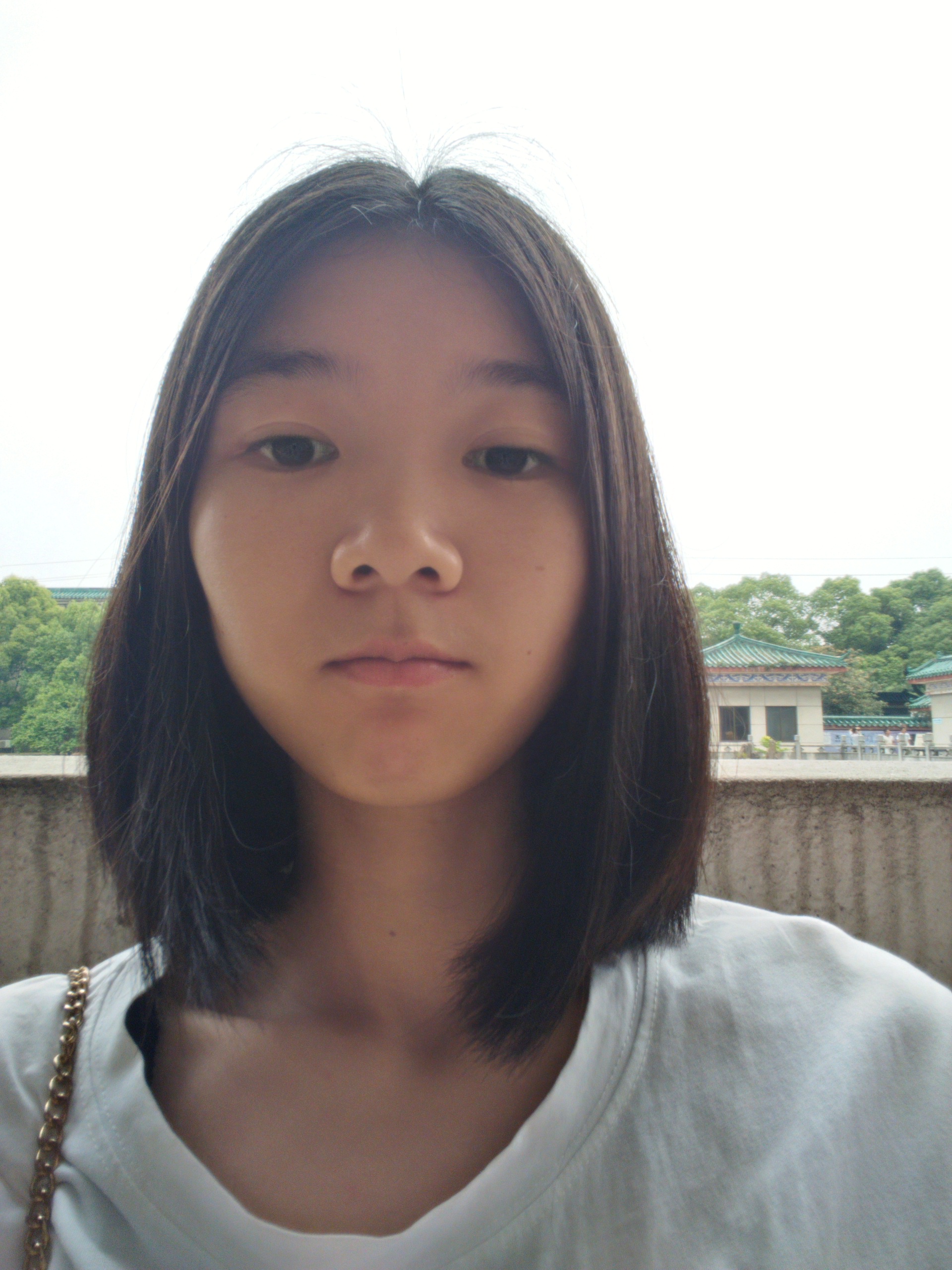


**设计模式与系统架构分析报告**



**系 （院）：**  计算机科学学院

**课 程：** 设计模式与系统架构

**指导教师：**  向华

**专业班级：** 计科11401

**姓 名：** 胡谦

**学 号：** 201403583

**设计时间：** 2017.6.12 - 2017.6.25

目录

[基于MVC博客系统分析报告 3](#_Toc486081830)

[一． 引言 3](#_Toc486081831)

[二． 对于MVC的简单介绍 3](#_Toc486081832)

[三． 设计分析 4](#_Toc486081833)

[功能需求模块分析 4](#_Toc486081834)

