上，为数据库分配单一的服务器，阿里云的数据库本身就带有数据恢复功能，并且在此基础上再给数据库定时备份，这样可以保证数据库的数据不容易丢失；数据的立体呈现体现在对数据的展示，对文章的类别加以区分，展现时更加清晰，图表的展示使数据易于读懂。

本系统是基于组件的开发，使用组件化开发后，我们只需要对每一个模块的对应的组件进行维护，在修改过程中不会影响到同一个系统的其它文件。而且组件化模块化使文件结构更加清楚，同时方便后台开发的使用；块状化结构，增加CSS代码的复用性，并且方便扩展；一个模块一个标识，标识加元素标签定位书写CSS，减少大量因需要定位DOM元素而添加的类名，网状试开发，大块套小块，扩展更加灵活！

根据学术论坛系统的特性，学术论坛系统主要功能主要由四部分组成，分别是：用户管理、论文管理、评论管理、系统管理。

## 3.2 用户登录组件

用户登录页面使用户进入个系统性化的一个入口，用户登录页并不是进入系统的唯一入口，只是登录后，可以增加用户的很多会员式的功能，可以使用大部分对注册用户开放的功能，若未登录用户只可以浏览热门文章。

系统通过登录页对用户的登录信息进行校验，用户名校验是否为空，是否包含非法字符，防止恶意提交篡改权限和用户误操作；密码校验是否为空，防止错误操作，是否包含非法字符，防止篡改数据库。

点击登录按钮，系统提交数据给后台，若后台返回数据正常，显示用户登录成功，进入首页。界面如图3-1所示，由于使用elementUI框架，校验是在数据里定义规则，关键如代码段3-1所示。



图3-1

|  |
| --- |
| rules: {  username: [  { required: true, message: '请输入用户名', trigger: 'blur' },  { min: 3, max: 15, message: '长度在 3 到 15 个字符', trigger: 'blur' }  ],  password: [  { required: true, message: '请输入密码', trigger: 'blur' }  ]  } |

代码段3-1

## 3.3 用户注册组件

用户注册页面是用户向系统申请使用更多更全面的功能的一个入口，用户注册页包含用户的登录名、密码、微信号、QQ号、性别、年龄、电话号、用户头像的URL、电子邮箱地址等；在输入时校验用户名是否可用，包含用户名是否重复和用户名是否含有非法字符；校验密码是否超过4位，是否含有非法字符；校验邮箱、微信号、手机号、QQ号是否是正常的号码。

全部校验完成后，点击注册向后台提交数据，提示完成注册，跳转到登录页。注册页面如图3-2所示，关键代码如代码段3-2所示。



图3-2

|  |
| --- |
| 针对用户名校验：  var checkUsername = (rule, value, callback) => {  if (value.replace(/(^\s\*)|(\s\*$)/g, "") === '') {  return callback(new Error('用户名不能为空'));  }else if(/[@#\,\;\'\"\$%\^&\\*]+/g.test(value)){  return callback(new Error('含有非法字符'));  }else{  this.$http.post('http://localhost:3000/register', { 'username': value })  .then(function(res){  console.log(value)  console.log(res)  if(res.data){  return callback(new Error('用户名已被占用'))  }  })  }  用户名和密码规则：  rules: {  username: [  { required: true, message: '请输入用户名', trigger: 'blur' },  { min: 3, max: 15, message: '长度在 3 到 15 个字符', trigger: 'blur' }  ],  password: [  { required: true, message: '请输入密码', trigger: 'blur' }  ]  } |

代码段3-2

## 3.4 头部导航组件

头部导航组件包含系统logo、各个模块入口、全局模糊搜索框、写文章入口和登录注册入口；全局搜索框是用来按关键字搜索用户可能需要的文章、作者或文章相关评论等，搜索结果会展示在首页的内容区域，提供一个快速、准确的查找入口，注册登录入口在用户登录完成后会变成用户的个人头像，展示用户的个性信息，点击头像用户便可以进如个人中心，编辑和查看自己的个性化信息。未登录状态如图3-3所示，登录状态如图3-4所示。关键代码如代码段3-3所示。



图3-3 未登录状态



图3-4 登录状态

|  |
| --- |
| 部分代码截取：  <ul class="nav-ul-pc nav-list-margin">  <li v-for="item in navli" v-bind:key="item.index"><a href="#" class="nav-link-pc">{{ item.name }}</a></li>  <li class="nav-form-margin">  <form action="" v-bind:class="navInputBlur ? 'nav-form-blur' : 'nav-form-focus'">  <input type="text" maxlength="32" v-on:blur="checkBlur" v-on:focus="checkFocus" v-bind:class="navInputBlur ? 'nav-search-input-focus' : 'nav-search-input-blur'"><icon name="search" class="nav-search-icon-blur" v-bind:class="navInputBlur ? 'nav-search-icon-blur' : 'nav-search-icon-focus'"></icon>  </form>  </li>  <li class="nav-writePage mainColor" @click="LinkTowritePage">  <icon name="file-text-o" class="writePageIcon mainColor"></icon><span class="writePage mainColor">写文章 </span>  </li>  <li v-if="loginState">  <div class="indexTopBar\_userDiv" @click='userHome' :style="{backgroundImage: 'url('+ userInfo.headImgUrl +')'}">  </div>  </li>  <li v-else>  <span class="nav-login mainColor" v-on:click="loginShow">登录</span>  <span class="nav-register mainColor" v-on:click="registerShow">注册</span>  </li>  </ul> |

