

**《软件设计综合课程设计》**

**报 告**

题目： 个人博客系统

学 院： 计算机信息工程学院

专 业： 软件工程

班 级： 2019级数软件工程一班

学 号： 2019090640114

姓 名： 梁渲

指导教师： 张丽梅

完成日期： 2021年6月9日

**目 录**

[1 需求分析 1](#_Toc105598784)

[1.1 项目背景和意义 1](#_Toc105598785)

[1.2 系统需求分析 1](#_Toc105598786)

[2 系统设计 2](#_Toc105598787)

[2.1 总体设计 2](#_Toc105598788)

[2.2 数据库方案设计 4](#_Toc105598789)

[3 系统编码 8](#_Toc105598790)

[3.1 登陆和注册功能 8](#_Toc105598791)

[3.2 验证码功能 11](#_Toc105598792)

[3.3 文章添加功能 12](#_Toc105598793)

[3.4 文章搜索功能 15](#_Toc105598794)

[3.5 评论功能 16](#_Toc105598795)

[4 系统功能调试 18](#_Toc105598796)

[4.1 开发环境 18](#_Toc105598797)

[4.2 调试环境 18](#_Toc105598798)

[4.3 功能测试 18](#_Toc105598799)

[5 参考文献 22](#_Toc105598800)

1. 需求分析
   1. 项目背景和意义

进入二十一世纪，以互联网为核心的现代网络和通信技术已经得到了快速的发展和广泛的应用，各种网络通信工具也随时代而生。其中就有QQ、论坛、贴吧、社区等较受广大人民欢迎，也是现在发展的比较成熟的信息交流工具。随着网络技术的日渐成熟，大部分普通人记录生活的方式是文图形式，工具仍是纸质、QQ空间或者微信朋友圈，但是这类工具的记录有限，如果想要发表更多想法和见解我认为个人博客是最好的选择。

以前我们记录个人生活的时候，我们可以在QQ空间写日志，现在博客也成为人们发表网络日记的主要方式之一，它是供用户在网络上发表个人文章的社交平台，发表者通常为博主，博客通常由博主个人管理，不定期更新发表新文章。博客网站主要是为用户提供一个社交平台，可以让那些兴趣爱好相同、工作方向相关、学习内容相近的人有一个共同的社交圈子，博主们可以互相交流、相互评论，网络上提供博客平台的网站也有不少，如CSDN、博客园等，但他们或多或少对博主都有技术上的限制，所以本文设计了这个个人博客平台。博主可以根据自己的需求选择喜欢的功能，也可以没有任何限制地发表自己的文章。

* 1. 系统需求分析

个人博客系统主要分为三个角色：普通访客，注册用户、管理员。普通访客可以查看所有的博客、所有的分类、所有的标签、可以查询博客、可以打赏管理员。管理员拥有所有权限，可以对博客内容、网站设置和用户信息等进行添加，删除，修改，查询等操作，同时更新数据库里的数据。

前台参考"Hexo"的"Butterfly"设计，美观简洁，响应式体验好。后台参考"element-admin"设计，侧边栏，历史标签，自动生成。文章采用Markdown编辑器，写法简单。

课题前端采用vue，vuex，vue-router，axios，vuetify，element，echarts，等网页编写技术和框架，后端采用SpringBoot，nginx，docker，SpringSecurity，Swagger2，MyBatisPlus，Mysql，RedisRabbitMQ，Websocket框架实现了数据接收与处理。系统采用了前后端分离的方式，极大地缓解了后台的压力，提升了网站的安全性、可靠性、易用性，提高了数据库的可恢复性与安全保护。

1. 系统设计
   1. 总体设计

博客网站究其根本是一个社交网站，它有功能俱全、页面简介、博客编写方便以及管理简单的特点。其前端模块包括用户对博客和评论页面的浏览以及用户的登录、注册、注销界面，用户既可以对博客进行评论，又能够对评论进行回复。分为前台模块和后台模块，

前台模块如图2-1所示，详细功能为：（1）用户的登录、注册。用户进入网站后可以选择登录账号与否，若用户没有账号则可以注册账号，在用户输入所要求的信息后，系统将用户数据传入数据库，用户即可登录。用户可以修改账号信息账号，退出网站登录。（2）博文、评论、留言板及主页信息的显示。用户、游客可以浏览博客以及下方的评论，还可以在主页面中浏览主页信息。（3）博客评论功能。只有已注册用户可以对博客进行评论，评论后他人或自己还能对评论进行回复。（4）留言板功能。用户可以在留言板上留言，表达自己对网站的想法或提出建设性的意见。

