|  |  |
| --- | --- |
|  | 惠州学院  HUIZHOU UNIVERSITY |

**毕 业 论 文（设 计）**

**中文题目： 课室自动点名系统前台开发**

**英文题目： Client Development of Classroom**

**Attendance auto-taking System**

**姓 名**  **林 泽 全**

**学 号 091402116**

**专业班级 网络工程1班**

**指导教师 彭 刚**

**提交日期 2013年04月**

# 摘 要

**课室自动点名系统(已申请国家专利)是对目前学校的考勤实际情况而进行设计、开发的，目的是为了用一种智能的自动化的方式实现课室的考勤任务。利用PHP作为后台语言，ExtJS作为前台开发库以及MySQL数据库访问技术实现了基于B/S架构的应用系统。本系统运用ThinkPHP框架MVC结构开发，利用ThinkPHP本身特点进行模板分离，使模块之间的耦合性降低，减轻重新建立解决复杂问题方案的负担，增强了系统的可扩展性，便于进行内部的定制化。通过使用Linux、PHP、MySQL和Apache技术建设动态网站，充分发挥了LAMP结构所独有的易用性、跨平台性和安全性，从而构建了一个运行高效、安全可靠、适用性广的管理系统，实现了课室考勤信息的管理与交互，满足自动点名的需求。**

**论文首先简要介绍了课室自动点名系统的一些研究与应用背景，其次介绍了该系统所采用的开发工具、平台以及开发环境。在此基础上，论文详尽描述了课室自动点名系统前台(包括学生端和教室端)的设计与实现过程。**

**关键词：****PHP，MySQL，ThinkPHP，ExtJS，课室自动点名系统**

**Abstract**

**In order to take classroom attendances in a more intelligent and automatic way, Classroom Attendance Auto-Taking System, which takes the real-life situation of classroom attendance into consideration and has obtained the national patent, has been designed. The system, based on PHP as its back-end language, ExtJS as its front-end development library, and MySQL as its database technology, is a B/S architecture application. With the help of the MVC architecture and Template system of ThinkPHP, the coupling among different modules lowered the burden of reconstruction for solving complicated problems relieved, the extensibility for easy inner custom-made building enhanced. The implementation of requirements for classroom attendance management and interaction are made possible through the construction of a high-effective, safe, and broadly-applicable management system which is built on top of the accessibility, cross-platform, and safety of LAMP architecture, through the technical combination of Linux, PHP, MySQL, and Apache for building dynamical websites.**

**At first, this paper introduces the researches and the application background. Then, the developing tools, platform and development environment are illustrated. The rest of this paper detailed describes the whole process of the design and implementation of the system, both clients for students and classrooms.**

**Keywords: PHP, MySQL, ThinkPHP, ExtJS, Attendance Auto-Taking System**

目录

[摘 要 2](#_Toc354093377)

[前 言 5](#_Toc354093378)

[第一章 绪论 6](#_Toc354093379)

[1.1 研究背景 6](#_Toc354093380)

[1.2 设计目标 8](#_Toc354093381)

[1.3 本文结构 9](#_Toc354093382)

[第二章 系统开发环境与技术 9](#_Toc354093383)

[2.1 系统开发环境 9](#_Toc354093384)

[2.1.1 Apache服务器介绍 9](#_Toc354093385)

[2.1.2 Linux服务器介绍 10](#_Toc354093386)

[2.1.3 MySQL服务器介绍 11](#_Toc354093387)

[2.1.4 Git软件版本控制管理 11](#_Toc354093388)

[2.2 系统开发技术 12](#_Toc354093389)

[2.2.1 PHP脚本语言 12](#_Toc354093390)

[2.2.2 ThinkPHP开发框架 13](#_Toc354093391)

[2.2.3 MVC模式 13](#_Toc354093392)

[2.2.4 JavaScript与ExtJS库 14](#_Toc354093393)

[2.2.5 新浪微博应用授权与开发流程 15](#_Toc354093394)

[第三章 系统需求分析与前台设计 23](#_Toc354093395)

[3.1 需求分析 23](#_Toc354093396)

[3.1.1 系统前端操作平台设计概述 23](#_Toc354093397)

[3.1.2 系统用例图 24](#_Toc354093398)

[3.2 系统设计 26](#_Toc354093399)

[3.2.1 系统层次划分 26](#_Toc354093400)

[3.2.2 数据库设计 26](#_Toc354093401)

[3.2.3 系统时序图 32](#_Toc354093402)

[第四章 系统详细设计与实现 34](#_Toc354093403)

[4.1 系统项目的文件结构 34](#_Toc354093404)

[4.2 设计思路 35](#_Toc354093405)

[4.3 核心功能实现代码及运行效果 35](#_Toc354093406)

[第五章 总结与展望 56](#_Toc354093407)

[5.1 课题总结 56](#_Toc354093408)

[5.2 进一步开发的展望 56](#_Toc354093409)

[第六章 参考文献 57](#_Toc354093410)

[第七章 致谢 57](#_Toc354093411)

[第八章 附录 58](#_Toc354093412)

[外文翻译 58](#_Toc354093413)

[Original (Excerpt from Pro PHP: Patterns, Frameworks, Testing and More) 58](#_Toc354093414)

[译文（节选自《PHP高级程序设计 模式、框架、测试》） 68](#_Toc354093415)

# 前 言

随着科技的快速发展，我们身边的一切改变得如此之快，把固定通信改变为移动通信，把触感按钮改变为触摸按钮，IBM甚至提出在5年内生产出用人的意志控制的设备，这意味着人们在追求更为便捷的生活。大学的课堂中，老师往往需要话费许多时间去点名，处理请假、旷课、迟到、早退，最终记录学生的出勤率，给予评价，事情是如此的繁琐。那么我们是否能研发一套系统，能够完全替代老师所要做的学生出勤率统计的工作，并且阶段性对考勤情况进行分析，让老师的工作更为轻松便捷呢？这就是我们的教室自动点名系统。

本文的结构如下：

第一章是绪论部分，叙述了本系统的开发背景、开发目标、开发意义以及国内外的一些比较。

第二章介绍了本系统的开发环境以及开发技术。

第三章是系统需求分析与前台设计，阐述了在系统开发前的分析工作、系统的整体设计以及数据库设计。

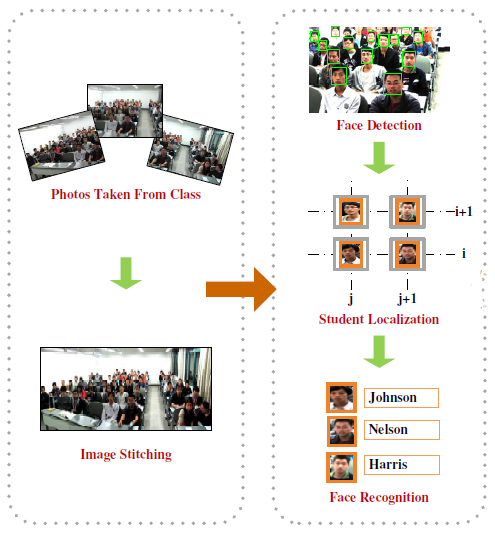
第四章是系统的详细设计与功能实现。

第五章是总结和展望，为整个项目的过程和感悟进行总结，并对项目未来的开发做出展望。

# 第一章 绪论

## 研究背景

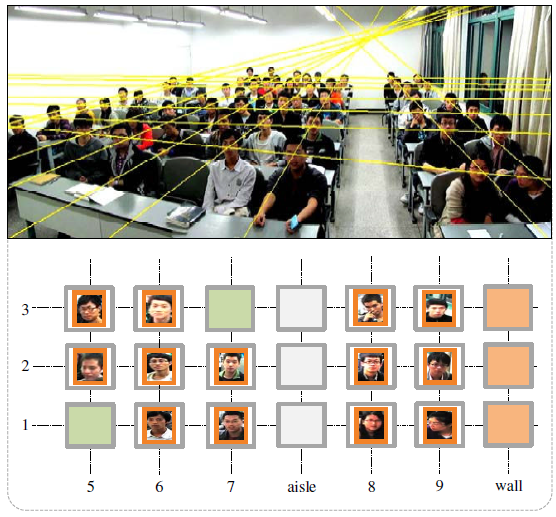
当今时代信息技术飞速发展，人们已经处于数字化的生存空间之中。科技已经从根本上改变了人民的生活方式和工作方式，也改变了课堂上的各种传统。本系统通过对课室在有人和没有人的时候分别获取图像，经过算法进行对比得到出勤信息，再通过管理系统进行管理，通过人性化和多样化的展示方式展示给教师和学生。下图是对点名系统工作过程的一个工作流程图。



对于摄像机拍好的照片，根据实际情况进行人脸识别处理。首先固定好各自的座位，取出照片上各人的脸部，放入二维表中，与正常情况下进行对比。图片进行拆分和处理主要如下面两图所示。







教室自动点名系统是为实现教室点名、学生出勤统计处理的系统，它可以智能地对教室的出勤情况进行分析，并统计好学生的请假、旷课、迟到和早退情况，并记录在学生的出勤信息库里。学生可以根据系统查询自己的考勤情况，给自己阶段性总结；教师可以根据系统分析出勤状况，给予合理的综合评价。

我们生活在一个科技日新月异的时代，传统而繁琐的事情需要利用科技去改变，去追求更为便捷轻松的生活。

## 设计目标

开发设计本系统的任务是完成只能考勤功能，通过该系统能够自动获取出勤信息，并且用户能对考勤相关的信息（如课程、人员、考勤等）进行管理。

开发系统的目标是实现点名自动化及对考勤信息进行各种管理的功能，满足的要求具体可包括如下几个方面：

1. 完成智能考勤功能；
2. 对考勤信息进行统计及考勤相关信息的管理；
3. 系统运行安全稳定；
4. 要求编写完善的文档，方便后期程序设计工作。

## 本文结构

本文的章节安排如下：

第一章绪论部分，说明本文的研究背景和设计目标，为接下来的系统后台的分析与设计做基础准备。

第二章是背景技术，简单介绍系统实现所用到的平台与技术。如LAMP开发环境、ExtJS框架、ThinkPHP框架、MVC三层结构、新浪微博应用开发以及GIT版本控制管理等等。

第三章是系统分析与设计，进行需求分析，根据需求分析设计相应的功能与表结构。

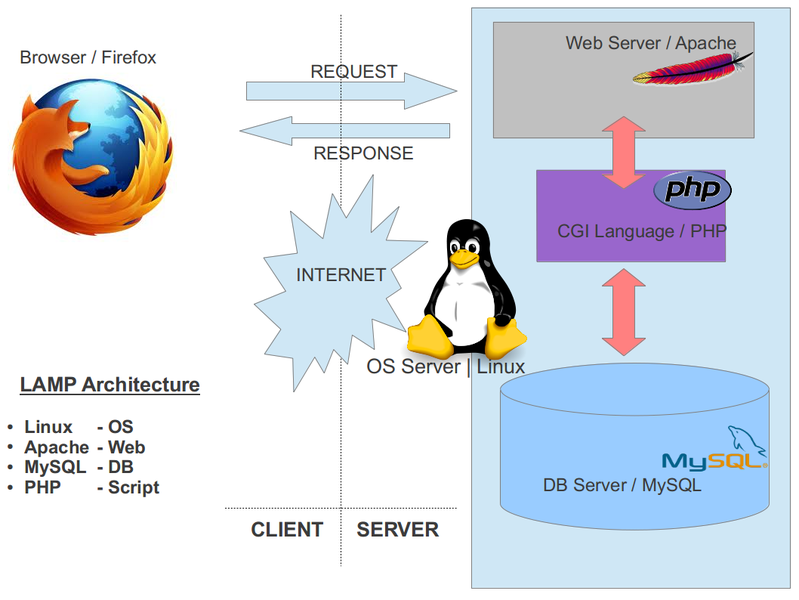
第四章是系统实现，给出系统后台的具体实现，包括实现效果和核心代码说明。

第五章是总结与展望，对本文所做的工作进行总结，并针对目前工作中的不足以及对未来工作的展望。

# 第二章 系统开发环境与技术

## 系统开发环境

教室自动点名系统以LAMP（即Linux、Apache、MySQL和PHP）为开发平台，新浪SAE分布式WEB服务为运行平台，使用ExtJS框架作为前台开发库，ThinkPHP框架作为后台，Git进行版本控制，利用MVC模式实现整个系统的开发。



### Apache服务器介绍

Apache HTTP Server（简称Apache）是Apache软件基金会的一个开放源码的网页服务器，可以在大多数计算机操作系统中运行，由于其多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。它快速、可靠并且可通过简单的API扩展，将Perl/Python等解释器编译到服务器中。

Apache起初由伊利诺伊大学香槟分校的国家超级电脑应用中心（NCSA）开发。此后，Apache被开放源代码团体的成员不断的发展和加强。Apache 服务器拥有牢靠可信的美誉，已用在超过半数的因特网站中，特别是几乎所有最热门和访问量最大的网站。

开始，Apache只是Netscape网页服务器（现在是Sun ONE）之外的开放源代码选择。渐渐的，它开始在功能和速度超越其他的基于Unix的HTTP服务器。1996年4月以来，Apache一直是Internet上最流行的HTTP服务器: 1999年5月它在 57% 的网页服务器上运行；到了2005年7月这个比例上升到了69%。在2005年11月的时候达到接近70％的市占率，不过随着拥有大量域名数量的主机域名商转换为微软IIS平台，Apache市场占有率近年来呈现些微下滑。而Google自己的网页服务器平台GWS推出后，加上Lighttpd这个轻量化网页服务器软件使用的网站慢慢增加，反应在整体网页服务器市场占有率上，根据netcraft在2007年7月的最新统计数据，Apache的市场占有率已经降为52.65％，8月时又滑落到50.92％。尽管如此，它仍旧是现阶段因特网市场上，市场占有率最高的网页服务器软件。

作者宣称因为这个名字好记才在最初选择它，但是流传最广的解释是（也是最显而易见的）:这个名字来自这么一个事实:当Apache在1995年初开发的时候，它是由当时最流行的HTTP服务器NCSA HTTPd 1.3的代码修改而成的，因此是“一个修补的（a patchy）”服务器。然而在服务器官方网站的FAQ中是这么解释的:“‘Apache’这个名字是为了纪念名为Apache(印地语)的美洲印第安人 土著的一支，众所周知他们拥有高超的作战策略和无穷的耐性”。无论如何，Apache 2.x 分支不包含任何 NCSA 的代码。

Apache支持许多特性，大部分通过编译的模块实现。这些特性从服务器端的编程语言支持到身份认证方案。一些通用的语言接口支持Perl，Python，Tcl和 PHP。流行的认证模块包括mod\_access，mod\_auth和mod\_digest。其他的例子有SSL和TLS支持（mod\_ssl），代理服务器(proxy)模块，很有用的URL重写（由mod\_rewrite实现），定制日志文件（mod\_log\_config），以及过滤支持（mod\_include 和 mod\_ext\_filter）。Apache日志可以通过网页浏览器使用免费的脚本AWStats或Visitors来进行分析。

### Linux服务器介绍

Linux（i/ˈlɪnəks/ LIN-əks）是一种自由和开放源代码的类UNIX操作系统。定义Linux的组件是Linux内核，该操作系统内核由林纳斯·托瓦兹在1991年10月5日首次发布。

严格来讲，术语Linux只表示操作系统内核本身，但通常采用Linux内核来表达该意思。Linux则常用来指基于Linux内核的完整操作系统，包括GUI组件和许多其他实用工具。由于这些支持用户空间的系统工具和库主要由理查德·斯托曼于1983年发起的GNU计划提供，自由软件基金会提议将该组合系统命名为GNU/Linux。

Linux最初是作为支持英特尔x86架构的个人电脑的一个自由操作系统。目前Linux已经被移植到更多的计算机硬件平台，远远超出其他任何操作系统。Linux是一个领先的操作系统，可以运行在服务器和其他大型平台之上，如大型主机和超级计算机。世界上500个最快的超级计算机90％以上运行Linux发行版或变种，包括最快的前10名超级电脑运行的都是基于Linux内核的操作系统。 Linux也广泛应用在嵌入式系统上，如手机，平板电脑，路由器，电视和电子游戏机等。在移动设备上广泛使用的Android操作系统就是创建在Linux内核之上。

Linux也是自由软件和开放源代码软件发展中最著名的例子。只要遵循GNU通用公共许可证,任何个人和机构都可以自由地使用Linux的所有底层源代码，也可以自由地修改和再发布。通常情况下，Linux被打包成供个人计算机和服务器使用的Linux发行版，一些流行的主流Linux发布版，包括Debian（及其派生版本Ubuntu，Linux Mint），Fedora（及其相关版本Red Hat Enterprise Linux，CentOS）和openSUSE等。Linux发行版包含Linux内核和支撑内核的实用程序和库，通常还带有大量可以满足各类需求的应用程序。个人计算机使用的Linux发行版通常包X Window和一个相应的桌面环境，如GNOME或KDE。桌面Linux操作系统常用的应用程序,包括Firefox网页浏览器，LibreOffice办公软件，GIMP图像处理工具等。由于Linux是自由软件，任何人都可以创建一个符合自己需求的Linux发行版。

