

分类号：TP311.5

U D C：D10621-408-(2018)2479-0

密 级：公 开

编 号：2014081052

# 成 都 信 息 工 程 大 学

## 学 位 论 文

### 在线博客系统的设计与实现

论文作者姓名：	孔浩
申请学位专业：	软件工程
申请学位类别：	工学学士
指导教师姓名(职称)：	郑文斌（副教授）
论文提交日期：	2018 年 05 月 22 日

# 在线博客系统的设计与实现

## 摘 要

随着互联网技术的飞速发展，人们的生活与互联网紧密的结合在一起，到处都是互联网与日常生活结合的产物。同样，互联网使人们的日常学习更加方便和高效，通过互联网人们可以快速，有效地获取知识，博客就是一种方便的学习途径。博客永远是共享与分享精神的体现，通过博客我们可以自由地发表想法，分享知识，和他人交流讨论。

该博客系统的主要功能分为用户管理、博客管理、评论管理、点赞管理、分类管理、标签管理几个模块。该系统采用了 Spring Boot + Hibernate 框架技术进行开发，前端页面使用了 Thymeleaf 模板引擎，后台数据库采用的是 Oracle 旗下的关系型数据库 mysql，服务器端使用 Apache 的 Tomcat，使用了 Spring security 安全框架进行安全设置，使用了 Elasticsearch 搜索引擎实现了全文检索，开发工具使用的是 JetBrains 公司的 IntelliJ IDEA。

最终该博客系统实现了需求的大部分功能，由于使用了 Spring Boot 的轻型架构和 Spring security 的安全框架提高了系统的稳定性和可移植性。页面布局使用的 Bootstrap 框架，使得页面简洁明了且响应式布局自适应了不同的客户端，为用户提供了更好的交互式体验。

**关键词：**在线博客系统；Spring Boot；java；Thymeleaf；

# **Design and Implementation of Online Blog System**

## **Abstract**

With the rapid development of Internet technology, people's lives are closely combined with the Internet, and everywhere are the product of the combination of Internet and daily life. In the same way, the Internet makes people's daily learning more convenient and efficient. People can get knowledge quickly and effectively through the Internet. Blog is a convenient way to learn. Blog is always the embodiment of sharing and sharing spirit. We can freely express ideas, share knowledge and exchange discussions with others through blogs.

The main functions of the blog system include user management, blog management, comment management, point management, classification management and label management. The system uses the Spring Boot + Hibernate framework technology to develop, the front page uses the Thymeleaf template engine, the backstage database uses the relational database mysql under Oracle, the server side uses the Tomcat of Apache, uses the Spring security security framework to secure the security, uses the Elasticsearch search. The cable engine implements full-text search, and the development tool uses JetBrains IntelliJ IDEA.

In the end, the blogger system implements most of the functionality of the requirement, because the light architecture of Spring Boot and the security framework of Spring security make the stability and portability of the system outstanding. The Bootstrap framework used for page layout makes the page concise and responsive layout adaptable to different clients and provides a better interactive experience for the user.

**Key words:** Online blog system; Spring Boot; java; Thymeleaf;

# 目 录

论文总页数：37 页

1 引言.....	1
1.1 课题背景.....	1
1.2 课题的研究意义.....	1
1.3 国内外研究现状.....	1
1.4 本文的基本结构与主要工作.....	1
2 系统需求分析.....	2
2.1 需求概述 .....	2
2.2 功能性需求 .....	2
2.2.1 用例模型 .....	2
2.2.2 用例描述 .....	3
2.3 非功能性需求 .....	7
3 系统模块划分与概要设计 .....	8
3.1 系统模块划分 .....	8
3.2 系统模块设计 .....	10
3.2.1 登录注册模块 .....	10
3.2.2 博客发布模块 .....	11
3.2.3 博客查看模块 .....	12
3.3 数据库设计 .....	12
3.3.1 数据库模型 .....	13
3.3.2 数据库表结构设计 .....	13
4 系统详细设计与实现 .....	15
4.1 开发环境的搭建 .....	15
4.1.1 Mean 项目管理工具 .....	15
4.1.2 IntelliJ IDEA.....	16
4.1.3 MySQL.....	16
4.1.4 JDK 1.8.....	16
4.2 持久层的详细设计与实现.....	16
4.2.1 domian 实体的设计与实现 .....	16
4.3 登录注册模块的详细设计与实现.....	19
4.4 博客发布模块的详细设计与实现.....	21
4.5 博客查看模块的详细设计与实现.....	23
4.6 个人中心模块的详细设计与实现.....	26
5 系统运行及测试.....	28

5.1	测试环境.....	28
5.2	系统单元测试.....	28
5.3	系统测试结果.....	30
	结 语.....	31
	参考文献.....	32
	致 谢.....	33
	声 明.....	34

# 1 引言

## 1.1 课题背景

互联网的飞速发展，改变着人们的生活，同时网络上出现了很多新时代的产物，博客就是一个典型的例子，博客是一种重要的网络交流方式。人们可以通过博客查找自己需要的知识、信息，在博客上与别人分享自己日常生活中的趣事，发表自己对一些事物的见解，总而言之就是将自己想要分享的东西分享出去。随着博客逐渐流行，写博客成为人们日常生活的一部分，各大博客网站因运而生，但很多博客网站的用户体验不尽人意，人们需要一个开放自由、界面友好、简洁易用的博客网站。

## 1.2 课题的研究意义

该系统采用了 Spring Boot, Hibernate, Thymeleaf 等前沿技术，构建了灵活，稳定的系统架构。使用了 Elasticsearch 搜索引擎，实现了全文检索，方便用户快速的搜索出自己想要的內容<sup>[10]</sup>。该系统的页面使用 Bootstrap 构建的响应式布局，可以很好的适应不同设备的访问需求。该博客系统将为广大网友提供了一个自由，和谐，简洁的交流空间，用户可以自由的浏览发表文章，以及对文章进行点赞、评论，管理员也可以通过管理员页面对整个博客系统进行管理和监控。

## 1.3 国内外研究现状

博客 (blog) 是上个世纪末在美国兴起并迅速风靡世界的一种全新的网络媒体形式，简而言之就是网络日记。“它的出现标志着互联网从传统的信息共享开始渐渐跨过‘思想共享’的门槛，真正开始凸显其无穷的知识价值”<sup>[1]</sup>近些年来博客在中国飞速发展，国内涌现了很多博客网站，目前有 15 家博客专业网站，23 家非专业博客网站，其中专业博客网站以中国博客网，博客动力，新浪博客等为代表，非专业博客网站以天涯博客，donnews 等为代表<sup>[2]</sup>。博客用户规模持续快速发展。截至 2009 年 6 月底，用户规模已经达到 1.81 亿人，博客空间超过 3 亿，博客使用者体现了年轻化的特点，30 岁及 30 岁以下的博客使用者占到总数的 86.1%以上<sup>[3]</sup>。随着各大博客网站的兴起，出现了很多个人博客网站，相比来说个人博客网站更加的自由，开放，适合年轻人展现自我、凸显个性。

## 1.4 本文的基本结构与主要工作

本文的主要工作有以下几点：

1. 介绍博客的发展状况以及发展前景
2. 介绍了用于在线博客系统的框架技术：Spring Boot , Hibernate 等
3. 对在线博客系统的各个功能模块进行设计与分析
4. 按照需求分析实现在线博客系统
5. 对在线博客系统进行单元测试和模块测试

本文的基本结构如下：

1. 引言，介绍课题的背景以及研究意义，说明了本文的组织结构
2. 理论基础与相关技术的分析，介绍了本系统所用到的相关技术
3. 系统需求分析，对本系统的功能，性能进行了分析
4. 系统模块划分设计，对博客系统的各个模块进行划分并进行相应设计
5. 系统详细设计与实现，对系统进行详细设计
6. 系统运行及测试结果，查看系统实际运行效果，并进行测试

## 2 系统需求分析

### 2.1 需求概述

系统的功能主要为用户管理、博客管理、评论管理、点赞管理、分类管理、标签管理。

- 用户管理：用户可以进行登录注册，可以浏览自己的个人主页并对个人信息进行修改，可以修改的有个人邮箱、姓名、密码等，账号名称无法修改。
- 博客管理：用户能够浏览自己与他人的博客，对博客进行最热、最新排序。可以对自己的博客进行搜索，删除，修改等操作。
- 评论管理：用户可以对自己或者他人的博客进行评论，同时可以对自己发表的评论进行删除操作。
- 点赞管理：用户可以对自己或者他人的博客进行点赞操作，同时可以删除自己的点赞
- 分类管理：用户可以对自己的博客进行分门别类，方便管理自己的博客
- 标签管理：用户在发表博客时可以给自己的博客添加标签，方便别人浏览。

### 2.2 功能性需求

#### 2.2.1 用例模型

上述功能模块虽多，但大体分为四个部分，登录注册，个人中心，博客的发布，博客的查看以及互动（评论，点赞）。通过对功能的分析，细化得到了如图 2.1 所示的系统用例图。

