**分类号：TP311.5 U D C：D10621-408-(2018)3479-0**

**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。

本文严格按照软件工程的一般流程，从头至尾地对博客系统的需求，设计与实现进行了全面的讲解与分析。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，本文对上述功能的具体设计进行了详细的分析

该博客采用了先进的Spring Boot + Hibernate框架技术进行开发，前端页面使用了Thymeleaf模板引擎，后台数据库采用的时Oracle旗下的关系型数据库mysql，服务器端使用Apache的Tomcat，采用了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。

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

**Design and Implementation of Online Blog System**

**Summary**

**Abstract**

With the rapid development of Internet technology, people's lives are closely integrated with the Internet. Everywhere is the combination of the Internet and everyday life. In the same way, the Internet makes people's daily learning more convenient and efficient. Through the Internet, people can acquire knowledge quickly and effectively. A blog is a convenient way of learning. Blogs are always the embodiment of the spirit of sharing and sharing. Through blogs, we can freely express ideas, share knowledge, and communicate with others. This article strictly follows the general flow of software engineering, from the beginning to the end of the blog system requirements, design and implementation of a comprehensive explanation and analysis. The main functions of the blog system are divided into several modules: user management, blog management, comment management, point-alike management, classification management, and tag management. The detailed design of the above functions is analyzed in this paper. The blog is developed using the advanced Spring Boot + Hibernate framework technology. The front page uses the Thymeleaf template engine. The back-end database uses Oracle's relational database mysql. The server uses Apache's Tomcat. It uses the Spring security security framework. Security settings, using the ElasticSearch search engine to achieve full-text search, development tools using JetBrains company's IntelliJ IDEA

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

**目 录**

论文总页数：37页

[1 引言 1](#_Toc513813650)

[1.1 课题背景 1](#_Toc513813651)

[1.2 课题的研究意义 1](#_Toc513813652)

[1.3 国内外研究现状 1](#_Toc513813653)

[1.4 本文的基本结构与主要工作 1](#_Toc513813654)

[2 理论基础与相关技术的分析 2](#_Toc513813655)

[2.1 B/S结构简介 2](#_Toc513813656)

[2.2 Spring Boot技术简介 2](#_Toc513813657)

[2.3 Hibernate简介 2](#_Toc513813658)

[2.4 ElasticSearch简介 3](#_Toc513813659)

[2.5 IntelliJ IDEA简介 3](#_Toc513813660)

[3 系统需求分析 4](#_Toc513813661)

[3.1 需求概述 4](#_Toc513813662)

[3.2 功能性需求 4](#_Toc513813663)

[3.2.1 用例模型 4](#_Toc513813664)

[3.2.2 用例描述 5](#_Toc513813665)

[3.3 非功能性需求 9](#_Toc513813666)

[4 系统模块划分与概要设计 9](#_Toc513813667)

[4.1 系统模块划分 9](#_Toc513813668)

[4.2 系统模块设计 10](#_Toc513813669)

[4.2.1 登录注册模块 11](#_Toc513813670)

[4.2.2 博客发布模块 12](#_Toc513813671)

[4.2.3 博客查看模块 13](#_Toc513813672)

[4.3 数据库设计 13](#_Toc513813673)

[4.3.1 数据库模型 14](#_Toc513813674)

[4.3.2 数据库表结构设计 14](#_Toc513813675)

[4.4 系统页面设计 16](#_Toc513813676)

[5 系统详细设计与实现 20](#_Toc513813677)

[5.1 开发环境的搭建 20](#_Toc513813678)

[5.1.1 Mean项目管理工具 20](#_Toc513813679)

[5.1.2 IntelliJ IDEA 20](#_Toc513813680)

[5.1.3 MySQL 20](#_Toc513813681)

[5.1.4 Mean项目管理工具 21](#_Toc513813682)

[5.2 持久层的详细设计与实现 21](#_Toc513813683)

[5.2.1 domian实体的设计与实现 21](#_Toc513813684)

[5.3 登录注册模块的详细设计 24](#_Toc513813685)

[5.4 博客发布模块的详细设计 25](#_Toc513813686)

[5.5 博客查看模块的详细设计 28](#_Toc513813687)

[5.6 个人中心模块的详细设计 28](#_Toc513813688)

[6 系统运行及测试结果 28](#_Toc513813689)

[结 论 29](#_Toc513813690)

[参考文献 30](#_Toc513813691)

[致 谢 31](#_Toc513813692)

[声 明 32](#_Toc513813693)

# 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，博客成为了一种重要的网络交流方式。我们可以通过博客查找、访问到全球互联网中最有价值的信息、知识与资源，也可以将你个人工作过程、生活故事、思想历程、闪现的灵感等及时记录和发布。写博客成为了人们日常生活的一部分。

## 1.2 课题的研究意义

本系统采用了Sping 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内容。使用BootStrap构建的响应式布局，很好的兼容了移动设备的访问需求。本文开发的博客系统为广大网友提供了一个自由，和谐的交流空间，用户可以自由的浏览文章，评论，点赞，管理员也可以对整个博客系统进行管理。

## 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”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。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。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 理论基础与相关技术的分析

## 2.1 B/S结构简介

B/S结构是WEB兴起后的一种网络结构模式，WEB浏览器是客户端最主要的应用软件。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。客户机上只要安装一个浏览器，服务器安装数据库。浏览器通过Web Server 同数据库进行数据交互。