Linux发行版一直被用来作为服务器的操作系统，并且已经在该领域中占据重要地位。根据2006年9月en:Netcraft的报告显示，十个最大型的网络托管公司有八个公司在其Web服务器运行Linux发行版。

Linux发行版是构成LAMP（Linux操作系统， Apache， MySQL ， Perl / PHP / Python）的重要部分，LAMP是一个常见的网站托管平台，在开发者中已经得到普及。

Linux发行版也经常使用作为超级计算机的操作系统，2010年11月公布的超级计算机前500强，有459个（91.8％）运行Linux发行版。曾经是世界上最强大的超级计算机——IBM的红杉(IBM Sequoia)，已于2011年交付劳伦斯利福摩尔国家实验室，并于2012年6月开始运作，也是选择Linux作为操作系统。

### MySQL服务器介绍

MySQL（发音：[ˌmaɪ ɛs kjuː ˈɛl]，但也经常读作My-SEQuel）是一个开放源代码的关系数据库管理系统，原开发者为瑞典的MySQL AB公司，该公司于2008年被升阳微系统（Sun Microsystems）收购。2009年，甲骨文公司（Oracle）收购升阳微系统公司，MySQL成为Oracle旗下产品。

MySQL在过去由于性能高、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。随着MySQL的不断成熟，它也逐渐用于更多大规模网站和应用，比如维基百科、Google和Facebook等网站。非常流行的开源软件组合LAMP中的“M”指的就是MySQL。

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

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，因此使用这种方式不用花一分钱就可以创建起一个稳定、免费的网站系统。MySQL加PHP的配对在互联网上的应用相比LAMP来说更为常见，并获得了“动态配对”（Dynamic Duo）的雅号，大部分Blog网站基于的WordPress系统主要运用MySQL加PHP的配对。除了LAMP之外，用于Solaris、Windows和Mac上的网站构架也分别被称为SAMP、WAMP和MAMP。

### Git软件版本控制管理

Git是一个分布式版本控制／软件配置管理软件，原来是linux内核开发者林纳斯·托瓦兹（Linus Torvalds）为了更好地管理linux内核开发而创立的。需要注意的是和GNU Interactive Tools，一个类似Norton Commander界面的文件管理器有所不同。

Git最初的开发动力来自于BitKeeper和Monotone。Git最初只是作为一个可以被其他前端比如Cogito或StGIT包装的后端而开发的。不过，后来Git内核已经成熟到可以独立地用作版本控制。很多有名的软件都使用Git来进行版本控制，其中有Linux内核、X.Org服务器和OLPC内核开发。

Git是用于Linux内核开发的版本控制工具。与CVS、Subversion一类的集中式版本控制工具不同，它采用了分布式版本库的作法，不需要服务器端软件，就可以运作版本控制，使得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追踪（merge tracing）能力。

实际上内核开发团队决定开始开发和使用Git来作为内核开发的版本控制系统的时候，世界开源社群的反对声音不少，最大的理由是Git太艰涩难懂，从Git的内部工作机制来说，的确是这样。但是随着开发的深入，Git的正常使用都由一些友善的命令稿来执行，使Git变得非常好用。现在，越来越多的著名项目采用Git来管理项目开发，例如：wine、U-boot等。

作为开源自由原教旨主义项目，Git没有对版本库的浏览和修改做任何的权限限制，通过其他工具也可以达到有限的权限控制，比如：gitosis, CodeBeamer MR。原本Git的使用范围只适用于Linux/Unix平台，但在Windows平台下的使用也逐渐成熟，这主要归功于Cygwin、msysgit环境与TortoiseGit这样易用的GUI工具。其实Git的源代码中已经加入了对Cygwin与MinGW编译环境的支援，且逐渐完善，为Windows使用者带来福音。

## 系统开发技术

教室自动点名系统的开发技术是基于ThinkPHP框架和ExtJS框架的组合开发技术。开发平台是Linux，以VIM为编辑器，Apache作为服务器，MySQL作为数据库，新浪SAE分布式WEB服务器为运行平台。

### PHP脚本语言

PHP（PHP：Hypertext Preprocessor）是一种在电脑上运行的脚本语言，主要用途是在于处理动态网页，也包含了命令行运行接口（command line interface），或者产生图形用户界面（GUI）程序。

PHP最早由丹麦人拉斯姆斯·勒多夫在1995年发明，而现在PHP的标准由PHP Group和开放源代码社区维护。PHP以PHP License作为许可协议，不过因为这个协议限制了PHP名称的使用，所以和开放源代码许可协议GPL不兼容。

PHP的应用范围相当广泛，尤其是在网页程序的开发上。一般来说PHP大多运行在网页服务器上，通过运行PHP代码来产生用户浏览的网页。PHP可以在多数的服务器和操作系统上运行，而且使用PHP完全是免费的。根据2007年4月的统计数据，PHP已经被安装在超过2000万个网站和100万台服务器上。

据National Vulnerability Database数据显示，与PHP有关的数据库攻击比例为：20% 2004, 28% 2005, 43% 2006, 36% 2007, 35% 2008 and 32% 2009。其中很多的漏洞都可以通过远程操作完成，如：黑客可以通过网络连接攻击服务器，达到盗取或毁坏数据，发送垃圾邮件或进行分布式拒绝服务攻击。但是随着更多的关注，PHP也变得越来越安全了。

### ThinkPHP开发框架

ThinkPHP 是一个免费开源的，快速、简单的面向对象的轻量级PHP开发框架，遵循Apache2开源协议发布，是为了敏捷WEB应用开发和简化企业应用开发而诞生的。ThinkPHP从诞生以来一直秉承简洁实用的设计原则，在保持出色的性能和至简的代码的同时，也注重易用性。并且拥有众多的原创功能和特性，在社区团队的积极参与下，在易用性、扩展性和性能方面不断优化和改进，众多的典型案例确保可以稳定用于商业以及门户级的开发。

经过6年的不断积累和重构，3.0版本又是一个新的里程碑版本，在框架底层的定制和扩展方面趋于完善，使得应用的开发范围和需求适应度更加扩大，能够满足不同程度的开发人员的需求。而且引入了全新的CBD（核心+行为+驱动）架构模式，旨在打造DIY框架和AOP编程体验，让ThinkPHP能够在不同方面都能快速满足项目和应用的需求，并且正式引入SAE、REST和Mongo支持。

使用ThinkPHP，你可以更方便和快捷的开发和部署应用。当然不仅仅是企业级应用，任何PHP应用开发都可以从ThinkPHP的简单和快速的特性中受益。ThinkPHP本身具有很多的原创特性，并且倡导大道至简，开发由我的开发理念，用最少的代码完成更多的功能，宗旨就是让WEB应用开发更简单、更快速。为此ThinkPHP会不断吸收和融入更好的技术以保证其新鲜和活力，提供WEB应用开发的最佳实践！经过6年来的不断重构和改进，ThinkPHP达到了一个新的阶段，能够满足企业开发中复杂的项目需求，足以达到企业级和门户级的开发标准。

### MVC模式

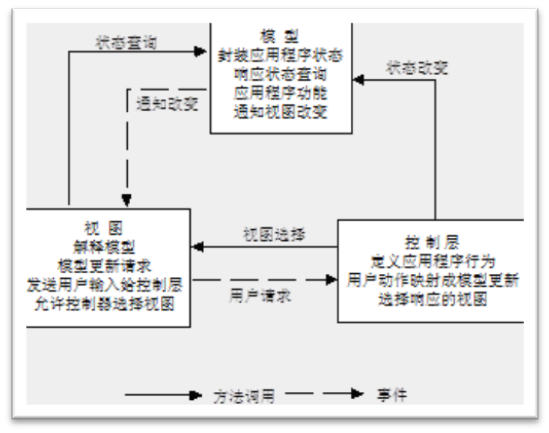
MVC即Model-View-Controller，M代表业务逻辑，V代表视图，C代表控制层。通过使用MVC模式就把软件层次分为三个层：模型层、视图层、控制层。

模型（Model）层：也就是软件业务的实现部分。一个软件有什么功能，它如何实现功能，这些功能之间的联系都是在模型层进行处理的。可以这样说，模型层是一个软件的核心，软件功能的好坏、软件功能是否完善，业务层起到了决定性的作用。

视图（View）层：就是用户能看到的并与之交互的界面。视图向用户显示相关的数据，并能接受用户的输入数据，但是它不能进行任何实际的业务处理。视图还可以接受模型发出来的数据更新事件，从而对用户界面进行同步更新。

控制（Controller）层：调用业务层，然后进行视图的切换。控制层几乎不做任何与业务有关的事情，比如登录验证这些。所以在软件的分层结构设计中，控制层的代码量几乎是最少的，但它的作用也是非常重要的。通过将模型、视图与控制层的分离，各个层之间的联系非常少，但这样做的好处是，控制层可以随意的切换到自己想要展示的页面，然后控制层调用业务逻辑层和持久化保存数据层，来实现软件的功能，再转回到控制层，最后将最终的结果显示到视图层，这实际上是一种模型的变化——传播机制。

模型、视图、控制层三者之间的关系和各自的主要功能如图3-2所示。



### JavaScript与ExtJS库

JavaScript，也称ECMAScript，是一种广泛用于客户端网页开发的脚本语言，最早是在HTML上使用的，用来给HTML网页添加动态功能，然而现在JavaScript也可被用于网络服务器，如Node.js。

一般来说，完整的JavaScript包括以下几个部分：

* ECMAScript，描述了该语言的语法和基本对象；
* 文档对象模型（DOM），描述处理网页内容的方法和接口；
* 浏览器对象模型（BOM），描述与浏览器进行交互的方法和接口。

它的基本特点如下：

* 是一种解释性脚本语言（代码不进行预编译）；
* 主要用来向 HTML 页面添加交互行为；
* 可以直接嵌入 HTML 页面，但写成单独的js文件有利于结构和行为的分离。

JavaScript常用来完成以下任务：

* 嵌入动态文本于HTML页面
* 对浏览器事件做出响应
* 读写HTML元素
* 在数据被提交到服务器之前验证数据
* 检测访客的浏览器信息
* 控制cookies，包括创建和修改等



ExtJS 主要用来开发RIA富客户端的AJAX应用，主要用于创建前端用户界面，与后台技术无关的前端ajax框架。因此，可以把ExtJS用在.Net、Java、PHP等各种开发语言开发的应用中。ExtJS最开始基于YUI技术，由开发人员JackSlocum开发，通过参考JavaSwing等机制来组织可视化组件，无论从UI界面上CSS样式的应用，到数据解析上的异常处理，都可算是一款不可多得的JavaScript客户端技术的精品。

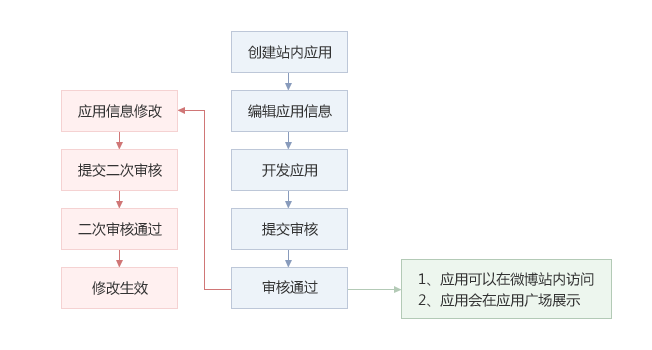
Ext的UI组件模型和开发理念脱胎、成型于Yahoo组件库YUI和Java平台上Swing两者，并为开发者屏蔽了大量跨浏览器方面的处理。相对来说，EXT要比开发者直接针对DOM、W3C对象模型开发UI组件轻松。

特点如下：

* 高性能, customizable UI widgets
* Well designed, documented and extensible Component model
* Commercial and Open Source licenses available

### 新浪微博应用授权与开发流程

1. **站内应用创建及发布流程图**



1. **登录微博账号**

建议由技术开发人员使用官方微博账号来登录。



1. **设置开发者信息**

在开放平台首页上选择“我是开发者”，进入到应用开发页面； 在应用开发页面中，点击“创建应用”按钮：



若您还没有填写开发者信息，系统会自动跳转到编辑开发者信息页面。



填写完毕，请点击保存按钮进行保存。

1. **创建站内应用**

在应用开发页面点击“创建应用”，会弹出浮层：



选择创建站内应用，即可进入创建站内应用的页面。

创建站内应用需要完善以下信息：

* 应用名称：该名称用于来源显示，用户通过应用发布微博信息后，应用的名称会出现在微博信息下方，点击后直接进入应用地址。
* 绑定域名：请填写根域名，绑定域名后其他域名无法使用您的appkey。
* 应用分类：请依据您所开发的应用类型来选择相关的分类信息；请确认您的应用分类正确，否则无法通过您的审核申请。



1. **编辑应用基本信息**
2. 应用页面设置：

* 站内应用地址：该地址是应用在新浪微博展示的应用基本地址。
* 应用实际地址：该地址是第三方应用的实际地址，微博服务器会向该地址发送请求获取应用内容。
* Iframe高度：开发者可以设置Iframe的高度，也可以选择Iframe的高度根据你的应用页面自适应，但需要开发者在应用页面中部署一段JS代码：

<script src="http://tjs.sjs.sinajs.cn/t35/apps/opent/js/frames/client.js" language="JavaScript"></script>

提示：若你的应用有多个页面，并且每个页面的高度不一致，则每个页面都需要部署以上JS代码。



1. 运营信息

开发者想要对应用进行维护、升级等操作时，可以开启维护状态，并填写维护原因告知平台工作人员，此时用户再访问你的应用时，会提示用户你的应用正在维护中，请稍后访问。



1. 应用截图

请上传相关的应用介绍照片，这些图片会在应用频道中展示，让用户能更直观的了解你的应用。



1. 测试用户

应用在未通过审核之前，不能大范围推广，您可以设置最多15个测试用户来测试尚在开发中的应用。

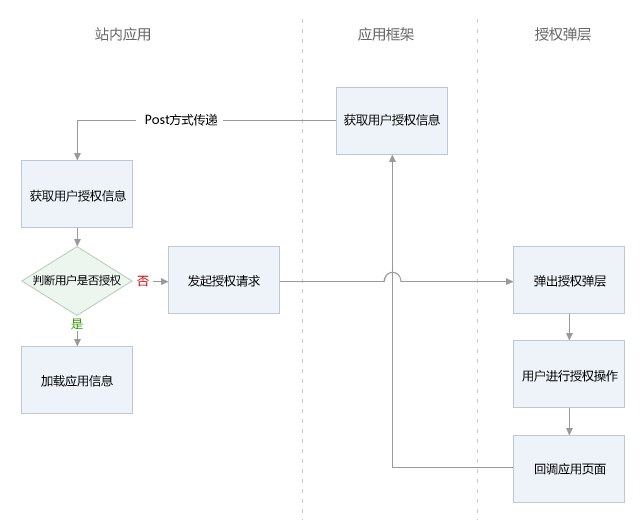


相关内容填写完毕后，请点击保存按钮进行保存。

1. **技术开发**

应用信息保存成功后，将会获得该应用的App Key和Secret Key，您可以通过这两个Key数据开始进行相关的技术开发工作。App Key是应用的唯一标识，开放平台通过App Key来鉴别应用的身份。AppSecret是给应用分配的密钥，开发者需要妥善保存这个密钥，这个密钥用来保证应用来源的可靠性，防止被伪造。

站内应用授权流程如下：



站内应用框架会POST给您的应用以下信息：

1. 用户未授权传递参数

|  |  |  |  |
| --- | --- | --- | --- |
|  | **必选** | **类型及范围** | **说明** |
| **user** | true | array | 当前用户对象 |
| **algorithm** | true | string | 签名算法，暂时用HMAC-SHA256 |
| **issued\_at** | true | int | 服务端生成时间, unix timestamp格式 |

用户未授权时应用需要自动判断，并调用OAuth 2.0 授权弹层。

1. 用户授权后传递参数

|  |  |  |  |
| --- | --- | --- | --- |
|  | **必选** | **类型及范围** | **说明** |
| **user** | true | array | 当前用户对象 |
| **algorithm** | true | string | 签名算法，暂时用HMAC-SHA256 |
| **issued\_at** | true | int | token生成时间, unix timestamp格式 |
| **expires** | true | int | token过期时间, unix timestamp格式 |
| **oauth\_token** | true | string | access token |
| **user\_id** | true | uint64 | 当前用户微博user id |