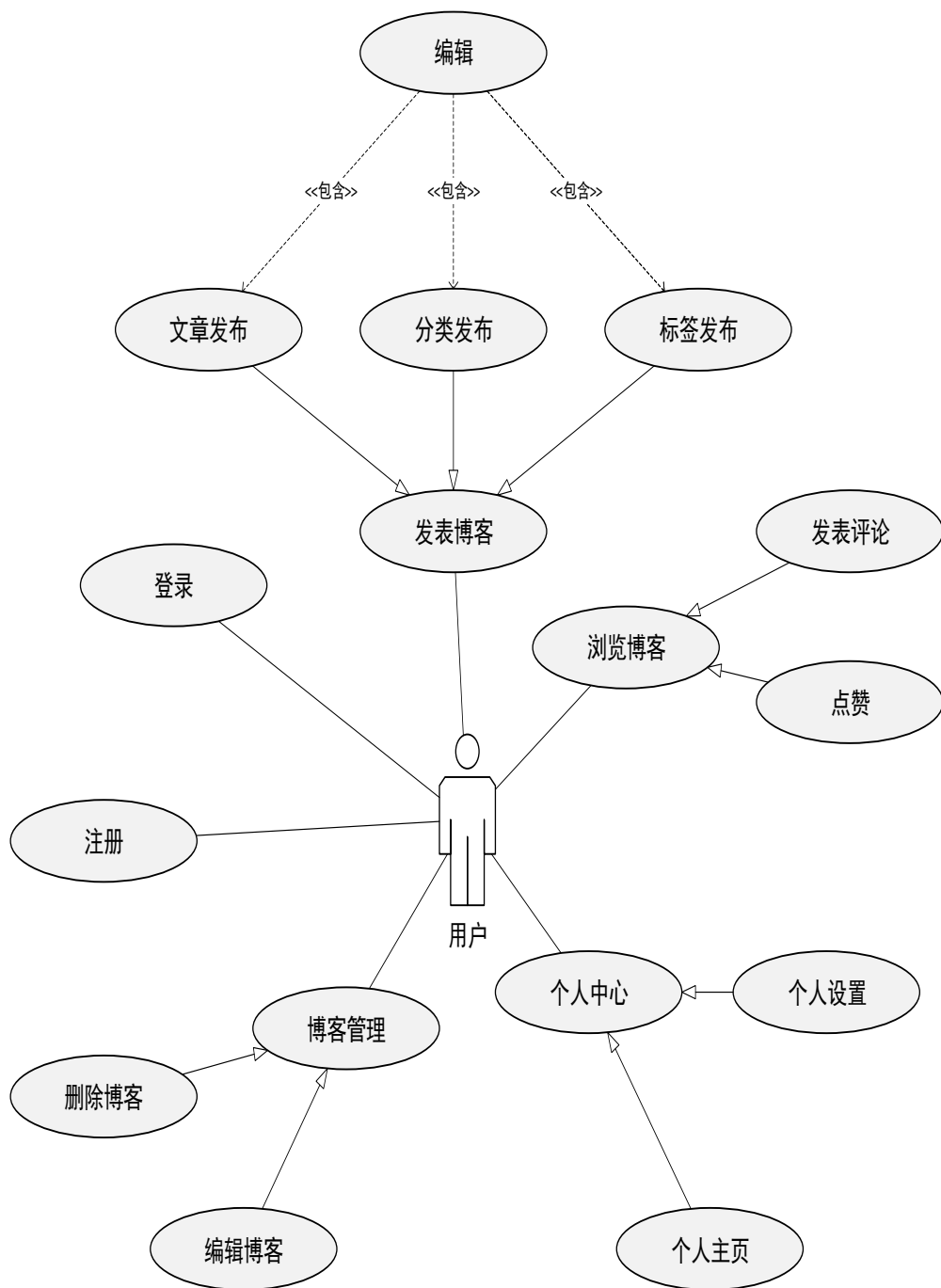


图 2.1 系统用例图

### 2.2.2 用例描述

- **登录与注册用例：**用户点击登录按钮之后可以进行登录，在登录之前用户需要先注册用户，只有登录之后才能进行其他操作，该用例活动图如图 2.2 所示。



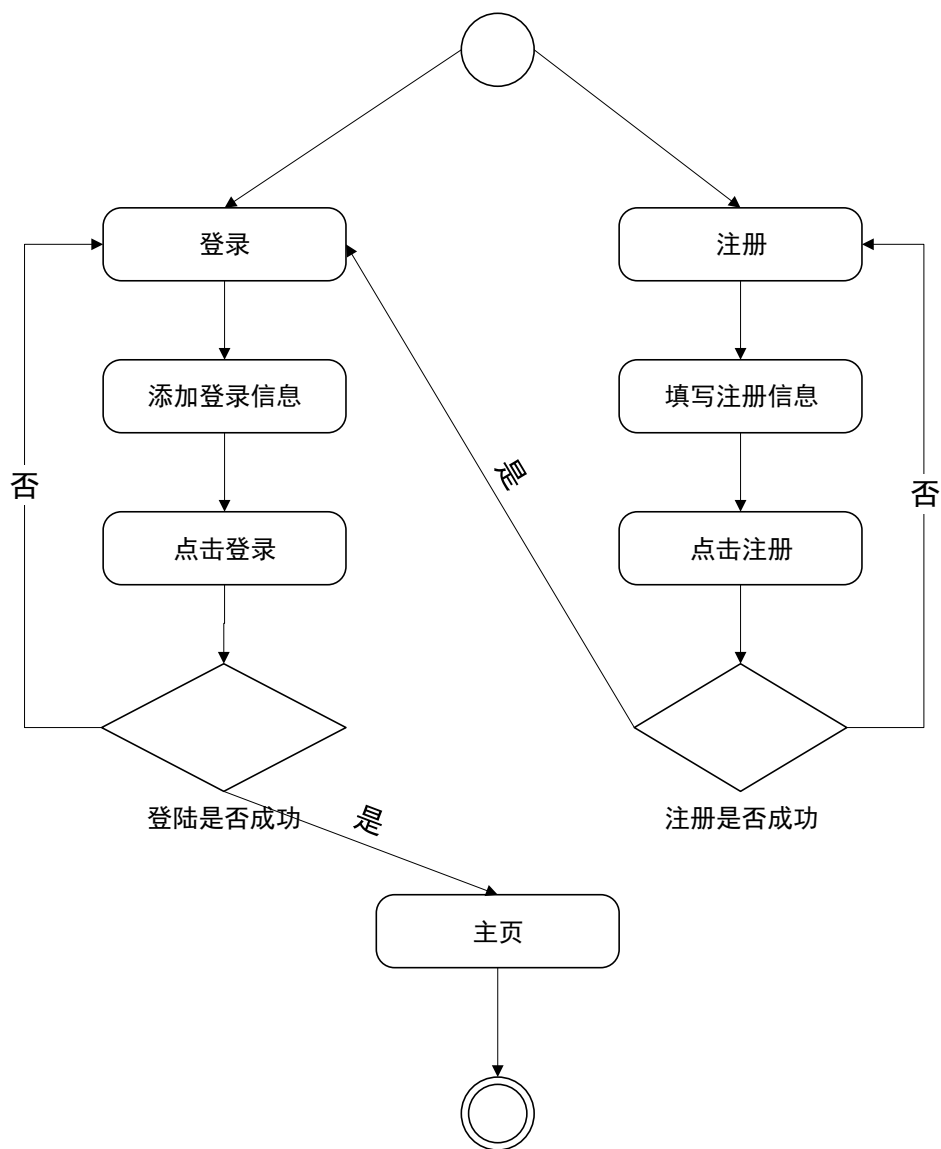


图 2.2 登录注册活动图

活动流程：

用户进入主页

(1) 点击登录按钮

1. 填写登录信息后提交
2. 若登录信息正确，跳转至主页
3. 若登录信息错误，跳转至登录错误页面

(2) 点击注册页面

1. 填写注册信息后提交
2. 若注册成功跳转至登录页面
3. 若注册失败跳转至注册页面并返回失败信息

■ 发表博客用例：用户可以编写博客标题，博客摘要，博客正文然后发布

博客，在发布博客时还可以给博客添加标签、分类，用例活动图如 2.3 所示。

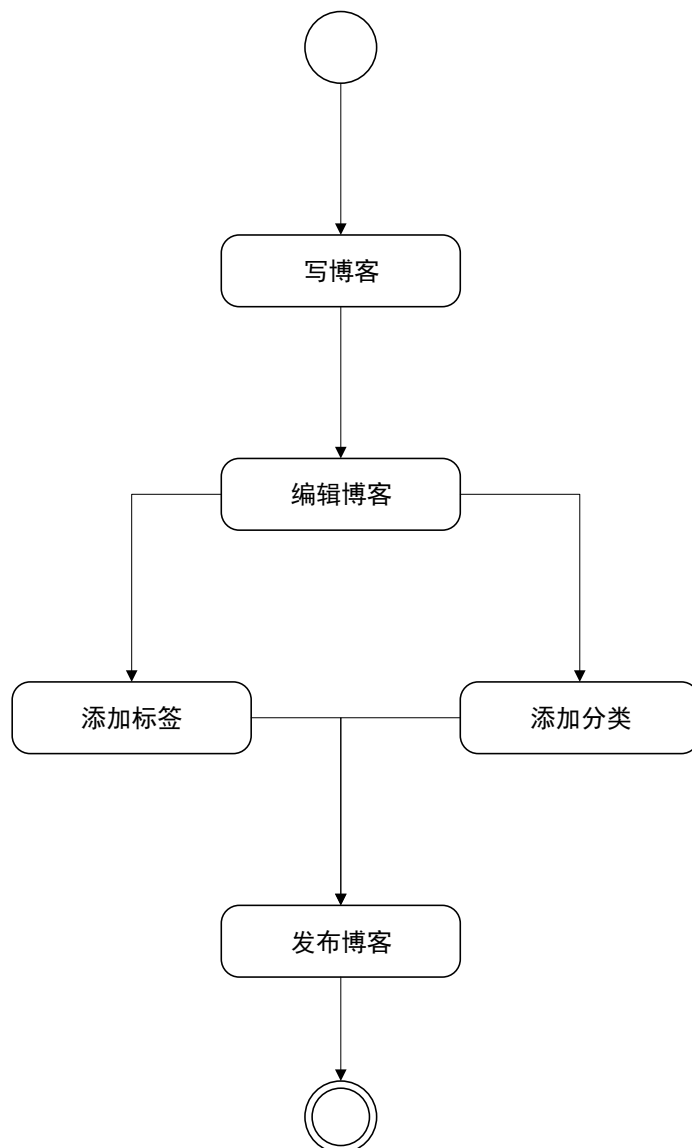


图 2.3 发布博客活动图

活动流程：

用户点击写博客按钮，进入写博客页面

1. 填写博客标题、摘要、正文
2. 为博客添加标签
3. 点击分类按钮，为博客选择分类
4. 点击发布按钮，发布博客

■ 博客查看用例：用户登录之后可以查看自己和他人的博客列表，浏览每个博客的详情页面，可以在每篇博客下发表评论和点赞，博客查看用例活动图 2.4 所示。

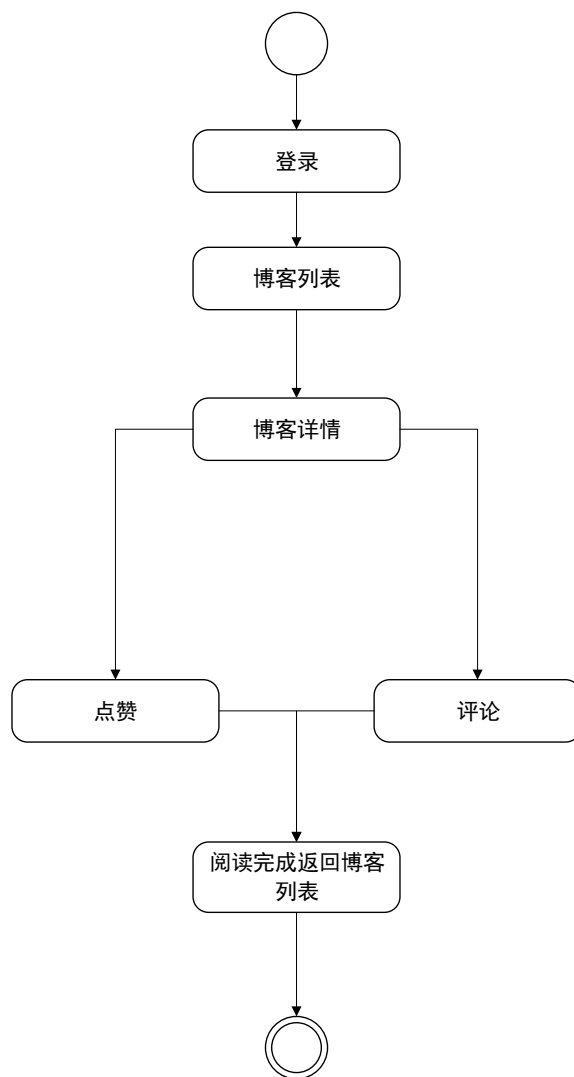


图 2.4 博客查看活动图

活动流程:

用户登录成功后，显示博客列表

1. 点击某个博客，进入博客详情页面
2. 点击点赞按钮为博客点赞
3. 在评论框中输入评论，点击发表，发表评论
4. 阅读完博客后返回博客列表

■ 个人中心用例：用户登录之后可以访问自己的用户中心，可以选择个人主页或者个人设置。在个人主页中，用户可以查看自己的个人信息，发布的博客。在个人设置里，用户可以修改自己的个人信息，个人中心用例活动图如图 2.5 所示。

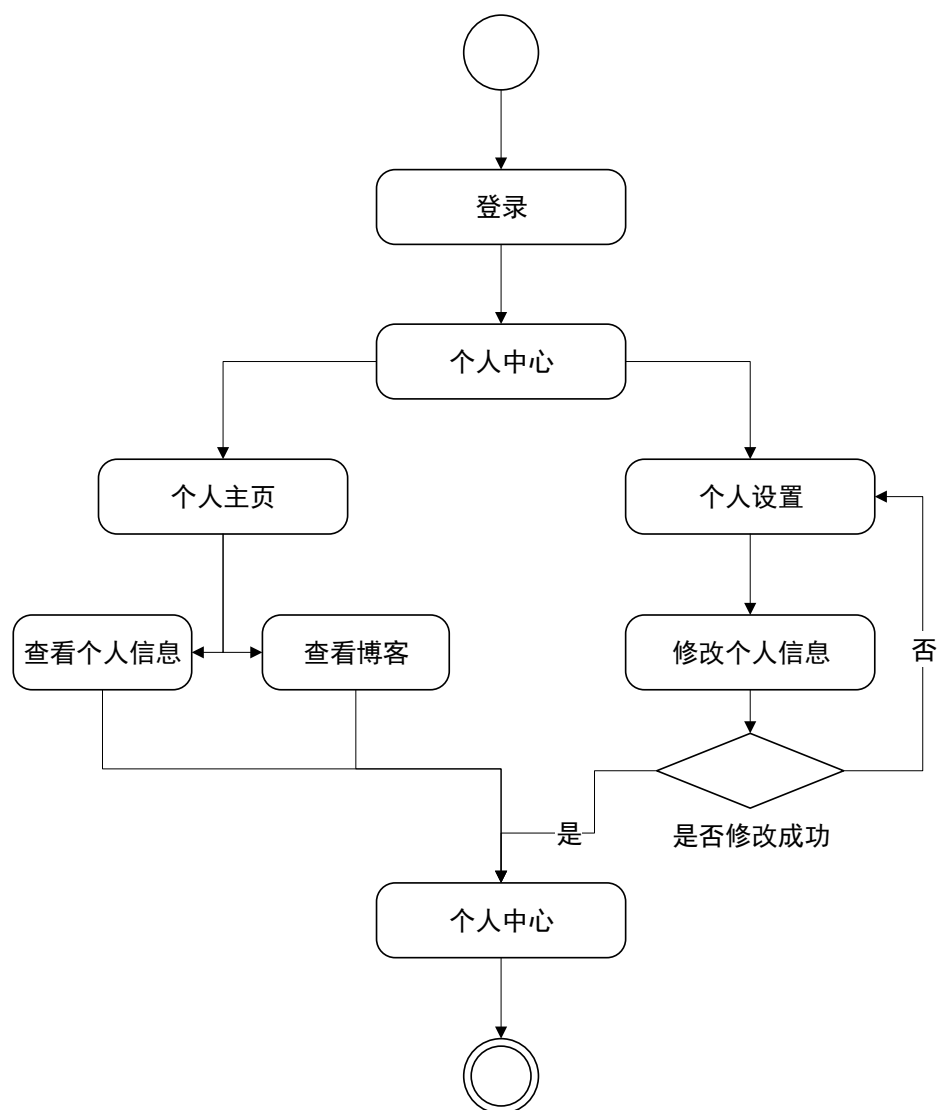


图 2.5 个人中心活动图

活动流程：

用户登录成功后，点击个人中心，显示个人中心菜单

(1) 点击个人主页

1. 显示个人信息，发表过的博客

(2) 点击个人设置

1. 显示个人信息

2. 修改个人信息，点击保存按钮

3. 保存成功，返回个人设置页面

4. 保存失败，返回个人设置页面显示失败原因

## 2.3 非功能性需求

- 易用性：系统页面采用了 Bootstrap 前端框架，外观明朗大方，操作按钮醒目易懂，使用了响应式布局，使得移动设备也可以很好的使用。

- 安全性：使用了 Spring Security 安全框架，提供了安全级别很高的用户认证和用户授权，每个用户有着自己的角色，只能使用自身角色可以使用的功能。
- 可维护性：使用了 Spring Boot 框架，用依赖注入的方式创建实例大大减少了系统的耦合程度。本系统采用了接口化编程，后续向系统中添加，修改功能都十分的方便<sup>[4]</sup>。
- 稳定性：系统使用了 Java 语言编写，Java 语言自带的垃圾回收机制使得系统可以长期稳定地运行而不会无故死机。
- 可扩展性：使用了接口化编程，使得系统可以很方便的进行升级。
- 可移植性：该系统使用了 Spring Boot 作为主体框架，内置了 Tomcat，只需将系统打包成 jar 包，便可在任意安装了 Java 环境的设备中直接运行。

## 3 系统模块划分与概要设计

### 3.1 系统模块划分

在线博客系统是基于 Spring Boot 的 WEB 系统，功能主要分为四个部分：登录注册、个人中心、博客的发布、博客的查看以及互动（评论，点赞）。对于这四个模块的结构、功能继续进行细分：

- 登录注册：用户登录需要填写账号和密码，注册时需要填写账号、邮箱、姓名和密码，账号至少 3 个字符，至多 20 个；邮箱必须为正确邮箱格式；姓名至少 2 个字符，至多 20 个；密码可以为字母或特殊符号和数字结合。
- 个人中心：个人主页可以查看用户的基本信息，以及用户曾经发表的文章，可以编辑，修改，删除自己的博客分类。个人设置可以对自己的个人信息（邮箱，姓名，密码，头像）进行修改。
- 博客的发布：可以编写博客标题，博客摘要和博客正文，可以为博客添加标签、分类。
- 博客的查看：可以查看博客列表，以及某个博客的详细界面，可以进行添加、删除评论，添加、删除点赞等操作。

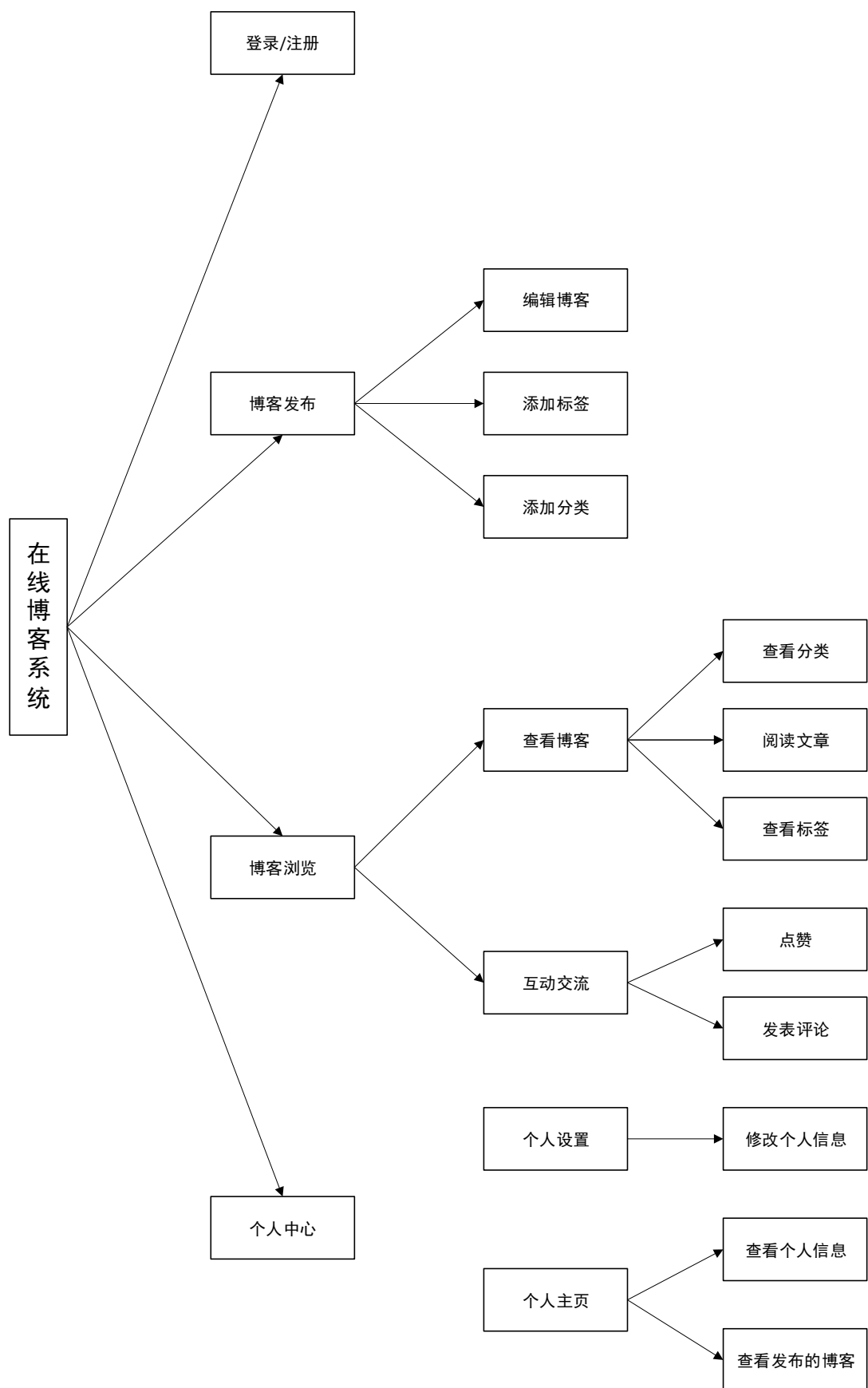


图 3.1 系统模块图

### 3.2 系统模块设计

在 4.1 中对系统功能模块进行了大致的划分，接下来对四个大模块的功能进行设计。系统的整体流程是，用户注册登录，通过验证登陆成功后选择博客发布、博客查看、个人中心等操作，再在相应的模块下进行相应操作。系统整体的活动图如图 3.2 所示。