其主要特点：

* 维护和升级方式简单。B/S架构的产品明显体现着更为方便的特性。对一个稍微大一点单位来说，系统管理人员如果需要在几百甚至上千部电脑之间来回奔跑，效率和工作量是可想而知的，但B/S架构的软件只需要管理服务器就行了，所有的客户端只是浏览器，根本不需要做任何的维护。
* 成本降低，选择更多。使用B/S架构的应用管理软件，只需安装在Linux服务器上即可，而且安全性高。所以服务器操作系统的选择是很多的。
* 应用服务器运行数据负荷较重。由于B/S架构管理软件只安装在服务器端上，导致应用服务器运行数据负荷较重，一旦发生服务器“崩溃”等问题，后果不堪设想。

## 2.2 Spring Boot技术简介

Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。

其主要特点：

* 创建独立的Spring应用程序。
* 嵌入的Tomcat，无需部署WAR文件
* 自动配置Spring
* 提供生产就绪型功能，如指标，健康检查和外部配置
* 绝对没有代码生成和对XML没有要求配置

## 2.3 Hibernate简介

Hibernate是一个开放源代码的对象关系映射框架，它对JDBC进行了非常轻量级的对象封装，它将POJO与数据库表建立映射关系，是一个全自动的orm框架，hibernate可以自动生成SQL语句，自动执行，使得Java程序员可以随心所欲的使用对象编程思维来操纵数据库。

其主要特点：

* 将对数据库的操作转换为对Java对象的操作，从而简化开发。通过修改一个“持久化”对象的属性从而修改数据库表中对应的记录数据。
* 提供线程和进程两个级别的缓存提升应用程序性能。
* 有丰富的映射方式将Java对象之间的关系转换为数据库表之间的关系。
* 屏蔽不同数据库实现之间的差异。在Hibernate中只需要通过“方言”的形式指定当前使用的数据库，就可以根据底层数据库的实际情况生成适合的SQL语句。
* 非侵入式：Hibernate不要求持久化类实现任何接口或继承任何类，POJO即可。

## 2.4 ElasticSearch简介

ElasticSearch是一个基于Lucene的搜索服务器。它提供了一个分布式多用户能力的全文搜索引擎，基于RESTful web接口。Elasticsearch是用Java开发的，并作为Apache许可条款下的开放源码发布，是当前流行的企业级搜索引擎。设计用于云计算中，能够达到实时搜索，稳定，可靠，快速，安装使用方便。

## 2.5 IntelliJ IDEA简介

IntelliJ IDEA是java语言开发的集成环境，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、Ant、JUnit、CVS整合、代码审查、 创新的GUI设计等方面的功能可以说是超常的。IDEA是JetBrains公司的产品，这家公司总部位于捷克共和国的首都布拉格，开发人员以严谨著称的东欧程序员为主。

其主要特点：

* 智能的选取。在很多时候我们要选取某个方法，或某个循环或想一步一步从一个变量到整个类慢慢扩充着选取，IDEA就提供这种基于语法的选择，在默认设置中Ctrl+W，可以实现选取范围的不断扩充，这种方式在重构的时候尤其显得方便。
* 丰富的导航模式。IDEA提供了丰富的导航查看模式，例如Ctrl+E显示最近打开过的文件，Ctrl+N显示你希望显示的类名查找框（该框同样有智能补充功能，当你输入字母后IDEA将显示所有候选类名）。在最基本的project视图中，你还可以选择多种的视图方式。
* 完美的自动代码完成。智能检查类中的方法，当发现方法名只有一个时自动完成代码输入，从而减少剩下代码的编写工作。
* 智能代码。自动检查代码，发现与预置规范有出入的代码给出提示，若程序员同意修改自动完成修改。

# 3 系统需求分析

## 3.1 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。

## 3.2 功能性需求

### 3.2.1 用例模型

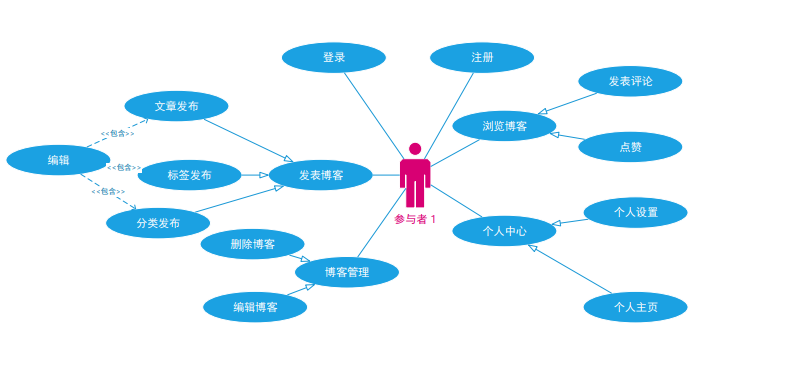
上述功能模块虽多，但大体分为四个部分，登录注册，个人中心，博客的发布，博客的查看以及互动（评论，点赞）。通过对功能的分析，细化得到了如图3.1所示的系统用例图。

图 3.1 系统用例图

### 3.2.2 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3.2所示。

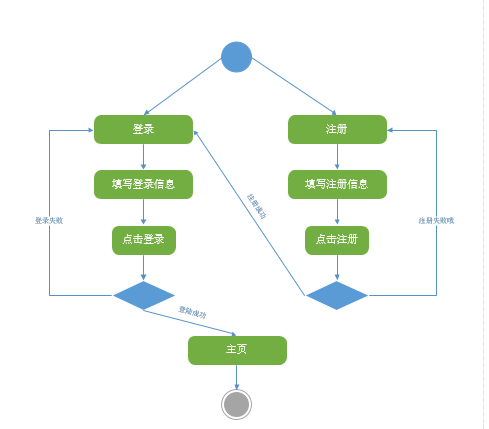


图 3.2 登录注册活动图

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原因

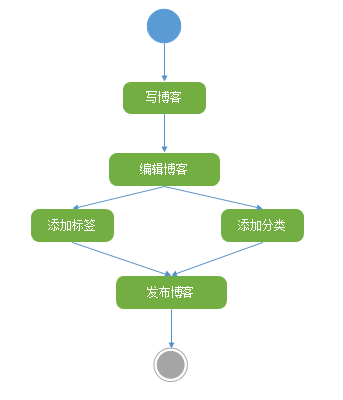
* 发表博客用例：用户可以编写博客标题，博客摘要，博客正文然后发布博客，在发布博客时还可以给博客添加标签、分类。

图 3.3 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3.4.

