

课程设计说明书

设计题目： 个人博客系统设计与实现

学　　院 **计算机科学与信息工程学院**

专业年级 **2016级 计算科学与技术**

学生姓名 程欣  学号  **2016131232**

指导教师 **唐灿** 职称 **副教授**

设计地点 **重庆工商大学**

日 期 **2019年6月1日\_— 6月30日**

个人博客系统设计与实现

重庆工商大学 计算机科学与技术 2016级

指导老师：唐灿

**中文摘要：**

对个人而言，博客可以调动个人的积极性，充分发挥个人的创造性。近些年，国内博客开始迅猛发展起来，博客系统致力于为广大用户提供优质的互动交流平台。

本文首先首先论述了系统的开发背景以及可行性，确定在当前条件下，开发个人博客系统是可行的，并对系统开发中的技术进行了简单的介绍，本系统基于springboot框架和mysql数据库，并将两者有机结合在一起。该子系统主要采用bootstrap前端设计框架，并结合jQuery及其相关插件等进行设计开发，后台则基于springboot框架进行相应功能的编写。

**关键字：**Spring Boot、mysql、博客

**The design and implementation of the Blog system**

Chongqing Technology and Business University

Computer Science and Technology

The class of 2016

Instructor : Can Tang

**Abstract:**

For individuals, blogs can mobilize their enthusiasm and give full play to their creativity. In recent years, domestic blogs have begun to develop rapidly, and blog system is committed to providing users with high-quality interactive communication platform.

Firstly, this paper discusses the development background and feasibility of the system, confirms that it is feasible to develop a personal blog system under the current conditions, and briefly introduces the technology in the system development. The system is based on spring boot framework and MySQL database, and organically combines the two. This subsystem mainly uses bootstrap front-end design framework, and combines jQuery and related plug-ins to design and develop, while the background is based on spring boot framework to write the corresponding functions.

**Key Words:** Spring Boot、MySql、Blog

**目 录**

[一、 前言 5](#_Toc518132098)

[1、 开发背景 5](#_Toc518132099)

[2、 系统可行性分析 5](#_Toc518132100)

[二、 需求分析 5](#_Toc518132101)

[1、 概述 5](#_Toc518132102)

[2、 功能需求 5](#_Toc518132103)

[3、 UML用例图 5](#_Toc518132104)

[三、 概要设计 6](#_Toc518132105)

[1、 系统概述 6](#_Toc518132106)

[2、 软件体系结构 6](#_Toc518132107)

[3、 系统功能模块 6](#_Toc518132108)

[4、 个人博客系统设计与实现—子系统模块设计 6](#_Toc518132109)

[5、 数据库设计 7](#_Toc518132110)

[四、 详细设计及编码 7](#_Toc518132111)

[五、 软件测试 10](#_Toc518132112)

[六、 总结 12](#_Toc518132113)

[参考文献 12](#_Toc518132114)

1. **前言**
2. 开发背景

博客是近些年比较流行的一种传播媒介,它是开放的私人空间的窗口，人们可以通过博客系统记录每天的一些想法和感受，在博客上展示出来，在当前的社会压力下，人们尤为需要这样一种方式来释放压力，将自己的愉快和忧愁分享出来，使自己的思想能够的到解放。

1. 系统可行性分析

简单分析博客系统需要实现的功能，可分为前端显示界面和后台管理界面，前端显示界面有多种方式实现，而我们要能够实现动态的显示博客内容，故而简单的html显然不能实现，jsp虽然能很好的支持页面的动态显示，但是由于没有学习过，所以采用jsp实现前端界面并不现实，但是我们可以采用html结合ajax使用Jquery框架和bootstrap的方式来实现动态显示前端界面，而后台方面则采用Spring Boot作为核心框架，使用Thymeleaf模版引擎来实现后台系统。

