

攀枝花学院本科毕业设计（论文）

基于lamp架构的个人网站

学生姓名： 刘杰

学生学号： 201610802018

院 （系）： 数学与计算机学院

年级专业： 2016级信息与计算科学

指导教师： 赖国勇

助理指导教师：

二〇二零年五月

# 摘 要

**关键词**

# Abstract

**Keywords**:

目 录

[摘 要 I](#_Toc29309281)

[Abstract II](#_Toc29309282)

[**1.** 绪 论 6](#_Toc29309283)

[1.1. 立题依据： 6](#_Toc29309284)

[1.2. 面临的问题： 6](#_Toc29309285)

[1.3. 研究目的： 6](#_Toc29309286)

[1.4. 研究意义： 6](#_Toc29309287)

[1.5. 基本目标 7](#_Toc29309288)

[2. 开发工具及和环境介绍 9](#_Toc29309289)

[2.1. 开发工具 9](#_Toc29309290)

[2.2. LAMP简介 9](#_Toc29309291)

[2.3. MVC简介 10](#_Toc29309292)

[2.4. Django简介 11](#_Toc29309293)

[2.5. MySQL简介 12](#_Toc29309294)

[3. 系统可行性分析 15](#_Toc29309295)

[3.1. 经济可行性 15](#_Toc29309296)

[3.2. 技术可行性 15](#_Toc29309297)

[3.3. 发展预测 15](#_Toc29309298)

[4. 需求分析 16](#_Toc29309299)

[4.1. 功能需求 16](#_Toc29309300)

[4.1.1. 前端页面共有需求 16](#_Toc29309301)

[4.1.2. 主页需求 16](#_Toc29309302)

[4.1.3. 成果展示页需求 16](#_Toc29309303)

[4.1.4. IT技术论坛需求 16](#_Toc29309304)

[4.1.5. 精品收藏页需求 17](#_Toc29309305)

[4.1.6. 问题及其解答页需求 17](#_Toc29309306)

[4.1.7. 汉译文档页需求 17](#_Toc29309307)

[4.1.8. 个人文档页需求 17](#_Toc29309308)

[4.1.9. 登录页需求 17](#_Toc29309309)

[4.1.10. 游客注册页需求 17](#_Toc29309310)

[4.1.11. 后台需求 18](#_Toc29309311)

[4.2. 非功能性需求 18](#_Toc29309312)

[4.3. 设计约束 19](#_Toc29309313)

[5. 系统主体设计 18](#_Toc29309314)

[5.1. 设计原则 18](#_Toc29309315)

[5.2. 网站访问者结构 18](#_Toc29309316)

[5.3. 功能结构设计 19](#_Toc29309317)

[5.4. 系统流程分析 20](#_Toc29309318)

[5.5. 网站主体数据模型 22](#_Toc29309319)

[6. 系统主体详细设计 23](#_Toc29309320)

[6.1. 站长管理 23](#_Toc29309321)

[6.1.1. 站长信息ER图 23](#_Toc29309322)

[6.1.2. 前端基础设置ER图 23](#_Toc29309323)

[6.1.3. 导航栏ER图 24](#_Toc29309324)

[6.1.4. 数据表 24](#_Toc29309325)

[6.2. 账户管理模块 25](#_Toc29309326)

[6.2.1. 功能结构 25](#_Toc29309327)

[6.2.2. 流程图 25](#_Toc29309328)

[6.2.3. ER图 26](#_Toc29309329)

[6.2.4. 数据表 26](#_Toc29309330)

[6.3. 功能管理模块 27](#_Toc29309331)

[6.3.1. 功能结构 27](#_Toc29309332)

[6.3.2. 流程图 27](#_Toc29309333)

[6.3.3. ER图 28](#_Toc29309334)

[6.3.4. 数据表 28](#_Toc29309335)

[6.4. 数据管理模块 28](#_Toc29309336)

[6.4.1. 功能结构 28](#_Toc29309337)

[6.4.2. 流程图 29](#_Toc29309338)

[6.4.3. ER图 29](#_Toc29309339)

[6.4.4. 数据表 30](#_Toc29309340)

[6.5. 用户-功能关系表 30](#_Toc29309341)

[6.5.1. ER图 30](#_Toc29309342)

[6.5.2. 数据表 30](#_Toc29309343)

[7. 系统主体设计实现 27](#_Toc29309344)

[7.1. 网站安装系统的具体实现 27](#_Toc29309345)

[7.1.1. 检测程序的设计实现 27](#_Toc29309346)

[7.1.2. 创建链接文件和基础数据表 29](#_Toc29309347)

[7.1.3. 基本信息的录入 30](#_Toc29309348)

[7.2. 网站登录检测系统与注册系统的具体实现 31](#_Toc29309349)

[7.2.1. 检测系统的实现 31](#_Toc29309350)

[7.2.2. 登录系统的实现 31](#_Toc29309351)

[7.2.3. 注册系统的实现 33](#_Toc29309352)

[7.3. 网站后台管理系统的实现 33](#_Toc29309353)

[7.3.1. 数据展示 33](#_Toc29309354)

[7.3.2. 数据修改 34](#_Toc29309355)

[7.3.3. 添加数据 34](#_Toc29309356)

[7.3.4. 删除数据 35](#_Toc29309357)

[7.3.5. 访问保护 36](#_Toc29309358)

[8. 系统扩展的设计与实现 43](#_Toc29309359)

[8.1. 系统扩展 43](#_Toc29309360)

[8.1.1. 系统扩展的设计 43](#_Toc29309361)

[8.1.2. 系统扩展的实现 44](#_Toc29309362)

[8.2. 日志模块 45](#_Toc29309363)

[8.2.1. 数据模型 45](#_Toc29309364)

[8.2.2. 数据表 45](#_Toc29309365)

[8.2.3. 代码实现 45](#_Toc29309366)

[8.2.4. 最终结果 45](#_Toc29309367)

[9. 总结 50](#_Toc29309368)

[参考文献 51](#_Toc29309369)

# 绪 论

## 立题依据：

21世纪是网络的世纪，人类生活现在大部分依赖于互联网。互联网为人们提供方便，快捷，高效的服务，拉近了人与人之间的距离，更是在工作中提高了工作的效率和时间。在访问互联网的过程中，网站访问的时间又占据了相当大的比例。网站类型多样，其中的个人网站是服务于个人的综合型服务类网站，通过使用个人网站，无论是在平时的工作，生活中，还是学习，娱乐中，它都能提供积极的正面影响，从而提高生活满意度。所以建立一个给予lamp架构的个人网站不仅仅只是对我大学4年所学知识的一个梳理总结实践的过程，其成果对我以后的生活依然有很大帮助。

## 面临的问题：

从自身来讲，现代社会的高节奏的生活中，有效地提高工作和学习效率是我们的追求。但是，我们对自己的性格，爱好，能力等了解不清楚，对自己的总结和对未来的规划相当模糊，从而引发了相当多的工作，学习和生活问题，苦恼人生艰难；对他人来讲，由于时间短暂且接触不多，了解一个陌生人的各项特征都非常困难的原因，寻找志同道合的人相处变得十分困难，引发了很多社交问题。