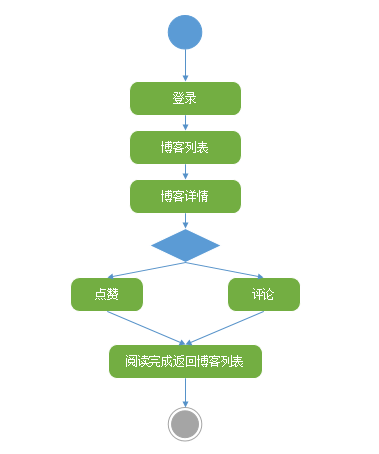


图 3.4 博客查看活动图

活动流程：

用户登录成功后，显示博客列表

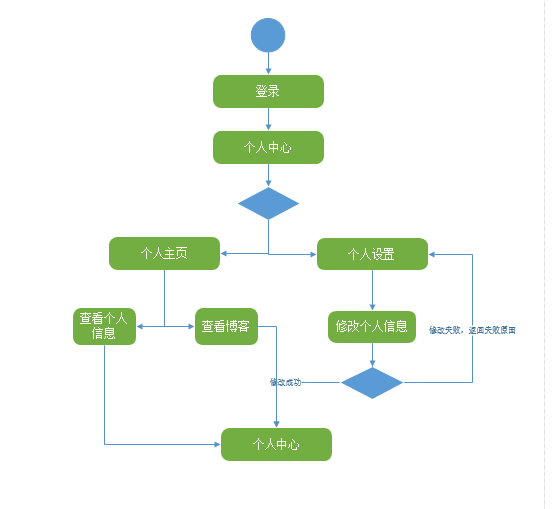
1.点击某个博客，进入博客详情页面

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3.5

图 3.5 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
2. 显示个人信息，发表过的博客
3. 点击个人设置
4. 显示个人信息
5. 修改个人信息，点击保存按钮
6. 保存成功，返回个人设置页面
7. 保存失败，返回个人设置页面显示失败原因

## 3.3 非功能性需求

非功能性需求是指依一些条件判断系统运作情形或其特性，而不是针对系统特定行为的需求。包括安全性、可靠性、互操作性、健壮性、易使用性、可维护性、可移植性、可重用性、可扩充性。

* 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。
* 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扩展。
* 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。

# 4 系统模块划分与概要设计

## 4.1 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：

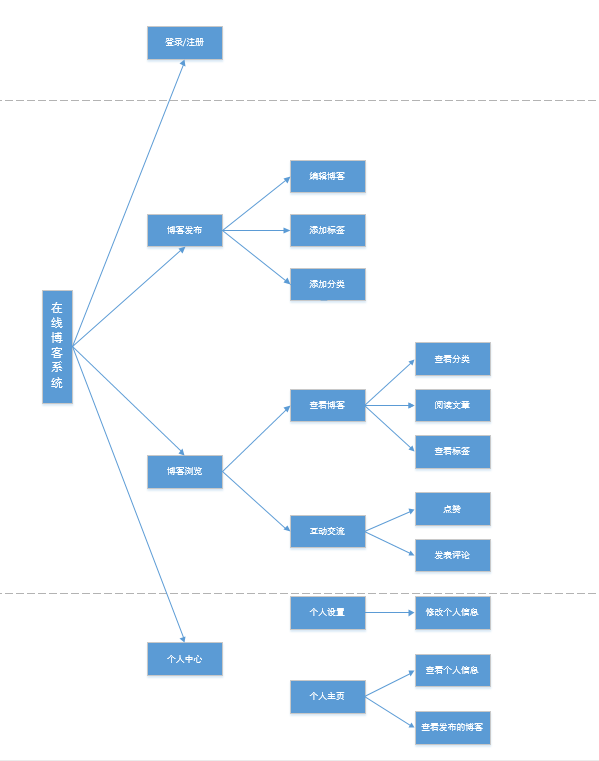
* 登录注册：用户登录需要填写账号和密码，注册时需要填写账号、邮箱、姓名和密码，账号至少3个字符，至多20个；邮箱必须为正确邮箱格式；姓名至少2个字符，至多20个；密码可以为字母或特殊符号和数字结合。
* 个人中心：个人主页可以查看用户的基本信息，以及用户曾经发表的文章，可以编辑，修改，删除自己的博客分类。个人设置可以对自己的个人信息（邮箱，姓名，密码，头像）进行修改。
* 博客的发布：可以编写博客标题，博客摘要和博客正文，可以为博客添加标签、分类。
* 博客的查看：可以查看博客列表，以及某个博客的详细界面，可以进行添加、删除评论，添加、删除点赞等操作。

