

# 华北水利水电大学

North China University of Water Resources and Electric Power

## 人机交互的软件工程方法

### 实验报告

2021-2022 学年

第一学期

学 院 信息工程学院

专 业 软件工程

班 级 2018201 班

指导教师 陈欢欢

小组成员 刘大禾 201820110

冯海龙 201820106

刘张澳 201820109

白梦醒 201820108

冯家豪 201820107

代向龙 201820105

---

## 目录

第一章 绪论.....	6
第二章 概述.....	7
2.1 项目背景.....	7
2.2 开发环境.....	8
2.3 技术介绍.....	8
2.4 功能介绍.....	8
第三章 相关技术介绍.....	9
3.1 B/S 简介.....	9
3.2 JAVA 简介.....	9
3.3 JSP 简介.....	10
3.4 SpringMVC 简介.....	10
第四章 可行性分析.....	11
4.1 用户需求分析.....	11
4.2 技术可行性分析.....	11
4.2 经济可行性分析.....	12
4.3 操作可行性.....	12
第五章 系统设计.....	13
5.1 系统总流程.....	13
5.2 博主用例.....	14
5.3 游客用例.....	15
5.4 E-R 图.....	17
5.5 系统表设计.....	19
第六章 系统实现.....	22
6.1 登录模块.....	22
6.2 博客管理模块.....	22
6.2.1 博客查询.....	23
6.2.2 博客新建.....	24
6.2.3 博客修改.....	25
6.2.4 博客删除.....	26
6.3 博客类别管理模块.....	27
6.3.1 添加博客类别.....	27
6.3.2 修改博客类别.....	28
6.3.3 删除博客类别.....	29
6.3.4 显示博客类别.....	30
6.4 评论管理模块.....	31
6.4.1 审核评论.....	31
6.4.2 删除评论.....	32
第七章 界面展示以及人机交互体现.....	33
7.1 后台界面.....	33

---

7.2 文章管理.....	33
7.3 页面子界面.....	34
7.4 链接子界面.....	35
7.5 公告页面.....	35
7.6 评论页面.....	36
7.7 菜单页面.....	37
7.8 前台页面--修改前.....	37
7.10 人机交互目标体现.....	39
7.10.1 可用性目标: .....	39
7.10.2 高效率目标: 采用 markdown 语法格式.....	40
7.10.3 主观满意度: 代码框显示代码.....	40
第八章 调试过程中的问题及系统测试情况.....	41
8.1 调试过程中问题.....	41
8.2 系统测试情况.....	41
8.2.1 前台模块测试.....	41
8.2.2 后台模块测试.....	42
第九章 项目创新部分.....	44
9.1 想法由来.....	44
9.2 实现简介.....	44
9.3 编辑器使用 Markdown.....	45
9.4 使用 OpenTag.....	45
参考文献.....	46
致谢.....	47
附: 组内分工情况.....	48

---

## 摘 要

随着互联网的发展，博客在自由和共享的精神的影响下应运而生。博客的出现让我们有更多机会在日常生活中向他人展示自己的观点和信息。博客就是开放的私人空间。在互联网中，你可以尽情的发表自己的思想和言论。可以在网络上结交各式各样的人。

互联网在大家的方方面面都提供了极大的帮助。所以很明显的趋势就是网站的出现。个人博客系统为树立个人在网络上的形象提供了一个良好的平台。

JAVA 作为目前最盛行的技术之一，J2EE（JAVA 企业版本）十分适合于 Web 应用程序的设计和实现。此外，目前基于 J2EE 的开源框架在市场上久经考验，java 社区也是十分活跃。本文的目的就是利用 J2EE 技术设计和实现个人博客系统。

关键词：博客系统；B/S 结构；SpringMVC；Spring；MyBatis

---

## ABSTRACT

blog comes into being under the influence of the spirit of freedom and sharing. The advent of blogs gives us more opportunities to present our views and information to others in our daily lives. Blog is open private space. On the Internet, you can express your thoughts and opinions. You can make all kinds of people on the internet.

The Internet has provided great help in all aspects of our. So the obvious trend is the emergence of the site. Personal blog system provides a good platform for setting up the image of the individual on the network.

JAVA is currently the most popular technology, J2EE (JAVA enterprise version) is very suitable for the design and implementation of Web applications. In addition, the current open source framework based on J2EE has been tested on the market, Java community is also very active. The purpose of this paper is to use J2EE technology to design and implement personal blog system.

**Key words:** Blog System; B / S structure; SpringMVC; Spring ; MyBatis

---

## 第一章 绪论

博客是人们在网络上展现自己个性的一种方式。自从互联网走进千家万户，大受人们喜爱。大家都喜欢通过网络来获取各种各样的信息。而在网络上，你也可以畅所欲言，而博客也是其中一种。在博客上，你可以发表自己的心路旅程，发表自己的工作经验，发表技术博客等等。

首先介绍一下什么是博客，它也被称为网络日志。博客其实就是一个网页，它是由各种各样的帖子组成。这些帖子可以是你某一天观看电影的感慨，也可以是阅读到某篇书籍的读后感，更可以是你在自己技术领域的心得。它就是一个网页，但是大家可以在上面用文字或者图片分享自己的心情等，它提供的内容可以用来进行交流。

博客最开始的时候是论坛下的一个子系统。最早参与博客的都是一些信息技术的一些专家和学者，讨论的都是专业技能。直到发现了博客的商业价值，一些开放了自发式的博客，大家也都发现了博客分享和讨论的趣味。越来越多的人参与进来，博客也越来越风靡盛行。

博客与论坛有许多的相似之处，许多朋友都分不清他们两的区别。首先，论坛和博客都可以进行交友和交流等。而其中论坛注重的是集体讨论，博客注重的是个人分享。它们两者的核心点就有着很大的区别。论坛的创建的用户是基于为众人服务，而博客是为了博主服务。它们的不同点还有在形式上的不同。博客是独立存在的。而论坛并不是。总结起来，论坛是多人一起交流的地方，互动性强，社交性强。博客是发表个人文章的地方，虽然也可以交流，但更多的还是自娱自乐。

---

## 第二章 概述

### 2.1 项目背景

进入二十一世纪,以Internet为核心的现代网络积水和通信技术已经得到了飞速的发展和广泛的应用,各种网络交流互动工具也应运而生。其中以论坛、博客、社区、空间最为受广大网民朋友的欢迎,也是目前为止发展的比较成熟的信息交流工具。在网络技术逐渐渗入社会生活各个层面的今天,传统的交流方式也面临着变革,而网络博客则是一个很重要的方向。基于Web技术的网络考试系统可以借助于遍布全球的Internet进行,因此交流既可以是本地进行,也可以是异地进行,大大拓展了沟通与交流的灵活性。博客在现如今这个飞速发展的网络时代已经成为人们不可或缺的一部分

博客,又译为网络日志、部落格或部落阁等,是一种通常由个人管理、不定期张贴新的文章的网站。博客它经常是由简短且经常更新的帖子构成,它可以发表有关个人构思,日记,或者诗歌,散文,小说等等。博客可以是你纯粹个人的想法和心得,包括你对时事新闻、国家大事的个人看法,或者你对一日三餐、服饰打扮的精心料理等,也可以是在基于某一主题的情况下或是在某一共同领域内由一群人集体创作的内容。写博客是为了把自己各种各样的想法在网上表达、释放出来,把一时的想法变成观点展示给大家。而浏览博客的用户,也可以对发表观点和看法的文章进行评论,博客就是这样一个平台。 博客最初的名称是Weblog,由web和log两个单词组成,按字面意思就为网络日记,后来喜欢新名词的人把这个词的发音故意改了一下,读成we blog,由此,blog这个词被创造出来。中文意思即网志或网络日志,不过,在中国大陆有人往往也将 Blog本身和 blogger(即博客作者)均音译为“博客”。“博客”有较深的涵义:“博”为“广博”;“客”不单是“blogger”更有“好客”之意。看Blog的人都是“客”。而在台湾,则分别音译成“部落格”(或“部落阁”)及“部落客”,认为Blog本身有社群群组的意含在内,借由Blog可以将网络上网友集结成一个大博客,成为另一个具有影响力的自由媒体。

随着我国经济的飞速发展,生活水平不断提高,越来越多的人选择用各种各样的方式记录下自己的生活点滴,跟朋友分享自己的生活乐趣,在这种背景下,各式各样的博客网站如同雨后春笋般出现在了人们的视野里。

我们小组打算开发一个简洁,方便的博客网站,用来让更多的人能在更加人性化,更加舒适的环境下分享自己的生活,分享自己的经验.在本项目中我们会使用各种各样人机交互学的知识,各种UI设计方面的知识,来对项目进行制作。

---

## 2.2 开发环境

操作系统	Window10
开发语言	Java
开发环境	IDEA
Jdk版本	1.8
数据库	Mysql

## 2.3 技术介绍

本项目是采用了springmvc, spring, mybatis框架进行设计与实现的。项目采用了jdk1.8的版本, web服务器采用tomcat 8.5, 数据库服务器采用了mysql, 前台页面的显示使用了JavaScript, CSS, jquery以及Layui前端框架等技术进行编写。使用java面向对象的思想对类和接口进行设计, 使网站具有可扩展性, 便于维护等优点。

## 2.4 功能介绍

本系统为用户提供实现了以下功能:

注册功能:

系统为多用户系统, 用户可通过注册获取用户名和密码。

登录功能:

用户注册后通过登录, 进入操作界面。用户名和密码, 通过服务器验证方可运行, 否则显示消息提示。

网站前台功能:

浏览: 博主个人信息、文章;

发表: 留言, 评论

网站后台功能:

文章管理模块: 可以发布文章, 查看文章, 编辑文章

博文分类与管理模块: 实现分类的增、删、改

公告管理: 实现公告的增删改查

用户管理模块: 实现用户的查询、编辑、删除功能

评论管理模块: 实现删除评论功能

系统管理模块: 实现博客版面的定义与管理、博客的数据统计



---

## 第三章 相关技术介绍

### 3.1 B/S 简介

软件系统体系结构分为两种，是客户机/服务器结构和浏览器/服务器结构。其中的浏览器/服务器结构就是B/S结构。

C/s模式：是客户端/服务器(Client/Server)模式，主要指的是传统的桌面级的应用程序。比如我们经常用的信息管理系统。

B/S模式：是浏览器/服务器(Browser/Server)模式，主要指的是web应用程序，就像电子商务网站，如淘宝，阿里巴巴等。相对于C/S模式的应用程序来说，B/S模式最大的优势在于客户端只要有浏览器就可以运行。而C/S模式的应用程序需要在客户端进行安装，而且升级也不太方便。而B/S模式的应用程序对于客户端来说，永远都是最新版本的。

### 3.2 JAVA 简介

Java的前身是Oak，sun公司在1995年申请注册商标的时候，发现已经Oak已经被注册了，最终才另外取了一个名字叫JAVA（其中还有一个趣味故事，有兴趣的可以去查找看看），

要使用Java首先要到官网甲骨文官网上下载Java的软件开发工具包，其中分为windows版本和Linux版本，我使用的是Windows的。而在Windows上要使用Java，要设置JAVA的环境变量。设置环境变量的步骤如下：第一步，点击计算机属性中的高级系统设置，点击系统属性高级的环境变量。第二步，设置一个系统变量为JAVA\_HOME, 其中的值是你的Java安装目录。它的作用还在于一些使用Java的工具时候需要用到这个变量，否则打不开，例如eclipse. 第三步，设置一个classpath系统变量，它的作用是在你写Java源文件导包的时候去查找的类路径。第三步，在Windows自有的系统变量path中加入Java安装目录下的bin路径。既可以写成%JAVA\_HOME%/bin;这就将Java的运行环境搭配好了。

介绍一下Java的文件目录。首先最重要的bin目录下面包含的是一些JDK包含的一些开发工具执行文件，例如像编译器javac.exe. 运行Java的java.exe等等都在这个目录下面。其中Java启动的虚拟机在Java目录的jre目录中。

---

### 3.3 JSP 简介

Jsp是服务端技术，与微软的ASP一样，是一种动态网页的技术，不过jsp是使用Java语言技术开发的技术，jsp使用的时候会被jsp引擎解析成Servlet代码，而jsp虽然可以写Java代码，但是jsp主要的作用是为了展示视图，所以jsp+javaBean 可以很好的解决代码混淆问题，然后将视图与逻辑分离。

### 3.4 SpringMVC 简介

SpringMVC是一种现今十分流行的WEB框架，它的作用是使用MVC思想设计模式对web层进行解耦。是用来接受请求和处理请求的。

SpringMVC处理请求的流程是如下：首先用户从浏览器中发送请求；之后它SpringMVC的核心组件DispatcherServlet 前端控制器根据请求信息也即是url的信息来选择一个页面控制器进行处理；页面控制器接受得到委托之后，它首先会收集和绑定参数到某个对象中，然后调用业务对象进行业务逻辑代码处理，处理结束后返回ModelAndView。前端控制器根据视图的名字渲染相应的视图然后将响应传到前台去。

---

## 第四章 可行性分析

### 4.1 用户需求分析

在对用户进行调查后,初步得出一部分用户的需求,就目前来看,用户所需求的必要功能包括但不限于:书写博客,博客浏览,博客评论,私信留言,用户所觉得最好拥有的功能包括但不限于:查询自己的访问记录,查询他人对自己的访问记录,查询相关的公开博客,博客可以进行相对的上锁,能批量对博客进行管理,有黑名单功能.

