|  |
| --- |
|  |
| 高性能内容发布网站系统探究 |
| 中大毕业设计 |
|  |
| **ming123jew** |
| **2018-2-4** |

|  |
| --- |
|  |

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 说明 | 作者 |
| 2018-02-05 | 1.00 | 提纲 | ming123jew |
| 2018-03-31 | 1.01 | 论文初稿（少部分未完成） | ming123jew |
| 2018-04-03 | 1.02 |  |  |

目录

[1. 引言 6](#_Toc512501377)

[1.1. 研究背景 6](#_Toc512501378)

[1.2. 研究目的 6](#_Toc512501379)

[2. 系统分析 7](#_Toc512501380)

[2.1. 可行性分析 7](#_Toc512501381)

[2.1.1. 经济方面 7](#_Toc512501382)

[2.1.2. 技术方面 7](#_Toc512501383)

[2.1.3. 管理方面 7](#_Toc512501384)

[2.2. 系统功能需求 7](#_Toc512501385)

[2.2.1. 访客 8](#_Toc512501386)

[2.2.2. 注册用户 8](#_Toc512501387)

[2.2.3. 管理员 8](#_Toc512501388)

[2.3. 数据流图 8](#_Toc512501389)

[3. 数据库设计 9](#_Toc512501390)

[3.1. 数据库概念结构设计 9](#_Toc512501391)

[3.2. 数据库逻辑结构设计 10](#_Toc512501392)

[3.3. 数据字典 11](#_Toc512501393)

[3.4. 数据库的完整性和安全性 11](#_Toc512501394)

[4. 系统设计 12](#_Toc512501395)

[4.1. 系统B/S结构 12](#_Toc512501396)

[4.2. MVC架构 13](#_Toc512501397)

[4.3. 系统总体结构设计 13](#_Toc512501398)

[4.4. 前台功能实现 14](#_Toc512501399)

[4.4.1. 首页 14](#_Toc512501400)

[4.4.2. 前台登录 18](#_Toc512501401)

[4.4.3. 前台注册 22](#_Toc512501402)

[4.4.4. 系统栏目 22](#_Toc512501403)

[4.4.5. 系统文章 25](#_Toc512501404)

[4.4.6. 系统评论 28](#_Toc512501405)

[4.5. 后台功能 31](#_Toc512501406)

[4.5.1. 登录 31](#_Toc512501407)

[4.5.2. 角色权限管理 32](#_Toc512501408)

[4.5.3. 文章管理 33](#_Toc512501409)

[4.5.4. 用户管理 33](#_Toc512501410)

[4.5.5. 栏目管理 33](#_Toc512501411)

[4.5.6. 菜单管理 33](#_Toc512501412)

[4.5.7. 评论管理 33](#_Toc512501413)

[5. 技术框架、运行环境与开发工具 33](#_Toc512501414)

[5.1. 技术框架 33](#_Toc512501415)

[5.1.1. HTML前端框架之Bootstrap 33](#_Toc512501416)

[5.1.2. PHP服务框架之swoole 34](#_Toc512501417)

[5.1.3. SwooleDistributed框架 34](#_Toc512501418)

[5.2. 运行环境 36](#_Toc512501419)

[5.2.1. centos7.4服务器 36](#_Toc512501420)

[5.2.2. 超文本预处理器之PHP7 36](#_Toc512501421)

[5.2.3. git版本管理工具 38](#_Toc512501422)

[5.2.4. 简约客户端hiredis 38](#_Toc512501423)

[5.2.5. mysql服务器 39](#_Toc512501424)

[5.2.6. 测试运行环境 41](#_Toc512501425)

[6. 系统安全 42](#_Toc512501426)

[6.1. centos服务器安全 42](#_Toc512501427)

[6.2. 服务进程安全 42](#_Toc512501428)

[6.3. 程序模块安全 42](#_Toc512501429)

[6.3.1. 防止内存溢出 42](#_Toc512501430)

[6.3.2. 参数过滤 42](#_Toc512501431)

[6.3.3. 数据库安全 43](#_Toc512501432)

[7. 系统测试 43](#_Toc512501433)

[8. 总结 43](#_Toc512501434)

高性能内容发布网站系统探究

[摘要] PHP（外文名:PHP: Hypertext Preprocessor，中文名：“超文本预处理器”）是一种通用开源脚本语言。语法吸收了C语言、Java和Perl的特点，利于学习，使用广泛，主要适用于Web开发领域。[1]本项目就PHP编程语言之扩展框架（SwooleDistributed[2]，下文简称SD）进行高性能内容发布网站系统（偏重个人博客）探究，研究SD框架的性能表现，以及开发流程以及使用各个细节，最终完成一套属于本人甚至企业使用的高性能内容管理系统。

系统主要提供管理员/编辑发表文章、用户浏览文章等功能，具有CMS（Content Management System）通用属性，即同一类型的信息进行分类。每一类信息都可以进行新建（Add）、查看（View）、编辑（Edit）和删除（Delete）四种不同的操作。除了使用分类的方式组织信息以外，又采用了统一的用户和权限管理对信息的使用进行控制。即构成了一个完整的信息组织和管理的体系。

[关键词] PHP、SD、高性能、内容发布 、并发、功能模块

1. 引言
   1. 研究背景

21世纪以来，由于计算机技术和网络技术迅猛发展，世界已经进入了信息社会。随着个人上网用户和企事业单位用户上网的增加，并且近年来电子商务的流行，使得人们越来越依赖网络。越来越多的企业和个体成员投入到互联网行列中，如企业官方网站、个人博客、社交平台、电子商务等平台。多种多样性质的网站推动着开发网站技术的发展，由初期多数基于asp编程技术建设的网站，逐步转向PHP、.NET、JSP等语言平台编程方向，随着互联网的不断发展，对大数据的需求，特别部分社交平台对访问并发技术的需求量也是越来越大，逐渐服务架构+编程语言的并发能力越来越得到重视。本设计就PHP编程语言之扩展框架（SwooleDistributed）进行高性能内容发布网站系统（偏重博客）探究。

* 1. 研究目的

现在各行各业都需要利用网站作为宣传的媒体，因此网站的整体需求是很大的，网站内容管理系统的前景也是很广阔，特别高性能内容管理系统，无论在并发访问量或降低企业硬件成本，或者技术需求上更是受到媒体以及互联网的青睐。高性能内容管理系统成为我国各行各业、各领域的基础设施建设重点。在这种大趋势下，无论是在理论上还是在应用上，对于一个标准的完整的高性能内容管理系统的研究与应用都是具有重要意义的。基于此，本项目研究的目的就PHP编程语言之扩展框架（SwooleDistributed）完成对网站内容管理系统中的角色管理、栏目管理、内容发布等核心功能，实现一个属于自己/企业的高性能内容管理系统。

1. 系统分析

本章将详细介绍系统的分析过程，包括系统的可行性分析，各类用户的功能以及系统流程。

* 1. 可行性分析

可行性分析(Feasibility Analysis)也称为可行性研究，是在系统调查的基础上，针对新系统的开发是否具备必要性和可能性，对新系统的开发从技术、经济、社会的方面进行分析和研究，以避免投资失误，保证新系统的开发成功。可行性研究的目的就是用最小的代价在尽可能短的时间内确定问题是否能够解决。

* + 1. 经济方面

本系统对系统开发者来说并不需要太高的成本支出，只是对系统的管理者付出管理报酬即可，而且开发周期不需要太长，节省了人力、物力、财力资源，所以本系统在经济上是可行的。

* + 1. 技术方面

在软件方面，本系统采用编程语言PHP(PHP Hypertext Preprocessor)进行开发，前端网页框架使用的是Bootstrap框架，而数据库系统采用的是MYSQL，相关技术都是目前较为流行且开源的技术。通过多方面分析，在软、硬件方面现有工具与环境完全可以实现系统的开发，因此具有技术上的可行性。目前越来越多的应用程序都已经是转向基于Web的开发，并且Internet已经广泛使用，因此系统的设计具有时机可行性。

* + 1. 管理方面

系统的开发主要就是为了方便个人信息的管理，补充现在传统管理方式的不足，因此具备了管理上的可行性。

综上所述以及结合当前背景情况，本系统开发目标已明确，在技术和经济等方面具备可行性，并且投入少、见效快，因此系统的开发是完全可行的。

* 1. 系统功能需求

系统涉及到的用户角色分为三种：访客、注册用户和管理员。下面分别介绍各类用户角色的主要功能。

* + 1. 访客

访客无须注册即可浏览页面的文章，并且还可以对自己感兴趣的文章发表评论，给文章留言。

* + 1. 注册用户

系统同时提供简单的用户注册功能，用户可通过注册页面进行注册用户，目前的注册用户与访客类似，仅提供阅读文章，以及发表评论功能。未来该系统可以迭代注册用户功能，如用户上线积分、评论积分、登录积分等。

* + 1. 管理员

管理员是该系统的核心成员，通过后台登录的超级管理员，拥有整个系统的所有权限，可分配角色、设置权限、管理用户、管理栏目、管理文章等。

* 1. 数据流图

系统运行过程中，主要涉及到两部分的数据流向，一部分是管理员在系统后台管理界面上发布的对数据库操作的数据流，另一部分是用户在应用前台使用系统时所产生的数据流。如图2.3-1和图2.3-2：

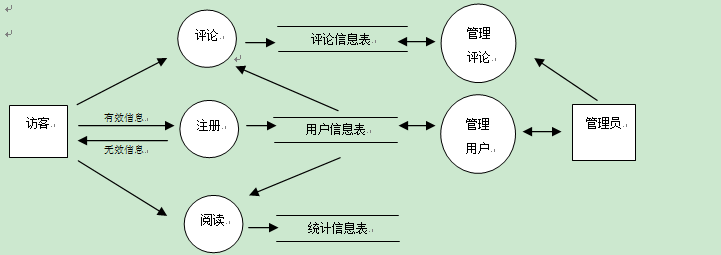


图2.3-1前台用户数据流图

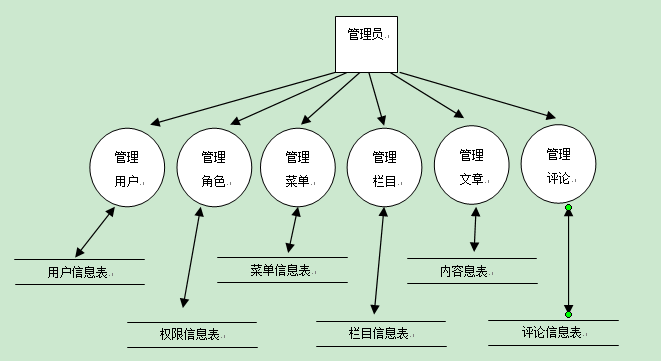


图2.3-2后台用户数据流图

1. 数据库设计

数据库技术概述(Database)，是按照数据结构来组织、存储和管理数据的仓库，是用于查询的大量数据的存储区域。使用数据库可以带来许多好处：如减少了数据的冗余度，从而大大地节省了数据的存储空间，实现数据资源的充分共享等等。近年来推出的计算机关系数据库管理系统，还具有操作直观，使用灵活，编程方便，功能强大，环境适应广泛，数据处理能力极强等诸多优点。

总的来说，数据库的设计对系统的经济性、功能性和效率有很大的影响，一个好的数据库，要求在设计时尽量避免数据的多余，另外，还要尽可能提高数据的存取速度。数据库设计时速度与空间在范式上是相互矛盾的。一方面按规范化理论的要求，关系模式应尽量取高级范式，尤其对记录数较多的关系，低级范式将会造成存储的大量重复，是空间遭受严重浪费；另一方面从存取速度上考虑，应该尽量做到一个模式涉及的属性越多越好，相互运算越少越好，这样，又应取低级范式。因此我们在设计数据库时要严格地探讨它的使用环境，充分了解用户的需求，清楚地掌握数据库系统的特点。经过综合的评价和分析后设计出的数据库应满足以下要求：