图 4.1 系统模块图

## 4.2 系统模块设计

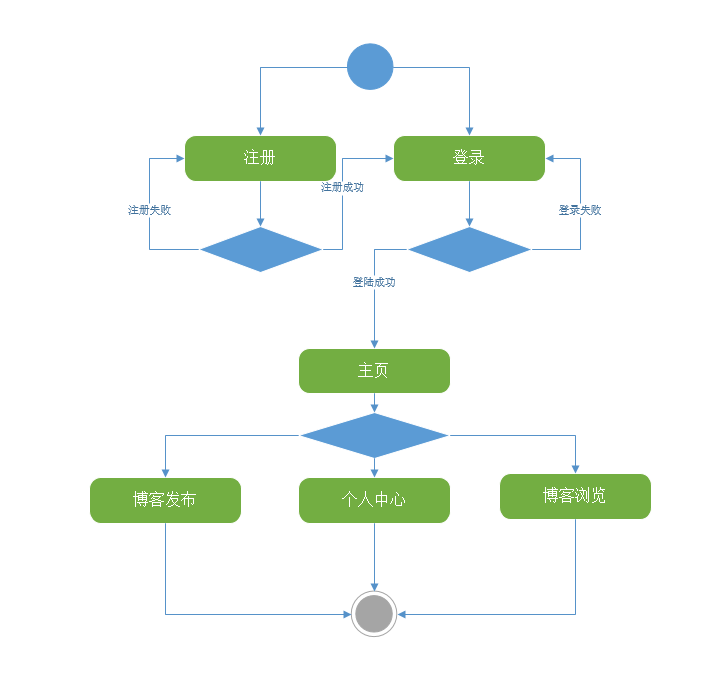
在4.1中对系统功能模块进行了大致的划分，接下来对四个大模块的功能进行设计。系统的整体流程是，用户注册登录，通过验证登陆成功后选择博客发布、博客查看、个人中心等操作，再在相应的模块下进行相应操作。系统整体的活动图如图4.2

图 4.2 系统活动图

### 4.2.1 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4.3。

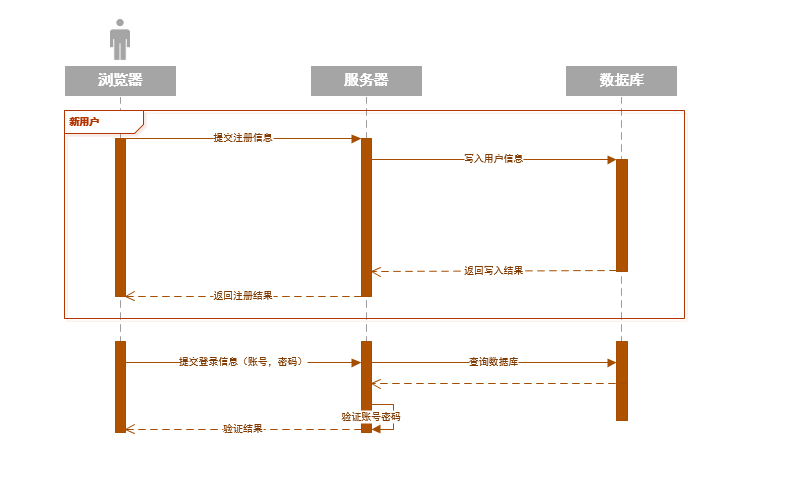


图 4.3 登录注册时序图

### 4.2.2 博客发布模块

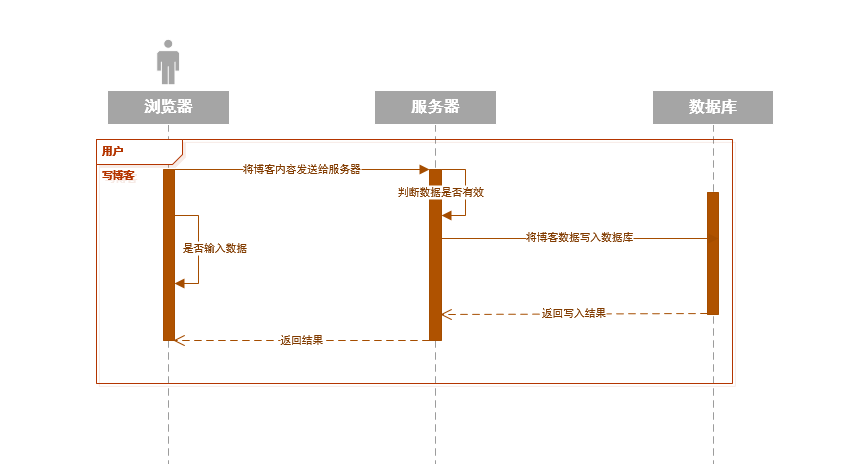
 用户登录之后，点击“写博客”按钮后跳转到写博客页面，编写博客标题、摘要、正文，然后添加标签，分类点击发布按钮，页面将数据传输给服务器，服务器审核、判断博客是否符合标准，若符合标准则将数据写入数据库中，返回成功信息跳转到博客详情页面，若不符合标准则返回结果和错误信息。