在对用户喜好进行调查的过程中,初步得到了用户对UI界面的一些需求:没有广告,能清楚的找到相应的功能,博客页面清晰,博客内容占据页面的主要部分,在博客内容界面可以直接进行各种跳转,想要对页面进行自定义设计.

用户在性能方面要求主要为:能够在2S内打开博文,回复时尽量不重复出现卡顿,历史记录能保存至少一个月或1000条,有一定的保存空间.

### 4.2 技术可行性分析

本系统管理项目使用的是maven,数据库连接使用的是mybatis,数据库使用的是mysql,接受浏览器请求并处理的是使用springmvc,他们都是经过市场检验的,都已经十分成熟了,在安全性、可用性、可靠性等方面都是可以值得肯定的.

使用JAVA作为开发语言,MySQL作为数据库。MySQL拥有灵活、易维护、使用方便、快捷的特点,在小型项目开发中有着广泛的实际应用。JAVA作为面向对象的开发语言在这种存在大量学生、教师个体的项目中也是一个十分合适的开发语言,同时JAVA还有着JDBC与后台数据库进行连接,能简单的实现增删改查的功能。同时JAVA语言也能跟HTML和JSP进行十分良好的对接,不会存在大量的接口问题,同时我们也可以使用比较多的框架来对代码进行精简,这在实践中是可以实现的一件事.

同时目前HTML5中有着大量的官方UI,并且HTML的UI并非特别难以设计和编写,在UI设计方面将设计中的UI具现到网页上并非特别困难的问题。

---

## 4.2 经济可行性分析

本系统使用的技术框架都是开源框架，比如spring，springmvc,mybatis.而使用的数据库也是开源免费的mysql,使用的开发软件idea对学生是免费的。所以在经济上消费很低，在经济可行性分析中是可以确定可行的。

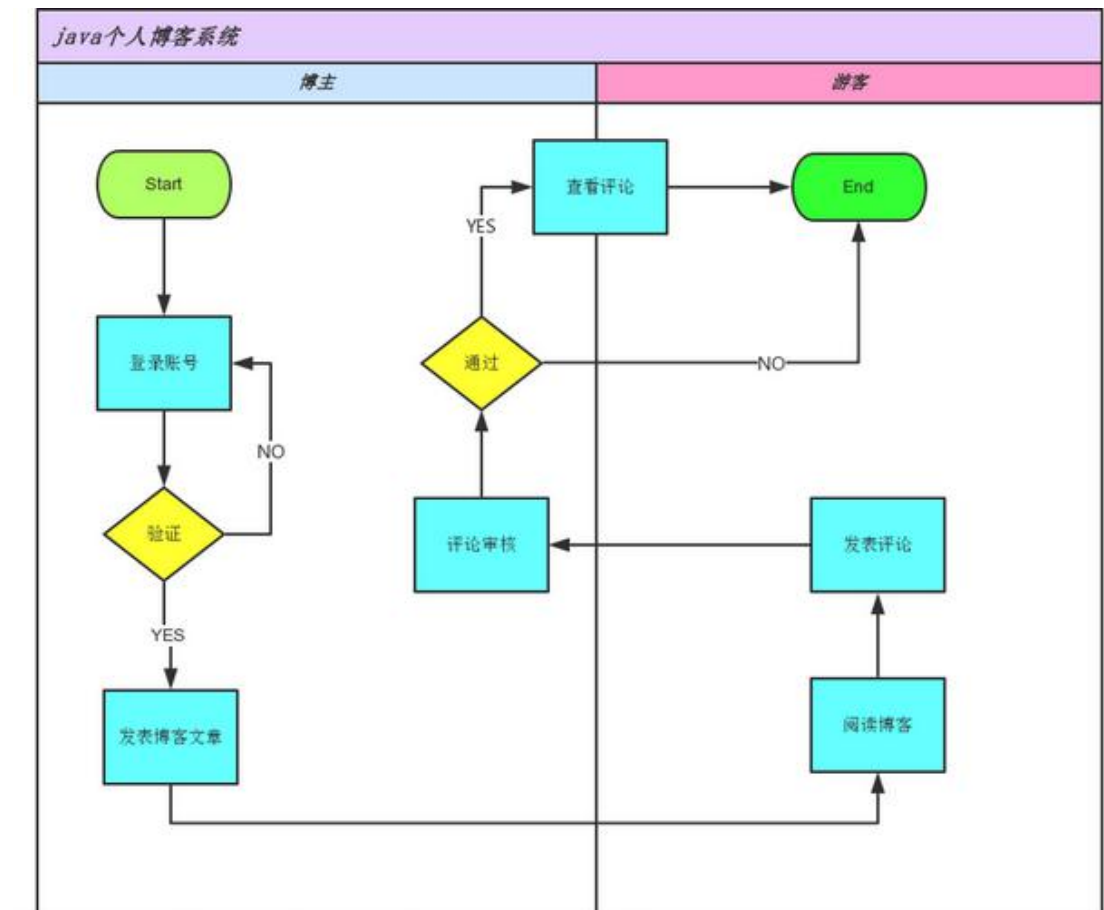
## 4.3 操作可行性

目前，绝大多数的个人电脑都能进行网络的链接和使用浏览器，而目前市面上的大部分浏览器也已经支持HTML5，包括但不限于chrome内核和IE内核的各种浏览器，在用户的操作上不存在特别难的操作空间，用户现在对网络上各种博客的应用也有相对的知识，不会存在特别困难的操作难度。本项目的页面使用相对而言不会让用户产生困难，而且也会有些导航功能让用户熟悉项目的功能。

博客系统开发结合个人需求而开发，能解决用户可以通过互联网来展现自己等诸多问题，因此该项目符合开发条件，具有成熟的基础，并且，从前面的分析来看，技术上的操作是十分成熟并且开源免费广为人们使用，从系统的操作上是可行的。

## 第五章 系统设计

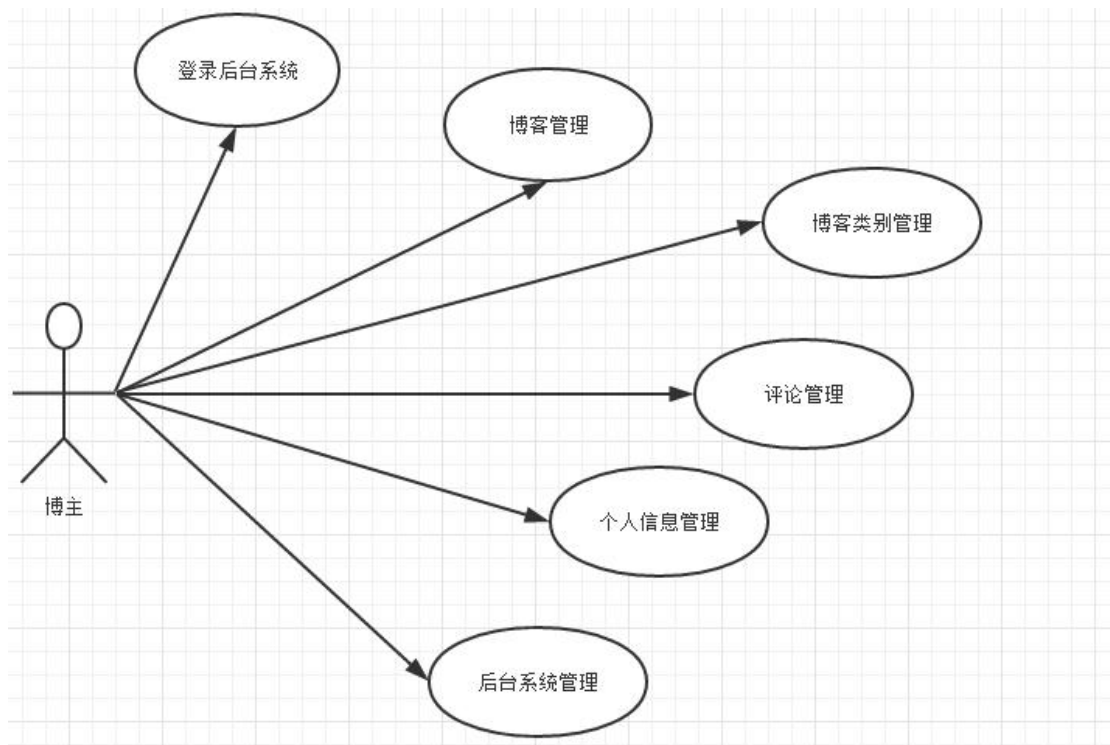
### 5.1 系统总流程



本个人博客系统分为两个角色，分别是博主和游客。只有博主才能登录后台管理系统，进行博客发表。

1. 首先要通过用户名和密码登录后台管理系统，才能发布博客。
2. 博主和游客都可以阅读博客。
3. 博主和游客都可以发表评论。
4. 经过博主的评论审核，评论才可以发布。
5. 此时博主和游客才可以观看到评论。

## 5.2 博主用例



博主的功能简介：

### 1. 登入功能

登入后台管理系统：首先进入登录页面，需要输入账号和密码。它会使用Shiro进行安全管理，对前台输入的密码进行加密运算，然后与数据库中的进行比较。成功后才能登入后台系统。

### 博客管理功能

博客管理功能分为写博客和博客信息管理。写博客是博主用来发表编写博客的，需要博客标题，然后选择博客类型，最后将博客内容填入编辑器中，点击发布博客按钮即可发布博客。

### 博客类别管理系统

博主类别管理系统可以添加，修改和删除博客类型名称和排序序号。将会显示到首页的按日志类别区域。游客可以从这里查找相关的感兴趣的博客内容。

### 4. 评论管理功能

评论管理功能分为评论审核和评论信息管理两部分。评论审核是当有游客或自己发表了评论之后，博主需要在后台管理系统中审核评论。若想将此评论显示在页面上则点击审核通过，否则点击审核不通过。

### 5. 个人信息管理功能

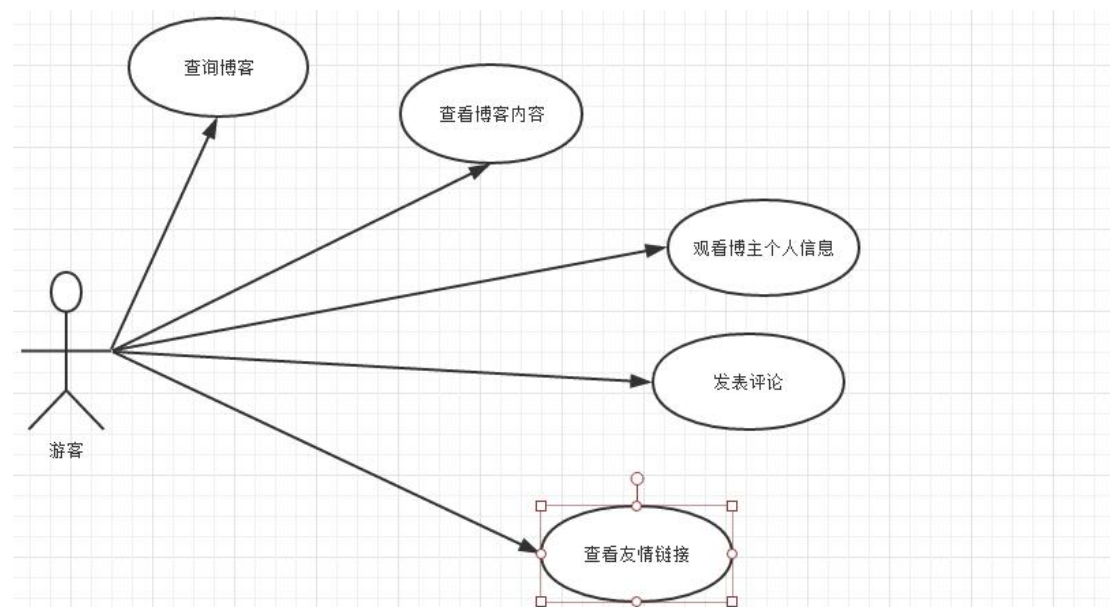
在这里可以修改博主的个人信息，可以修改昵称，个性签名，可以添加个人头像，修改个人简介。