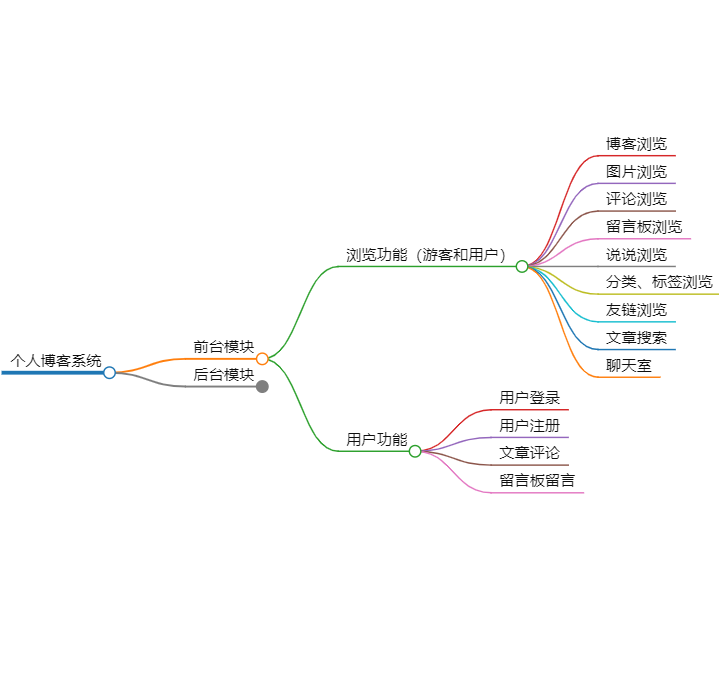


图2-1 前台模块

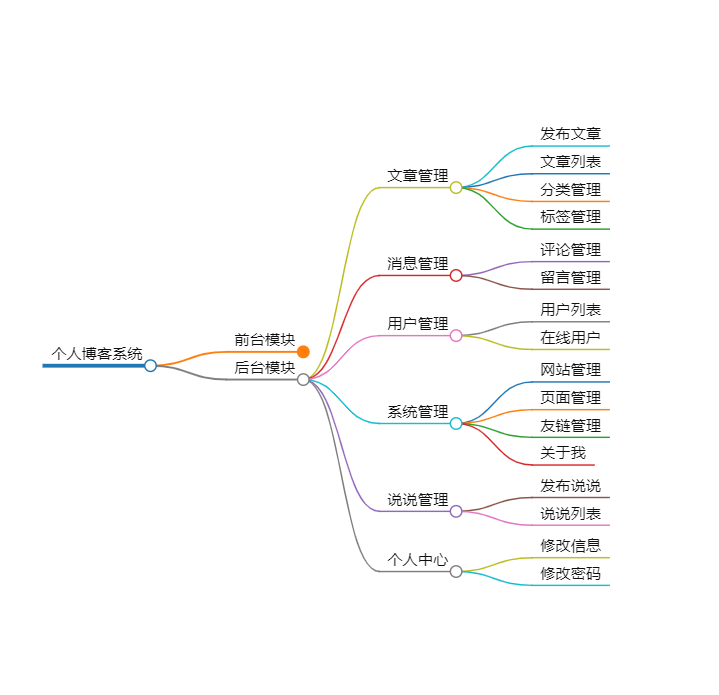
前台模块如图2-2所示，详细功能为：（5）文章管理。博主可以发表文章、浏览文章列表、给文章添加标签和分类、预览图。也可以对其进行修改、置顶等操作。（6）消息管理。用户在文章下评论、留言板留言后后会交由管理员审核，只有通过审核后才会显示。审核功能管理员也可以关闭。（7）用户管理。显示当前已注册用户和在线用户的信息，包括昵称、登陆ip、登陆地址、创建时间和上次登录时间，管理员也可以对用户赋予不同的权限：管理员、用户和测试，能在后台进行相应的操作。如果对用户所作所为有不满之处也可以禁用该用户的登陆。（8）系统管理。负责网站的主页信息、社交信息、和其他功能的修改；对各个页面添加不同的页面图；增删改友联信息；填写网站作者的个人信息，让大家能更了解你。（9）说说管理。此功能是从QQ空间的说说、微信的朋友圈功能延申出来的，用于分享当日心情或用于记事本。（10）个人中心。可以修改管理员信息及修改密码。