图 4.4 博客发布时序图

### 4.2.3 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。

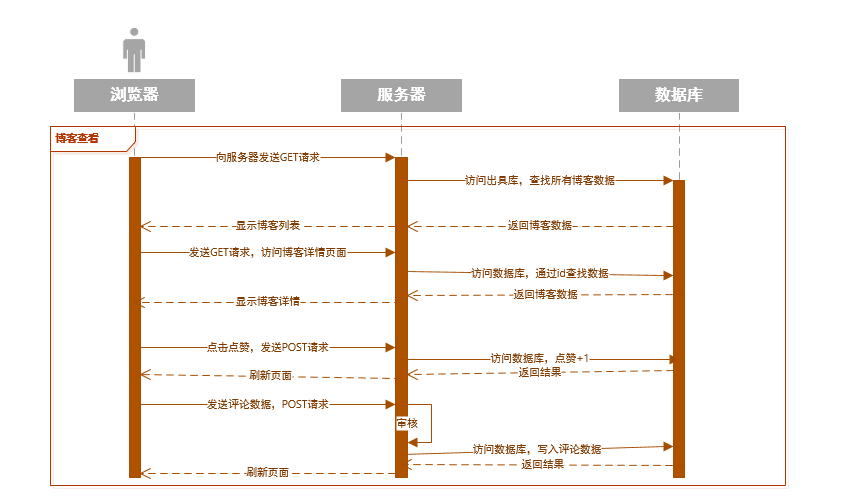
当用户点击点赞按钮时，向服务器发送POST请求，服务器访问数据库，将点赞数加一。当用户在评论框中填写数据并点击发表时，向服务器发送POST请求并将评论数据发送给服务器，服务器审核评论数据，审核通过则访问数据库将评论数据写入，若审核未通过返回错误信息。

图 4.5 博客查看时序图

## 4.3 数据库设计

数据库设计是指对于一个给定的应用环境，构造最优的数据库模式，建立数据库及其应用系统，使之能够有效地存储数据，满足各种系统的应用需求。数据库建模是开发中非常关键的一步。数据库的设计直接影响着系统的性能和开发的效率。

### 4.3.1 数据库模型

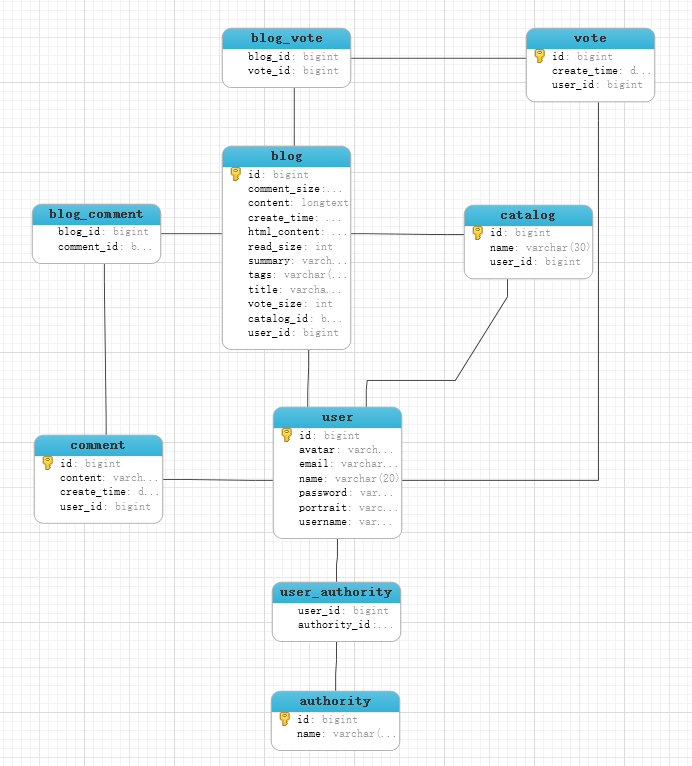
数据库的整体框架如图4.6

图 4.6 数据库E-R图

### 4.3.2 数据库表结构设计

4.3.1展示了数据库的总体设计，接下来对每个表的结构进行详细设计。

表 4.1 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 4.2 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 4.3 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 4.4 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 4.5 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 4.6 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 4.7 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 4.8 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 4.9 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 |

## 4.4 系统页面设计

系统页面是用户体验很重要的部分，好看、合理的页面布局，快捷明了的页面跳转可以很好地增加用户体验。该系统进入是显示主页，此时用户未登陆，可以查看博客，但无法进行评论点赞，无法访问个人中心。此时点击登录跳转至登录界面，登陆成功跳转到主页，若是点击注册则跳转至注册页面，注册成功后自动跳转至登录页面。在主页面点击博客进入博客详情页面，点击用户头像进入用户个人主页。在主页点击个人主页跳转至个人主页页面显示个人信息，点击个人设置则跳转个人设置页面。具体跳转情况如图4.7

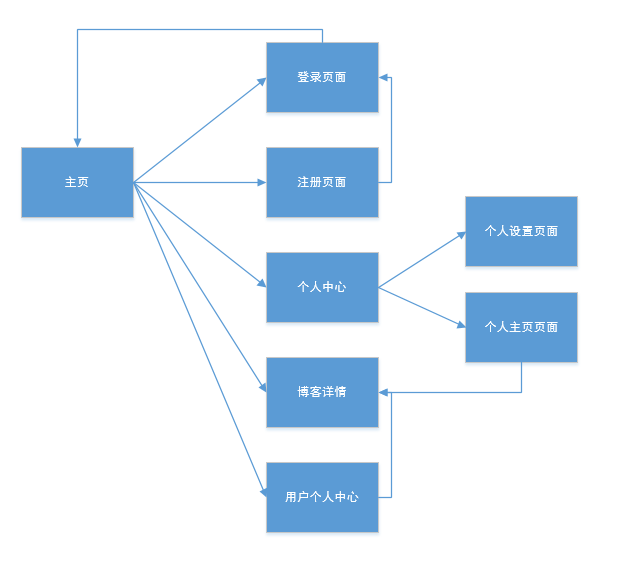


图 4.7 页面跳转图

此系统主页十分简洁大方，导航栏将系统所具有的功能简单直了的展现在用户面前，导航栏下方左侧为博客列表，右侧为热门标签、热门用户、热门文章以及最新发布的博客。博客列表显示了每个博客的标题、摘要、发布时间、点赞次数、阅读次数、评论条数，将每一篇博客简洁有效地展现了出来，如图4.8