（1）能够支持用户对数据进行的所有处理；

（2）能够容易被数据库管理系统运行维护；

（3）数据一致性、完整性好，无更新异常。

* 1. 数据库概念结构设计

概念结构设计就是将需求分析得到的用户需求抽象为信息结构，即概念模型。再这里采用E-R模型。概念结构的E-R模型设计如图3.1-1：

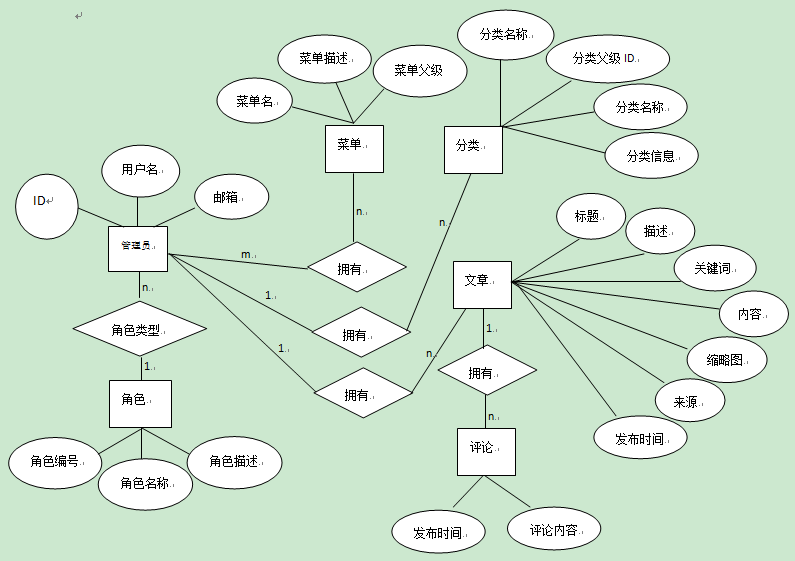


图3.1-1 系统后台E-R模型图

* 1. 数据库逻辑结构设计

概念结构设计阶段得到的E-R模型是用户的模型，它独立于任何一种数据模型，独立于任何一种具体的DBMS。为了建立用户所需求的数据库，需要把上述概念模型转为某个具体的DBMS所支持的数据模型。数据逻辑结构设计的任务就是将概念结构转换为特定DBMS的所支持的数据模型的过程。该系统数据库的逻辑结构设计如图3.2-1：

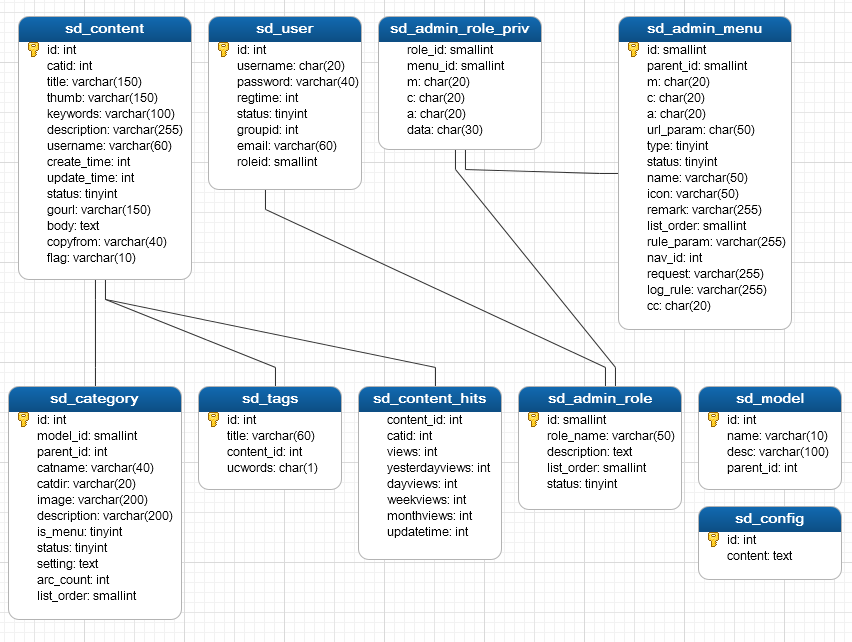


图3.2-1 数据库逻辑关系图

* 1. 数据字典

数据字典较多，请参阅附件。

* 1. 数据库的完整性和安全性

数据库的完整性是指数据的正确性和相容性。数据库管理系统(DBMS)用一定的机 制来检查数据库中的数据是否满足规定的条件——完整性约束条件，数据的约束条件是语义的体现，将作为模式的一部分存入数据库中。

数据库的完整性约束主要包括以下三方面要求：

(1)实体完整性 实体完整性是指若属性A是基本关系R的主属性，则属性A不能取空值。对于每个表来说，主键都不可以为空。在用户提交信息时，系统会对用户所填写的信息进行验证，对于其中的关键信息没有填写或者数据非法时，系统会提示用户更改后再提交。这样保证了系统实体完整性。

(2)参照完整性

由于表和表之间会存在某些联系，当关系R和关系S之间是一对多关系时(R的主键是A，S的主键是b，c是s中的外码)，在插入数据时就要保证外码c的值或者为空或者等于R中某个元组的主码值。在对被参照表进行删除操作时应同时进行及联删除。

(3)用户定义完整性

任何关系数据库系统都应该支持实体完整性和参照完整性。除此之外，不同的关系数据库系统根据其应用环境的不同，往往还需要一些特殊的约束条件，用户定义的完整性就是针对某一具体关系数据库的约束条件。它反映某一具体应用所涉及的数据必须满足的语义要求。例如某个属性必须取唯一值、某个属性的取值范围等。

1. 系统设计

对系统进行系统分析、数据库设计之后，本章将主要介绍系统各功能模块的设计及其实现。 软件所实现的功能强弱是衡量一个软件的最根本的标准。通过对系统的全面分析并结合调研的情况，确定了本系统的功能模块图。本章先介绍B/S结构，MVC架构、然后对系统总体结构设计介绍，接着分别对前台、后台功能模块加以描述。

* 1. 系统B/S结构

B/S结构分为三层结构，建立在广域网的基础之上，客户端只需要安装浏览器，通过HTTP协议向WebServer提出请求，Web Server站点收到请求后，通过CGI（ISAPI）得到相关数据，然后以HTML文档的形式返回给浏览器。结构如图4.1-1：

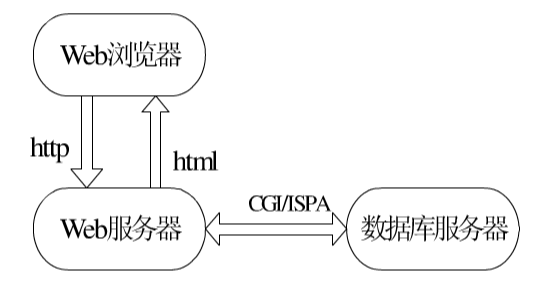


图4.1-1 B/S三层结构图

B/S建立在广域网之上，不必是专门的网络硬件环境，有比C/S更强的适应范围，一般只要有操作系统和浏览器即可。在B/S结构的系统中，用户可以通过浏览器向分布在网络上的许多服务器发出请求。B/S结构极大地简化了客户机的工作，客户机上只需要安装、配置较少量的客户端软件即可，服务器将担任更多的工作，对数据库的访问和应用程序的执行将在服务器上完成。B/S体系的优点是：系统安装维护简单、数据集中管理；便于分散用户使用，适应互联网时代软件的发展趋势。根据B/S结构的特点和本系统的性能特点，采用B/S结构进行开发更合适。

* 1. MVC架构

MVC全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

其中这里需要特别说明，系统中一部分较为复杂的数据表（如内容表sd\_conteng），其Model层又细分为数据逻辑层与数据访问层，以简洁控制器代码编写，同时更好的实现“高内聚低耦合”的思想。

业务逻辑层：主要是针对具体的问题的操作，也可以理解成对数据层的操作，对数据业务逻辑处理，如果说数据层是积木，那逻辑层就是对这些积木的搭建。

数据访问层：主要是对非原始数据（数据库或者文本文件等存放数据的形式）的操作层，而不是指原始数据，也就是说，是对数据库的操作，而不是数据，具体为业务逻辑层或表示层提供数据服务。

* 1. 系统总体结构设计

根据系统的功能分析画出系统的功能结构图，分别从前台、后台界面对功能模块加以描述。前台界面的系统功能如图4.3-1：

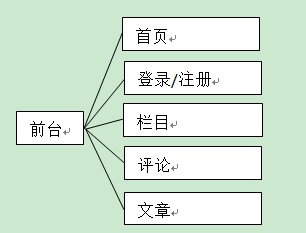


图4.3-1

后台界面系统功能如图4.3-2：

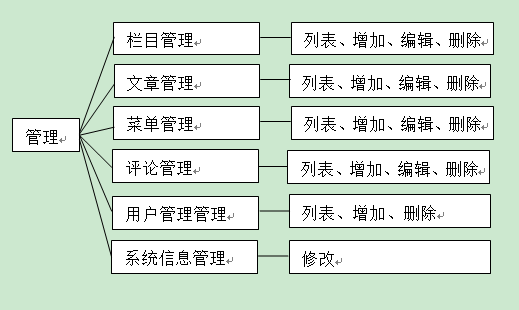


图4.3-2

* 1. 前台功能实现

本次毕业设计以后台管理为主，前台以访客/用户阅读文章、评论为主。因此，根据此特性，系统将前台读取文章相关信息进行了逻辑封装，其封装文件为：src\app\Models\Business\HomeBusiness.php。通过调用封装函数，只需要根据封装函数的传参设置，传入相关参数，即可获得对应的文章列表/内容。下面各小节进行详细介绍。

* + 1. 首页
       1. 页面设计

根据系统需求，以及页面美观，其设计布局图如图4.4-1：



图4.4-1

* + - 1. 导航

由于此设计偏向于博客风格，无论是页面还是功能上都追求简洁、高性能，因此导航方面不进行读取后台管理（栏目预留是否显示按钮，暂不使用），而是直接写入模板公共文件src\app\Views\Home\nav.php，以便提供性能，且方便其他页面调用。同时，为了提供系统性能，快读读取列表/内容，在逻辑层都进行了缓存设置开关（默认是开启，且默认缓存时间为24\*3600秒，后期可在【系统设置】、【基本设置】、【内容相关】进行设置缓存时间、是否开启缓存）。