图2-2 后台模块

* 1. 数据库方案设计

该项目共有16个数据表，数据表之间不直接联系，都有各自属性为id的主键，且没有外键。如图2-3：

图2-3 E-R图

文章表，如图2-4所示。存储文章id、发布者id、分类id、文章预览图、文章标题、文章内容、文章类型、原文链接、是否置顶、是否展示、及发表时间

和更新时间。

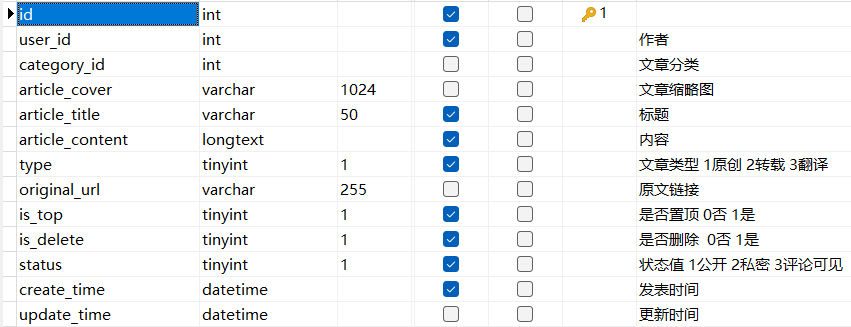


图2-4 文章表

评论表，如图2-5所示。存储用户id、文章id或友联id或说说id，依据此id去不同的表中找数据、评论内容、回复用户的id、父评论id、评论类型、评论时间等。

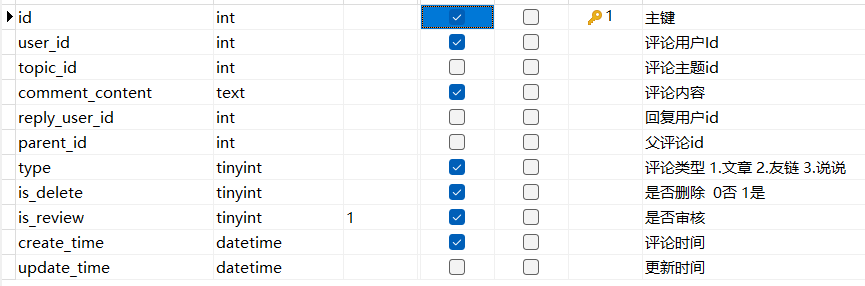


图2-5 评论表

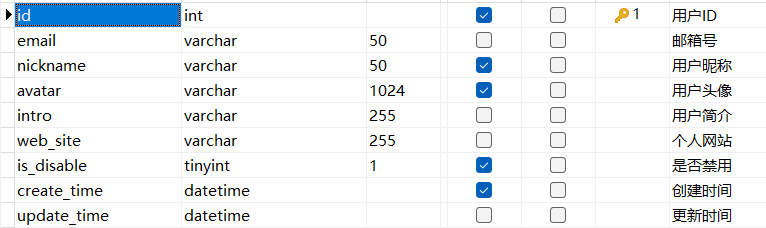
用户个人信息表，如图2-6所示。存储已注册用户的id、注册邮箱、用户昵称、头像、简介、个人网站、创建时间、修改时间等信息，除了后三个其余都是用户自己可以修改的，拥有用户id后才可以使用评论和留言等等在网站上留下信息的功能。

图2-6 用户个人信息表

分类表，如图2-7所示。每个文章至多可以有一个分类，所以是一对一的关系，所以只需要一个分类表，且分类表中对每个值都给予一个独特的id值，将值赋给文章表中的category\_id属性，每次查找的时候只需要在文章表和分类表中找出相同的结果即可。

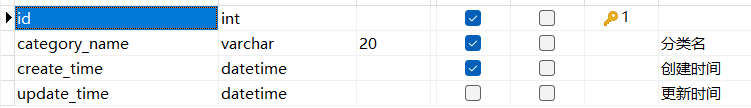


图2-7 分类表

留言板表，如图2-8所示。每个用户或来访游客都可以在前台页面“留言板”中留言。该表中存放了的用户id也包括游客、昵称有游客、管理员和用户昵称、头像、留言内容、用户IP、用户地址、弹幕速度默认都为1、发布时间等信息。

