



电子科技大学

University of Electronic Science and Technology of China

学 士 学 位 论 文

BACHELOR DISSERTATION

论文题目 团队信息网络管理系统的设计与实现

学生姓名 曾路洋

学 号 2010013060009

专 业 通信工程

学 院 通信与信息工程学院

指导教师 马立香副教授

指导单位 电子科技大学

2014年5月15日

摘 要

PHP 中文摘要

关键词：信息管理系统，B/S架构，MVC架构，动态网站，MySQL

ABSTRACT

PHP English Abstract.

Keywords: Informtaion Management System, B/S Framework, MVCFramework, Dynamic Website, MySQL

目 录

第1章 需求分析	1
1.1 总体需求	1
1.2 论文管理模块	2
1.2.1 数据表项	2
1.2.2 录入	3
1.2.3 修改与删除	4
1.2.4 显示	4
1.2.5 查询与导出	4
1.3 科研项目管理模块	5
1.3.1 数据表项	5
1.3.2 录入	6
1.3.3 显示	7
1.4 其它数据管理模块	7
1.4.1 专利管理模块	7
1.4.2 人员管理模块	7
1.5 权限管理模块	8
1.5.1 数据表项	8
1.5.2 权限的分级配置	8
1.5.3 用户管理	8
第2章 方案设计	9
2.1 B/S结构与AJAX技术	9
2.1.1 传统C/S结构的缺点	9
2.1.2 B/S结构的优点	9
2.1.3 AJAX技术	10
2.1.4 小结	10
2.2 LAMP	11
2.2.1 服务器操作系统 Linux	11
2.2.2 网页服务器 Apache	11

2.2.3 数据库管理系统 MySQL	12
2.2.4 脚本语言 PHP	12
2.2.4.1 PHP框架	12
2.2.4.2 Yii框架	13
2.3 MVC设计模式	13
2.4 单次请求时序	14
2.5 本章小结	15
第3章 方案实现	16
3.1 开发工具	16
3.1.1 开发服务器搭建	16
3.1.2 版本控制工具	17
3.2 数据库表实现	17
3.2.1 数据库引擎选择	17
3.2.2 数据库字符集选择	18
3.2.3 数据库表项实现	18
3.2.3.1 论文	19
3.2.3.2 科研项目	21
3.2.3.3 专利	23
3.2.3.4 人员	24
3.2.4 数据表关系	24
3.3 模型实现	26
3.3.1 建立数据库连接	27
3.3.2 定义AR类	27
3.3.3 实现增删操作	28
3.3.4 定义数据库表关系	29
3.3.5 验证数据有效性	31
3.3.6 搜索与筛选	32
参考文献	35
致 谢	36

第1章 需求分析

1.1 总体需求

为科研团队开发、设计一个“团队信息网络管理系统”，目标是建立及时、准确、全面的科研团队信息管理平台。通过与系统使用者进行沟通与调研，完成了本系统的需求分析。

本系统的目的针对科研团队信息管理的实际情况，全面覆盖科研团队在论文、科研项目、专利、人员管理等多个方面，并提供对这些数据的搜索、筛选以及分析功能，提高科研团队的工作以及沟通效率，并且为科研团队的决策、考核提供有力的支撑。同时，本系统需要利用这些数据，将论文、科研项目、专利等作为内容对外展示，方便其他人对团队进行了解和交流。根据以上需求，可将本系统划分为前台模块和后台模块。

前台模块主要用于对外展示，主要针对角色为游客的使用者进行设计，主要包括首页和展示团队论文、科研项目、专利、人员情况的子页面。其中，根据展示内容的特点，需要以不同形式进行展示。

后台模块用于科研团队内部对各项数据进行管理，主要针对角色为子模块维护者、系统管理员进行设计。根据以上需求，后台模块又可以划分为数据管理模块和权限控制模块。其中，又可根据需要管理的数据分为论文管理、项目管理、专利管理、人员管理这四个子模块。

对于每个数据管理模块，需要实现以下功能：

1. 对各子模块的数据进行录入、删除、修改、查询。
2. 对各子模块的数据进行多条件搜索、筛选。
3. 实现对各子模块的分级权限管理。
4. 对各子模块的数据进行格式化的导入和导出。

对于权限控制模块，需要实现以下功能：

1. 对用户进行管理，实现管理员的建立和其它用户的注册、删除、修改功能
2. 划分用户的权限级别，对某个用户是否能够操作某个数据管理模块进行认证和鉴权

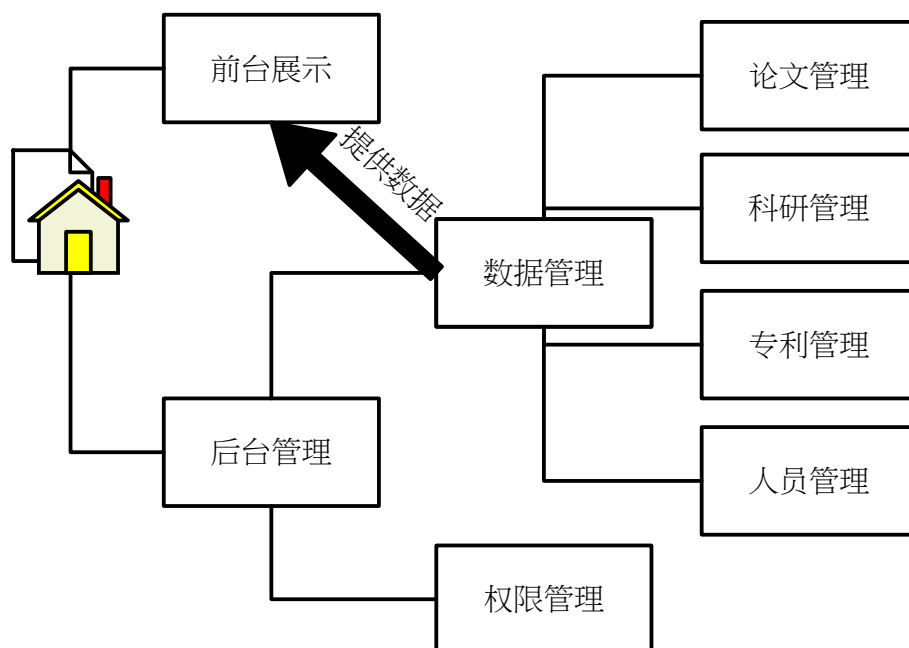


图 1-1 系统总体框图

本系统的系统总体框图如图 1-1 所示：

限于篇幅所限，对于四个数据管理子模块，本文仅选取数据表项最为复杂、功能最多的论文管理模块和科研项目管理模块进行详细介绍；专利管理模块以及人员管理模块数据表项与功能相对简单，且在功能上与论文管理模块和科研项目管理模块相类似，在本文中，仅给出它们的数据表项。

1.2 论文管理模块

通过将科研团队已发表和拟发表的论文录入到本系统中进行管理，可以及时地跟踪论文的发表状态，准确地根据作者、时间段、类别、支撑项目等不同条件搜索、筛选以及导出特定的论文，供科研团队里面的老师和学生进行使用，为团队的决策、考核提供有力的数据支撑。

论文管理模块主要要实现以下几个功能。

1.2.1 数据表项

需要设计一个数据库表，存储以下论文数据：

1. 论文信息，如：“Xiaoyan Huang, Yuming Mao, Fan Wu, “Low Complexity Utility-based Scheduling Algorithm for Heterogeneous Services in OFDM Wireless Networks,” In: Proc. of ICCAS 2009, San Jose, USA, vol 1, pp.48-52.”
2. 作者：按顺序存储第一作者至第五作者
3. 状态：录用待发、已发表或已检索
4. 时间：录用时间、发表时间和检索时间
5. 检索类型：SCI、EI和ISTP的检索号
6. 论文级别：一级、核心、其他刊物、期刊、会议、国际、高水平
7. 论文文件：存储论文对应的PDF或者Microsoft Word文档
8. 支柱项目：数个支柱项目
9. 报账项目：数个报账项目