1. **需求分析**
2. 概述

完成博客系统前端动态的显示博客内容，并能够并能够通过点击标题进入详细信息页，能够实现游客浏览以及用户评论。后台方面则实现对博客内容、评论内容等进行基本管理。

1. 功能需求

前端展示： 动态显示博客内容，按分类查询博客内容，按标签查询博客内容，博客内容详细展示，博客内容的评论。

后台管理：博客管理，评论管理。

1. UML用例图



图1 系统总体用例图

1. **概要设计**
2. 系统概述

系统主要实现博客主人对博客页面的管理以及用户和游客对博客进行浏览。

管理方面，需要实现能够对已发博客进行修改并记录修改时间，能够发布新的博客并自动获取当前时间作为创建时间，能够对已发博客进行删除，删除是出现提示，是否确认删除。

浏览方面，提供一个用户界面良好，交互体验流畅的web页面，实现浏览博客内容的流畅性。

1. 软件体系结构

本系统采用时下十分流行的B/S架构，免除了C/S架构安装客户端的烦恼，保留了博客系统的简洁大方感。

Web服务器端：采用java作为主要的开发语言来实现web服务器上的所有的业务逻辑功能，对应的服务器软件选用tomcat来作为应用服务器软件。

数据库服务器：采用mysql来作为数据库存储数据。

1. 系统功能模块



1. 个人博客设计与实现—子系统模块设计



1. 数据库设计

表3.5.1 tb\_blog表的结构



表3.5.2 tb\_blog\_category表的结构



表3.5.3 tb\_blog\_comment表的结构

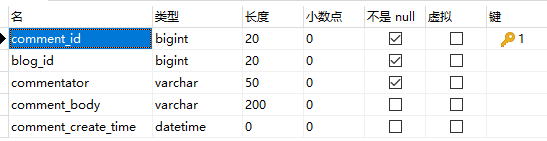


表3.5.4 tb\_tag表的结构



1. **详细设计及编码**

后台博客管理部分代码

**dao层：**

package cn.edu.ctbu.sbadmin.blog.dao;

import cn.edu.ctbu.sbadmin.blog.domain.BlogDO;

import cn.edu.ctbu.sbadmin.common.core.MyMapper;

import cn.edu.ctbu.sbadmin.common.utils.MQuery;

import java.util.List;

public interface BlogDAO extends MyMapper<BlogDO> {

/\*\*

\* 查询视图

\* @param mQuery

\* @return

\*/

public List<BlogDO> listView(MQuery mQuery);

/\*\*

\* 查询计数

\* @param mQuery

\* @return

\*/

public int countView(MQuery mQuery);

}

**controller层：**

package cn.edu.ctbu.sbadmin.blog.conrtoller;

import cn.edu.ctbu.sbadmin.common.controller.BaseController;

import org.apache.shiro.authz.annotation.RequiresPermissions;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

@Controller

public class BlogController extends BaseController {

@RequestMapping("/blog/backstage/list")

@RequiresPermissions("blog:backstage:list")

public String list(){

return "blog/backstage/list";

}

@RequestMapping("/blog/backstage/edit")

@RequiresPermissions("blog:backstage:edit")

public String edit(){

return "blog/backstage/edit";

}

}

**service层：**

package cn.edu.ctbu.sbadmin.blog.service;

import cn.edu.ctbu.sbadmin.blog.domain.BlogDO;

import cn.edu.ctbu.sbadmin.common.utils.MQuery;

import java.util.List;

public interface BlogService {

List<BlogDO> listView(MQuery mQuery);

int countView(MQuery mQuery);

}

**service.impl层：**

package cn.edu.ctbu.sbadmin.blog.service.impl;

import cn.edu.ctbu.sbadmin.blog.dao.BlogDAO;

import cn.edu.ctbu.sbadmin.blog.domain.BlogDO;

import cn.edu.ctbu.sbadmin.blog.service.BlogService;

import cn.edu.ctbu.sbadmin.common.core.AbstractService;

import cn.edu.ctbu.sbadmin.common.utils.MQuery;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import java.util.List;

@Service

@Transactional

public class BlogServiceImpl extends AbstractService<BlogDO> implements BlogService {

@Autowired

BlogDAO blogDAO;

