

# 基于 SpringBoot+Mybatis 的个人博客系统设计与实现

刘子凡, 郭昱君

(四川大学锦城学院 计算机与软件学院, 四川 成都 611731)

**摘要:** 随着计算机技术的更新发展, 信息技术也产生了革命性的变化, 信息的传播方式开始向网络化转变, 博客网站在此形势下应运而生。然而, 现有的第三方博客网站存在各种问题, 例如网站的广告多、恶意弹窗等。鉴于此, 该文设计了一个小型的个人博客网站, 使用的开发语言为 Java, 采用后端框架 SpringBoot、持久层框架 Mybatis、前端框架 SemanticUI 以及模板引擎 Thymeleaf。该博客系统具有简洁、轻量化的特点, 使人们免去了各种广告、恶意弹窗等烦扰, 提供更加方便的博客平台。

**关键词:** 个人博客; SpringBoot; Mybatis; Thymeleaf

**中图分类号:** TP393.092

**文献标识码:** A

**文章编号:** 2096-4706 (2021) 08-0104-05

## Design and Implementation of Personal Blog System Based on SpringBoot+Mybatis

LIU Zifan, GUO Yujun

(School of Computer and Software, Jincheng College of Sichuan University, Chengdu 611731, China)

**Abstract:** With the update and development of computer technology, information technology also has a revolutionary change, the way of information dissemination came to the network, blog website came into being in this situation. However, the existing third-party blog websites have various problems, such as more advertisements, malicious pop ups and so on. In view of this, this paper designs a small personal blog website, using Java as the development language, using the back-end framework SpringBoot, persistence layer framework Mybatis, front-end framework SemanticUI and template engine Thymeleaf. The blog system has the characteristics of simplicity and lightweight, so that people can avoid all kinds of advertisements, malicious pop ups and other annoyances, and provide a more convenient blog platform.

**Keywords:** personal blog; SpringBoot; Mybatis; Thymeleaf

## 0 引言

随着互联网的快速发展, 信息传递的方式也逐渐网络化, 博客网站就是一种用来公布个人消息与实现沟通的专业网站, 提供基本的沟通和个性化设计<sup>[1]</sup>功能。然而, 过于追求商业化的博客网站却给用户带来了广告、弹窗的烦扰; 另外, 过于丰富的功能也使网站变得臃肿, 操作起来更加烦琐, 这无形中为那些追求简洁便利的用户带来莫大的困扰。

## 1 主要技术介绍

### 1.1 SpringBoot

SpringBoot 是由 Pivotal 团队设计开发的框架结构, 通过简化配置来优化 Spring 应用的搭建和开发过程。另外, SpringBoot 通过集成大量的框架使得依赖包的版本冲突, 以及应用的不稳定性问题得到很好的解决<sup>[2]</sup>。

### 1.2 Mybatis

Mybatis 与 Hibernate 是 Java 中最常用的两种 ORM 框架, 其中 Hibernate 在处理多表关联时可能会出现 N+1 问题, 性能会受到很大影响。若要解决性能受影响的问题, 需要使用

者具备深厚的 Hibernate 知识基础和丰富的项目经验<sup>[3]</sup>。所以本系统中采用了 Mybatis, 自己编写 SQL 语句, 以便具有一定的可控性, 并通过简单的 XML 或特定注解将 Java 中的普通对象映射为数据库里的记录。

### 1.3 Thymeleaf

Thymeleaf 是服务于 Java 的一个模板引擎, 能够处理 HTML、JS、XML 等, 并且不会影响 Java 原本的表现, 利用 Thymeleaf 可极大地提高开发速度, 同时还能完美地实现页面渲染<sup>[4]</sup>。

## 2 总体设计

本个人博客系统采用 B/S 架构进行设计, 分为游客端和管理员端两个子系统, 通过 SpringBoot 下的 MVC 分层结构来完成整个项目。游客端子系统主要是针对游客在博客网站的操作, 游客通过访问博客网站来浏览博客文章, 如果访客有特殊需求, 则可以通过分类、添加标签等操作进行文章筛选, 如果访客对自己所阅读的文章有所感悟或者想予评价, 那么可以在文章的底部给出留言和评价。管理员端子系统是针对管理员操作, 管理员通过登录自己的账号进入博客的管理系统, 在系统里面可以对文章进行增加、删除、修改、更新操作, 同时更新数据库里的数据。