1.2.2 录入

分别需要实现对论文数据的批量录入和逐个录入。

批量录入指用户通过上传指定格式的Microsoft Excel表格，批量地将论文数据导入到数据库中。在批量录入中，需要实现替换已有表项和对已有表项进行添加的功能。

逐个录入：通过一个表单，提示用户输入或者选择以下内容：

1. 论文维护人员：从人员模块获取人员的姓名，生成下拉菜单进行选择，也可以通过输入拼音首字母快速选择
2. 论文信息
3. 作者1——作者5：与选择论文维护人员时相同
4. 状态：在“录用待发”、“已发表”、“已检索”中三选一
5. 时间：
 - 若状态为录用待发，则显示输入“录用时间”的表单
 - 若状态为已发表，则显示输入“发表时间”的表单
 - 若状态为已检索，则显示输入“发表时间”和“检索时间”的表单
6. 检索类型：仅当状态为已检索时，需要用户输入，提示用户输入SCI、EI或ISTP的检索号
7. 发表级别：在会议、期刊、国际、一级、核心、其他期刊中勾选一项或者多项

8. 支柱项目：在科研项目的列表中选择，也可以通过拼音首字母搜索
9. 报账项目：与选择支柱项目时相同
10. 高水平：在“是”、“否”中二选一
11. 论文文件：提示用户选择论文对应的pdf文件或者Microsoft Word文件进行上传

1.2.3 修改与删除

对已有表项进行修改与删除，能够先按条件搜索到需要修改或删除的论文，然后对相关项修改或删除。论文修改功能的用户界面与流程与论文录入的用户界面与流程一致，只是表单各条目需要显示已有数据。论文删除功能需要实现将选中的论文条目从存储论文的数据库表中删除。

1.2.4 显示

论文的显示功能需要分为对内显示和对外显示。

对外显示：仅选择类型为高水平的论文，具体功能如下：

1. 排序原则：最新日期置顶
2. 显示内容：数据表项中的论文信息
3. 显示方式：表格分页形式

对内显示：需要显示所有的论文，显示格式与对外显示一致。另外，需要在页面上醒目位置设置“查询”按钮，通过点击“查询”按钮，进入查询和导出功能。

1.2.5 查询与导出

需要可接单条件和组合条件查询，以表格形式显示并以Excel表格形式导出（显示和导出内容相同）。单条件包括：维护者、作者、状态、时间、检索类型、发表级别、支柱项目、报账项目。组合条件即是以上2个及以上条件的组合。

由于篇幅所限，本文在这里仅给出“按维护者查询”、“按作者查询”和“按时间段查询”的需求，其它条件、以及多条件查询的需求类似在这里给出的三种需求。

按维护者查询：排序按时间，最新时间置顶（录用、发表时间统一考虑），如时间相同，则以已检索、已发表、已录用再排。显示以及导出的格式如表 1-1 所

示,其中:时间一项要求只录用的为录用时间,其它显示为发表时间;检索一项需要包含检索类型和检索号。导出的Excel文件名需要命名为“由XXX维护的论文”。

表 1-1 论文按维护者查询导出格式

序号	论文信息	状态	时间	检索
1	Xiaoyan Huang, Yuming Mao...	已检索	2011.01.01	EI:20111713930271
2	黄晓燕, 毛玉明, 吴凡, 冷甦鹏., “基于...	已发表	2009.05.01	

按作者查询:排序按时间,最新时间置顶(录用、发表时间统一考虑),如时间相同,则以已检索、已发表、已录用再排。导出的Excel文件名需要命名为“XXX发表的论文”。

按时间段查询:如2000年之后、2012年9月至2013年6月等,排序按时间,最新时间置顶(录用、发表时间统一考虑),如时间相同,则以已检索、已发表、已录用再排。导出的Excel文件名需要命名为“2000年至2013年之间发表的论文”。

1.3 科研项目管理模块

与论文管理模块类似,通过将科研项目信息录入到本系统,可以及时地跟踪各个项目的当前状态和信息,按照人员、年份、级别等不同条件对科研项目进行搜索、筛选和显示,供科研团队里面的老师和学生进行使用,为团队的决策、考核提供有力的数据支撑。

1.3.1 数据表项

需要设计一个数据库表,存储以下信息:

1. 维护人员
2. 项目名称
3. 项目编号
4. 经本费编号
5. 级别:国际级、国家级、省部级、市级、校级、横向、国家自然科学基金、973、863、科技支撑计划、教育部高校博士点基金、重大专项和XX项目中的一项或者多项

6. 开始时间
7. 截止时间
8. 结题时间
9. 申报时间
10. 立项时间
11. 申报经费
12. 立项经费
13. 实际执行人员
14. 责任书人员：数个责任书人员

1.3.2 录入

同论文管理模块类似，分别需要实现对论文数据的批量录入和逐个录入。

批量录入指用户通过上传指定格式的Microsoft Excel表格，批量地将论文数据导入到数据库中。在批量录入中，需要实现替换已有表项和对已有表项进行添加的功能。

在逐个录入中，通过一个表单，提示用户输入或者选择以下内容：

1. 维护人员：从人员模块获取人员的数据，生成下拉菜单进行选择，也可以通过输入拼音首字母快速选择
2. 项目名称
3. 项目编号
4. 经本费编号
5. 级别：在国际级、国家级、省部级、市级、校级、横向、国家自然科学基金、973、863、科技支撑计划、教育部高校博士点基金、重大专项和XX项目中勾选一项或者多项
6. 时间：开始时间、截止时间、结题时间、申报时间、立项时间
7. 经费：申报经费、立项经费
8. 实际执行人员：在人员的列表中进行多项选择，也可以通过输入拼音首字母快速选择
9. 责任书人员：与实际执行人员的录入类似

1.3.3 显示

对外显示：要求校级项目和XX项目不显示，申报项目不显示。排序原则：项目级别和类型按时间排序（最新截至时间置顶），级别类型顺序如下级别类型从高到低：国际合作，国家自然科学基金，国家973，国家863，国家科技支持计划，重大专项，省部级，市级，横向。显示格式需要以表格的形式进行显示。

对内显示：研究项目分为责任版和执行版（主要是人员可能有差异）两个菜单可选。选择版本后，其显示栏目完全相同。栏目内容都显示。级别类型中，将XX项目放在最后。默认显示所有数据表项。对于申报项目，默认显示的数据表项：项目名称、级别、类型、申报时间、立项时间、人员。查询时置顶按相关条件。如按人员查询，则按其排名先后和时间后先（排名，最新时间），等等。

1.4 其它数据管理模块

1.4.1 专利管理模块

专利管理模块的数据表项：

1. 维护人
2. 专利名称
3. 申请时间
4. 申请号
5. 授权时间
6. 授权号
7. 级别：国际或国内二选一
8. 类型：发明专利或实用新型专利二选一
9. 发明人：数个（少于五个）发明人
10. 报账项目

1.4.2 人员管理模块

1. 中文名
2. 英文名
3. 类型：学生或教师二选一
4. 介绍

1.5 权限管理模块

1.5.1 数据表项

需要设计一个数据库表，存储以下权限管理数据

1. 用户名
2. 密码散列值
3. 权限等级：系统超级管理员、论文模块管理员、科研项目模块管理员、专利管理员或人员管理员中的一个或者多个
4. 电子邮箱：用于忘记密码后的找回

1.5.2 权限的分级配置

在系统第一次部署时，需要提示当前用户（系统的部署者）输入一个电子邮箱地址，生成一个用户名为“admin”密码为随机生成的用户作为系统超级管理员，并将这些信息显示和发送给输入的电子邮箱地址。

有了系统超级管理员之后，管理员可以按需的查看、增加、删除、修改用户，并且为这些用户配置各个数据管理模块的管理权限。

1.5.3 用户管理

查看用户：按表格的形式列出当前系统所有用户，在每个用户条目位置显示增加、删除、修改的链接。

增加、修改用户：通过一个表单，提示管理员输入或选择以下内容：

1. 用户名：要求只能是英文字母和下划线的组合
2. 密码和重复密码：若密码和重复密码相符合，将密码的散列值存储在数据库中，不能存储明文密码，以免泄漏用户隐私
3. 权限等级：在系统超级管理员、论文模块管理员、科研项目模块管理员、专利管理员或人员管理员中的一个或者多个
4. 电子邮箱