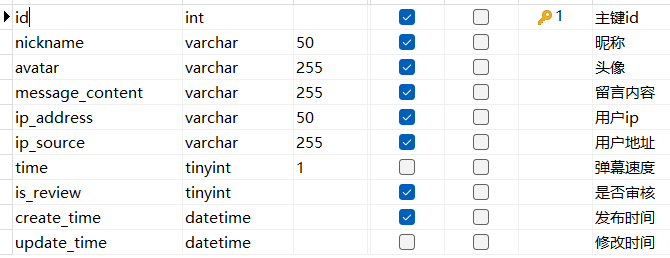


图2-8 留言板表

用户注册表，如图2-9所示。存储用户id、用户名、密码、用户类型有管理员，测试和普通用户、ip地址及物理地址、创建时间等信息。

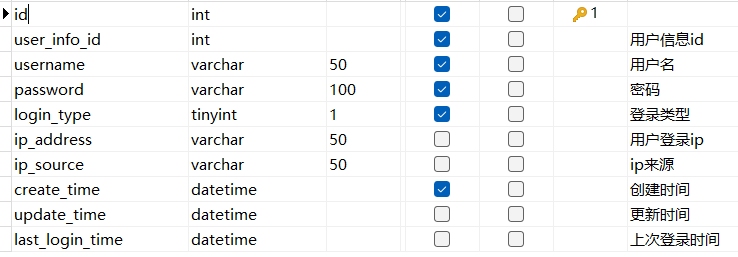


图2-9 用户注册信息表

照片表，如图2-10所示。存放管理员上传在网站上的图片信息，如相册id、照片名、照片描述、照片的存放链接、及创建时间等信息。

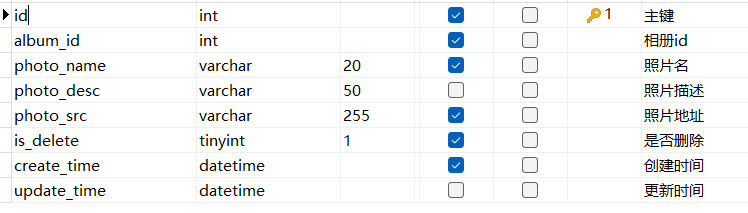


图2-10 照片表

标签表，如图2-11所示。存放标签信息，如标签id、标签名、创建时间和更新时间

图2-11 标签表

文章标签表，如图2-12所示。每个文章至少可以有一个标签，是一对多的关系，所以需要一个中间表来代替文章表存放多余的标签，这样在查找同一个文章时可以查到多个标签。

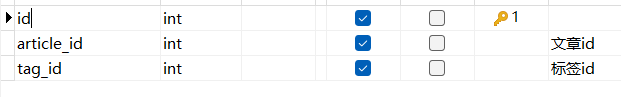


图2-12 文章标签表

用户角色表，如图2-13所示。用户有三种类型，也称其为角色，分别为管理员、测试用户、普通用户。每个用户的用户id的值完全不同，但角色id的值可以也只能为“1、2、3”的一种，分别对应三种角色，不同的角色可以在网站后台拥有不同的权限。



图2-13 用户角色表

1. 系统编码
   1. 登陆和注册功能

（1） 前端中导入axios请求库。

import axios from "axios";

import VueAxios from "vue-axios";

创建一个登陆实例。

const routes = [

{

path: "/login",

name: "登录",

hidden: true,

component: () => import("../views/login/Login.vue")

}

];

向后端发送登录请求，后端处理后返回数据，如图3-1、3-2。

let param = new URLSearchParams();

param.append("username", that.loginForm.username);

param.append("password", that.loginForm.password);

that.axios.post("/api/login", param).then(({ data }) => {

if (data.flag) {

// 登录后保存用户信息

that.$store.commit("login", data.data);

// 加载用户菜单

generaMenu();

that.$message.success("登录成功");

that.$router.push({ path: "/" });

} else {

that.$message.error(data.message);

}

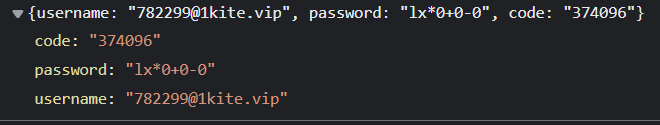
 });

图3-1 发送数据

图3-2 返回数据

（2） UserAuthController.java控制类。处理用户登陆或者注册的请求。使用userAuthService对象从数据库中找寻数据，然后调用UserAuthServiceImpl中的register方法对数据进行处理，也就是判断邮箱是否被注册，如果没有则新建用户信息、绑定用户角色、新增用户账号。

/\*\*

\* 用户注册

\*

\* @param user 用户信息

\* @return {@link Result<>}

\*/

@ApiOperation(value = "用户注册")

@PostMapping("/register")

public Result<?> register(@Valid @RequestBody UserVO user) {

userAuthService.register(user);

return Result.ok();

}

UserAuthServiceImpl.java数据服务实现类。在此处判断邮箱是否被注册，进而创建用户。

@Transactional(rollbackFor = Exception.class)

@Override

public void register(UserVO user) {

// 校验账号是否合法

if (checkUser(user)) {

throw new BizException("邮箱已被注册！");

}

// 新增用户信息

UserInfo userInfo = UserInfo.builder()

.email(user.getUsername())

.nickname(CommonConst.DEFAULT\_NICKNAME + IdWorker.getId())

.avatar(blogInfoService.getWebsiteConfig().getUserAvatar())

.build();

userInfoDao.insert(userInfo);

// 绑定用户角色

UserRole userRole = UserRole.builder()

.userId(userInfo.getId())

.roleId(RoleEnum.USER.getRoleId())

.build();

userRoleDao.insert(userRole);

// 新增用户账号

UserAuth userAuth = UserAuth.builder()

.userInfoId(userInfo.getId())

.username(user.getUsername())

.password(BCrypt.hashpw(user.getPassword(), BCrypt.gensalt()))

.loginType(LoginTypeEnum.EMAIL.getType())

.build();

userAuthDao.insert(userAuth);

}

* 1. 验证码功能

（1） 在pom.xml中引入mail依赖

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-mail</artifactId>

</dependency>

在application.yml文件中配置邮箱邮箱主机名和邮箱授权码，

# 邮箱配置

mail:

host: smtp.qq.com

username: 1274872144@qq.com #邮箱用户名

password: szvvqocdfmwebabh #邮箱授权码

default-encoding: UTF-8

port: 587

（2） 使用EmailDTO类发送包含验证码的邮件。

@Override

public void sendCode(String username) {

// 校验账号是否合法

if (!checkEmail(username)) {

throw new BizException("请输入正确邮箱");

}

// 生成六位随机验证码发送

String code = getRandomCode();

// 发送验证码

EmailDTO emailDTO = EmailDTO.builder()

.email(username)

.subject("验证码")

.content("您的验证码为 " + code + " 有效期15分钟，请不要告诉他人哦！")

.build();

rabbitTemplate.convertAndSend(EMAIL\_EXCHANGE, "\*", new Message(JSON.toJSONBytes(emailDTO), new MessageProperties()));

