 **JIANGXINORMALUNIVERSITY**

本科生毕业设计（论文）

**基于Spring Boot的博客系统的设计与实现**

**Design and Implementation of Blog System Based on Spring Boot**

**学院名称：计算机信息工程学院**

**专业名称： 网络工程**

**姓 名： 林臣哲**

**学 号： 201526203055**

**指导教师： 周新宇**

**完成日期：**

## 声明

本人郑重声明：

本人的毕业论文（设计）是在指导教师指导下独立进行研究取得的成果。其中引用的地方都加了标注和致谢，在法律规定允许的情况下，不包含任何其他个人或集体已经撰写完成或已经发表的以某种方式公开过的研究成果，不包含为了获得教育机构的证书或学位而作的材料。任何对本研究有贡献的个人和集体都在文中作了明确的说明并且表达了十分的谢意。本人完全意识到本声明的法律责任由本人承担。

本人的毕业论文（设计）成果在江西师范大学读书期间，在指导教师精心的指导下取得的，成果归江西师范大学计算机信息工程学院所有。  
 特此声明。

声明人（毕业设计（论文）作者）学号：  
声明人（毕业设计（论文）作者）签名：  
  
签名日期：　　　　年　　月　　日

**摘要**

近些年的网络急剧发展，带来了许多的便利与商机，人与人之间的沟通方式越来越多元化，其中博客是许多多获取知识信息的重要途径，也是承载现在信息爆炸时代大量信息的重要载体

本项目是一个基于Spring Boot框架的多用户博客系统，集成了Spring Security、Spring Session、Thymeleaf、BootStrap、HTML、CSS、MySQL、JavaScript，同时使用Gradle项目构建工具对项目进行管理，使用Git对项目进行版本管理，整体项目符合RESTful风格

**关键字：**Spring Boot ; MySQL; BootStrap;

1. **绪论**

**1.1开发背景**

**身处**信息时代，任何知识信息可以说信手拈来，但海量的信息看似又杂乱无章，个人如何进行有效的信息提取以及自身知识的管理，博客就发挥了巨大的作用。博客是一种由个人管理、张贴新的文章、图片或影片的网站或在线日记，用来纪录、抒发情感或分享信息。博客上的文章通常根据张贴时间，以倒序方式由新到旧排列。许多博客作者专注评论特定的课题或新闻，其他则作为个人日记。一个典型的博客结合了文字、图像、其他博客或网站的超链接、及其它与主题相关的媒体。能够让读者以互动的方式留下意见，是许多博客的重要要素。大部分的博客内容以文字为主，也有一些博客专注艺术、摄影、视频、音乐、播客等各种主题。博客是社会媒体网络的重要组成部分。

**1.2开发目的**

开发的目的是为了提供记录个人想法和知识的平台，有以下几方面：

1、博客可以作为一个见证，它既可以见证过去也可以见证未来看自己以前的文章，可以回顾自己的成长，就算是黑历史，也可以和现在的我打个照面，看看自己的改变，目前正在写的博文也将是未来我的“参考历史”，给自己留下一个悬念，期待一下现在的自己和未来的自己又会有怎样的不同。博客可以作为一个时间的媒介，它可以历尽沧海仍不变，供它的主人来了解其本身。

2、在学习到新知识时可以在博客上进行总结，并与他人进行交流。

**第2章 需求分析**

在系统工程和软件工程中，需求分析是指在创建新系统或更改现有系统或产品时确定新系统的目的，范围，定义和功能所需的所有工作。 需求分析是软件工程中的关键过程。 在此过程中，系统分析员和软件工程师确定客户的需求。 只有在确定了这些需求后，他们才能分析并寻求新系统的解决方案。

**2.1系统目标**

游客可浏览所有公开文章，如想自己发布文章或者与其他博主互动，需使用账号登录到系统中，对文章评论进行增删改查操作。在发布文章时同时指定文章所属分类，和相应的标签，在每篇文章下面提供了评论功能，可以与对文章进行评论。首页展示热门标签，分类，热门文章，最新文章等用户感兴趣的板块。在后台方面，系统提供了对分类，标签，文章，账号，权限进行统一管理。可对系统进行精细化管理。