#### 6. 系统管理功能

这里的功能有友情链接管理，修改密码，刷新系统缓存和安全退出。

友情链接管理可以添加，修改，删除友情链接网址。

### 5.3 游客用例



#### 用例一

用例名称：查询博客

参与者：游客

主要流程：  
在查询搜索处填写所需的条件  
点击查询  
便会显示出符合条件的所有博客

#### 用例二

用例名称：查看博客内容

参与者：游客

主要流程：  
点击想要查看的博客，进入博客内容页面

---

观看博客内容信息
----------

用例三
-----

用例名称：查看博主个人信息
---------------

参与者：游客
--------

主要流程： 点击关于博主，进入博主个人信息页面 可以看到博主的个人信息内容
---

用例四
-----

用例名称：发表评论
-----------

参与者：游客
--------

前置条件：首先要进入到博客内容页面
-------------------

主要流程： 在发表评论的评论框中写入自己想要发表的评论 填入验证码 点击发表评论
---

用例五
-----

用例名称：查看友情链接
-------------

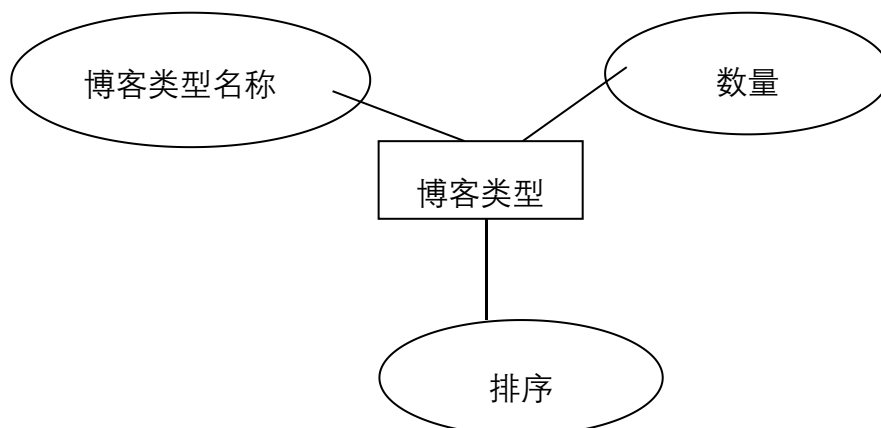
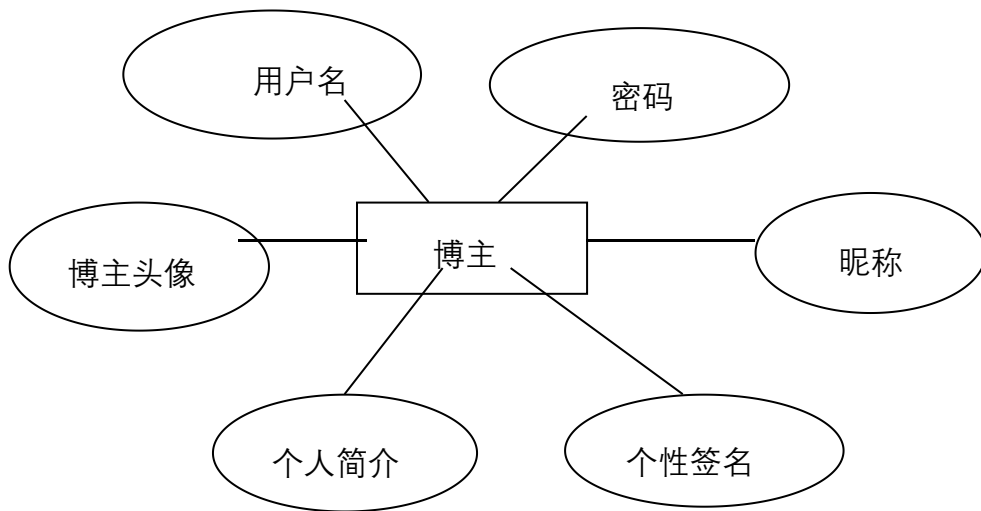
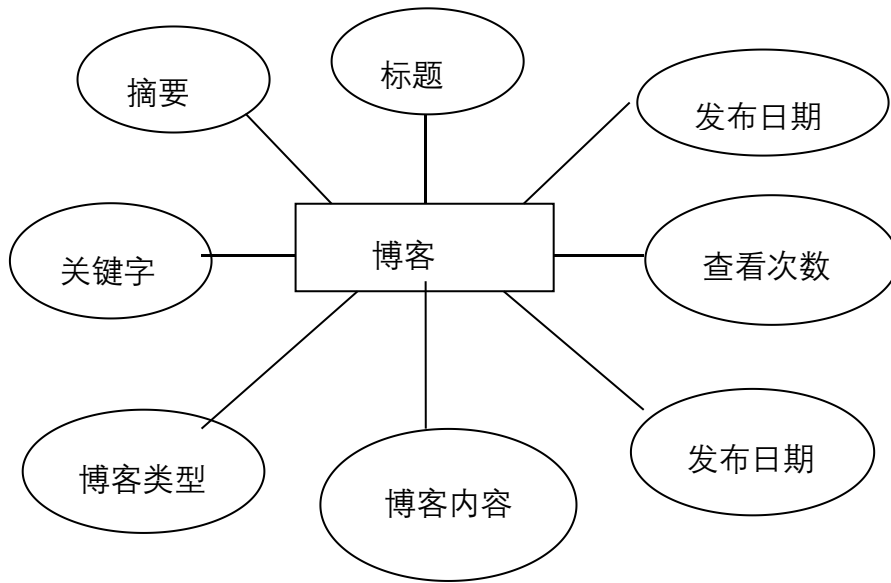
参与者：游客
--------

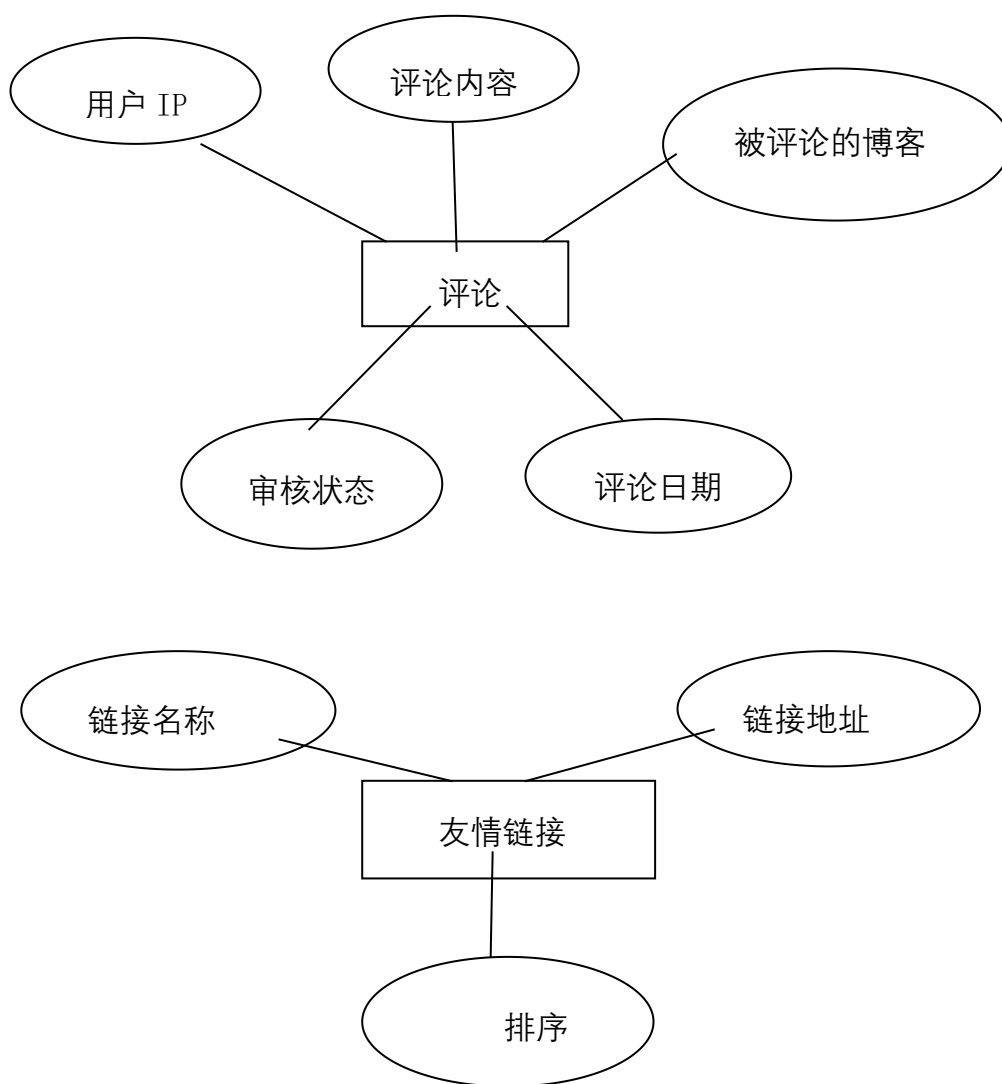
主要流程： 在每个页面的右下角可以看到友情链接 若想跳转到某个友情链接则点击相应的友情链接即可 观看友情链接内容
---



---

## 5.4 E-R 图





## 5.5 系统表设计

article表:

[illegible]

### article\_tag\_ref表:

[illegible]

category表:

[illegible]

comment表:

列名	数据类型	长度	默认	主键?	非空?	Unsigned	自增?	Zerofill?	更新	注释	虚拟	表
comment_id	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	<input type="button" value="表"/>
comment_pid	int	11	0								(n...	<input type="button" value="表"/>
comment_pname	varchar	255				<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	<input type="button" value="表"/>
comment_article_id	int	11				<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	<input type="button" value="表"/>
comment_author_name	varchar	50				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	<input type="button" value="表"/>
comment_author_email	varchar	50				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	<input type="button" value="表"/>
comment_author_url	varchar	50				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	<input type="button" value="表"/>
comment_author_avatar	varchar	100				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	<input type="button" value="表"/>
comment_content	varchar	1000				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	<input type="button" value="表"/>
comment_agent	varchar	200				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	<input type="button" value="表"/>
comment_ip	varchar	50				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	<input type="button" value="表"/>
comment_create_time	datetime					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	<input type="button" value="表"/>
comment_role	int	1				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	<input type="button" value="表"/>
												<input type="button" value="表"/>

link表:

列名	数据类型	长度	默认	主键?	非空?	Unsigned	自增?	Zerofill?	更新	注释	虚拟
link_id	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...
link_url	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...
link_name	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...
link_image	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...
link_description	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...
link_owner_nick	varchar	40		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...
link_owner_contact	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...
link_update_time	datetime			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...
link_create_time	datetime			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...
link_order	int	2	1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...
link_status	int	1	1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...

menu表:

列名	数据类型	长度	默认	主键?	非空?	Unsigned	自增?	Zerofill?	更新	注释	虚拟	表达
menu_id	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
menu_name	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
menu_url	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
menu_level	int	11		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
menu_icon	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
menu_order	int	11		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	

notice表:

列名	数据类型	长度	默认	主键?	非空?	Unsigned	自增?	Zerofill?	更新	注释	虚拟	表达
notice_id	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
notice_title	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
notice_content	varchar	10000		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
notice_create_time	datetime			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
notice_update_time	datetime			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
notice_status	int	1	1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
notice_order	int	2		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	

options表:

列名	数据类型	长度	默认	主键?	非空?	Unsigned	自增?	Zerofill?	更新	注释	虚拟	表达
option_id	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
option_site_title	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
option_site_desc	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
option_meta_desc	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
option_meta_keywo	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
option_aboutsite_	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
option_aboutsite_	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
option_aboutsite_	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
option_aboutsite_	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
option_aboutsite_	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
option_aboutsite_	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
option_aboutsite_	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
option_status	int	1	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	

page表:

列名	数据类型	长度	默认	主键?	非空?	Unsigned	自增?	Zerofill?	更新	注释	虚拟	表达
page_id	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
page_key	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
page_title	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
page_content	mediumtext			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
page_create_time	datetime			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
page_update_time	datetime			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
page_view_count	int	10	0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
page_comment_count	int	5	0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
page_status	int	1	1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	

tag表:

列名	数据类型	长度	默认	主键?	非空?	Unsigned	自增?	Zerofill?	更新	注释	虚拟	表达
tag_id	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
tag_name	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
tag_description	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	

user表:

<input type="checkbox"/> 列名	数据类型	长度	默认	主键?	非空?	Unsigned	自增?	Zerofill?	更新	注释	虚拟	表名
<input type="checkbox"/> user_id	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
<input type="checkbox"/> user_name	varchar	255	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
<input type="checkbox"/> user_pass	varchar	255	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
<input type="checkbox"/> user_nickname	varchar	255	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
<input type="checkbox"/> user_email	varchar	100	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
<input type="checkbox"/> user_url	varchar	100	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
<input type="checkbox"/> user_avatar	varchar	255		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
<input type="checkbox"/> user_last_login_ip	varchar	255		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
<input type="checkbox"/> user_register_time	datetime			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
<input type="checkbox"/> user_last_login_time	datetime			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	
<input type="checkbox"/> user_status	int	1	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	状态	(n...	
<input type="checkbox"/> user_role	int	1	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	用户权限	(n...	
<input type="checkbox"/>				<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(n...	

