

# **毕业设计（论文）**

题 目 基于java的手机网站发布系统设计与实现

学 院 软件与计算机学院

专 业 软件工程

学生姓名 陈锦华 学号 149000623

指导教师 石磊 职称

2018年 3 月 19 日

**学生毕业设计（论文）原创性声明**

本人以信誉声明：所呈交的毕业设计（论文）是在导师的指导下进行的设计（研究）工作及取得的成果，设计（论文）中引用他（她）人的文献、数据、图件、资料均已明确标注出，论文中的结论和结果为本人独立完成，不包含他人成果及为获得重庆工程学院或其它教育机构的学位或证书而使用其材料。与我一同工作的同志对本设计（研究）所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

毕业设计（论文）作者（签字）：

年 月 日

# 摘 要

随着科技的发展，特别是移动互联网的发展，现阶段的PC网站发布系统越来越不能满足企业的需要，特别是对于一些企业仍然没有搭建手机网站发布系统，这样不仅显得十分落后而且难以适应激烈的市场竞争。设计一种基于Java的手机网站发布系统就应运而生了，手机网站发布系统基本实现了PC网站发布系统的基本应用，包括企业产品管理的增、删、改、查，文章管理的增、删、改、查，用户权限等基本应用，设计的信息管理涉及MySQL数据库的操作，Eclipse以及jdbc数据库的连接等相关知识。

本文主要对手机网站发布系统的设计与开发过程进行了详细阐述。包括系统需求分析、系统设计、数据库设计、编码实现以及系统测试五个方面。系统分析中主要包括了系统设计前的需求分析、业务流程分析、数据流程分析和数据字典分析；而系统设计则是对系统体系结构、各功能模块、数据库结构设计等具体的系统实现过程进行剖析展示，在系统设计章节中有详细的文字解释并配有表图说明。文章还列举了关键的功能模块实现代码。

关键词：手机网站 发布系统

# **ABSTRACT**

With the development of science and technology, especially the development of mobile Internet, the current PC website publishing system is increasingly unable to meet the needs of enterprises, especially for some companies still do not build a mobile website publishing system, this not only looks very backward and difficult to adapt Fierce competition in the market. Designing a Java-based mobile website publishing system came into being. The mobile website publishing system basically implemented the basic application of the PC website publishing system, including the addition, deletion, modification, and inspection of corporate product management, and the addition and deletion of article management. Basic application such as change, check, user authority, etc. The design of information management involves the operation of MySQL database, Eclipse and jdbc database connection and other relevant knowledge.

This article describes the design and development process of the mobile website publishing system in detail. Including system requirements analysis, system design, database design, coding and system testing. The system analysis mainly includes requirements analysis, business process analysis, data flow analysis and data dictionary analysis before system design; and the system design analyzes the specific system implementation process such as system architecture, functional modules, and database structure design. The display is explained in detail in the system design chapter and is provided with a graphic description. The article also lists the key functional module implementation code.

**Keywords:** mobile website；publishing system

目 录

[毕业设计（论文） 1](#_Toc510878458)

[摘 要 I](#_Toc510878459)

[ABSTRACT II](#_Toc510878460)

[1 绪 论 1](#_Toc510878461)

[1.1系统开发背景及其意义 1](#_Toc510878462)

[1.2国内外现状 1](#_Toc510878463)

[2 系统开发环境与技术简介 2](#_Toc510878464)

[2.1 系统开发环境 2](#_Toc510878465)

[2.1.1 Intellij IDEA 2](#_Toc510878466)

[2.1.2 JDK 2](#_Toc510878467)

[2.1.3 MySQL 2](#_Toc510878468)

[2.2 主要技术 3](#_Toc510878469)

[2.2.1 MVC模型 3](#_Toc510878470)

[2.2.2 Spring、Spring MVC与Spring Boot 3](#_Toc510878471)

[2.2.3 Semantic UI与jQuery 4](#_Toc510878472)

[2.2.4 响应式网页设计 4](#_Toc510878473)

[3 需求分析 6](#_Toc510878474)

[3.1可行性分析 6](#_Toc510878475)

[3.1.1技术可行性 6](#_Toc510878476)

[3.1.2 经济可行性 6](#_Toc510878477)

[3.1.3操作可行性 6](#_Toc510878478)

[3.1.4法律可行性 7](#_Toc510878479)

[3.2 性能分析 7](#_Toc510878480)

[3.3 系统主要功能 7](#_Toc510878481)

[3.4 系统用例 7](#_Toc510878482)

[4 系统总体设计与分析 9](#_Toc510878483)

[4.1功能模块设计 9](#_Toc510878484)

[4.2数据字典 9](#_Toc510878485)

[4.3 E-R图 10](#_Toc510878486)

[4.4总体E-R图 13](#_Toc510878487)

[4.5数据表设计 13](#_Toc510878488)

[5 系统详细设计 15](#_Toc510878489)

[5.1数据表脚本设计 15](#_Toc510878490)

[5.2控制层的设计 18](#_Toc510878491)

[5.3业务层的设计 19](#_Toc510878492)

[5.4持久层的设计 20](#_Toc510878493)

[6 系统实现 20](#_Toc510878494)

[6.1用户登录 21](#_Toc510878495)

[6.2系统管理主页面 22](#_Toc510878496)

[6.3文章管理界面 23](#_Toc510878497)

[6.4栏目管理界面 24](#_Toc510878498)

[6.5标签管理界面 24](#_Toc510878499)

[6.6角色管理界面 24](#_Toc510878500)

[6.7网站前台首页 25](#_Toc510878501)

[6.8文章详情界面 26](#_Toc510878502)

[7 系统测试 27](#_Toc510878503)

[7.1软件测试的目的 27](#_Toc510878504)

[7.2软件测试的方法 27](#_Toc510878505)

[7.2.1单元测试 27](#_Toc510878506)

[7.2.2系统功能测试 27](#_Toc510878507)

[7.2.3自动测试与人工测试 28](#_Toc510878508)

[8 结束语 29](#_Toc510878509)

[参考文献 30](#_Toc510878510)

# 1 绪 论

## 1.1系统开发背景及其意义

随着互联网的崛起，网络在当今社会各项事务中都扮演着至关重要的角色，更是成为继报刊、电视、广播等传统媒体之后而兴起的一新传媒介质，网络宣传已成为众多宣传手段的重中之重，各种各样的网站也就应运而生。

因此，当代企业纷纷建立自己的网站，借助于互联网快速的信息传递加速自身的发展。一个企业网站的规划与建设，自然而然也就成了众多类型的网站规划与建设的典范；而目前大部分企业的网站都是针对于PC机的浏览方式，但是在移动互联网的冲击下，建立一个针对于手机用户浏览的网站可以很方便的宣传自己的新产品，可以很及时的了解产品的反馈信息和市场动态，不管是对用户还是企业都有很大的开发价值。因此手机网站无疑成为当今乃至今后一段长时间内，企业内部和互联网上的一个相当重要的讨论对象。

## 1.2 国内外现状

无论是在国内还是在国外，手机网站的建设都还是在刚刚发展的阶段。最近几年得到了一定的发展。尽管各个国内外之间情况不同，但是基于对企业在市场竞争力的角度考虑，企业在建立站点时都需要考虑到手机用户，并且积极的把精力投入到手机站点的用户体验上。

在国内，据工信部统计，中国手机用户数量接近13亿，其中智能手机已经超过了9亿，国内移动互联网的发展始于2005年，目前已进入快速发展期，以WAP为主要方式。在手机网站方面，在过去一年时间里，手机友好型网站的数量增长了7倍，从原来的15万家猛增至大约110万家。手机的普及程度远远超过电脑，由于有线网络的“缺席”，移动网络已经超越前者。

在国外，比如美国，移动互联网发展已进入高速成长期，在过去一年中，使用移动终端浏览新闻、获取信息上升了52%，且呈现出不断加速的趋势，每周至少一次和每天至少一次使用移动互联网的用户数分别同比增张87%和107%。在手机网站方面，从10年的19万家到15年的260万家。

# 2 系统开发环境与技术简介

## 2.1 系统开发环境

### 2.1.1 Intellij IDEA

IntelliJ IDEA 是JetBrains出品的一个Java的IDE（集成开发环境），是目前Java中最强大和最流行的集成开发环境之一。它的功能十分强大，通过一些插件，使得IDEA不仅可以支持Java相关项目的开发，也可以支持包括JavaScript、Scala、Android、Python、PHP等语言的相关项目的开发。

IntelliJ IDEA的主要功能包括编码完成（Code Completion），此功能可以通过分析代码上下文，来给出当前上下文最适合的符号列表，并且允许直接跳转到代码中的类或声明的代码导航。此外，它内置了许多构建系统，比如grunt、bower、gradle、maven等流行的构建工具，还集成了版本控制系统（VCS），像[Git](https://en.wikipedia.org/wiki/Git_(software)" \o "Git (software))、[Mercurial](https://en.wikipedia.org/wiki/Mercurial" \o "Mercurial)、[Perforce](https://en.wikipedia.org/wiki/Perforce" \o "Perforce)和[SVN](https://en.wikipedia.org/wiki/Apache_Subversion" \o "Apache Subversion)等。

Intellij IDEA目前主要由两个版本，一个是免费的社区版，另一个是收费的完全版，完全版可以开发包括Java EE以及Android的项目，而社区版只支持Java程序的运行及开发。