// 从POST过来的signed\_request中提取oauth2信息

if(!empty($\_REQUEST["signed\_request"])){

$o = new SaeTOAuth( WB\_AKEY , WB\_SKEY );

$data=$o->parseSignedRequest($\_REQUEST["signed\_request"]);

if($data=='-2'){

die('签名错误!');

}else{

$\_SESSION['oauth2']=$data;

}

}

* 判断是否获取到access token

//判断用户是否授权

if (empty($\_SESSION['oauth2']["user\_id"])) {//若没有获取到access token，则发起授权请求

include "auth.php";

exit;

} else {//若已获取到access token，则加载应用信息

$c = new SaeTClient( WB\_AKEY , WB\_SKEY ,$\_SESSION['oauth2']['oauth\_token'] ,'' );

}

* 调用应用授权弹层

调用授权弹层时，页面背景用一张应用图片填充，给用户带来更好的体验。

<style>

body{

background:url("http://qimeng.appsina.com/images/1.jpg");

}

</style>

<script src="http://tjs.sjs.sinajs.cn/t35/apps/opent/js/frames/client.js" language="JavaScript"></script>

<script>

// 弹出授权弹层：

function authLoad(){

App.AuthDialog.show({

client\_id : '<?=WB\_AKEY;?>', //必选，appkey

redirect\_uri : '<?=$canvas\_page;?>', //必选，授权后的回调地址

height: 120, //可选，默认距顶端120px

scope: ''friendships\_groups\_read, friendships\_groups\_write"//可选，授权页scope参数

});

}

</script>

* 用户进行授权操作

用户进行授权操作后，会回调站内应用页面（即apps.weibo.com/<个性域名>），重新进行以上流程。

* 直接跳转至指定页面

假设您的站内应用地址是：apps.weibo.com/liwu，iframe中应用实际地址是：www.liwu.com。

当浏览器地址栏是apps.weibo.com/liwu/demo/a1.php时，iframe中的页面会定位至www.liwu.com/demo/a1.php

* 应用内超链接

站内框架支持两类形式的超链接：

1. 在当前窗口打开，地址栏不变

调用方法:<a href="应用实际地址链接" target="\_self"></a>或<a href="应用实际地址链接"></a>

1. 在当前窗口打开，刷新整个框架，地址栏改变

调用方法:<a href="<http://apps.weibo.com/xxx>" target="\_top"></a>

1. **站内应用审核**

当您完成站内应用的开发并测试没有问题后，可以在站内应用的汇总信息页面点击“提交审核”按钮，将您的应用提交给平台工作人员进行审核。 我们的工作人员在收到您的申请后，会和您进行联络，并试用您所开发的应用，在确认应用开发已经完成并且可以正常使用后，会通过您的申请。



站内应用通过审核之后，将突破15个测试账号的限制，可以被所有的微博用户所使用，并且自动被推荐到应用广场。

# 第三章 系统需求分析与前台设计

## 需求分析

在计算机广泛运用的今天，应用计算机已经大大地提高了社会各方面的效率，而作为学校，社会知识的高度集中区，计算机已经成为一种必需品。有了计算机，那么我们就应该在方方面面想到用它来提高我们的效率。对于大学生的考勤方面，现在越来越体现出其重要性。虽然大学生都已经步入成年，但是其自制力还是很不乐观，纪律方面还需要老师和学校的监督。但是对于上课点名很耗费时间，而且老师点名以后对学生的出勤状况也没有一个总结的感知，学生也不能即时对自己的考勤情况有一个分析反省。通过自动点名系统的服务器会根据摄像头的采样对比结果分析学生的出勤情况，并保存在数据库中。然后通过系统对出勤情况的统计分析，最终得出各类出勤相关结果。此过程无需任课老师参与，全由计算机后台实现。而学生通过网络，便可以轻松查看自己一天，一个星期，一个月任何一个阶段的出勤情况。本系统重在实现无纸化点名，还有对点名结果的统计与分析以及一系列点名系统的附属功能，同时节约时间与物质成本，让老师的教学更加便捷。

### 系统前端操作平台设计概述

教室自动点名系统按操作平台可分为前台操作与后台操作，从用户角度出发，可以分为“超级管理员”、“学生”、“老师”三大角色。本毕业设计主要是针对自动点名系统的前台开发中的“学生端”和“教师端”的开发。

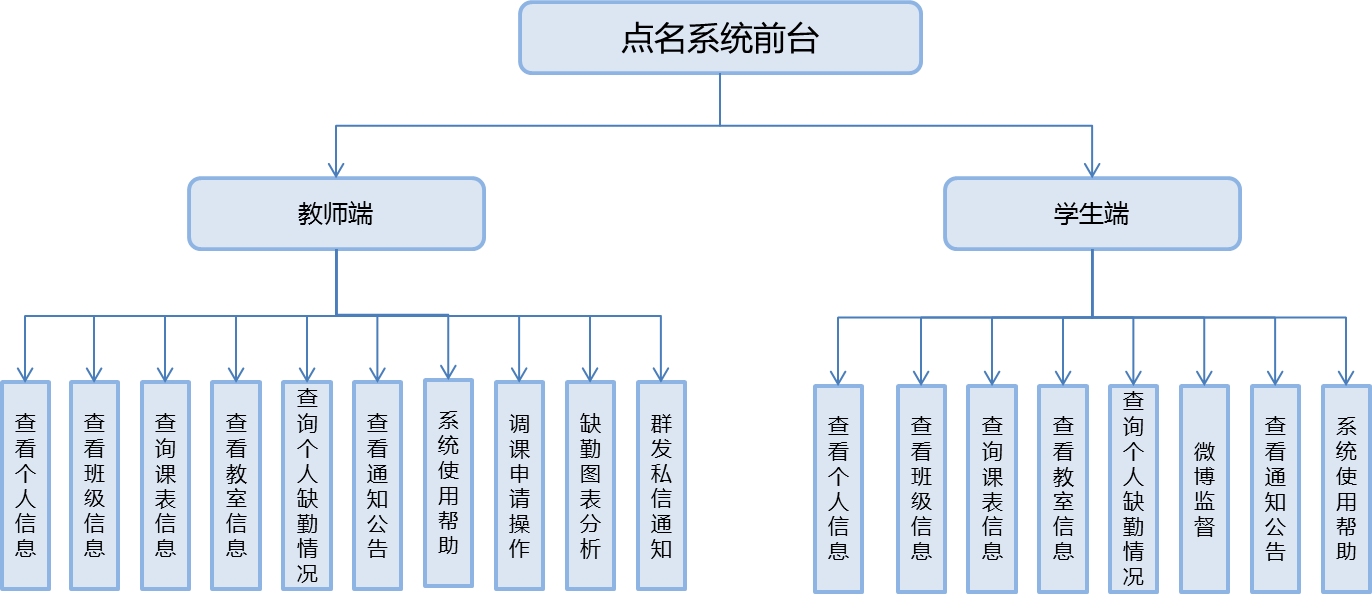
下面为“教师端”的功能说明：

1. 教师登陆后，可以查看并且修改个人资料以及登陆密码
2. 可以查看某个班级的学生信息，包括学号、姓名、性别、联系方式等等
3. 教师可以查询某个班级某节课的上课座位表
4. 教师可以查询某个班级的课表信息
5. 教师登陆后，可以进行调课申请操作，同时查看自己的申请记录
6. 教师可以查看各个教室的情况，例如“是否多媒体”、“可容纳学生数”等信息
7. 教师可以查询某个班级或者某个学生的出勤情况，同时对拍照出现异常或者其他必要的时候进行拍照更正操作，重新进行拍照取样
8. 教师可以图表分析功能，查看考勤动态、院系分析、班级分析、课程分析的图表分析情况
9. 可以发送和接收公告通知，发布对象可以是群发班级，也可以是私信某个同学
10. 通过系统帮助模块，在使用系统过程中遇到问题时可以得到技术支持

下面是“学生端”的功能说明：

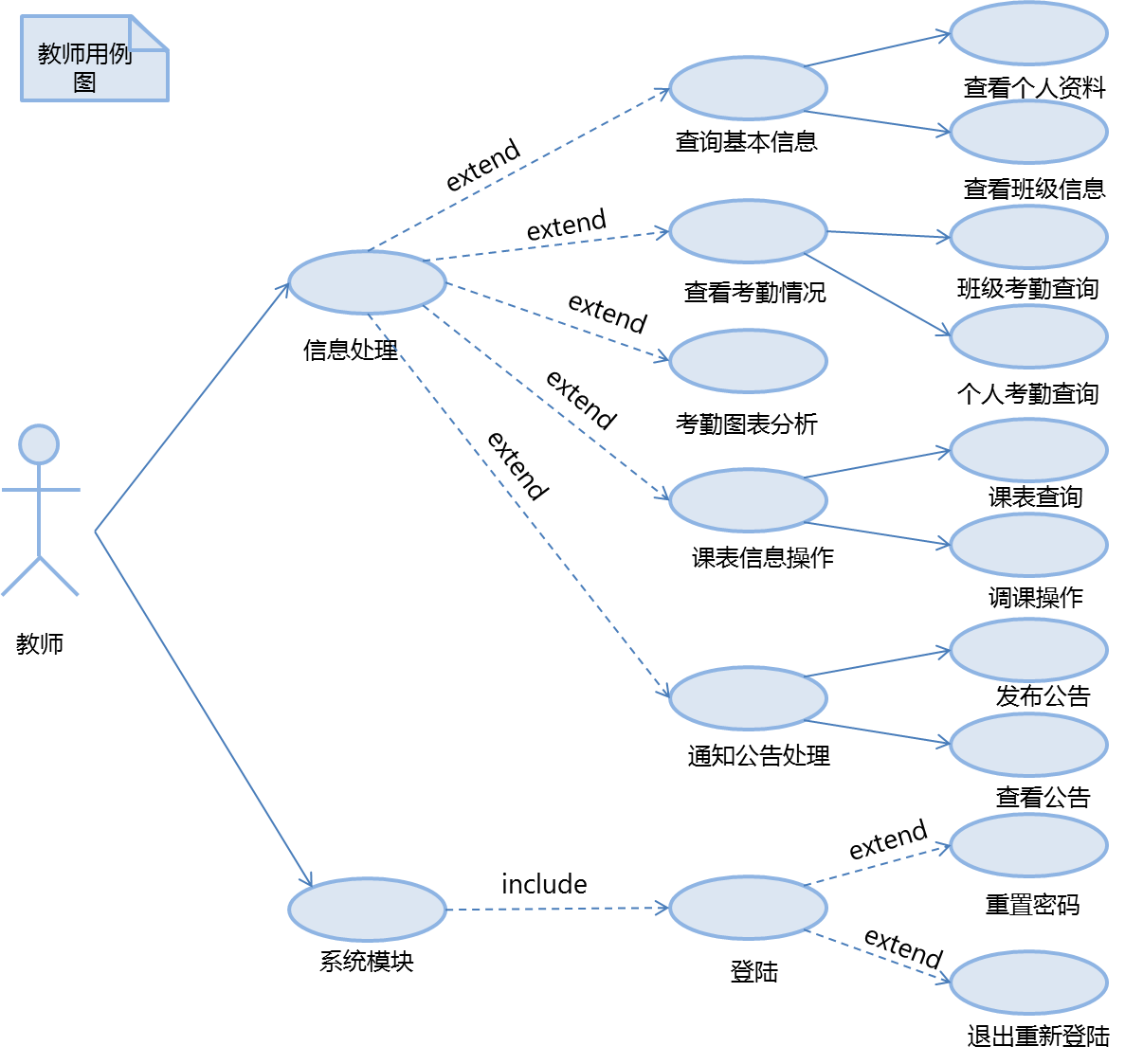
1. 学生登陆后，可以查看并且修改个人资料以及登陆密码
2. 可以查看某个班级的学生信息，包括学号、姓名、性别、联系方式等等
3. 学生可以查询某个班级某节课的上课座位表
4. 学生可以查询某个班级的课表信息
5. 学生可以查看某个教室的情况，例如“是否多媒体”、“可容纳学生数”等信息
6. 学生查询某个学生的出勤情况
7. 学生可以通过微博监督功能，设置符合条件时进行推送微博监督
8. 学生可以接收公告通知
9. 通过系统帮助模块，在使用系统过程中遇到问题是可以得到技术支持

系统结构总图如下图所示

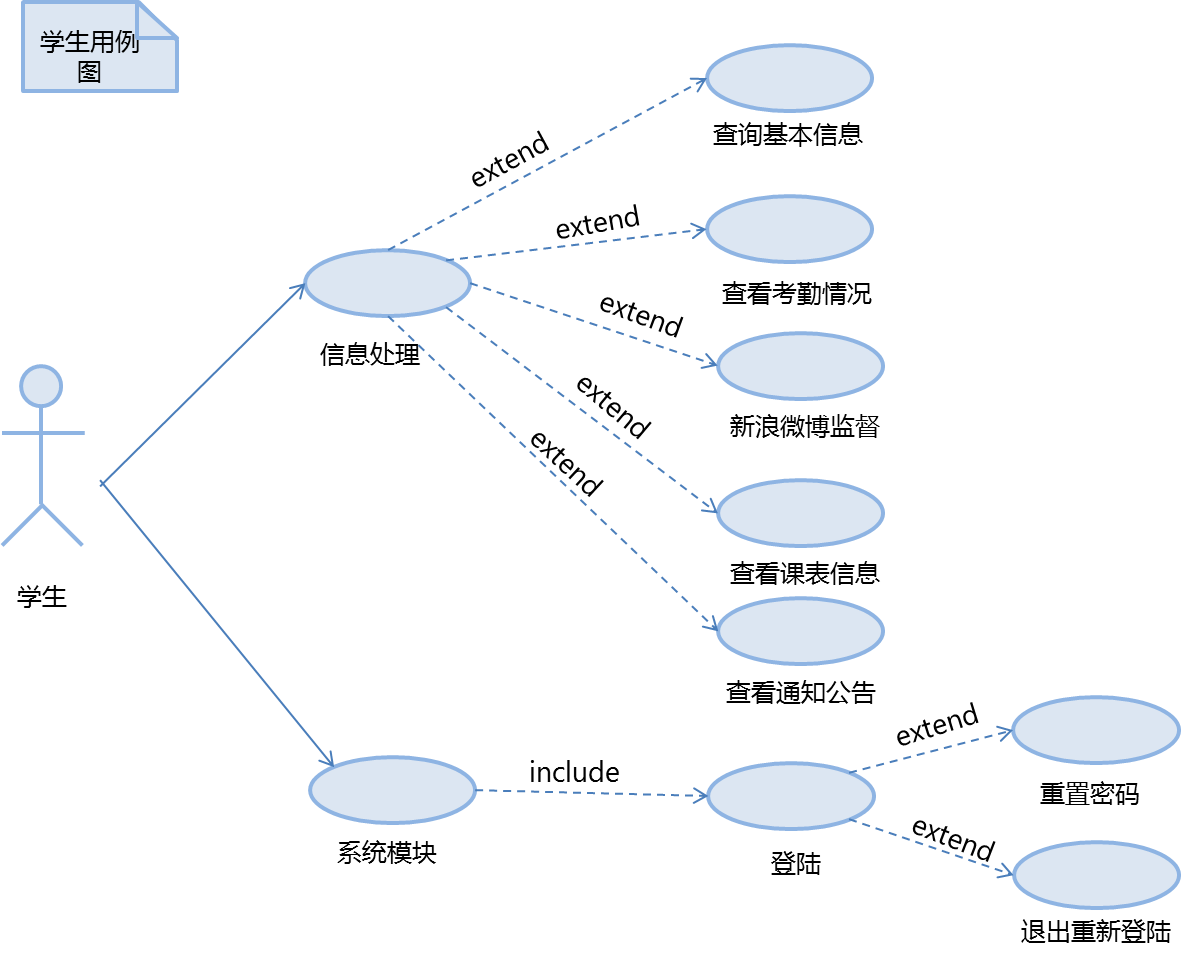


### 系统用例图

教师操作用例图如下图所示：



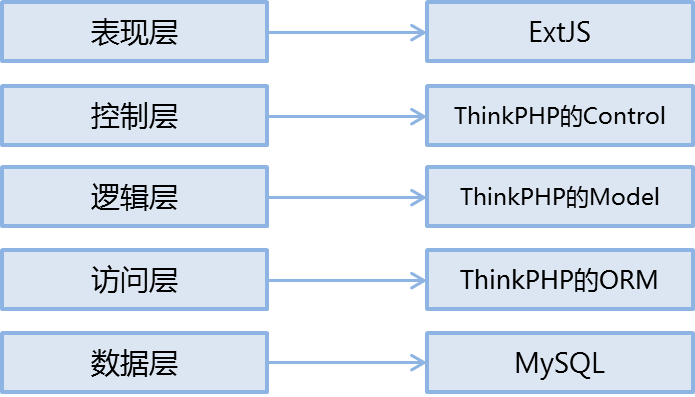
学生操作用例图如下图所示：



## 系统设计

### 系统层次划分

该系统模块中，模块系统共分为五层，分别为表现层、控制层、业务逻辑层、数据访问层、数据层，如下图所示：



表现层中，系统主要采用了开源ThinkPHP框架技术，将Action传递过来的数据传递到Model层进行一定的处理之后，再转化为JSON格式的数据反馈给前台页面。整个前台页面使用ExtJS库进行仿桌面化的开发，最终在计算机操作系统的仿真界面中展示交互的数据结果。