代码段3-3

## 3.5 主页文章展示模块

主页文章展示模块默认按观看最多的顺序自上而下展示论文，展示每篇论文的种类、作者、发布时间、标题和展示图片等；论文根据不同种类有不同种颜色；点击作者可查看作者部分选择性公开的详细信息，点击种类可筛选此种类的观看最多的论文，其他部分是进入论文详细的入口，点击即可查看论文的详细信息。展示如图3-5所示，关键代码如代码段3-4所示。



图3-5 首页论文展示

|  |
| --- |
| For循环li元素：  <li v-for="(item1, index) in index\_content" v-bind:key='item1.index' class="index\_content\_li">  <a class="index\_content\_li\_a" @click="goThesis(item1.thesisID)">  <div class="index\_content\_li\_a\_div">  <div>  <div>  <a href="">{{ item1.thesisTitle }}</a>  </div>  <div class="index\_content\_li\_div">  <ul>  <li><span class="index\_content\_li\_span" v-bind:class="'index\_content\_li\_span\_color\_' + item1.thesisKind">{{ item1.thesisKind }}</span></li>  <li class="index\_content\_li\_2"><div>{{ item1.username }}</div></li>  <li class="index\_content\_li\_3">{{ item1.time }}</li>  </ul></div></div></div><div></div> </a></li> |

代码段3-4

## 3.6 写文章模块

用户点击头部导航的写文章按钮时，系统会校验用户是否登录，若未登录则进行路由拦截，路由拦截如代码段3-5所示。

|  |
| --- |
| router.beforeEach((to, from, next) => {  let sessionUserInfo = window.sessionStorage.getItem('userInfo') || '{}';  let userInfo = JSON.parse(sessionUserInfo).data || '';  if(to.meta.requireAuth && (userInfo.username === '' || userInfo.username === undefined)){  console.log('require')  next({path: '/login'})  }else{  next()  }  }) |

代码段3-5

用户点击写文章进入写文章页面时，便立即创建草稿，草稿保存在用户的个人中心的‘我的管理’下，鼠标光标会自动在输入标题的输入框下获取焦点，提示用户输入文章标题，右边可以添加展示图片、添加论文附件和选择论文种类，最右边是用户的个人中心的入口。

编辑文章内容支持Markdown语法，左边用户输入后，右边会即时出现编辑效果，编辑器还支持友好的快捷操作栏，可以帮助用户快速的使用markdown语法编辑出样式丰富的文章，除对markdown的语法进行快捷输入支持外，编辑器还支持通用的键盘组合快捷键操作，如Ctrl+Z撤销、Ctrl+S保存、Ctrl+C复制、Ctrl+V粘贴；并且还支持预览、全屏和收起右边效果展示栏等。用户编辑后并不需要保存操作，草稿是即时性保存的。页面效果如图3-6所示，关键代码如代码段3-6所示。

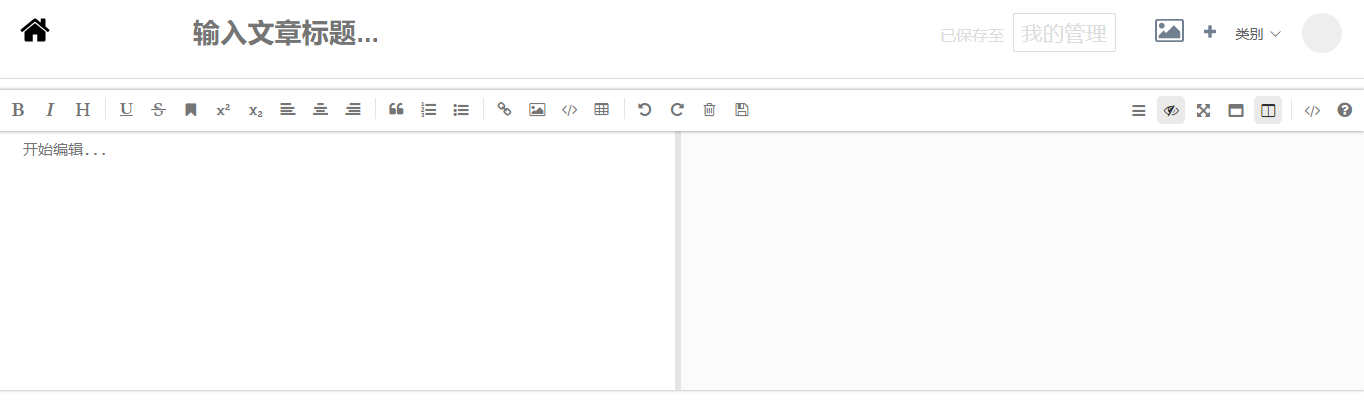


图3-6 写文章页面展示

|  |
| --- |
| 标题部分：  <input type="text" class="WP\_input1" v-model="title" placeholder="输入文章标题...">  编辑器插件：  <div class="WP\_write">  <mavon-editor @change="$change" @save='$save'></mavon-editor>  </div> |