**2.2系统需求**

**2.2.1软硬件需求**

硬件需求

PC机

(1)CPU：i7

(2)内存：8G

(3)固态硬盘：128G

(4)操作系统：Windows 10 Pro

软件需求：

(1)数据库：MySQL

开发工具：IntelliJ IDEA

浏览器：Chrome

Web容器 :Tomcat

其他工具：Git

**2.2.2技术需求**

对系统开发来说，技术可行性分析至关重要，做技术可行性分析时需注意全面考虑系统开发过程所涉及的所有技术问题，尽可能采用成熟技术，慎重引入先进技术。

1、Java语言

Java是一种通用的计算机编程语言，它是并发的，基于类的，面向对象的，并且专门设计为具有尽可能少的实现依赖性。它旨在让应用程序开发人员“编写一次，随处运行”，这意味着编译的Java代码可以在支持Java的所有平台上运行，而无需重新编译。Java应用程序通常编译为可以在任何Java虚拟机（JVM）上运行的字节码，而不管计算机体系结构如何。截至2016年，Java是最常用的编程语言之一，特别适用于客户端 - 服务器Web应用程序。该语言从SmallTalk中获得了大部分原始特性，其语法类似于C和C ++。

2、MySQL

MySQL是一个关系型数据库管理系统，MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的 RDBMS 应用软件。MySQL是一种关系数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL所使用的 SQL 语言是用于访问数据库的最常用标准化语言。

3、Spring Boot

Spring Boot可以轻松创建可以运行的独立的，生产级的基于Spring的应用程序。对Spring平台和第三方库进行了一种自以为是的观点，这样就可以轻松上手了。大多数Spring Boot应用程序只需要很少的Spring配置。 使用Spring Boot创建可以使用java -jar或更传统的war部署启动的Java应用程序。Spring Boot还提供了一个运行“spring脚本”的命令行工具。使Spring开发提供从根本上更快且可广泛访问的入门体验。 开箱即用，但随着需求开始偏离默认值而迅速摆脱困境。 提供大型项目（例如嵌入式服务器，安全性，度量标准，运行状况检查和外部化配置）通用的一系列非功能性功能。 绝对没有代码生成，也不需要XML配置。

4、Spring Security

Spring Security是一个安全框架，提供针对常见攻击的身份验证，授权和保护。通过对命令式和反应式应用程序的一流支持，它是保护基于Spring的应用程序的事实标准。

Spring Security是一个能够为基于Spring的企业应用系统提供声明式的安全访问控制解决方案的安全框架。它提供了一组可以在Spring应用上下文中配置的Bean，充分利用了Spring IoC，DI（控制反转Inversion of Control ,DI:Dependency Injection 依赖注入）和AOP（面向切面编程）功能，为应用系统提供声明式的安全访问控制功能，减少了为企业系统安全控制编写大量重复代码的工作。

5、Thymeleaf

Thymeleaf是一个现代服务器端Java模板引擎，适用于Web和独立环境，能够处理HTML，XML，JavaScript，CSS甚至纯文本。 Thymeleaf的主要目标是提供一种优雅且高度可维护的模板创建方式。为实现这一目标，它以自然模板的概念为基础，将其逻辑注入模板文件，其方式不会影响模板被用作设计原型。这改善了设计沟通，缩小了设计和开发团队之间的差距。 Thymeleaf也从一开始就设计了Web标准 - 特别是HTML5 - 允许创建完全验证的模板。

6、BootStrap

Bootstrap是一个免费的开源前端Web框架。它包含用于排版，表单，按钮，导航和其他界面组件的HTML和基于CSS的设计模板，以及可选的JavaScript扩展。与许多早期的Web框架不同，它只关注前端开发。 BootStrap是上最受欢迎的前端组件库在Web上构建响应式，移动优先项目。 Bootstrap是一个用于使用HTML，CSS和JS进行开发的开源工具包。使用我们的Sass变量和mixins，响应式网格系统，广泛的预构建组件以及基于jQuery构建的强大插件，快速构建您的想法或构建整个应用程序。

7、Spring Session

Spring Session提供了用于管理用户会话信息的API和实现，同时也使得支持群集会话变得微不足道，而不依赖于特定于应用程序容器的解决方案。它还提供透明集成： HttpSession - 允许以应用程序容器（即Tomcat）中立方式替换HttpSession，支持在头文件中提供会话ID以使用RESTful API。 WebSocket - 提供在接收WebSocket消息时保持HttpSession活动的能力 WebSession - 允许以应用程序容器中立方式替换Spring WebFlux的WebSession。

8、Hibernate

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， Hibernate负责从Java类到数据库表的映射，以及从Java数据类型到SQL数据类型的映射。此外，它还提供数据查询和检索功能。它可以显着缩短在SQL和JDBC中使用手动数据处理所花费的开发时间。 Hibernate的设计目标是通过消除使用SQL和JDBC手动，手工制作数据处理的需要，使开发人员从95％的常见数据持久性相关编程任务中解脱出来。但是，与许多其他持久性解决方案不同，Hibernate不会隐藏SQL的强大功能，并保证对关系技术和知识的投资与往常一样有效。