---

## 第六章 系统实现

因功能较多，只展示部分功能。

为防止插入代码引起文章排版乱，此处不再复制代码，代码打包进压缩包，此处只写功能的实现描述。

### 6.1 登录模块

登入系统后台管理登录页面，博主首先输入用户名和密码，它回去调用 Controller 层代码，然后进入业务层调用数据库的一些操作确认是否用户名密码正确，然后返回到前台就会登进去。

### 6.2 博客管理模块

博客管理模块为个人博客系统的用户（即博主）提供写博客和博客信息管理功能。在个人博客系统的首页上的博客就是从这里进行发布的。博客管理包含

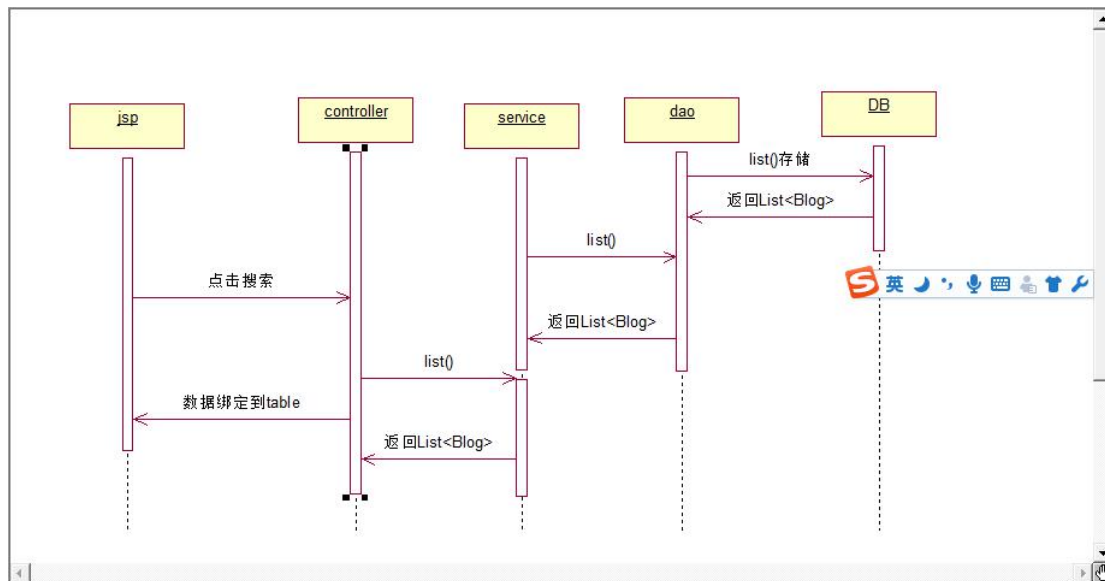
**新建：**对博客的新建，博客可以含有图片，视频，音频附件。新建博客必须要有博客标题，博客类别自己选择所需要的博客类别，然后填入博客内容，最后发表文章。

**搜索：**在博客信息管理中，可以输入自己想要搜索的博客信息，系统会自动筛选出适合的博客展现给用户。

**修改：** 点击你想要修改的博客，系统会弹出修改博客页面，之后博客的标题，所属的博客类型，博客内容等都可以修改。

**删除：** 该系统支持单个删除和批量删除。

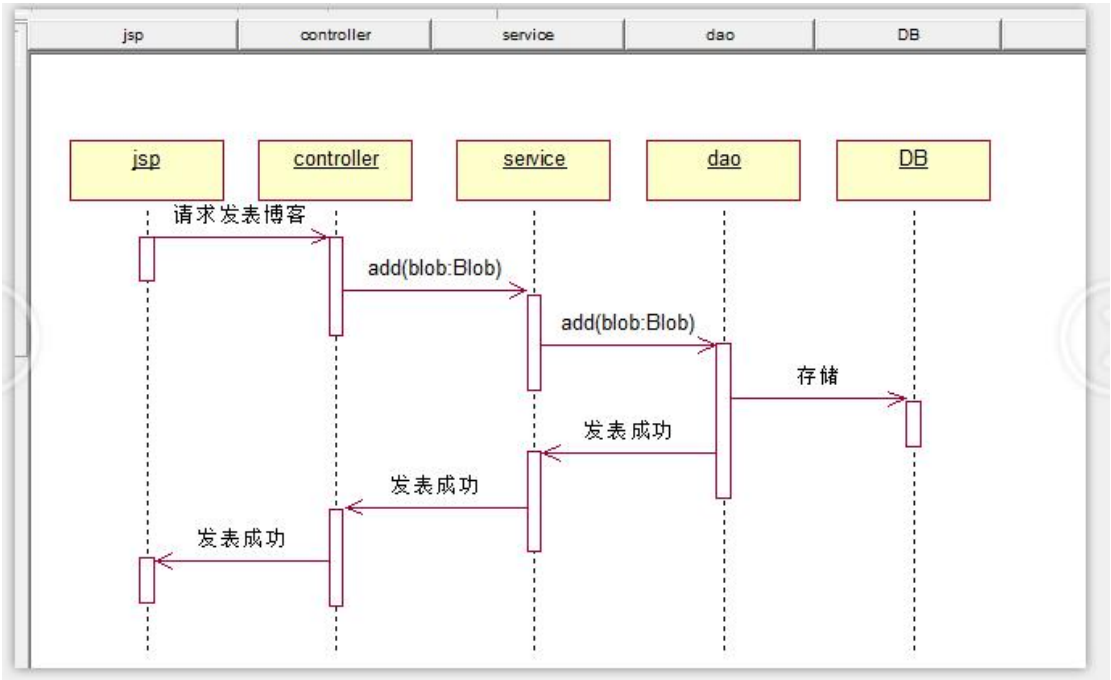
## 6.2.1 博客查询



步骤	系统行为描述
1	博客信息管理页面点查询按钮
2	页面表单提交调用控制层控制层的list方法
3	控制层list方法中调用服务层的list方法
4	服务层调用持久层list方法返回博客集合

按条件查询博客, 调用控制层的list方法, 在其中调用服务层list方法, 返回blog集合: `List<Blog> list`。获得集合后使用方法把查询到符合要求的数据传到前台, 在前台对数据进行处理。

6.2.2 博客新建

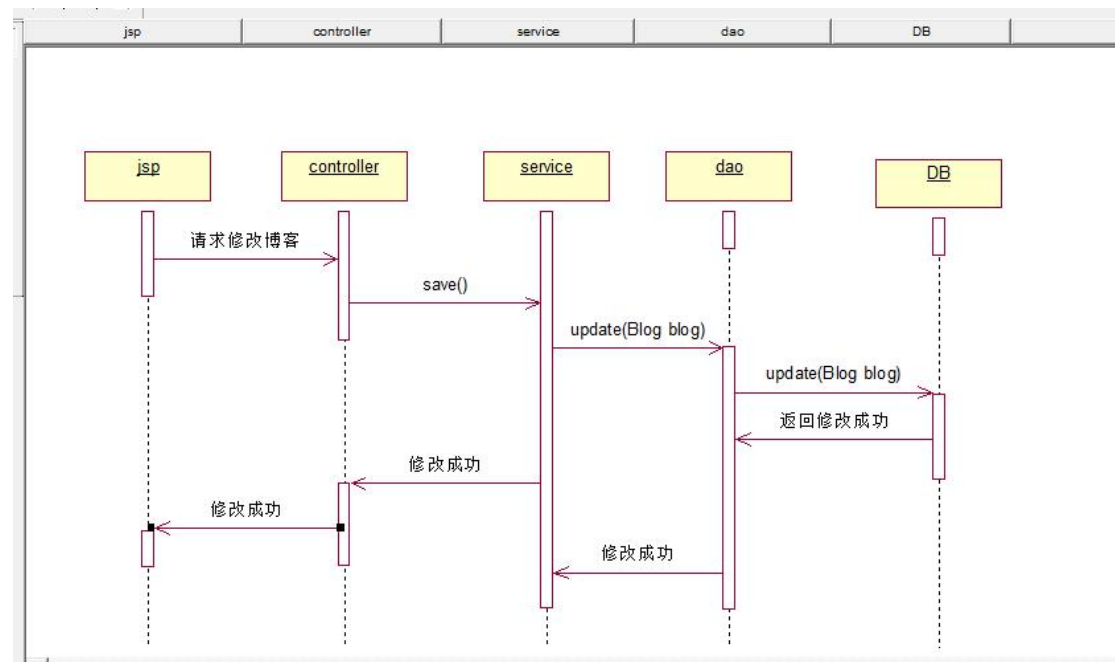


步骤	系统行为描述
1	博主在后台页面填写博客信息点击发布博客按钮
2	Jsp表单提交调用控制层controller的save方法
3	控制层save方法中调用service层的add方法
4	调用blogMapper持久层保存博客信息
5	添加索引

点击写博客进入博客发布界面，填写博客内容信息，点击发布博客，提交到控制层的save方法，传入参数：Blog，该参数包含了该博客所有信息内容，在提交的时候使用javascript对博客的内容进行校验。校验成功后便会调用控制层的方法，在控制层的save（）方法中调用服务层的add（）方法，对该博客进行保存，持久到数据库中。



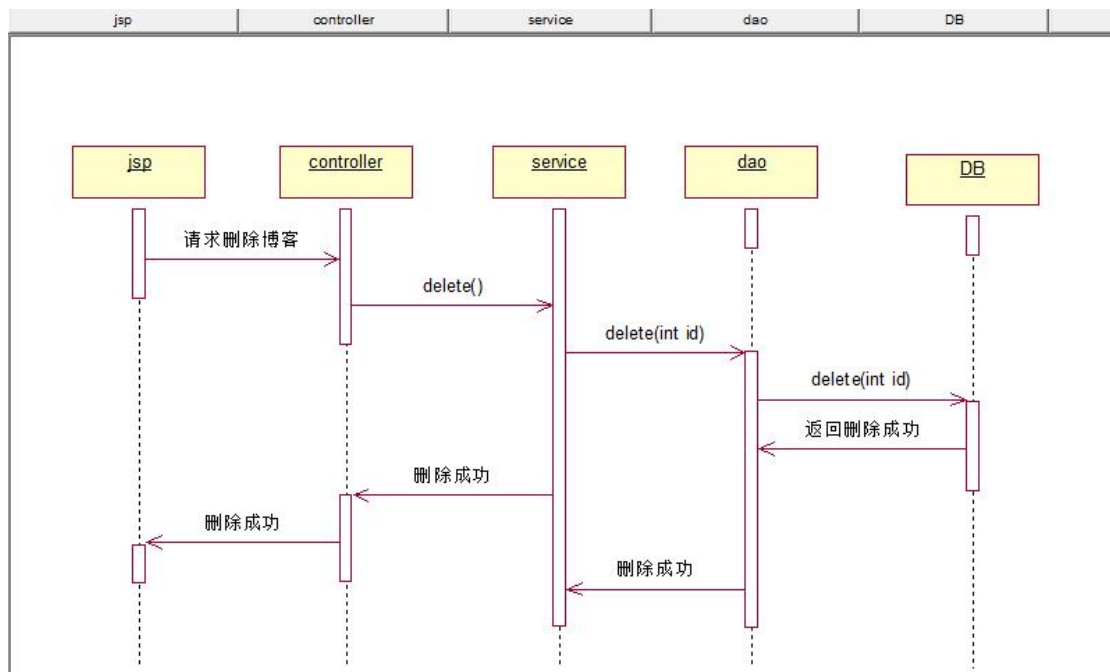
### 6.2.3 博客修改



步骤	系统行为描述
1	博主在博客信息管理页面选择需要修改的博客打开博客修改页面
2	Jsp表单提交调用控制层controller的save方法
3	控制层save方法中调用service层的update方法
4	调用blogMapper持久层保存修改后的博客信息
5	更新索引

进入博客修改页面，修改博客内容信息，点击发布博客，提交到控制层的save方法，传入参数：Blog，该参数包含了修改后的博客所有信息内容，在提交的时候使用javascript对博客的内容进行校验。校验成功后便会调用控制层的方法，在控制层的save（）方法中调用服务层的update（）方法，对该博客进行保存，持久到数据库中。

## 6.2.4 博客删除

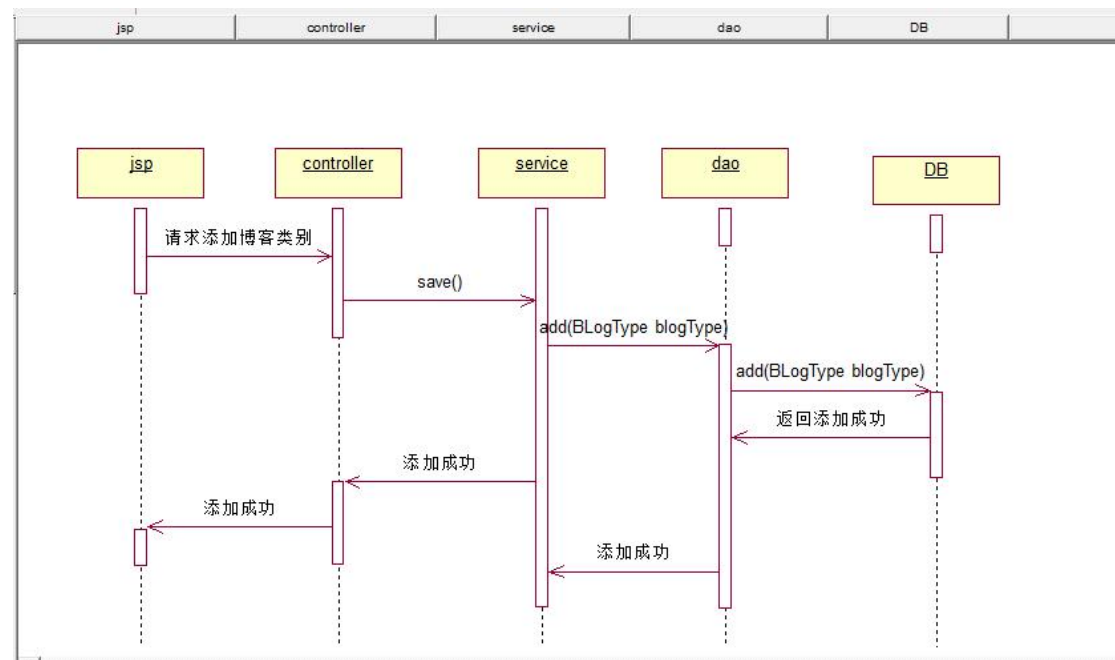


步骤	系统行为描述
1	博主在博客信息管理页面选择需要删除的博客(可以选择多条)
2	调用控制层controller的delete方法
3	控制层delete方法中调用service层的delete方法
4	调用blogMapper持久层将对应的博客信息删除
5	删除对应博客的索引