### 2.1.2 JDK

JDK（Java Developer Kit）包含开发Java程序所需的工具，以及运行程序的JRE。这些工具包括编译器（javac.exe），Java应用程序启动器（java.exe），Appletviewer等等。编译器将java代码转换为字节代码。Java应用程序启动器打开JRE，加载类并调用其主要方法。如果我们想编写自己的程序并编译，则需要JDK。对于运行Java程序，JRE就足够了。JRE的目标是执行Java文件，即JRE = JVM + Java包类（如util、math、lang、awt、swing等）+运行时库。JDK主要针对Java开发。也就是说，我们可以创建一个Java文件（借助Java包），编译一个Java文件并运行一个java文件。

新接触Java的人可能会对是否使用JRE或JDK感到困惑。要运行Java应用程序和Applet，只需下载JRE。但是，为了开发Java应用程序和applet并运行它们，需要JDK。

### 2.1.3 MySQL

MySQL 是应用最广泛的开源关系数据库，是许多常见网站、应用程序和商业产品使用的主要关系数据存储。MySQL 有 20 多年的社区开发和支持历史，是一种可靠、稳定而安全的基于 SQL 的数据库管理系统。MySQL 适用于各种使用案例，包括任务关键型应用程序、动态网站以及用于软件、硬件和设备的嵌入式数据库。

MySQL几乎可以在所有平台上运行，包括[Linux](http://searchenterpriselinux.techtarget.com/definition/Linux)，[UNIX](http://searchenterpriselinux.techtarget.com/definition/Unix)和[Windows](http://searchwindowsserver.techtarget.com/definition/Windows)。虽然它可以用于各种应用程序，但MySQL通常与基于Web的应用程序和在线发布相关联，并且是[开放源代码](http://searchenterpriselinux.techtarget.com/definition/open-source)企业堆栈LAMP的重要组成部分。LAMP是一个Web开发平台，它使用[Linux](http://searchenterpriselinux.techtarget.com/definition/Linux)作为操作系统，Apache作为Web服务器，MySQL作为关系数据库管理系统，PHP作为面向对象的脚本语言。

## 2.2 主要技术

### 2.2.1 MVC模型

随着业务逻辑越来越复杂与多样化，用户所需要的功能也越来越多，传统的业务模型已经无法适用于当前的商业状况，所以就把用户所用的界面与商业逻辑分开了，这就产生了所谓的MVC（Model、View、Controller）架构。

MVC通过将应用程序分成三部分来完成这个工作：模型，视图和控制器。模型管理应用程序的基本行为和数据。它可以响应信息请求，响应指令改变信息状态，甚至在信息变化时通知事件驱动系统的观察者。这可能是一个数据库，或任何数量的数据结构或存储系统。简而言之，它是应用程序的数据和数据管理。视图有效地提供了应用程序的用户界面元素。它会将来自模型的数据呈现为适合用户界面的表单。控制器接收用户输入并调用模型对象和视图以执行适当的操作。

### 2.2.2 Spring、Spring MVC与Spring Boot

Spring是Java EE平台上的一个开源框架，是为了解决企业开发的复杂性而创建的，Spring框架的一个最主要的功能就是依赖注入（Dependency Injection）。从全局来说，它主要包括以下几个部分：Spring Core、Spring AOP、Spring ORM、Spring DAO、Spring Web、Spring Web MVC等。有了这几大模块，只需要简单几步就可以搭建起框架，并可以帮助我们开发者简化开发，在企业开发上显著的提升我们的开发效率。

Spring mvc是Spring框架所提供的一个MVC框架，Spring mvc框架是围绕一个DispatcherServlet将请求分派给处理程序的框架设计的，所以说Spring的Web MVC框架与其他许多Web MVC框架一样，是以请求为驱动的，围绕一个中央Servlet进行设计，该中央Servlet将请求分发给控制器，并提供其他功能来促进Web应用程序的开发。Spring MVC具有可配置的处理程序映射，区域设置和主题解析以及对上传文件的支持。默认处理程序基于@Controller和 @RequestMapping注释，提供了广泛的灵活的处理方法。随着Spring 3.0的推出，该@Controller机制还允许我们通过@PathVariable注释和其他功能创建RESTful Web站点和应用程序。

Spring boot是为了简化spring的配置使用，一种用于快速配置spring相关功能的实用工具，提供开箱即用的配置，以构建Spring应用程序。正如我们前文所提到的，Spring在它的平台中集成了各种不同的模块，如Spring core、spring data、spring web等等。有了这个工具，我们会得到一个快速的设置（以后可以自己改变它）。

### 2.2.3 Semantic UI与jQuery

Semantic UI是一个前端UI框架，主要提供了一些常见的开发控件，比如表格框、按钮等，还有一些栅格系统，这个栅格系统可以帮助我们开发者更快速的布局页面，使得开发前端页面更有效率。

jQuery是用来简化前端开发的一个JavaScript库，它的主旨是“Write Less，Do More”，jQuery的出现大大提高了开发JavaScript相关应用的效率，以前完成同一个任务可能需要十几行甚至上百行的JavaScript代码，而有了jQuery，完成同一个任务可能就只需要一两行代码。同时，jQuery也简化了AJAX的使用。jQuery的主要功能包括DOM操作、CSS操作、事件委托与调用、AJAX以及一些工具函数。此外，jQuery还支持插件用来扩展jQuery本身不具备的功能。

### 2.2.4 响应式网页设计

响应式网页设计（Responsive Web Design）是一种网页设计方法，可使网页在各种设备和窗口或屏幕尺寸上呈现良好效果。随着移动设备拥有量的增加和传统个人电脑销售的放缓，响应式设计在过去几年变得越来越重要。在网页设计和开发领域，我们很快就无法跟上无尽的新决议和新设备。对于许多网站来说，为每个分辨率和新设备创建网站版本将是不可能的，或者至少是不切实际的。

响应式网页设计由三个开发原则组成。要正常工作，所有这三项都需要实施：

①流体网格

灵活的基于网格的布局是响应式设计的基石。它使用相对大小来适应设备的屏幕大小。术语“网格”有点误导，因为没有必要实现任何可用的网格框架。相反，CSS用于定位内容。这种方法基于百分比，与传统的基于像素的设计原则背道而驰。响应式设计远离基于像素的方法，因为一个设备上的像素可能是另一个设备上的八个像素。通过以百分比为基础的文本大小，宽度和边距，固定大小可以变成相对于其显示空间的大小。

②媒体查询

媒体查询（也称为断点）可用于根据设备的功能应用不同的样式。该网站会检测您正在使用的设备类型或您的网络浏览器的大小，并正确显示该页面。要看到这一点，请将网页浏览器的窗口拉伸至不同的尺寸。注意页面如何调整。功能可用于控制宽度，高度，最大宽度，最大高度，设备高度，方向，高宽比等。

③灵活的图像和媒体

此功能允许您调整图像和其他媒体的加载方式，具体取决于设备，可以通过缩放或使用CSS溢出属性进行加载。在CSS中缩放比较简单 - 媒体元素的最大宽度可以设置为100％，Web浏览器将根据其容器使图像缩小和展开。缩放媒体的替代方法是使用CSS对其进行裁剪。应用overflow：hidden可以让图像动态裁剪，以便它们适合其容器

# 3 需求分析

## 3.1 系统功能需求

根据相关的实际情况，本系统主要功能设计有四大部分，分别为文章管理、栏目管理、标签管理和用户角色管理。

文章管理：管理员和普通用户可以发布自己的文章，并且可以对发布后的文章进行增删改查。

①栏目管理：管理员能够对文章所属的栏目进行管理，包括对栏目的增删改查。

②标签管理：管理员能对不同的标签进行管理

③角色管理：管理员能够对普通用户的访问权限进行管理

## 3.2非功能需求

### 3.2.1 性能需求

本系统要正常流畅的运行可以从以下几个角度着手：第一就是这个系统需要稳定的运行，主流程上不能有任何的故障，一旦运行出现一些故障，我们也能够及时的修复；第二是系统的扩展性强，我们能够在系统现有功能稳定运行的基础之上，额外添加一些新功能而不会影响到现有的功能；第三是本系统尽量能够不影响客户机中的其他系统，保持独立运行；最后是保证本系统的页面尽量考虑到美观与良好的用户体验，也要考虑到手机端的用户，页面操作尽量简单不能太复杂。

### 3.2.2 设计约束

本系统操作应简洁易懂，主要采用了B/S的开发模式，换句话说，只要用户具备基本的计算机操作知识，他就可以使用本系统，并且本系统采用的SemanticUI框架的一些控件也是参考到了主流用户的操作习惯，非常的简洁易懂，错误提示也非常的人性化，只要用户有过浏览网站的经验，那么他就可以正常的使用本系统。

### 3.2.3 硬件需求

服务器：双核及以上CPU,内存2G及以上,160G及以上硬盘；

客户端：奔腾III及以上CPU,内存2G及以上,20G以上硬盘；兼容IE9及以上浏览器

## 3.3 系统用例

用例图主要描述系统的功能模型，由参与者（Actor）和用例（Use Case）组成，

用例图(User Case)是一个系统功能的模型图，它可以表示一些参与者和一些用例，并对它们之间的关系进行建模，主要用于系统、子系统或类的功能行为。用例图显示了用例和用例参与者是如何相互关联的。用例图用于可视化系统、子系统或类的行为，使用户能够理解如何使用这些元素，并使开发人员能够实现这些元素。本发布系统的普通用户用例图如图3.1所示：