9、富文本编辑器CKEditor 5

CKEditor 5允许您快速，轻松地初始化它在您的应用程序中提供的众多类型的编辑器之一。同时，它是一个用于创建定制的富文本编辑解决方案的框架。CKEditor 5提供了可以想象的所有类型的WYSIWYG编辑解决方案。从类似于Google Docs和Medium的编辑器到Slack或类似Twitter的应用程序，所有这些都可以在一个编辑框架内完成。CKEditor 5有三种风格：Classic、Inline、Balloon。这里选用Classic风格更适合博客编辑

2.2.3功能需求

在软件工程和系统工程中，功能需求定义了系统或其组件的功能，其中功能被描述为输出和输入之间的行为规范。功能需求可能涉及计算，技术细节，数据操作和处理，以及定义系统应该完成什么的其他特定功能。在用例中捕获描述系统使用功能需求的所有情况的行为要求。功能要求由非功能性要求（也称为“质量要求”）支持，这些要求对设计或实施（例如性能要求，安全性或可靠性）施加约束。通常，功能要求以“系统必须<要求>”的形式表示，而非功能要求采用“系统应<要求>”的形式。系统设计中详细说明了实现功能要求的计划。而非功能性要求在系统架构中有详细说明。

用户前端：

首页：展示用户发布的最新文章，最热门文章，分类以及热门标签

文章详情：展示文章详情，并可以评论

用户信息：展示用户发布的所有文章评论，并提供修改邮箱密码和头像的功能

管理员前端：

分类管理：对分类进行管理，增删改操作

标签管理：对标签进行管理，增删改操作

文章管理：对文章进行管理，删除操作，和隐藏操作

权限管理：对用户权限进行管理，指定能否发布文章或者评论浏览之类的

**第3章 开发平台介绍与环境搭建**

**3.1 开发平台**

本博客系统前期基于Windows 10 64位平台开发，后期转向使用Mac平台开发，同时远程部署到Linux服务器进行运行调试

**3.2 Tomcat容器**

Apache Tomcat，通常称为Tomcat Server，是由Apache Software Foundation（ASF）开发的开源Java Servlet容器。 Tomcat实现了几个Java EE规范，包括Java Servlet，JavaServer Pages（JSP），Java EL和WebSocket，并提供了一个可以运行Java代码的“纯Java”HTTP Web服务器环境。

Tomcat包含以下几个组件：

Catalina：Catalina是Tomcat的servlet容器。 Catalina实现了Sun Microsystems的servlet和JavaServer Pages（JSP）规范。

Coyote：Coyote是Tomcat的Connector组件，它支持HTTP 1.1协议作为Web服务器。这使得Catalina（名义上是Java Servlet或JSP容器）也可以充当普通的Web服务器，将本地文件作为HTTP文档提供。

Jasper：Jasper是Tomcat的JSP引擎。 Jasper解析JSP文件，将它们编译为Java代码作为servlet（可以由Catalina处理）。在运行时，Jasper检测JSP文件的更改并重新编译它们。这也提供了在开发时进行热部署，修改代码无需重启Tomcat的功能基础

**3.3MVC架构**

Model–View–Controller（通常称为MVC或MVVC变体的模型WC）是一种通常用于开发用户界面的体系结构模式，用户界面将应用程序划分为三个互连的部分。这样做是为了将信息的内部表示与信息呈现给用户和从用户接受的方式分开。MVC设计模式将这些主要组件分离，从而实现高效的代码重用和并行开发。 传统上用于桌面图形用户界面（GUI），这种架构已经成为设计Web应用程序甚至移动，桌面和其他客户端的热门。流行的编程语言，如Java，C＃，Python，Ruby，PHP都有MVC框架，可直接用于Web应用程序开发。

Model： 模式的核心组成部分。它是应用程序的动态数据结构，独立于用户界面。[4]它直接管理应用程序的数据，逻辑和规则。

View： 任何信息表示，如图表，图表或表格。可以使用相同信息的多个视图，例如管理条形图和会计师的表格视图。

Controller： 接受输入并将其转换为模型或视图的命令。 除了将应用程序划分为这些组件之外，Model–View–Controller设计还定义了它们之间的交互。该模型负责管理应用程序的数据。它从控制器接收用户输入。 视图意味着以特定格式呈现模型。 控制器响应用户输入并对数据模型对象执行交互。控制器接收输入，可选地验证它，然后将输入传递给模型。 与其他软件模式一样，MVC表达了问题的“核心解决方案”，同时允许它适用于每个系统。