## 研究目的：

开发一款基于lamp架构的个人网站，囊括（1）角色和用户管理（注册、维护、登陆等）\*；（2）主页。（3）IT技术论坛。（4）学习心得（问题解答）；（5）成果展示。（6）精品收藏：好文共读，音视频分享。（7）个人简历；（8）系统备份与恢复等主要页面或功能，解决学习效率低，自我认识不足等问题，达到服务个人生活，展示个人性格，分享学习成果，记录人生旅途的目的。

## 研究意义：

对自己来讲，个人网站是管理生活的一款工具，它使生活更具条理性，目标性，规范性，能突出自身的优势，暴露自身的劣势；对他人来讲，个人网站能不限时间，不限地点，随时随地都能浏览，是别人初步了解自己的一条极为有效的途径。

## 基本目标

本次课题的基本目标是开发基于lamp架构的小型分享型网站。能实现当访问者访问网站时页面时响应速度较快，展示的页面结构基本合理，功能丰富，操作简单。能使访问者基本了解到网站做了什么，有什么作用。使本站拥有者能够轻松的管理自己的生活，学习甚至工作，展示自己兴趣，爱好甚至才能。

# 开发工具及和环境介绍

## 开发工具

本网站是基于lamp架构开发的小型网站，其主要开发工具为PHPstudy,和Hbuilder, Notepad++以及一台云主机。

phpStudy是一个PHP调试环境的程序集成包。该程序包集成最新的[Apache](https://baike.baidu.com/item/Apache)+[PHP](https://baike.baidu.com/item/PHP)+MySQL+[phpMyAdmin](https://baike.baidu.com/item/phpMyAdmin)+[ZendOptimizer](https://baike.baidu.com/item/ZendOptimizer),一次性安装,无须配置即可使用,是非常方便、好用的PHP调试环境.该程序不仅包括PHP调试环境,还包括了开发工具、开发手册等。

Notepad++是[Wind](https://baike.baidu.com/item/Wind)[ows](https://baike.baidu.com/item/ows)操作系统下的一套[文本编辑器](https://baike.baidu.com/item/%E6%96%87%E6%9C%AC%E7%BC%96%E8%BE%91%E5%99%A8/8853160)(软件版权许可证: [GPL](https://baike.baidu.com/item/GPL))，有完整的中文化接口及支持多国语言编写的功能([UTF8](https://baike.baidu.com/item/UTF8)技术)。Notepad++功能比 [Windows](https://baike.baidu.com/item/Windows) 中的 [Notepad](https://baike.baidu.com/item/Notepad)(记事本)强大，除了可以用来制作一般的纯文字说明文件，也十分适合编写计算机程序代码。Notepad++ 不仅有语法高亮度显示，也有语法折叠功能，并且支持宏以及扩充基本功能的外挂模组。

HBuilder是[DCloud](https://baike.baidu.com/item/DCloud)（[数字天堂](https://baike.baidu.com/item/%E6%95%B0%E5%AD%97%E5%A4%A9%E5%A0%82/244387)）推出的一款支持[HTML5](https://baike.baidu.com/item/HTML5)的[Web](https://baike.baidu.com/item/Web/150564)开发[IDE](https://baike.baidu.com/item/IDE/8232086)。 HBuilder的编写用到了[Java](https://baike.baidu.com/item/Java/85979)、[C](https://baike.baidu.com/item/C/7252092)、Web和[Ruby](https://baike.baidu.com/item/Ruby/11419)。HBuilder[本身](https://baike.baidu.com/item/%E6%9C%AC%E8%BA%AB/126627)主体是由Java编写。HBuilder的最大优势，通过完整的语法提示和[代码](https://baike.baidu.com/item/%E4%BB%A3%E7%A0%81)输入法、代码块等，大幅提升[HTML](https://baike.baidu.com/item/HTML)、[js](https://baike.baidu.com/item/js/10687961)、[css](https://baike.baidu.com/item/css/5457)的开发效率。

云主机是[云计算](https://baike.baidu.com/item/%E4%BA%91%E8%AE%A1%E7%AE%97/9969353)在基础设施应用上的重要组成部分，位于云计算产业链金字塔底层，产品源自[云计算平台](https://baike.baidu.com/item/%E4%BA%91%E8%AE%A1%E7%AE%97%E5%B9%B3%E5%8F%B0/9471851)。该平台整合了互联网应用三大核心要素：计算、存储、网络，面向用户提供公用化的互联网基础设施服务。云主机是一种类似VPS主机的[虚拟化技术](https://baike.baidu.com/item/%E8%99%9A%E6%8B%9F%E5%8C%96%E6%8A%80%E6%9C%AF/276750)， VPS是采用[虚拟软件](https://baike.baidu.com/item/%E8%99%9A%E6%8B%9F%E8%BD%AF%E4%BB%B6/207683)，VZ或VM在一台主机上虚拟出多个类似独立主机的部分，能够实现单机多用户，每个部分都可以做单独的操作系统，管理方法同主机一样。 而云主机是在一组集群主机上虚拟出多个类似独立主机的部分，集群中每个主机上都有云主机的一个镜像，从而大大提高了[虚拟主机](https://baike.baidu.com/item/%E8%99%9A%E6%8B%9F%E4%B8%BB%E6%9C%BA/208420)的安全稳定性，除非所有的集群内主机全部出现问题，云主机才会无法访问

## LAMP简介

LAMP是指一组通常一起使用来运行动态网站或者服务器的[自由软件](https://baike.baidu.com/item/%E8%87%AA%E7%94%B1%E8%BD%AF%E4%BB%B6)名称首字母缩写，包括[Linux](https://baike.baidu.com/item/Linux)（[操作系统](https://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F)），[Apache](https://baike.baidu.com/item/Apache)（网页服务器），[MariaDB](https://baike.baidu.com/item/MariaDB)或[MySQL](https://baike.baidu.com/item/MySQL)，（[数据库管理系统](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F)（或者[数据库服务器](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E6%9C%8D%E5%8A%A1%E5%99%A8)）， [PHP](https://baike.baidu.com/item/PHP)、[Perl](https://baike.baidu.com/item/Perl)或[Python](https://baike.baidu.com/item/Python)（[脚本语言](https://baike.baidu.com/item/%E8%84%9A%E6%9C%AC%E8%AF%AD%E8%A8%80)），是一组常用来搭建动态网站或者服务器的开源软件。它们本身都是各自独立的程序，但是因为常被放在一起使用，拥有了越来越高的兼容度，共同组成了一个强大的Web应用程序平台。随着开源潮流的蓬勃发展，开放源代码的LAMP已经与J2EE和.Net商业软件形成三足鼎立之势，并且该软件开发的项目在软件方面的投资成本较低，因此受到整个IT界的关注。其架构图如下：