在博客管理页面选择需要删除的博客，点击删除按钮，提交到控制层的delete方法，传入参数：需要删除博客的id，调用控制层的方法，在控制层的delete()方法中调用服务层的delete()方法，对该博客或多个博客进行删除，持久到数据库中。

## 6.3 博客类别管理模块

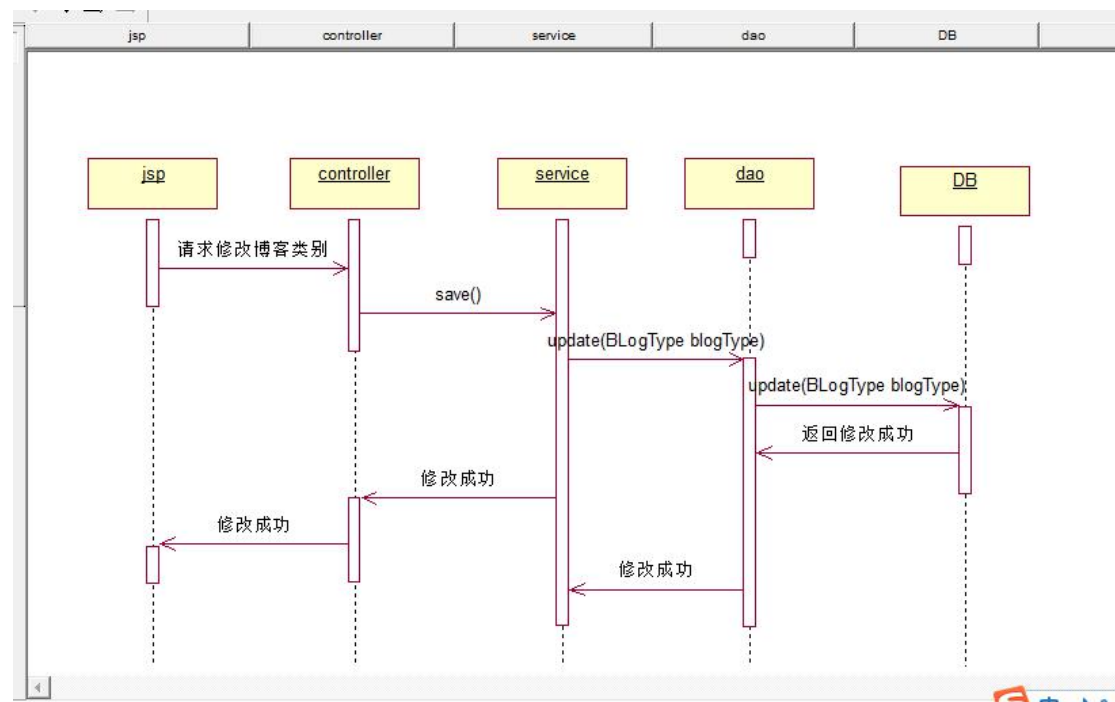
### 6.3.1 添加博客类别



步骤	系统行为描述
1	博主在博客类别信息管理页面点击添加按钮打开添加博客类别弹窗
2	调用blogType控制层controller的save方法
3	控制层save方法中调用service层的add方法
4	调用blogTypeMapper持久层将对应的博客类别信息添加
5	返回成功添加信息

在博客类别管理页面打开添加博客类别弹窗，填写博客类别名称和排序，点击保存按钮，提交到控制层的save方法，传入参数：需要添加的博客类型blogType，调用控制层的方法，在控制层的save（）方法中调用服务层的add（）方法，添加博客类别的相关信息，持久到数据库中。

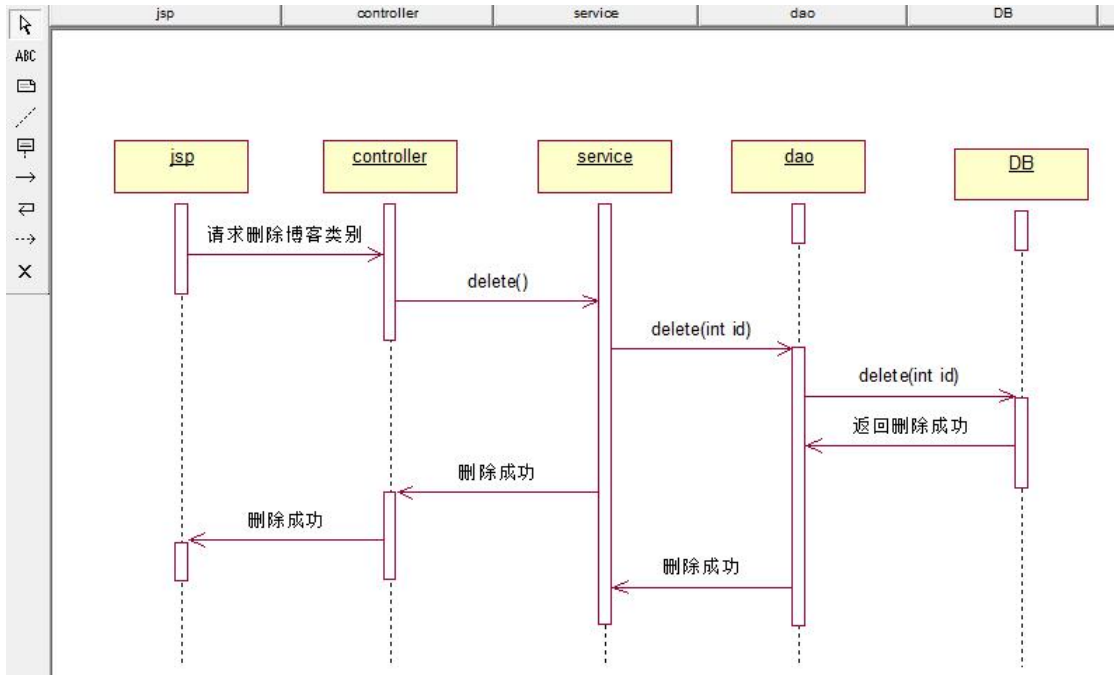
### 6.3.2 修改博客类别



步骤	系统行为描述
1	博主在博客类别信息管理页面选择要修改的博客类别点击修改按钮弹出弹窗
2	调用blogType控制层controller的save方法
3	控制层save方法中调用service层的update方法
4	调用blogTypeMapper持久层将对应的博客类别信息修改
5	返回成功系修改信息

在博客类别管理页面先选择要修改的数据，点击修改按钮，打开修改博客类别弹窗，填写修改后博客类别名称和排序，点击保存按钮，提交到控制层的save方法，传入参数：需要添加的博客类型blogType，调用控制层的方法，在控制层的save（）方法中调用服务层的update（）方法，修改博客类别的相关信息，持久到数据库中。

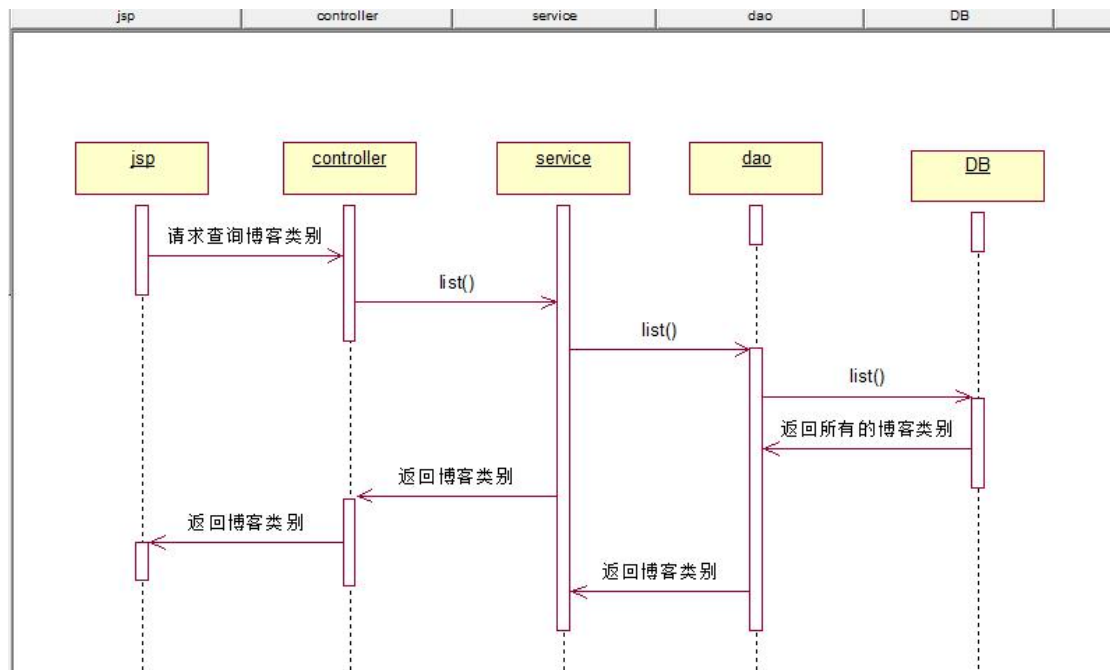
### 6.3.3 删除博客类别



步骤	系统行为描述
1	博主在博客类别信息管理页面勾选上要删除的博客类别，点击删除按钮
2	调用blogType控制层controller的delete方法
3	控制层delete方法中调用service层的delete方法
4	调用blogTypeMapper持久层将对应的博客类别信息删除
5	返回成功删除信息

在博客类别管理页面先勾选上要删除的数据，点击删除按钮，提交到控制层的delete方法，传入参数：需要删除的博客类型的id(可以是多个)，调用控制层的方法，在控制层的delete（）方法中调用服务层的delete()方法，删除博客类别的相关信息，持久到数据库中。

### 6.3.4 显示博客类别

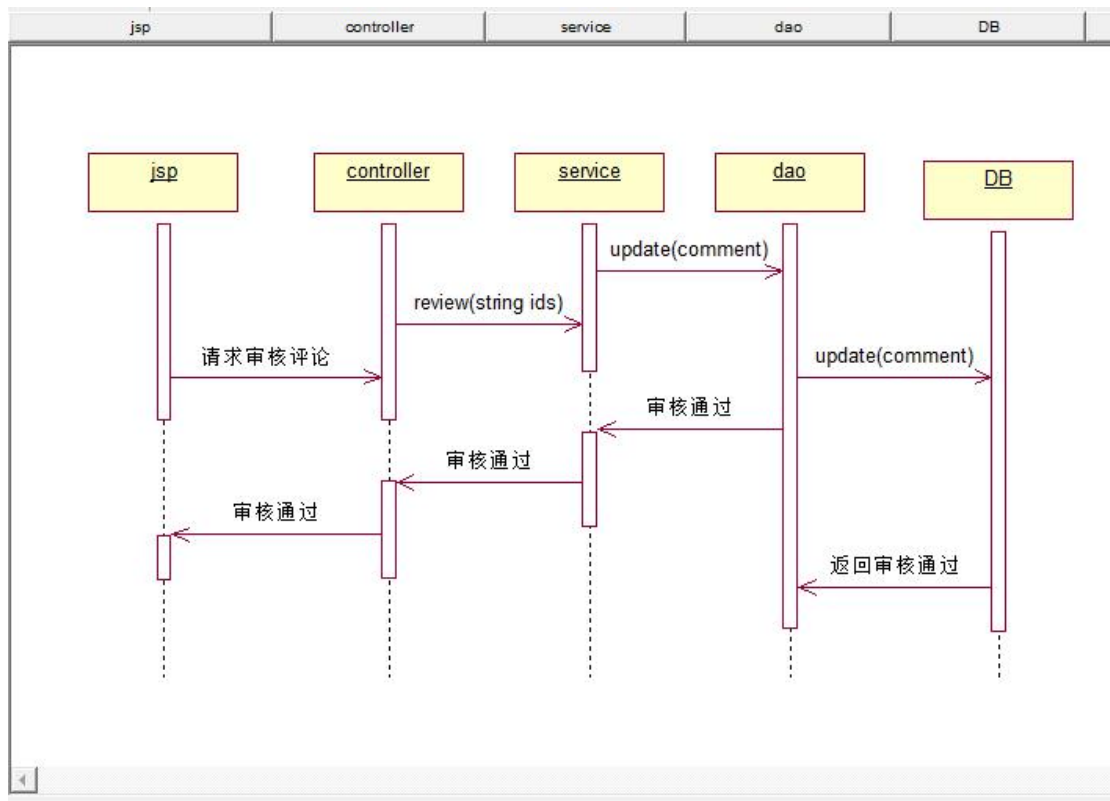


步骤	系统行为描述
1	博主在后台系统打开博客类别信息管理页面就会调用查询方法
2	调用blogType控制层controller的list方法
3	控制层list方法中调用service层的list方法
4	调用blogTypeMapper持久层查询所有的博客类别信息
5	返回查询到的博客类别信息

在打开博客类别管理页面时，调用控制层的list方法，传入参数：当前页数和每页数量，调用控制层的方法，在控制层的list()方法中调用服务层的list()方法，从数据库中查询到博客类别信息，然后将博客类别的相关数据渲染页面上。