图 4.8 系统首页

由于系统使用了Bootstrap进行了响应式布局，在移动设备上的显示效果也十分美观如图4.9



图 4.9 移动端系统首页

# 5 系统详细设计与实现

## 5.1 开发环境的搭建

### 5.1.1 Mean项目管理工具

Maven项目对象模型，可以通过一小段描述信息来管理项目的构建，报告和文档的软件项目管理工具。通过使用Maven，我们能够很方便地搭建开发环境，通过在pom.xml中添加依赖就可以完成整个项目的构建。

<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-thymeleaf</artifactId>  
</dependency>

添加Spring Boot 对 web支持

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

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。

### 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，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。

### 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。

*# DataSource  
spring.datasource.url*=jdbc:mysql:///blog  
*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。

### 5.1.4 Mean项目管理工具

JDK 1.8是JDK的最新稳定版本。

## 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4.3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。

### 5.2.1 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5.1

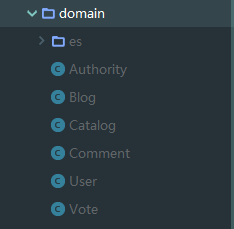


图 5.1 domain实体设计图

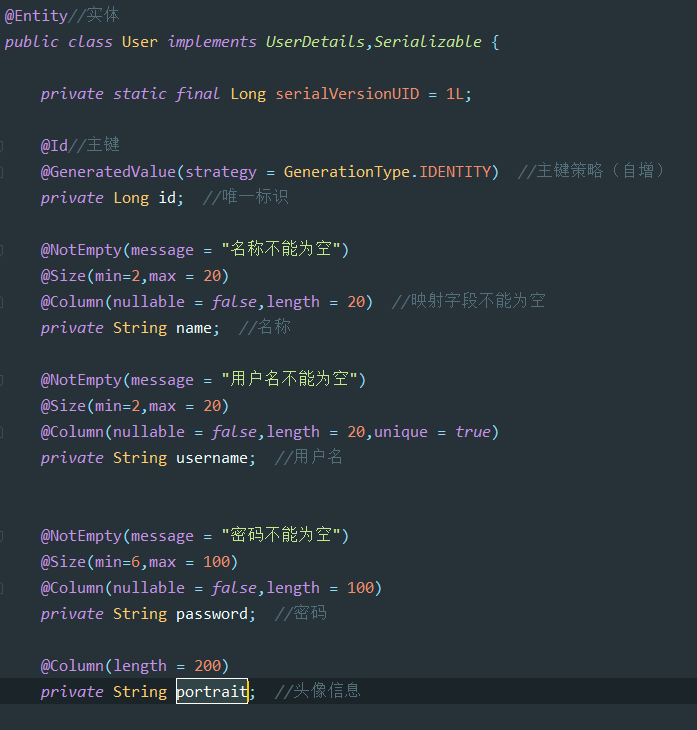
要在实体与数据库各表之间建立ORM的关系，需要用到一些注解来将类与数据库联系到一起，以用户类为例如图5.2

图 5.2 User实体设计图

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，@Colum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，只需要按章Spring Data的规定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，如图5.3