代码段3-6

## 3.7 论文详细组件

论文详细包含三个子组件，第一个是头部导航组件，上面已经介绍过，这里就不再赘述；第二个是论文内容组件，论文内容组件是展示作者编辑并保存的论文的内容；第三个是评论组件，展示用户的评论和对评论的回复。

### 3.7.1 论文内容组件

论文内容组件首先展示作者信息：作者姓名、作者头像，论文创建时间，剩下就是展示论文的内容，效果如图3-7所示，关键代码如代码段3-7所示。

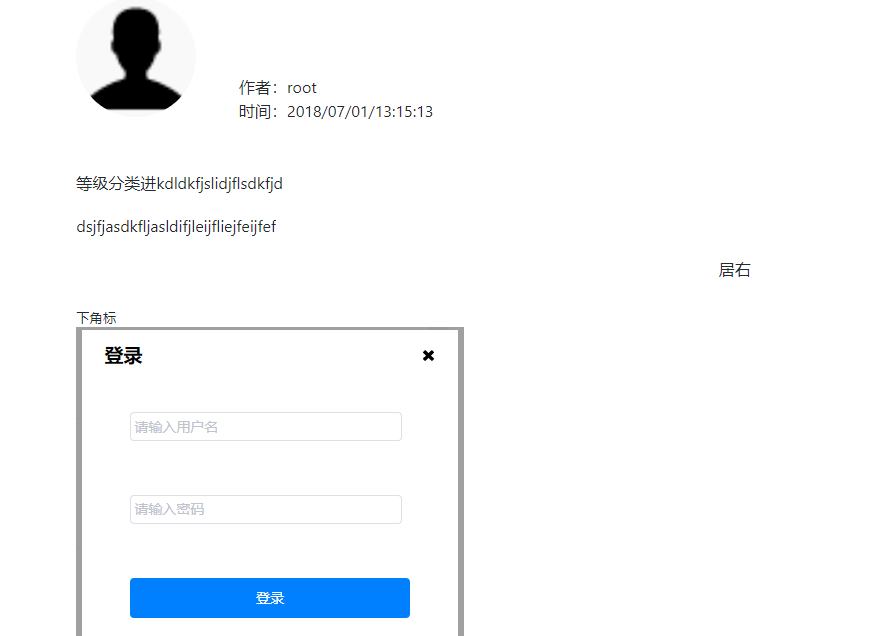


图3-7 论文详细

|  |
| --- |
| <div class="UH\_info" :style="{backgroundImage: 'url('+ thesisInfo.headImgUrl +')'}">  </div>  <div class="t\_thesisUserInfo">  <span>作者：{{ thesisInfo.username }}</span><br>  <span>时间：{{ thesisInfo.time }}</span>  </div>  </div>  <div class="v-note-show">  <div class="v-show-content scroll-style" id="t\_content">  </div>  </div> |

代码段3-7

### 3.7.2 论文评论部分

论文评论是用户在浏览论文时可以对论文进行评论，或者对已经有的评论进行回复。在用户进行评论时系统会检测用户是否登录，若检测到用户未登录，则会跳转至登录页提示用户登录。效果页如图3-8，关键代码如代码段3-8.



图3-8 评论效果图

|  |
| --- |
| <div class="UH\_info" :style="{backgroundImage: 'url('+ thesisInfo.headImgUrl +')'}">  </div>  <div class="t\_thesisUserInfo">  <span>作者：{{ thesisInfo.username }}</span><br>  <span>时间：{{ thesisInfo.time }}</span>  </div>  </div>  <div class="v-note-show">  <div class="v-show-content scroll-style" id="t\_content">  </div>  </div>  <!-- 评论评论的评论 -->  <ul>  <li v-for="item1 in item.childDiscuss" :key="item1.id" class="t\_c\_li">  <div class="indexTopBar\_userDiv" :style="{backgroundImage: 'url('+ item1.headImgUrl==null ? 'http://localhost:3000/image/user.png' : item1.headImgUrl +')'}">  </div>  <!-- 评论信息 -->  <div class="t\_c\_comment\_header">  <div class="t\_c\_user\_info">  <div class="t\_c\_user\_name">  {{ item1.username }}  </div>  </div>  </div>  <div class="t\_c\_response">  <el-popover  placement="right"  :title="'回复:' + item1.username"  trigger="hover">  <textarea placeholder="说说你的看法" v-model.trim="responseMsg" class="t\_content-input" style="overflow: hidden; word-wrap: break-word; height: 60px;"></textarea>  <el-button slot="reference">回复</el-button>  <el-button type="primary" icon="el-icon-edit" circle @click="t\_c\_response(item.thesisDiscuss.discussID)"></el-button>  </el-popover>  </div>  <div class="t\_c\_content">  <span v-html="item1.discuss"></span>  </div>  </li>  </ul>  </div> |

代码段3-8

## 3.8 个人中心