控制层主要是为了协调控制系统中各模块的执行流程。在控制层中，ThinkPHP框架中的Control充当了控制系统流程转向的角色。

业务逻辑层，主要是通过ThinkPHP框架中的Model层来实现具体系统的业务逻辑处理。

数据访问层，主要是采用数据库的ORM连贯操作，以组合条件的方式进行数据库的操作。

数据层存放系统中的数据，本系统采用MySQL数据库。

### 数据库设计

数据库就是存放数据的仓库，即数据按照一定的规律存放在计算机中，是信息系统的核心和基础，是信息系统开发和建设的重要组成部分。把信息系统中大量的数据按一定的模型组织起来，提供存储、维护、检索数据的功能，使信息系统可以方便、及时、准确地从数据库中获得所需的信息。数据库设计是信息资源管理最有效的手段，数据库设计就是根据具体的应用环境，构造合理的数据库模式，建立数据库逻辑结构，有效存储数据，满足用户实际需求和系统应用处理要求。在动态网站系统设计中数据库的重要性不言而喻。合理地设计数据库结构可以提高系统运行的效率。如果设计不当，查询起来就非常吃力，程序的性能也会收到影响。

数据库的设计一般主要包括两个部分：

1. 对网站系统的设计进行需求分析，确定所需要在数据库中保存的信息
2. 确定每个表的字段属性等数据库信息

根据系统功能需求分析，可以得出以下几个数据结构：管理员信息、学生信息、教师信息、班级信息、系别信息、部门信息、教室信息、科目信息、课表信息、缺勤信息、公告信息、假期信息、座位信息、换课申请信息、缺勤信息重查申请、个人敏感信息更改申请、微博监督设置。定义如下：

1. 名称：管理员信息

描述：系统后台管理员信息

定义：管理员信息=管理员编号+姓名+密码+邮箱

数据来源：后台添加

数据流向：后台操作

1. 名称：学生信息

描述：学生个人详细信息

定义：学生信息=学生编号+姓名+学号+密码+性别+照片+班级编号+系别编号+住址+邮箱+电话+监护人姓名+监护人邮箱+监护人电话+监护人与学生关系+新浪微博账号

数据来源：后台添加

数据流向：学生端操作

1. 名称：教师信息

描述：教师个人详细信息

定义：教师信息=教师编号+姓名+职工号+密码+部门编号+职称+联系电话+邮箱+照片+民族+政治面貌+性别

数据来源：后台添加

数据流向：教师端操作

1. 名称：班级信息

描述：学生所属班级信息

定义：班级信息=班级编号+班级名称+系别编号+年级

数据来源：后台添加

数据流向：前台学生所属班级记录操作

1. 名称：系别信息

描述：学生所属系别信息

定义：系别信息=系别编号+系别名称

数据来源：后台添加

数据流向：前台学生所属系别记录操作

1. 名称：部门信息

描述：教师所属部门信息

定义：部门信息=部门编号+部门名称

数据来源：后台添加

数据流向：前台教师所属部门记录操作

1. 名称：教室信息

描述：所有教室的情况

定义：教室信息=教室编号+教室名称+是否多媒体教室+最多可以容纳学生数+一排座位数+一列座位数

数据来源：后台添加

数据流向：前台教师更换上课教室查看操作

1. 名称：科目信息

描述：组成功课表的科目的信息

定义：科目信息=科目编号+科目名称+系别编号+学分

数据来源：后台添加

数据流向：创建功课表

1. 名称：功课表信息

描述：学生上课的功课表详细信息

定义：功课表信息=功课表编号+科目编号+学生编号+教师编号+教室编号+班级编号+系别编号+学年+学期+起始周+结束周+星期几+第几节课+是否单双周

数据来源：后台添加

数据流向：生成功课表

1. 名称：缺勤信息

描述：学生缺勤情况的记录

定义：缺勤信息=缺勤编号+学生编号+功课表编号+缺勤情况+是否可以申请复查+缺勤周

数据来源：视频服务器添加

数据流向：生成缺勤记录

1. 名称：公告信息

描述：教师向学生发布的通知公告消息

定义：公告信息=公告编号+公告名称+公告内容+公告教师编号+发布时间+接收方学生编号+阅读状态+是否群发

数据来源：教师发送

数据流向：学生接收

1. 名称：假期信息

描述：学生放假信息

定义：假期信息=假期编号+假期名称+学年+学期+起始周+结束周+星期几

数据来源：后台添加

数据流向：禁止视频服务器拍照采样操作

1. 名称：座位表信息

描述：学生上课时的作为记录

定义：座位表信息=座位表编号+学生编号+功课表编号+座位X坐标+作为Y坐标

数据来源：后台添加

数据流向：生成前台座位表

1. 名称：换课申请信息

描述：教师申请更换功课的时间地点

定义：换课申请信息=换课申请编号+功课表编号+教室号编号+更换到星期几+更换到第几节课+申请状态+审批说明+审核管理员姓名+申请说明标题+申请说明内容+申请教师编号

数据来源：教师在教师端添加

数据流向：后台管理员审批

1. 名称：学生缺勤重查申请

描述：学生对有疑问的缺勤情况进行申请重查记录

定义：学生缺勤重查申请=重查编号+功课表编号+教室编号+申请状态+申请重查说明+申请学生编号+审批管理员编号

数据来源：学生在学生端添加

数据流向：后台管理员审批

1. 名称：教师重查缺勤申请

描述：教师对有疑问的学生的缺勤情况进行申请重查记录

定义：教师重查缺勤申请=重查编号+功课表编号+教室编号+申请状态+申请重查说明+审批说明+审批管理员编号+需纠正的学生编号+申请教师编号

数据来源：教师在教师端添加

数据流向：后台管理员审批

1. 名称：微博监督设置

描述：设置微博监督的详细

定义：微博监督设置=微博监督编号+学生编号+是否启用发表微博功能+是否暴力发表微博+是否自定义微博内容+暴力保存多长时间+微博类型+微博内容

数据来源：学生在学生端设置

数据流向：学生考勤存在问题时推送微博

根据以上数据字典，创建符合需求的数据表下面是部分与系统相关的关键数据表结构。如下表所示，该表为student表结构，用于保存学生信息。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 逻辑名称 | 列名 | 数据类型 | 长度 | 主键 | 允许为空 | 说明 |
| 编号 | id | int | 8 | Y | N |  |
| 姓名 | name | vchar | 20 | N | N |  |
| 学号 | number | vchar | 12 | N | N |  |
| 密码 | password | vchar | 32 | N | N |  |
| 性别 | sex | int | 1 | N | Y |  |
| 照片 | image | int | 15 | N | Y |  |
| 班级编号 | classid | int | 6 | N | N | 外键class(id) |
| 系别编号 | familyid | int | 4 | N | N | 外键family(id) |
| 住址 | address | vchar | 50 | N | Y |  |
| 邮箱 | email | vchar | 50 | N | N |  |
| 联系电话 | phone | vchar | 13 | N | N |  |
| 监护人姓名 | guardername | vchar | 20 | N | N |  |
| 监护人邮箱 | guarderemail | vchar | 50 | N | Y |  |
| 监护人电话 | guarderphone | vchar | 13 | N | Y |  |
| 监护人关系 | guarderrelation | vchar | 20 | N | Y |  |
| 新浪微博 | sinaweibo | vchar | 15 | N | Y |  |

如下表所示，该表为teacher表结构，用于保存教师信息。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 逻辑名称 | 列名 | 数据类型 | 长度 | 主键 | 允许为空 | 说明 |
| 编号 | id | int | 8 | Y | N |  |
| 姓名 | name | vchar | 20 | N | N |  |
| 职工号 | number | vchar | 12 | N | N |  |
| 密码 | password | vchar | 32 | N | N |  |
| 照片 | image | int | 15 | N | Y |  |
| 部门编号 | departed | int | 4 | N | N | 外键depart(id) |
| 邮箱 | email | vchar | 50 | N | N |  |
| 联系电话 | phone | vchar | 13 | N | Y |  |
| 职称 | position | vchar | 20 | N | N |  |
| 民族 | nation | vchar | 30 | N | Y |  |
| 政治面貌 | polite | vchar | 30 | N | Y |  |
| 性别 | sex | int | 1 | N | Y |  |

如下表所示，该表为schedule表结构，用于保存由科目组成的功课表。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 逻辑名称 | 列名 | 数据类型 | 长度 | 主键 | 允许为空 | 说明 |
| 编号 | id | int | 8 | Y | N |  |
| 科目编号 | subjected | int | 8 | N | N | 外键subject(id) |
| 学生编号 | studentid | int | 8 | N | N | 外键student(id) |
| 教师编号 | teacherid | int | 8 | N | N | 外键teacher(id) |
| 教室编号 | classroomid | int | 8 | N | N | 外键classroom(id) |
| 班级编号 | classid | int | 6 | N | N | 外键class(id) |
| 系别编号 | familyid | int | 3 | N | N | 外键family(id) |
| 学年 | year | int | 4 | N | N |  |
| 学期 | term | int | 1 | N | N |  |
| 起始周 | startweek | int | 2 | N | N |  |
| 结束周 | endweek | int | 2 | N | N |  |
| 星期几 | day | int | 1 | N | N |  |
| 星期几 | time | int | 1 | N | N |  |
| 单双周 | singledouble | int | 1 | N | N |  |

如下表所示，该表为check表结构，用于保存缺勤信息。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 逻辑名称 | 列名 | 数据类型 | 长度 | 主键 | 允许为空 | 说明 |
| 编号 | id | int | 8 | Y | N |  |
| 学生编号 | studentid | int | 8 | N | N | 外键student(id) |
| 功课表编号 | schedule | int | 8 | N | N | 外键schedule(id) |
| 缺勤状态 | status | int | 1 | N | N |  |
| 是否可重查 | checkagain | int | 1 | N | N |  |
| 第几周 | week | int | 2 | N | N |  |

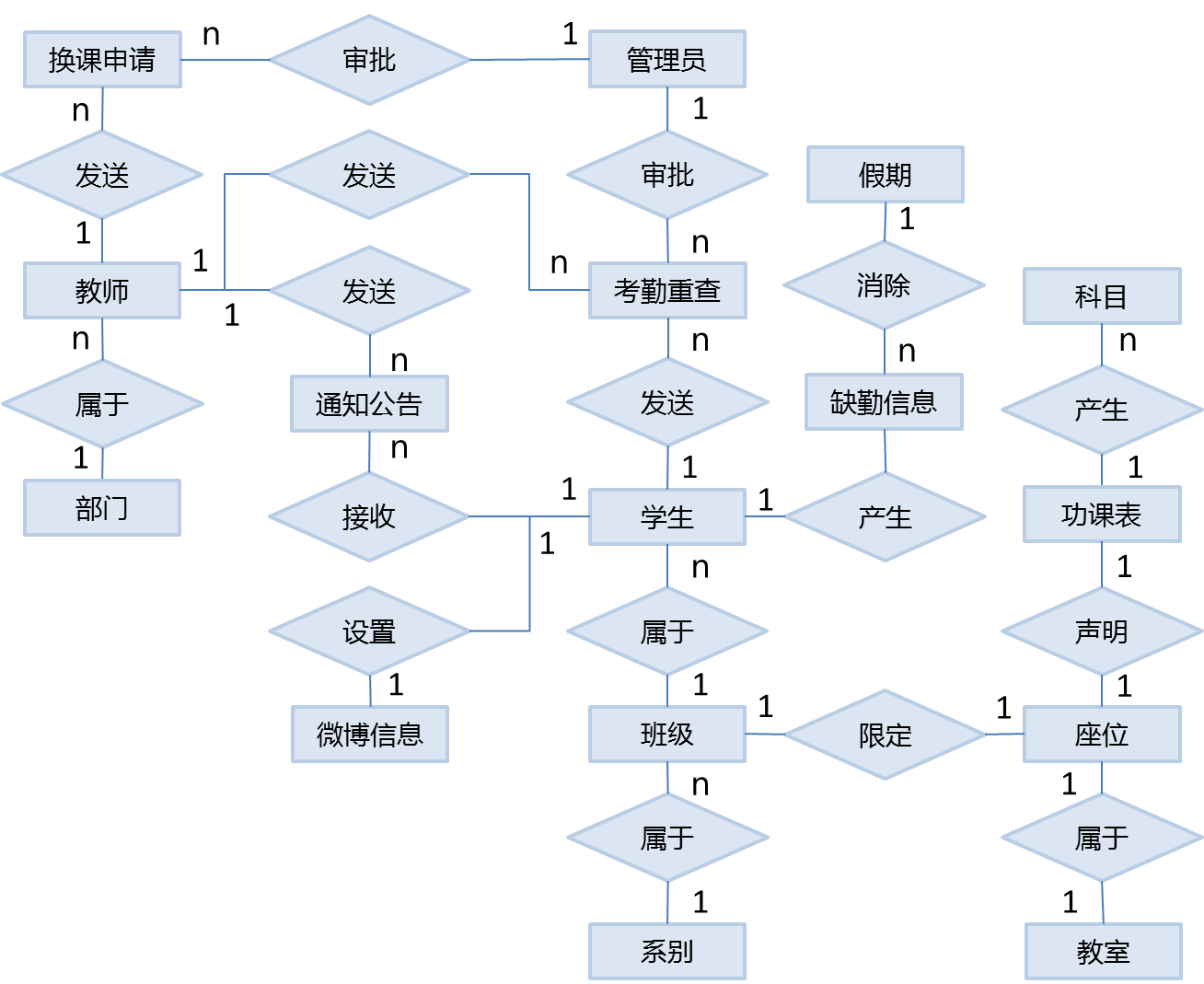
如下表所示，该表为notice表结构，用于保存公告通知信息。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 逻辑名称 | 列名 | 数据类型 | 长度 | 主键 | 允许为空 | 说明 |
| 编号 | id | int | 8 | Y | N |  |
| 公告名称 | name | vchar | 40 | N | N |  |
| 公告内容 | content | text |  | N | N |  |
| 教师编号 | teacherid | int | 8 | N | N | 外键teacher(id) |
| 发布时间 | time | int | 11 | N | N |  |
| 学生编号 | studentid | int | 8 | N | N | 外键student(id) |
| 阅读状态 | status | int | 1 | N | N |  |
| 是否群发 | isAll | int | 1 | N | N |  |

如下表所示，该表为seat表结构，用于保存座位表信息。

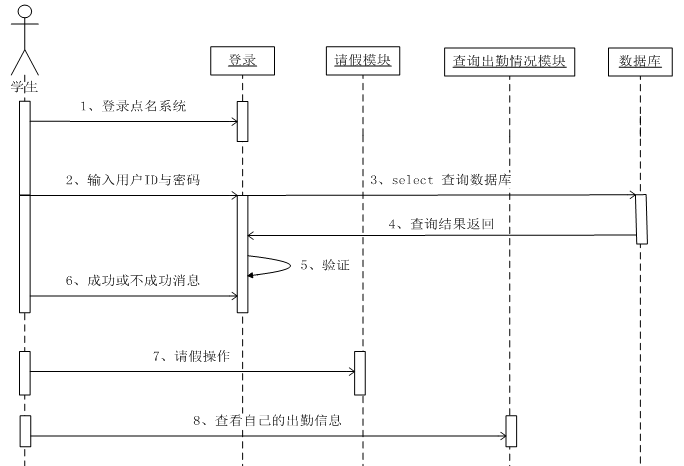
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 逻辑名称 | 列名 | 数据类型 | 长度 | 主键 | 允许为空 | 说明 |
| 编号 | id | int | 8 | Y | N |  |
| 功课表编号 | scheduleid | int | 8 | N | N | 外键schedule(id) |
| 学生编号 | studentid | int | 8 | N | N | 外键student(id) |
| 座位X坐标 | seatX | int | 3 | N | N |  |
| 座位Y坐标 | seatY | int | 3 | N | N |  |