// 将验证码存入redis，设置过期时间为15分钟

redisService.set(USER\_CODE\_KEY + username, code, CODE\_EXPIRE\_TIME);

}

* 1. 文章添加功能

（1） 前端创建文章实例：

import \* as imageConversion from "image-conversion";

export default {

created() {

const path = this.$route.path;

const arr = path.split("/");

const articleId = arr[2];

if (articleId) {

this.axios.get("/api/admin/articles/" + articleId).then(({ data }) => {

this.article = data.data;

});

} else {

const article = sessionStorage.getItem("article");

if (article) {

this.article = JSON.parse(article);

}

}

},

openModel() {

if (this.article.articleTitle.trim() == "") {

this.$message.error("文章标题不能为空");

return false;

}

if (this.article.articleContent.trim() == "") {

this.$message.error("文章内容不能为空");

return false;

（2） 调用ArticleDao.xml在对数据库执行SQL语句，添加文章内容。

<select id="listArticlesByCondition" resultMap="articlePreviewResultMap">

SELECT

a.id,

article\_cover,

article\_title,

a.create\_time,

a.category\_id,

category\_name,

t.id AS tag\_id,

t.tag\_name

FROM

(

SELECT

id,

article\_cover,

article\_title,

article\_content,

create\_time,

category\_id

FROM

tb\_article

<where>

<if test="condition.categoryId != null">

category\_id = #{condition.categoryId}

</if>

<if test="condition.tagId != null">

id IN (

SELECT

article\_id

FROM

tb\_article\_tag

WHERE

tag\_id = #{condition.tagId})

</if>

</where>

AND is\_delete = 0

AND status = 1

ORDER BY id DESC

LIMIT #{current},#{size}

) a

JOIN tb\_category c ON a.category\_id = c.id

JOIN tb\_article\_tag atg ON a.id = atg.article\_id

JOIN tb\_tag t ON t.id = atg.tag\_id

</select>

* 1. 文章搜索功能

（1） 后端ArticleServiceImpl.java数据服务实现类

@Override

PublicArticlePreviewListDTO listArticlesByCondition(ConditionVO condition) {

// 查询文章

List<ArticlePreviewDTO> articlePreviewDTOList = articleDao.listArticlesByCondition(PageUtils.getLimitCurrent(), PageUtils.getSize(), condition);

// 搜索条件对应名(标签或分类名)

String name;

if (Objects.nonNull(condition.getCategoryId())) {

name = categoryDao.selectOne(new LambdaQueryWrapper<Category>()

.select(Category::getCategoryName)

.eq(Category::getId, condition.getCategoryId()))

.getCategoryName();

} else {

name = tagService.getOne(new LambdaQueryWrapper<Tag>()

.select(Tag::getTagName)

.eq(Tag::getId, condition.getTagId()))

.getTagName();

}

return ArticlePreviewListDTO.builder()

.articlePreviewDTOList(articlePreviewDTOList)

.name(name)

.build();

}

* 1. 评论功能

（1） 查询回复用户邮箱号：

Integer userId = BLOGGER\_ID;

String id = Objects.nonNull(comment.getTopicId()) ? comment.getTopicId().toString() : "";

if (Objects.nonNull(comment.getReplyUserId())) {

userId = comment.getReplyUserId();

} else {

switch (Objects.requireNonNull(getCommentEnum(comment.getType()))) {

case ARTICLE:

userId = articleDao.selectById(comment.getTopicId()).getUserId();

break;

case TALK:

userId = talkDao.selectById(comment.getTopicId()).getUserId();

break;

default:

break;

}

}

（2） 发送邮件，如图3-1所示：

String email = userInfoDao.selectById(userId).getEmail();

if (StringUtils.isNotBlank(email)) {

EmailDTO emailDTO = new EmailDTO();

if (comment.getIsReview().equals(TRUE)) {

// 评论提醒

emailDTO.setEmail(email);

emailDTO.setSubject("评论提醒");

// 获取评论路径

String url = websiteUrl + getCommentPath(comment.getType()) + id;

emailDTO.setContent("您收到了一条新的回复，请前往" + url + "\n页面查看");

} else {

// 管理员审核提醒

String adminEmail = userInfoDao.selectById(BLOGGER\_ID).getEmail();

emailDTO.setEmail(adminEmail);

emailDTO.setSubject("审核提醒");

emailDTO.setContent("您收到了一条新的回复，请前往后台管理页面审核");

}

rabbitTemplate.convertAndSend(EMAIL\_EXCHANGE, "\*", new Message(JSON.toJSONBytes(emailDTO), new MessageProperties()));