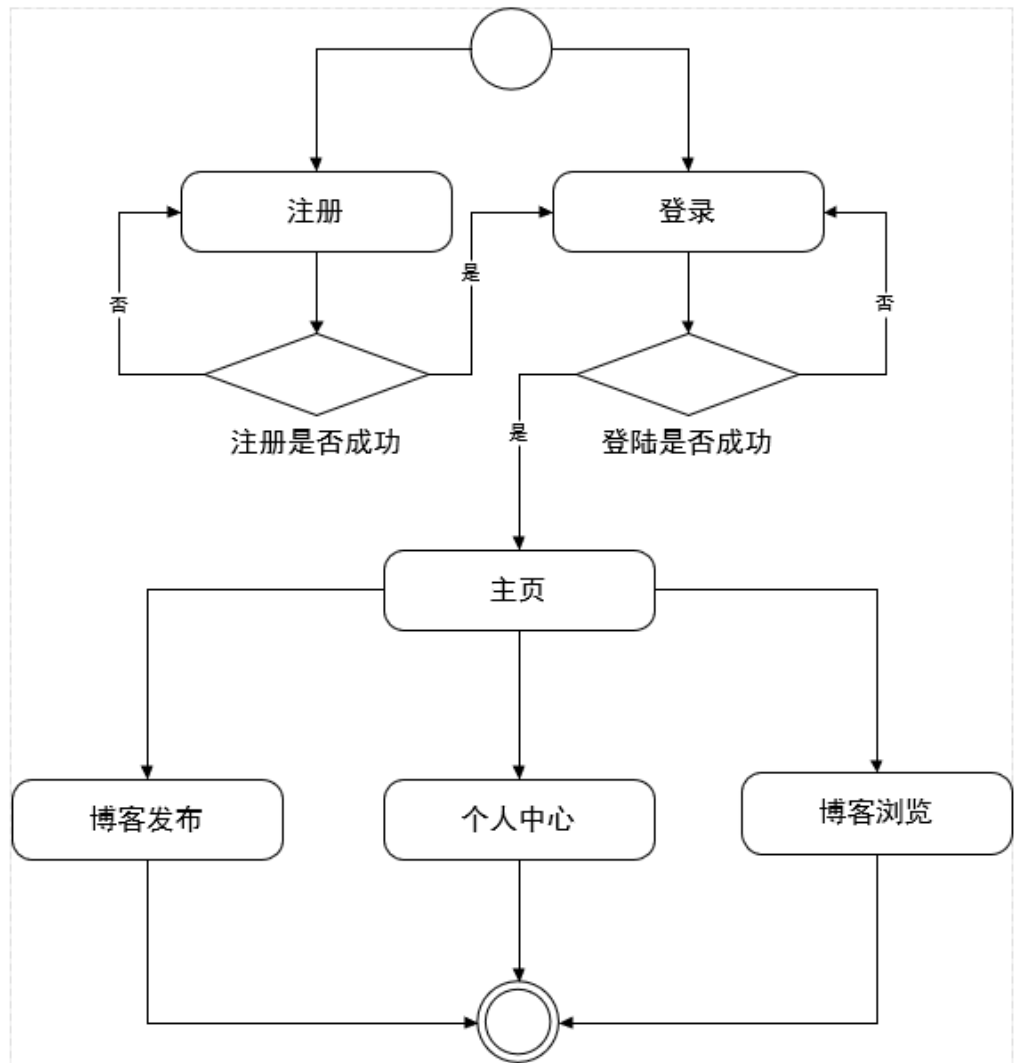


图 3.2 系统活动图

#### 3.2.1 登录注册模块

当用户注册时，点击注册按钮，页面将页面中的信息传入服务器中，服务器连接数据库访问数据，再由服务器对相关数据进行判断、审查，如果通过则将用户数据存入数据库中，如果不通过则返回失败原因。当用户登录时，点击登录按钮，页面将数据传给服务器，服务器连接数据库访问数据再由服务器进行判断，如果通过则登录成功跳转至主页，如果失败则返回错误信息，时序图如图 3.3 所示。

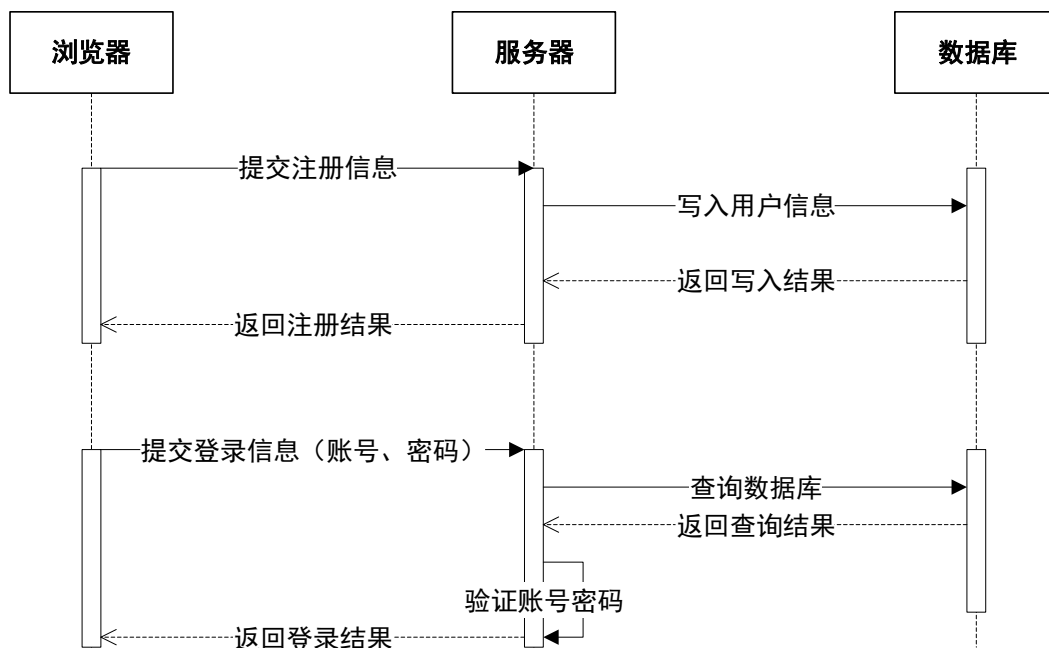


图 3.3 登录注册时序图

### 3.2.2 博客发布模块

用户登录之后，点击“写博客”按钮后跳转到写博客页面，编写博客标题、摘要、正文，然后添加标签，分类点击发布按钮，页面将数据传输给服务器，服务器审核、判断博客是否符合标准，若符合标准则将数据写入数据库中，返回成功信息跳转到博客详情页面，若不符合标准则返回结果和错误信息，时序图如图 3.4 所示。

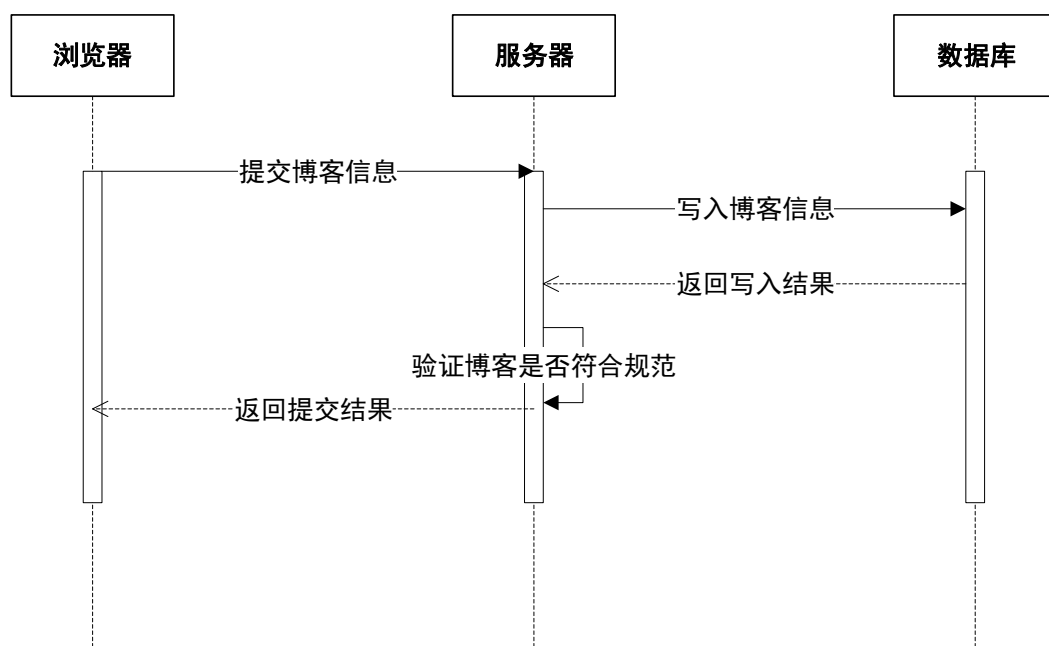


图 3.4 博客发布时序图



### 3.2.3 博客查看模块

用户登录成功后,跳转至主页,向服务器发出 GET 请求,服务器访问数据库,查找所有博客,然后将数据装入 Model 然后返回 view。主页接受到服务器传来的 Model,将 Model 里的数据遍历显示出来形成博客列表。用户点击博客列表中的数据,向服务器传入点击博客的 id,服务器访问数据库根据博客 id 查找出博客数据,将数据返回博客详情页面,博客详情页面将相关数据展示出来。通过博客的 id,服务器查找出相应评论数据,传输给博客详情页面,页面再将评论显示出来。

当用户点击点赞按钮时,向服务器发送 POST 请求,服务器访问数据库,将点赞数加一。当用户在评论框中填写数据并点击发表时,向服务器发送 POST 请求并将评论数据发送给服务器,服务器审核评论数据,审核通过则访问数据库将评论数据写入,若审核未通过返回错误信息,时序图如图 3.5 所示。