综合以上分析和数据结构情况，构建数据库E-R图，如下图所示：



### 系统时序图

对于学生，在登录以后，点击查询出勤情况，将向数据库发出一个请求，连接成功以后，返回数据请求的结果。学生操作时时序图如下图所示：



# 第四章 系统详细设计与实现

## 系统项目的文件结构



说明：

docs：系统开发文档文件夹

home：学生端程序

----Conf：学生端程序配置文件

----Lib：学生端逻辑程序实现，包括Action和Model等的实现

----Runtime：学生端运行时的缓存文件

----Tpl：系统模板文件，下面是各个模块模板文件，可定义多套模板

login：学生端与教师端登陆程序

Public：整个系统的一些公共文件，包括JavaScript、CSS和图片等

teacher：教师端程序

temp：系统开发过程中一些临时记录文件

ThinkPHP：系统开发所用框架

upload：学生端和教师端上传文件，如头像等

config.inc.php：系统公用配置文件

index.php：学生端程序主入口文件

login.php：登陆程序主入口文件

README.md：系统开发介绍说明

teacher.php：教师端程序主入口文件

## 设计思路

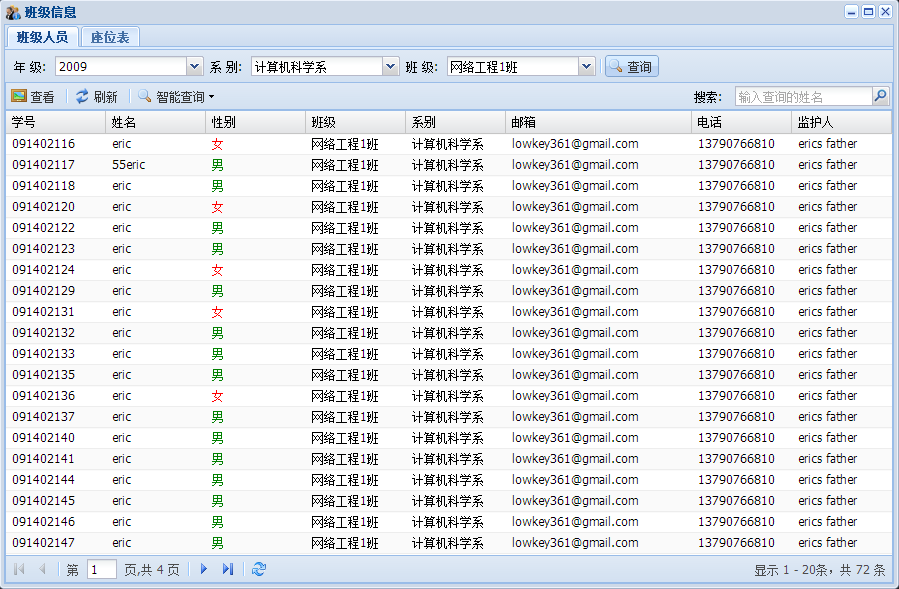
本系统定位为教室自动点名系统前台开发，主要是开发出学生与教师使用教师自动点名系统的一个交互平台。由于这样的特点，在逻辑代码上，系统采用了ThinkPHP框架进行开发，模版分离，便于后期维护修改；在界面上，系统采用ExtJS库进行开发，构建出模范操作系统界面的一个Web交互界面，在操作习惯上，也尽量与人们日常使用操作系统一致。

## 核心功能实现代码及运行效果

1. 系统登陆界面：
2. 输入学生学号或者教师教工号，选择对应的身份角色，敲击“登陆”可以登陆系统。



1. 班级人员查询界面：
2. 在条件栏中输入相应的条件，敲击“查询”即可返回相应的结果；
3. 设置智能查询时，可通过学号、姓名、班级、电话等信息查询记录；
4. 如果不输入搜索条件，则返回用户权限范围内的全部记录。



部分代码：

public function getClassmate() {

// 处理分页

$start = isset($\_GET['start']) ? $\_GET['start']:0; // 开始ID

$limit = isset($\_GET['limit']) ? $\_GET['limit']:20; // 每页条数

$query = isset($\_GET['query']) ? $\_GET['query']:''; // 检索关键字

$studentModel = M('Student');

// 是否三级联动检索

if(isset($\_GET['filterId'])) {

// 是

// 拆解联动条件，1为班级，2为年级

$filterIdArr = split(',', $\_GET['filterId']);

$classModel = M('Class');

$classinfo = $classModel->field('`id`, `name`')->where('`name` = "' . $filterIdArr[0] . '"')->select();

if(empty($query)) {

// 没有检索关键字

$classinfoid = $classinfo[0]['id'];

$info = $studentModel->query("SELECT `student`.`id`, `student`.`number`, `student`.`name`, `student`.`sex`, `student`.`classid`, `student`.`familyid`, `student`.`email`, `student`.`phone`, `student`.`guardername`, `class`.`name` as `classname`, `family`.`name` as `familyname` from `student` left join `class` on `class`.`id`=`student`.`classid` left join `family` on `family`.`id`=`student`.`familyid` where `student`.`classid`=$classinfoid limit $start, $limit");

$total = $studentModel->field('`id`, `number`, `name`, `sex`, `classid`, `familyid`, `email`, `phone`, `guardername`')->where("`classid` = " . $classinfo[0]['id'])->select();

echo '{"success":true,data:' . json\_encode($info) . ',"total": ' . count($total) . '}';

} else {

// 接收到检索关键字

$classinfoid = $classinfo[0]['id'];

$info = $studentModel->query("SELECT `student`.`id`, `student`.`number`, `student`.`name`, `student`.`sex`, `student`.`classid`, `student`.`familyid`, `student`.`email`, `student`.`phone`, `student`.`guardername`, `class`.`name` as `classname`, `family`.`name` as `familyname` from `student` left join `class` on `class`.`id`=`student`.`classid` left join `family` on `family`.`id`=`student`.`familyid` where `student`.`name` like '%$query%' and `student`.`classid`=$classinfoid limit $start, $limit");

$total = $studentModel->field('`id`, `number`, `name`, `sex`, `classid`, `familyid`, `email`, `phone`, `guardername`')->where("`name` like '%$query%' and `classid` = " . $classinfo[0]['id'])->select();

echo '{"success":true,data:' . json\_encode($info) . ',"total": ' . count($total) . '}';

}

} else {

// 否

if(empty($query)) {

// 没有检索关键字

$info = $studentModel->query("SELECT `student`.`id`, `student`.`number`, `student`.`name`, `student`.`sex`, `student`.`classid`, `student`.`familyid`, `student`.`email`, `student`.`phone`, `student`.`guardername`, `class`.`name` as `classname`, `family`.`name` as `familyname` from `student` left join `class` on `class`.`id`=`student`.`classid` left join `family` on `family`.`id`=`student`.`familyid` limit $start, $limit");

$total = $studentModel->field('`id`, `number`, `name`, `sex`, `classid`, `familyid`, `email`, `phone`, `guardername`')->select();

echo '{"success":true,data:' . json\_encode($info) . ',"total": ' . count($total) . '}';

} else {

// 接收到检索关键字

$info = $studentModel->query("SELECT `student`.`id`, `student`.`number`, `student`.`name`, `student`.`sex`, `student`.`classid`, `student`.`familyid`, `student`.`email`, `student`.`phone`, `student`.`guardername`, `class`.`name` as `classname`, `family`.`name` as `familyname` from `student` left join `class` on `class`.`id`=`student`.`classid` left join `family` on `family`.`id`=`student`.`familyid` where `student`.`name` like '%$query%' limit $start, $limit");

$total = $studentModel->field('`id`, `number`, `name`, `sex`, `classid`, `familyid`, `email`, `phone`, `guardername`')->where("`name` like '%$query%'")->select();

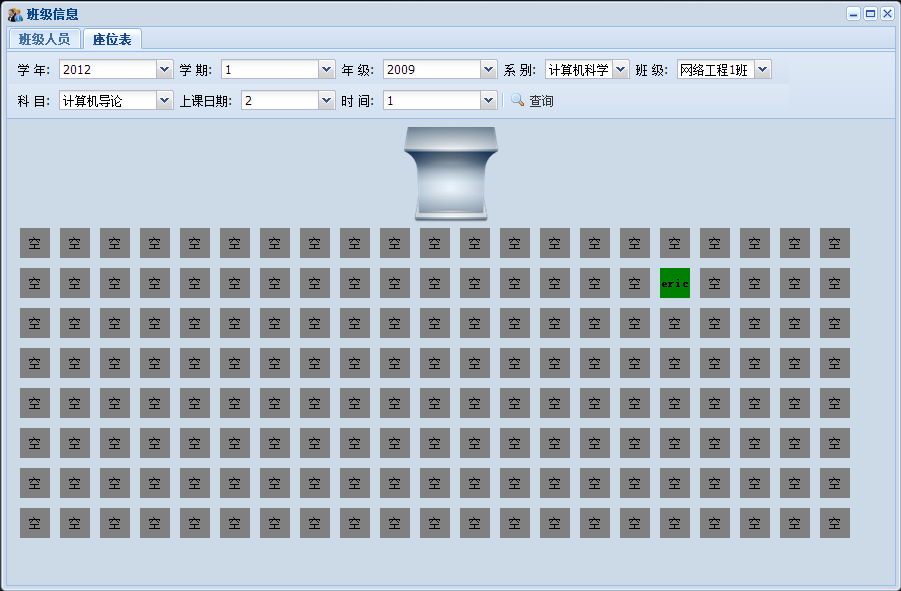
echo '{"success":true,data:' . json\_encode($info) . ',"total": ' . count($total) . '}';

}

}

}

1. 班级座位表查询
2. 在条件栏中输入相应的条件，敲击“查询”即可返回相应的结果；
3. 只能查询某个班级在某节课的座位。



部分代码：

public function getClassSeats() {

// 获取学年、学期、年级、系别、班级、科目、上课日期、时间信息

$query = $\_GET['seatQueryObj'];

if(isset($query)) {

$queryArr = split(',', $query);

// 创建座位表seat模型，找出上面查找出来的课表对应的座位

$classroomModel = M('Classroom');

$classInfo = $classroomModel->query("SELECT `classroom`.`sizex`, `classroom`.`sizey`, `classroom`.`id` AS `classroomd` FROM `classroom` WHERE `classroom`.`id` = (SELECT `classroomid` FROM `schedule` WHERE `year`=$queryArr[0] and `term`=$queryArr[1] and `classid`=$queryArr[4] and `subjectid`=$queryArr[5] and `day`=$queryArr[6] and `time`=$queryArr[7])");

// 座位数

$seatCount = $classInfo[0][sizex] \* $classInfo[0][sizey];

$seatModel = M('Seat');

$seatInfo = $seatModel->query("SELECT `seat`.`studentid`, `seat`.`seatX`, `seat`.`seatY`, `seat`.`scheduleid`, `student`.`name` AS `studentname`, `student`.`number` AS `studentnumber` FROM `seat` LEFT JOIN `student` ON `student`.`id`=`seat`.`studentid` WHERE `seat`.`scheduleid` in (SELECT `id` FROM `schedule` WHERE `year`=$queryArr[0] and `term`=$queryArr[1] and `classid`=$queryArr[4] and `subjectid`=$queryArr[5] and `day`=$queryArr[6] and `time`=$queryArr[7])");

$seatArray = array();

$count = 0;

for($i=0; $i<$classInfo[0][sizey]; $i++) {

for($j=0; $j<$classInfo[0][sizex]; $j++) {

for($k=0; $k<count($seatInfo); $k++) {

// 是否有人

if($seatInfo[$k][seatX]==$j && $seatInfo[$k][seatY]==$i) {

$seatArray[$count][name] = $seatInfo[$k][studentname];

$seatArray[$count][number] = $seatInfo[$k][studentnumber];

$seatArray[$count][has] = 'yes';

} else {

$seatArray[$count][has] = 'no';

}

// 是否需要换行

if(fmod($count+1, $classInfo[0][sizey]) == 0) {

$seatArray[$count][turn] = 'yes';

} else {

$seatArray[$count][turn] = 'no';

}

$count++;

}

}

}

// 分配模板变量

$this->assign('seatInfo', $seatArray);

$this->display();

} else {

$this->display('ClassroomInfo:noClassSeats');

}

}

1. 课表查询
2. 在条件栏中输入相应的条件，敲击“查询”即可返回相应的结果；
3. 可以查询某个班级一段时间内的课程安排。



部分代码：

public function getSchedule() {

// 获取传递的参数

$query = $\_GET['scheduleQueryObj'];

$queryArr = split(',', $query);

// 查询查询班级的课表信息

$scheduleModel = M('Schedule');

$info = $scheduleModel->query("SELECT `schedule`.`startweek` AS `startweek`, `schedule`.`endweek` AS `endweek`, `schedule`.`singledouble` AS `singledouble`, `schedule`.`time` AS `time`, `schedule`.`day` AS `day`, `subject`.`name` AS `subjectname`, `classroom`.`name` AS `classroomname` FROM `schedule` LEFT JOIN `subject` ON `subject`.`id`=`schedule`.`subjectid` LEFT JOIN `classroom` ON `classroom`.`id`=`schedule`.`classroomid` LEFT JOIN `class` ON `class`.`id`=`schedule`.`classid` WHERE `schedule`.`year`=$queryArr[0] AND `schedule`.`term`=$queryArr[1] AND `class`.`grade`=$queryArr[2] AND`schedule`.`familyid`=$queryArr[3] AND `schedule`.`classid`=$queryArr[4]");

if($info) {

$scheduleArray = array();

$count = 0;

// 检测是否课程为空

for($i=0; $i<11; $i++) {

for($j=0; $j<7; $j++) {

$has = 'no';

$k = 0;

for($k; $k<count($info); $k++) {

// 是否有课

if($info[$k][day]==$j && $info[$k][time]==$i) {

// 有课

// $scheduleArray[$count][has] = 'yes';

$has = 'yes';

break 1;

} else {

// 没课

// $scheduleArray[$count][has] = 'no';

continue;

}

}

if($has == 'yes') {

$scheduleArray[$count][has] = 'yes';

$scheduleArray[$count][startweek] = $info[$k][startweek];

$scheduleArray[$count][endweek] = $info[$k][endweek];

$scheduleArray[$count][singledouble] = $info[$k][singledouble];

$scheduleArray[$count][subjectname] = $info[$k][subjectname];

} else {

$scheduleArray[$count][has] = 'no';

}

// 是否需要换行

if(fmod($count, 7) == 0) {

$scheduleArray[$count][turn] = 'yes';

} else {

$scheduleArray[$count][turn] = 'no';

}

$count++;

}

}

// dump($scheduleArray);

$this->assign('info', $scheduleArray);

$this->display();

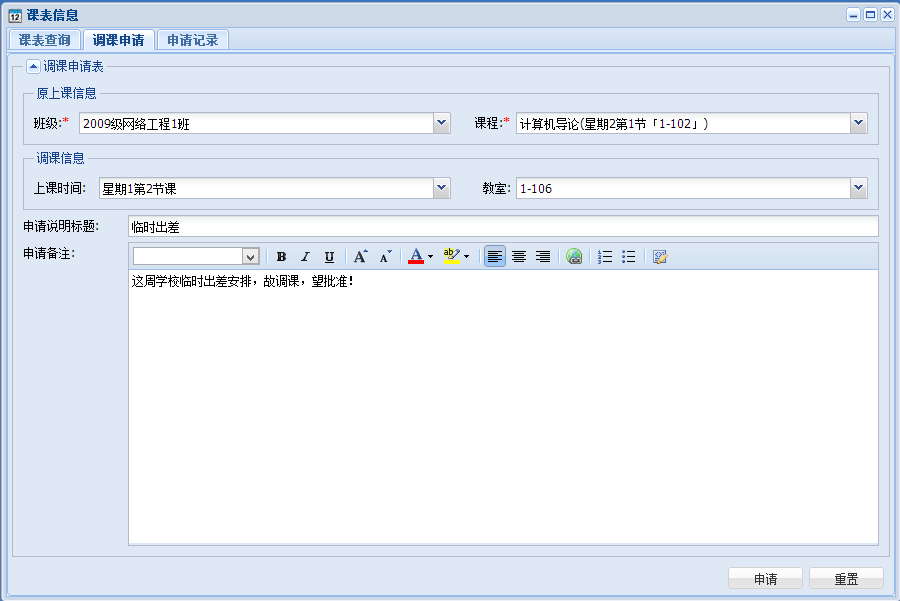
} else {

$this->display('Schedule:noSchedule');

}

}

1. 调课申请
2. 在调课申请表中选择班级、课程和调课后的上课时间、地点，同时填写申请说明，敲击“申请”，可以发送调课申请至后台审核；
3. 调课申请需后台管理员批准后生效；
4. 批准调课后，系统会以短信或者邮件的形式通知相应教师，教师也可在申请记录中查询申请记录及状态。
5. 该功能只供教师申请，学生端没有权限。



部分代码：

public function applySchedule() {

$id = session('teacher');

$applyScheduleOriginClass = isset($\_POST['applyScheduleOriginClass']) ? $\_POST['applyScheduleOriginClass'] : 0; // 班级ID

$applyScheduleOriginSubject = isset($\_POST['applyScheduleOriginSubject']) ? $\_POST['applyScheduleOriginSubject'] : 0; // 上课科目ID