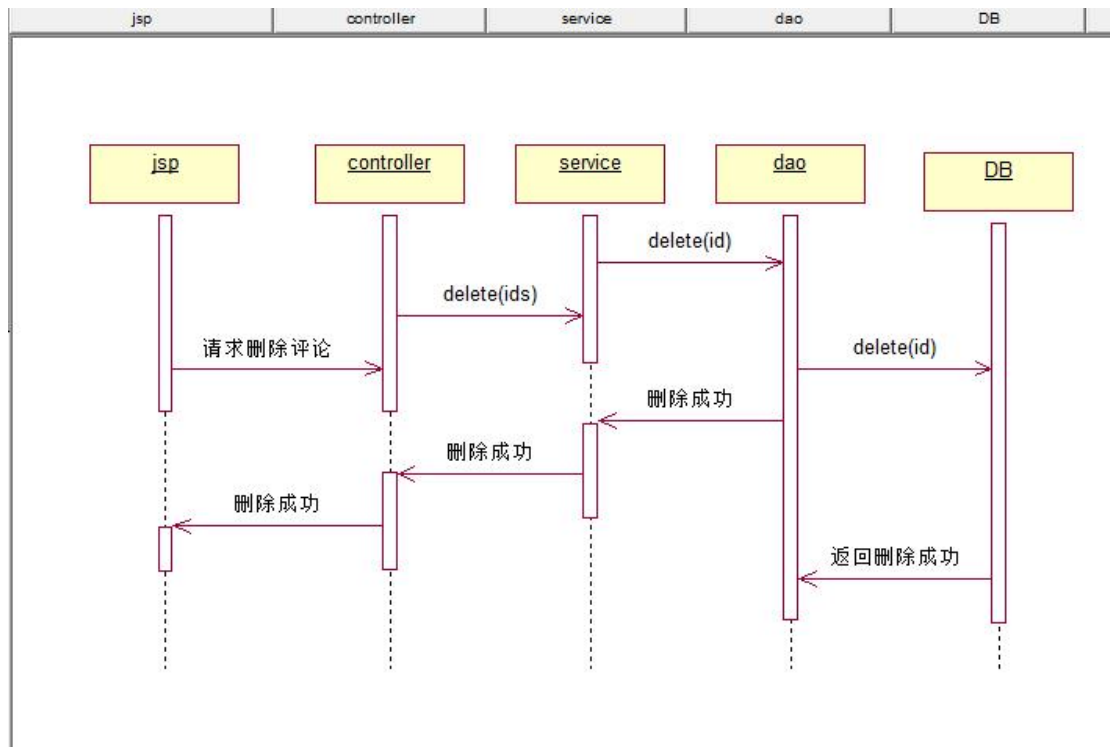
## 6.4 评论管理模块

### 6.4.1 审核评论



博主在后台页面打开评论管理页面，选择需要审核的评论，点击审核通过按钮，调用controller层的review方法，controller层调用service层的update方法，将评论的状态字段改为对应的代表通过审核的1，如果点击不通过那就是2。然后调用dao层的update方法持久化到数据库中。

## 6.4.2 删除评论

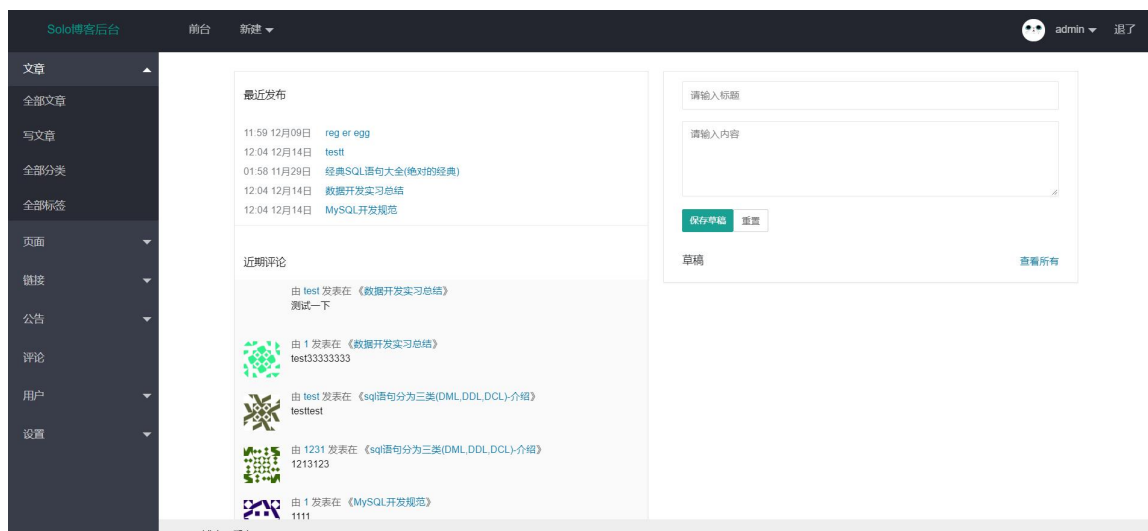


博主在后台打开评论管理页面，选择需要删除的评论，点击删除按钮。调用controller层的delete方法，在其中的方法中调用业务逻辑层的实现方法，然后调用dao层的方法，最后持久化到数据库中。



## 第七章 界面展示以及人机交互体现

### 7.1 后台界面



后台界面中，我们选择了在左边保留了完整的功能条，让后台的管理人员可以清晰，明显的找到自己想要的功能，并且可以在后台任意界面之间进行跳转。

左边的页面跳转条中，我们设定了大分类，每个大分类中包含了很多子页面，当后台管理人员用不到相应的大分类时可以将其折叠，这样页面显得比较干净，不会出现过于繁多的跳转链接在界面上。

子界面被放置在后台主页的中间处，来减少整体的刷新，增加用户体验。

### 7.2 文章管理



分类管理界面中我们选择在左边直接放置了一个添加分类的输入框，同时在右边将目前分类的状态进行了直接的显示，这样当管理人员在对分类进行检查后，可以直接在当前页面进行相对应的操作，不需要继续进行页面的跳转，这样减少了操作人员的页面跳转等待时间和操作量。

同理下面的标签管理也是这个布局，因此不再做单独叙述。

## 7.3 页面子界面

id	key	标题	内容	状态	操作
5	aboutSite	关于本站	<a href="#">关于项目</a> <a href="#">该项目使</a>	显示	<a href="#">编辑</a> <a href="#">删除</a>
6	testsolo	测试网站页面	测试网站页面	隐藏	<a href="#">编辑</a> <a href="#">删除</a>

别名	标题	内容
map	站点地图	<a href="#">点击查看</a>
articleFile	文章归档	<a href="#">点击查看</a>
message	留言板	<a href="#">点击查看</a>
applyLink	申请友链	<a href="#">点击查看</a>

温馨提示:  
1、自定义的页面，无法删除，别名已写入控制器

该界面采取了两部分，上部分为详细的页面列表，改部分中可以直观的看到页面的情况和进行对应的操作，下部为自定义页面，在该页面中我们能进行对自定义的页面进行查看和管理。

## 7.4 链接子界面

Solo博客后台

前台新建

admin 退了

文章

全部文章

写文章

全部分类

全部标签

页面

链接

公告

评论

用户

设置

首页 / 链接列表

名称	URL	联系方式	创建时间	Order	状态	操作	ID
江南一点雨	https://blog.csdn.net/u012702547/column/info/13987	2649060695	2019年12月14日	5	显示	编辑删除	10
唐亚峰   batton	https://blog.batton.com/		2019年12月14日	2	显示	编辑删除	14
Joker_Ye	https://blog.csdn.net/hj7jay/article/category/6135347/2	中国 广东省 广州市	2019年12月14日	2	显示	编辑删除	16
zhaoguowei的csdn链接	https://blog.csdn.net/qq_35885488	2649030395	2019年12月14日	1	显示	编辑删除	9
梁's Blog	https://blog.52listyle.vip/category/springBoot/	345849402@qq.com	2019年12月14日	1	显示	编辑删除	11
纯洁的微笑博客	http://www.iyouknow.com/spring-boot.html	一个有故事的程序员	2019年12月14日	1	显示	编辑删除	12
方志朋的博客	https://www.fangzhipeng.com/	网站描述: 生命不息, 奋斗不止	2019年12月14日	1	显示	编辑删除	13
Java3y	https://juejin.im/user/5a7005b46b9a01c995117a7	原创技术公众号: Java3y	2019年12月14日		显示	编辑删除	17

链接子界面同样采取了表单式的显示方式,在该页面中也能直观的观测到链接当前的状态和信息,同时能直接跳转到对应的操作界面,不需要再进行更多地页面跳转。

## 7.5 公告页面

Solo博客后台

前台新建

admin 退了

文章

页面

链接

公告

全部公告

添加公告

评论

用户

设置

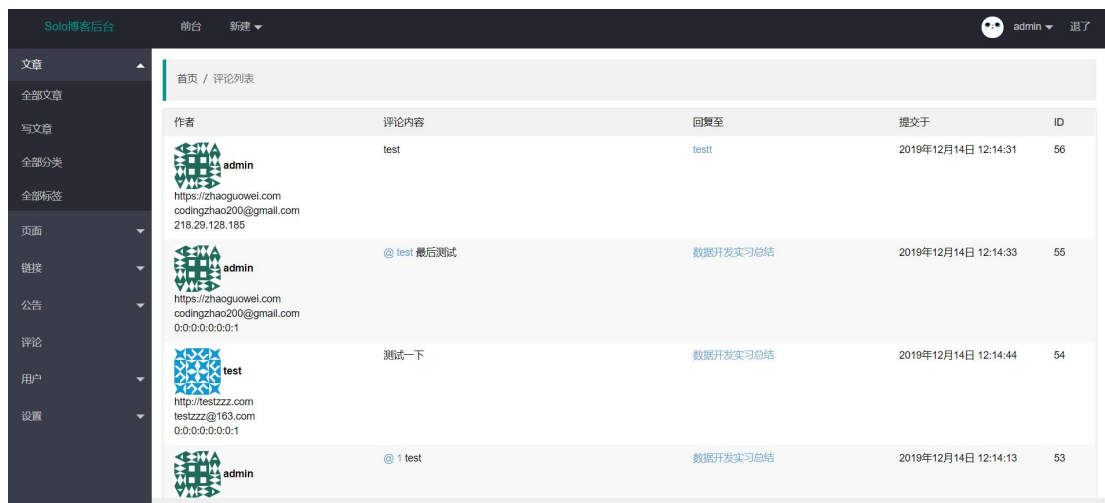
首页 / 公告列表

标题	Order	状态	操作	ID
欢迎来到我的个人博客网站	1	显示	编辑删除	4
本网站今天正式开放了	1	显示	编辑删除	5





温馨提示:  
Order的大小决定显示的顺序

该页面能显示出公告显示的等级,同时下面也有提示直接告诉操作人员公告显示的优先级取自于那个参数,减少了操作人员的记忆负担。

## 7.6 评论页面

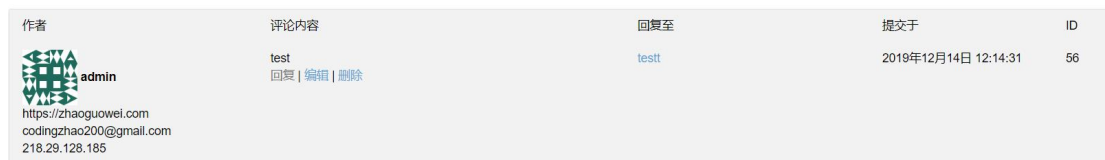


The screenshot shows the 'Solo博客后台' (Solo Blog Backend) interface. On the left is a sidebar menu with options like '文章' (Articles), '页面' (Pages), '链接' (Links), '公告' (Announcements), '评论' (Comments), '用户' (Users), and '设置' (Settings). The main content area is titled '首页 / 评论列表' (Home / Comment List). It displays a table of comments with the following data:


作者	评论内容	回复至	提交于	ID
 admin https://zhaoguowei.com codingzhao200@gmail.com 218.29.128.185	test	testt	2019年12月14日 12:14:31	56
 admin https://zhaoguowei.com codingzhao200@gmail.com 0:0:0:0:0:0:1	@ test 最后测试	数据开发实习总结	2019年12月14日 12:14:33	55
 test http://testzzz.com testzzz@163.com 0:0:0:0:0:0:1	测试一下	数据开发实习总结	2019年12月14日 12:14:44	54
 admin	@ 1 test	数据开发实习总结	2019年12月14日 12:14:13	53

评论界面中同样采用表单式布局，这样能更直观的看到评论者的评论。页面上显示出了评论时间，评论人，评论内容，评论ID和评论地点，这样方便操作者能够一眼看到评论的全部内容。