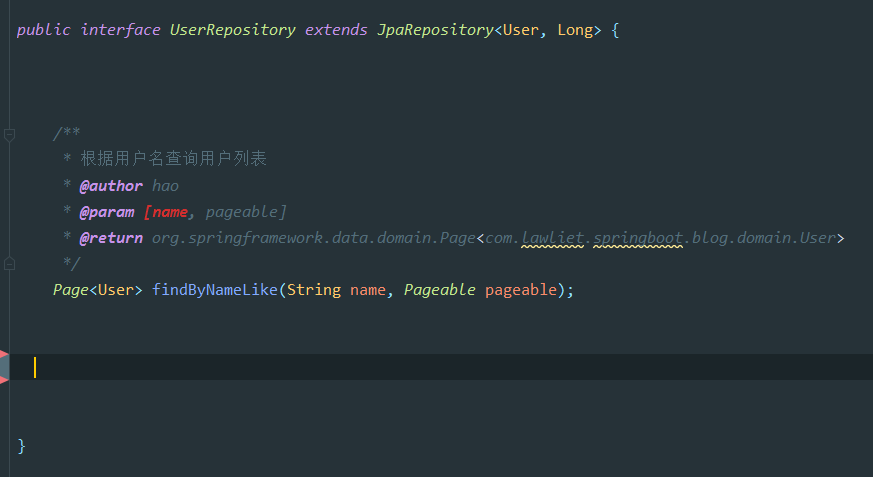


图 5.3 UserRepository图

只需要继承JpaRepository类既可以了，如方法findByNameLike（）便会查找所有名称中带有形参name的用户。

## 5.3 登录注册模块的详细设计

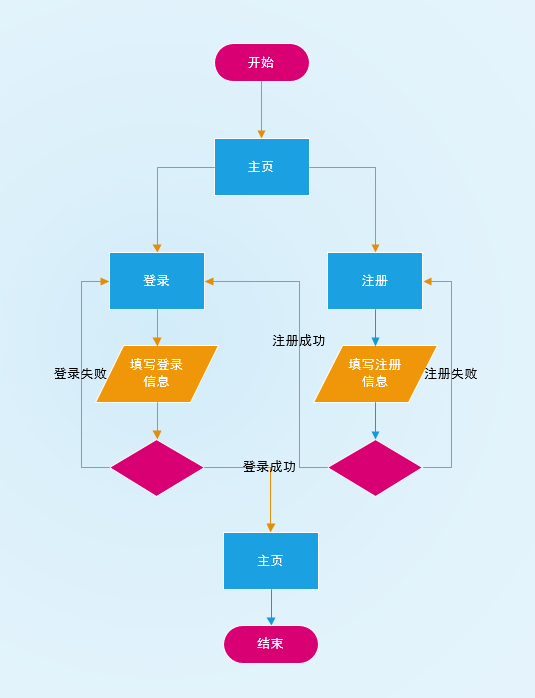
登录注册模块最重要的是其安全性，本系统作为一个自由的在线博客系统，即使用户没有登录，也可以阅读博客文章，只是无法进行点赞、评论等其他操作。具体的流程如图5.4。用户进入主页，选择登录或者注册。点击登录跳转至登录界面，点击登录，若登录成功进入主页。点击注册按钮，跳转至注册页面，填写注册信息，点击注册，注册成功则跳转至登录页面。

图 5.4 登录注册流程图

该系统的控制流程主要由Spring security控制，使用Spring security安全框架，提升了系统整体的安全性和可靠性。

图 5.5 重写configure方法

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。

通过在SecurityConfig类中重写configureGlobal方法来设置认证信息。如图5.6

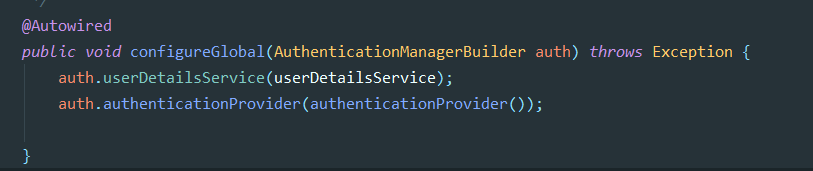


图 5.6 重写configureGLobal方法

## 5.4 博客发布模块的详细设计

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。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。

*// 对 Catalog 进行空处理  
if* (blog.getCatalog().getId() == *null*) {  
 User user = (User)userDetailsService.loadUserByUsername(username);  
 blog.setUser(user);  
 Catalog catalog = *new* Catalog(user,"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(*true*,"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。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。具体的代码如图5.7。将需要用到的UserDetailsService、userService、blogService、catalogService等使用set方法注入。



图 5.7 依赖注入

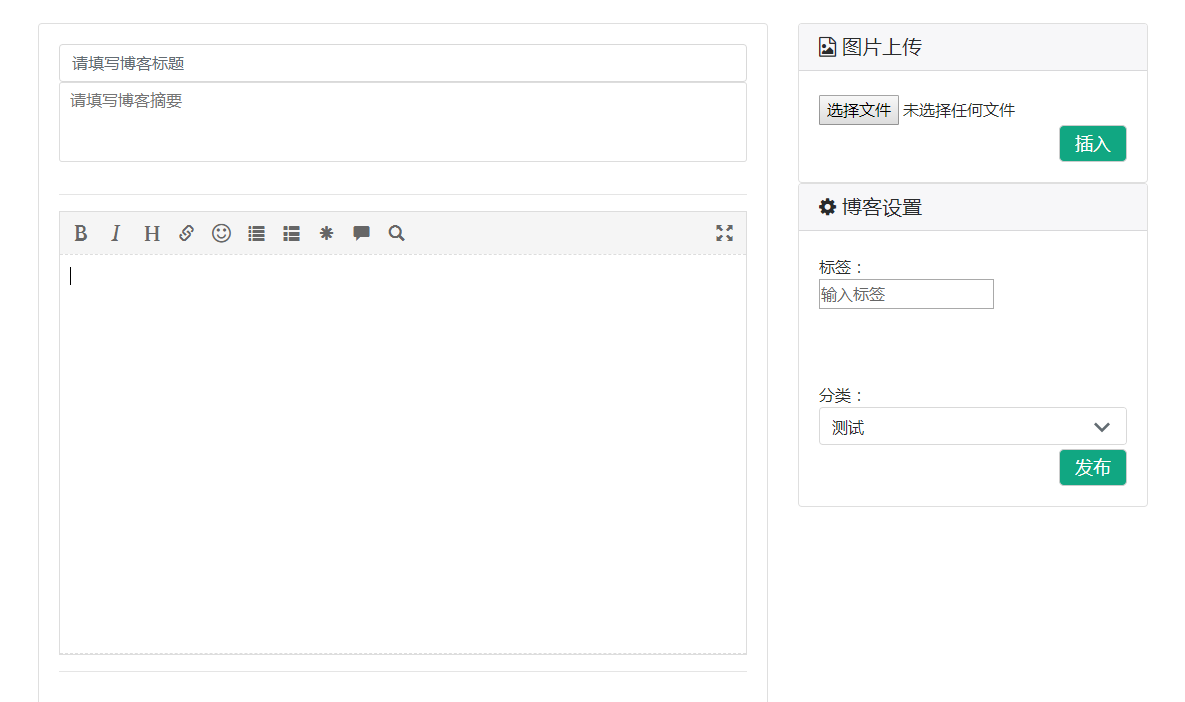
博客发布具体的页面如图5.8

图 5.8 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。

<div *class=*"col-lg-12">  
 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>  
  
</div>

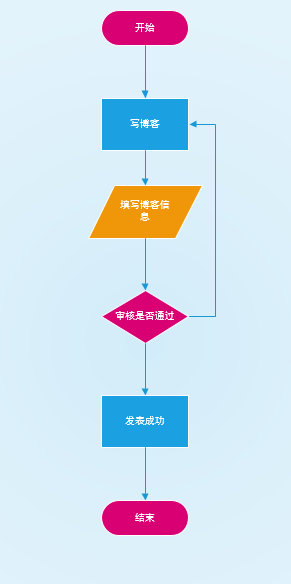
通过Thymeleaf的语法th:each="catalog : ${catalogs}"将model中的catalogs集合给遍历显示出来。博客发布总体的流程图如图5.9