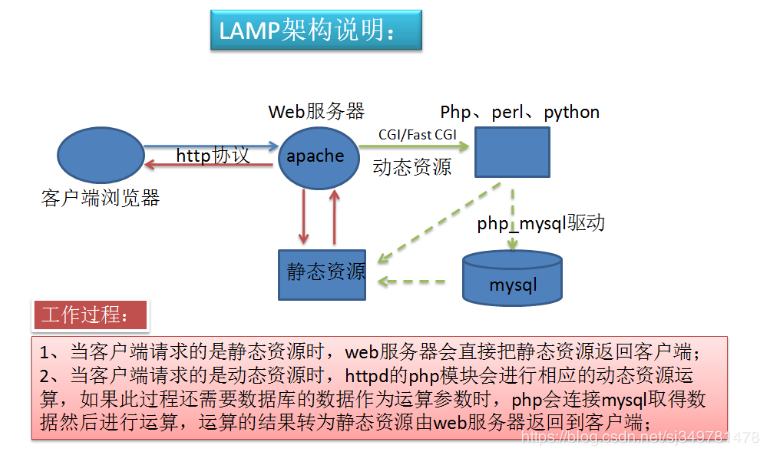


图2.1 lamp架构图

## MVC简介

[MVC](https://baike.baidu.com/item/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。其中，Model（模型）表示应用程序核心（数据库），通常模型对象负责在数据库中存取数据。；View（视图）显示数据（前端），通常是依据模型数据创建的；Controller（控制器）处理输入（后端），通常负责从视图读取数据，控制用户输入，并向模型发送数据。其工作流程如下：



图2.2 MVC工作流程图

## Django简介

Django是一个遵循 MVC 设计模式的框架。MVC是Model、View、Controller三个单词的简写，分别代表模型、视图、控制器。Django其实也是一个MTV 的设计模式。MTV是Model、Template、View三个单词的简写，分别代表模型、模版、视图  。但是在Django中，控制器接受用户输入的部分由框架自行处理，所以 Django 里更关注的是模型（Model）、模板(Template)和视图（Views），称为 MTV模式。它们各自的职责如下：

|  |  |
| --- | --- |
| 层次 | 职责 |
| 模型（Model），即数据存取层 | 处理与数据相关的所有事务： 如何存取、如何验证有效性、包含哪些行为以及数据之间的关系等。 |
| 模板(Template)，即表现层 | 处理与表现相关的决定： 如何在页面或其他类型文档中进行显示。 |
| 视图（View），即业务逻辑层 | 存取模型及调取恰当模板的相关逻辑。模型与模板的桥梁。 |

Django 视图不处理用户输入，而仅仅决定要展现哪些数据给用户，Django 模板 仅仅决定如何展现Django视图指定的数据。或者说, Django将MVC中的视图进一步分解为 Django视图 和 Django模板两个部分，分别决定 “展现哪些数据” 和 “如何展现”，使得Django的模板可以根据需要随时替换，而不仅仅限制于内置的模板。

Django控制器部分，由Django框架的URLconf来实现。URLconf机制是使用正则表达式匹配url，然后调用合适的Python函数。URLconf对于URL的规则没有任何限制，你完全可以设计成任意的URL风格，不管是传统的，RESTful的，或者是另类的。框架把控制层给封装了，无非与数据交互这层都是数据库表的读,写,删除,更新的操作。在写程序的时候，只要调用相应的方法就行了，感觉很方便。程序员把控制层东西交给Django自动完成了。 只需要编写非常少的代码完成很多的事情。所以，它比MVC框架考虑的问题要深一步，因为我们程序员大都在写控制层的程序。这个工作交给了框架，仅需写很少的调用代码，大大提高了工作效率。其工作流程图如下：



图2.3 Django工作流程图

## MySQL简介

MySQL是一种[开放源代码](https://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E6%BA%90%E4%BB%A3%E7%A0%81)的关系型[数据库管理](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86)系统（RDBMS），MySQL[数据库系统](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%B3%BB%E7%BB%9F)使用最常用的数据库管理语言--[结构化查询语言](https://baike.baidu.com/item/%E7%BB%93%E6%9E%84%E5%8C%96%E6%9F%A5%E8%AF%A2%E8%AF%AD%E8%A8%80)（SQL）进行数据库管理。

由于MySQL是开放源代码的，因此任何人都可以在General Public License的许可下下载并根据个性化的需要对其进行修改。MySQL因为其速度、可靠性和适应性而备受关注。大多数人都认为在不需要[事务](https://baike.baidu.com/item/%E4%BA%8B%E5%8A%A1)化处理的情况下，MySQL是管理内容最好的选择。

MySQL，虽然功能未必很强大，但因为它的开源、广泛传播，导致很多人都了解到这个数据库。它的历史也富有传奇性。

# 系统可行性分析

可行性分析是通过对[项目](https://baike.baidu.com/item/%E9%A1%B9%E7%9B%AE/477803)的主要内容和配套条件，如[市场需求](https://baike.baidu.com/item/%E5%B8%82%E5%9C%BA%E9%9C%80%E6%B1%82/2233476)、资源供应、建设规模、工艺路线、设备选型、环境影响、资金筹措、盈利能力等，从[技术](https://baike.baidu.com/item/%E6%8A%80%E6%9C%AF/832247)、经济、工程等方面进行调查研究和分析比较，并对项目建成以后可能取得的财务、经济效益及社会环境影响进行预测，从而提出该项目是否值得投资和如何进行建设的咨询意见，为[项目决策](https://baike.baidu.com/item/%E9%A1%B9%E7%9B%AE%E5%86%B3%E7%AD%96/10196438)提供依据的一种综合性的[系统分析方法](https://baike.baidu.com/item/%E7%B3%BB%E7%BB%9F%E5%88%86%E6%9E%90%E6%96%B9%E6%B3%95/9549282)。由此，我做了以下分析：

## 经济可行性

建立并上线一个基于lamp架构的个人网站需要购买一台虚拟云主机和一个域名，差不多花费一百多人民币一年，虚拟云主机也可以申请免费的，所以最少可以花费十几元人民币就可以在一年中拥有自己的个人网站，这个个人网站能为学习，工作，生活带来巨大帮助，节省资金，符合经济可行性。

## 技术可行性

2010年后期，随着国家对IT人才的大力培养，现在的网站建设者日益增多，我国的网站建设整体水平也有了全面提升，各种教学资料唾手可得，学习资源丰富，技术性论坛繁多。本人通过学校教学和网络自主学习已初步掌握建设网站的各个基本要点，并已经私下做了功能测试，有能力完成本次工作 ，具备技术可行性。

## 发展预测

个人分享型网站在知识共享、知识创新、展现自我方面有着显著的作用。知识共享水平的高低对一个企业甚至一个国家的创新能力有着显著的影响，个人分享型网站是发展知识共享水平，激发创新能力，展现个人独特魅力的有效方式，网站小型化，私人化将可能是未来发展的主要方向之一。另一方面，目前我国许多企业、学校建立了内部的虚拟社区、学习论坛、知识共享库等，然而却大多数都比较大众化，奉行拿来主义，无法有效的提高知识共享和创新，这也导致了网上知识杂乱，难以系统有效的形成知识体系，开发创新意识，因此，研究如何创建自己独有的知识体系，形成自己看待事情的观点，分享自己的创新方法也可能是未来研究的主要方向之一。

# 需求分析

需求分析是软件计划阶段的重要活动，也是软件生存周期中的一个重要环节，该阶段是分析系统在功能上需要“实现什么”，而不是考虑如何去“实现”。需求分析的目标是把用户对待开发软件提出的“要求”或“需要”进行分析与整理，确认后形成描述完整、清晰与规范的文档，确定软件需要实现哪些功能，完成哪些工作。我由此对我的项目做了以下分析：