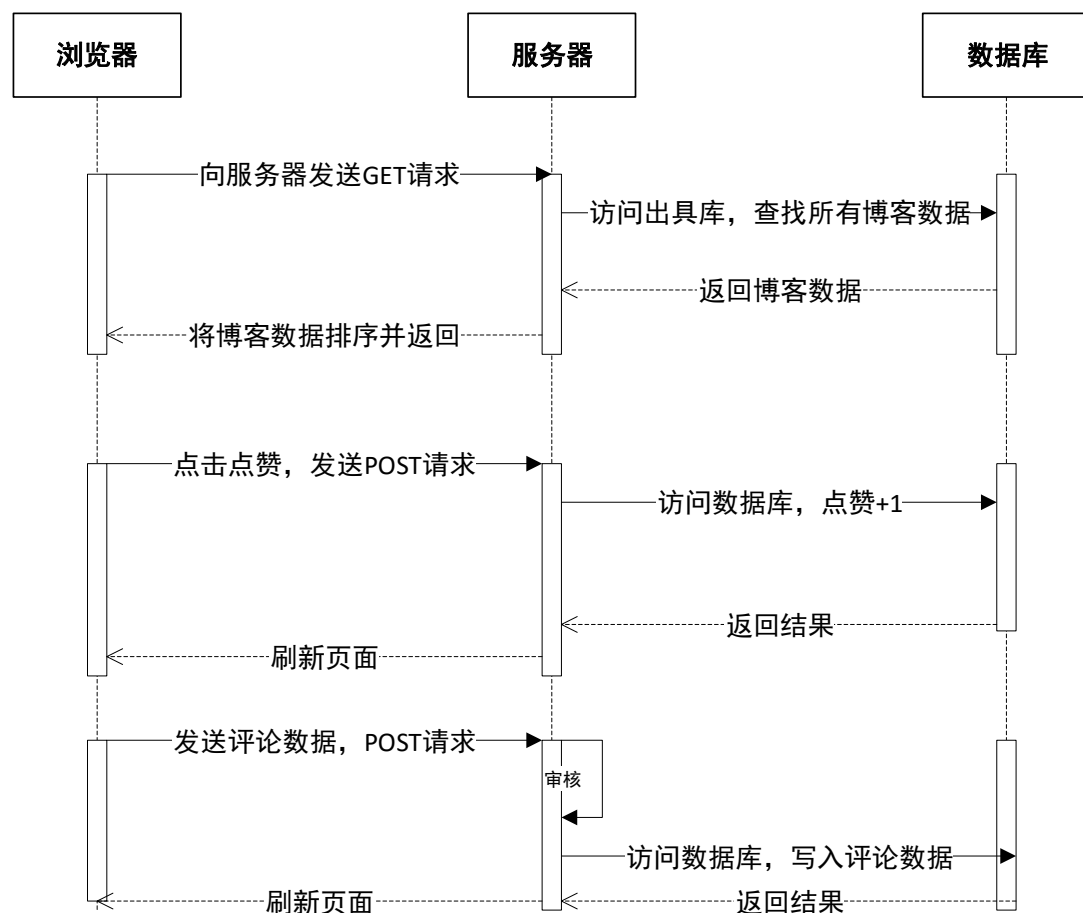


图 3.5 博客查看时序图

### 3.3 数据库设计

数据库设计是通过系统的需求,分析系统中参与的实体自身的属性以及实体之间的关系来设计数据库表。数据库设计是开发中至关重要的一步,数据库设计

一定要分析地透彻、建模地合理，否则在项目开发过程中再来修改数据库的话，往往会牵一发而动全身，造成巨大的工作量降低开发效率。

3.3.1 数据库模型

数据库的整体框架如图 3.6

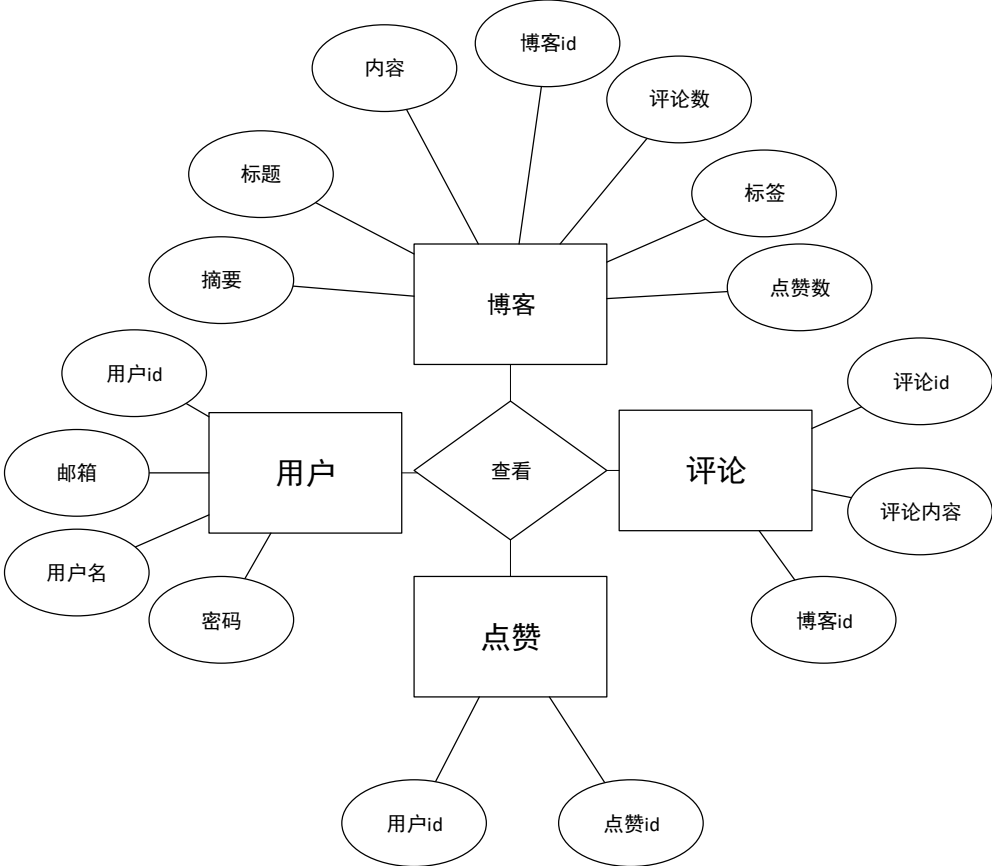


图 3.6 数据库 E-R 图

3.3.2 数据库表结构设计

3.3.1 展示了数据库的整体框架，接下来对每个表的具体结构进行详细设计。

表 3.1 用户 (user) 表

字段名	数据类型	约束	描述
id	bigint(20)	Auto Increment	用户 ID
avatar	varchar(200)	Allow Null	用户头像
email	varchar(50)		用户邮箱
name	varchar(20)		用户账号
password	varchar(100)		用户密码
username	varchar(20)		用户名称

表 3.2 权限 (authority) 表

字段名	数据类型	约束	描述
-----	------	----	----

id	bigint(20)	Auto Increment	权限 ID
name	varchar(255)		权限名称

表 3.3 用户\_权限 (user\_authority) 表

字段名	数据类型	约束	描述
user_id	bigint(20)	FOREIGN KEY	用户 ID
authority_id	bigint(20)	FOREIGN KEY	权限 ID

表 3.4 博客 (blog) 表

字段名	数据类型	约束	描述
id	bigint(20)	Auto Increment	博客 ID
comment_size	int(11)	Allow Null	评论数
content	longtext		博客正文
create_time	datetime		创建时间
html_content	longtext		html_content
read_size	int(11)	Allow Null	浏览次数
summary	varchar(300)		摘要
tags	varchar(100)	Allow Null	标签
title	varchar(50)		标题
vote_size	int(11)	Allow Null	点赞数
catalog_id	bigint(20)	Allow Null FOREIGN KEY	分类 ID
user_id	bigint(20)	FOREIGN KEY	用户 ID

表 3.5 评论 (comment) 表

字段名	数据类型	约束	描述
id	bigint(20)	Auto Increment	评论 ID
content	varchar(500)		评论内容
create_time	datetime		创建时间
user_id	bigint(20)	FOREIGN KEY	用户 ID

表 3.6 博客\_评论 (blog\_comment) 表

字段名	数据类型	约束	描述
blog_id	bigint(20)	FOREIGN KEY	博客 ID

comment_id	bigint(20)	FOREIGN KEY	评论 ID
------------	------------	-------------	-------

表 3.7 点赞 (vote) 表

字段名	数据类型	约束	描述
id	bigint(20)	Auto Increment	点赞 ID
create_time	datetime		创建时间
user_id	bigint(20)	FOREIGN KEY	用户 ID

表 3.8 博客\_点赞 (blog\_vote) 表

字段名	数据类型	约束	描述
blog_id	bigint(20)	FOREIGN KEY	博客 ID
vote_id	bigint(20)	FOREIGN KEY	点赞 ID

表 3.9 分类 (catalog) 表

字段名	数据类型	约束	描述
id	bigint(20)	Auto Increment	分类 ID
name	varchar(30)		类名
user_id	bigint(20)	Allow Null	用户 ID

## 4 系统详细设计与实现

### 4.1 开发环境的搭建

#### 4.1.1 Mean 项目管理工具

Maven 项目管理工具可以通过一些很简单的代码实现整个项目的构建，使用 Maven 可以很方便快速地构建开发环境，同时 Maven 非常易于管理且有着十分强大的资源库。

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>1.5.9.RELEASE</version>
  <relativePath/> <!-- lookup parent from repository -->
</parent>
```

通过上述代码搭建一个 Spring Boot 项目。

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

添加上述代码，实现了 Spring Boot 对 web 支持。

Spring Boot 很好地集成了 Thymeleaf 模板引擎，我们只要在 pom.xml 中加入以上依赖就可以添加 Thymeleaf 支持。

#### 4.1.2 IntelliJ IDEA

开发此系统使用的是 JetBrains 公司旗下的 IntelliJ IDEA 集成开发工具，

```
<!--集成 thymeleaf-->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
```

IntelliJ 在业界被公认为最好的 java 开发工具之一，尤其在智能代码助手、代码自动提示、重构、J2EE 支持、各类版本工具(git、svn、github 等)、JUnit、CVS 整合、代码分析、创新的 GUI 设计等方面的功能可以说是超常的<sup>[7]</sup>。

#### 4.1.3 MySQL

系统的数据库采用的是 Oracle 旗下的关系型数据库 MySQL。由于 MySQL 的体积小，速度快，免费等特点，所有选用 MySQL 作为系统的数据资源库<sup>[6]</sup>。