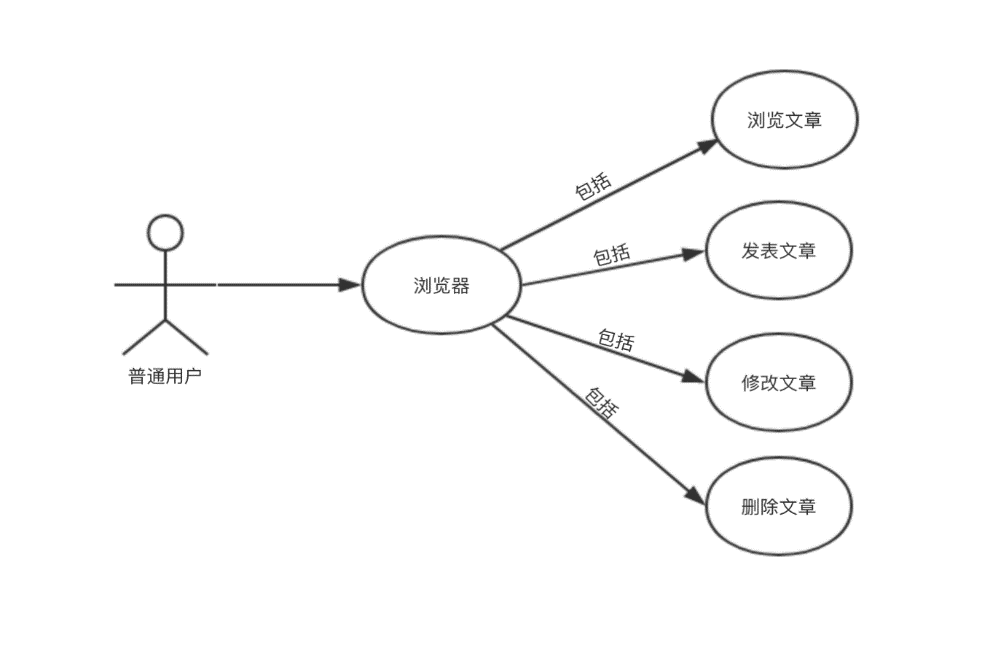


图3.1 普通用户用例图

本发布系统的管理员用例图如图3.2所示：

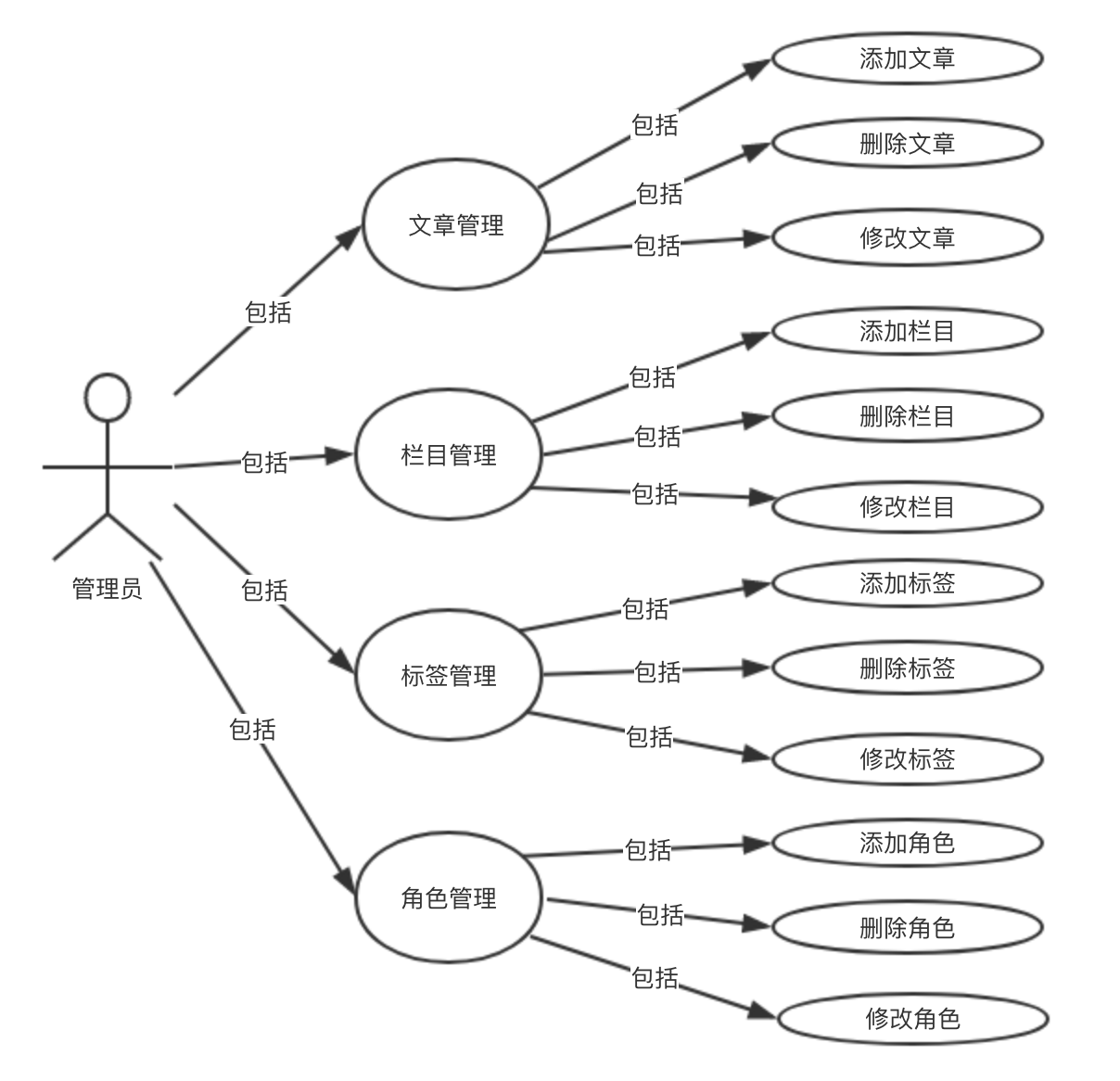


图3.2 系统管理员用例图

# 4. 系统总体设计与分析

## 4.1功能模块设计

根据分析，本发布系统可分为以下三个模块：登录模块、前台模块、后台管理模块。其系统模块图4.1所示。

前台展示模块

后台管理模块

登录模块

用户登录

手机网站发布系统

文章浏览

栏目浏览

评论发表

文章管理

栏目管理

角色管理

标签管理

文章查找

图4.1 系统功能模块图

管理员登录

## 4.2数据字典

数据字典是系统中各种数据描述的集合，是收集和分析数据的主要结果。

数据字典在数据库设计中扮演着重要的角色。

数据字典部分如下：

表3.1 文章信息表

|  |  |
| --- | --- |
| 数据流  系统名：手机网站发布系统  条目名：文章信息 | |
| 来源：管理员 | 去处：文章浏览与查询 |
| 数据结构：  文章信息：{ 编号，标题，正文，feature 图片，作者，评论列表，创建时间，修改时间 } | |
| 简要说明：文章信息，主要是登录用户可以对已发表文章进行增删改查 | |

表3.2 栏目信息表

|  |  |
| --- | --- |
| 数据流  系统名：手机网站发布系统  条目名：栏目信息 | |
| 来源：栏目信息 | 去处：栏目信息查询 |
| 数据结构：  栏目信息：{ 编号，名称，文章列表} | |
| 简要说明：栏目信息表主要是查看栏目的当前名称和下属文章列表 | |

表3.3 评论信息表

|  |  |
| --- | --- |
| 数据流  系统名：手机网站发布系统  条目名：评论信息 | |
| 来源：评论信息 | 去处：评论的查看 |
| 数据结构：评论信息表主要是查看评论的相关信息  菜单信息：{ 编号，作者名称，内容，评论创建时间} | |
| 简要说明：评论详情表，主要是用户对文章下评论的添加与修改 | |

表3.4 用户信息表

|  |  |
| --- | --- |
| 数据流  系统名：手机网站发布系统  条目名：用户信息表 | |
| 来源：用户信息 | 去处：用户信息查询 |
| 数据结构：  用户信息：{ 编号，用户名，密码，邮箱，文章列表，是否是管理员，角色名称，创建时间} | |
| 简要说明：用户信息，管理员能查看所有注册用户 | |

## 4.3 E-R图

实体关系（ER）图是一种流程图，用于说明诸如人员，对象或概念之类的“实体”如何在系统内相互关联。ER图最常用于设计或调试软件工程，商业信息系统，教育和研究领域的关系数据库。也称为ERD或ER模型，它们使用一组定义的符号，如矩形，菱形，椭圆形和连接线来描述实体，关系及其属性的相互关系。它们反映语法结构，实体作为名词和关系作为动词。ER模型通常是系统分析的结果，用于定义和描述在一个业务领域中对流程的重要性。它没有定义业务流程，它仅以图形形式呈现业务数据模式。它通常用图形形式表示为框（实体），它们通过表示实体之间的关联和依赖关系的线（关系）相连。