## 功能需求

功能性需求即软件完成哪些事，实现哪些功能，以及为了向其用户提供有用的功能所需执行的动作。功能性需求是软件需求的主体。由此，我做了以下功能需求分析：

### 前端页面共有需求

1. 每个前端页面都有顶端目录
2. 每个前端页面都有登录链接，注册链接，(Github链接)
3. 每个页面都有网站认证

### 主页需求

（1）进入主页的响应时间越低越好，避免访问者流失

（2）主页的页面展示应该美观，结构性好，吸引访问者眼球

（3）主页应该概括性的展示本网站是做什么的，有哪些功能，为访问者提供基础信息

（4）主页应该提供各个功能页面的跳转链接，为访问者提供路径

### 成果展示页需求

（1）具有展示成果功能

（2）具备所得成果的说明及其相关描述，便于访问者了解

（3）具有下载功能，便于访问者下载体验

（4）所显示成果应该按最近时间展示，便于展示最新成果。

（5）应该是模块化展示，使结构清晰，美观。

### IT技术论坛需求

（1）应该具有针对不同主题的分类展示功能

（2）应该提供权限用户的反馈提交功能

### 精品收藏页需求

（1）应该具有针对不同主题的分类展示功能

（2）具备音频，视屏操控功能

（3）应该具备原创作品和收藏作品区别标识，收藏作品应该提供相关链接

### 问题及其解答页需求

（1）具备问题搜索功能，方便用户精确定位

（2）具备问题及其解答的分页展示功能

（3）具备小知识总结模块

### 汉译文档页需求

（1）具备单词在线翻译功能

（2）具备已翻译单词收集功能

（3）具备已翻译文章浏览功能

（4）按时间排序

### 个人文档页需求

（1）具备个人基本信息展示模块

（2）具备技能图形化展示模块

（3）具备重要事件展示模块

（4）具备日志记录功能

### 登录页需求

（1）验证码检验功能

（2）找回密码功能

（3）记住账号密码

（4）权限识别功能

### 游客注册页需求

（1）账号检测功能

（2）邮件发送功能

### 后台需求

（1）具备账户管理功能，包括密码更改，权限修改等

（2）具备根据不同权限展现不同后台的功能

（3）如果是最高权限者拥有修改他人权限及其他人的功能

（4）如果是最高权限者拥有上传文件，管理文件的功能

（5）如果是最高权限者拥有设置前端展示信息的功能

（6）如果是普通权限者拥有管理自己信息及提交评论和自己作品的功能

（7）如果是最高权限者拥有添加新模块的功能

## 非功能性需求

1. 运行环境需求分析

客户端：

硬件： 不限；

操作系统： 不限

软件： 网页浏览器。

服务器：

硬件： 虚拟云主机；

操作系统： Linux；

软件： Apache， [MariaDB](https://baike.baidu.com/item/MariaDB)或[MySQL](https://baike.baidu.com/item/MySQL)。

1. 安全性需求分析。

第一，在登录功能做好防sql注入和密码加密，保护个人信息安全；第二，做好反爬虫设置，避免恶意攻击；第三，做好系统以及数据库备份与还原功能，保证数据安全。

（3）设计可扩展性分析。

采用mvc设计模式，实现前后端分离，使前端界面可因需求改变而不影响功能。采用模块嵌入法，为以后网站功能添加保留可扩展接口。

## 设计约束

（1）网站建设采取lamp架构，即运行时必须基于[Linux](https://baike.baidu.com/item/Linux)环境，网页服务器必须是Apache, [数据库服务器](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E6%9C%8D%E5%8A%A1%E5%99%A8)必须是[MariaDB](https://baike.baidu.com/item/MariaDB)或[MySQL](https://baike.baidu.com/item/MySQL)，[脚本语言](https://baike.baidu.com/item/%E8%84%9A%E6%9C%AC%E8%AF%AD%E8%A8%80)是[PHP](https://baike.baidu.com/item/PHP)、[Perl](https://baike.baidu.com/item/Perl)或[Python](https://baike.baidu.com/item/Python)。

# 系统主体设计

## 设计原则

基于lamp架构的个人网站是一个比较容易实现的系统，因为它已经是一个明确，成熟的体系。但对于其中功能，结构的设计往往关乎使用者和访问者的满意度。所以，对访问者来说，网站的结构必须清晰，内容丰富有趣；对使用者来说，功能必须丰富，对于数据库的设计必须合理可靠，系统必须稳定，数据必须正确，操作必须简洁。因此，在设计该服务系统的时候，我们必须遵守以下几个原则：

(1) 单一职责原则 ：一个接口或类只有一个原因引起变化，也就是一个接口或类只有一个职责，它就负责一件事情，其优点是类的复杂性降低，实现什么职责都有清晰明确的定义，可读性提高，复杂性降低，可维护性提高。

(2) 开闭原则：不以改动原有类的方式来实现新需求，而是应该以实现事先抽象出来的接口（或具体类继承抽象类）的方式来实现，其优点是在于可以在不改动原有代码的前提下给程序扩展功能。增加了程序的可扩展性，同时也降低了程序的维护成本。

(3) 依赖倒置原则：即高层模块不应该依赖底层模块，两者都应该依赖抽象；抽象不应该依赖细节；细节应该依赖抽象。

(4) 接口分离原则：即客户端不应该依赖它不需要实现的接口。不建立庞大臃肿的接口，应尽量细化接口，接口中的方法应尽量少。需要注意的是接口的力度也不能太小，如果过小，则会造成接口数量过多，使设计复杂化。

（5）迪米特法则：即一个类应该只和它的成员变量，方法的输入，返回参数中的类作交流，而不应该引入其他的类，其优点是降低类与类之间的耦合，减少类与类之间的关联程度，让类与类之间的协作更加直接。

（6）里氏替换原则：所有引用基类的地方必须能透明地使用其子类的对象，也就是说子类对象可以替换其父类对象，而程序执行效果不变，其优点是可以检验继承使用的正确性，约束继承在使用上的泛滥。

## 网站访问者结构

在设计原则下，个人网站面向的使用人物对象有三类，分别为普通网站访问者，权限访问者和网站管理员。

普通网站访问者是属于单纯的浏览网页而极少有交互操作的网站访问者，普通网站访问者浏览到的内容是网站管理员开放展示的内容，一些网站功能将受到限制，比如说留言等功能。

权限访问者由普通网站访问者通过注册功能被网站管理者赋予功能权限，使权限访问者拥有使用网站特定权限功能的能力，比如下载，上传审批，留言等功能。

网站管理员管理网站内容和注册人员信息。具体结构图如下：

5.1 关系结构图

## 功能结构设计

功能结构图是对硬件、软件、[解决方案](https://baike.baidu.com/item/%E8%A7%A3%E5%86%B3%E6%96%B9%E6%A1%88)等进行解剖，详细描述功能列表的结构，构成，剖面的从大到小，从粗到细，从上到下等而描绘或画出来的结构图。从概念上讲，上层功能包括 (或控制)下层功能，愈上层功能愈笼统，愈下层功能愈具体。所以，我按需求分析绘出如下功能结构图：