* + - 1. 幻灯列表

幻灯列表属于文章推荐的一种，在后台编辑文章/添加文章时选择其中一个标签，则会自动归类到相关列表，如下图4.4-2：

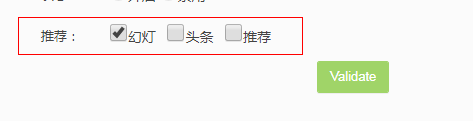


图4.4-2

文章推荐选择了幻灯，则其文章将会显示到首页幻灯列表处，且按最新时间排序。

数据层代码如下图4.4-3：

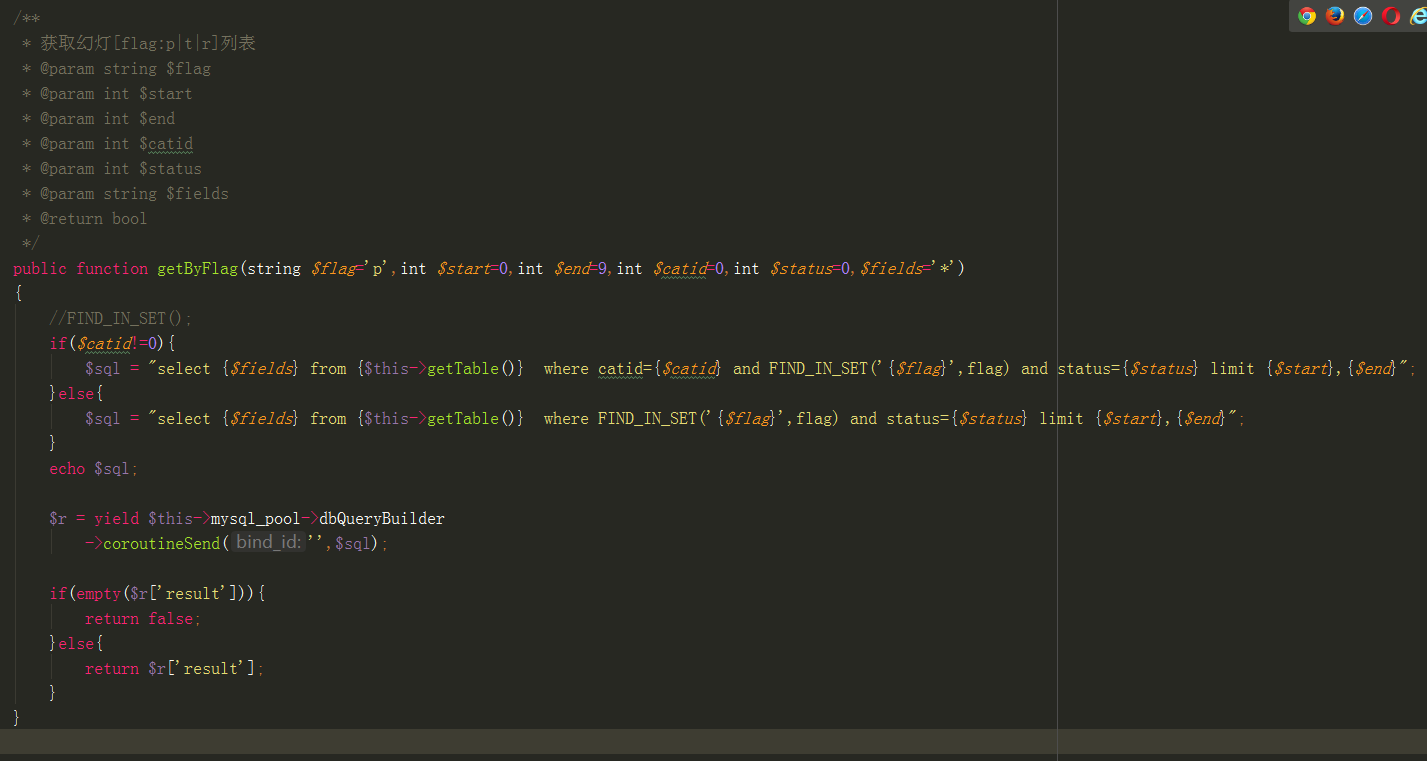


图4.4-3

逻辑代码如下图4.4-4：

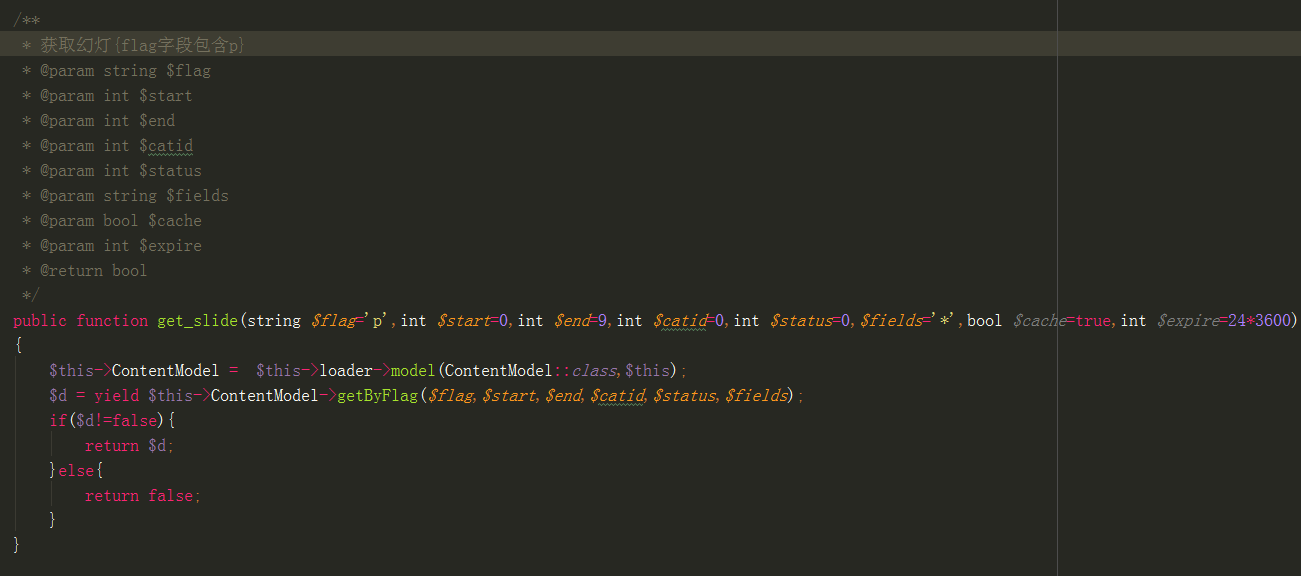


图4.4-4

控制器中调用代码如下图4.4-5：

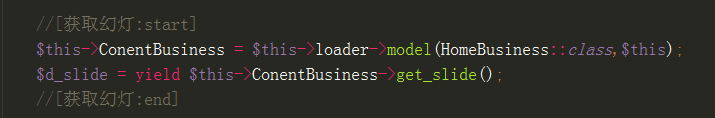


图4.4-5

系统通过MVC架构方式，使其代码结构清晰，管理便捷，实现了数据与页面分离。同时，对部分模型（M层）进行分解（数据层与逻辑层），使系统实现思路更加清晰、方便维护。

* + - 1. 热门推荐

实现方式与4.4.1.3节类似，通过在后台编辑文章/添加文章时选择其中一个标签，则会自动归类到相关列表，在这不再详细描述。

* + - 1. 最新文章

最新文章，顾名思义是系统中最新的文章列表，即按文章发布时间倒序查询的列表，同样的，系统对此列表进行了函数封装。实现与调用方式如下：

数据层代码如下图4.4-6：



图4.4-6

逻辑代码如下图4.4-7：



图4.4-7

控制器中调用代码如下图4.4-8：

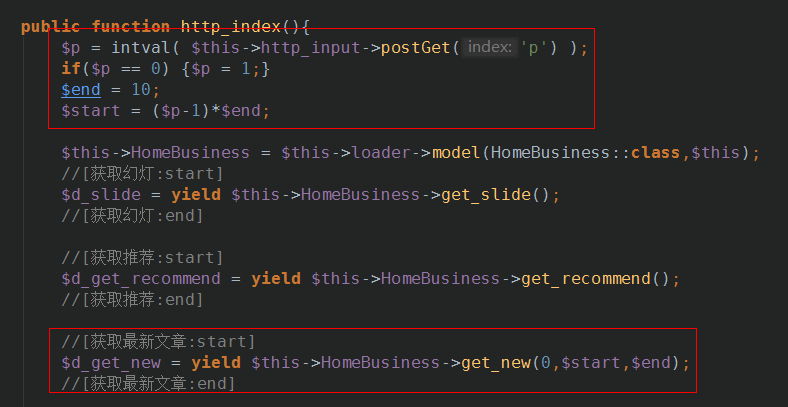


图4.4-8

* + - 1. 每日一句

每日一句展示当天的日期，然后从每日一句数据库中获取一条语录，简单而又有韵味的语录主要增加网站的整体美观以及提高品质。

* + - 1. 最新评论

最新评论是从系统中获取游客/用户的评论信息，每个页面右侧下方都有显示，如对评论信息感兴趣，可通过点击评论进入文章，查阅相关文章。

* + 1. 前台登录
       1. 登陆流程图

系统前台登陆流程图如图4.4-9

帐号+密码

验证

首页

y

n

提示

图4.4-9

* + - 1. 验证过程

用户在未登录状态下进行后台相关页面或接口访问，后台程序会自动检测管理员是否已经登录，如未登录则跳转到登录界面，提示用户输入账号和密码进行登录。

用户输入账号和密码后，前端使用JavaScript进行简单的验证操作，如系统对账号是否为空、密码是否为空，长度等做了简单的验证，JavaScript验证通过后，提交到服务器进行验证登录。后台的登录验证功能主要分为用户登录时的身份认证与操作后台功能权限认证。用户提交登录后，程序首先会进对数据进行过滤操作，然后进行身份认证操作，所输入账号与密码能够与数据库查询到的数据相匹配，则认证成功，否则提示相关登录失败信息。如登录成功，程序会把认证成功的用户信息保存到服务器session，然后进入主页。由于时间有限，暂不进行个人主页的开发。

* + - 1. 程序实现模式（AOP）

在软件业，AOP为Aspect Oriented Programming的缩写，意为：面向切面编程，通过预编译方式和运行期动态代理实现程序功能的统一维护的一种技术。AOP是OOP的延续，是软件开发中的一个热点，也是Spring框架中的一个重要内容，是函数式编程的一种衍生范型。利用AOP可以对业务逻辑的各个部分进行隔离，从而使得业务逻辑各部分之间的耦合度降低，提高程序的可重用性，同时提高了开发的效率。

系统前台登录功能使用了AOP的编程模式进行实现，在探究AOP思想编程的同时降低业务逻辑的耦合度，基于AOP模式很好的预留了验证、积分模块。

* + - 1. 代码实现

登录扩展接口如图：

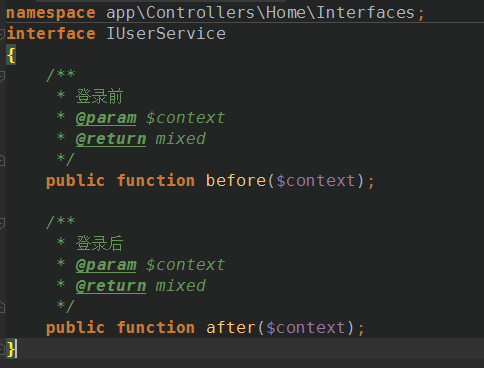


图4.4-10

接口实现类如图4.4-11：



图4.4-11

控制器实现如图4.4-12:

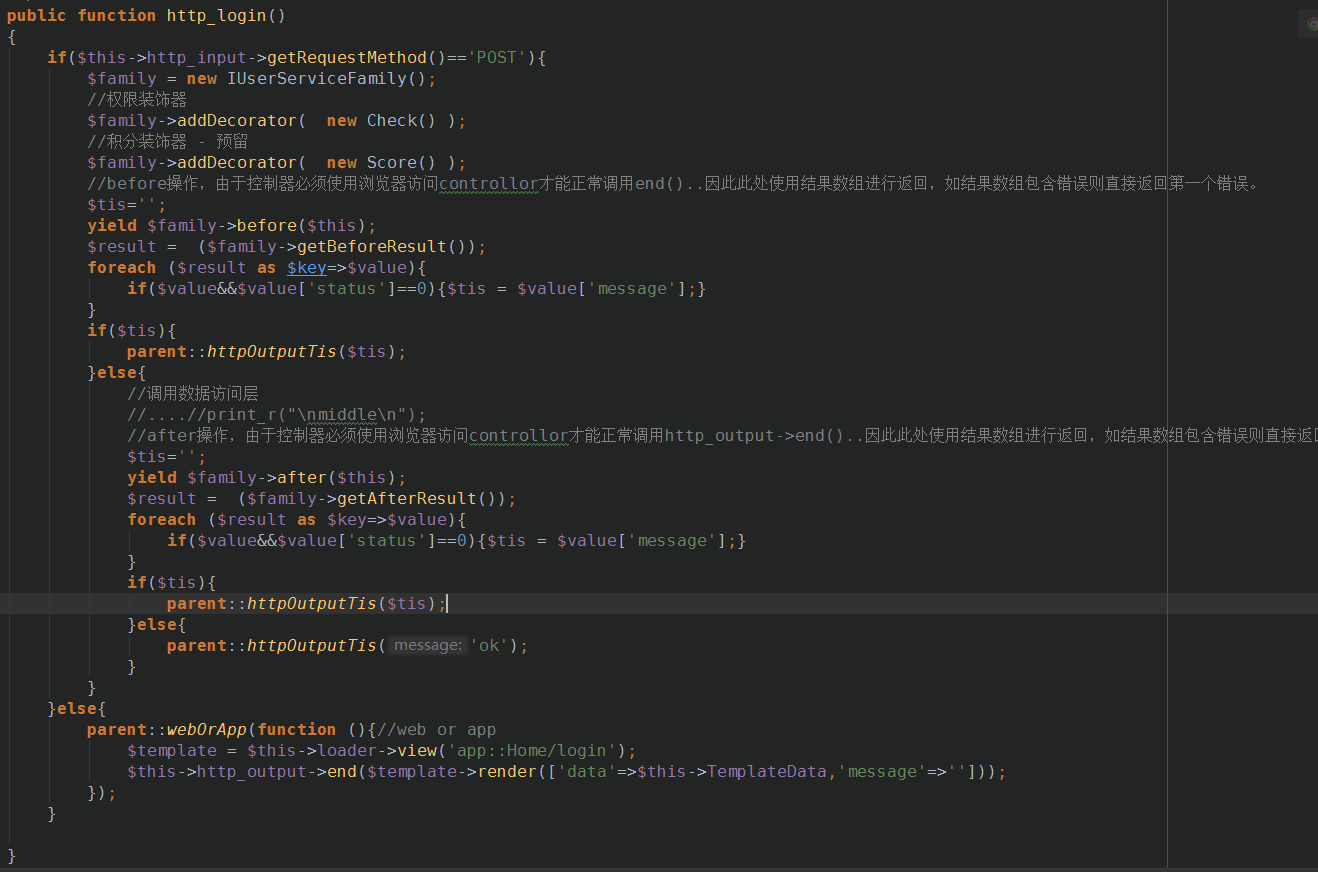


图4.4-12

登陆前置操作如图4.4-13:

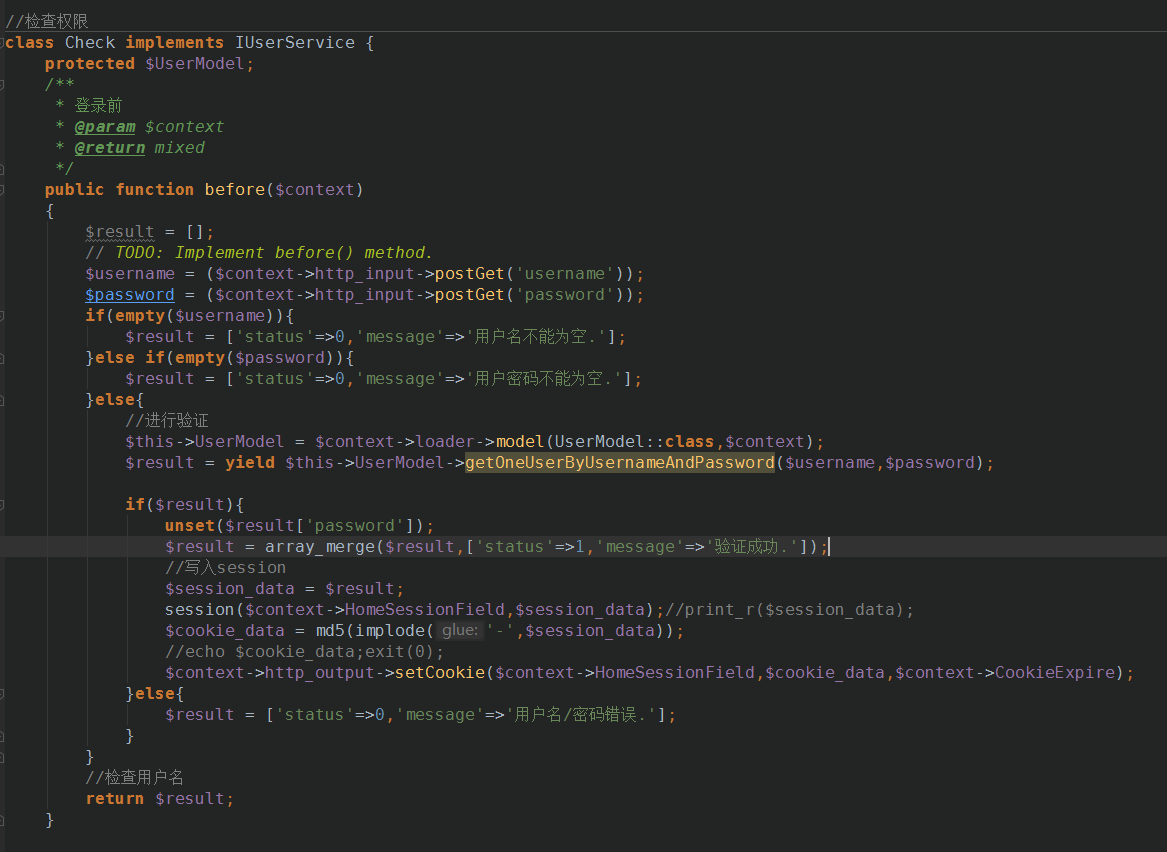


图4.4-13

登陆后置操作(预留,暂不实现)如图4.4-14:



图4.4-14

* + 1. 前台注册

前台注册与前台登录实现方式基本一致，在这里不再详细介绍。

* + 1. 系统栏目

系统栏目页面设计参考首页设计,保持美观的同时删减了幻灯模块,使栏目看起来更便捷、内容更集中、一目了然。因此，着重介绍栏目功能的实现，栏目左侧以及右则与首页左右则功能类似，不再作介绍。

* + - 1. 界面设计

系统前台栏目界面设计如图4.4-15:



图4.4-15

* + - 1. 功能实现

栏目实现控制器如图4.4-16

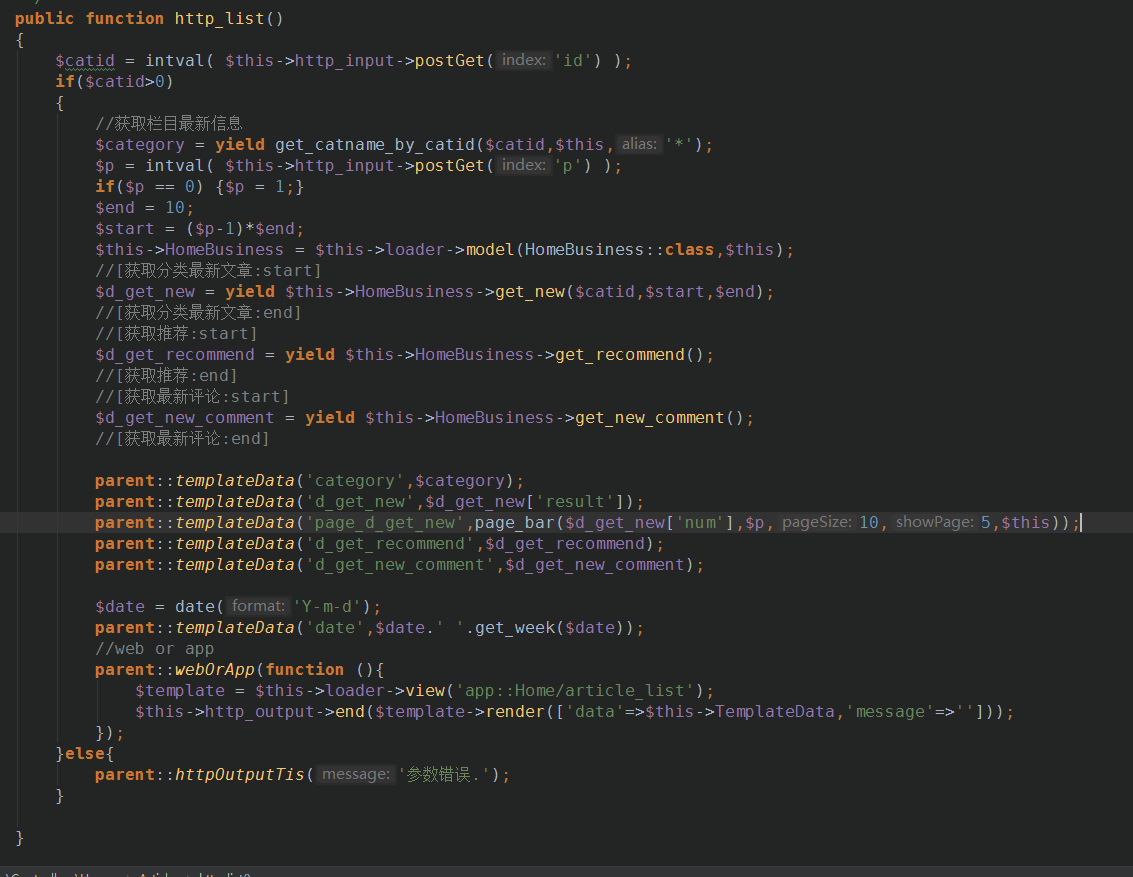


图4.4-16

栏目实现相关逻辑主要方法如图4.4-17：

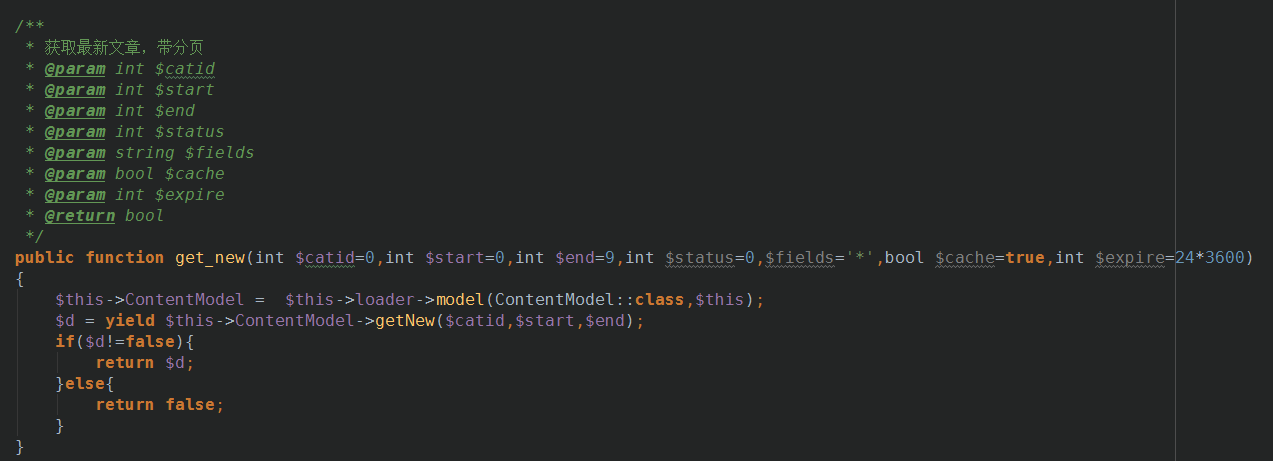


图4.4-17

栏目实现相关model主要方法如图4.4-18：

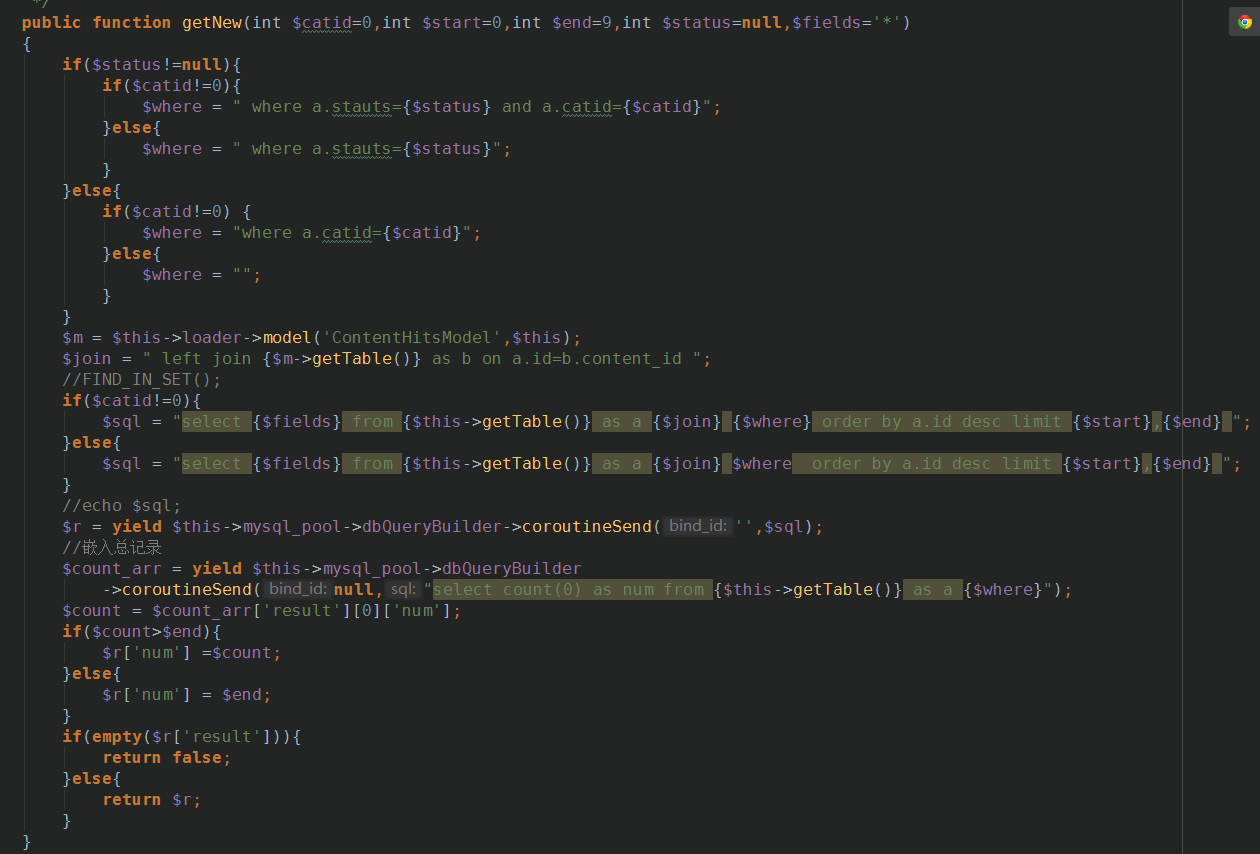


图4.4-18

栏目实现相关分页方法如图4.4-19：

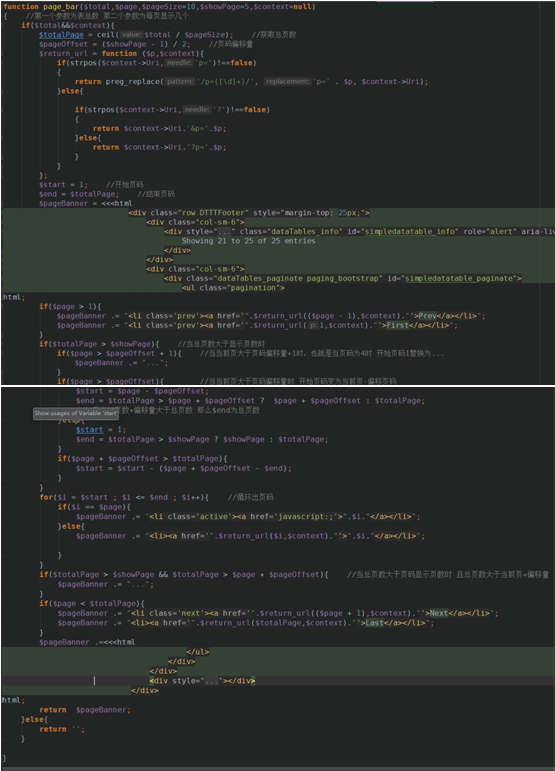


图4.4-19

* + 1. 系统文章
       1. 界面设计

系统文章界面设计如图4.4-20:



图4.4-20

* + - 1. 功能实现

系统文章读取控制器如图4.4-21

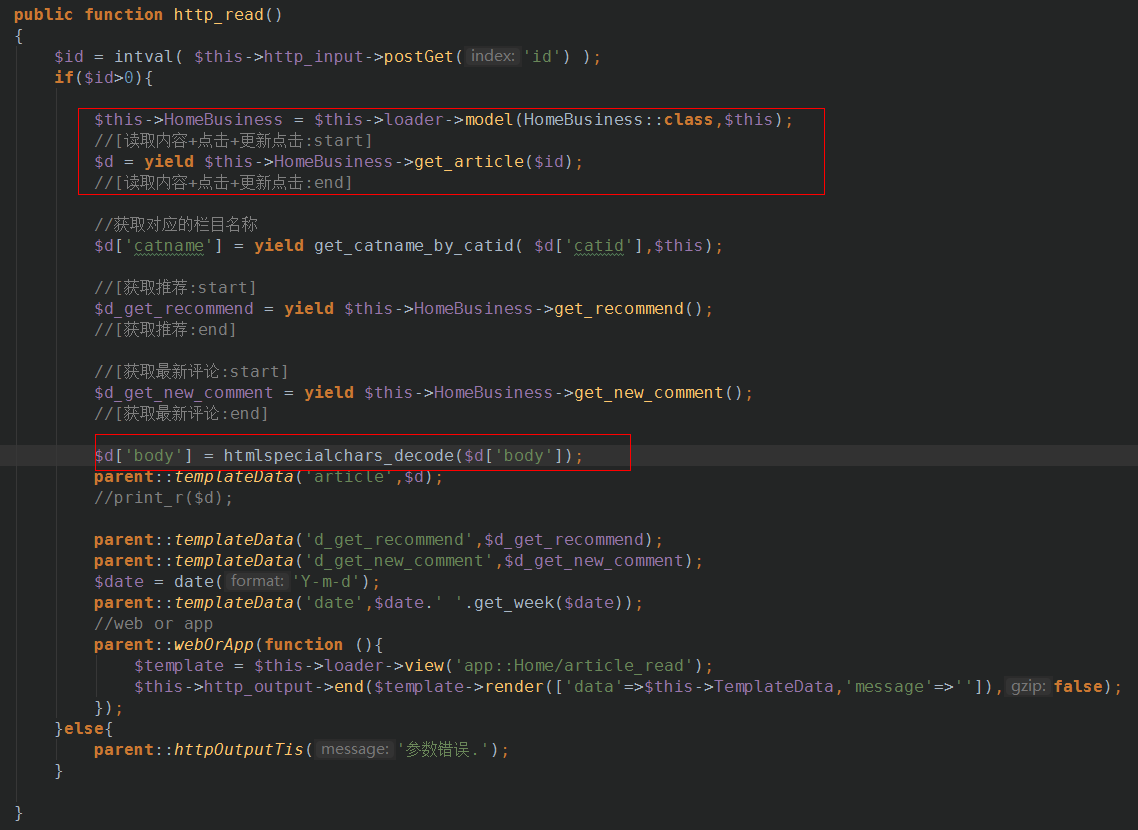


图4.4-21

系统文章相关逻辑主要方法如图4.4-22



系统文章相关数据主要方法如图4.4-23：

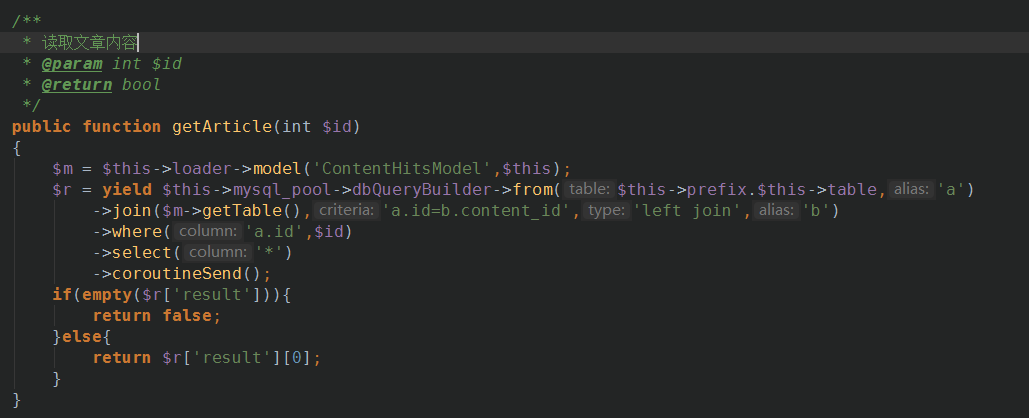


图4.4-23

系统文章获取上下篇方法如图4.4-24：

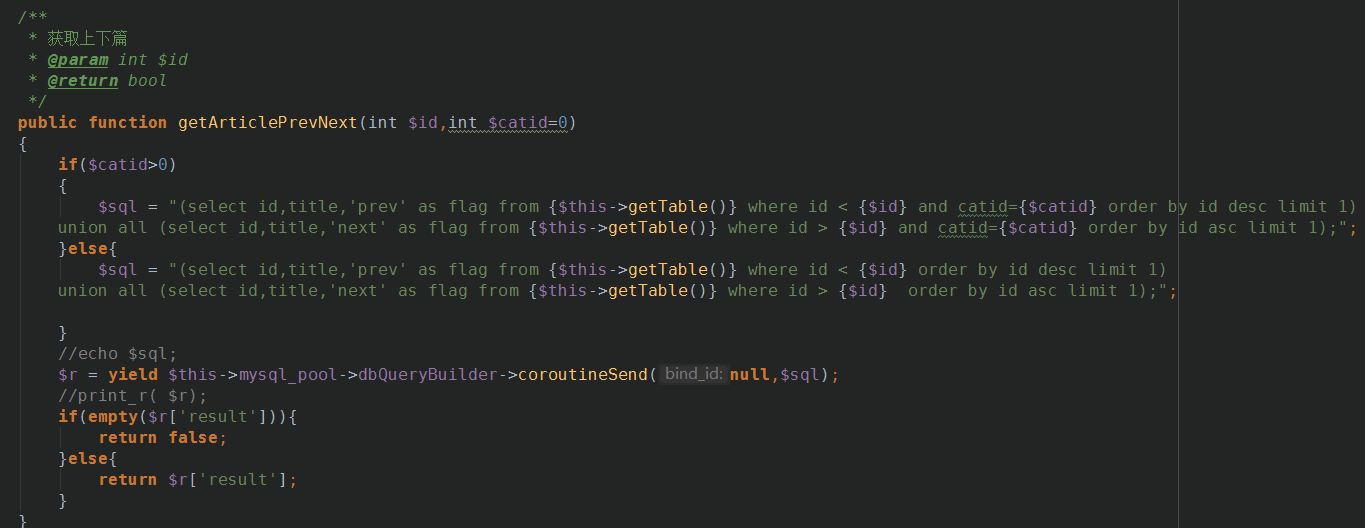


图4.4-24

* + 1. 系统评论
       1. 界面设计

系统评论界面设计如图4.4-25：

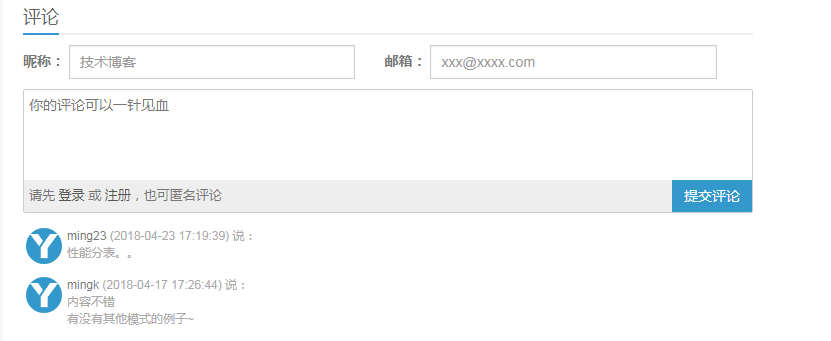


图4.4-25

* + - 1. 发表评论功能实现

系统发表评论控制器代码实现如图4.4-26：

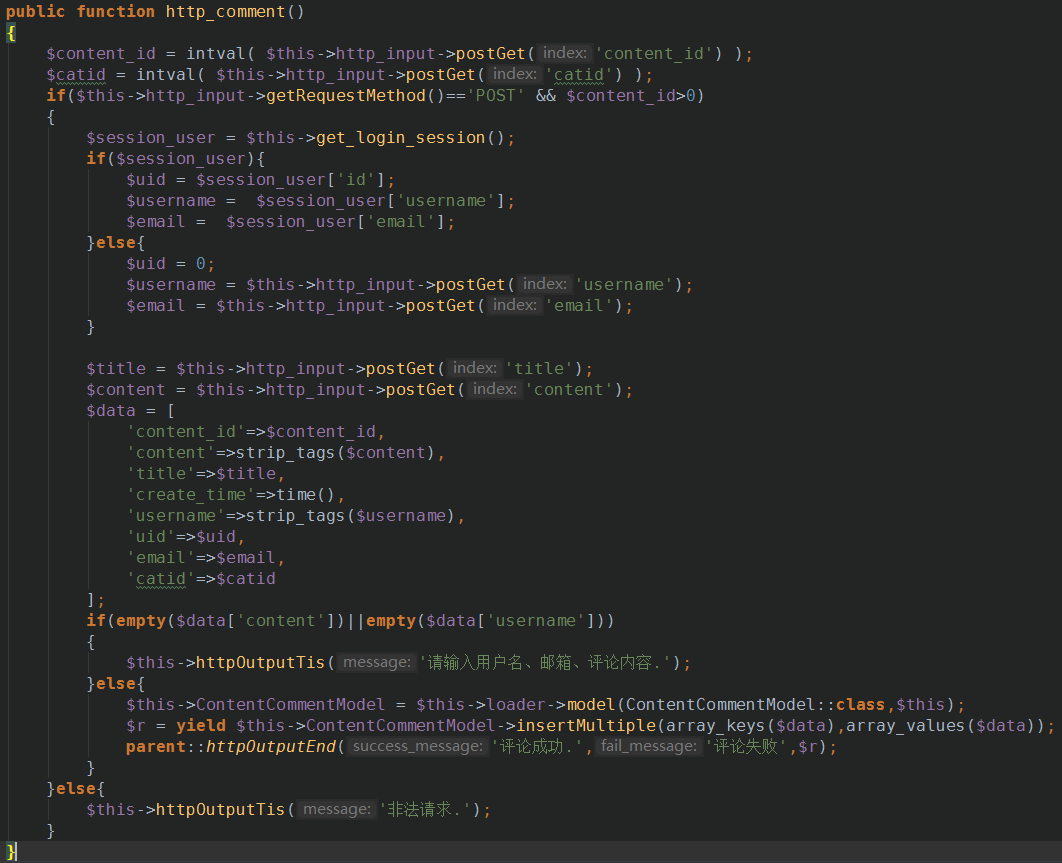


图4.4-26

系统发表评论数据层代码实现如图4.4-27：



图4.4-27

* + - 1. 获取评论功能实现

系统获取评论控制器代码实现如图4.4-28：



图4.4-28

系统获取评论数据层代码实现如图4.4-29：

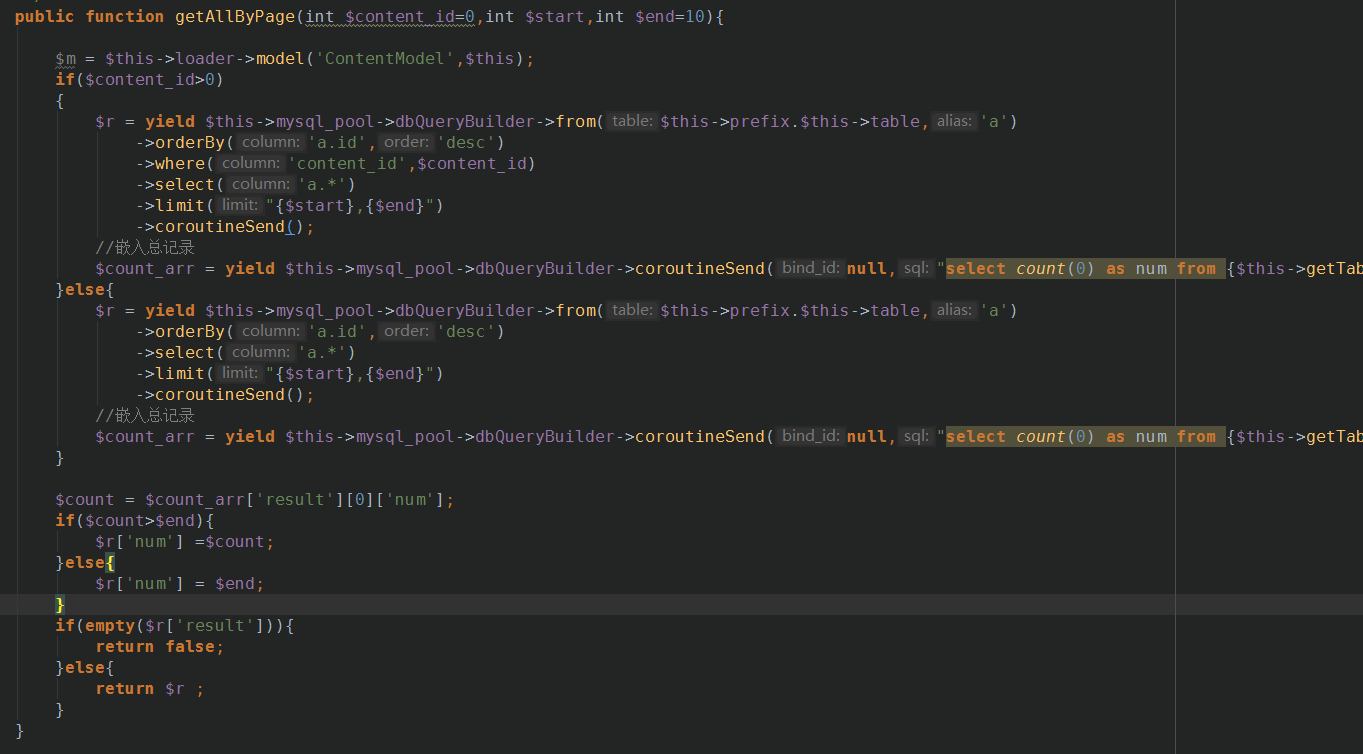


图4.4-29

* 1. 后台功能

系统后台功能如图4.5-1:

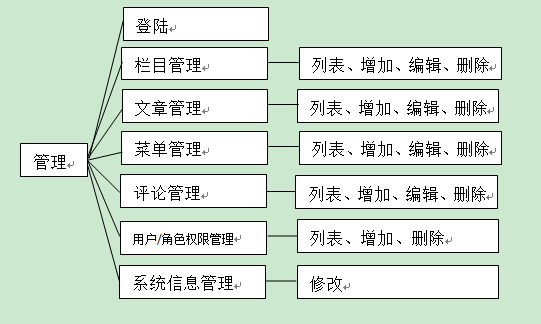


图4.5-1

由于后台系统栏目管理、菜单管理、用户管理、评论管理、文章管理中的列表、增加、编辑、删除实现方式较为类似，因此，本章重点介绍登录、角色权限、以及文章管理。

* + 1. 登录
       1. 流程图

后台登录流程如图4.5-2：

帐号+密码

验证

后台

y

n

提示

图4.5-2

* + - 1. 验证过程

用户在未登录状态下进行后台相关页面或接口访问，后台程序会自动检测管理员是否已经登录，如未登录则跳转到登录界面，提示用户输入账号和密码进行登录。后台程序开放了登录控制器的检测，因此登录界面可无条件进行访问。

用户输入账号和密码后，前端使用JavaScript进行简单的验证操作，如系统对账号是否为空、密码是否为空，长度等做了简单的验证，JavaScript验证通过后，提交到服务器进行验证登录。后台的登录验证功能主要分为用户登录时的身份认证与操作后台功能权限认证。用户提交登录后，程序首先会进对数据进行过滤操作，然后进行身份认证操作，所输入账号与密码能够与数据库查询到的数据相匹配，则认证成功，否则提示相关登录失败信息。如登录成功，程序会把认证成功的用户信息保存到服务器session，然后进行权限认证操作。