```
# DataSource
spring.datasource.url=jdbc:mysql:///blog?useUnicode=true&characterEncoding=utf-8
spring.datasource.username=root
spring.datasource.password=653772348
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
```

在 application.properties 中配置好数据库的路径，驱动，以及账号密码，以连接 MySQL 数据库。

#### 4.1.4 JDK 1.8

JDK 是 java 语言开发工具包支持大部分平台，JDK1.8 是 JDK 的最新稳定版本<sup>[8]</sup>。

### 4.2 持久层的详细设计与实现

该系统采用了 Hibernate 框架来操纵数据库，Hibernate 的中心思想是 ORM（对象关系映射），将对数据库的操作转换为对 Java 对象的操作，从而简化开发。所以对象实体的设计就是数据库的设计，在 3.3 中已经进行了数据库设计，接下在将这些数据库都转化成实体对象，就可以完成数据库的构建了<sup>[5]</sup>。

#### 4.2.1 domian 实体的设计与实现

在整个系统的运行流程中，总共涉及了六个实体，分别是用户、权限、博客

评论、分类、点赞等六个对象，具体的类名如图 4.1 所示。

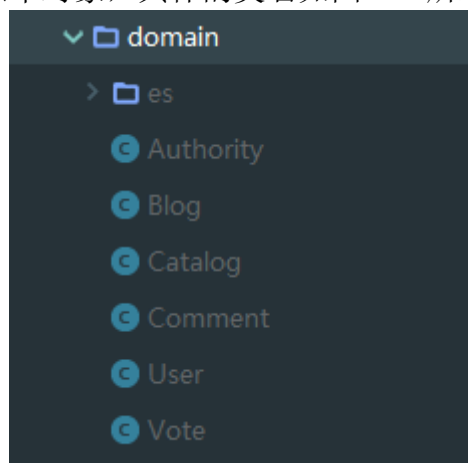


图 4.1 domain 实体设计图

要在实体与数据库各表之间建立 ORM 的关系，需要用到一些注解来将类与数据库联系到一起，以用户类为例关键代码如下：

```
@Entity//实体
public class User implements UserDetails,Serializable {

    private static final Long serialVersionUID = 1L;

    @Id//主键
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id; //唯一标识

    @NotEmpty(message = "名称不能为空")
    @Size(min=2,max = 20)
    @Column(nullable = false,length = 20)
    private String name; //名称

    @NotEmpty(message = "用户名不能为空")
    @Size(min=2,max = 20)
    @Column(nullable = false,length = 20,unique = true)
    private String username; //用户名

    @NotEmpty(message = "密码不能为空")
    @Size(min=6,max = 100)
    @Column(nullable = false,length = 100)
    private String password; //密码

    @Column(length = 200)
    private String portrait; //头像信息

    @NotEmpty(message = "邮箱不能为空")
    @Size(min=2,max = 50)
    @Email(message = "邮箱格式不正确")
    @Column(nullable = false,length = 50,unique = true)
    private String email; //邮箱
```

使用@ Entity 注解来标注此为一个实体对象，@id 标识着 id 为表的主键，@Column 代表表中的每一列，@Size 注解可以设置表中字段的长度。通过这些注解可以完成实体与表的映射。

实体类设计完之后，再通过仓库类来实现对数据库的具体操作，在这里使用了 Spring Data 框架，非常方便地实现了对数据库的操作。使用 Spring Data 框架后，只需要按照 Spring Data 的规范写出方法名就可以直接实现对数据库的操作，不用去写具体的方法，实例代码如下

```
public interface UserRepository extends JpaRepository<User, Long> {

    /**
     * 根据用户名查询用户列表
     * @author hao
     * @param [name, pageable]
     * @return
     */
    Page<User> findByNameLike(String name, Pageable pageable);

    /**
     * @author hao
     * @param [pageable]
     * @return
     */
    Page<User> findAll(Pageable pageable);

    /**
     * 根据用户账号查询用户
     * @author hao
     * @param [username]
     * @return com.lawliet.springboot.blog.domain.User
     */
    User findByUsername(String username);
}
```

只需要继承 JpaRepository 类就可以了，如方法 findByNameLike() 便会查找所有名称中带有形参 name 的用户，而 findAll() 方法可以查找所有的用户。每个实体对应一个数据库表，每个数据库表有一个对应的仓库类。具体的仓库如

图 4.2 所示。

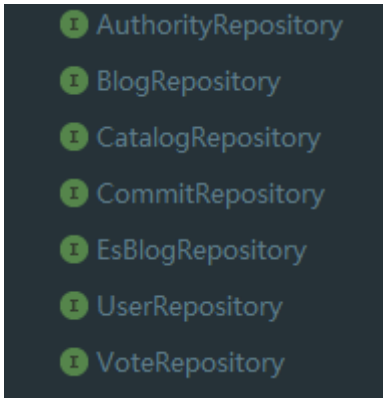


图 4.2 repository 仓库类设计图

4.3 登录注册模块的详细设计与实现

登录注册模块最重要的是其安全性，本系统作为一个自由的在线博客系统，即使用户没有登录，也可以阅读博客文章，只是无法进行点赞、评论等其他操作。具体的流程如图 5.3 所示。用户进入主页，选择登录或者注册。点击登录跳转至登录界面，点击登录，若登录成功进入主页。点击注册按钮，跳转至注册页面，填写注册信息，点击注册，注册成功则跳转至登录页面如图 4.3 所示

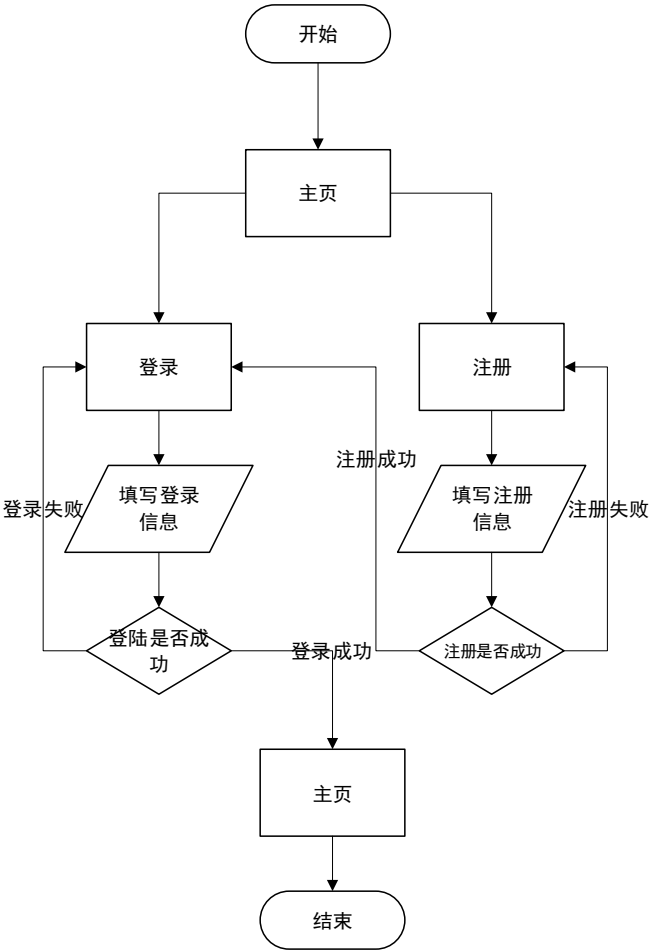


图 4.3 登录注册流程图



该系统的用户验证主要由 Spring security 安全框架控制，通过 Spring security 来控制用户认证，提升了系统整体的安全性和可靠性<sup>[9]</sup>，关键代码如下：

```
@Override
protected void configure(HttpSecurity http) throws Exception {
    http.authorizeRequests()
        .antMatchers( ...antPatterns: "/static/css/**", "/static/js/**",
            "/fonts/**", "/index")
        .permitAll()
        .antMatchers( ...antPatterns: "/admins/**").hasRole("ADMIN")
        .and()
        .formLogin()
        .loginPage("/login").failureUrl("/login-error")
        .and().rememberMe().key(KEY)
        .and().exceptionHandling().accessDeniedPage("/403") ;
}
```

通过撰写 SecurityConfig 类，继承 WebSecurityConfigurerAdapter 类，标识这是一个 Spring security 配置类，通过设置使得所有用户都可以访问静态资源，而管理员页面则只有拥有“ADMIN”角色的用户才能访问。设置登录页面为 /login，登录失败则跳转/login-error 页面。

```
public void configureGlobal(AuthenticationManagerBuilder auth)
    throws Exception {
    auth.userDetailsService(userDetailsService);
    auth.authenticationProvider(authenticationProvider());
}
```

通过在 SecurityConfig 类中重写 configureGlobal 方法来设置认证信息，关键代码如下。

最终实现页面如图 4.4 所示：

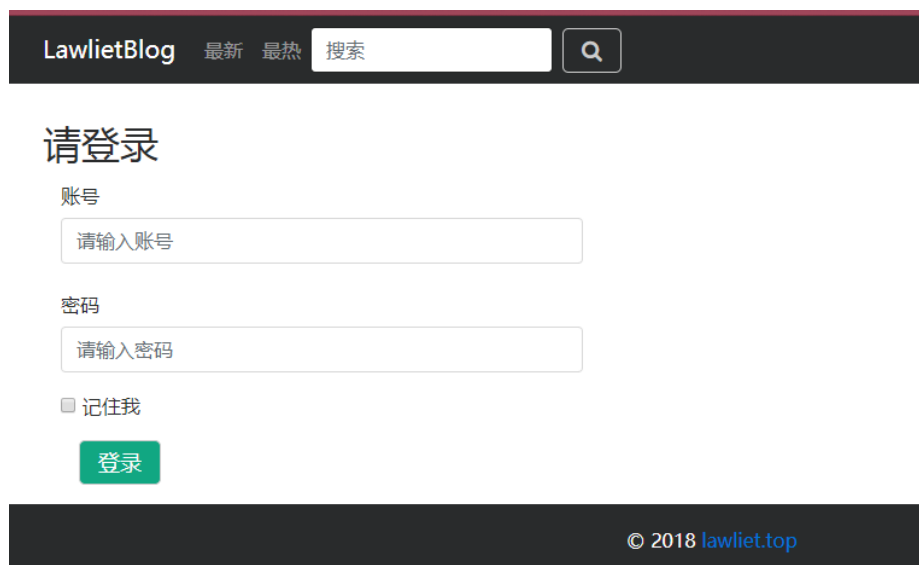


图 4.4 登录页面

## 4.4 博客发布模块的详细设计与实现

博客发布主要功能是编辑博客，发布博客。当用户点击写博客按钮时跳转至博客编辑页面，输入博客信息，为博客选择分类、设置标签，然后点击发布按钮，发布操作成功后返回发布博客的博客详情页面，具体流程图如图 4.5 所示。

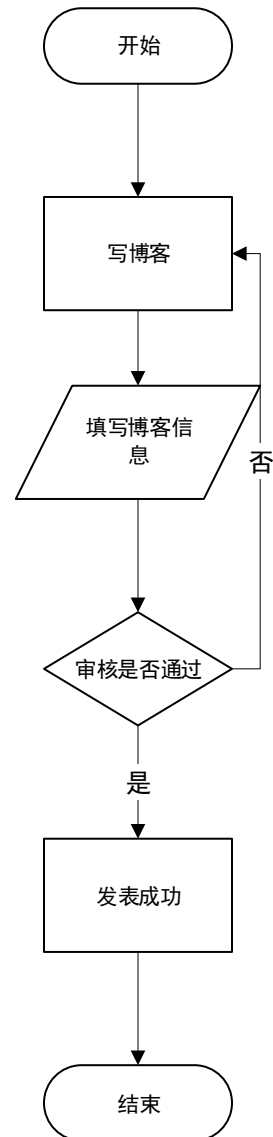


图 4.5 博客发布流程图

点击发布按钮向服务器发送 POST 请求，首先对 catalog 进行空处理，若 catalog 为空说明用户未给博客分类，此时将博客放入默认分类中，关键代码如下

下:

```
if (blog.getCatalog().getId() == null) {  
    User user = (User)userDetailsService.loadUserByUsername(username);  
    blog.setUser(user);  
    Catalog catalog = new Catalog(user, name: "默认分类");  
    catalogService.saveCatalog(catalog);  
    blog.setCatalog(catalog);  
    blogService.saveBlog(blog);  
    String redirectUrl = "/u/" + username + "/blogs/" + blog.getId();  
    return ResponseEntity.ok().body(new Response  
        ( success: true, message: "未选择分类, 为您放入默认分类", redirectUrl));  
}
```

操作成功后返回发布博客的博客详情页面, 通过 controller 类调用 service 类, 再通过 service 类调用 repository 类, 再由 repository 类对实体进行操作, 然后由实体映射数据库, 从而完成整个操作, 关键代码如下:

```
private UserDetailsService userDetailsService;
```

```
@Autowired  
public void setUserDetailsService  
    (UserDetailsService userDetailsService) {  
    this.userDetailsService = userDetailsService;  
}
```

```
private UserService userService;
```

```
@Autowired  
public void setUserService(UserService userService) {  
    this.userService = userService;  
}
```

```
private BlogService blogService;
```

```
@Autowired  
public void setBlogService(BlogService blogService) {  
    this.blogService = blogService;  
}
```

```
private CatalogService catalogService;
```

```
@Autowired  
public void setCatalogService  
    (CatalogService catalogService) {  
    this.catalogService = catalogService;  
}
```

在这一系列过程中，使用 Spring 依赖注入，减少了整个系统的耦合程度。依赖注入有两种方式，一种是通过构造方法注入，另一种是通过 set 方法注入。此系统采用的是 set 方法的依赖注入，因为使用 set 方法依赖注入可以防止循环依赖问题的产生，增加系统的稳定性。将需要用到的 UserDetailsService、userService、blogService、catalogService 等使用 set 方法注入。

博客发布具体的页面如图 4.6 所示。

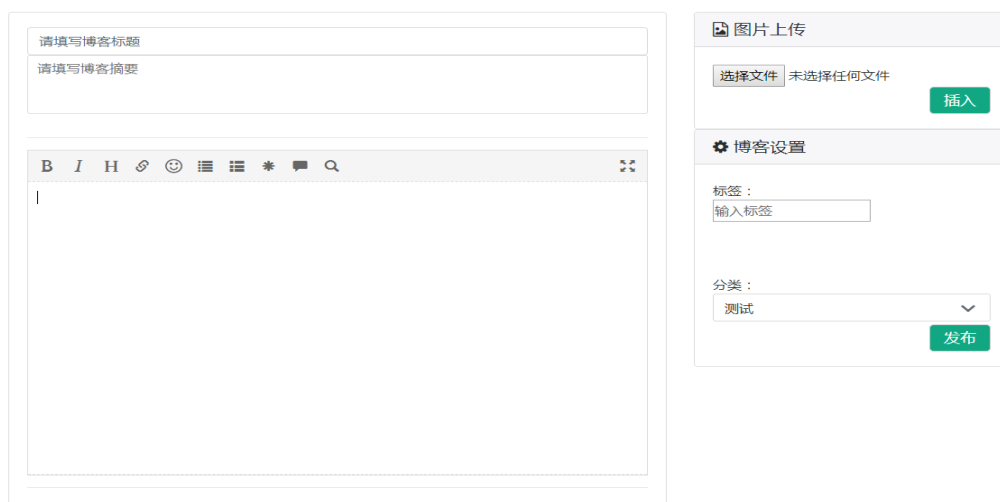


图 4.6 博客发布页面

左侧用来填写博客信息，右侧可以给博客设置标签，设置分类。此处的分类是通过用户 ID 查找该用户的分类，然后遍历出来，关键代码如下：

```
分类:
<select id="catalogSelect"
        class="form-control form-control-chosen"
        data-placeholder="请选择">
    <option th:value="*{id}" th:text="*{name}"
            th:each="catalog : ${catalogs}"
            th:object="${catalog}"
            th:selected="${blog.catalog eq catalog }">
        java
    </option>
</select>
```

通过 Thymeleaf 的语法 `th:each="catalog : ${catalogs}"` 将 model 中的 catalogs 集合给遍历显示出来。

## 4.5 博客查看模块的详细设计与实现

当用户进入网站首页之后，服务器会查询所有博客将其显示在主页上。当用户没有登陆时，点击博客进入博客详情页面，可以查看博客内容，评论内容但无法进行点赞评论等操作，其中的权限控制是使用 Spring Security 控制的。当用户登录之后，可以在博客详情页面中进行评论，点赞，取消点赞等操作。具体的

流程如下图 4.7 所示

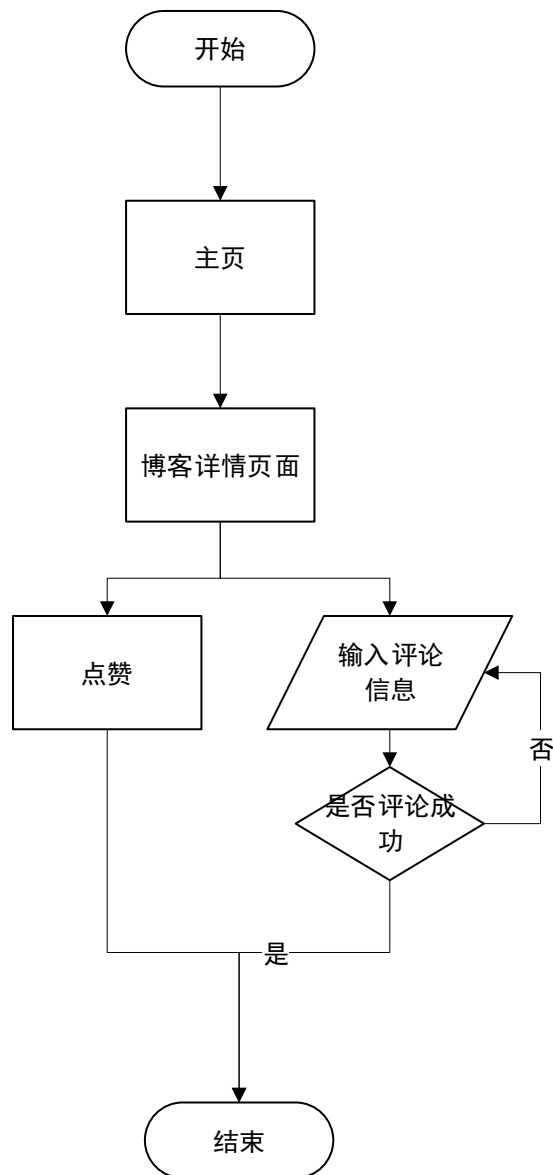


图 4.7 博客查看流程图

用户进入首页后，向服务器发送“/blogs” GET 请求，通过 BlogService 调用 BlogRepository 查询数据库，将查询到的所有 blog 放入 Page 然后显示在主页上。在 BlogController 里设置 Request 参数实现首页博客的最新最热排序，设置 order 实现最新最热排序，通过设置 keyword 实现关键词查询，关键代码如下：

```

@GetMapping
public String listEsBlogs(
    @RequestParam(value = "order", required = false, defaultValue = "new")
    String order,
    @RequestParam(value = "keyword", required = false, defaultValue = "")
    String keyword,
    @RequestParam(value = "async", required = false)
    boolean async,
    @RequestParam(value = "pageIndex", required = false, defaultValue = "0")
    int pageIndex,
    @RequestParam(value = "pageSize", required = false, defaultValue = "10")
    int pageSize,
    Model model) {

```

默认状态下 order 为 new，也就是按照时间先后排序，当点击最热按钮时，order 变为 hot，将执行 listHotestEsBlogs 方法，关键代码如下：

```

@Override
public Page<EsBlog> listHotestEsBlogs(String keyword, Pageable pageable) {
    Sort sort = new Sort(Sort.Direction.DESC,
        ...properties: "readSize",
        "commentSize",
        "voteSize",
        "createTime");
    if (pageable.getSort() == null) {
        pageable = new PageRequest
            (pageable.getPageNumber(),
            pageable.getPageSize(), sort);
    }

    return esBlogRepository.
        findDistinctEsBlogByTitleContainingOrSummary
            (keyword, keyword);
}

```

通过调用 esBlogRepository 中的方法查询数据库，然后排序显示在主页上。在博客详情页面中，只有登录的用户可以点赞和评论。在 CommentController 和 VoteController 中通过判断 SecurityContextHolder.getContext().getAuthentication() 是否为 null 来确认用户是否登录。当点击评论或者点赞时，会向服务器发送 POST 请求，向数据库中添加相应信息。

最终实现的页面如图 4.8 所示：



图 4.8 博客列表

#### 4.6 个人中心模块的详细设计与实现

用户登录系统后可以进入自己的个人中心，个人中心分为两块分别是个人主页和个人设置。在个人主页页面，服务器通过用户 ID 查询所有具有相同用户 ID 的博客并显示出来，用户可以在个人主页添加自己的博客分类，点击添加分类按钮，向服务器发送 POST 请求，向数据库中添加一条分类。具体流程如下图 4.9 所示。

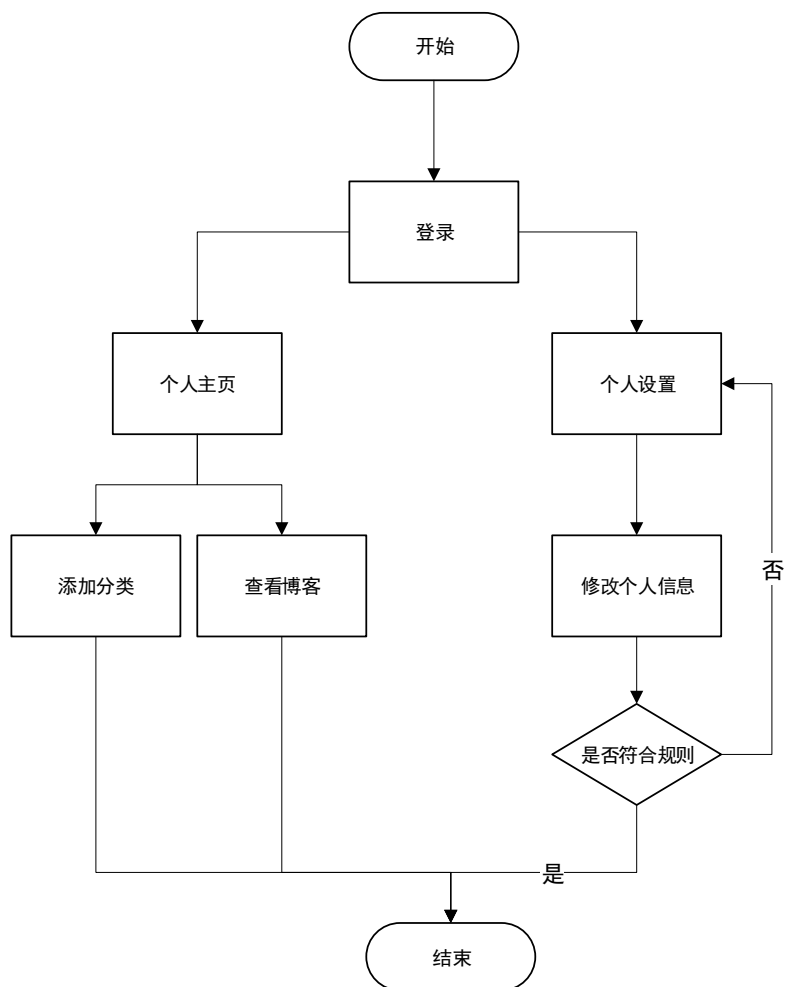


图 4.9 个人中心流程图

用户进入个人设置中修改个人设置时，需要先判断当前登录用户是否认证，所以只有自己可以修改自己的个人设置关键代码如下：