将鼠标移至对应的评论上时，会出现下图中的链接

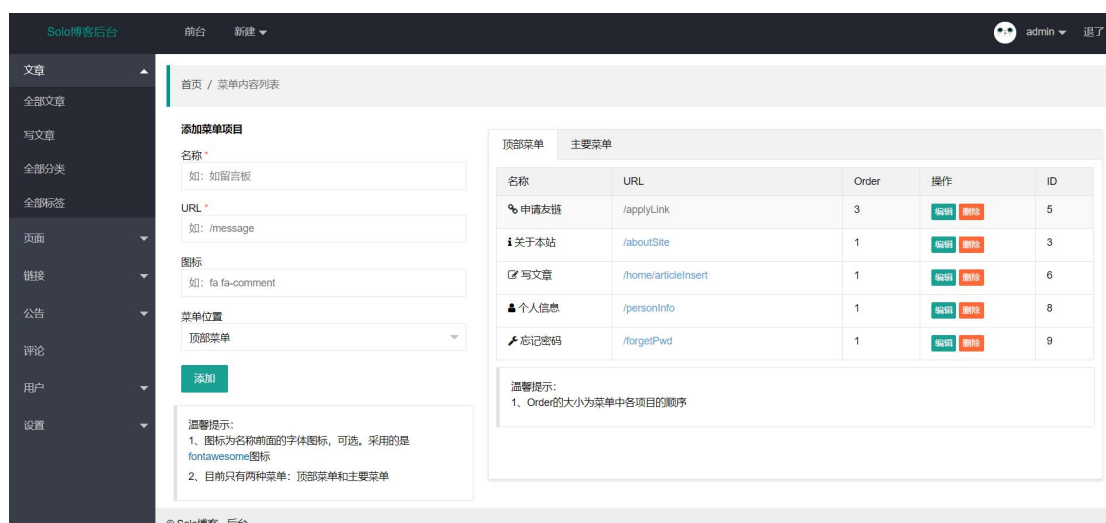


This screenshot shows a single row from the comment list, highlighting the interactive options available when hovering over a comment. The data is as follows:

作者	评论内容	回复至	提交于	ID
 admin https://zhaoguowei.com codingzhao200@gmail.com 218.29.128.185	test 回复   编辑   删除	testt	2019年12月14日 12:14:31	56

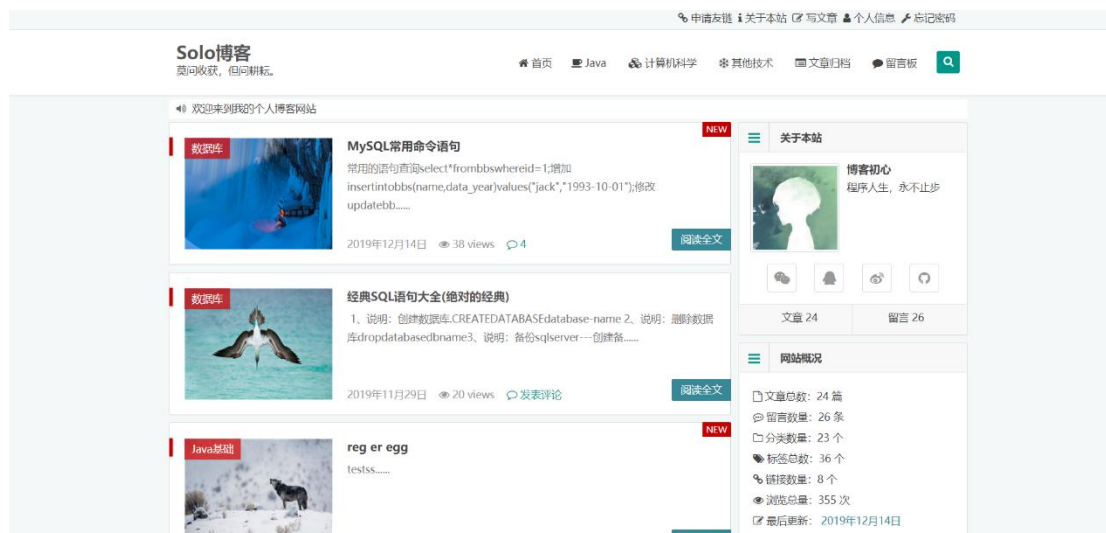
这样可以让操作者在想对某条评论进行某种操作时只能对那一条进行操作，不会出现误操作的情况，同时将所有的的操作直接显示出来供操作者选择，减少了操作者的操作量和记忆量，不需要来回的进行页面的跳转。

## 7.7 菜单页面



菜单界面跟分类管理界面类似，左边是新建菜单的操作区域，右边是对菜单的直接显示和管理，左下角的提示减轻了操作人员的记忆压力，右边的两个子窗口让菜单显示不是那么混乱，方便操作人员进行管理。

## 7.8 前台页面--修改前



首页中我们采取的布局是右边上部个人的信息，中部最显眼的地方是各种博主的简洁，这种布局能让用户第一时间将注意力集中在博文信息中。同时右边个人的信息也已经进行了大致的显示，上部的 TOP 条中有各种大致分类，这样能让用户在主页中直接进行跳转，同时有搜索按钮能对博文进行筛选，采取了较为传统的图案，减少了用户的理解压力和操作压力。

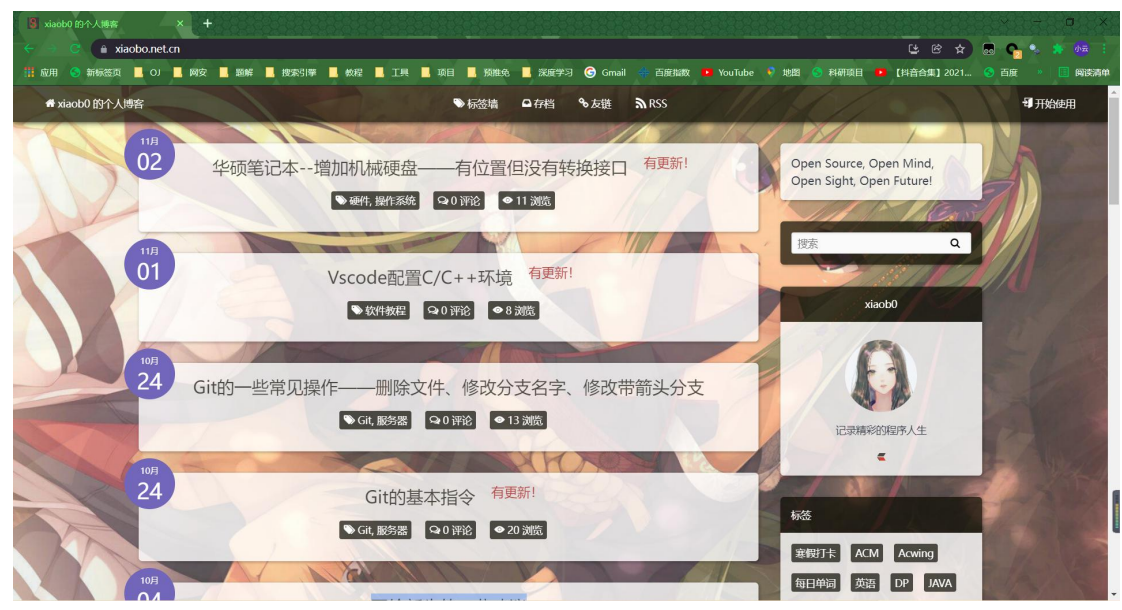
## 7.9 前台页面--修改后

本来博客的界面比较传统，开发出的界面只体现人机交互的可用性目标，对于用户体验目标采用 css+js 进行美化

### 5.1 交互设计的目标

交互设计不仅关心最终产品是否能够满足用户的需要，帮助用户更加高效地完成任务，同时还关注其他诸如用户是否满意以及系统是否容易使用等特性。

- 可用性目标
- 用户体验目标





## 7.10 人机交互目标体现

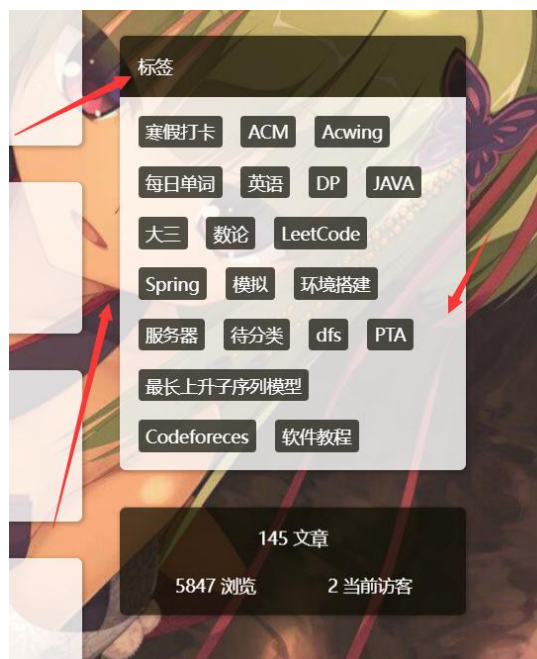
### 7.10.1 可用性目标：

易学性、易记性：

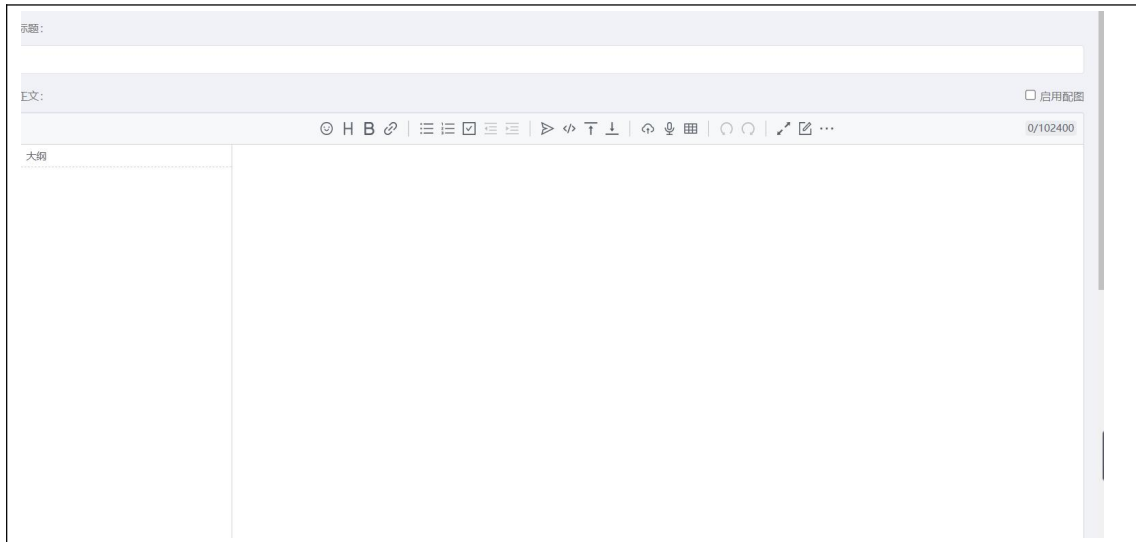
实现方法：文章标题+时间排序



根据标签页分类排序：



### 7.10.2 高效率目标：采用 markdown 语法格式



### 7.10.3 主观满意度：代码框显示代码

通过代码片段显示代码，右上侧显示整个文章的大纲，便于用户阅读





## 第八章 调试过程中的问题及系统测试情况

### 8.1 调试过程中问题

1、窗体和各种监听的连接，各种监听在书写时出现了一定程度的问题。在经历过多次更改后才成功将各类窗体跟监听联合了起来。

2、窗体中成员的制作，对窗体中出现的表格等一系列产物进行制作的时候经常出现各种各样的错误而导致返工。

3、String和Int参数的获取问题，通过函数将对应的数据进行转换

4、数据库连接问题，通过mybatis框架将数据库进行了链接。

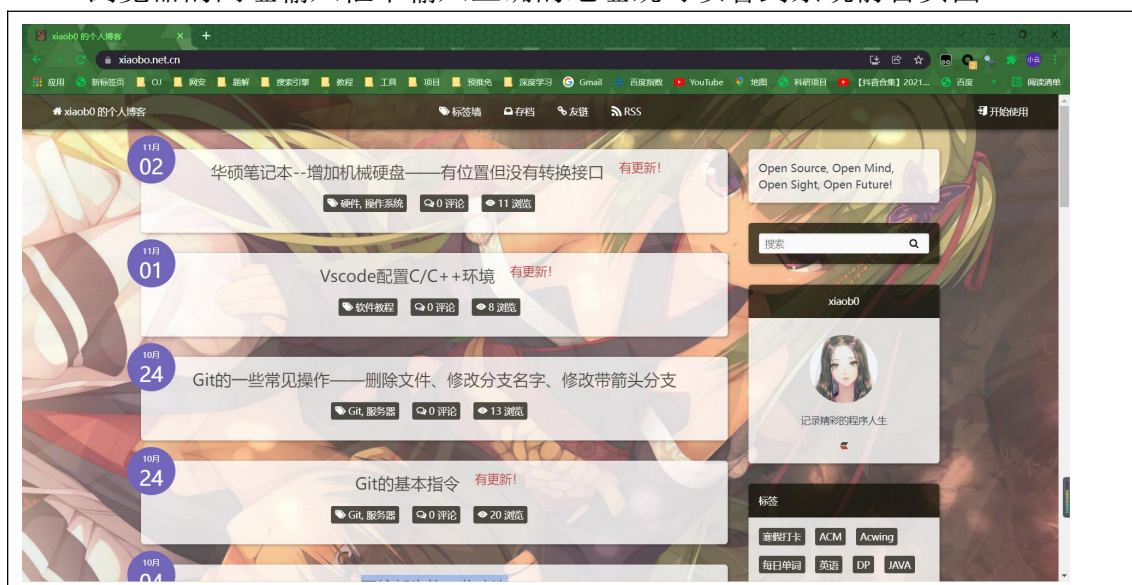
5、再调用各种框架时出现了一些BUG，登陆时也有些问题，现在暂时通过某种方式处理了。