图5.2系统功能结构图

从功能结构图中可以看出，本网站主要分为两个部分：前端和后端。前端的主要作用是展示信息和登录注册接口，后端的主要作用是数据操作。我把前端根据展示的内容分为主页，即网站的初始页，访问者通过主页可以了解到网站基本信息并能通过链接到其他页面；成果展示页，主要是以已形成结果的形式向访问者提供展示，并提供成果的下载链接；IT技术论坛主要是对IT技术的分享讨论；精品收藏页保存站主喜好的视屏音频和文章；汉译文档可作为学习英语和收集单词页面；个人文档页记录站长信息和日志；问题及其解答页记录生活学习中的问题及其答案。我把后端根据数据类型分为账户管理，可调整已注册用户的基本信息；权限管理，仅站长拥有该功能，可调整已注册用户对网站已有功能的权限；文件管理调整非音频视屏文件基本信息及其对应模块；媒体管理主要是对音频视屏文件的管理；前端设置主要是对整个网站主题的设置，比如背景，字体等；数据管理主要是对提交数据的审批调整；功能管理主要是网站功能的扩展和禁用。

## 系统流程分析

由系统构架可知，本网站主要面向3类用户，分别为普通网站访问者，权限访问者和网站管理员。而这3类用户，访问网站的流程不完全相同，分别有他们的特点。

针对站长，站长需要首先配置网站的基础设置才能使网站能够被访问。所以当网站开始运行时，网站的检测程序首先检测本网站是否具有数据库链接文件且文件中的配置项是否正确，如果满足条件则重定向到网站主页。如果不瞒足条件则重定向到网站配置页。在网站配置页填写链接数据库的基本信息后，系统根据所填写的基本信息尝试链接网站对应数据库；如果网站链接失败，提示失败原因，返回网站配置页重新填写链接数据库的基本信息，直到链接成功后，网站程序根据事先定义好的基础数据库模板建立数据表，创建数据库链接文件，注入网站基本信息，重新执行网站的检测程序。具体网站配置流程图如图5.3：



图5.3网站配置流程图

针对访问者，网站对普通访问者和注册访问者具有不同的响应措施。当网站被访问时，网站将检测该用户是否登录；如果未登录，则显示登录与注册链接并加载普通网站界面；如果登录则显示后台链接，并根据权限加载页面。可利用cookies实现。如图5.4:



图5.4网站访问流程图

普通的访问者可以通过登录和注册链接成为注册访问者获得权限，所以具备一个登录网站的流程。访问者通过前台的登录链接进入登录界面，填写账号密码，网站检测到账号密码匹配后，根据其拥有权限响应不同后台功能。如图5.5



图5.5网站登录流程图

## 网站主体数据模型

由网站基本结构，我分析了本网站的基本数据模型。一个网站只有一名站长，一名站长可以管理多名注册者，一名注册者可以提交多条数据被站长审核，一名注册者可以拥有多项功能，一项功能可以被多名注册者使用，一项功能只能处理一类资源。资源属于可扩展数据。如图5.6：



图5.6网站数据模型图

# 系统主体详细设计

前面我进行了整个网站的整体设计，接下来我将针对系统的各个主要功能模块进行更加详细的设计，包括模块功能结构的进一步划分，对该模块的操作流程和模块关联的数据结构都将一一叙述。

## 站长管理

站长管理包括两个子模块，分别是站长信息设置和前端设置，而前端设置又分为网站基础设置和导航栏设置。前端设置模块的主要内容是网站每个页面所共有的内容设置

### 站长信息ER图

站长信息包含的属性有姓名，年龄，学历，专业，爱好，特长，优缺点，人生规划，自我描述。



图6.1.1 站长信息E-R图

### 前端基础设置ER图

前端设置属性包括网站的标题和网站图标，以及网站的备注信息。



图6.1.2 前端基础设置E-R图

### 导航栏ER图

导肮栏的基础对象是每一个导航链接，每个导航链接都有它的tid，father\_id,名称，对应目标链接地址；father\_id表示它的父导航tid，顶级导航的父导航tid为0。



图6.1.3导航栏E-R图

### 数据表

表6.1 站长信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **空** | **默认** | **说明** |
| id | int(11) | 否 |  | 主键 |
| name | varchar(30) | 否 |  | 名称 |
| age | int(3) | 否 |  | 年龄 |
| rfs | varchar(30) | 否 |  | 学历 |
| profession | varchar(30) | 否 |  | 专业 |
| like | text | 是 | NULL | 爱好 |
| specialty | text | 是 | NULL | 特长 |
| pac | text | 是 | NULL | 优缺点 |
| plan | text | 是 | NULL | 人生规划 |
| selfintro | text | 是 | NULL | 自我描述 |

表6.2前端设置表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **空** | **说明** |
| set\_id | int(2) | 否 | 主键 |
| web\_name | varchar(255) | 否 | 网站名称 |
| webimg\_url | varchar(255) | 否 | 网站图标 |
| webfooter\_info | varchar(255) | 否 | 网站备注信息 |

表6.3导航栏信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **空** | **说明** |
| tid | int(2) | 否 | 主键 |
| father\_id | int(2) | 否 | 父导航 |
| tname | varchar(255) | 否 | 名称 |
| turl | varchar(255) | 否 | 链接地址 |

## 账户管理模块

账户管理模块是网站的基础模块之一，它是网站使用者管理网站内容的基本保障，是仅仅只用站长才能使用的模块之一，是网站最高权限的体现，是网站管理者对注册者的资源调度中心。

### 功能结构

账户管理模块的功能主要是对注册者信息的各种处理。比如说用户名更改，密码更改，用户添加，用户删除。当然，因为通过注册功能成为网站用户的访问者仅仅拥有网站的最低级权限，账户管理模块的主要功能还是对用户权限的调配与账户禁用。



图6.1.1账户管理结构图

### 流程图

当进入账户管理模块时，系统将查询数据库获得所有账户信息并以行列式的方式展示出来，展示出来的信息将在末尾追加功能按钮，点击相应按钮将触发相应提示，如果是则提交数据，修改数据库中的内容后，更新展示页面；如果否则直接刷新页面。



图6.1.2账户管理流程图

### ER图

账户管理的基础对象是用户，所以每个用户都有自己的编号uid,用户名，密码，email；用户名，密码用于登录网站后台；email将可以用来找回密码，所以应该具有唯一限制；状态表示该账户能否使用；权限限制该账户在本网站能使用的功能；系统会自动记录用户的注册时间。



图6.1.3账户管理E-R模型图

### 数据表

表6.4用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **空** | **默认** | **说明** |
| uid | int(11) | 否 |  | 主键 |
| uname | varchar(255) | 否 |  | 用户名 |
| upwd | varchar(255) | 否 |  | 密码 |
| utime | date | 否 |  | 注册时间 |
| ustatus | tinyint(1) | 否 | 1 | 状态 |
| uemail | varchar(255) | 否 |  | 邮箱 |