图 5.9 博客发布流程图

## 5.5 博客查看模块的详细设计

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5.10

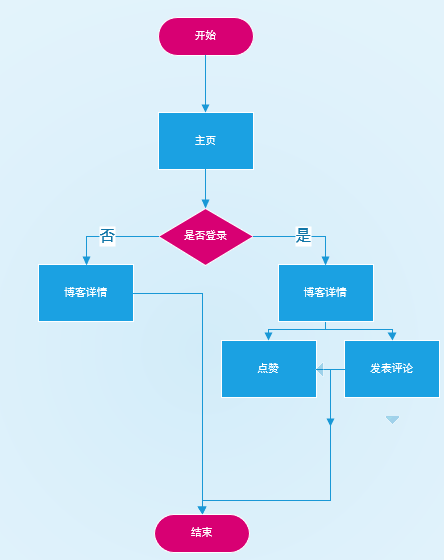
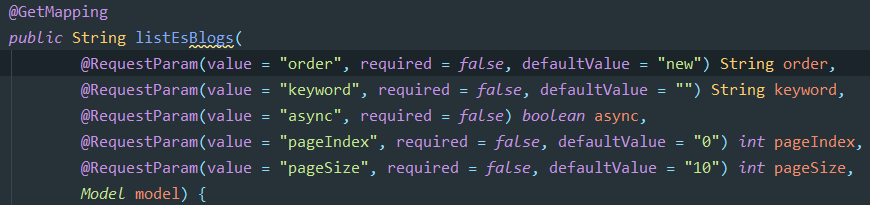


图 5.10 博客发布流程图

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，如图5.11.设置order实现最新最热排序，通过设置keyword实现关键词查询。

图 5.11 listEsBlogs方法

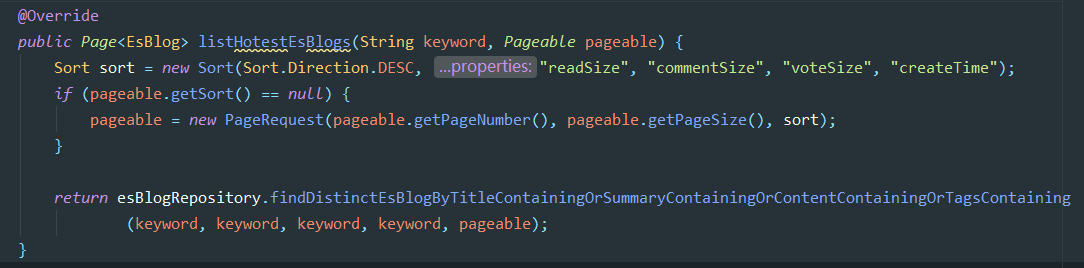
默认状态下order为new，也就是按照时间先后排序，当点击最热按钮时，order变为hot，将执行listHotestEsBlogs方法。

图 5.12 listEsBlogs方法

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().getAuthen

tication()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。

## 5.6 个人中心模块的详细设计

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5.13

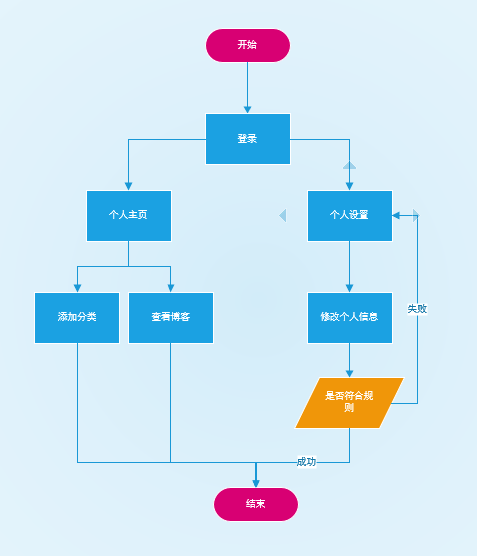


图 5.13 个人中心流程图

# 6 系统运行及测试结果

# **结 论**

。

# [**参考文献**](file:///C:\Users\wzlsyy\Desktop\毕业设计参考资料（To%20王梓懿）\AppData\Roaming\Microsoft\Word\论文格式摸板(2007).doc)

# 致 谢

该论文是在成都信息工程大学软件工程学院李代伟老师的悉心指导下完成的。李代伟老师是一位见多识广、拥有丰富经验的教师，具有大量的软件设计知识和丰富的项目开发经验，在整个系统的开发和论文撰写的过程中，李老师对我进行了悉心的指导，也为我提出了许多建设性意见，同时李老师时刻监督着我在整个毕业设计期间，有效地保证了我的毕业设计进度。每当我遇到问题时，李老师都会为我答疑解惑，勉励我大胆创新，在这几个月中，我在李老师的带领下学习到了许多解决问题的方法，也收获了许多宝贵的经验。在此，我向我的指导老师表示最真诚的感谢！

最后，我的毕业设计论文也算到了结尾，非常感谢我的指导老师对我的悉心教导，也感谢学院的苦心栽培，同时感谢学校几年来为我提供了这么好的学习环境和学习条件，希望毕业以后还能有机会为学校作出自己的贡献。

作者简介：

姓 名： 卢勇 性别： 男

出生年月： 1995.05 民族： 汉

E-mail: 411316917@qq.com

# 声 明

本论文的工作是 2016年12月至2017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：

2017年06月 日