第2章 方案设计

2.1 B/S结构与AJAX技术

B/S (Browser/Server) 结构^[1]即浏览器和服务器结构。它是随着Internet技术的兴起,对C/S结构的一种变化或者改进的结构。在这种结构下,用户工作界面是通过WWW浏览器来实现,极少部分事务逻辑在前端 (Browser) 实现,但是主要事务逻辑在服务器端 (Server) 实现,形成所谓三层结构。

2.1.1 传统C/S结构的缺点

C/S (Client/Server) 结构,即大家熟知的客户机和服务器结构。它是软件系统体系结构,通过它可以充分利用两端硬件环境的优势,将任务合理分配到Client端和Server端来实现,降低了系统的通讯开销。

传统的C/S体系结构虽然采用的是开放模式,但这只是系统开发一级的开放性,在特定的应用中无论是Client端还是Server端都还需要特定的软件支持。由于没能提供用户真正期望的开放环境,C/S结构的软件需要针对不同的操作系统系统开发不同版本的软件,加之产品的更新换代十分快,已经很难适应百台电脑以上局域网用户同时使用。而且代价高,效率低。

其次,采用C/S架构,网络管理工作人员既要服务器维护管理,又要对客户端维护和管理,这需要高昂的投资和复杂的技术支持,维护成本很高,维护任务量大。

2.1.2 B/S结构的优点

与C/S结构相比较,B/S结构的优点主要有跨平台性和易维护性^[2]。

以目前的技术看,局域网建立B/S结构的网络应用,并通过Internet/Intranet模式下数据库应用,相对易于把握、成本也是较低的。它是一次性到位的开发,能实现不同的人员,从不同的地点,以不同的接入方式 (比如LAN、WLAN、WAN,Internet/Intranet等)、不同的终端 (PC机、智能手机以及平板电脑等)、不同的操作系统 (Windows、Linux、Mac OS X、iOS、Android等) 访问和操作共同的

数据库；它能有效地保护数据平台和管理访问权限，服务器数据库也很安全。特别是在JAVA、PHP这样的跨平台语言出现之后，B/S架构管理软件更是方便、快捷、高效。

另外，软件系统的改进和升级越来越频繁，B/S架构的产品明显体现着更为方便的特性。对一个稍微大一点单位来说，系统管理人员如果需要在几百甚至上千部电脑之间来回奔跑，效率和工作量是可想而知的，但B/S架构的软件只需要管理服务器就行了，所有的客户端只是浏览器，根本不需要做任何维护。无论用户的规模有多大，有多少分支机构都不会增加任何维护升级的工作量，所有的操作只需要针对服务器进行。

2.1.3 AJAX技术

AJAX即“Asynchronous JavaScript and XML”（异步的JavaScript与XML技术），指的是一套综合了多项技术的浏览器端网页开发技术。Ajax的概念由Jesse James Garrett所提出^[3]。

传统的Web应用允许用户端填写表单（form），当提交表单时就向Web服务器发送一个请求。服务器接收并处理传来的表单，然后送回一个新的网页，但这个做法浪费了许多带宽，因为在前后两个页面中的大部分HTML码往往是相同的。由于每次应用的沟通都需要向服务器发送请求，应用的回应时间依赖于服务器的回应时间。这导致了用户界面的回应比本机应用慢得多。

与此不同，AJAX应用可以仅向服务器发送并取回必须的数据，并在客户端采用JavaScript处理来自服务器的回应。因为在服务器和浏览器之间交换的数据大量减少（大约只有原来的5%），服务器回应更快了。同时，很多的处理工作可以在发出请求的客户端机器上完成，因此Web服务器的负荷也减少了。

通过Ajax技术，采用B/S架构的程序也能在客户端电脑上进行部分处理和刷新，从而大大的减轻了服务器的负担；并增加了交互性，能进行局部实时刷新，能够达到和传统C/S结构相同的用户体验。

2.1.4 小结

综上所述，本系统选择采用基于AJAX技术的B/S结构，这样既能使本系统具有跨平台性和易维护性的优点，也能够提供和传统C/S结构相同的用户体验。

2.2 LAMP

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名称首字母缩写：

1. Linux, 操作系统
2. Apache, 网页服务器
3. MySQL, 数据库管理系统
4. PHP, 脚本语言

目前Internet上流行的网站构架方式是LAMP (Linux Apache MySQL PHP), 即是用Linux作为操作系统, Apache作为Web服务器, MySQL作为数据库, PHP (部分网站也使用Perl或Python) 作为服务器端脚本解释器。由于这四个软件都是开放源代码软件, 因此使用这种方式可以以较低的成本创建起一个稳定、免费的网站系统。LAMP软件可以说是当前最为流行的动态网页解决方案^[4]。因此, 在本系统中, 系统程序运行在Linux操作系统下, 由Apache HTTP服务器提供内容, 在MySQL数据库中存储内容, 利用PHP来实现程序逻辑。

2.2.1 服务器操作系统Linux

Linux作为本系统服务器的操作系统, 主要有开源免费、良好的生态系统、更好的性能几点优势。

Linux 内核源代码可以免费下载。大多数Linux 发布版本, 包括GNU/Linux 的发行版本和商业的发行版本几乎都提供免费下载服务。免费意味着零试用成本, 也不需要为安装在第二台机器上付费。

Linux 作为服务器的优势是, 他目前具有最好的生态系统, 服务器端的各种软件都为它而设计, 默认都认为你是在Linux上运行, Apache、MySQL等软件随能够在Windows下运行, 但是性能却显著地比在Linux下运行时低^[5]。

2.2.2 网页服务器Apache

Apache HTTP Server (简称Apache) 是Apache软件基金会的一个开放源代码的网页服务器, 可以在大多数计算机操作系统中运行, 由于其跨平台和安全性被广泛使用, 是最流行的Web服务器端软件之一^[6]。它快速、可靠并且可通过简单的API扩充, 将Perl、PHP、Python等解释器编译到服务器中。

2.2.3 数据库管理系统MySQL

MySQL是一个开放源代码的关系数据库管理系统（RDBMS），MySQL性能高、成本低、可靠性好，已经成为最流行的开源数据库，因此被广泛地应用在Internet上的中小型网站中。

2.2.4 脚本语言PHP

PHP（全称：PHP: Hypertext Preprocessor，即“PHP: 超文本预处理器”）是一种开源的通用计算机脚本语言，尤其适用于网络开发并可嵌入HTML中使用^[7]。PHP的语法借鉴吸收了C语言、Java和Perl等流行计算机语言的特点，易于一般程序员学习。PHP的主要目标是允许网络开发人员快速编写动态页面，但PHP也被用于其他很多领域。

2.2.4.1 PHP框架

PHP框架提供了一个用以构建web应用的基本框架，从而简化了用PHP编写web应用程序的流程。换言之，PHP框架有助于促进快速应用开发，不但节省开发时间、有助于建立更稳定的应用，而且减少了重复编码的开发。通过确保适当的数据库交换和在表现层编码，框架还可以帮助初学者建立更稳定的应用服务。这可以让你花更多的时间去创建实际的Web应用程序，而不是花时间写重复的代码。

PHP框架的作用相当于模型-视图-控制器（Model View Controller）。MVC是种编程的架构模式，将业务逻辑从UI中分离出来，允许一个一个单独修改（也称为关注点分离）。在MVC中，Model指数据，View指表现层，Controller则指应用程序或业务逻辑。基本上，MVC打破了一个应用的开发进程，这样各组件就可以不受影响地各自工作。从本质上讲，这使得用PHP编码更快更简单。