## 功能管理模块

功能管理模块也是网站的基础模块之一，此模块控制整个网站的功能，也是网站扩展功能的接口。每个功能必须在该模块备注后才能正常使用。它也是整个网站对权限的实际操作者。功能管理模块仅站长拥有权限操作。

### 功能结构

功能管理模块的功能主要是对已有功能的查询，修改，删除，以及对新功能的添加。



图6.2.1功能管理结构图

### 流程图

当进入功能管理模块时，系统将查询数据库获得所有功能表中的信息并以行列式的方式展示出来，展示出来的信息将在末尾追加功能按钮，点击相应按钮将触发相应提示，如果是则提交数据，修改数据库中的内容后，更新展示页面；如果否则直接刷新页面。



图6.2.2功能管理流程图

### ER图

功能管理的对象是网站对应的每一个功能，所以每一个功能都必须有它独有的编号fid，名称最好能体现其功能；关联页面是该功能的一个具体实现地址；关联父功能是其上级的直接父级功能的编号，其中顶级功能的父级功能编号为0；功能级别代表它的应用层级，顶级功能的层级为1； 状态控制该功能能否使用。



图6.2.3功能管理E-R图

### 数据表

表6.5功能表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **空** | **默认** | **说明** |
| fid | int(11) | 否 |  | 主键 |
| fpid | int(11) | 否 |  | 父功能 |
| fname | varchar(255) | 否 |  | 名称 |
| furl | varchar(255) | 否 |  | 关联页面地址 |
| fgrade | int(2) | 否 | 0 | 功能级别 |
| fstatus | tinyint(1) | 否 | 1 | 状态 |

## 数据管理模块

数据管理模块主要是站长对注册者提交数据的审批检验模块。是站长与注册者的主要交流平台。

### 功能结构

数据管理模块的主要功能是接收前端提交的数据给予站长审核，处理，分类，删除。



图6.3.1数据管理结构图

### 流程图

当进入数据管理模块时，网站将查询未被审核的历史提交记录展示给站长审核，如果审核通过则修改数据状态表示该数据可推送，否则删除该条数据。



图6.3.2数据管理流程图

### ER图

数据管理的对象是注册者提交的每一条数据，它们都有属于自己的编号，创作人，内容，系统将根据提交点将数据类型作为附加属性提交，系统会自动记录用户的提交时间。



图6.3.3数据管理E-R图

### 数据表

表6.6提交数据表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **空** | **说明** |
| Did | int(11) | 否 | 主键 |
| Uid | int(11) | 否 | 用户id |
| Data | text | 否 | 内容 |
| Dtype | varchar(255) | 否 | 数据类型 |
| Dtime | date | 否 | 提交时间 |

## 用户-功能关系表

户-功能关系表也叫权限表，此表用于描述注册者和可用功能的关系。

### ER图

权限表现为用户-功能关系，所以每个对应关系它们都有属于自己的编号Aid，层级表示该用户的权限等级。



图6.5.1用户-功能 E-R图

### 数据表

表6.6用户-功能关系表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **空** | **说明** |
| Aid | int(11) | 否 | 主键 |
| Agrede | tinyint(2) | 否 | 权限层级 |
| Uid | int(11) | 否 | 用户id |
| Fid\_having | varchar(255) | 否 | 对应功能 |

# 系统主体设计实现

系统主体设计实现部分是我对前面做的主体设计部分的具体实现部分，在这一部分我将阐述我是怎样根据我的设计来实现具体的功能，具体的实现代码，最终的结果反馈和实现过程中所遇到的问题及其解决方案。

## 网站安装系统的具体实现

网站安装系统是站长配置网站时使用的系统。我的设想是把我的网站文件夹放置在网站可访问域名空间下，当网站被访问时就会触发安装检测程序，如果已安装则跳转至主页，否则进入网站配置流程，完成网站配置后跳转至主页。检测程序的设计实现如下：

### 检测程序的设计实现

在检测程序中，设想的功能是检测数据库链接信息存储文件（sql\_conn.txt）是否存在,数据库链接信息是否正确以及页面跳转，所以该程序的主要代码有：

dirname(\_\_FILE\_\_)：文件路径获取

file\_exists(file)：检测文件是否存在

Header("Location:$url"):域名重定向

include\_once "model/conn.php"：文件引用

Class\_exists("Mysql")：检测类是否存在

在conn.php文件中存在Mysql类，Mysql类中有两个方法conn()与sucessConn()；conn()是不知道sql\_conn.txt文件是否存在时调用的函数，sucessConn()已知sql\_conn.txt存在时调用的函数。代码如下：

/\*$hostname:主机名

$dbname:数据库名

$username:数据库用户名

$pwd:数据库链接密码\*/

class Mysql{

/\*数据库链接测试函数，不存在sql\_conn.txt文件时使用\*/

public function conn($hostname,$dbname,$username,$pwd){

$mysql\_conf = array(

'host' => $hostname,

'db' => $dbname,

'db\_user' => $username,

'db\_pwd' => $pwd,

);

$mysql\_conn = @mysql\_connect($mysql\_conf['host'], $mysql\_conf['db\_user'], $mysql\_conf['db\_pwd']);

if (!$mysql\_conn) {

die("could not connect to the host:\n" . mysql\_error());//诊断连接错误}

mysql\_query("set names 'utf8'");//编码转化

$select\_db = mysql\_select\_db($mysql\_conf['db']);

if (!$select\_db) {die("could not connect to the database,please create a new database:\n" . mysql\_error());

}

return $mysql\_conn;

}

/\*数据库链接函数，存在sql\_conn.txt文件时使用

preg\_match("/\s/", $str):检查字符串是否有隐藏空格

trim($arr):去掉字符串两端空格\*/

public function sucessConn(){

$sql\_conn = dirname(dirname(\_\_FILE\_\_))."/sql\_conn.txt";

//echo $sql\_conn;

if(file\_exists($sql\_conn)){

include\_once "filecontrol.php";

$fobj = new FileControl();

$arr = $fobj->freader($sql\_conn);

$arr = array\_filter($arr);

$hostname = trim($arr[0]);

$dbname = trim($arr[1]);

$username = trim($arr[2]);

$pwd = trim($arr[3]);

$mysql\_conn=$this>conn($hostname,$dbname,$username,$pwd);

return $mysql\_conn;

}else{

die("the sql\_conn file is not exists");

} }}

### 创建链接文件和基础数据表

当检测程序没有检测到数据库链接信息存储文件（sql\_conn.txt）时，将跳转到数据收集页面（web\_conn\_sql.html）收集数据库链接信息，测试信息正确性，创建数据库链接信息存储文件（sql\_conn.txt）并在连接成功的数据库创建网站基本数据表。部分重要代码示例如下：

$hostname = $\_POST['hostname']; //获取通过post方法提交的数据

/\*创造数据库信息文件函数

$sql\_conn:数据库信息文件地址

fopen():打开并创建文件

fwrite()：将信息写入文件\*/

public function fcrate($hostname,$dbname,$username,$pwd){

$dpath = dirname(dirname(\_\_FILE\_\_));

$sql\_conn = $dpath."/sql\_conn.txt";

echo $sql\_conn;

if(file\_exists($sql\_conn)){

unlink($sql\_conn);

fcrate($hostname,$dbname,$username,$pwd);