### 8.2 系统测试情况

通过采用黑盒测试，逐步实现系统，基本实现目标系统功能，逐步完善系统功能。

#### 8.2.1 前台模块测试

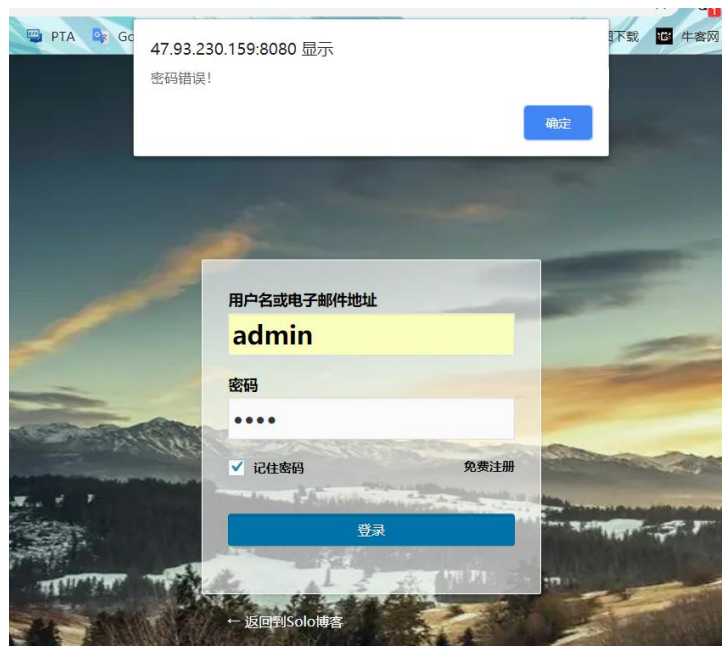
浏览器的网址输入框中输入正确的地址既可以看到系统前台页面：



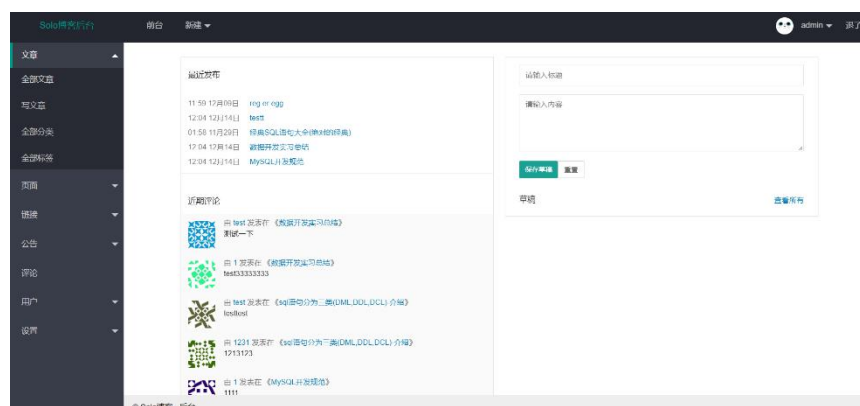
## 8.2.2 后台模块测试

在前台首页可以点击登录按钮可以登录到后台管理页面，在后台管理页面可以进行博客管理，博客类别管理，评论审核和个人信息管理还有系统管理。但是首先要获取用户名和密码，拥有博主权限才可以进入到后台管理页面，首先输入用户名：admin，然后输入密码：admin。结果有两种，一种是密码正确成功登录，另外一种是不成功，登录失败，若是失败，则会在页面上提示：用户名或密码错误。

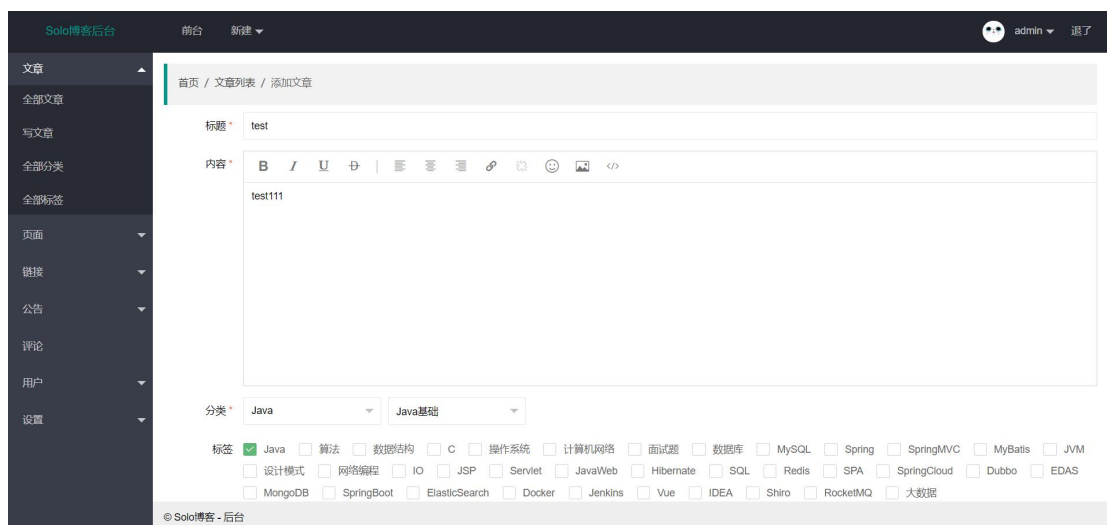
如下图所示：



登录成功后，可以进入到后台管理页面，在这个页面，博主可以管理博客，可以新增博客，可以删除修改博客，也可以管理评论等等。



测试写博客：



点击发布，成功。

---

## 第九章 项目创新部分

### 9.1 想法由来

博客是一个表达情感，分享经验、生活的地方，最重要的就是能够随心记录点点滴滴。

关于选择独立博客还是选择个人博客服务的讨论比比皆是。究其根本，如果有廉价、稳定的空间搭建独立博客，且安装部署简单、后续维护简洁、使用方便，那别人提供的博客服务就完全没有优势了。

随着云服务的普及，搭建一个廉价、甚至是免费的独立博客已经不再是只有“高手”才能玩的了，我们只需要自己动手就能搭建。

使用博客服务就没有那么简单了，写文章时得注意内容关键字；稍微敏感的文章会被删除；国家法定节假日不让发表文章；说维护就停止服务；说封杀就封杀。用户的理由都不是理由，注册账号的时候已经告知过了。

从大一到大四，体验了很多博客，比如CSDN，博客园，简书，知乎专栏，Github Page，每个博客都有自己的优点和缺点，CSDN广告多，定制性差，博客需审核，博客园比较陈旧，众多原因一直没找到一个适合自己的博客，因此，也就诞生了自己做一个博客的想法。

### 9.2 实现简介

数据库与功能设计参考forest开源博客的基础上，融入自己的想法，规划了整个系统的功能，后台使用ssm框架实现。在设计之初，参考了几个开源博客的优点以及所缺乏的功能，又根据人机交互的原则，从功能方面尽量给用户良好的体验。

前端的UI设计和前端制作. 前端制作方面, 主要使用的是LayUi框架来实现的前端制作, 由于所使用的大部分UI功能和实现都是框架内自带的功能, 基本上都能做到比较容易的, 如整体网页的跳转以及子界面的构成, 子界面的嵌入和对信息的调用。

整体UI设计方面我采取了比较传统的UI形式, 大部分都是采用了顶部, 中部, 左/右部的设计形式, 子页面的跳转也直接设计在了主页上, 来方便用户进行跳转.

---

整体的风格我采取了主页使用白色,而后台使用了黑白交替,这样个人认为可以在主页上显示的整洁一点,同时看起来更为明亮舒畅.后台的黑色部分则想要给人一种较为庄重的感觉,这样可以让操作员在操作的时候更加谨慎,也给了操作者一种暗示,让他认为他的工作需要谨慎对待.

### 9.3 编辑器使用 Markdown

Markdown 是一种轻量级标记语言,它允许人们使用易读易写的纯文本格式编写文档。Markdown受广大用户的热爱,非常适用于博客系统。

用户可以在后台导出.md或者.html文件,用于生成.md文档和生成html静态页面,方便用户使用,不论是在线上还是线下。

### 9.4 使用 OpenTag

Tag在以信息为主的站点中已经非常普及了。举个例子,一篇文章“Java 持久化中的Hibernate”被发布了,那么Tag的建立方式可以有如下两种:

1. 作者对这篇文章进行内容分类。

至少作者应该添加“Java”与“Hibernate”这两个 Tag,其突出的是作者对文章内容的归纳,也即关键字。

2. 读者对这篇文章进行个人分类。

读者 A 可能会添加“Java”与“Hibernate”;读者 B 可能会添加“冬眠”、“Java”与“英语”....其突出的是对各读者对文章内容的归纳,是个性化的标记。

从它的使用者的角度对这两种Tag建立方式进行了统一。可以看到作者的Tag,同时作为读者还可以对其加入自己的Tag。这样做是非常大的一个进步(至少从“开放式应用”的角度上来看)。

但是,这个进步也许还不是足够开放。我们在Tag的建立方式上创新了一下:文章作者在发布文章的时候添加文章的Tag,但该文章的浏览者可以对其 Tag作出评分,并且可以补充Tag。

还是前面的例子:文章作者添加了“Java”与“Hibernate”,浏览者 A 对Tag“Java”进行了加分,“Hibernate”进行了减分,并补充了一个Tag:

“Spring”;浏览者 B 也对“Java”进行了加分,同时也对“Spring”进行了加分;一段时间后,这篇文章的Tag可以从高分到低分进行排序,从而可以确定该文章最符合的分类。这样,根据这个Tag就可以更准确地找到自己想要的内容。

---

## 参考文献

- [1] 李洋. SSM 框架在 Web 应用开发中的设计与实现[J]. 计算机技术与发展, 2016, (12):190-194.
- [2] 蒋启明, 彭小宁, 张文. “轻量级 Java EE”课程的 Spring 框架教学改革探讨[J]. 科教文汇(中旬刊), 2016, (11):57-58.
- [3] 马杰川. 基于 JSP 的企业 Blog 系统的设计与实现[D]. 电子科技大学, 2013.
- [4] 韦耿, 钟亮. 基于 JSP 的个人博客系统[J]. 电脑知识与技术, 2013, (21):4865-4869+4876.
- [5] 杨嘉群. 基于 JSP 的博客系统[J]. 电子制作, 2013, (09):68-69+32.
- [6] 祁新安, 李凯. 基于 JSP 的多媒体博客系统的研究[J]. 电脑开发与应用, 2013, (04):33-35.
- [7] 尚云龙. 关于网络博客系统的设计与实现[D]. 电子科技大学, 2013.
- [8] 邬心云. 日志式个人博客的自我呈现心理研究[D]. 华中科技大学, 2012.
- [9] 何发胜. 基于 SSH 模式架构的博客系统的设计与实现[D]. 湖南大学, 2012.
- [10] 崔程, 张钦. 一种基于 JSP 技术的博客系统设计[J]. 安阳师范学院学报, 2011, (05):93-95.
- [11] 郭向阳, 李红娟. 基于 JSP 技术的都市博客系统的设计与实现[J]. 软件导刊, 2010, (10):151-152.

---

## 致谢

此课程为时十一周，由于时间和能力的限制，有许多还未学到的人机交互知识。但是在这一段的实践和学习中，很大程度上提高了解决实际问题的能力，也确实使我收获很多知识，相信这些收获会对我以后的工作和生活带来很多好处。虽然在这个过程中遇到了很多困难，但是在老师和同学的帮助下，逐渐解决了一些问题，认真完成了实验报告和课后作业。

这一学期的课程中，有同学们的互帮互助，一起欢乐，有老师们的孜孜不倦，传授知识，也有大学生生活中的种种挑战。随着时间的流逝，这些终将变成美好的回忆留在我的心中。我永远不会忘记给我支持和帮助的老师 and 同学们。首先，我要感谢老师，老师的帮助和指导才帮助我更加完美的度过了这学期，学会了更多的东西。此外，我还要感谢那些给予我帮助的同学和朋友们，也正是他们的鼓励才让我更加坚定，同时他们也和我共同探讨找打了各种解决问题的方法，在看了我的实验报告听了我的思路后给了我建议，帮助我进一步完善了实验报告。要是没有你们的帮忙，或许我不会进步如此之快。在此我也由衷得感谢你们的帮助与支持，感谢你们在我本学期的大学生活中做出的贡献与陪伴。

在完成项目开发之后，我将整个项目部署到了阿里云，并让老师查看页面效果。老师也指出了整个项目所存在的问题，首先我比较注重一个项目从idea产生到项目开发再到项目发布整个全体的过程，因为本地的项目写的再好也没办法持久化使用，所以就忽略了前台人机交互的设计。但后来我也通过了html+css对页面的整体布局进行了修改。只对页面的首页进行了修改，而对页面的后台管理模块则不需要进行再次修改。因此我也十分感谢老师的帮助和指正。

最后，再次衷心的感谢每个老师和同学的帮助和陪伴。

---

## 附：组内分工情况

系统想法以及设计：201820110刘大禾、201820106冯海龙、201820109刘张澳 201820108白梦醒、201820107冯家豪、201820105代向龙
后台功能实现：201820106冯海龙、201820109刘张澳
前端界面实现：201820110刘大禾
数据库设计：201820109刘张澳、201820107冯家豪
系统设计文档撰写：201820110刘大禾、201820106冯海龙
工作量比重： 201820110刘大禾：30% 201820106冯海龙：20% 201820109刘张澳：20% 201820108白梦醒：10% 201820107冯家豪：10% 201820105代向龙：10%