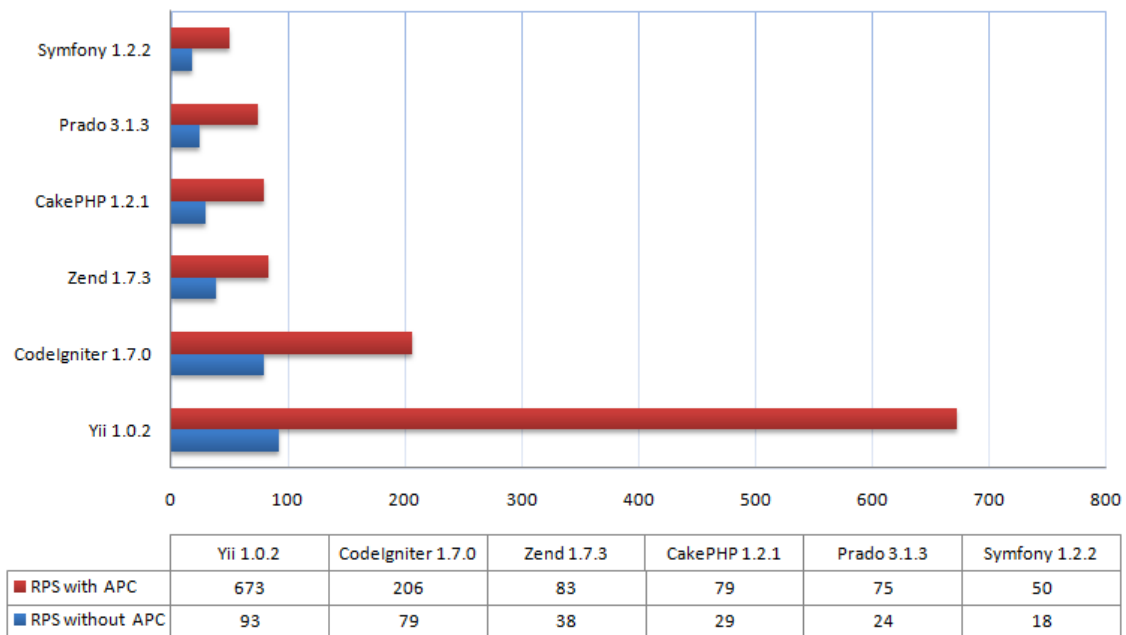


图 2-1 主流PHP框架性能比较

2.2.4.2 Yii框架

Yii框架是一个基于组件的高性能PHP 框架，用于快速开发大型Web 应用。

在图 2-1 中给出了与其他主流框架的性能比较，其中PPS（Requets Per Second）指每秒钟请求数，反映了使用不同框架编写同样功能程序每秒钟所能处理的请求数目，越高越好；红色部分是没有开启APC（Alternative PHP Cache，是一个开放自由的PHP opcode 缓存。它的目标是提供一个自由、开放，和健全的框架用于缓存和优化PHP的中间代码^[8]）的结果，蓝色部分是开启了APC缓存的结果。

因此，综合性能^[9]、文档完善程度^[10]、学习曲线三方面因素考虑，本系统选择使用了Yii框架进行开发。

2.3 MVC设计模式

MVC模式（Model-View-Controller）是软件工程中的一种软件架构模式，把软件系统分为三个基本部分：模型（Model）、视图（View）和控制器（Controller）。

MVC模式最早由Trygve Reenskaug在1978年提出^[11]。MVC模式的目的是实现一种动态的程序设计，使后续对程序的修改和扩展简化，并且使程序某一部分的

重复利用成为可能。除此之外，此模式通过对复杂度的简化，使程序结构更加直观：

1. 控制器Controller:负责转发请求，对请求进行处理。
2. 视图View:图形界面设计
3. 模型Model:实现程序应有的功能（实现算法等等）、实现数据管理和数据库设计(可以实现具体的功能)。

2.4 单次请求时序

如 2-2 所示，这是本系统使用MVC设计模式某一次从服务器收到客户端的HTTP请求到服务器返回给客户端HTTP响应的时序图。在这里以用户查看论文

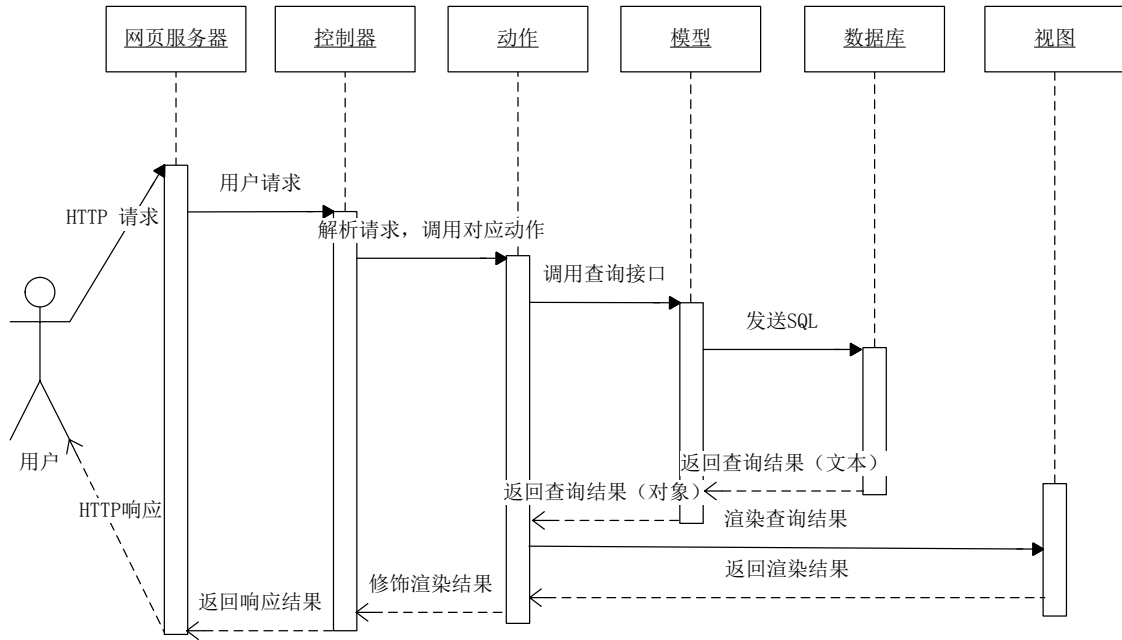


图 2-2 单次请求时序图

管理模块首页为例，分析系统的整个时序流程。

1. 用户点击“论文”超链接
2. 浏览器向URL为<http://domain.com/index.php?r=paper/index>发送HTTP请求
3. Apache服务器收到请求，调用mod_php模块，对index.php执行，并且传递查询参数（Query Parameter）：r=paper/index

4. index.php解析查询参数，调用控制器PaperController的动作actionIndex
5. 动作actionIndex调用模型Paper的查询接口findAll
6. 模型Paper的查询接口findAll通过TCP Socket，发送SQL语句：“SELECT * FROM TABLE_PAPER ” 到数据库
7. 数据库返回文本查询结果，模型Paper将这些查询结果解析并使用记过例化一个新的模型类返回给动作actionIndex
8. actionIndex根据这些结果，按照视图模版渲染生成HTML结果返回给控制器
9. 控制器在HTML结果的基础上进行修饰：添加头部和尾部，生成最终结果，返回给Apache
10. Apache最终结果通过HTTP响应返回给浏览器
11. 浏览器根据W3C的标准将HTTP响应绘制成响应的文字和图形，呈现给用户

2.5 本章小结

本章主要根据本系统的需求，介绍B/S结构、LAMP架构、Yii框架、MVC设计模式的主要特点，并分析了选择它们作为实现本系统方案的原因，最后给出了本系统单次请求的时序关系，解释了了B/S结构、LAMP架构、Yii框架、MVC设计模式是如何结合在本系统中的。

第3章 方案实现

3.1 开发工具

3.1.1 开发服务器搭建

在实现本系统之前，需要搭建出第 11 页的第 2.2 节提出的LAMP服务器环境，然后在此开发环境下进行开发与调试。首先安装 Ubuntu 12.04 LTS 操作系统，Ubuntu是一个以桌面应用为主的GNU/Linux操作系统。打开终端，输入`sudo apt-get install tasksel`，安装tasksel。tasksel是一个Debian下的安装任务套件，如果你为了使你的系统完成某一种常规功能，而需要安装多个软件包时，可以使用它进行方便快捷的安装。安装成功后打开tasksel，如 3-1 所示，选择LAMP Server，一个默认配置的LAMP服务器便搭建配置完毕了。此时，在浏览器中访