实体不仅可以通过关系进行表征，还可以通过附加属性（属性）进行表征，其中包括称为“主键”的标识符。为了表示属性以及实体和关系而创建的图可以称为实体 - 属性 - 关系图，而不是实体 - 关系模型。ER模型通常作为数据库来实现。在一个简单的关系数据库实现中，表的每一行代表一个实体类型的实例，并且表中的每个字段代表一个属性类型。在关系数据库中，实体之间的关系通过将一个实体的主键存储为另一个实体的表中的指针或“外键”来实现。

ER模型中包含“实体”，“联系”和“属性”。在本系统中，多对多的关系用（m-n）表示。本系统的主要的实体有文章，评论，标签，用户，角色等。实体ER图如下所示：

图4.1 文章ER图

文章

评论

图4.2 评论ER图

图4.3 栏目ER图

栏目

用户

图4.4 用户ER图

图4.5 标签ER图

标签

1

角色

图4.6

## 4.4总体E-R图

通常进行逻辑设计是为了将概念设计模型转换为符合所选机器上DBMS支持的数据模型的逻辑结构。这些模型应该满足用户在功能、完整性和一致性约束以及数据库可扩展性方面的要求。因此，根据上节的ER图，我们的总体ER图如下：

栏目

文章

评论

用户

标签

拥有

拥有

发表

拥有

发表

角色

1

m

m

n

m

1

m

m

1

1

图4.7

## 4.5数据表设计

下面将详细说明关键数据表的数据库设计。

表4.5 文章表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 说明 | 索引 | 是否为空 |
| id | Int | 4 | 编号 | Primary Key | NO |
| title | nVarchar | 50 | 文章标题 |  | NO |
| content | text |  | 文章内容 |  | Yes |
| comments | Comment |  | 评论列表 |  | Yes |
| authorId | nVarchar |  | 文章作者id |  | No |
| createTime | Date |  | 创建时间 |  | Yes |

该表是用来记录文章相关信息，在文章表里面有个authorid，这个是用来关联发表文章的用户的信息

表4.6 栏目表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 说明 | 索引 | 是否为空 |
| id | Int | 4 | 编号 | Primary Key | NO |
| name | nVarchar | 50 | 栏目名称 |  | NO |
| articles | nVarchar |  | 文章列表 |  | Yes |
| createTime | Date |  | 创建时间 |  | Yes |

该表是用来记录栏目相关信息。

表4.7 标签表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 说明 | 索引 | 是否为空 |
| id | Int | 4 | 编号 | Primary Key | NO |
| name | nVarchar | 50 | 标签名称 |  | NO |
| articles | nVarchar |  | 文章列表 |  | Yes |
| createTime | Date |  | 创建时间 |  | Yes |

该表是用来记录标签相关信息，标签和文章是多对多的关系，表示一个标签可以有多篇文章，一篇文章可以有多个标签，而栏目和文章是一对多的关系，一个栏目可以有多篇文章，一篇文章只能对应一个栏目。

表4.8 角色表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 说明 | 索引 | 是否为空 |
| id | Int | 4 | 编号 | Primary Key | NO |
| name | nVarchar | 50 | 角色名称 |  | NO |

该表是用来记录角色相关信息，默认有两个角色，一个是管理员，另一个是普通用户

表4.9 用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 说明 | 索引 | 是否为空 |
| id | Int | 4 | 编号 | Primary Key | NO |
| username | nVarchar | 50 | 用户名称 |  | NO |
| password | nVarchar |  | 注册密码 |  | Yes |
| createTime | Date |  | 创建时间 |  | Yes |
| role | nVarchar |  | 角色名称 |  | No |
| isAdmin | boolean | 2 | 是否为管理员 |  | No |

该表是用来记录用户信息，里面有role字段，这个字段用来记录用户的角色信息，每个角色对应一定的权限，默认普通用户角色。还有一个字段是isAdmin，这个字段表示此用户是否为管理员，默认为否

# 5 系统详细设计

## 5.1数据表脚本设计

根据平台的需求分析和功能模块的划分，所涉及的实体表有：

1）文章信息：用于保存和更改各个文章的详细信息，包括文章标题、内容、创建时间等。

2）栏目信息：用于保存和更改栏目的信息。

3）标签信息：用于保存和更改标签的信息。

4）角色信息：用于保存和更改用户所对应的角色信息。

5）用户信息：用于保存和更改用户的详细信息。

6）评论信息：用户保存和更改文章下的评论信息

因此系统的定义数据库主要包含7张表：文章表t\_blog, 栏目表t\_type, 标签表t\_tag,角色表t\_role, 用户表t\_user, 评论表t\_comment，文章评论表：t\_blog\_comments

文章表t\_blog：

CREATE TABLE `t\_blog` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`commentabled` bit(1) NOT NULL,

`content` longtext COLLATE utf8\_bin,

`create\_time` datetime DEFAULT NULL,

`description` varchar(255) COLLATE utf8\_bin DEFAULT NULL,

`first\_picture` varchar(255) COLLATE utf8\_bin DEFAULT NULL,

`published` bit(1) NOT NULL,

`title` varchar(255) COLLATE utf8\_bin DEFAULT NULL,

`update\_time` datetime DEFAULT NULL,

`views` int(11) DEFAULT NULL,

`type\_id` bigint(20) DEFAULT NULL,

`user\_id` bigint(20) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `FK292449gwg5yf7ocdlmswv9w4j` (`type\_id`),

KEY `FK8ky5rrsxh01nkhctmo7d48p82` (`user\_id`),

CONSTRAINT `FK292449gwg5yf7ocdlmswv9w4j` FOREIGN KEY (`type\_id`) REFERENCES `t\_type` (`id`),

CONSTRAINT `FK8ky5rrsxh01nkhctmo7d48p82` FOREIGN KEY (`user\_id`) REFERENCES `t\_user` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

栏目表t\_type:

CREATE TABLE `t\_type` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`name` varchar(255) COLLATE utf8\_bin NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

标签表t\_tag:

CREATE TABLE `t\_tag` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`name` varchar(255) COLLATE utf8\_bin DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

角色表t\_role：

CREATE TABLE `t\_role` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`name` varchar(255) COLLATE utf8\_bin DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

用户表t\_user:

CREATE TABLE `t\_user` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`create\_time` datetime DEFAULT NULL,

`email` varchar(255) COLLATE utf8\_bin DEFAULT NULL,

`username` varchar(255) COLLATE utf8\_bin DEFAULT NULL,

`password` varchar(255) COLLATE utf8\_bin DEFAULT NULL,

`update\_time` datetime DEFAULT NULL,

`role\_id` bigint(20) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `FK76cs7cu4h4y8vc1vd4qyw36rt` (`role\_id`),

CONSTRAINT `FK76cs7cu4h4y8vc1vd4qyw36rt` FOREIGN KEY (`role\_id`) REFERENCES `t\_role` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

评论表t\_comment：

CREATE TABLE `t\_comment` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`content` varchar(255) COLLATE utf8\_bin DEFAULT NULL,

`create\_time` datetime DEFAULT NULL,

`email` varchar(255) COLLATE utf8\_bin DEFAULT NULL,

`username` varchar(255) COLLATE utf8\_bin DEFAULT NULL,

`blog\_id` bigint(20) DEFAULT NULL,

`parent\_comment\_id` bigint(20) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `FKke3uogd04j4jx316m1p51e05u` (`blog\_id`),

KEY `FK4jj284r3pb7japogvo6h72q95` (`parent\_comment\_id`),

CONSTRAINT `FK4jj284r3pb7japogvo6h72q95` FOREIGN KEY (`parent\_comment\_id`) REFERENCES `t\_comment` (`id`),

CONSTRAINT `FKke3uogd04j4jx316m1p51e05u` FOREIGN KEY (`blog\_id`) REFERENCES `t\_blog` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

文章标签表t\_blog\_tags:

CREATE TABLE `t\_blog\_tags` (

`blogs\_id` bigint(20) NOT NULL,

`tags\_id` bigint(20) NOT NULL,

KEY `FK5feau0gb4lq47fdb03uboswm8` (`tags\_id`),

KEY `FKh4pacwjwofrugxa9hpwaxg6mr` (`blogs\_id`),

CONSTRAINT `FK5feau0gb4lq47fdb03uboswm8` FOREIGN KEY (`tags\_id`) REFERENCES `t\_tag` (`id`),

CONSTRAINT `FKh4pacwjwofrugxa9hpwaxg6mr` FOREIGN KEY (`blogs\_id`) REFERENCES `t\_blog` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

## 5.2控制层的设计

该系统使用spring mvc来处理控制层。SpringMVC是一个核心部分主要由DispatcherServlet处理的MVC框架，它首先访问DispatcherServlet负责将每个请求转发给相应的处理程序并返回到相应的模型（Model）和视图（View）。 在使用注解时，Spring MVC处理器处理程序使用@ Controller和@ RequestMapping注释进行注释，这些注释提供了非常灵活的匹配和处理方法。在使用注解时，Spring MVC处理器handler使用@ Controller和@ RequestMapping注释进行注释，这些注释提供了非常灵活的匹配和处理方法。

作为springmvc的主调度程序，DispatcherServlet默认从DispatcherServlet.properties文件创建由springmvc使用的子程序集。如果在springmvc.xml中配置，则存在DispatcherServlet，如果在springmvc.xml中配置，则会减少springmvc程序集之间的耦合。本系统的控制器类图如图所示：