## 2.1 前台功能设计

### 2.1.1 首页文章的分页功能

游客能够在网站首页中浏览各式各样的博客文章,然而由于文章数量庞大,因此需要有一个分页的功能,游客通过点击按钮的操作可以实现翻页。通过 Thymeleaf 技术,给标签 href 赋值来接收后台已经编写好的数据。实现代码为:

```
<div class="column">
  <a href="#" th:href="@{/pageNum=${pageInfo.hasPreviousPage}?${pageInfo.prePage}:1}" class="ui mini teal basic button"> 上一页 </a>
</div>
<div class="right aligned column">
  <a href="#" th:href="@{/pageNum=${pageInfo.hasNextPage}?${pageInfo.nextPage}:${pageInfo.pages}}" class="ui mini teal basic button"> 下一页 </a>
</div>;
```

### 2.1.2 文章分类查询功能

游客可以通过点击分类按钮,根据需要查找筛选自己感兴趣的文章。给 type-name 赋值为数据库里存放的文章类别名的数据, type-id 为文章类别的对应编号,通过文章类别名以及编号来确定该文章的类别,如果点击某个类别,主界面会显示该类别的相关文章信息。实现代码为: <a href="#" th:href="@{/types/{id}{id=\${type-id}}}" class="ui basic button" th:classappend="\${type-id==activeTypeId} ? 'teal'" th:text="\${type-name}">java

```
</a>
<div class="ui basic left pointing label" th:classappend="${type-id==activeTypeId} ? 'teal'"
  th:text="${#arrays.length(type-blogs)}">24
</div>;
```

## 2.2 后台功能设计

### 2.2.1 管理员账号登录功能

管理员输入自己的账号和密码登录后台管理系统,这一功能是通过以 Post 方式提交 from 表单来实现的。如果登录失败,会显示账户名称错误或密码错误。通过 if 条件语句判断管理员的账号 user 和密码 password 是否为空、是否对应数据库的信息等。实现代码为:

```
@PostMapping("/login")
public String login(@RequestParam String username, @RequestParam String password,
    HttpSession session, RedirectAttributes attributes){
    User user = userService.checkUser(username,password);
    if (user != null){
        user.setPassword(null);
        session.setAttribute("user",user);
        return "/admin/index";
    }else {
        attributes.addFlashAttribute("message","用户名或密码错误");
    }
}
```

```
return "redirect:/admin";
}
};
```

### 2.2.2 文章的管理功能

管理员登录后台后,可以对文章进行一系列的操作。对博客文章的新增操作,在前台提交数据后,后台会依次获取文章名字、文章的类型等,再通过 blogService.saveBlog() 方法去保存博客。实现代码为:

```
blog.setUser((User) session.getAttribute("user"));
blog.setType(typeService.getType(blog.getType().getId()));
blog.setTypeId(blog.getType().getId());
blog.setUserId(blog.getUser().getId());
int b = blogService.saveBlog(blog);
```

文章删除功能的实现是通过调用 blogService.deleteBlog() 方法,查找文章的编号来删除文章。实现代码为:

```
blogService.deleteBlog(id);
attributes.addFlashAttribute("message","删除成功");
管理员上传了文章后想对文章进行更新、修改,以下代码则为文章的修改功能,调用 blogService.updateBlog() 方法,重新传输修改的数据。
```

```
int b = blogService.updateBlog(showBlog);
if(b == 0){
    attributes.addFlashAttribute("message","修改失败");
}else {
    attributes.addFlashAttribute("message","修改成功");
};
```

### 2.2.3 文章的分类功能

管理员登录后台后,可以对文章内容进行分类操作。例如添加文章类型功能的实现是通过 type1 获取类型名字,添加类型的同时增加数据库里的信息,并判断所添加的类型是否有重复。实现代码为:

```
Type type1 = typeService.getTypeByName(type.getName());
if (type1 != null) {
    attributes.addFlashAttribute("message","不能添加重复的分类");
    return "redirect:/admin/types/input";
}
int t = typeService.updateType(type);
if (t == 0) {
    attributes.addFlashAttribute("message","编辑失败");
} else {
    attributes.addFlashAttribute("message","编辑成功");
};
```

## 2.3 数据库设计

为保证网站中的数据可以正确显示,需要将不同的数据类型进行分类。对于不同字段值的设计要够精准,以免在填写信息的时候出现数据添加不正确的情况<sup>[5]</sup>。这时数据库的设计就显得非常重要。本系统设计了多张表格,各个表之间相互关联,数据库按照三范式进行设计。各个表之间相互关