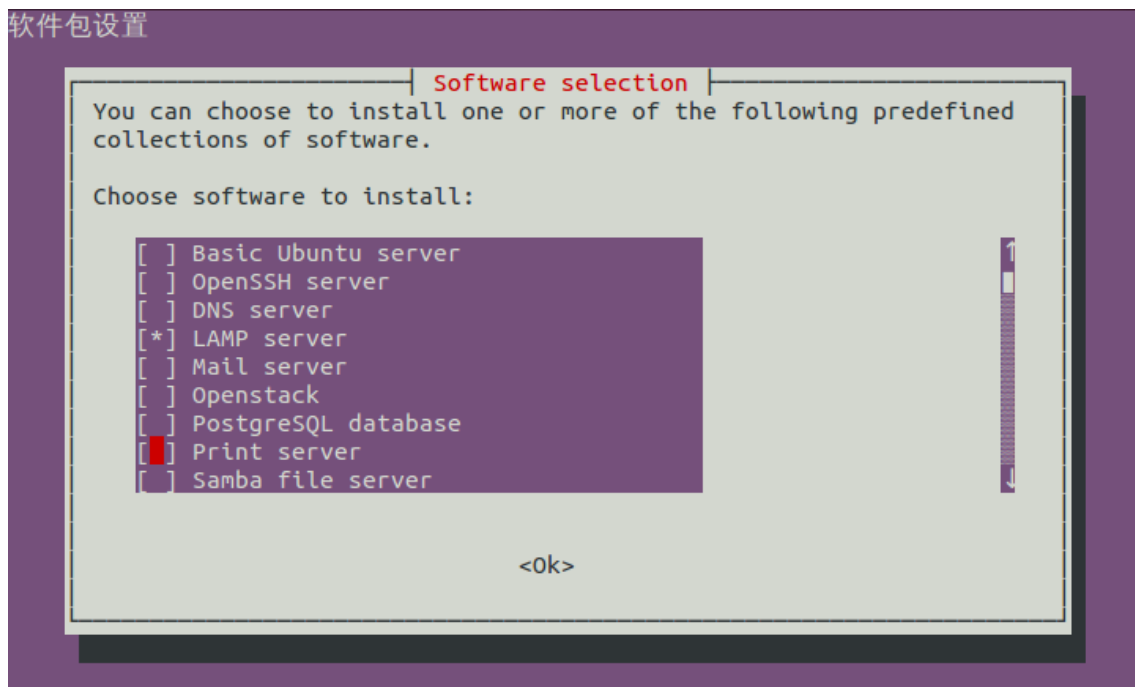


图 3-1 利用tasksel工具安装配置LAMP开发服务器

问 `http://localhost` 就能够打开一个标题为 “It works!” 的默认网页，说明开发服务器已经正常运行了。

3.1.2 版本控制工具

git是一个版本控制系统，用来保留工程源代码历史状态的命令行工具。可以利用它来追踪项目中的文件，并且得到某些时间点提交的项目状态。通过使用git，可以方便地在开发本系统的过程中，任意回溯到源码的不同版本上，提高开发效率，保证了源码库的安全。图 3-2 展示了在开发本系统过程中的部分git日志，其中每个条目代表对代码库的一次提交，可以在不同的提交之间回溯。

```
commit 2418a935a6a5ba928e55711e7057c548d7947e5f
Author: Zeng Luyang <zengluyang@gmail.com>
Date:   Fri May 16 12:51:45 2014 +0800

    MOD: change index page layout

commit 79d6ade8b89f6d5d1f1cf242401a85add8809a6b
Author: Zeng Luyang <zengluyang@gmail.com>
Date:   Fri May 16 12:20:49 2014 +0800

    MOD: search condition

commit ef62647eddf142adc3cefe98f343d5e31b0111fd
Author: Zeng Luyang <zengluyang@gmail.com>
Date:   Fri May 16 12:20:25 2014 +0800

    MOD: add a banner in template

commit 06f477d6b5b1a5cf3f6af93a9641ff74504fa2c2
Author: Zeng Luyang <zengluyang@gmail.com>
Date:   Fri May 16 09:39:37 2014 +0800

    MOD: implement for search options in Paper MVC
```

图 3-2 本系统开发过程中git的部分日志

3.2 数据库表实现

3.2.1 数据库引擎选择

InnoDB和MyISAM是MySQL中最常用的两个数据库引擎。MyISAM是MySQL关系数据库管理系统的默认储存引擎。这种MySQL表存储结构从旧的ISAM代码扩展出许多有用的功能。InnoDB是MySQL的另一个存储引擎，正成为目前MySQL AB所发行新版的标准，被包含在所有二进制安装包。较之于其它的存储引擎它的优点是它支持兼容ACID的事务（类似于PostgreSQL），以及参数完整性（对外键的约束）。在新版本的MySQL中，InnoDB引擎由于其对事务，参照完整性，以及更高的并发性等优点开始广泛的取代MyISAM。

考虑到在本系统中，需要在各个数据管理模块的数据库表中保存一个维护者字段，在科研项目项目管理模块中为每个科研项目记录保存不定数量的人员，在论文管理模块中为每个项目记录保存不定数量的作者，以及在专利管理模块中为每个专利记录保存不定数量的发明人，需要使用到外键和创建额外的关系数据表，详见第24页3.2.4节所介绍的数据表关系；考虑到InnoDB对与参数完整性的支持能够很好的保证本系统数据的一致性和完整性，本系统中的所有数据库表选用InnoDB引擎。在MySQL中指定数据库引擎非常简单，只需要在创建数据库表的SQL语句中指定“ENGINE=InnoDB”

3.2.2 数据库字符集选择

字符集是一套符号和编码的规则，不论是在Oracle数据库还是在MySQL数据库，都存在字符集的选择问题，而且如果在数据库创建阶段没有正确选择字符集，那么可能在后期需要更换字符集，而字符集的更换是代价比较高的操作，也存在一定的风险。

UTF-8（8-bit Unicode Transformation Format）是一种针对Unicode的可变长度字符编码，是用以解决国际上字符的一种多字节编码，它对英文使用8位（即一个字节），中文使用24位（三个字节）来编码。UTF-8包含全世界所有国家需要用到的字符，是国际编码，通用性强。

考虑到通用性和易用性，本系统中的所有数据库表选用UTF-8字符集。在MySQL中指定数据库字符集同样非常简单，只需要在创建数据库表的SQL语句中指定“DEFAULT CHARSET=utf8”

3.2.3 数据库表项实现

根据需求分析中提出的各数据管理模块需要实现存储的表项，在MySQL中分别创建了论文、专利、科研项目、人员四个数据库表。

3.2.3.1 论文

表 3-1 tbl_paper表的结构

字段名称	数据类型	说明
id	整型	主键，自动编号
info	文本	论文信息，不能为空
status	整形	状态
pass_date	日期	录用时间
pub_date	日期	发表时间
index_date	日期	检索时间
sci_number	变长字符串	SCI检索号
ei_number	变长字符串	EI检索号
istp_number	变长字符串	ISTP检索号
is_first_grade	布尔	是否一级
is_core	布尔	是否核心
is_journal	布尔	是否期刊
is_conference	布尔	是否会议
is_intl	布尔	是否国际
is_domestic	布尔	是否国内
file_name	变长字符串	论文文件名
file_type	变长字符串	论文文件类型
file_content	二进制数据	论文文件数据
is_high_level	布尔	是否高水平
maintainer_id	整形	维护人员id