}else{

$myfile = fopen($sql\_conn, "w+") or die("Unable to open file!");

$txt = $hostname."\r\n";

fwrite($myfile, $txt);

$txt = $dbname."\r\n";

fwrite($myfile, $txt);

$txt = $username."\r\n";

fwrite($myfile, $txt);

$txt = $pwd."\r\n";

fwrite($myfile, $txt);

fclose($myfile);

} return $sql\_conn; }

因为创建表的模式都一样，下面列举一个例子，其中check\_table\_is\_exist()f是检测表存在 性的方法。

/\*创建前端导航基本信息表\*/

$nav\_sql = "CREATE TABLE IF NOT EXISTS `navigationbar` (

`tid` int(2) NOT NULL AUTO\_INCREMENT,

`father\_id` int(2) NOT NULL COMMENT '父导航id',

`tname` varchar(255) NOT NULL COMMENT '名称',

`turl` varchar(255) NOT NULL COMMENT '链接地址',

PRIMARY KEY (`tid`)

)";

if((int)$this->check\_table\_is\_exist('navigationbar',$mysql\_conn)==0){

$nav\_res = mysql\_query($nav\_sql,$mysql\_conn);

if(!$nav\_res){

die(json\_encode(array('code'=>0,'msg'=>'couldn not get the res:'.mysql\_error())));

}}

### 基本信息的录入

在数据表和链接信息文件都已经创建好以后，网站安装系统将执行最后录入网站信息。网站信息包括网站基础信息（后台列表，网站基本信息等）和站长信息（密码，权限等），其中涉及数据库插入操作，主要代码示例如下：

$utime = date('Y-m-d', time()); //格式化获取当前系统时间

/\*act\_manage表插入操作

$uname:用户名

$upwd:密码

$utime:注册时间

$uemail:电子邮箱\*/

public function act\_insert($uname,$upwd,$utime,$uemail){

$str = "INSERT INTO `act\_manage` ( `uname` , `upwd` , `utime` , `uemail` )

VALUES ('".$uname."', '".$upwd."', '".$utime."', '".$uemail."');";

$re = mysql\_query($str,$this->mysql\_conn);

if (!$re) {die("couldn't get the res:\n" . mysql\_error());}

mysql\_close($this->mysql\_conn);

return $re; }

## 网站登录检测系统与注册系统的具体实现

登录检测系统是根据网站访问者结构设计的，我的主要设想是根据访问者不同展示不同的功能页面，所以其中一个子系统为访问者权限检测子系统。当然，第一次访问网站肯定都是一样的，为了区分我又设计了登录子系统和注册子系统。

### 检测系统的实现

检测系统的实现主要依赖于cookie而获取不同的内容，所以此子系统主要涉及cookie的获取 操作和数据获取操作。主要代码示例如下：

//cookic查询函数

function getCookie(name){

var arr,reg=new RegExp("(^| )"+name+"=([^;]\*)(;|$)");

if(arr=document.cookie.match(reg))

return unescape(arr[2]);

else

return false;

}

/\*navigationbar表查寻操作\*/

public function navigationbar\_queryAll(){

$str = "SELECT \*FROM `navigationbar`";

$re = mysql\_query($str,$this->mysql\_conn);

if (!$re) { die("couldn't get the res:\n" . mysql\_error());}

$res\_arr = array();

while ($row = mysql\_fetch\_assoc($re,MYSQL\_ASSOC))

{array\_push($res\_arr, $row); }

mysql\_close($this->mysql\_conn);

return $res\_arr;

}

### 登录系统的实现

当首访问网站时，所有访问者都是游客模式，想要获取更多功能和内容就得依赖于录入权限cookie，而录入cookie则由登录系统实现。在登录系统中主要的功能点是验证码操作程序（包括验证码生成，刷新，匹配检验）和用户检验程序及创建cookie。主要示例代码如下：

//随机验证码函数

function createVerificationCode(){

code = '';//首先默认code为空字符串

var codeLength = 4;//设置长度，这里看需求，我这里设置了4

var random = new Array(0,1,2,3,4,5,6,7,8,9,'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R', 'S','T','U','V','W','X','Y','Z');//设置随机字符

for(var i = 0; i < codeLength; i++){ //循环4次

var index = Math.floor(Math.random()\*36); //设置随机数范围,这设置为0 ~ 36

code += random[index]; //字符串拼接 将每次随机的字符 进行拼接}

$('#showVerificationCode').html(code);}

//刷新验证码函数

$('#refresh').click(function () {

createVerificationCode();

});

//验证码验证函数

function checkVerificationCode(){

var oValue = $("#verificationCode").val().toUpperCase();

if(oValue ==""){

alert('请输入验证码'); return 0;

}else if(oValue != code){

alert('验证码不正确，请重新输入');

oValue = "";

$("#verificationCode").val('');

createVerificationCode();

return 0;

}else{return 1; }}

//用户检测和cookie创建

header('Content-Type:text/html;Charset=utf-8');

$uname = $\_POST['uname']; //获取用户信息

setcookie("url","../../admin/view/index.html",time()+36000,"/");//后台地址cookie

setcookie("unameid",$act\_res->getUid(),time()+36000,"/");//用户cookie

### 注册系统的实现

## 网站后台管理系统的实现

根据网站主体数据模型而建立的后台管理系统主要作用是对各种数据的增删改查和网站的各种功能控制，它的主体分为用户管理，功能管理，网站管理，和站长信息4个模块。这些模块管理的内容不同，但功能基本相似。所以我将按照功能来叙述怎样实现后台管理系统。

### 数据展示

数据查询程序在前文已经有叙述，在这里我将解决如何实现使用分页技术展示大量数据。在设想中，通过查询程序返回回来的数据是不具备规律性的，很难实现分页，所以在返回数据到展示层之前，我做了数据处理，代码如下：

$fun = new connFun;

$fun\_res =$fun->fun\_queryAll(); //获取数据

$arr = [];

for($i=0; $i<count($fun\_res);$i++ ){ //数据重构

$j = (int)$i/10+1;

$arr[$j][$i%10] = $fun\_res[$i];

}

$arr = json\_encode($arr);

print\_r($arr); //展示数据接口

重构后的数据通过特定接口展示，那么我只需通过调用这个接口就能得到我想要的数据。数据已经得到，接下来是分页按钮的设置。在分页按钮中主要的难点是当前页码的判断，这一点我通过设置隐藏div解决，主要代码如下：

//翻页控制函数

function pageContro(status){

var page =parseInt($("#hiddenpage").html());

if (status) {

if (page+1 > maxpage) {alert('已达最大页');

} else{setData(page+1); }

} else{if (page-1 < 1) {

alert('已达最小页');

} else{setData(page-1); }}}

### 数据修改

数据修改程序大多都是对数据表的修改，所以需要一个唯一标识符来查找修改那一类数据，主要示例代码如下：

include\_once "../../model/sqlfunctions.php"; //函数调用

$fun = new connFun;

$fun\_res =$fun->fun\_update($fid, $fpid, $fname, $furl, $fgrade, $fstatus);