权限认证详细介绍将在角色权限进行介绍。

* + 1. 角色权限管理
       1. 流程图

登陆流程图如图4.5-3:

后台URL

模块名+控制器名+方法名

用户信息(role\_id)

所在角色权限

验证

执行

y

提示

n

图4.5-3

* + - 1. RBAC权限验证

权限认证（RBAC），基于角色的权限访问控制（Role-Based Access Control）作为传统访问控制（自主访问，强制访问）的有前景的代替受到广泛的关注。在RBAC中，权限与角色相关联，用户通过成为适当角色的成员而得到这些角色的权限。这就极大地简化了权限的管理。在一个组织中，角色是为了完成各种工作而创造，用户则依据它的责任和资格来被指派相应的角色，用户可以很容易地从一个角色被指派到另一个角色。角色可依新的需求和系统的合并而赋予新的权限，而权限也可根据需要而从某角色中回收。角色与角色的关系可以建立起来以囊括更广泛的客观情况。

RBAC支持三个著名的安全原则：最小权限原则，责任分离原则和数据抽象原则。

（1）最小权限原则之所以被RBAC所支持，是因为RBAC可以将其角色配置成其完成任务所需要的最小的权限集。

（2）责任分离原则可以通过调用相互独立互斥的角色来共同完成敏感的任务而体现，比如要求一个计帐员和财务管理员共参与同一过帐。

（3）数据抽象可以通过权限的抽象来体现，如财务操作用借款、存款等抽象权限，而不用操作系统提供的典型的读、写、执行权限。然而这些原则必须通过RBAC各部件的详细配置才能得以体现。

* + 1. 文章管理

系统中文章管理主要包括文章列表、添加文章、编辑文章、删除文章功能。

* + - 1. 页面设计

系统文章列表页面设计如图4.5-4:

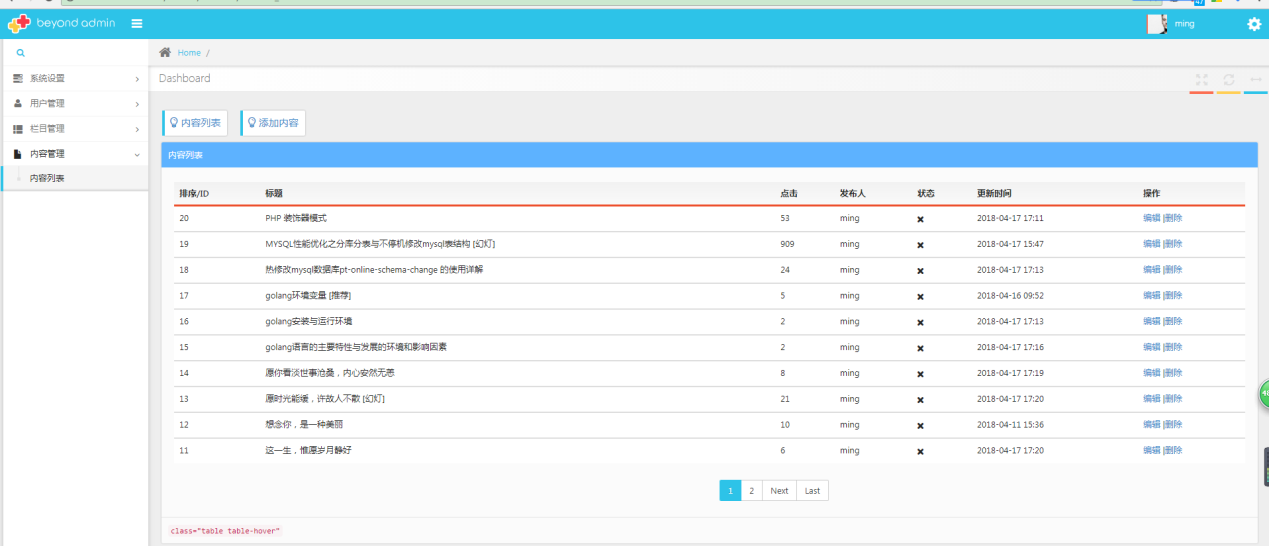


图4.5-4

系统文章添加/编辑页面设计如图4.5-5:

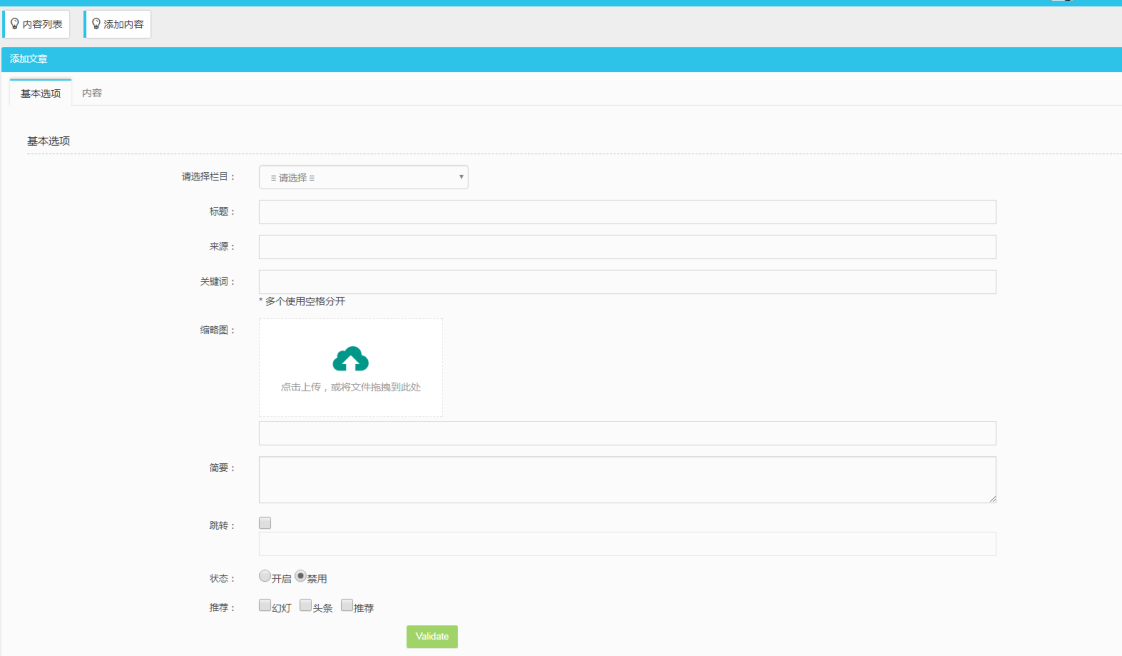


图4.5-5

* + - 1. 文章列表

后台文章列表主要显示所有文章的列表信息，包括标题、点击、发布人、更新时间、状态等。功能实现过程中，由于文章点击率以及点赞属于经常操作的数据表，因此文章相关信息与点击率设计在不同的数据表中，以提高系统整体性能。查询列表过程中，数据库使用了联合查询将数据表对应数据行以及统计表数据行联合起来返回给结果集。实现过程如图4.5-6：