[Spring MVC重要组件分析 5](#_Toc486081835)

[涉及到的设计模式分析 6](#_Toc486081836)

[四． 作者介绍 7](#_Toc486081837)

[五． 总结体会 7](#_Toc486081838)

[六．致谢 7](#_Toc486081839)

# 基于MVC博客系统分析报告

## 引言

随着计算机的发展和普及，Blog得到了极大的发展。Blog是Weblog（网络日志）的简称。Blog是即E-mail、BBS、ICQ之后的第四种网络交流方式，是曾经风靡互联网的信息发布工具。本系统采用MVC模式设计实现了一个典型的博客系统，可以很好地表达用户与系统的交互模式以及整个系统的程序架构模式。

## 对于MVC的简单介绍

模型视图控制器（MVC）是80年代Smalltalk 80出现的一种软件设计模式，现在已经被广泛的使用。它包含3个对象：

模型（model）

模型是应用程序的主体部分。模型表示业务数据，或者业务逻辑。

视图（View）

视图是应用程序中用户界面相关的部分，是用户看到并与之交互的界面。

控制器（controller)

控制器的工作就是根据用户的输入，控制用户界面数据显示和更新model对象的状态。

模型

封装应用程序状态

响应状态查询

应用程序功能

通知视图改变

状态改变

状态查询

通知改变

视图

解释模型

模型更新请求

发送用户输入给控制器

允许控制器选择视图

控制器

定义应用程序行为

用户动作映射成模型更新

选择相应视图

视图选择

用户请求

## 设计分析

### 功能需求模块分析

#### 用户注册和登录模块：

此功能模块的功能包括新用户的注册，已注册用户的登陆。用户需要登陆博客系统才能进行相关操作，否则只能浏览和评论。如果不是博客系统的用户，需要先行注册。

#### 博客页面显示模块：

根据用户的设定将博客内容显示给用户，这些内容包括用户的文章及相关评论，用户的个性化信息以及其他信息导航栏目。

#### 文章管理模块：

功能包括新增文章（保存），文章修改（保存），文章删除。

#### 用户信息维护模块：

管理维护用户的个性化信息，包括昵称，个性logo，签名等。

系统模块设计如下图所示

（注）：本模块功能作者并未完全实现，稍微自行添加了一些。

文章管理模块

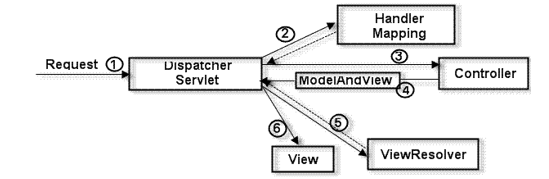
用户信息维护模块

博客页面显示模块

用户注册登陆模块

博客系统

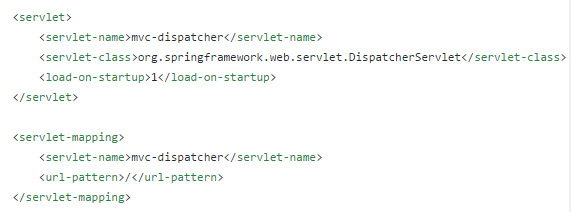
### Spring MVC重要组件分析



一．前端控制器

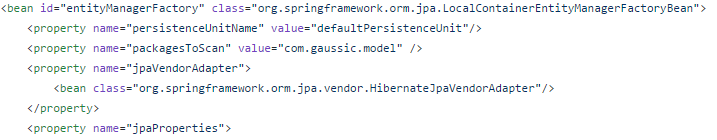
DispatherServlet:是spring mvc的中央调度器，创建时会默认从DispatherServlet.properties文件加载spring mvc所用的各种组件。

用户发送请求到前端控制器



二．处理器映射器

这里用的是simpleURLHandlerMapping，它将url和处理器bean的id进行统一映射配置。

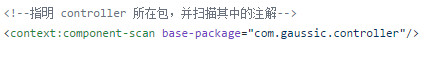


三．处理器适配器

HandlerAdapter会根据适配器及接口对Hander进行包装适配，包装后即可对处理器进行执行(使用了适配器模式)这里使用了简单控制器处理器适配器(SimpleControllerHanderAdapter)，所有实现了org.springframework.web.servlet.mvc.Controller接口的bean通过此适配器进行适配执行。