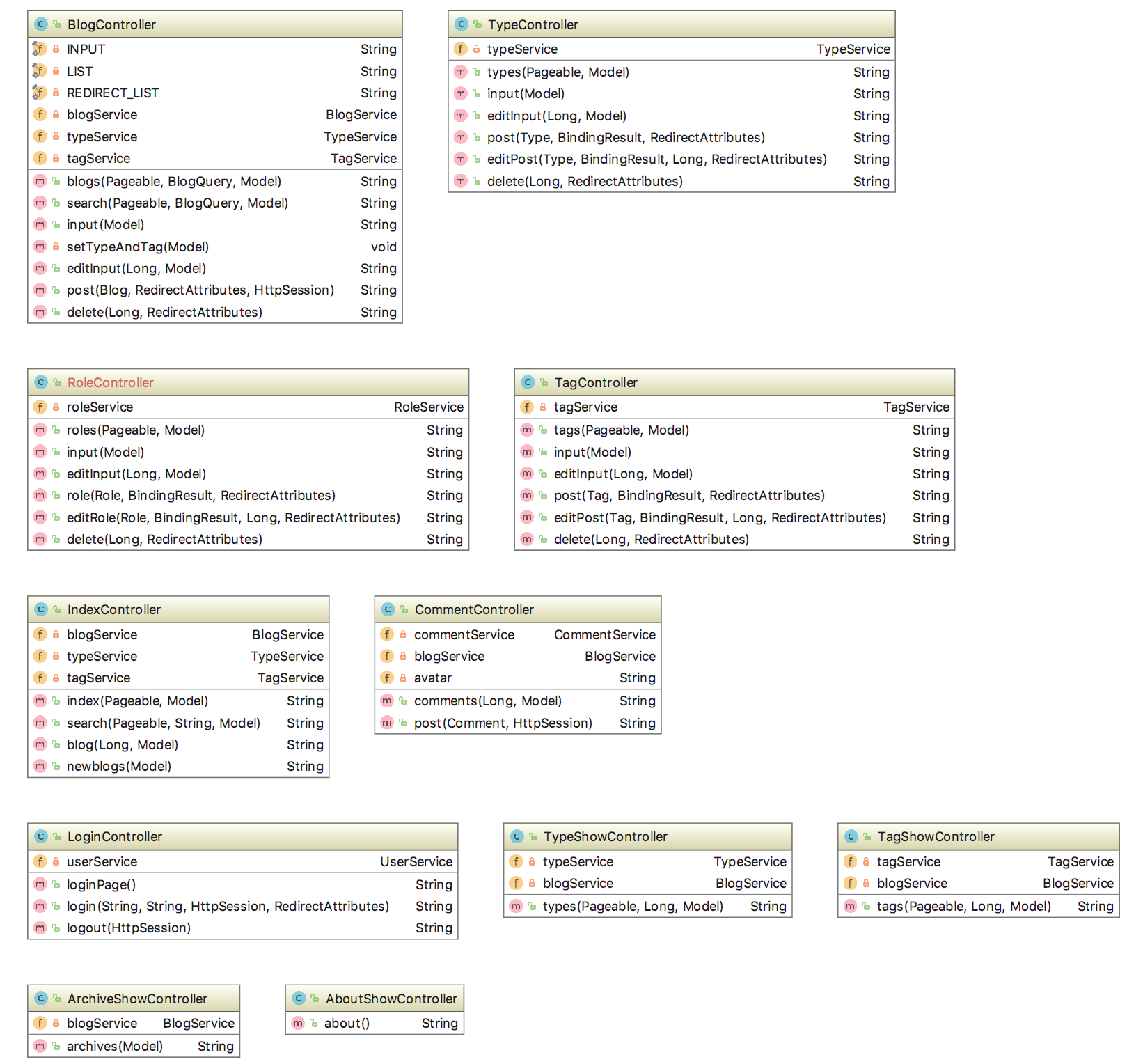


图5.1 控制层类图

## 5.3业务层的设计

系统的业务处理主要是由业务层来完成的。当用户的请求被控制层接收时，将在业务层上调用一个方法，该方法调用持久层以将数据保存到数据库或将消息返回给控制层。 一般来说，业务层可以持久化数据以将数据存储到数据库中，但该系统定义了业务层和持久层以便实现系统的灵活扩展。调用方法主要由业务层来负责，而数据库的操作则有持久层来完成。

业务层类图如图所示：

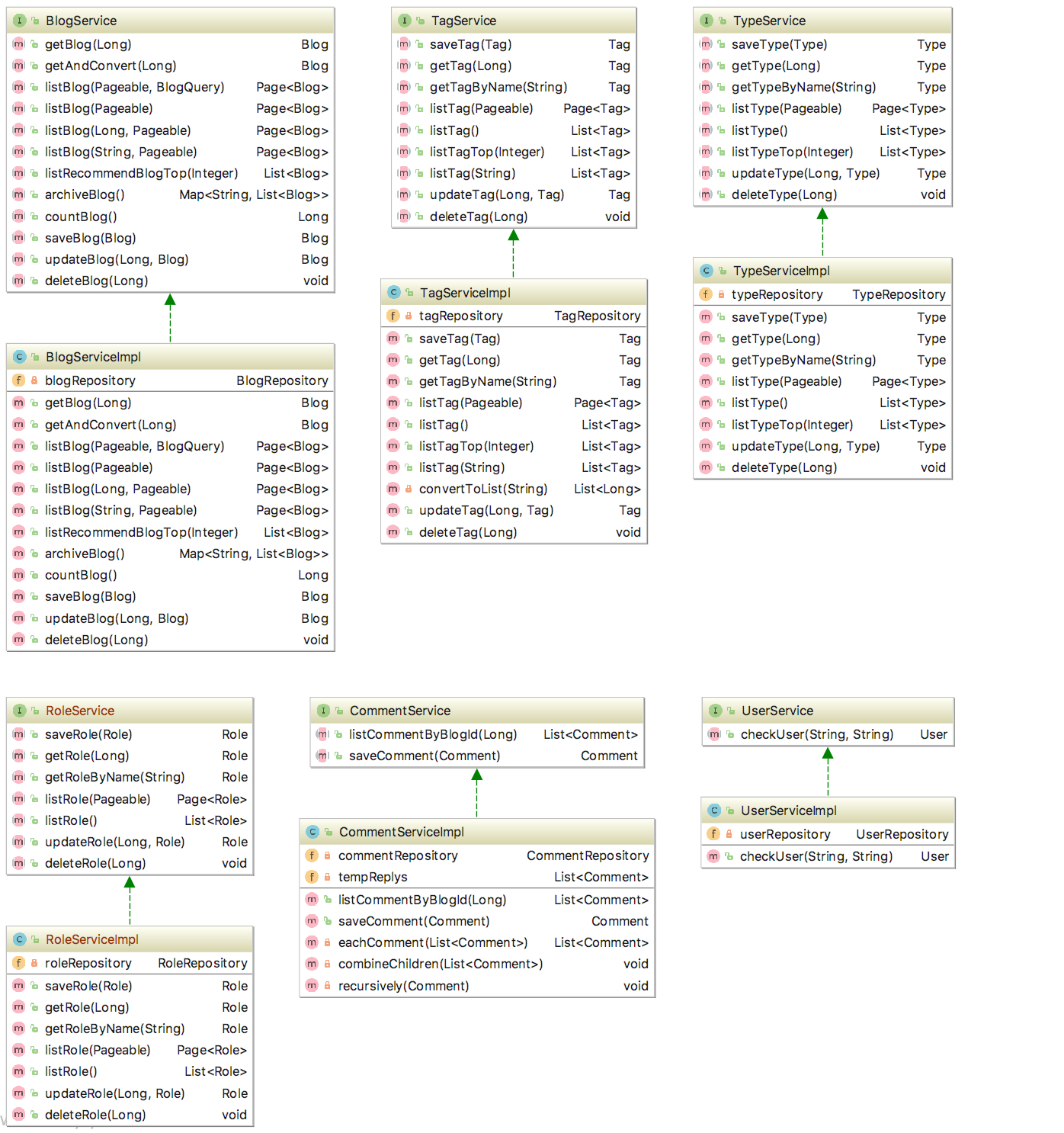


图5.2 业务层类图

## 5.4持久层的设计

一些数据库的增删改查方法主要包括在持久层里。系统中用户操作产生的数据是临时的，随时更新，系统处理后这些数据将消失，因此需要使用持久层来将有用数据存储到数据库中，以确保及时获得所需数据。

该系统通过JDBC完成数据库访问。现在不同的数据库厂商提供访问数据库的JDBC接口提供对各种数据库接口的访问，在数据库对象中有关系模型;因此，在访问数据库时，每个字段对应于每个字段。由于该链接很容易出错，因此该系统使用Hibernate持久层来实现对象和关系之间的一一对应关系。

添加Hibernate持久层提高了生产力，改善了软件架构，并且在编写代码和系统维护时变得更加容易。特别是在大型应用中，它将更加有利。同时，持久层作为一个单独的层可以为该层独立开发一个软件包，允许它们坚持各种应用数据并为上层提供服务。