}



图3-1 发送邮件

1. 系统功能调试
   1. 开发环境

硬件资源：CPU为i5-9300H，内存8GB，无公网ip。

软件资源：操作系统：windows11。数据库：mysql。开发工具：IDEA。浏览器：Microsoft Egde。

* 1. 调试环境

硬件资源：内存1GB，公网ip：150.230.40.181，网络带宽：0.48。

软件资源：操作系统：CentOS-7。数据库：mysql。测试工具：Docker。浏览器：Microsoft Egde

* 1. 功能测试

以下测试均采用黑盒测试

首页功能无误，如图4-1。

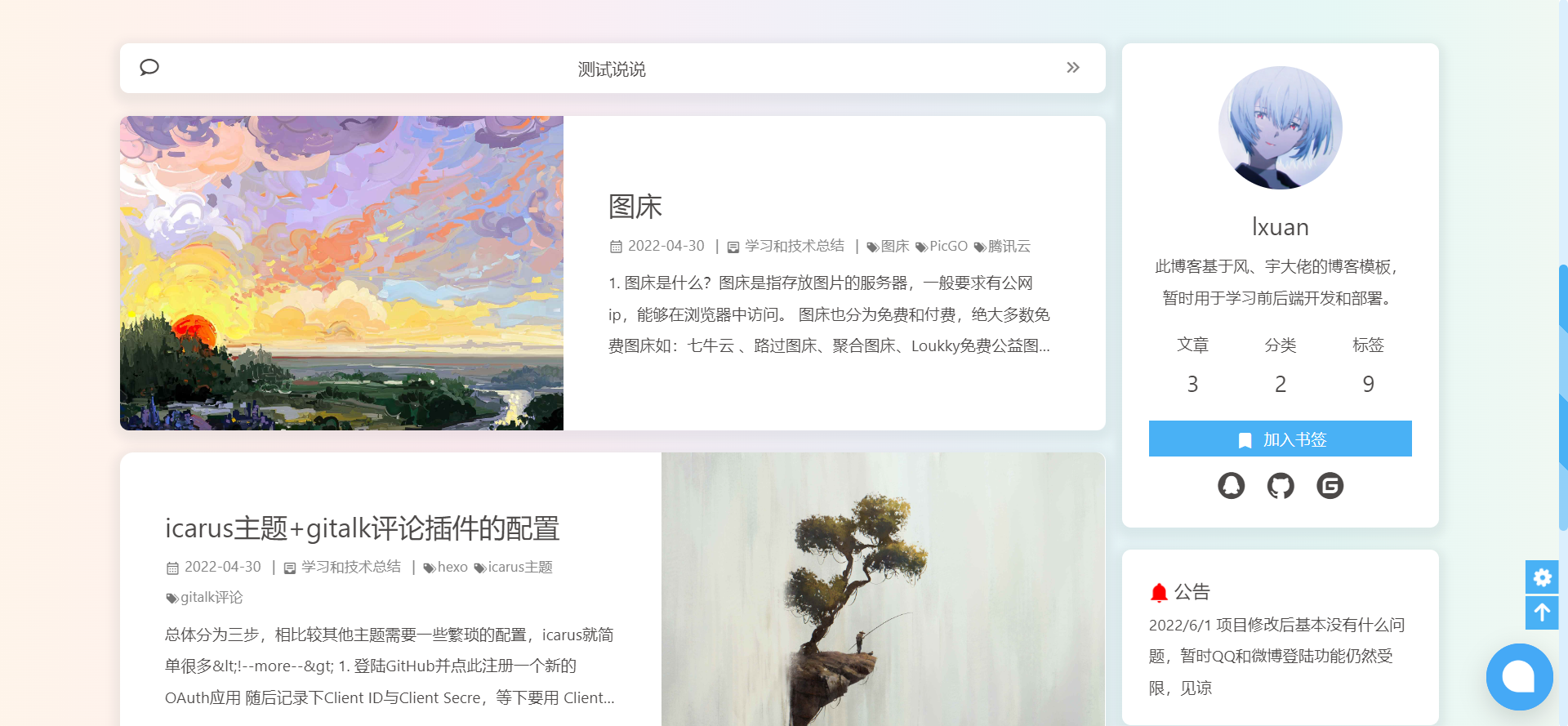


图4-1 首页测试

文章搜索功能无误，如图4-2。

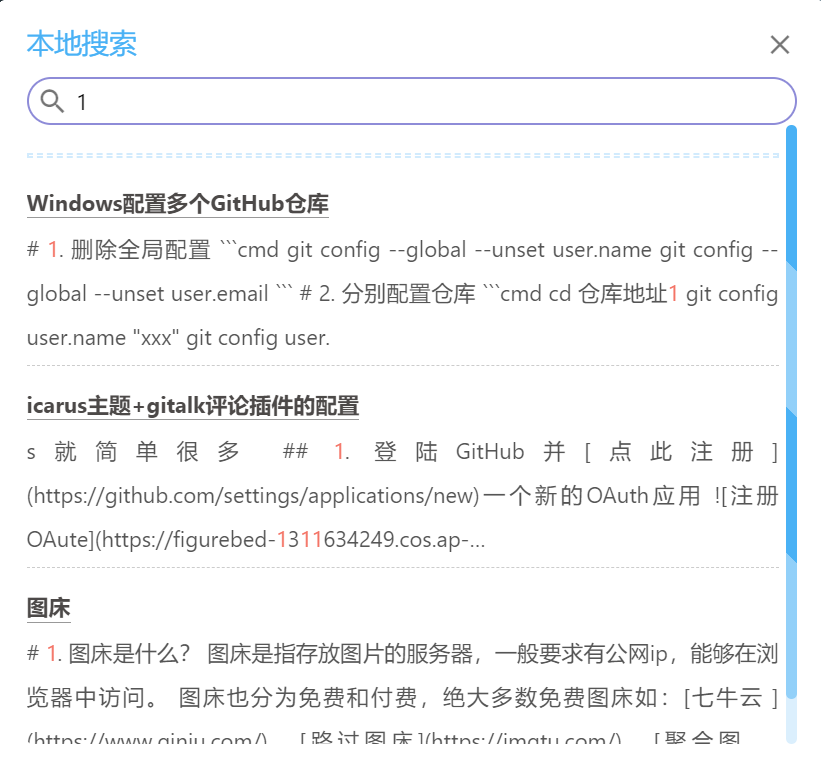


图4-2 文章搜索

文章显示功能无误，如图4-3。

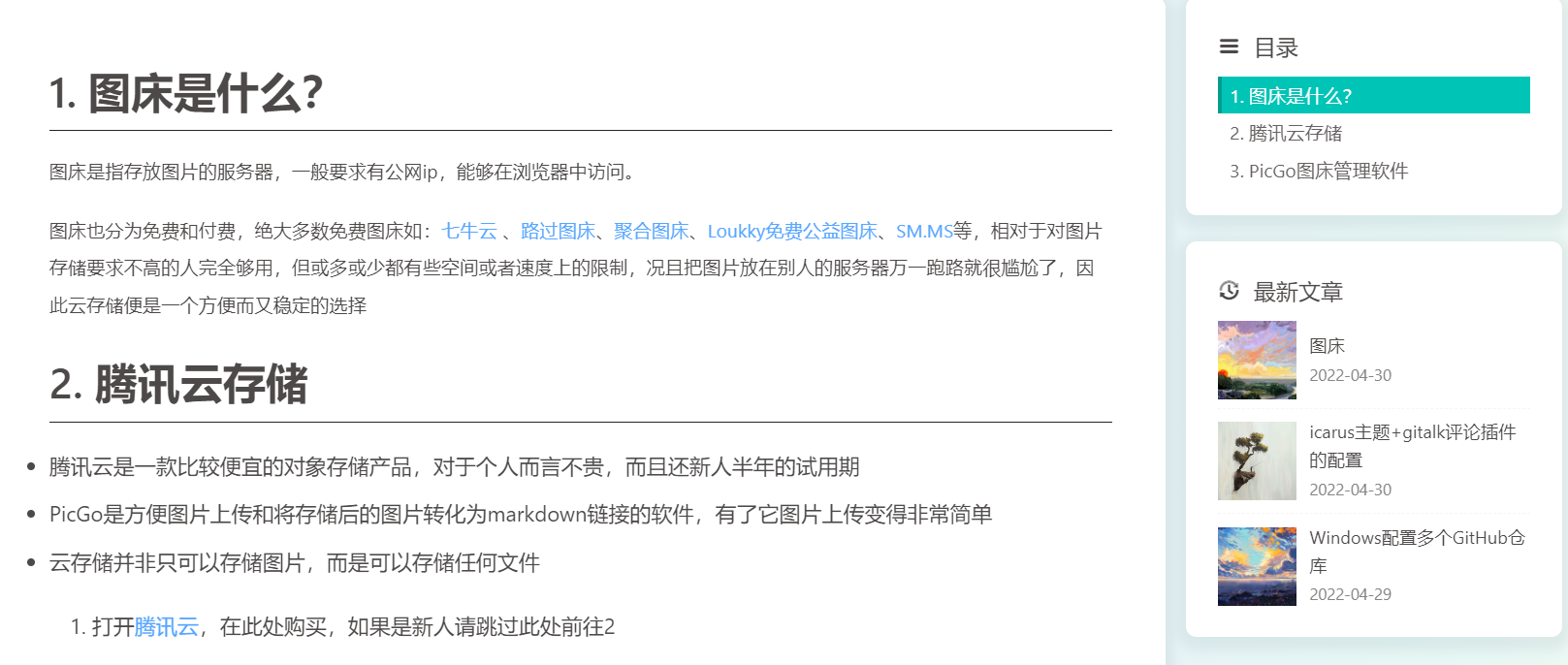


图4-3 文章显示

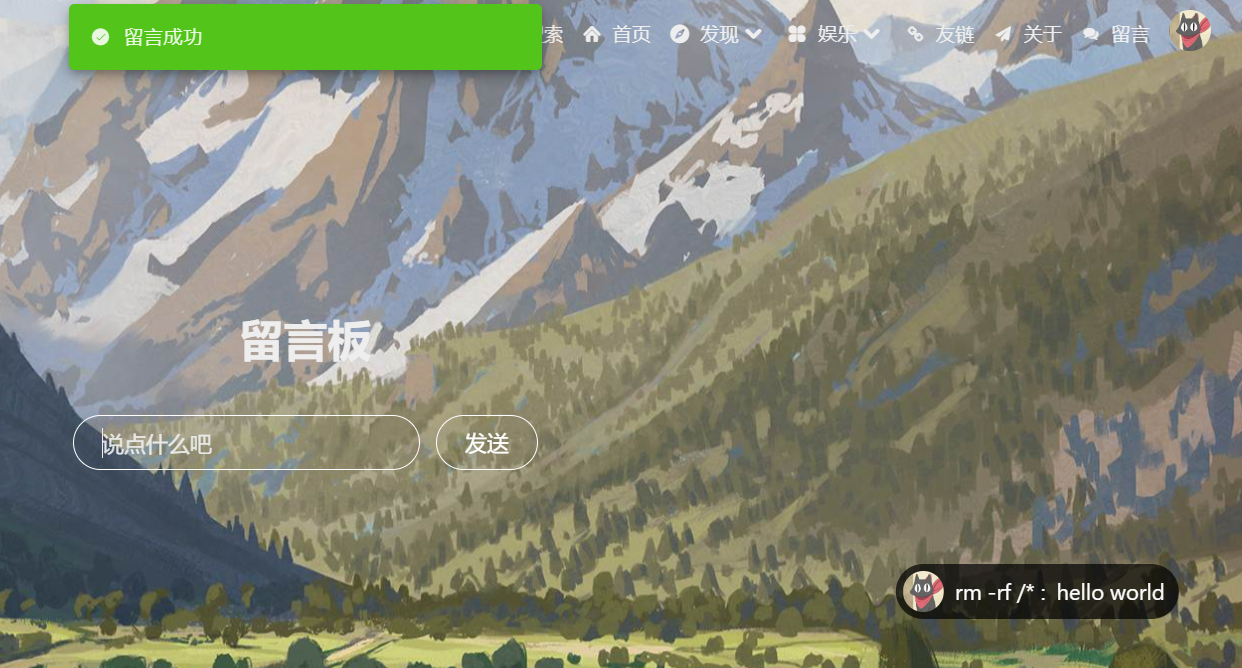
留言功能无误，如图4-4。

图4-4 留言板

后台登陆功能无误，如图4-5。



图4-5 后台首页

文章添加功能无误，如图4-6。



图4-6 文章添加功能

评论管理功能无误，如图4-7。