表 3-1 给出了论文数据库表的结构，其中论文信息比较长，可能超过255个字符，因此采用MySQL中的“mediumtext”数据类型，其他变长字符串均采用“varchar(255)”数据类型，日期采用“date”数据类型，主键id与外键维护人员id采用“int(11)”数据类型，是否一级、是否核心等布尔字段均采用“tinyint(1)”型数据类型。综合上述各字段数据类型的选择，使用下面的SQL语句，在MySQL数据库中生成论文管理模块的数据库表。

```
CREATE TABLE IF NOT EXISTS `tbl_paper` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `info` mediumtext COLLATE utf8_bin NOT NULL,
```

```

`status` tinyint(4) DEFAULT NULL,
`pass_date` date DEFAULT NULL,
`pub_date` date DEFAULT NULL,
`index_date` date DEFAULT NULL,
`sci_number` varchar(255) COLLATE utf8_bin DEFAULT NULL,
`ei_number` varchar(255) COLLATE utf8_bin DEFAULT NULL,
`istp_number` varchar(255) COLLATE utf8_bin DEFAULT NULL,
`is_first_grade` tinyint(1) DEFAULT NULL,
`is_core` tinyint(1) DEFAULT NULL,
`other_pub` varchar(255) COLLATE utf8_bin DEFAULT NULL,
`is_journal` tinyint(1) DEFAULT NULL,
`is_conference` tinyint(1) DEFAULT NULL,
`is_intl` tinyint(1) DEFAULT NULL,
`is_domestic` tinyint(1) DEFAULT NULL,
`file_name` varchar(255) COLLATE utf8_bin DEFAULT NULL,
`file_type` varchar(255) COLLATE utf8_bin NOT NULL,
`file_size` int(11) NOT NULL,
`file_content` mediumblob NOT NULL,
`is_high_level` tinyint(1) DEFAULT NULL,
`maintainer_id` int(11) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `tbl_paper_ibfk_1` (`maintainer_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

```

3.2.3.2 科研项目

表 3-2 tbl_project表的结构

字段名称	数据类型	说明
id		整型主键，自动编号
name	变长字符串	项目名称
number	变长字符串	编号
fund_number	变长字符串	经费本编号
is_intl	布尔	是否国际
is_national	布尔	是否国家级
is_provincial	布尔	是否省部级
is_city	布尔	是否市级
is_school	布尔	是否校级
is_enterprise	布尔	是否横向
is_NSF	布尔	是否国家自然科学基金
is_973	布尔	973
is_863	布尔	863
is_NKTRD	布尔	是否科技支撑计划
is_DFME	布尔	是否教育部博士点专项基金
is_major	布尔	是否重大专项
start_date	日期	开始时间
deadline_date	日期	截至时间
conclude_date	日期	结题时间
app_date	日期	申报时间
pass_date	日期	立项时间
app_fund	货币	申报经费
pass_fund	货币	立项经费

表 3-2 给出了科研项目数据库表的结构，其中所有类型为变长字符串均的字段均采用MySQL中的“varchar(255)”数据类型，日期采用“date”数据类型，主键id与外键维护人员id采用“int(11)”数据类型，是否国际、是否国家级等类型为布尔的字段均采用“tinyint(1)”型数据类型；申报经费、立项经费不使用字符串类型或浮点类型存储，而是采用整数部分为15位，小数部分为2位的数值类型存储，方便比较和计算，且没有误差，在MySQL对应“decimal(15,2)”数据类型。综合上述各字段数据类型的选择，使用下面的SQL语句，在MySQL数据库中生成

科研项目数据库表。

```
CREATE TABLE IF NOT EXISTS `tbl_project` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) COLLATE utf8_bin DEFAULT NULL,
  `number` varchar(255) COLLATE utf8_bin DEFAULT NULL,
  `fund_number` varchar(255) COLLATE utf8_bin DEFAULT NULL,
  `is_intl` tinyint(1) DEFAULT NULL,
  `is_national` tinyint(1) DEFAULT NULL,
  `is_provincial` tinyint(1) DEFAULT NULL,
  `is_city` tinyint(1) DEFAULT NULL,
  `is_school` tinyint(1) DEFAULT NULL,
  `is_enterprise` tinyint(1) DEFAULT NULL,
  `is_NSF` tinyint(1) DEFAULT NULL,
  `is_973` tinyint(1) DEFAULT NULL,
  `is_863` tinyint(1) DEFAULT NULL,
  `is_NKTRD` tinyint(1) DEFAULT NULL,
  `is_DFME` tinyint(1) DEFAULT NULL,
  `is_major` tinyint(1) DEFAULT NULL,
  `start_date` date DEFAULT NULL,
  `deadline_date` date DEFAULT NULL,
  `conclude_date` date DEFAULT NULL,
  `app_date` date DEFAULT NULL,
  `pass_date` date DEFAULT NULL,
  `app_fund` decimal(15,2) DEFAULT NULL,
  `pass_fund` decimal(15,2) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

3.2.3.3 专利

表 3-3 tbl_people表的结构

字段名称	数据类型	说明
id	整形	主键, 自动编号
name	变长字符串	专利名称
app_date	日期	申请时间
app_number	变长字符串	申请号
auth_date	日期	授权时间
auth_number	变长字符串	授权号
is_intl	布尔	是否国际
is_domestic	布尔	是否国内
abstract	变长字符串	专利摘要

表 3-4 给出了专利数据库表的结构, 其中所有类型为变长字符串均的字段均采用MySQL中的“varchar(255)”数据类型, 日期采用“date”数据类型, 主键id“int(11)”数据类型, 是否国际、是否国家级等类型为布尔的字段均采用“tinyint(1)”型数据类型; 申报经费、立项经费不使用字符串类型或浮点类型存储, 而是采用整数部分为15位, 小数部分为2位的数值类型存储, 方便比较和计算, 且没有误差, 在MySQL对应“decimal(15,2)”数据类型。综合上述各字段数据类型的选择, 使用下面的SQL语句, 在MySQL数据库中生成专利模块的数据库表。

```
CREATE TABLE IF NOT EXISTS `tbl_patent` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) COLLATE utf8_bin NOT NULL,
  `app_date` date NOT NULL,
  `app_number` varchar(255) COLLATE utf8_bin NOT NULL,
  `auth_number` varchar(255) COLLATE utf8_bin DEFAULT NULL,
  `auth_date` date DEFAULT NULL,
  `is_intl` tinyint(1) NOT NULL,
  `is_domestic` tinyint(1) NOT NULL,
  `abstract` text COLLATE utf8_bin NOT NULL,
  PRIMARY KEY (`id`)
```

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

3.2.3.4 人员

表 3-4 tbl_people表的结构

字段名称	数据类型	说明
id	整形	主键, 自动编号
name	变长字符串	姓名
name_zh	日期	姓名拼音或英文名
type	布尔	教师或者学生
description	变长字符串	介绍

表 3-4 给出了人员数据库表的结构, 其中所有类型为变长字符串均的字段均采用MySQL中的“varchar(255)”数据类型, 主键id“int(11)”数据类型, type字段采用“tinyint(1)”型数据类型; 综合上述各字段数据类型的选择, 使用下面的SQL语句, 在MySQL数据库中生成人员模块的数据库表。

```
CREATE TABLE IF NOT EXISTS 'tbl_people' (
    'id' int(11) NOT NULL AUTO_INCREMENT,
    'name' varchar(255) COLLATE utf8_bin NOT NULL,
    'name' varchar(255) COLLATE utf8_bin,
    'type' tinyint(1),
    'description' varchar(255),
    PRIMARY KEY ('id')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

3.2.4 数据表关系

在科研项目管理中, 需要存储和维护不定数目的项目参与人员, 科研项目和参与人员之间存在着多对多的关系。在这里显然不能直接在科研项目的数据表中添加若干个字段, 存储人员的id; 而是要单独建立一个数据表, 存储项目和参与人员之间的关系, 这样才能满足数据库第二范式, 保持数据有效性, 节约存储空间。

所谓数据库第二范式（2NF），它要求实体的属性完全依赖于主关键字。所谓完全依赖是指不能存在仅依赖主关键字一部分的属性，如果存在，那么这个属性和主关键字的这一部分应该分离出来形成一个新的实体，新实体与原实体之间是一对多的关系。为实现区分通常需要为表加上一个列，以存储各个实例的惟一标识。简而言之，第二范式就是属性完全依赖于主键。