/\*function\_table表修改操作

$fid:功能编号

$fpid:父功能id

$fname:功能名称

$furl:功能实现地址

fgrade:功能层级

fstatus:状态\*/

public function fun\_update($fid, $fpid, $fname, $furl, $fgrade, $fstatus){

$str = "UPDATE `function\_table` SET `fpid` =".$fpid.",

`fname` = '".$fname."',

`furl` = '".$furl."',

`fgrade` =".$fgrade.",

`fstatus` =".$fstatus." WHERE fid =".$fid.";";

$re = mysql\_query($str,$this->mysql\_conn);

if (!$re) {

die("couldn't get the res:\n" . mysql\_error());

}

mysql\_close($this->mysql\_conn);

return $re;

}

### 添加数据

添加数据时程序通过post方法向数据操作层提交数据，数据操作层获取并验证数据后可通过调用模型方法更改数据，这些前文已有叙述，不再累述；在但有一些特殊的关联数据需要做特殊处理，主要示例代码如下：

//权限更改

$fun2 = new connFun;

$fun\_res2 =$fun2->fun\_queryAll();

$Fid\_having ='';

for($i=0; $i<count($fun\_res2);$i++ ){

$strnum = strval($fun\_res2[$i]['fid']);

$Fid\_having .= $strnum.'|';

}

$Fid\_having = substr($Fid\_having, 0, -1);

$fun3 = new connFun;

$aut\_res =$fun3->authority\_update(1, 1, $Fid\_having);

### 删除数据

删除数据时需要一个唯一标识符来查找删除那一类数据，这是一个很危险的操作，所以我设计了提示框来确认用户是否错误操作。同样的当删除数据时一些特殊的关联数据需要做特殊处理，主要示例代码如下：

/\*act\_manage表删除操作 $uid:用户编号\*/

public function act\_delete($uid){

$str = "DELETE FROM `act\_manage` WHERE uid =".$uid;

$re = mysql\_query($str,$this->mysql\_conn);

if (!$re) {

die("couldn't get the res:\n" . mysql\_error());

}

mysql\_close($this->mysql\_conn);

return $re; }

/\*数据模型调用与关联数据处理 \*/

include\_once "../../model/sqlfunctions.php";

$fun = new connFun;

$fun\_res =$fun->fun\_delete($fid);

//权限更改

$fun2 = new connFun;

$fun\_res2 =$fun2->fun\_queryAll();

$Fid\_having ='';

for($i=0; $i<count($fun\_res2);$i++ ){

$strnum = strval($fun\_res2[$i]['fid']);

$Fid\_having .= $strnum.'|';

}

$Fid\_having = substr($Fid\_having, 0, -1);

$fun3 = new connFun;

$aut\_res =$fun3->authority\_update(1, 1, $Fid\_having);

### 访问保护

网站后台管理系统是整个系统中很重要的一部分。所以为了避免未经过登录的过程而访问网站后台，我设计了cookie保护和访问来源检测函数，cookie保护是指只有通过相应的cookie，访问者才能使用相应功能，没有经过登录的过程就无法录入相应cookie，相应内容前文已有叙述。访问来源检测函数代码如下：

//访问来源检测函数

function checkReferrer(){

var fromurl = document.referrer;

if (fromurl=='') {

window.location.href ="http://www.newweb.com/index/view/dologin.html"

}

}

# 系统扩展的设计与实现

在系统主体设计实现中我已经实现了网站的基本 架构，但一个完整的网站还需要内容来充实这个网站，所以我设想以扩展模块的形式向网站嵌入内容，这样做的好处是不仅仅易于降低管理网站内容的难度，还能降低网站内容的耦合性，使网站结构清晰明了。

## 系统扩展

系统扩展是网站可扩展接口，是为了满足可扩展性需求而设计实现的模块。

### 系统扩展的设计

系统扩展依赖于功能扩展。当添加一个功能时，系统会根据功能连接找到扩展文件包并寻找扩展文件包中是否存在数据库操作文件，如果存在则表示需要添加/删除数据库，调用添加/删除函数，返回结果；如果不存在则直接返回结果。



图8.1.1扩展功能流程图

在扩展的功能文件中的一些文件需要遵循特别的 命名规则来满足扩展功能的需求。比如文件名必须和模块控制主页的名称相同，数据操作文件必须是模块控制主页的名称加上\_sql后缀，数据操作文件中的类名必须为Sqlcontra，创建数据表函名必须为createTable()，删除数据表函数名必须为deleteTable()。



图8.1.1扩展文件结构图

### 系统扩展的实现

在系统扩展设计时，我的想法是扩展的功能以一个文件夹的形式添加到预留的扩展文件夹Extra中。所以实现这部分我使用到了很多关于文件夹的操作代码，部分代码示例如下：

/\*createMod():数据库操作文件检测函数

$url：模块控制主页地址\*/

public function createMod($url)

{

$dpath = dirname(dirname($url)); //获取添加的模块地址

if(!file\_exists($dpath)){ //模块检测

die("该模块不存在");

}

$array=explode('/', $url); //字符串切割

$a = $array[count($array)-1];

$name=explode('.', $a )[0];

$dpath = $dpath."/".$name."\_sql.php"; //地址拼接

if(file\_exists($dpath)){ //数据操控文件存在性检测

include\_once $dpath;

$obj = new Sqlcontra();

$status = $obj->createTable(); //数据表创建函数调用

if($status){

return true;

}else{

return false;

}

}else{

return true;

}

}

## 日志模块

日志模块是我设计的记录日常生活信息的模块，这个模块其实也就是一个在线日记簿，功能虽然简单，但对于我来说却是一个非常重要的模块。

### 数据模型

日志模块的数据模型属性主要有编号，时间属性用于查找排序，标题概括内容，内容记录当日事件。



图8.2.1日志表E-R图

### 数据表

表8.1日志表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **空** | **默认** | **说明** |
| lid | int(11) | 否 |  | 编号 |
| ltime | date | 否 |  | 时间 |
| ltitle | varchar(255) | 是 | NULL | 标题 |
| ltext | text | 是 | NULL | 内容 |

### 代码实现

### 最终结果

# 总结

# 参考文献

[1] (美)普雷斯曼 . Software Engineering A Practitioner’s Approach软件工程：实践者的研究方法（英文版第8版） [M]. 北京：机械工业出版社. 2015年2月

[2] 张鑫.PHP+MySQL开发实战[M]. 北京：清华大学出版社. 2013年

[3] 韩超，梁泉著. Android系统原理及开发要点详解 [M]. 北京：电子工业出版社. 2017.

[4] 张利国,代闻,龚海平. Android移动开放案例详解[M]. 北京：人民邮电出版社. 2018.

[5] (印)阿南德•巴拉钱德拉•皮莱.软件架构：Python语言实现[M]．北京:机械工业出版社. 2018.03

[6] 何敏煌. Python新手使用Django架站的16堂课[M]．北京：清华大学出版社. 2017.5

[7] 王友钊, 黄静编. Django开发宝典[M]. 北京：清华大学出版社. 2017.

[8] 吴艳. Java Web开发基础与案例教程[M].北京：机械工业出版社.2016.

[9] 陈光辉，黎连业，王萍等.网络综合布线系统与施工技术（第五版）[M].北京：机械工业出版社. 2018.