联, 每张表都存储着用户的数据等, 以下为表的具体说明:

(1) 管理员表。管理员表主要设计的内容包括管理员账号、管理员密码、账号创建时间等字段, 其中 id 为主键, 管理员表如表 1 所示。

表 1 管理员表 (t\_user)

编号 (主键)	id	varchar (10)
管理员账号	username	varchar (25)
管理员密码	password	varchar (25)
账号创建时间	create_time	datetime

(2) 文章信息表。文章信息表主要设计的内容包括上传文章的相关信息, 例如文章标题、文章上传时间、文章简介、文章标签等字段, 其中字段名文章类型为外键, 文章信息表如表 2 所示。

表 2 文章信息表 (t\_blog)

文章编号 (主键)	id	int (32)
文章标题	title	varchar (50)
文章上传时间	create_time	datetime
文章简介	description	varchar (32)
文章标签	flag	varchar (16)
文章更新时间	update_time	datetime
文章类型 (外键)	type_id	bigint (20)

(3) 文章类型表。文章类型表主要设计的内容包括相关文章的类型名、相关文章所对应的类型编号等字段, 文章类型表如表 3 所示。

表 3 文章类型表 (t\_type)

文章类型编号	id	bigint (20)
文章类型名	name	varchar (25)

### 3 系统的具体功能实现

#### 3.1 博客系统的登录注销功能实现

管理员进行登录操作时, 是通过以 post 方式提交 from 表单来实现的, 提交的数据在 LoginController 类进行判断。首先针对用户提交的数据通过 if 语句进行合法性判断, 判断其中是否含有非法字符。其次把传入的用户名当作参数调用 UserService 类的 checkUser() 方法, 如果返回的值为 null, 就给前端页面返回“账号或密码错误!”; 如果用户提交的数据没有问题, 系统就将用户的数据通过 Dao 层的类存储到数据库中。关于账号的注销功能, 通过使用 session.removeAttribute("user"), 对 session 进行移除就能实现将账号退出登录。当管理员成功登录至后台后, 如果想要退出登录, 只需点击“注销”按钮即可退出账号并跳转至登录页面。

#### 3.2 文章的增删查改功能实现

对于本后台管理系统, 虽然可以在数据库中进行增删查改操作, 但是不够方便快捷, 所以如何能够快速简便地对文章进行增删查改操作, 是本系统的关键所在。本系统将利用 Mybatis 框架对映射文件进行编写, 在根元素 <mapper> 下创建各种子元素。对于增加、删除、查找、修改文章

功能, 分别使用 <insert>、<delete>、<select>、<update> 子元素, 在其子元素里编写插入的 SQL 语句, 并通过相应的 Controller 调用 Service 层, Service 层调用 Dao 层, 在 Dao 层里找到同名的 Mapper 文件处理数据, 返回前端。在增加文章的操作中, 前端页面使用了 Markdown 插件和 typo-css 插件, 能够有效地帮助我们快速编辑、排版, 前端页面清新简洁, 能够有效地提高工作效率。同时, 在页面上进行文章的增删查改操作, 其数据也会记录在数据库里。

#### 3.3 文章分类功能的实现

文章分类的功能和文章的管理功能类似, 都有增加、删除、查找、修改等功能, 然而两种功能的不同之处在于, 当管理员进行文章的编辑操作时, 前端页面会有单独的“添加分类”按钮用于文章分类。点击分类按钮时页面会有弹窗提示分类的名称, 且名称是通过数据库本来的数据显示在页面上, 在文章上传成功后, 页面会有弹窗显示“上传成功!”, 游客就可以在该分类栏里找到该文章。在前端通过 Thymeleaf 的标签 <a href="#" th:href="@{/admin/types}" 来接受后台传上来的分类数据。

### 4 测试及结果

项目完成之后, 对本个人博客系统进行了功能性测试。

从网址 http://localhost:8080/login 进入网站, 显示登录界面, 在界面中输入账号密码, 点击登录按钮, 登录成功并跳转至后台管理页面, 如图 1、图 2 所示。