$applyScheduleAfterTime = isset($\_POST['applyScheduleAfterTime']) ? $\_POST['applyScheduleAfterTime'] : 0; // 调课后的时间，格式为'星期-1-第-1-节课'

$applyScheduleAfterClassroom = isset($\_POST['applyScheduleAfterClassroom']) ? $\_POST['applyScheduleAfterClassroom'] : 0; // 调课后的课室，格式为'1-101'

$applyScheduleTitle = isset($\_POST['applyScheduleTitle']) ? $\_POST['applyScheduleTitle'] : ''; // 申请说明标题

$applyScheduleContent = isset($\_POST['applyScheduleContent']) ? $\_POST['applyScheduleContent'] : ''; // 申请说明内容

// 对传递过来的参数进行格式化处理

$timeArr = explode('-', $applyScheduleAfterTime);

$day = $timeArr[1];

$time = $timeArr[3];

$modifyclassModel = M('modifyclass');

$teacherinfo = $modifyclassModel->query("SELECT `teacher`.`id` FROM `teacher` WHERE `teacher`.`number`=$id");

$teacherid = $teacherinfo[0][id];

$info = $modifyclassModel->query("INSERT INTO `modifyclass` (`scheduleid`, `classroomid`, `day`, `time`, `status`, `teacherid`, `title`, `content`) VALUES ($applyScheduleOriginSubject, $applyScheduleAfterClassroom, $day, $time, 0, $teacherid, '$applyScheduleTitle', '$applyScheduleContent')");

// echo $modifyclassModel->getLastSql();

if($info) {

echo '{"success":true}';

}

}

1. 教室信息
2. 可以查看全校所有教室的基本情况，供调课等情况使用。



部分代码：

// 教室信息面板

createClassroom: function() {

// 是否多媒体格式转换

function videoChange(val) {

if(val==0) {

return '<span style="color:green;">是</span>';

} else {

return '<span style="color:red;">否</span>';

}

};

Ext.define('classroomModel', {

extend: 'Ext.data.Model',

fields: [{

name: 'id',

type: 'int'

}, {

name: 'name',

type: 'string'

}, {

name: 'video',

type: 'int'

}, {

name: 'people',

type: 'int'

}, {

name: 'sizex',

type: 'int'

}, {

name: 'sizey',

type: 'int'

}]

});

classroomStore = Ext.create('Ext.data.Store', {

storeId: 'classroomStore',

model: 'classroomModel',

pageSize: 20,

autoLoad: { start:0, limit:20 },

proxy: {

type: 'ajax',

url: AppUrl + '/Classroom/getClassroom',

reader: {

type: 'json',

root: 'data'

}

},

autoLoad: true

});

classroomGrid = Ext.create('Ext.grid.Panel', {

store: Ext.data.StoreManager.lookup('classroomStore'),

loadMask: true,

id: 'classroomGrid',

columns: [{

text: '教室名称',

dataIndex: 'name',

flex: 1

}, {

text: '是否多媒体',

dataIndex: 'video',

renderer: videoChange

}, {

text: '容纳学生数',

dataIndex: 'people'

}, {

text: '一排座位数',

dataIndex: 'sizex'

}, {

text: '一列座位数',

dataIndex: 'sizey'

}],

header: false,

//paging bar on the bottom

bbar: {

xtype: 'pagingtoolbar',

store: classroomStore,

displayInfo: true

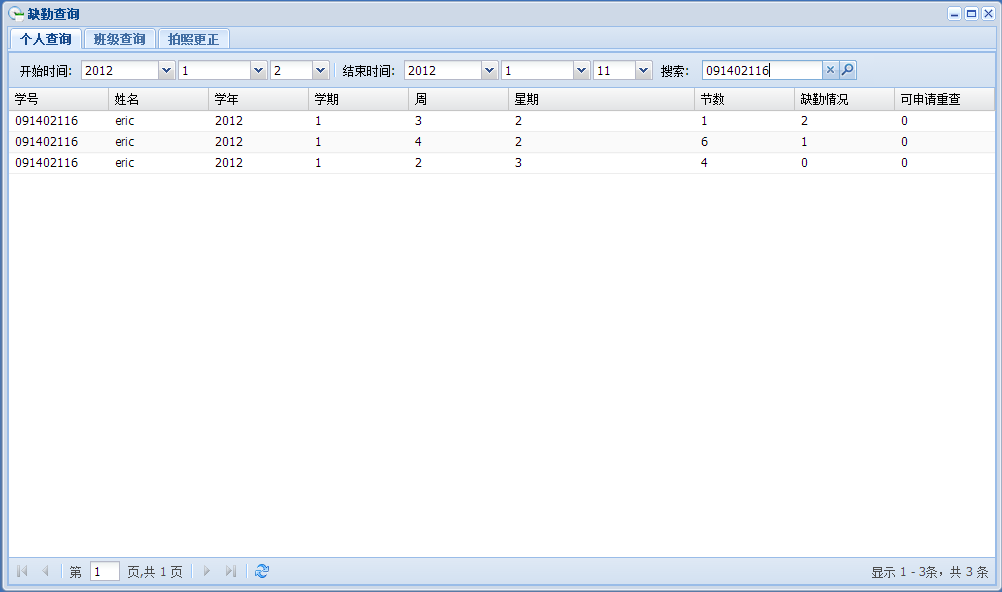
}

});

return classroomGrid;

}

1. 缺勤个人查询
2. 在条件栏中输入查询开始和结束时间，输入需要查询的学生学号，敲击“查询”即可返回相应的结果。



部分代码：

public function getPersonAttendance() {

$start = $\_GET['start']; // 开始时间参数

$end = $\_GET['end']; // 结束时间参数

$query = $\_GET['query']; // 查询关键字参数

if(!empty($query)) { // 传递了关键字参数

if(!empty($start) && !empty($end)) { // 传递了时间参数

$startArr = split(',', $start);

$endArr = split(',', $end);

$checkModel = M('Check');

$checkInfo = $checkModel->query("SELECT `student`.`number`, `student`.`name`, `schedule`.`year`, `schedule`.`term`, `schedule`.`day`, `check`.`week`, `schedule`.`time`, `check`.`status`, `check`.`checkagain` from `check` LEFT JOIN `student` ON `student`.`id`=`check`.`studentid` LEFT JOIN `schedule` ON `schedule`.`id`=`check`.`scheduleid` LEFT JOIN `class` ON `class`.`id`=`schedule`.`classid` LEFT JOIN `family` ON `family`.`id`=`student`.`familyid` WHERE `schedule`.`year`>=$startArr[0] AND `schedule`.`year`<=$endArr[0] AND `schedule`.`term`>=$startArr[1] AND `schedule`.`term`<=$endArr[1] AND `check`.`week`>=$startArr[2] AND `check`.`week`<=$endArr[2] AND `student`.`number`=$query");

if($checkInfo) {

echo '{"success":true,data:' . json\_encode($checkInfo) . '}';

}

} else { // 没有传递时间参数

$checkModel = M('Check');

$checkInfo = $checkModel->query("SELECT `student`.`number`, `student`.`name`, `schedule`.`year`, `schedule`.`term`, `schedule`.`day`, `check`.`week`, `schedule`.`time`, `check`.`status`, `check`.`checkagain` from `check` LEFT JOIN `student` ON `student`.`id`=`check`.`studentid` LEFT JOIN `schedule` ON `schedule`.`id`=`check`.`scheduleid` LEFT JOIN `class` ON `class`.`id`=`schedule`.`classid` LEFT JOIN `family` ON `family`.`id`=`student`.`familyid` WHERE `student`.`number`=$query");

if($checkInfo) {

echo '{"success":true,data:' . json\_encode($checkInfo) . '}';

}

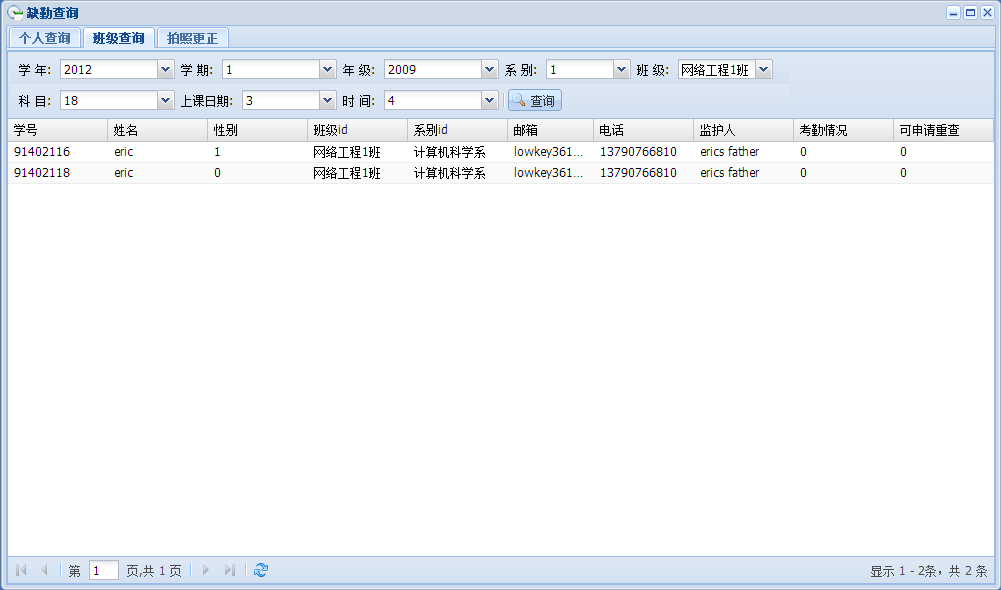
}

} else { // 没有传递关键字参数

}

}

1. 缺勤班级查询
2. 在条件栏中输入相应的条件，敲击“查询”即可返回相应的结果；
3. 该功能只限教师使用，学生端没有权限。



部分代码：

public function getClassAttendance() {

// 获取学年、学期、年级、系别、班级、科目、上课时间、时间信息

$query = $\_GET['attendanceQueryObj'];

// 判断参数是否为空

if(empty($query)) {

// 为空，说明第一次加载，没有输入信息。显示所有有缺勤的信息

$checkModel = M('Check');

$checkInfo = $checkModel->query("SELECT `student`.`name`, `student`.`number`, `student`.`sex`, `class`.`name` as classname, `family`.`name` as familyname, `student`.`email`, `student`.`phone`, `student`.`guardername`, `check`.`status`, `check`.`checkagain` FROM `check` LEFT JOIN `student` ON `student`.`id`=`check`.`studentid` LEFT JOIN `schedule` ON `schedule`.`id`=`check`.`scheduleid` LEFT JOIN `class` ON `class`.`id`=`schedule`.`classid` LEFT JOIN `family` ON `family`.`id`=`student`.`familyid`");

if($checkInfo) {

echo '{"success":true,data:' . json\_encode($checkInfo) . '}';

}

} else {

$queryArr = split(',', $query);

$checkModel = M('Check');

$checkInfo = $checkModel->query("SELECT `student`.`name`, `student`.`number`, `student`.`sex`, `class`.`name` as classname, `family`.`name` as familyname, `student`.`email`, `student`.`phone`, `student`.`guardername`, `check`.`status`, `check`.`checkagain` FROM `check` LEFT JOIN `student` ON `student`.`id`=`check`.`studentid` LEFT JOIN `schedule` ON `schedule`.`id`=`check`.`scheduleid` LEFT JOIN `class` ON `class`.`id`=`schedule`.`classid` LEFT JOIN `family` ON `family`.`id`=`student`.`familyid` WHERE `schedule`.`year`=$queryArr[0] and `schedule`.`term`=$queryArr[1] and `schedule`.`classid`=$queryArr[4] and `schedule`.`subjectid`=$queryArr[5] and `schedule`.`day`=$queryArr[6] and `schedule`.`time`=$queryArr[7]");

if($checkInfo) {

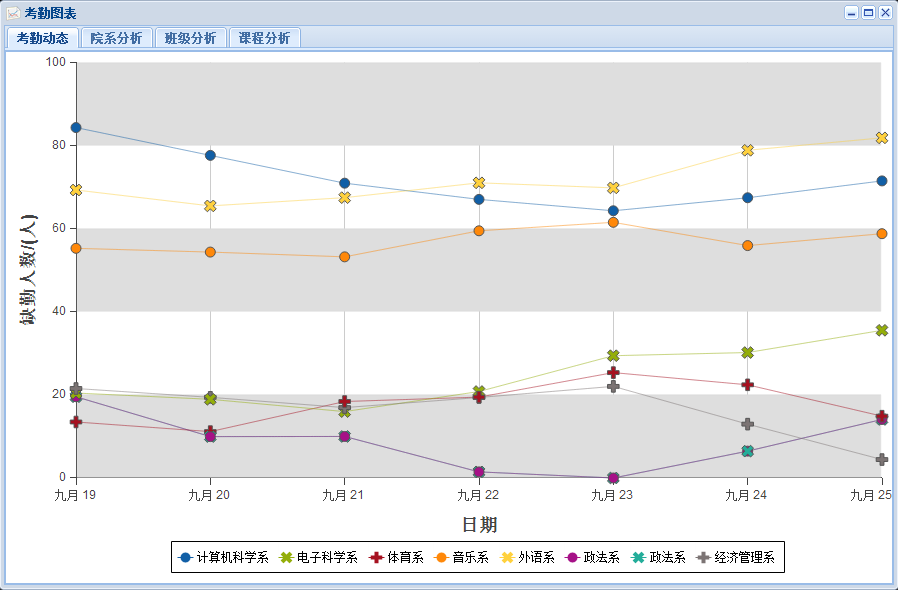
echo '{"success":true,data:' . json\_encode($checkInfo) . '}';

}

}

}

1. 考勤动态图表
2. 系统默认显示所有系每天的考勤动态，以动态折线图的方式表现；
3. 可以点击折线图或者下方对应的系别名称，切换展示或者隐藏该系的考勤动态；
4. 该功能仅教师端可以使用，学生端没有权限。



1. 考勤院系分析图表
2. 系统展示不同系考勤状态的详细缺勤情况；
3. 上方是不同系的缺勤总数柱状图，左下方是各个系各个类型具体的缺勤人数，右下方是选择对应的系别后显示的具体缺勤雷达图；
4. 可以点击上方柱状图或者左下角列表中具体的系，切换右下方展示的具体系别信息；
5. 该功能仅教师端可以使用，学生端没有权限。



部分代码：

/\*

\* 获取院系分析雷达图表数据

\* 查询某个系各个类型的缺勤情况

\*

\* \*/

public function getRadar() {

$checkModel = M('Check');

$index = $checkModel->query("SELECT `check`.`week` AS `checkweek`, COUNT('`check`.`id`') AS `count`, `schedule`.`year`, `schedule`.`term`, `schedule`.`day` FROM `check` LEFT JOIN `schedule` ON `schedule`.`id`=`check`.`scheduleid` LEFT JOIN `class` ON `class`.`id`=`schedule`.`classid` GROUP BY `schedule`.`day`, `check`.`week` ORDER BY `check`.`week` DESC");

$i = $index[$queryIndex][checkweek];

$info = $checkModel->query("SELECT COUNT('`check`.`id`') AS `Data`, `check`.`status` AS `Name` FROM `check` LEFT JOIN `schedule` ON `schedule`.`id`=`check`.`scheduleid` LEFT JOIN `class` ON `class`.`id`=`schedule`.`classid` WHERE `check`.`week`=$i GROUP BY `check`.`status`");

if($info) {

echo '{"success":true,data:' . json\_encode($info) . '}';

}

}

/\*

\* 获取院系分析Grid和bar series的共享数据集

\*

\* \*/

public function getFamilyDs() {

$checkModel = M('Check');

$info = $checkModel->query("select `schedule`.`familyid` AS `familyId`, `family`.`name` AS `familyName`, COUNT('`check`.`id`') AS `count`,

sum(case when `check`.`status`=0 then 1 else 0 end) as `kuangke`,

sum(case when `check`.`status`=1 then 1 else 0 end) as `zaotui`,

sum(case when `check`.`status`=2 then 1 else 0 end) as `chidao`,

sum(case when `check`.`status`=3 then 1 else 0 end) as `bingjia`,

sum(case when `check`.`status`=4 then 1 else 0 end) as `shijia`,

sum(case when `check`.`status`=5 then 1 else 0 end) as `gongjia`

from `check`

LEFT JOIN `schedule` ON `schedule`.`id`=`check`.`scheduleid`

LEFT JOIN `family` ON `family`.`id`=`schedule`.`familyid`

group by `schedule`.`familyid`");