因此，在企业中进行应用程序开发的开发人员不需要执行数据持久性的底层实现，而不需要直接调用由持久层提供的API。

持久层类图如图所示：

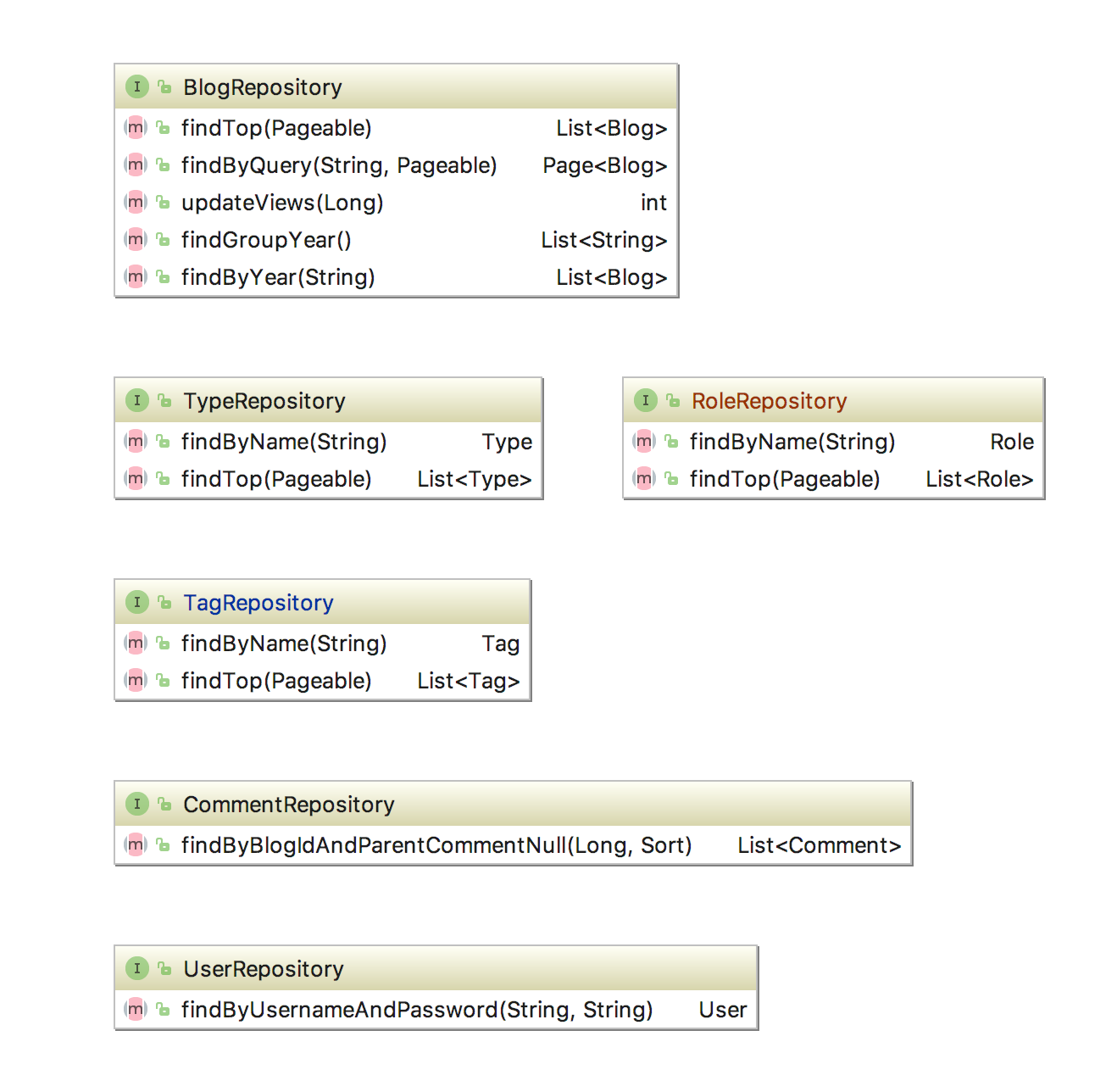


图5.3 持久层类图

6 系统实现

## 6.1用户登录

登录界面如图6.1所示：



图6.1 用户登录界面

登录关键代码如下：

@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"**, **"用户名和密码错误"**);  
 **return "redirect:/admin"**;  
 }  
}

本段代码是程序的登陆代码，首先从获取从页面输入的用户名和密码。登录成功后就会跳转到管理后台主页面；登录失败后就会重定向至原页面，并提示用户用户名或者密码错误。

## 6.2系统主菜单管理

后台管理主页面如图6.2所示：

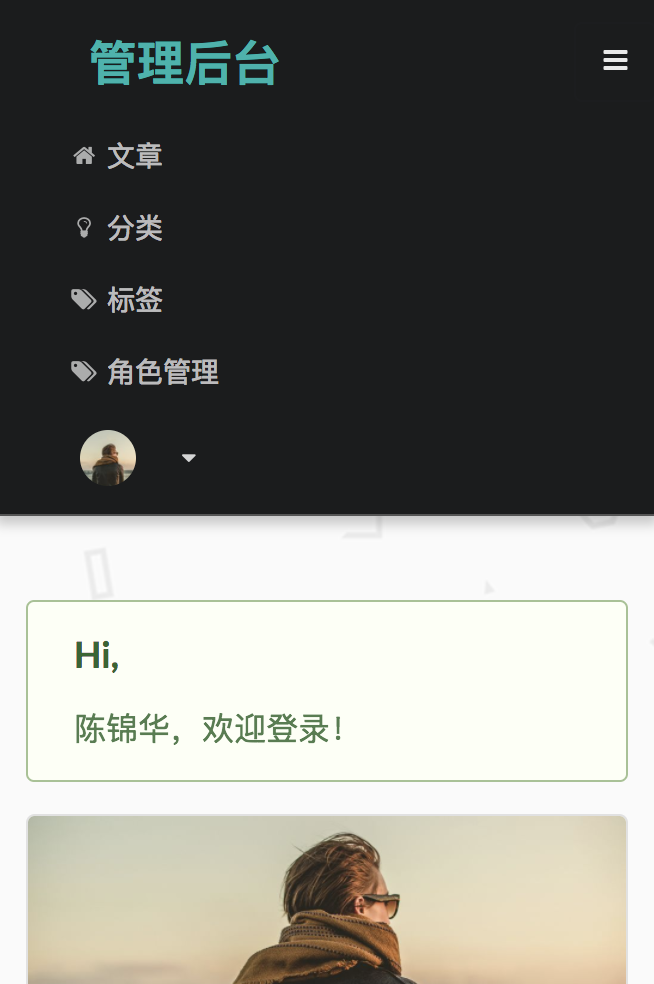


图6.2 后台管理主界面

用户登录成功后，根据用户的角色值的不同显示不同的菜单。本系统所实现的功能，以及本系统的一些业务概况。主要包括文章的添加，删除，修改，查看。栏目信息的添加删除，修改、查阅。标签的添加，删除，修改，更新，等等。各种功能的实现。

关键代码如下：

@GetMapping(**"/"**)  
**public** String index(@PageableDefault(size = 8, sort = {**"updateTime"**}, direction = Sort.Direction.***DESC***) Pageable pageable,  
 Model model) {  
 model.addAttribute(**"page"**,**blogService**.listBlog(pageable));  
 model.addAttribute(**"types"**, **typeService**.listTypeTop(6));  
 model.addAttribute(**"tags"**, **tagService**.listTagTop(10));  
 model.addAttribute(**"recommendBlogs"**, **blogService**.listRecommendBlogTop(8));  
 **return "index"**;  
}