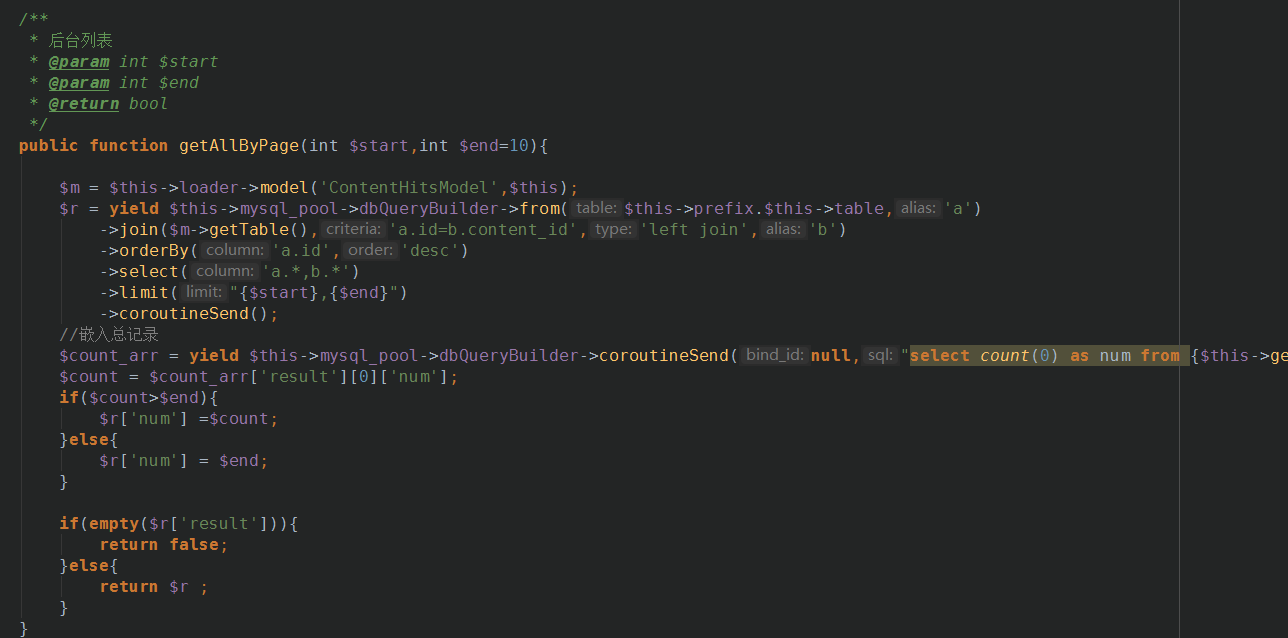


图4.5-6

* + - 1. 添加文章
      2. 编辑文章
      3. 删除文章
    1. 用户管理

系统中用户管理主要包括用户列表、添加用户、删除用户功能，编辑用户暂未实现。

* + 1. 栏目管理

系统中栏目管理主要包括栏目列表、添加栏目、编辑栏目、删除栏目功能。

* + 1. 菜单管理

系统中菜单管理主要包括菜单列表、添加菜单、编辑菜单、删除菜单功能。

* + 1. 评论管理

系统中评论管理主要包括评论列表、查看评论、删除评论功能。

1. 技术框架、运行环境与开发工具
   1. 技术框架
      1. HTML前端框架之Bootstrap
         1. Bootstrap的介绍

Bootstrap，来自 Twitter，是目前很受欢迎的前端框架。Bootstrap 是基于 HTML、CSS、JavaScript 的，它简洁灵活，使得 Web 开发更加快捷。它由Twitter的设计师Mark Otto和Jacob Thornton合作开发，是一个CSS/HTML框架。Bootstrap提供了优雅的HTML和CSS规范，它即是由动态CSS语言Less写成。Bootstrap一经推出后颇受欢迎，一直是GitHub上的热门开源项目，包括NASA的MSNBC（微软全国广播公司）的Breaking News都使用了该项目。国内一些移动开发者较为熟悉的框架，如WeX5前端开源框架等，也是基于Bootstrap源码进行性能优化而来。

* + - 1. Bootstrap Admin Template前端模板

Bootstrap Admin Template最漂亮的Bootstrap后台模板之一，支持屏幕自适应，UI丰富多彩，控制完整，包括各种表单控件、图表控制、表格控件，支持从右向左的阅读习惯，支持多级菜单，提供表单编辑器，动画效果酷炫非凡。本系统程序后台搭建用此框架进行后台制作可减免后台页面设计工作，大大提高后台页面制作效率。

* + 1. PHP服务框架之swoole
       1. swoole框架介绍

Swoole是一种PHP高级Web开发框架，框架不是为了提升网站的性能，是为了提升网站的开发效率。最少的性能损耗，换取最大的开发效率。利用Swoole框架，开发一个复杂的Web功能，可以在很短的时间内完成了。

* + - 1. swoole框架特点

(1)类似ORM的数据查询，提供SQL封装器，让MySQL的SQL与PHP的Array，会话，Cache无缝结合。

(2)MVC分层结构，有效的程序结构分层，提高程序的可维护性和扩展性，实现低耦合，基于接口开发。

(3)集成大量，实用的功能，比如方便的数据库操作，模板操作，缓存操作，系统配置，表单处理，分页，数据调用，字典操作，上传处理，内容编辑，调试等。

(4)模板-数据反射系统，可以直接在模板中调用数据，提供很多标签，可是无需修改程序，只修改模板，即可实现网站各类更新维护工作。

(5)Swoole包含了大量类，提供众多的功能扩展，基本上Web开发能够用到的功能类，大部分都可以在Swoole框架中找到。

(6)Swoole拥有插件系统，Fckeditor、Adodb、pscws中文分词、中文全文索引系统、最新的Key-Value数据库思想，TokyoTyrant，可以无限扩展框架的功能。

* + 1. SwooleDistributed框架
       1. SwooleDistributed框架介绍

SD框架全称SwooleDistributed，从名称上看一个是Swoole一个是Distributed，他是基于Swoole扩展的可以分布式部署的应用服务器框架。 借助于PHP的高效开发环境，Swoole的高性能异步网络通信引擎，以及其他的高可用的扩展和工具，SD框架提供给广大开发者一个稳定的高效的而且功能强大的应用服务器框架。

* + - 1. SwooleDistributed框架特点

(1)混合协议 SD框架支持长连接协议TCP，WebSocket，短连接协议HTTP，以及UDP。 通过配置开放不同的端口开发者可以轻松管理不同的协议，并且可以共用一套业务代码，当然你可以通过智能路由进行代码的隔离。 长连接可以配置不同的数据传输协议，比如二进制协议文本协议等等，通过框架提供的封装器解包器接口可以自定义各种各种的协议封装，并且各种协议之间可以自动转换，比如你通过广播发送一个信息，该信息流向不同客户端，客户端间采用不同协议，那么框架会根据不同的端口自动转换不同的协议封装。 你也可以通过Http给所有长连接客户端发送推送消息，类似这种混合协议协作的业务在SD框架上会异常简单。

(2)MVC以及智能路由 框架的设计是MVC架构，其中每一个层级都可以继续划分子层级，开发者可以将Controller继续分层通过不同文件夹进行管理，也可以将Model进行细分，划分为业务层和数据层，这都看开发者自身的系统设计。智能路由将处理解包器解包后的数据，负责将这些数据传递到Controller层。

(3)中间件 SD框架还提供了中间件，中间件可以对流入的数据进行处理，比如清理异常数据，修改数据，流量统计，搜集日志等功能。中间件可以设置多个，他们和端口进行绑定。

(4)对象池 SD框架内大多数的对象都使用了对象池技术，对象池技术有利于系统内存的稳定，减少GC的次数，提高系统的运行效率，事实证明对象池对系统稳定做出了极大的贡献，开发者也可以使用这一套对象池技术，增加对对象的复用，减少GC和NEW的频率，对系统毛刺现象和内存泄露方面都有很大的稳定性提升。

(5)异步客户端以及连接池 Mysql，Redis，Http客户端，Tcp客户端，等等其他更为复杂的客户端，在SD框架中均为异步的模式，异步解决了系统整体的并发能力，但异步客户端需要提供连接池维持，SD框架提供了连接池，开发者不需要自己管理连接池，只需要使用即可。

(6)协程、异步事件回调解决的是并发性能，但造成的是业务代码的混乱。SD框架提供了协程解决了这一问题，通过yield关键字提供对异步的同步写法，消除了业务书写上的大量回调嵌套，你可以通过yield+同步的写法实现异步的性能。 协程提供了一整套完整的体系，包括超时，异常，休眠，多路选择，以及创建用户协程等等功能。

(7)定时任务 顾名思义定时执行的任务。

(8)任务投递 支持将耗时任务投递到Task进程。

(9)自动Reload 可以开启框架的自动Reload功能，这样代码修改会被立即响应。

* 1. 运行环境
     1. centos7.4服务器
        1. centos介绍

CentOS（Community Enterprise Operating System，中文意思是：社区企业操作系统）是Linux发行版之一，它是来自于Red Hat Enterprise Linux依照开放源代码规定释出的源代码所编译而成。由于出自同样的源代码，因此有些要求高度稳定性的服务器以CentOS替代商业版的Red Hat Enterprise Linux使用。两者的不同，在于CentOS并不包含封闭源代码软件。

CentOS 是一个基于Red Hat Linux 提供的可自由使用源代码的企业级Linux发行版本。每个版本的 CentOS都会获得十年的支持（通过安全更新方式）。新版本的 CentOS 大约每两年发行一次，而每个版本的 CentOS 会定期（大概每六个月）更新一次，以便支持新的硬件。这样，建立一个安全、低维护、稳定、高预测性、高重复性的 Linux 环境。 [1] CentOS是Community Enterprise Operating System的缩写。

CentOS 是RHEL（Red Hat Enterprise Linux）源代码再编译的产物，而且在RHEL的基础上修正了不少已知的 Bug ，相对于其他 Linux 发行版，其稳定性值得信赖。CentOS在2014初，宣布加入Red Hat。

* + - 1. 最低硬件配置

为了满足系统开发设计的需要，表2.1为实现系统设计开发对硬件要求的最低配置，建议使用当前中等或以上计算机配置。

|  |  |
| --- | --- |
| CPU | 1核及其以上 |
| 内存 | 1G及其以上（建议2G以上） |
| 硬盘 | 20GB及其以上 |
| 光驱 | 无要求 |
| 软驱 | 无要求 |
| 显示器 | 无要求 |
| 其它计算机设备 | 网卡等联网设备 |

* + 1. 超文本预处理器之PHP7
       1. PHP7介绍

PHP（外文名:PHP Hypertext Preprocessor，中文名：“超文本预处理器”）是一种通用开源脚本语言。

PHP语法吸收了C语言、Java和Perl的特点，易于学习，使用广泛，主要适用于Web开发领域。PHP 独特的语法混合了C、Java、Perl以及PHP自创的语法。它可以比CGI或者Perl更快速地执行动态网页。用PHP做出的动态页面与其他的编程语言相比，PHP是将程序嵌入到HTML（标准通用标记语言下的一个应用）文档中去执行，执行效率比完全生成HTML标记的CGI要高许多；PHP还可以执行编译后代码，编译可以达到加密和优化代码运行，使代码运行更快的效果。

* + - 1. PHP7特点

PHP7要打破一切。 PHP开发人员应该接受打破版本之间向下兼容的定律。只要不允许大量的向后兼容，PHP7将是一个高度尊重的语言。其特点有：

（1）创建一个具体的核心语言 删除所有库方法，并保持在对象集中的核心方法。 您应该能够编写无需任何外部库或扩展PHP7和对基本输入/输出，字符串处理和数学一个很好的完整的语言。库以外的任何应该通过批准扩展。

（2）一切都当作一个对象 以从Ruby，Smalltalk和(主要)的Java对象，并把它一切当作对象。 整数是对象，字符串是对象，他们每个人都可以操作的方法， 我不相信PHP需要的Ruby和Smalltalk在对象之间传递彼此讯息的观念，而调用对象的方法才是最好的。

（3）一致的命名方法和类 由于PHP的最大的抱怨之一是不断要检查，(needle,haystack) 或(haystack, needle)，或some\_function()，或function\_some()，或someFunction()，一个一致的格式需要制定。