表 3-5 给出了存储科研项目与实际执行人员的关系的数据表结构。这样在查找某一个科研项目记录的时候，只需要在存储科研项目与实际执行人员的关系的数据表中找到满足科研项目id等于当前科研项目记录id，就可以得到该项目的所有实际执行人员id（可能有多个），再依次地按照这些实际执行人员id在人员数据表中查找，便可以得到某一个科研项目的所有实际执行人员。

表 3-5 tbl_project_people_execute表的结构

字段名称	数据类型	说明
project_id	整形	科研项目id，外键，不能为空
people_id	整形	人员id，外键，不能为空
seq	整形	人员在单个项目中的顺序

使用下面的SQL语句生成存储科研项目与实际执行人员的关系的数据表：

```
CREATE TABLE IF NOT EXISTS `tbl_project_people_execute` (
  `project_id` int(11) NOT NULL,
  `people_id` int(11) NOT NULL,
  `seq` int(11) NOT NULL,
  PRIMARY KEY (`project_id`,`people_id`),
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

使用下面的SQL语句给该数据表添加外键限制，以增强本系统数据的有效性和一致性：

```
ALTER TABLE `tbl_project_people_execute`
ADD CONSTRAINT `tbl_project_people_execute_ibfk_1`
FOREIGN KEY (`project_id`)
REFERENCES `tbl_project` (`id`) ON DELETE CASCADE ON 2
UPDATE CASCADE,
ADD CONSTRAINT `tbl_project_people_execute_ibfk_2`
```

```
FOREIGN KEY (`people_id`)
REFERENCES `tbl_people` (`id`) ON DELETE CASCADE ON 2
UPDATE CASCADE;
```

通过使用SQL的“JOIN”命令，可以从存储科研项目与实际执行人员的关系的数据表中得到某个项目的实际执行人员。例如，下面的SQL查询得到id为1的科研项目实际执行人员：

```
SELECT `execute_`.`id` AS `t1_c0`, `execute_`.`name` AS
`t1_c1` FROM `tbl_people` `execute_` INNER JOIN
`tbl_project_people_execute` `execute_peoples_execute_` ON
(`execute_peoples_execute_`.`project_id`=1) AND
(`execute_`.`id`=`execute_peoples_execute_`.`people_id`) 2
ORDER BY
execute_peoples_execute_.seq;
```

同样地，在本系统中，还需要分别存储和维护科研项目与责任书人员的关系、科研项目与维护人员的关系、论文与作者的关系、论文与维护人员的关系、发明和发明人的关系，皆采用类似上述建立一个关系数据库表的方式来实现，限于篇幅关系，在这里不再赘述。

3.3 模型实现

在本小节中，将介绍数据表项最多，数据关系最复杂的科研项目模型的实现，人员模型、论文模型、专利模型的实现方法与科研项目模型相类似，在这里不再赘述。

Yii框架提供了易用的AR(Active Record)实现，AR 是一个流行的对象-关系映射(ORM) 技术。每个AR 类代表一个数据表（或视图），数据表（或视图）的列在AR 类中体现为类的属性，一个AR 实例则表示表中的一行。常见的CRUD 操作作为AR 的方法实现。在本系统中，使用AR技术来实现各个模型。

3.3.1 建立数据库连接

AR 依靠一个数据库连接以执行数据库相关的操作。在Yii框架目录下的config.php中按照下面的PHP代码配置数据库链接。其中，host是MySQL服务器的地址，在开发服务器上，即为本机，填写本地环回地址；dbname为MySQL中存储本系统的数据库名称，根据需要填写，在开发服务器上填写为testdrive；username和password分别为mysql的密码；charset填写在第 18页 第 3.2.2 节中选择的utf-8。

```
return array(  
    //其它配置  
    'db'=>array(  
        'connectionString' => 'mysql:host=127.0.0.1;  
        dbname=testdrive',  
        'emulatePrepare' => true,  
        'username' => 'root',  
        'password' => 'test',  
        'charset' => 'utf8',  
    ),  
);
```

3.3.2 定义AR类

Yii 框架提供了一个名为 CActiveRecord 的 AR 类，如下面的 PHP 代码所示：让科研项目模型类 Project 继承自 CActiveRecord 类，并复写父类的 tableName() 方法，使它返回我们存储科研项目的数据库表的名称，即“tbl_project”，便可以使用 CActiveRecord 提供的接口以面向对象的方式对科研项目的数据库表进行增加、删除、修改、查询了。

```
class Project extends CActiveRecord  
{  
    public function tableName()  
    {  
        return 'tbl_project';  
    }  
}
```

```
}  
}
```

3.3.3 实现增删操作

增删改查操作指对数据库表记录的增加、删除、修改和查询，在本系统的控制器层中，可以使用面向对象的方式对科研项目的数据库表进行操作了：

1. 增加新的科研项目的条目：

```
$project=new Project;  
$project->name='项目名称';  
//对其他字段进行赋值  
$project->save();
```

2. 对现有科研项目的条目进行查询：

```
// 查找满足指定条件的结果中的第一行  
$project=Project::model()->find($condition,$params);  
// 查找具有指定主键值的那一行  
$project=Project::model()->findByPk($projectID,$condition,2  
$params);  
// 查找具有指定属性值的行  
$project=Project::model()->findByAttributes($attributes,2  
$condition,$params);  
// 通过指定的 SQL 语句查找结果中的第一行  
$project=Project::model()->findBySql($sql,$params);  
  
$projects=Project::model()->findAll($condition,$params);  
// 查找带有指定主键的所有行  
$projects=Project::model()->findAllByPk($projectIDs,2  
$condition,$params);  
// 查找带有指定属性值的所有行  
$projects=Project::model()->findAllByAttributes($attributes,2
```

```
$condition,$params);
// 通过指定的SQL语句查找所有行
$projects=Project::model()->findAllBySql($sql,$params);
```

3. 对现有科研项目的条目进行修改:

```
$project=Project::model()->findByPk(10);
$project->name='新的项目名称';
$project->save(); // 将更改保存到数据库
```

4. 对现有科研项目的条目进行删除:

```
$project=Project::model()->findByPk(10); // 2
假设有一个科研项目，其 ID 为 10
$project->delete(); // 从数据表中删除此行

// 删除符合指定条件的行
Project::model()->deleteAll($condition,$params);
// 删除符合指定条件和主键的行
Project::model()->deleteByPk($pk,$condition,$params);
```

3.3.4 定义数据库表关系

在上一节中，已经实现了使用Active Record (AR) 从单个数据表对条目进行增删改查的操作。在本节中，将使用AR 连接多个相关数据表并取回关联（对应MySQL中的“JOIN”语句）后的数据表。

在使用AR 执行关联查询之前，需要让AR 知道一个AR 类是怎样关联到另一个的。

两个AR 类之间的关系直接通过AR 类所代表的数据表之间的关系相关联。在AR 中，定义了三种关系类型：

1. BELONGS_TO: 一对多
2. HAS_MANY: 多对一
3. MANY_MANY:多对多

在本系统的科研项目管理模块中，科研项目与维护者之间的关系是多对一，而科研项目与执行人员之间的关系是多对多。

如下面的PHP代码所示，通过在Project类中复写父类 AR 中定义关系的 relations() 方法,可以定义科研项目与其它数据库表的关系。

```
public function relations()
{
    return array(
        'liability_peoples' => array(
            self::MANY_MANY,
            'People',
            'tbl_project_people_liability(project_id, 2
            people_id)',
            'order'=>'liability_peoples_liability..seq',
            'alias'=>'liability_'
        ),
        'execute_peoples' => array(
            self::MANY_MANY,
            'People',
            'tbl_project_people_execute(project_id, 2
            people_id)',
            'order'=>'execute_peoples_execute..seq',
            'alias'=>'execute_'
        ),
    );
}
```

在定义了数据库表关系之后，可以在控制器层中方便地执行关联查询以及增删改查，就像访问科研项目的模型类Project本身的属性一样：