## 6.3文章管理

文章管理界面如图6.3所示：

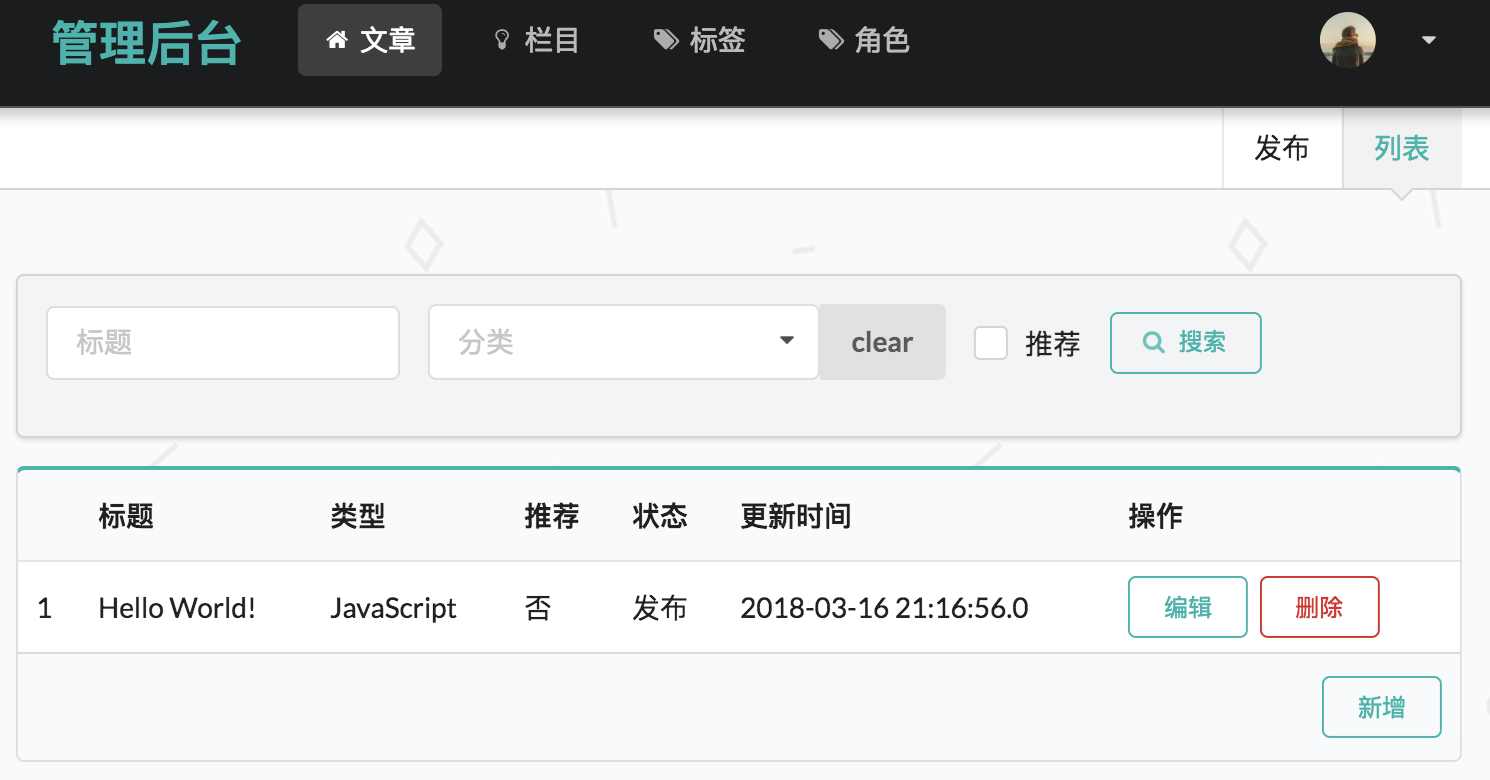


图6.3 文章管理主界面

用户登录成功后，可以点击页面上方的菜单栏中的“文章”选项进入此界面，在这里用户可以发表文章、查看已发表的文章、根据文章标题搜索文章、修改文章以及删除已发布的文章。

关键代码如下：

**public interface** BlogRepository **extends** JpaRepository<Blog, Long>, JpaSpecificationExecutor<Blog> {  
 @Query(**"select b from Blog b where b.recommend = true"**)  
 List<Blog> findTop(Pageable pageable);  
 @Query(**"select b from Blog b where b.title like ?1 or b.content like ?1"**)  
 Page<Blog> findByQuery(String query,Pageable pageable);  
 @Transactional  
 @Modifying  
 @Query(**"update Blog b set b.views = b.views+1 where b.id = ?1"**)  
 **int** updateViews(Long id);  
 @Query(**"select function('date\_format',b.updateTime,'%Y') as year from Blog b group by function('date\_format',b.updateTime,'%Y') order by year desc "**)  
 List<String> findGroupYear();  
 @Query(**"select b from Blog b where function('date\_format',b.updateTime,'%Y') = ?1"**)  
 List<Blog> findByYear(String year);  
}

## 6.4栏目管理

栏目管理界面如图6.3所示：



图6.4 栏目管理主界面

用户在这里可以增加新的栏目、查看已有的栏目、编辑栏目、以及删除栏目。

关键代码如下：

@PostMapping(**"/types"**)  
**public** String post(@Valid Type type,BindingResult result, RedirectAttributes attributes) {  
 Type type1 = **typeService**.getTypeByName(type.getName());  
 **if** (type1 != **null**) {  
 result.rejectValue(**"name"**,**"nameError"**,**"不能添加重复的栏目"**);  
 }  
 **if** (result.hasErrors()) {  
 **return "admin/types-input"**;  
 }  
 Type t = **typeService**.saveType(type);  
 **if** (t == **null** ) {  
 attributes.addFlashAttribute(**"message"**, **"新增失败"**);  
 } **else** {  
 attributes.addFlashAttribute(**"message"**, **"新增成功"**);  
 }  
 **return "redirect:/admin/types"**;  
}

## 6.5标签管理

标签管理界面如图6.5所示：



图6.5 标签管理主界面

用户在这里可以增加新的标签、查看已有的标签、编辑标签、以及删除标签。

关键代码如下：

@GetMapping(**"/tags"**)  
**public** String tags(@PageableDefault(size = 10,sort = {**"id"**},direction = Sort.Direction.***DESC***)  
 Pageable pageable, Model model) {  
 model.addAttribute(**"page"**,**tagService**.listTag(pageable));  
 **return "admin/tags"**;  
}  
  
@GetMapping(**"/tags/input"**)  
**public** String input(Model model) {  
 model.addAttribute(**"tag"**, **new** Tag());  
 **return "admin/tags-input"**;  
}  
  
@GetMapping(**"/tags/{id}/input"**)  
**public** String editInput(@PathVariable Long id, Model model) {  
 model.addAttribute(**"tag"**, **tagService**.getTag(id));  
 **return "admin/tags-input"**;  
}

## 6.6角色管理

角色管理界面如图6.6所示：



图6.6 角色管理主界面

用户在这里可以增加新的角色、查看已有的角色、编辑角色、以及删除角色。

关键代码如下：

@Autowired  
**private** RoleService **roleService**;  
  
@GetMapping(**"/roles"**)  
**public** String roles(@PageableDefault(size = 10,sort = {**"id"**},direction = Sort.Direction.***DESC***)  
 Pageable pageable, Model model) {  
 model.addAttribute(**"page"**,**roleService**.listRole(pageable));  
 **return "admin/roles"**;  
}  
  
@GetMapping(**"/roles/input"**)  
**public** String input(Model model) {  
 model.addAttribute(**"role"**, **new** Role());  
 **return "admin/roles-input"**;  
}

## 6.7网站前台

网站前台主界面在手机上的效果如图6.7所示：

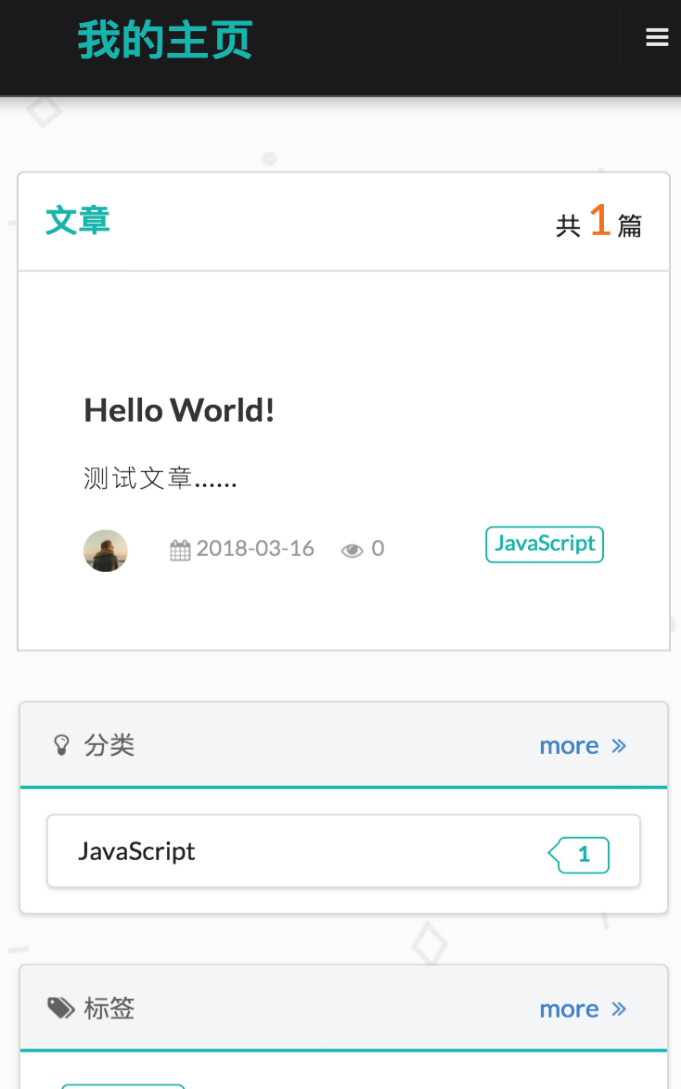


图6.7 网站前台首页

进入网站的所有人都可以查看已发表的文章、分类以及标签。点击文章标题就可以查看当前文章的内容了。

关键代码如下：

*<!--导航-->*<**nav th:replace="\_fragments :: menu(1)" class="ui inverted attached segment m-padded-tb-mini m-shadow-small"** >  
 <**div class="ui container"**>  
 <**div class="ui inverted secondary stackable menu"**>  
 <**h2 class="ui teal header item"**>Blog</**h2**>  
 <**a href="#" class=" m-item item m-mobile-hide"**><**i class="mini home icon"**></**i**>首页</**a**>  
 <**a href="#" class="m-item item m-mobile-hide"**><**i class="mini idea icon"**></**i**>分类</**a**>  
 <**a href="#" class="m-item item m-mobile-hide"**><**i class="mini tags icon"**></**i**>标签</**a**>  
 <**a href="#" class="m-item item m-mobile-hide"**><**i class="mini clone icon"**></**i**>归档</**a**>  
 <**a href="#" class="m-item item m-mobile-hide"**><**i class="mini info icon"**></**i**>关于我</**a**>  
 <**div class="right m-item item m-mobile-hide"**>  
 <**div class="ui icon inverted transparent input m-margin-tb-tiny"**>  
 <**input type="text" placeholder="Search...."**>  
 <**i class="search link icon"**></**i**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 <**a href="#" class="ui menu toggle black icon button m-right-top m-mobile-show"**>  
 <**i class="sidebar icon"**></**i**>  
 </**a**>  
</**nav**>