```

@PostMapping("/{username}/profile")
@PreAuthorize("authentication.name.equals(#username)")
public String saveProfile(@PathVariable("username")
                          String username, User user) {
    User originUser = userService.findUserById(user.getId());
    originUser.setEmail(user.getEmail());
    originUser.setName(user.getName());
    // 判断密码是否做了变更
    String rawPassword = originUser.getPassword();
    PasswordEncoder encoder = new BCryptPasswordEncoder();
    String encodePasswd = encoder.encode(user.getPassword());
    System.out.println("修改过密码: " + rawPassword +
                       "    原始密码: " + encodePasswd);
    boolean isMatch = encoder.matches(rawPassword, encodePasswd);
    if (!isMatch) {
        originUser.setEncodePassword(user.getPassword());
    }

    userService.saveOrUpdateUser(originUser);
    return "redirect:/u/" + username + "/profile";
}

```



最终实现的页面如图 4.10 所示。



⚙ 个人设置

账号

admin

邮箱

653772348@qq.com

姓名

管理员

密码

\*\*\*\*\*

保存

图 4.10 个人设置页面图

5 系统运行及测试

5.1 测试环境

表 5.1 测试环境表

硬件配置	计算机类型	微星笔记本电脑
	CPU	Intel Core i7
	内存	DDR3 8G
	硬盘	Lenovo SSD 128G
软件配置	Windows10 操作系统 IntelliJ IDEA	

5.2 系统单元测试

在线博客系统的各个模块的测试流程以及结果如下表 5.2 所示。

表 5.2 在线博客系统单元测试表

序号	功能模块	功能	操作步骤	预期结果	实际结果
1	登录注册模块	注册	1、 打开在线博客系统； 2、 点击“注册”按钮进入注册页面； 3、 向信息框中输入注册信息； 4、 点击“注册”按钮；	当用户输入注册信息为空时，或者注册信息不符合规定时显示注册失败反之显示注册成功并跳转到登录界面。	一致
2	登录注册模块	登录	5、 打开在线博客系统； 6、 点击“登录”按钮进入登录页面； 7、 向信息框中输入登录信息 8、 点击“登录”按钮；	如果登录信息正确则跳转至主页，反之则显示注册失败。	一致

3	博客查看模块	查看博客详情	1、 进入系统首页； 2、 点击博客列表中的博客；	跳转到对应博客的博客详情页面。	一致
4	博客查看模块	添加点赞	1、 进入系统首页； 2、 点击博客列表中的博客，进入博客详情页面； 3、 点击“点赞”按钮；	点赞成功，博客点赞数加一，点赞按钮变成取消点赞按钮。	一致
5	博客查看模块	取消点赞	1、 进入系统首页； 2、 点击博客列表中的博客，进入博客详情页面； 3、 点击“取消点赞”按钮；	取消点赞成功，博客点赞数减一，取消点赞按钮变成点赞按钮。	一致
6	博客查看模块	添加评论	1、 进入系统首页； 2、 点击博客列表中的博客，进入博客详情页面； 3、 在评论框中输入评论信息； 4、 点击“发表评论”按钮；	提示“评论成功”，在博客详情页面显示评论内容。	一致
7	博客查看模块	删除评论	1、 进入系统首页； 2、 点击博客列表中的博客，进入博客详情页面； 4、 点击评论右侧的“删除”按钮；	删除成功，博客详情页面的评论内容消失。	一致
8	博客发布模块	发布博客	1、 进入系统首页； 2、 点击“写博客”按钮；	进入发布博客编辑页面。	一致
9	博客发布模块	添加标签	1、 进入系统首页； 2、 点击“写博客”按钮进入博客编辑页面； 3、 输入博客信息，添加标签； 4、 点击“发布”按钮；	发布成功，跳转至博客详情页面，博客标签显示为刚才添加的标签。	一致
10	博客发布模块	为博客添加分类	1、 进入系统首页； 2、 点击“写博客”按钮进入博客编辑页面； 3、 输入博客信息，选择分类； 4、 点击“发布”按钮；	发布成功，跳转至博客详情页面，博客分类显示为刚才选择的分类。	一致
11	个人中心模块	个人主页	1、 进入系统首页； 2、 点击“个人主页”按钮进入个人主页页面；	个人主页显示用户名，个人邮箱，博客分类，以及个人博客。	一致

12	个人中心模块	添加博客分类	1、 进入系统首页； 2、 点击“个人主页”按钮进入个人主页页面； 3、 点击“添加分类按钮” 4、 输入分类名称 5、 点击“提交”按钮	提示“添加分类成功”，博客分类中多处刚添加分类	一致
13	个人中心模块	个人设置	1、 进入系统首页； 2、 点击“个人设置”按钮进入个人设置页面；	显示个人设置页面，页面上显示账号名、邮箱、姓名、密码	一致
14	个人中心模块	修改个人设置	1、 进入系统首页； 2、 点击“个人设置”按钮进入个人设置页面； 3、 编辑个人设置信息； 4、 点击“保存”按钮；	显示“修改成功”，并刷新页面，页面上的信息已经刷新。	一致

### 5.3 系统测试结果

在线博客系统能够满足用户登录注册、发布博客、编辑博客、查看博客、点赞、评论等功能，界面十分友好，功能按钮简洁明了，布局合理清晰，基本实现了预期的效果。同时该系统在稳定性和可移植性上表现良好，由于使用的 Spring Boot 框架，只需运行 jar 包即可启动系统。

综上所述，在线博客系统是一个良好的博客平台，但不适合大范围内使用，适合小规模使用。

## 结 语

随着博客逐渐流行，写博客成为人们日常生活的一部分，各大博客网站因运而生，但是很多博客网站的用户体验并不友好，所以本文的作者设计并实现了一个界面简洁，功能丰富，开放自由的在线博客系统。

本文按照软件开发的流程，对博客系统的需求，设计与实现进行了详细的讲解与分析，对用户管理、博客管理、评论管理、点赞管理、分类管理、标签管理几个模块进行了详细的设计与实现，最后对在线博客系统进行了严格的单元测试并分析了测试结果得出结论。

通过此次在线博客系统的设计与实现，作者对一个软件的开发流程有了更深刻的体会。经过了几个月的努力，作者最终完成了在线博客系统并将其部署在linux服务器上，在后期的实际使用中在线博客系统表现良好，但仍然存在着不足，作者会不断地对其进行改进，希望有更多的人来使用在线博客系统。

## 参考文献

- [1] 杜运年. 从博客到微信: 网络社交的窄化[J]. 新媒体研究, 2017, 3(2):19-20
- [2] 常松主编. 博客舆情的分析与研判[M]. 社会科学文献出版社, 2014.
- [3] 李刚,魏康林,王梅等.医院专家博客在医学信息中的应用[J].西南军医,2009,11(3):535-536.
- [4] 王福强著.SpringBoot 揭秘:快速构建微服务体系:microservices quick starter[M]. 机械工业出版社, 2016.
- [5] 黄佩虹,,张冰晶编著.精通 Hibernate Java 数据库持久层开发核心编程[M]. 清华大学出版社, 2009.
- [6] 任进军,,林海霞主编.MySQL 数据库管理与开发 慕课版[M]. 人民邮电出版社, 2017.
- [7] IntelliJIDEA2016. 2 功能升级[J]. 电脑编程技巧与维护, 2016, 0(14):4-4.
- [8] 林信良著. JAVA JDK 8 学习笔记[M]. 清华大学出版社, 2015.
- [9] (美),Nicholas,S. 等. Java Web 高级编程:涵盖 WebSockets、Spring Framework、JPA Hibernate 和 Spring Security[M]. 清华大学出版社, 2015.
- [10] 朱林编著.Elasticsearch 技术解析与实战[M]. 机械工业出版社, 2017.

## 致 谢

该论文是在成都信息工程大学软件工程学院郑文斌老师的悉心指导和严格要求下完成的。从设计选题到论文完成这一过程中,无一不包含着郑老师的心血。在整个毕业设计期间,郑老师的指导与教诲一直陪伴着我,每当我遇到解决不了的问题时,郑老师便会一步步地引导我去分析问题、解决问题,而不是直接给出问题的答案。在郑老师的引导下,我学会了如何客观地分析问题以及如何有效地解决问题。在此,我向郑老师表达诚挚的谢意。

在论文即将完成之际,我由衷的感谢指导老师对我的指导与教诲,感谢学院对我的栽培,感谢同学对我的帮助与鼓励,感谢所有支持我的人。

作者简介:

姓 名: 孔浩

性别: 男

出生年月: 1996.12

民族: 汉

E-mail: cz653772348@163.com

## 声 明

本论文的工作是 2017 年 12 月至 2018 年 5 月在成都信息工程大学软件工程学院完成的。文中除了特别加以标注地方外，不包含他人已经发表或撰写过的研究成果，也不包含为获得成都信息工程大学或其他教学机构的学位或证书而使用过的材料。

关于学位论文使用权和研究成果知识产权的说明：

本人完全了解成都信息工程大学有关保管使用学位论文的规定，其中包括：

(1) 学校有权保管并向有关部门递交学位论文的原件与复印件。

(2) 学校可以采用影印、缩印或其他复制方式保存学位论文。

(3) 学校可以学术交流为目的复制、赠送和交换学位论文。

(4) 学校可允许学位论文被查阅或借阅。

(5) 学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

除非另有科研合同和其他法律文书的制约，本论文的科研成果属于成都信息工程大学。

特此声明！

作者签名：

2018 年 06 月 日