四．注解处理器映射器和适配器

可以使用组件扫描，而省去在spring容器中配置每个controller类。配置如下：

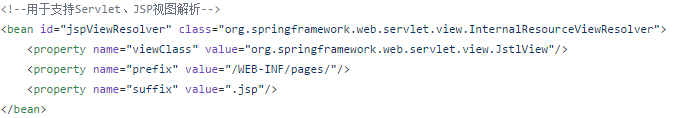


1. 注解处理器映射器，对类中标记的@RequestMapping的方法进行映射，根据RequestMapping定义的url匹配RequestMapping标记的方法，匹配成功返回HandlerMethod对象给前端控制器。
2. 注解处理适配器，对标记的@RequestMapping的方法进行适配。

注：这里采用的如下配置的

C:\Users\谦谦\AppData\Local\Microsoft\Windows\INetCache\Content.Word\开启注解.png

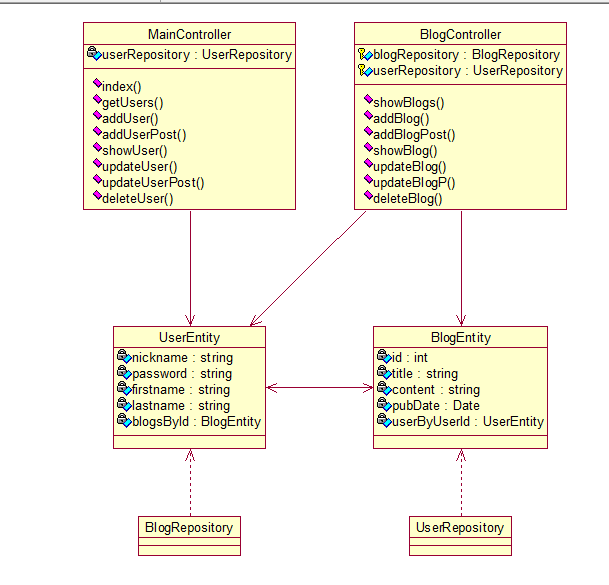
五．视图解析器(View Resolver)，负责将处理结果生成view视图。View Resolver首先根据逻辑视图名解析成物理视图名即具体的页面地址，再生成View视图对象，最后对View进行渲染将处理结果通过页面展示给用户。



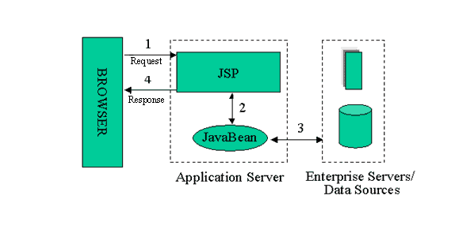
### 涉及到的设计模式分析

用MVC实现的观察者模式，观察目标是MVC模式中的模型（model），而观察者就是MVC中的视图（view），控制器充当两者之间的中介。当模型的数据发生改变时，视图将自动改变其显示内容。

以下为其UML类图：



所用的JSP Model架构如下图：



分析：在UML类图中，没有画出视图（jsp），从上到下，分别为控制层，模型层，数据层。根据不同的页面的请求，推送给不同的控制器，对其做相应的处理使相应的模型改变并更新数据库，随后将改变后的信息再返回给页面。这里就用到了观察者模式，当模型改变时，视图也会随之改变。

## 作者介绍

籍贯：上海

爱好：喜欢AI研究

签名：一位致力于AI研究却不得不兼顾项目的研究生

性格：文艺，内心矛盾但能坚持自己的内心

自己完成的demo：百度新闻热词抓取，OpenBR人脸识别等。然后基本上都是一些自己的学习笔记。

## 总结体会

这个项目是早就找好了的一直放着没看，之所以选择它是因为其内容较简单，而且博客系统理解起来也不会太困难，毕竟知道该有什么模块及功能之类。在仔细看过之后感觉这个项目并不成熟，作者只是搭了个架子（好吧，我不能理解为什么这样的项目能点击量高，也许因为它简单还是），比如说处理的六个具体步骤它一个都没写，起初也没深入研究它，就想着让它运行起来，过程我也不知道有什么问题，反正报错，找不到原因。

通过写这个报告，虽然实际的技术没学到什么，但是基本思想都弄懂了，还有一些不懂的通过百度结合这个例子杂七杂八的也了解了些东西，等有时间系统的学习一下应该能收获更多。

## 六．致谢

感谢老师这么久以来认真负责的教学，通过这门课程让我明白代码原来也可以有这么多的讲究，以前只是注重实现，面向对象的思想虽然有但自己却很少有意识的去用到它，用一个人的评价我就是：老是用面向对象的语言做着面向过程的事情。

附：

开源代码链接：<https://github.com/gaussic/SpringMVCDemo>

个人主页地址: https://rainqian.github.io/ArainQain/