/\*\*

\* 通用查询，返回视图

\*

\* @param mQuery

\* @return

\*/

public List<BlogDO> listView(MQuery mQuery) {

return blogDAO.listView(mQuery);

}

/\*\*

\* 通用查询，返回数量

\*

\* @param mQuery

\* @return

\*/

public int countView(MQuery mQuery) {

return blogDAO.countView(mQuery);

}

}

**service.impl测试**

package cn.edu.ctbu.sbadmin.blog.service.impl;

import cn.edu.ctbu.sbadmin.blog.domain.BlogDO;

import cn.edu.ctbu.sbadmin.common.utils.MQuery;

import cn.edu.ctbu.sbadmin.common.utils.MQueryHelper;

import org.junit.Test;

import org.junit.runner.RunWith;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.test.context.junit4.SpringRunner;

import java.util.List;

import static org.junit.Assert.\*;

@RunWith(SpringRunner.class)

@SpringBootTest

public class BlogServiceImplTest {

@Autowired

BlogServiceImpl blogServiceImpl;

@Test

public void listView()throws Exception {

MQuery mQuery = MQueryHelper.GenQuery("",10,0,"","");

List<BlogDO> blogDOList=blogServiceImpl.listView(mQuery);

System.out.println(blogDOList);

}

}

1. **软件测试**

1、对登录进行测试

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 项目/软件 | 个人博客系统 | | 编制时间 | 2019.6.20 | | |
| 功能模块名 | 登录 | | 用例编号 | 01 | | |
| 功能特性 | 登录 | | | | | |
| 测试目的 | 检验登录模块的准确性 | | | | | |
| 测试数据 | 1、输入错误账号zhangsan 密码123456登录  2、输入正确账号admin与正确密码123456进行登录  3、输入正确账号admin与错误密码123进行登录 | | | | | |
| 操作步骤 | 操作描述 | 数 据 | | 期望结果 | 实际结果 | 测试状态 |
| 1 | 对错误账号进行登录 | 账号：zhangsan  密码：123456 | | 提示非法参数 | 提示非法参数 | 正确 |
| 2 | 对正确账号密码进行登录 | 账号：admin  密码：111111 | | 正确登录 | 正确登录 | 正确 |
| 3 | 对正确账号不正确密码进行登录 | 账号：admin  密码：123 | | 提示非法参数 | 提示非法参数 | 正确 |

2、博客发布模块测试

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 项目/软件 | 个人博客系统 | | 编制时间 | 2019.6.20 | | |
| 功能模块名 | 发布博客 | | 用例编号 | 02 | | |
| 功能特性 | 发布博客 | | | | | |
| 测试目的 | 检验发布博客的功能能否正常运行 | | | | | |
| 测试数据 | 1. 点击新增按钮，弹出发布博客界面，什么都不填写直接点击发布 2. 点击新增按钮，弹出发布博客界面，输入标题test，输入作者devil，选择分类java，选择标签python，输入内容test，点击发布。 | | | | | |
| 操作步骤 | 操作描述 | 数 据 | | 期望结果 | 实际结果 | 测试状态 |
| 1 | 发布空博客 | 无 | | 提示相关字段为必填字段 | 提示相关字段为必填字段 | 正确 |
| 2 | 正确发布博客 | 标题：test  作者：devil  分类：java  标签：python  内容：test | | 发布成功 | 发布成功 | 正确 |

1. **总结**

经过此次个人博客系统的设计与实现，我对于javaweb项目的认识又深刻了不少，虽然本次系统编写并没有采用jsp和servlet，但是在前期找资料的时候也是了解到可能采用jsp和servlet来进行编写的话会相对比较容易实现一些，但奈何自身技术不过硬，所以只能根据老师给出的框架，结合自己的理解进行设计实现，尽管如此本系统也做的不尽如人意，各方面也存在大大小小的bug，毕竟时间太紧，而任务繁重，故而采取了“轻量化”的方式来实现系统，好歹也算是对这段时间自己学习的一个交代吧。

# 参考文献

[1] <https://gitee.com/minttang/mintadmin>

[2] http://www.w3school.com.cn/