**3.6开发环境与工具**

**3.6.1 Tomcat容器**

Tomcat是由Apache软件基金会属下Jakarta项目开发的Servlet容器，按照Sun Microsystems提供的技术规范，实现了对Servlet和JavaServer Page（JSP）的支持，并提供了作为Web服务器的一些特有功能，如Tomcat管理和控制平台、安全局管理和Tomcat阀等。由于Tomcat本身也内含了HTTP服务器，因此也可以视作单独的Web服务器。但是，不能将Tomcat和Apache HTTP服务器混淆，Apache HTTP服务器是用C语言实现的HTTPWeb服务器；这两个HTTP web server不是捆绑在一起的。Apache Tomcat包含了配置管理工具，也可以通过编辑XML格式的配置文件来进行配置。

Tomcat 服务器是一个免费的开放源代码的Web 应用服务器，属于轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试JSP 程序的首选。对于一个初学者来说，可以这样认为，当在一台机器上配置好Apache 服务器，可利用它响应HTML（标准通用标记语言下的一个应用）页面的访问请求。实际上Tomcat是Apache 服务器的扩展，但运行时它是独立运行的，所以当你运行tomcat 时，它实际上作为一个与Apache 独立的进程单独运行的。

**3.6.2分布式版本控制系统**

Git是一个分布式版本控制系统，用于跟踪软件开发过程中源代码的变化。它旨在协调程序员之间的工作，但它可用于跟踪任何文件集中的更改。其目标包括速度，数据完整性，以及对分布式非线性工作流程的支持。与大多数其他分布式版本控制系统一样，与大多数客户端 - 服务器系统不同，每台计算机上的每个Git目录都是一个完整的存储库，具有完整的历史记录和完整的版本跟踪能力，独立于网络访问或中央服务器。

**第4章 系统实现详述**

**4.1系统设计**

**4.1.1 系统架构设计**

博客系统主要包括账号，文章，评论，标签，分类，权限模块

**4.1.2 数据库设计**

1、account（账号表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 描述 |
| account\_id | bigint | - | not null | 主键，账号id |
| account\_non\_expired | bit | - | not null | 账号是否过期 |
| account\_non\_locked | bit | - | not null | 账号是否被锁定 |
| avatar | varchar | 255 | not null | 头像地址 |
| create\_date | datetime | - | not null | 创建日期 |
| credentials\_non\_expired | bit | - | not null | 凭证是否过期 |
| email | varchar | 255 | not null | 电子邮箱 |
| enabled | bit | - | not null | 是否激活状态 |
| last\_modified\_date | datetime | - | not null | 上次修改时间 |
| password | varchar | 16 | not null | 密码 |
| username | varchar | 16 | not null | 用户名 |

2、account\_authority（账号权限表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 描述 |
| account\_id | bigint | - | not null | 主键，账号id |
| authority\_id | bigint | - | not null | 主键，权限id |

3、article（文章表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 描述 |
| article\_id | bigint | - | not null | 主键，文章id |
| content | mediumtext | - | not null | 富文本内容 |
| create\_date | datetime | - | not null | 创建时间 |
| is\_public | bit | - | not null | 是否公开 |
| last\_modified\_date | datetime | - | not null | 上次修改时间 |
| page\_view | bigint | - | not null | 浏览量 |
| raw\_content | mediumtext | - | not null | 纯文本内容 |
| title | varchar | 255 | not null | 标题 |
| account\_id | bigint | - | not null | 文章所属用户id |
| category\_id | bigint | - | not null | 所属分类id |

4、article\_tag（文章标签表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 描述 |
| tag\_id | bigint | - | not null | 主键，标签id |
| article\_id | bigint | - | not null | 主键，文章id |

5、authority（权限表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 描述 |
| authority\_id | bigint | - | not null | 主键，权限id |
| authority\_name | Varchar | 255 | not null | 权限名称 |
| create\_date | datetime | - | not null | 创建时间 |
| last\_modified\_date |  | - | not null | 上次修改时间 |

6、category（分类表）-----------------待定

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 描述 |
| category\_id | bigint | - | not null | 主键，分类id |
| category\_name | Varchar | 255 | not null | 分类名称 |

7、comment（评论表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 描述 |
| comment\_id | bigint | - | not null | 主键，评论id |
| content | Varchar | 255 | not null | 评论内容 |
| create\_date | datetime | - | not null | 创建时间 |
| last\_modified\_date | datetime | - | not null | 上一次修改时间 |
| account\_id | bigint | - | not null | 评论所属账号id |
| article\_id | bigint | - | not null | 评论所属文章id |

8、favorites\_article（用户文章收藏表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 描述 |
| account\_id | bigint | - | not null | 主键，账号id |
| article\_id | bigint | - | not null | 主键，文章id |

9、tag（标签表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 描述 |
| tag\_id | bigint | - | not null | 主键，标签id |
| tag\_name | Varchar | 255 | not null | 标签名称 |