图4-7 评论管理

用户管理功能无误，如图4-8。



图4-8 用户管理

密码修改功能无误，如图4-9。

图4-9 密码修改

1. 参考文献

[1] 罗路腾，王贵鑫 . 基于 Springboot 的博客网站的设计与 实现 [J]. 科学技术创新，2019（33）：64-66.

[2] 李孟津，杨丹 . 基于 SpringBoot 的在线招聘网站的设计 与实现 [J]. 科学技术创新，2020（26）：98-99.

[3] 熊永平 . 基于 SpringBoot 框架应用开发技术的分析与研 究 [J]. 电脑知识与技术，2019，15（36）：76-77.

[4] 杨伟凡 . 基于 Java 技术平台的在线考试系统的设计与实 现 [J]. 卫星电视与宽带多媒体，2020（3）：99-100.

[5] 王丹，孙晓宇，杨路斌，等 . 基于 SpringBoot 的软件统 计分析系统设计与实现 [J]. 软件工程，2019，22（3）：40-42.

[6] 余思源，张伟 . 基于 JAVA 的个人博客系统的设计与实现 [J]. 电脑知识与技术，2018，14（17）：135-137.

[7] 邓笑 . 基于 Spring Boot 的校园轻博客系统的设计与实现 [D]. 武汉：华中科技大学，2018.

[8] 唐炜 .Spring Data、MongoDB、Thymeleaf 的数据持久 化方案及分页技术实现 [J]. 陇东学院学报，2017，28（5）：9-13.