图 1 登录页面



图 2 后台管理页面

在搜索框中输入文章标题或分类, 点击“搜索”按钮, 主界面会显示文章内容相关信息, 再点击“编辑”按钮, 跳转至相关文章内容, 如图 3 所示。

在页面中点击“新增”按钮, 跳转至文章编辑页面, 如图 4 所示, 进行一系列编辑后, 再点击“发布”按钮, 页面会显示“发布成功”。





图 3 文章查询功能

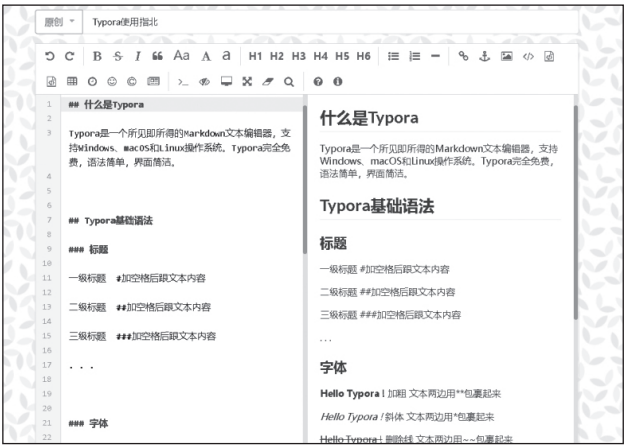


图 4 文章编辑页面

在页面中点击“删除”按钮，页面弹出弹窗“确定要删除吗？”，点击按钮“是”，页面自动刷新，将不再显示该文章的信息，如图 5 所示。



图 5 文章的删除

在页面中点击“分类”按钮，跳转至文章分类编辑页面，并通过文本框实现对文章类别的添加操作，再点击“新增”按钮，在输入框中输入新增的类别，并点击“提交”按钮，页面会显示“新增成功”，如图 6、图 7 所示。

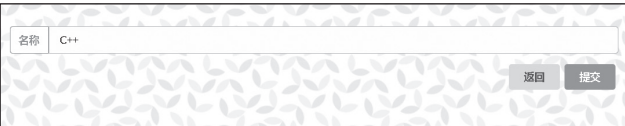


图 6 文章类型的新增

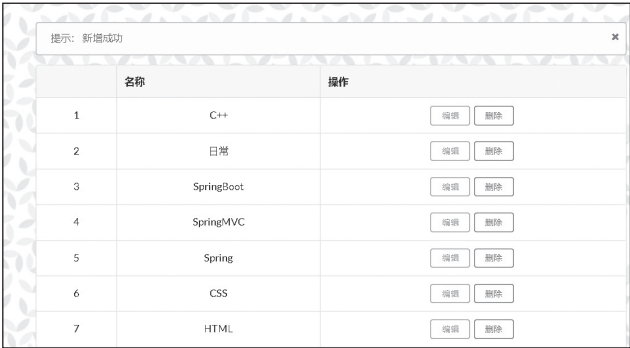


图 7 文章类型新增成功页面

游客访问前台时，网页上会显示所有文章的信息，并显示“首页”“分类”“关于我”等按钮，点击文章信息，跳转至相关文章内容页面，如图 8 所示。



图 8 首页页面

5 结 论

综上所述，本文叙述了基于 SpringBoot+Mybatis 技术的个人博客系统，该系统具有对博客文章的上传、增加、修改、删除等功能。SpringBoot 框架的“开箱即用”、无须配置等特点为整个项目的快速开发节省了大量时间，通过一些美化界面的插件以及前端框架 Semantic UI 对整个页面进行进一步的美化，从而达到个人博客系统的简洁、美观。

虽然本系统能够正常运行操作，但是站在更高的角度来看还可以做出相应的完善，由于 SpringBoot 框架的集成度很高，因此开发人员对一些底层代码是不熟悉的，其中比较复杂的函数方法还需要大量、全面地搜集资料，（下转 111 页）

原因进行总结展示,从视觉、听觉体验上进行感性认识后,对灾害发生的前因后果有更理性的认识,提升灾害体验课程的培训深度,提高学习效率。矿井顶板灾害原因分析流程如图8所示。