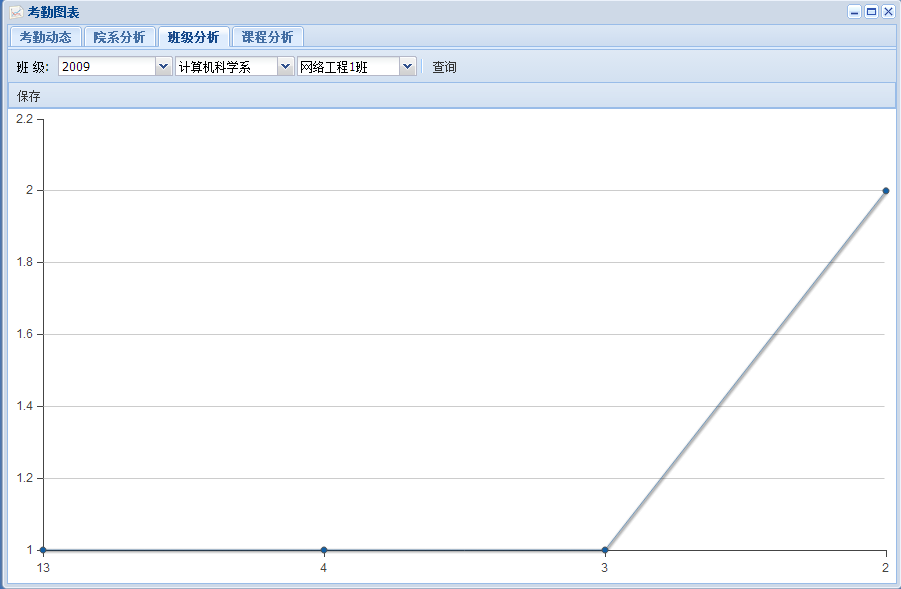
if($info) {

echo '{"success":true,data:' . json\_encode($info) . '}';

}

}

1. 考勤班级分析图表
2. 在条件栏中输入相应的条件，敲击“查询”即可返回相应的结果；
3. 可以查询不同班级一个月内每天的缺勤人数；
4. 鼠标停留在某一天的对应节点上，可以以饼图的形式展示该天各个类型的缺勤人数；
5. 该功能只有教师端可以使用，学生端没有权限。



1. 考勤课程分析图表
2. 以饼图形式展示不同课程的缺勤比例；
3. 可以以图片方式保存该柱状图，或者打印出该柱状图；
4. 该功能只有教师端可以使用，学生端没有权限。



部分代码：

/\*

\* 获取班级分析饼图数据

\*

\* \*/

public function getClassPie() {

// 获取查询条件：年级、系别、班级

$query = $\_GET['classQueryObj'];

$queryArr = split(',', $query);

// 获取查询条件：选择的节点

$queryIndex = $\_GET['index'];

$checkModel = M('Check');

$index = $checkModel->query("SELECT `check`.`week` AS `checkweek`, COUNT('`check`.`id`') AS `count`, `schedule`.`year`, `schedule`.`term`, `schedule`.`day` FROM `check` LEFT JOIN `schedule` ON `schedule`.`id`=`check`.`scheduleid` LEFT JOIN `class` ON `class`.`id`=`schedule`.`classid` WHERE `class`.`grade`=$queryArr[0] AND `schedule`.`familyid`=$queryArr[1] AND `schedule`.`classid`=$queryArr[2] GROUP BY `schedule`.`day`, `check`.`week` ORDER BY `check`.`week` DESC limit 0, 30");

$i = $index[$queryIndex][checkweek];

$info = $checkModel->query("SELECT COUNT('`check`.`id`') AS `absenceCount`, `check`.`status` AS `absenceName` FROM `check` LEFT JOIN `schedule` ON `schedule`.`id`=`check`.`scheduleid` LEFT JOIN `class` ON `class`.`id`=`schedule`.`classid` WHERE `class`.`grade`=$queryArr[0] AND `schedule`.`familyid`=$queryArr[1] AND `schedule`.`classid`=$queryArr[2] AND `check`.`week`=$i GROUP BY `check`.`status`");

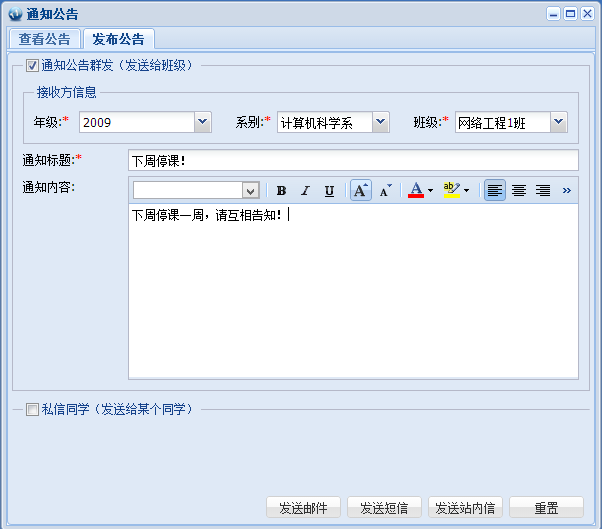
if($info) {

echo '{"success":true,data:' . json\_encode($info) . '}';

}

}

1. 发布通知公告
2. 可以选择“通知公告群发”或者“私信同学”，切换通知公告发送给具体某个同学还是发送给一个班级若干人；
3. 系统可以以邮件、短信或者系统站内信的方式发送给接收方；
4. 在条件栏中选择接收方信息，输入通知内容，敲击对应的发送方式，可以将通知内容发送给选择的接收方；
5. 该功能只有教师端可以使用，学生端没有权限。



部分代码：

/\*

\* 保存通知公告

\*

\* \*/

public function saveNotice() {

// 获取传递的参数

$classNoticeCheckbox = $\_POST['classNotice-checkbox']; // 选择班级通知公告

$studentNoticeCheckbox = $\_POST['studentNotice-checkbox']; // 选择私信通知公告

if(isset($classNoticeCheckbox)) {

// 班级通知

// 获取传递的参数

$grade = $\_POST['grade']; // 班级通知公告年级

$familyid = $\_POST['familyid']; // 班级通知公告系别ID

$classid = $\_POST['classid']; // 班级通知公告班级ID

$noticeClassTitle = $\_POST['noticeClassTitle']; // 班级通知公告标题

$noticeClassContent = $\_POST['noticeClassContent']; // 班级通知公告内容

// 查询所选择的班级的学生ID

$studentModel = M('Student');

$studentInfo = $studentModel->query("SELECT `student`.`number` AS `id` FROM `student` LEFT JOIN `class` ON `class`.`id`=`student`.`classid` WHERE `class`.`id`=$classid");

// 添加通知公告到每个查询出来的学生

$noticeModel = M('Notice');

$sqlStatus = true;

$time = time();

$teacherid = session('teacher');

for($i=0; $i<count($studentInfo); $i++) {

$studentId = $studentInfo[$i]['id'];

$info = $noticeModel->query("INSERT INTO `notice` (`name`, `content`, `teacherid`, `studentid`, `time`, `status`, `isAll`) VALUES ('$noticeClassTitle', '$noticeClassContent', '$teacherid', '$studentId', '$time', '0', '0')");

if(!$info) {

$sqlStatus = false;

}

}

if(!$sqlStatus) {

echo '{success:true}';

} else {

echo '{success:false}';

}

} else {

// 个人通知

// 获取传递的参数

$noticeStudentNumber = $\_POST['noticeStudentNumber']; // 私信通知公告学生学号

$noticeStudentTitle = $\_POST['noticeStudentTitle']; // 私信通知公告标题

$noticeStudentContent = $\_POST['noticeStudentContent']; // 私信通知公告内容

// 添加通知公告到指定学号的学生

$noticeModel = M('Notice');

$time = time();

$teacherid = session('teacher');

$info = $noticeModel->query("INSERT INTO `notice` (`name`, `content`, `teacherid`, `studentid`, `time`, `status`, `isAll`) VALUES ('$noticeStudentTitle', '$noticeStudentContent', '$teacherid', '$noticeStudentNumber', '$time', '0', '1')");

if(!$info) {

echo '{success:true}';

} else {

echo '{success:false}';

}

}

}

1. 微博应用爱监督
2. 首次进入该模块时，需要用户进行微博登陆授权该应用；
3. 可以切换监督状态选择是否接受监督，或者切换强制监督模式选择是否接受强制监督，输入监督内容，敲击确定可以设定；
4. 监督模式开启时，当该学生出现缺勤情况时，会自动调用新浪微博接口，推送设定好的微博内容到新浪微博；
5. 强制监督模式时，需要设定“强制时限”，在此时限前，学生无法取消监督；
6. 该功能仅学生端可以使用，教师端没有权限。



# 第五章 总结与展望

## 课题总结

经过这段时间的努力，这套教室自动点名系统前台的构思、设计以及实现，基本完成。通过这次毕业设计，在以下几方面有了自己的认识与体会：

1. 和以前相比，通过一个专利性的产品，自己更会站在产品的角度去思考产品的发展。在自己开发中考虑过后，才将自己的想法付诸实现于系统中。从这方面看，自己已经慢慢地不再停留在一个编码人员的层次上了。
2. 通过版本管理软件，合理部署自己的文件结构，对系统开发和日后维护有很大的帮助。
3. 在开发中能够不断接收工作中学习到的新技术，快速掌握并运用到毕业设计中来。

## 进一步开发的展望

在本系统的开发过程中，由于时间条件的关系，并没有太多精力去开发。我对这款产品还有一些觉得可以做的更好的地方：

1. 教室自动点名系统在这次毕业设计中，由于他人原因，并没有能够形成一个整体的点名系统应用，只是目前的教师端和学生端的实现，缺少管理员后台一些数据的支撑。所以后期如果有时间或者以后的师弟师妹能够接着做这个项目的话，希望能够将管理员平台添加上去，形成一个完整的管理系统。
2. 教室自动点名系统前台开发由于时间仓促，没有进行系统测试，所以在使用中会存在一些bug。目前仅仅对我已知存在的bug进行了修复。如果在正式投入使用前，应该做一次完整的系统测试。对测试得出的bug进行调整，或者联系我这边重新进行调整。
3. 教室自动点名系统目前局限在Web平台，但是由于移动终端的兴起，我觉得一个产品的发展，不能单单停留在一个平台上面，而必须支持Web、Android、IOS等各个平台。这次毕业设计原本打算将系统引入Android和IOS平台的开发，无奈时间不足，代码量过多，最终先完成了Web端的开发。不过我在这次开发过程中也体现了我希望引入手机终端的一些想法。同样，希望之后的师弟师妹有精力可以完善这部分的开发。

# 第六章 参考文献

1. 《深入浅出Ext JS》，徐会生 康爱媛 何启伟著，人民邮电出版社
2. 《PHP and MySQL Web Development》, Luke Welling and Laura Thomson(Author), Addison-Wesley Professional(Publisher)
3. 《MySQL Cookbook》, Paul DuBois(Author), O'Reilly Media(Publisher)
4. 《JavaScript Patterns》, Stoyan Stefanov(Author), O'Reilly Media(Publisher)

# 第七章 致谢

在此向所有在这次课程设计中给予我关心、支持和帮助的老师、朋友们，向你们致以最诚挚的谢意。

首先，我要感谢彭刚老师，在大学期间是他给我提供了实验室的环境，使我在这个环境中提早接触了Web开发，给这次毕业设计的开发实现打下了编程基础。同时，他的细心和严谨也是我一直学习的品质，他渊博的专业知识和亲切的待人方式都给我留下了深刻的印象。不论在学习和生活中，大学期间他都给了我巨大的提高和帮助。

接着，我要感谢我的同学们和同事们，感谢他们在生活中对我的帮助，也感谢他们给我提出那么多的建议和指导性意见，帮助我顺利完成我的毕业设计。

最后，我还要感谢我的家人，感谢他们给我提供经济和精神资助，让我顺利完成大学学业。特别要感谢来自家庭成员的支持。

# 第八章 附录

## 外文翻译

### Original (Excerpt from *Pro PHP: Patterns, Frameworks, Testing and More*)

**MVC Architecture**

**M**odel-View-Controller (MVC) is a design pattern that simplifies application development and maintenance. It achieves this by separating the application into three logical components:

*Model*: The model layer is responsible for the business logic of an application. It will encapsulate access to data stores and will provide a reusable class library. Typically, within the model, you will find facilities for database abstraction, e-mail delivery, validation, and authentication.

*View*: The view layer is typically what would be considered web design, or templating. It controls the look and feel of data and provides facilities to collect data from the user. Technologies exclusively found in the view are HTML, CSS, and JavaScript.

*Controller*: The controller layer glues everything together and merges the styling of the view with the functionality of the model. It is responsible for collecting input data from the view and deciding program execution. The controller will call model facilities and interpret the returning data so that it can be rendered by the view. It is also responsible for all application exception and flow control.

This chapter introduces the MVC pattern and its advantages. In the final section, you will create a minimal MVC framework to see how such a framework actually works.

**Why Use MVC?**

You might be wondering why MVC is so popular within web application development. While there are many compelling reasons to use an MVC framework—from cleanliness of code to ease of upgrading—the number one reason companies choose MVC development is the development workflow that it encourages.

In teams, MVC allows for efficient collaboration by splitting up responsibilities into three main roles:

*Development*: The developers are programmers working on the model. They will typically have skills related to PHP, database administration, algorithms, architecture, and data validation. This role will typically be responsible for the programming details (the “how”) of an application, and will provide APIs and enforce policy for interacting with data.

*Design*: The designers handle the view and are responsible for how an application looks and feels. They will be highly skilled in technologies like HTML, CSS, JavaScript, and graphic design. Typically, this role will be responsible for interacting with both internal and external communication sources to determine realistic business rules for which an application will be developed or improved. Design usually leads to prototype development, creating mock-up designs showing ideal functionality.

*Integration*: The integration exists in the controller layer, which glues the work of designers and developers together. Integrators will typically have less experience than developers, and will be responsible for cutting up static templates and making the required regions dynamic. They are also responsible for brokering data from request sources. They will take request information from forms, pass the information to the model, interpret the results, and pass the information along to the view.

By providing division of responsibilities, the details of *how* are separated from the details of *why*, which allows the creative and technical disciplines to interact more easily.

In a typical workflow, first a designer will put together a static, stand-alone, prototype design based on the existing business requirements and how the application should behave. The designers will then pass the design to the developers for review.

Developers are responsible for ensuring that all of the requested functionality is feasible and fits with the company’s security and privacy policies. If everything checks out, a development plan, including a mock-up API, will be created and passed to the integration role. If there are any issues with the prototype, the project will go back to the designer, and the process will continue again from the start.

Once the design prototype and the mock-up API are in place, the integration work begins. The integrators will dissect the prototype design and convert it into a template language like plain PHP, Smarty, and so on. This will enable the design to handle dynamic data, and will integrate the site URL structures for which the integrator role is solely responsible. Then a controller component is written to broker requests (forms, URL parameters, cookies, and so on) between the web server and the model information they are meant to manipulate. When a result is produced, the view will be retrieved and rendered, with the controller providing the values for any dynamic content.

Finally, when the mock-up API is fully implemented, the application can be handed off for testing. If the integrator wrote the controller to spec, and the model developers conformed to their API, then everything should work as expected.

As well as distributing responsibility between team members, the MVC architecture also offers several other key benefits. One is that it allows the presentation layer to be modified and updated at any time, as long as it does not break the contract with the controller. This means that copy editing or theme changes can be readily accomplished without changing the way the application works or sorting through a lot of PHP code.

The MVC architecture also provides a useful subdivision of the file structure. Each MVC framework has a different structure and layout, but with few exceptions, they all provide separate files for models, views, and controllers. This feature will reduce code conflicts and will save time that would otherwise be spent merging changes in your revision control system.

**MVC Application Layout**

The layout for MVC applications differs by framework; however, some concepts are common to all MVC frameworks. In the next chapter, you will learn about the specifics of the Zend Framework layout. The explanations in this section refer to the typical implementation of the MVC pattern in PHP.

**From the Web Server**

Unlike other PHP applications, MVC sites typically implement a centralized PHP script that handles all requests on a web site. So, instead of the browser going to /*path*/*to*/*somefile*.php, the browser goes to /*controller*/*action*. However, you won’t find a folder named controller or a file named action. Instead, a URL rewriter like mod\_rewrite is used to redirect *all* HTTP requests to a centralized script that is called the *bootstrap file*.

The bootstrap file is responsible for initializing the framework: loading files, reading configuration data, parsing the URL into actionable information, and populating the objects that encapsulate the request. Finally, it is responsible for initializing the controllers.

**Actions and Controllers**

After bootstrapping, a class called the front controller (FC) is instantiated and takes over. The front controller is typically a built-in class that is responsible for interpreting request variables and routing code execution to a user-defined class, which is called an action controller (AC). Usually, an action controller will implement a standard interface or descend from an abstract class, so that the front controller and action controller may interact using a common API.