[9] 陈秋玲 . 基于 JavaWeb 的图书购物网站的设计与实现 [J]. 电脑知识与技术，2019，15（16）：36-38.

|  |  |
| --- | --- |
| **软件设计综合课程设计总结：**  软件工程专业本科生的培养目标是面向软件产业界对软件工程专业技术人才的需求，培养具备扎实的计算机科学技术基本知识，受过良好的软件设计、开发和工程管理方法基本训练的应用型高级技术人才。  Java程序设计是软件工程专业的一门重要课程，该门课程的课程设计作为实践教学的重要环节，对培养学生的学习能力、问题分析与解决能力、实践创新能力起着重要的作用，同时还可以培养学生的软件工程思想，提高学生的软件开发能力。  我此次课程设计中项目使用的是spring boot后端框架和vue前端框架。项目部署时注册只有一种邮箱注册方式，QQ注册或者微博注册，需要去腾讯应用开放平台申请回调地址，然后加入到前端页面中，后端只需要获取对应数据就可以和邮箱注册一样完成注册。  此次项目仍存在一点不足，服务器是国外的机器，带宽小，延迟高，添加文章图片的时候不能本地上传服务器，必须使用阿里云的对象存储服务（OSS）来存储图片。希望以后可以在项目中添加压缩图片的功能，减小服务器带宽压力，也能容纳更多人访问。 | |
| 指导教师评语：  指导教师(签字)：　　　　　　 年 月 日 | |
| 课程设计成绩 |  |