## 6.8文章详情

文章详情界面如图6.8所示：

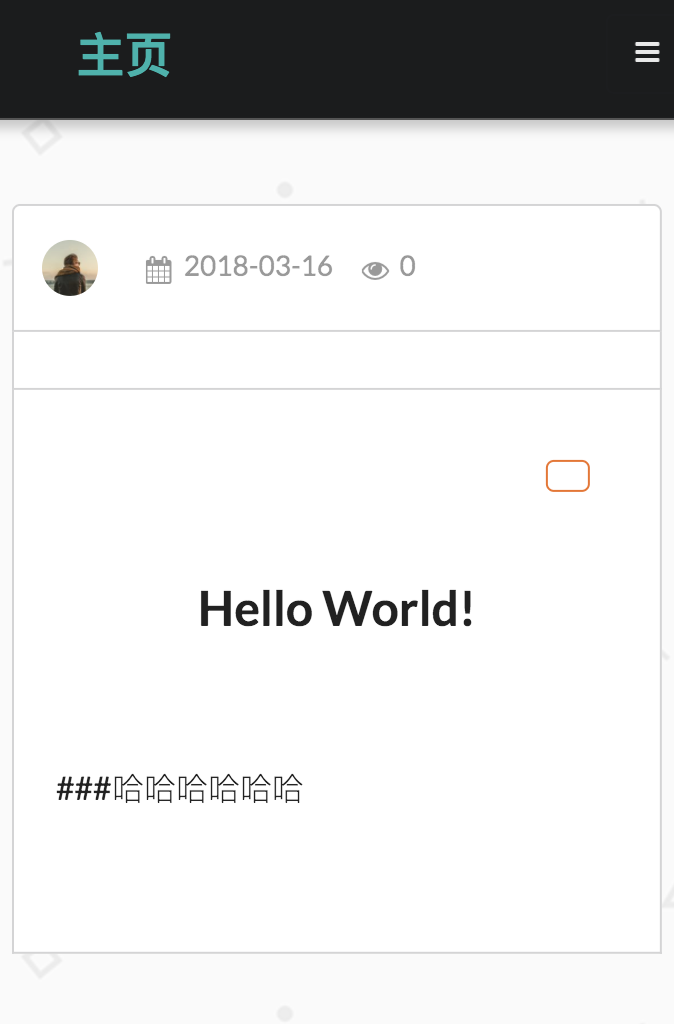


图6.8 文章详情界面

# 7 系统测试

## 7.1软件测试的目的

软件测试意味着检查开发的软件是否存在软件初始设计中的任何错误和问题。通过软件测试，可以完成软件程序的验证，并且可以轻松检测到任何人为错误。软件测试是一个持续的过程，而不是单一的活动。测试是贯穿整个软件开发生命周期的过程。

软件测试对于纠正软件应用程序中的任何错误并使其无错而言非常重要。一个人总是倾向于犯错误，通过使用这个测试，可以很容易地纠正人为错误。总体而言，软件测试具有许多优点，例如：

1.软件测试是一种帮助软件开发人员在编码阶段检测错误并为他们提供改进软件应用程序编码的正确建议的工具。

2.测试将帮助软件开发人员为他们的客户提供准确的帮助，并在一天结束时拥有满意的客户。

3.保证软件质量和让客户满意非常重要。它有助于软件开发人员通过向客户提供质量更好的软件来提高市场竞争力。

4.需要为客户提供高质量产品或软件应用所需的大量设施，这些产品或软件应用需要较低的维护成本，从而产生更准确，一致和可靠的结果。

5.构建能够按照客户要求工作的高性能软件应用程序非常重要。

6.在运行阶段测试软件可以提供准确的读数，如果在这个阶段发现任何错误，那么在这个阶段很容易纠正错误，并且它也比较便宜。

7.如果想在业务中停留更长时间，那么肯定需要使用软件测试。

## 7.2软件测试的方法

### 7.2.1单元测试

在小型系统中单元测试是最常用的。 单元测试是软件开发过程中进行的最低级别的测试活动。在单元测试活动中，将软件中的独立单元与程序的其他部分分离进行测试。 单元测试不仅可以作为一次性开发过程中的错误编码的辅助方法使用。在进行软件修改或移植到新的操作环境时，单元测试必须是能够重新进行的。因此，所有测试必须在软件系统的整个生命周期中保持不变。

### 7.2.2系统功能测试

黑盒测试和白盒测试是系统测试中两种常见的方法。黑盒测试也称为功能测试，它将软件看作一个黑匣子，测试软件时不考虑程序的内部结构和特点，只考虑外部特征。根据软件需求规范测试用例，测试程序的输入输出特性是否满足所设置的功能。白盒测试又称结构测试，它将软件看作透明的白盒，根据程序的内部结构和处理逻辑选择测试用例，测试软件的逻辑路径和过程，并检查测试是否与测试匹配。

①链接测试。

链接是Web应用程序系统的主要功能，也是切换页面和引导用户到未知地址的主要手段。链路测试分为三个领域，首先，我们测试所有链接是否按照指示访问链接的页面，然后测试链接页面是否存在，最后检查Web应用程序系统上没有孤立页面。孤立页面是指没有链接到该页面，只需知道正确URL的地址即可访问。

②表单测试。

当用户向Web应用系统管理员提交信息时，需要使用用户注册，登录，信息提交等表单操作。在这种情况下，您需要测试发送操作的完整性并验证发送到服务器的信息的准确性。如果只能使用表单中指定的一部分值，也可以运行测试。例如，您只能接受某些字母，并通过跳过系统是否不正确来进行测试。

③数据库测试。

数据库在一般web应用中起着重要的作用，数据库管理Web应用系统的大部分数据操作。在Web应用程序中，最常用的数据库类型是关系数据库，可以使用SQL处理信息。在使用数据库的Web应用系统中，通常会出现数据不一致性错误和输出错误的两个错误。数据不一致性主要是由于用户所发送的信息不正确，但输出错误主要是由于网络速度和代码相关的问题，这两种情况应当分开测试。

### 7.2.3自动测试与人工测试

人工测试（Manual Testing）：借鉴了黑盒白盒测试，测试人员根据自己的能力和经验来检测程序。

自动测试（Automation Testing）：将人工测试步骤用机器来替代。人工测试相对比较普遍。自检很笨拙，没有很好的灵活性。但是手工测量有很多过程，工作量比较大。由于测试环境操作系统的网络环境带宽等可能会影响需要长期积累经验的测试结果来判断并尝试保护测试环境。

# 8 结束语

经过一个学期的辛勤与忙碌，本次毕业设计已经接近尾声，由于实践经验的匮乏，在设计过程中难免有一些困扰和难题，许多问题也会考虑不周，如果没有石老师的悉心指导与督促，以及设计小组的同学们的帮助与支持，也许毕业设计不会这么顺利的完成。

毕业设计过程中，石磊老师耐心的指导使我受益匪浅，他扎实的理论知识与丰富的实践经验给我留下了深刻的印象，也正是老师认真负责的教学态度使我从毕业设计中真正的学会了学以致用，将课本知识与实际工程中所遇到的问题相联系，整合了自己在校所学的专业知识，同时提高了我们发现问题、解决问题的能力。从系统设计开始到设计完成，石老师对于专业知识严谨的学习精神以及对于设计过程严格的要求态度成为我们在今后学习以及工作中良好的榜样，在此谨向为我们毕业设计小组辛苦了一学期的老师致以最真诚的谢意。

回顾这一学期，似乎忙碌与紧张感充满了整个毕业设计的实现过程，但是因为有了老师和同学们的关怀与帮助，使得设计的过程中也充满了欢乐。我们共同研究，相互讨论，共同学习，遇到问题一起探讨解决，毕业设计不仅是对我们学习的检验，同时也是锻炼我们与他人沟通与合作能力的好机会。

在最后我要感谢学校、学院对我们的培养，为我们的学习以及毕业设计创造良好环境、提供便利条件。伴随着毕业设计的结束，我们也将离开校园，有过少耕耘就有多少收获，相信我们走出校园后能将自己的价值体现出来，回报父母、回报学校，为社会做自己力所能及的贡献。再一次感谢在毕业设计中给予我帮助的老师和同学们，祝大家在今后的工作和生活中一切顺利！

# 参考文献

[1] 郑阿奇．JSP程序设计教程．北京：机械工业出版社，2005：174-179，215.

[2] 施伯乐，丁宝康，汪卫．数据库系统教程［Ｍ］．北京：高等教育出版社，2003：147-151.

[3] 周新会，傅立宏．ASP通用模块及典型系统开发实例导航［Ｍ］．北京：人民邮电出版社，2006：11-19