```
//获取 ID 为 10 的科研项目
$project=Project::model()->findByPk(10);
//获取此科研项目的维护者的姓名：2
```

此处将执行一个关联查询。

```
$maintainter=$project->maintainter->name;
//获取此科研项目的所有实际执行人员
//将返回一个人员对象的数组
$exec_peoples=$project->execute_peoples;
```

3.3.5 验证数据有效性

当插入或更新一行时，我们常常需要检查列的值是否符合相应的规则。如果列的值是由最终用户提供的，这一点就更加重要。总体来说，为了防止恶意攻击，永远不能相信任何来自客户端的数据。为了维护本系统数据的有效性，在本系统的模型层需要定义验证输入数据有效性的规则。

当调用 AR 类的实例的save()方法时，AR 类的实例会自动执行数据验证。验证是基于在 AR 类的rules()方法中指定的规则进行的。如下面给出的PHP代码，通过根据在第 21 页第 3.2.3.2 节建立的数据库表的结构，在科研项目模型类Project中复写父类的rules()方法，可以完成对数据验证规则的定义。这样以来，控制层便能判断用户输入、提交数据是否有效，避免了无效数据的录入或者是恶意攻击的可能性。

```
public function rules()
{
    return array(
        array('name','required'),
        array('is_intl, is_national, is_provincial, is_city, 2
is_school, is_enterprise, is_NSF, is_973, is_863, 2
is_NKTRD, is_DFME, is_major', 'numerical', 2
'integerOnly'=>true),
        array('name, number, fund_number', 'length', 'max'=>2
255),
        array('app_fund, pass_fund', 'length', 'max'=>15),
        array('start_date, deadline_date, conclude_date, 2
app_date, pass_date', 'safe'),
```

```

        array('id', name, number, fund_number, is_intl, 2
        is_national, is_provincial, is_city, is_school, 2
        is_enterprise, is_NSF, is_973, is_863, is_NKTRD, 2
        is_DFME, is_major, start_date, deadline_date, 2
        conclude_date, app_date, pass_date, app_fund, 2
        pass_fund, ', 'safe', 'on'=>'search'),
    );
}

```

3.3.6 搜索与筛选

Yii框架中的 `CDbCriteria` 类代表了一条数据库查询的条件，比如说 MySQL 中的 `WHERE` 字句、`AND` 运算符、`OR` 运算符、`ORDER BY` 子句等。利用它来创建在搜索与查询中需要添加的条件，生成对应的SQL语句。利用 `CDbCriteria` 类提供的 `compare()` 方法，可以实现搜索与筛选功能。`compare()` 的作用是添加一个比较条件到最终生成的SQL语句的 `WHERE` 字句中去。还可以通过设置 `CDbCriteria` 类的 `condition`、`group`、`order` 等属性，配置SQL中对应的条件。

`CActiveDataProvider` 类使用AR的 `CActiveRecord::findAll()` 方法，从数据库中检索信息。它的 `criteria` 属性是 `CDbCriteria` 类的一个实例，能够用来查询多种指定条件。

如下面的PHP代码所示，利用 `CDbCriteria` 类，实现了科研项目模型类 `Project` 中的搜索方法 `search()`，提供给本系统的控制器层调用。`search()` 方法将 `Project` 类的各个属性通过一个 `CDbCriteria` 实例的 `compare()` 方法添加到了查询条件中，最后利用这些查询条件作为一个 `CActiveDataProvider` 类的实例构造函数的参数，传递给这个 `CActiveDataProvider` 类的实例，并返回给模型层使用。

```

public function search()
{

    $criteria=new CDbCriteria;
    $criteria->with=array(
        'execute_peoples',

```



```
'liability_peoples'
);
$criteria->together=true;
$criteria->group = 't.id';
$criteria->compare('execute_peoples.id',$this->2
searchExecutePeople,true);
$criteria->compare('liability_peoples.id',$this->2
searchLiabilityPeople,true);
$criteria->compare('name',$this->name,true);
$criteria->compare('number',$this->number,true);
$criteria->compare('fund_number',$this->fund_number,true)2
;
$criteria->compare('is_intl',$this->is_intl);
$criteria->compare('is_national',$this->is_national);
$criteria->compare('is_provincial',$this->is_provincial);
$criteria->compare('is_city',$this->is_city);
$criteria->compare('is_school',$this->is_school);
$criteria->compare('is_enterprise',$this->is_enterprise);
$criteria->compare('is_NSF',$this->is_NSF);
$criteria->compare('is_973',$this->is_973);
$criteria->compare('is_863',$this->is_863);
$criteria->compare('is_NKTRD',$this->is_NKTRD);
$criteria->compare('is_DFME',$this->is_DFME);
$criteria->compare('is_major',$this->is_major);
$criteria->compare('start_date',$this->start_date,true);
$criteria->compare('deadline_date',$this->deadline_date,2
true);
$criteria->compare('conclude_date',$this->conclude_date,2
true);
$criteria->compare('app_date',$this->app_date,true);
$criteria->compare('pass_date',$this->pass_date,true);
```

```
$criteria->compare('app_fund',$this->app_fund,true);  
$criteria->compare('pass_fund',$this->pass_fund,true);  
  
return new CActiveDataProvider($this, array(  
    'criteria'=>$criteria,  
));  
}
```

参考文献

- [1] L. Shuchun, L. Mengyang, W. Shixian, et al. The Design and Implementation of the Browser/Server Mode MIS [J][J]. Computer Engineering and Applications, 2000, 6:038.
- [2] 蒋磊宏, 董传良, 钟华. 上海交大基于校园网的 MIS 建设规划[J]. 中山大学学报 (自然科学版), 2001, 40(3):10–14.
- [3] J. J. Garrett, et al. Ajax: A new approach to web applications[J]. 2005.
- [4] Z. He-qin. Best scheme of design dynamic website: Apache+ PHP+ MySQL [J][J]. Computer Engineering and Design, 2007, 4:933–938.
- [5] U. Ramana, T. Prabhakar. Some experiments with the performance of LAMP architecture[C]//Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on. .[S.l.]: [s.n.] , 2005:916–920.
- [6] T. A. S. Foundation. The Apache HTTP Server Project[M].[S.l.]: [s.n.] , 2014. <http://httpd.apache.org/>.
- [7] T. P. Group. PHP 手册 序言[M].[S.l.]: [s.n.] , 2014. <http://www.php.net/manual/zh/preface.php>.
- [8] T. P. Group. Alternative PHP Cache[M].[S.l.]: [s.n.] , 2014. <http://www.php.net/manual/zh/intro.apc.php>.
- [9] Y. S. LLC. Performance of Yii[M].[S.l.]: [s.n.] , 2010. <http://www.yiiframework.com/performance/>.
- [10] A. Makarov. Yii 1.1 Application Development Cookbook[M].[S.l.]: Packt Publishing Ltd, 2011.
- [11] T. Reenskaug. THING-MODEL-VIEW-EDITOR-an Example from a planningssystem[J]. technical note, Xerox Parc, 1979.
- [12] 陈旭, 刘加伶. Client/Server 与 Browser/Server 结构的分析与比较[J]. 重庆工学院学报, 2000, 14(2):100–103.

致 谢

历时将近两个月的时间终于将这篇论文写完，在论文的写作过程中遇到了无数的困难和障碍，都在同学和老师的帮助下度过了。尤其要强烈感谢我的论文指导老师—XX老师，她对我进行了无私的指导和帮助，不厌其烦的帮助进行论文的修改和改进。另外，在校图书馆查找资料的时候，图书馆的老师也给我提供了很多方面的支持与帮助。在此向帮助和指导过我的各位老师表示最中心的感谢！

感谢这篇论文所涉及到的各位学者。本文引用了数位学者的研究文献，如果没有各位学者的研究成果的帮助和启发，我将很难完成本篇论文的写作。

感谢我的同学和朋友，在我写论文的过程中给予我了很多你问素材，还在论文的撰写和排版灯过程中提供热情的帮助。由于我的学术水平有限，所写论文难免有不足之处，恳请各位老师和学友批评和指正！