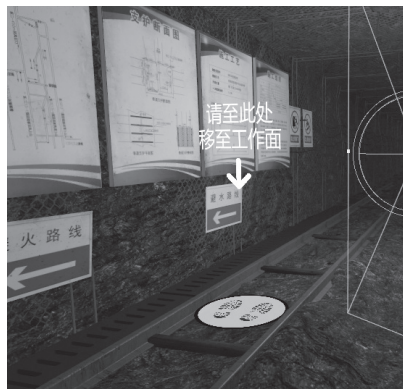


图7 灾害场景引入界面



图8 顶板灾害原因分析

#### 4.4 安全生产规范流程

通过对灾害原因分析学习后,设置与灾害发生原因紧密相关的设备操作、工艺流程操作等操作学习课程,使培训学员掌握正确的操作流程或处理措施,避免真实生产中同类灾害事故的发生,提升学员技能水平。具体顶板灾害处理规范流程如图9所示。

#### 4.5 安全知识问答

通过还原的灾害场景震撼性体验,学习过灾害发生原因分析和规范流程操作后,本系统提供了与灾害相关题库,包含灾害基本知识、应急救援知识等灾害相关的内容,供体验后的学员检验培训效果使用。具体如图10所示。

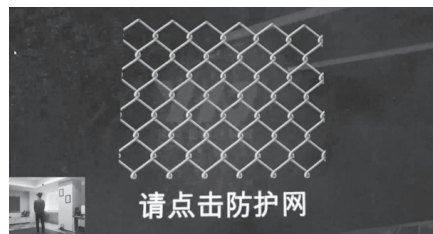


图9 顶板灾害处理规范流程

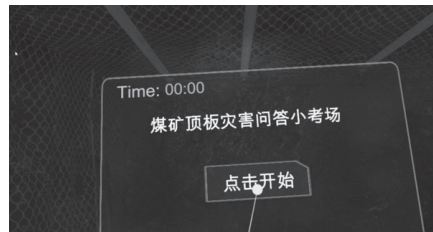


图10 安全知识问答

## 5 结 论

综上所述,虚拟现实技术在矿井灾害防治演练培训中的应用,可以有效地增强教学内容直观性和实践性,帮助学员更好地掌握比较复杂和抽象的专业知识,通过建模、动画制作、交互功能开发等一系列开发流程,打造了具有沉浸性、交互性、想象性的体验式培训课程,实现了提高学员学习积极性、提升煤矿灾害类课程培训效果,提高了下井人员的安全技能,为矿山企业的安全生产提供保障。

#### 参考文献:

- [1] 刘广超,李宝富,韦四江.辅助设计和可视化技术在采矿工程专业教学中的应用[J].煤炭技术,2009,28(10):145-146.
  - [2] 唐超礼,黄友锐,李继云,等.虚拟煤矿安全培训系统的设计与实现[J].煤矿安全,2011,42(2):87-88.
  - [3] 刘洋.基于VR技术的煤巷综掘仿真系统的研究与开发[D].西安:西安科技大学,2008.
  - [4] 戴正烈.基于虚拟现实技术的矿山应急救援培训研究[D].三河:华北科技学院,2015.
  - [5] 卞锋,江漫清,桑永英.虚拟现实及其应用进展[J].计算机仿真,2007(6):1-4+12.
- 作者简介:孟龙(1990—),男,汉族,安徽淮北人,工程师,硕士研究生,研究方向:数字化矿山设计。

(上接107页)还有其版本更新迭代快,更新后整个模块还需要重新学习。对于本系统而言,需要完善的地方是尚未实现将系统部署在服务器上,游客的访问只能从本地IP登录,具体功能方面就是游客对博客进行评论、打赏的功能并未实现,在以后的学习中,还需要积累知识,融会贯通,不断地改进和完善项目。

#### 参考文献:

- [1] 余思源,张伟.基于JAVA的个人博客系统的设计与实现[J].电脑知识与技术,2018,14(17):135-137.
- [2] 邓笑.基于Spring Boot的校园轻博客系统的设计与实现

[D].武汉:华中科技大学,2018.

- [3] 陈玲,夏汛.利用Mybatis的动态SQL实现物理分页[J].数字技术与应用,2011(11):227.
- [4] 唐炜.Spring Data、MongoDB、Thymeleaf的数据持久化方案及分页技术实现[J].陇东学院学报,2017,28(5):9-13.
- [5] 陈秋玲.基于JavaWeb的图书购物网站的设计与实现[J].电脑知识与技术,2019,15(16):36-38.

作者简介:刘子凡(1999—),男,汉族,四川绵阳人,本科在读,研究方向:软件工程;郭昱君(1994—),女,汉族,山西太原人,硕士研究生,研究方向:管理学。