（4）让事情严格尝试传递到一个方法浮动字符串? 这是一个警告。

（5）一切是Unicode 在PHP6中的所有字符串都是Unicode，这很好，我主张PHP7也应该保持。

（6）中央启动点 创建一个主类或初始化，所有代码执行源于此。

（7）清理C代码我不是一个C的专家，但如果你比较了解Ruby的C代码到PHP的C代码，可以很容易地了解了PHP与Ruby的内部。 我非常熟悉PHP，所以我自己的写扩展更容易。

（8）摆脱eval() eval()是邪恶的。 如果你正在使用它，那么这是一个错的主意：这将打破PHPUnit，抛弃它从现在开始。

（9）支持操作符重载 因为一切都是对象，开发者只需掌握操作对象的方法即可。

（10）允许的方法签名

* + - 1. 内存工具redis-3.0.8

Redis是一个开源的使用ANSI C语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value数据库，并提供多种语言的API。从2010年3月15日起，Redis的开发工作由VMware主持。从2013年5月开始，Redis的开发由Pivotal赞助。

* + 1. git版本管理工具
       1. git介绍

git(读音为/gɪt/。)是一个开源的分布式版本控制系统，可以有效、高速的处理从很小到非常大的项目版本管理。git 是 Linus Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的版本控制软件。

git 是用于 Linux内核开发的版本控制工具。与常用的版本控制工具 CVS, Subversion 等不同，它采用了分布式版本库的方式，不必服务器端软件支持（wingeddevil注：这得分是用什么样的服务端，使用http协议或者git协议等不太一样。并且在push和pull的时候和服务器端还是有交互的。），使源代码的发布和交流极其方便。 Git 的速度很快，这对于诸如 Linux kernel 这样的大项目来说自然很重要。 git 最为出色的是它的合并跟踪（merge tracing）能力。

* + - 1. git版本工具安装

本系统程序使用git版本工具进行管理，其系统相关的框架以及部分相关组件均在github平台上面。因此，git作为本系统开发相关的必备工具。登录centos，在控制台中输入命令，yum install git进行自动安装。

* + 1. 简约客户端hiredis
       1. hiredis介绍

Hiredis是Redis数据库的简约C客户端库。它是简约的，因为它只是增加了对协议的 最小支持，但是同时它使用了一个高级别的printf-like API，所以对于习惯了printf风格的C编程用户来说，其非常容易使用，而且API中没有明确的绑定每个Redis命令。本系统sd框架启用redis模式运行系统，需要以来hiredis工具支持。

* + - 1. hiredis安装

由于安装过程中，hiredis安装经常容易出现出错，因此在本论文作安装记录。

git clone https://github.com/redis/hiredis.git

make -j

sudo make install

sudo ldconfig

/etc/ld.so.conf文件，在新的一行中加入库文件所在目录；

备注：如php\_error提示:

[15-Nov-2017 05:48:46 UTC] PHP Warning: PHP Startup: Unable to load dynamic library '/home/work/php-7.1.0/lib/php/extensions/no-debug-non-zts-20160303/swoole.so' - libhiredis.so.0.13: cannot open shared object file: No such file or directory in Unknown on line 0

则加入libhiredis.so.0.13所在路径.

* + 1. mysql服务器
       1. mysql介绍

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，目前属于 Oracle 旗下产品。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 (Relational Database Management System，关系数据库管理系统) 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。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

由于其社区版的性能卓越，搭配 PHP 和 Apache 可组成良好的开发环境。

* + - 1. mysql应用环境

与其他的大型数据库例如 Oracle、DB2、SQL Server等相比，MySQL 自有它的不足之处，但是这丝毫也没有减少它受欢迎的程度。对于一般的个人使用者和中小型企业来说，MySQL提供的功能已经绰绰有余，而且由于 MySQL是开放源码软件，因此可以大大降低总体拥有成本。

Linux作为操作系统，Apache 或Nginx作为 Web 服务器，MySQL 作为数据库，PHP/Perl/Python作为服务器端脚本解释器。由于这四个软件都是免费或开放源码软件（FLOSS)，因此使用这种方式不用花一分钱（除开人工成本）就可以建立起一个稳定、免费的网站系统，被业界称为“LAMP“或“LNMP”组合。

* + - 1. mysql存储引擎

MyISAMMySQL 5.0 之前的默认数据库引擎，最为常用。拥有较高的插入，查询速度，但不支持事务

InnoDB事务型数据库的首选引擎，支持ACID事务，支持行级锁定, MySQL 5.5 起成为默认数据库引擎

BDB源 自 Berkeley DB，事务型数据库的另一种选择，支持Commit 和Rollback 等其他事务特性

Memory所有数据置于内存的存储引擎，拥有极高的插入，更新和查询效率。但是会占用和数据量成正比的内存空间。并且其内容会在 MySQL 重新启动时丢失

Merge将一定数量的 MyISAM 表联合而成一个整体，在超大规模数据存储时很有用

Archive非常适合存储大量的独立的，作为历史记录的数据。因为它们不经常被读取。Archive 拥有高效的插入速度，但其对查询的支持相对较差

Federated将不同的 MySQL 服务器联合起来，逻辑上组成一个完整的数据库。非常适合分布式应用

Cluster/NDB高冗余的存储引擎，用多台数据机器联合提供服务以提高整体性能和安全性。适合数据量大，安全和性能要求高的应用

CSV： 逻辑上由逗号分割数据的存储引擎。它会在数据库子目录里为每个数据表创建一个 .csv 文件。这是一种普通文本文件，每个数据行占用一个文本行。CSV 存储引擎不支持索引。

BlackHole：黑洞引擎，写入的任何数据都会消失，一般用于记录 binlog 做复制的中继

EXAMPLE 存储引擎是一个不做任何事情的存根引擎。它的目的是作为 MySQL 源代码中的一个例子，用来演示如何开始编写一个新存储引擎。同样，它的主要兴趣是对开发者。EXAMPLE 存储引擎不支持编索引。

另外，MySQL 的存储引擎接口定义良好。有兴趣的开发者可以通过阅读文档编写自己的存储引擎。

本毕业设计由于在删除操作使用了事务机制，因此所选的mysql储存引擎为InnoDB类型。

* + - 1. mysql系统特性

1．使用 C和 C++编写，并使用了多种编译器进行测试，保证了源代码的可移植性。

2．支持 AIX、FreeBSD、HP-UX、Linux、Mac OS、NovellNetware、OpenBSD、OS/2 Wrap、Solaris、Windows等多种操作系统。

3．为多种编程语言提供了 API。这些编程语言包括 C、C++、Python、Java、Perl、PHP、Eiffel、Ruby,.NET和 Tcl 等。

4．支持多线程，充分利用 CPU 资源。

5．优化的 SQL查询算法，有效地提高查询速度。

6．既能够作为一个单独的应用程序应用在客户端服务器网络环境中，也能够作为一个库而嵌入到其他的软件中。

7．提供多语言支持，常见的编码如中文的 GB 2312、BIG5，日文的 Shift\_JIS等都可以用作数据表名和数据列名。

8．提供 TCP/IP、ODBC 和 JDBC等多种数据库连接途径。

9．提供用于管理、检查、优化数据库操作的管理工具。

10．支持大型的数据库。可以处理拥有上千万条记录的大型数据库。

11．支持多种存储引擎。

12.MySQL 是开源的，所以你不需要支付额外的费用。

13.MySQL 使用标准的 SQL数据语言形式。

14.MySQL 对 PHP 有很好的支持，PHP是目前最流行的 Web 开发语言。

15.MySQL是可以定制的，采用了 GPL协议，你可以修改源码来开发自己的 MySQL 系统。

16.在线 DDL/更改功能，数据架构支持动态应用程序和开发人员灵活性（5.6 [4] 新增）

17.复制全局事务标识，可支持自我修复式集群（5.6 [4] 新增）

18.复制无崩溃从机，可提高可用性（5.6 [4] 新增）

19.复制多线程从机，可提高性能（5.6 [4] 新增）

20.3倍更快的性能（5.7 [5] 新增）

21.新的优化器（5.7 [5] 新增）

22.原生JSON支持（5.7 [5] 新增）

23.多源复制（5.7 [5] 新增）

24.GIS的空间扩展（5.7 [5] 新增）

* + 1. 测试运行环境

测试依赖环境是否安装成功：

php –m 看其模块是否包括redis bcmath swoole，如不出现，则需要重新部署环境。

1. 系统安全
   1. centos服务器安全

服务器安全作为系统运行的基本条件，只有对服务器进行安全加固以及不断的优化，运行在该系统的程序才能更加稳定、快速。本毕业设计只要探讨“高性能内容发布网站系统”这个主题，因此在此不进行系统安全方面的详细介绍。

* 1. 服务进程安全

除了服务器的安全问题，其程序相关的服务安全同意非常重要。一个系统的正常运行往往需要多种不同的服务支持，如本系统程序运行，需要依赖redis服务、mysql服务、自身服务等。本系统为了方便管理上述相关服务进程，引入Supervisor工具进行管理。

Supervisor是用Python开发的一套通用的进程管理程序，能将一个普通的命令行进程变为后台daemon，并监控进程状态，异常退出时能自动重启。它是通过fork/exec的方式把这些被管理的进程当作supervisor的子进程来启动，这样只要在supervisor的配置文件中，把要管理的进程的可执行文件的路径写进去即可。也实现当子进程挂掉的时候，父进程可以准确获取子进程挂掉的信息的，可以选择是否自己启动和报警。supervisor还提供了一个功能，可以为supervisord或者每个子进程，设置一个非root的user，这个user就可以管理它对应的进程。

* 1. 程序模块安全
     1. 防止内存溢出

系统中可能会遇到大数组，其操作会导致系统发生内存溢出，因此在处理大数组时，设置PHP使用内存键值[ini\_set('memory\_limit','256M');]，同时设置超时限制[set\_time\_limit(600);]，另外，在使用完大数组后，使用unset对其进行销毁。

避免读取大文件，如读取大文件操作，则使用PHP的fopen，fgets函数进行指针操作，或使用SplFileObject文件操作类。

递归处理，在递归处理方面，需要注意无限递归/循环导致内存溢出，递归函数逻辑需要非常清晰。

* + 1. 参数过滤

SD框架在Controller父类中对其进行实例化，其功能主要是对原生swoole\_http\_request对象进行包装，让开发者可以轻松的获取get、post、header、cookie、file等值。“高性能内容发布网站系统”基于SD框架进行架构开发，因此，在参数过滤方面续承了统一处理，轻松过滤用户提交的相关参数，提高系统安全。

* + 1. 数据库安全

程序中的所有错误信息必须关闭或者屏蔽，防止系统信息涉漏。

服务器端的magic\_quotes，虽然他们默认都是打开的（magic\_quotes\_gpc），但仍然可能存在被恶意注入，因此主动关闭magic\_quotes\_gpc，全部自己手工对所有变量添加magic\_quotes，就是添加addslashes进行过滤。

1. 系统测试
2. 总结

一句老话说的：万事开头难。的确是这样的，经过将近三个月的努力终于完成了毕业论文（设计）的写作以及程序实现工作，有种如释重负的感觉。此外，还得出一个结论：知识必须通过应用才能真正实现其价值！有些东西以为学会了，但真正到要用的时候才发现是两回事，所以我认为只有到真正用得时候才是真正的学会了，理论和时间是不能分开的两兄弟。

通过这次毕业设计，我更加深入的认识PHP编程语言，对mysql数据库事务也进一步认识，REDIS的使用也越来越熟练。同时，通过SWOOLE、SD框架支持进一步了解HTTP协议、TCP协议、WEBSOCK协助等相关网络编程。

在此要感谢我的指导老师黎培兴老师对我悉心的指导，感谢老师们给我的帮助。在系统设计过程中，我通过查阅大量有关资料，与同学交流经验和自学，并向老师请教等方式，使自己学到了不少知识，也经历了不少艰辛，但收获同样巨大。在整个系统设计中我懂得了许多东西，也培养了我独立工作的能力，树立了对自己工作能力的信心，相信会对今后的学习工作生活有非常重要的影响。而且大大提高了动手的能力，使我充分体会到了在创造过程中探索的艰难和成功时的喜悦。

最后还要感谢同学们，在这次毕业论文（设计）写作中给我的支持和帮助。

参考资料：

[1][百度百科](https://baike.baidu.com/).[词条PHP（超文本预处理器）](https://baike.baidu.com/item/PHP/9337?fr=aladdin)[引用日期2018-02-05]

[2][SwooleDistributed官网](http://sd.youwoxing.net/).SwooleDistributed定义[引用日期2018-02-05]

[3]分布式版本控制系统 Git ．开源社区网[引用日期2012-09-25]