The front controller will then invoke a method on the action controller specified by the URL. This method is called an *action*, and its name, like the action controller’s name, is determined from the URL. The action is responsible for all the dirty work. In this method, you will instantiate model classes, parse view templates, and output the results.

**Models**

The models are typically the easy part. They usually don’t follow any particular structure other than to exist in a common location, so that they may be automatically loaded by the framework. Models are simply utility classes that provide the required data manipulation and parsing functionality.

**Views**

The views are templates and can be written in any template language. The primary goal here is to not include any sort of processing logic or data manipulation in the view, and output only data that has been provided specifically by the controller. Typically, this means limiting yourself to displaying arrays, not data objects, when operating in the view.

The choice of templating language is your own, depending on whether it can be integrated with your framework. Personally, I find that plain PHP makes for the best templating language when working in MVC environments. FastTemplate and Smarty are viable second choices.

**Criteria for Choosing an MVC Framework**

Every MVC framework has certain advantages and certain downsides. You should consider five key factors when choosing a framework to use: architecture, documentation, community, support, and flexibility. I suggest that you sit down and score all the frameworks you are considering using these criteria, and then pick the most appropriate one for you.

**Architecture of the MVC Framework**

Every developer is different. What makes sense to one developer might be totally illogical to another. You should choose a framework that you understand. I have met far too many distressed developers using a complicated MVC framework because they have heard about all the benefits it may provide. I assure you that without an architectural understanding of your framework, it will hinder, not help, your development. Choose one that you are comfortable using.

You should also consider whether you want a framework that follows a linear function and callback style or an object-oriented approach. Personally, I prefer the latter because the concept of encapsulation really works wonders with model classes.

You will need to understand the framework’s technical requirements and file structure, so that you can be sure it will work effectively with your deployment procedures. For example, some frameworks have very backward-compatible APIs, whereas others do not even attempt to maintain backward compatibility. There are valid arguments for both approaches. Choose the one that works for your organization.

**MVC Framework Documentation**

All frameworks come with documentation, and that’s where the similarities end. Documentation can be as simple as an install file, or as complex as formal training courses.

When I look at a framework’s documentation, I ask three key questions:

• Is it current? Documentation is useless if it’s talking about a version that you are not using. Ensure that the version you download is the version that is documented.

• Is it clear? Is the documentation well written and easy to understand? If you can’t understand what it is trying to say, move on.

• Is it complete? Is the documentation comprehensive? Does it cover everything in the framework or just the basics? All too often, you will get a combination of great introductory documentation and machine-generated advanced docs.

Good docs are essential to meeting deadlines and keeping costs down.

**MVC Framework Community**

The existence of community around an MVC framework is critical. You should look at what amount of participation is welcomed, and how long it takes members of the community to respond to a bug and produce a patch.

Community is also the number one factor managers will need to be concerned with, as it will heavily affect their ability to hire new staff members that are knowledgeable with the framework.

You should be able to communicate directly with the developers of your framework to report bugs, suggest features, and obtain fixes. However, the framework developers should *not* be considered technical support, which is an entirely separate, though related, topic.

The time between releases will give you an idea of how active the project is. This usually gives a good indication of the frequency of future updates. You will need to evaluate how fast your business can reasonably adapt to a changing framework. There are definite trade-offs between API stability and the introduction of new features, so your mileage may vary.

**MVC Framework Support**

As I noted in the previous section, the developers of your framework are not support. If you send them so much as a single e-mail with an “ignorance” question, you will be highly unlikely to receive future assistance that only they can provide. Consider this “crying wolf” to the development team.

So with that in mind, who can you ask when you have a support question? Some MVC frameworks are supported through a support-revenue model, where the framework is free, but support will cost you. Others offer no support at all; in those cases, you will need to find a knowledgeable third party.

Mailing lists and support chat channels will provide free support for a given framework. These sources can be highly useful, but may also prove highly unreliable compared with paid support. To get the best results from these sources, a share-and-share-alike attitude must be adopted—those with the most to offer should also get the most help.

Before beginning development with any particular framework, be sure you fully understand your support position. Getting stuck on problems for which you cannot quickly find the answer can be very costly.

**MVC Framework Flexibility**

How flexible is the MVC framework’s implementation? Can you integrate your existing applications within the framework, and make sure it supports your databases?

Some frameworks are incredibly rigid and will place many restrictions on how you write and organize your code. They may dictate templating languages, and they may have very specific syntax that will incur a significant training cost.

Look for a flexible framework that imposes as few hard limits as possible.

**Roll Your Own MVC Framework**

It is generally recommended to choose from one of the publicly available frameworks, rather than creating your own. No matter how good a programmer you are, you will always flunk in the community department. However, to truly understand MVC, it helps to know how to write your own basic framework, so that’s what you’ll do now.

In this example, you will follow a layout similar to the Zend Framework, utilizing techniques you have learned in prior chapters and implementing a typical front controller/action routing system.

For this example, you will need a web server running Apache 2, PHP 5, mod\_rewrite, and a database like PostgreSQL. The instructions in this chapter are for a Debian Linux installation, but they will work on all PHP-supported platforms.

■**Note** If you don’t have a server with the required configuration, all the necessary applications can be downloaded to your desktop. Their installation is not complicated. Dive right in, and you’ll more than likely be able to figure it out.

**Setting Up a Virtual Host**

First, you need to create a virtual host and a new site. Follow these steps:

**1.** Execute the following commands to create a virtual host:

> mkdir -p /usr/local/www/*yourdomain*.com/*document\_root*

> cd /etc/apache2/sites-available

**2.** Edit a file called *yourdomain*.com.

> pico –w *yourdomain*.com

**3.** Place the virtual host definition in the *yourdomain*.com file.

<VirtualHost \*>

*DocumentRoot* /usr/local/www/*yourdomain*.com/*document\_root*

*ServerName yourdomain*.com

</VirtualHost>

**4.** Execute the following commands to activate your site:

> a2ensite *yourdomain*.com

> /etc/init.d/apache2 reload

**5.** Make your browser access your site. You can do this by pointing the Domain Name System (DNS) for the domain at the server or by editing your hosts file. To edit your hosts file on Windows, locate the file at C:\WINDOWS\system32\drivers\etc\hosts. Open it in a text editor and add a new line. Type the IP address of your server, press Tab, and then type the name of the file (*yourdomain*.com) you just created. Save the file.

**6.** Open a new browser instance (if you have a browser open, it may need to be closed first to pick up the hosts file change) and browse to your new site. You should see an empty directory index.

You’re now ready to start developing your very own MVC framework.

**Creating an MVC Framework**

At this point, you will need to start creating directories. I recommend the following layout:

> cd /usr/local/www/yourdomain.com

> mkdir document\_root/images document\_root/styles

> mkdir –p application/models application/views \

> application/controllers

> find

.

./document\_root

./document\_root/images

./document\_root/styles

./application

./application/models

./application/views

./application/controllers

Next, create a placeholder for your bootstrap file, as shown in Listing 14-1.

**Listing 14-1.** *Creating the Bootstrap index.php*

> pico document\_root/index.php

Hello, World!

At this point, reloading your browser should result in “Hello, World!” being shown. Navigate to http://yourdomain.com/nonexistent. You should get a 404 File Not Found error.

For this MVC framework to work, you need every single request to hit this index.php. To do this, you enlist mod\_rewrite through an .htaccess file, as shown in Listing 14-2.

**Listing 14-2.** *Using mod\_rewrite in an .htaccess File*

> pico document\_root/.htaccess

RewriteEngine on

RewriteRule !\.(js|gif|jpg|png|css)$ index.php

Reload your browser, and you should see “Hello, World!” If you don’t, you will need to ensure that your server has mod\_rewrite available and does not restrict .htaccess overrides with a nonstandard allowOverrides directive.

**Bootstrapping**

The next stage in this process is to create a bootstrapping script in your index.php file. This file will be responsible for including files and initializing the front controller. Most MVC frameworks will use SPL autoloading mechanisms to load files, but for simplicity, you will just require\_once for now. Create the file as shown in Listing 14-3.

**Listing 14-3.** *The Bootstrap Loader*

> pico –w document\_root/index.php

<?php

//Require Components

require\_once('../application/models/front.php');

require\_once('../application/models/icontroller.php');

require\_once('../application/models/view.php');

//Require Controllers

require\_once('../application/controllers/index.php');

//Initialize the FrontController

$front = FrontController::getInstance();

$front->route();

echo $front->getBody();

■**Note** You do not need a closing ?> in Listing 14-3, as the end of file will close the tag automatically. It is not recommended to include closing tags, as newlines at the end of the file can result in premature output and may cause issues sending HTTP headers.

**The Front Controller**

The front controller’s job is to parse the URL and to instantiate the controller and invoke the action method. This is a singleton class (see Listing 3-1 in Chapter 3), which is responsible for the application’s program flow.

The theory of operation is that URLs follow the format /controller/action/key1/value1/. . . . Once the URL has been parsed, you will use the reflection API (covered in Chapter 7) to invoke the action method on the IController implementing class.

To get started, first create the FrontController class, as shown in Listing 14-4.

**Listing 14-4.** *The FrontController Class*

> pico –w application/models/front.php

<?php

class FrontController {

protected $\_controller, $\_action, $\_params, $\_body;

static $\_instance;

public static function getInstance() {

if( ! (self::$\_instance instanceof self) ) {

self::$\_instance = new self();

}

return self::$\_instance;

}

private function \_\_construct() {

$request = $\_SERVER['REQUEST\_URI'];

$splits = explode('/', trim($request,'/'));

$this->\_controller = !empty($splits[0])?$splits[0]:'index';

$this->\_action = !empty($splits[1])?$splits[1]:'index';

if(!empty($splits[2])) {

$keys = $values = array();

for($idx=2, $cnt = count($splits); $idx<$cnt; $idx++) {

if($idx % 2 == 0) {

//Is even, is key

$keys[] = $splits[$idx];

} else {

//Is odd, is value;

$values[] = $splits[$idx];

}

}

$this->\_params = array\_combine($keys, $values);

}

}

public function route() {

if(class\_exists($this->getController())) {

$rc = new ReflectionClass($this->getController());

if($rc->implementsInterface('IController')) {

if($rc->hasMethod($this->getAction())) {

$controller = $rc->newInstance();

$method = $rc->getMethod($this->getAction());

$method->invoke($controller);

} else {

throw new Exception("Action");

}

} else {

throw new Exception("Interface");

}

} else {

throw new Exception("Controller");

}

}

public function getParams() {

return $this->\_params;

}

public function getController() {

return $this->\_controller;

}

public function getAction() {

return $this->\_action;

}

public function getBody() {

return $this->\_body;

}

public function setBody($body) {

$this->\_body = $body;

}

}

**The Controller**

The controller is where you create your application code. It is just a normal class, except that it implements the IController “marker” interface that does not contain any methods. Create the IController interface as shown in Listing 14-5.

**Listing 14-5.** *The IController Interface*

> pico –w application/models/icontroller.php

<?php

interface IController {}

Next, you will create a class to encapsulate your program logic. This class will be responsible for creating an instance of the view (which you’ll define shortly), binding a variable to it, and calling the render method, which will return the parsed result.

Once the view is rendered, the controller sets the body of the FrontController and returns. Create this class as shown in Listing 14-6.

**Listing 14-6.** *A Default Controller*

> pico –w application/controllers/index.php

<?php

class index implements IController {

public function index() {

$view = new View();

$view->name = "Kevin";

$result = $view->render('../views/index.php');

$fc = FrontController::getInstance();

$fc->setBody($result);

}

}

**The View**

The view model is one of the more complex classes in an MVC framework. It is designed to create a scope for variables (such as $view->name in Listing 14-6) and include the file returning the parsed output.

This is achieved by including from within a method, and thus limiting the scope to that method, and by using the output buffering feature of PHP. The actual properties support is provided by the ArrayObject SPL class, using a blank starting array and the special flag ARRAY\_AS\_PROPS, which allows for easy creation of a property overloaded object.

Create the view model as shown in Listing 14-7.

■**Note** This class is called View and is a model. It provides view rendering functionality. The “view” is actually the template defined later on, and will be found in the views folder.

**Listing 14-7.** *The View Model Class*

> pico –w application/models/view.php

<?php

class View extends ArrayObject {

public function \_\_construct() {

parent::\_\_construct(array(), ArrayObject::ARRAY\_AS\_PROPS);

}

public function render($file) {

ob\_start();

include(dirname(\_\_FILE\_\_) . '/' . $file);

return ob\_get\_clean();

}

}

Next, make a default view template, as shown in Listing 14-8.

**Listing 14-8.** *The Default View*

> pico application/views/index.php

Hello, <?php echo $this->name; ?>!

That’s it. Reload your browser, and you should see “Hello, Kevin!”

**URL Parameters**

You’re probably not named Kevin though, so let’s integrate your name with this script. Because of encapsulation, to do this, you need to modify only the controller.

Replace your application/controllers/index.php file with the file shown in Listing 14-9.

**Listing 14-9.** *Using URL Parameters*

> pico –w application/controllers/index.php

<?php

class index implements IController {

public function index() {

$fc = FrontController::getInstance();

$params = $fc->getParams();

$view = new View();

$view->name = $params['name'];

$result = $view->render('../views/index.php');

$fc->setBody($result);

}

}

Now change the URL to http://yourdomain.com/index/index/name/yourname =, and you should see “Hello, *yourname*!”

As you can see, creating an MVC framework is actually pretty basic. Well, this sample framework is so basic that it provides almost no functionality.

The frameworks you will find on the market have significant built-in features, such as database abstraction, data filtering and validation, and so on. In the next two chapters, you will learn about the built-in facilities a typical MVC framework provides.

**Just the Facts**

The MVC pattern is a design pattern that separates an application into three parts: the model, the view, and the controller. MVC is primarily designed to help with web application development workflow and allow teams to work together more efficiently by defining specific roles. These roles are developer, designer, and integrator. Developers are responsible for the model. Designers are responsible for the look and feel of the site. Integrators are in charge of putting it all together.

Most MVC frameworks follow similar concepts, providing classes for controllers and methods for actions. Views are templates, and can be created in plain PHP or a template language like Smarty.

There are a lot of considerations to bear in mind when choosing which framework is right for your project. The five key points are architecture, documentation, community, support, and flexibility.

The introduction to MVC in this chapter provides a basis for the next chapters, which discuss the specific implementation of the Zend Framework.

### 译文（节选自《PHP高级程序设计 模式、框架、测试》）

**MVC架构**

MVC（Model-View-Controller，模型-视图-控制器）是一种可以简化应用程序开发和维护的设计模式。为了实现这一目的，它将应用程序分隔到三个逻辑组件中。

模型（Model）。模型层负责应用程序的商业逻辑。它将封装访问数据库的方法并提供一个可重用的类库。通常，在模型中，你将发现用于数据库抽象、电子邮件发送、校验和验证等机制。

视图（View）。视图层通常包括Web设计或者模板设计等工作。它控制了数据的外观并且提供了从用户手机数据的机制。在视图中经常会找到视图特有的技术，包括HTML、CSS和JavaScript。

控制器（Controller）。控制器层将所有事项连接在一起，并且将视图设计和模型功能合并在一起。它负责从视图中收集输入数据，并决定程序的执行逻辑。控制器会调用模型的机制并解析返回的数据，以便使这些数据能够被视图所展示。它还负责所有应用程序的异常控制和流控制。

本文介绍了MVC模式及其特点。在最后一节，创建了一个小型的MVC框架，从而了解这样的一个框架是如何运作的。

1. **为什么使用MVC**

为什么MVC在Web应用程序开发领域中如此流行呢？虽然有很多理由值得你去使用MVC框架，其中包括代码的简介以及升级的便利等，但是业界公司选择MVC开发应用程序的首要原因是它提倡的开发流。

在开发团队中，MVC提供了有效的合作模式，它将各人的职责分成为三个主要的角色。

* 开发。开发人员是指处理模型的程序员。他们通常拥有与PHP、数据库管理、算法、架构和数据校验等方面相关的技能。这一角色通常会负责实现应用程序的编程细节（操作），它们会提供API，同时实施处理数据的策略。
* 设计。设计人员处理视图并负责实现应用程序的外观。他们具有如HTML、CSS、JavaScript和图形设计等方面的技能。通常，这一角色负责与内外部的通信源交互以决定要开发或者增强的应用程序的现实的业务规则。设计通常会导致原型的开发，也就是创建能够显示理想功能的模型设计。
* 集成。集成工作存在于控制器层中，它将设计师和开发人员的工作连接在一起。集成人员的经验通常比开发人员要少，他们负责切割静态模板并且制作应用程序所需的动态区域。他们还负责代理来自请求源的数据。他们将从表格中获取请求信息，将这些信息传递给模型，解释结果，并且将结果传递给视图。

通过职责分工，开发方法的细节与开发需求的细节分开了，这使得创意学科和技术学科能够更加容易地交互。

在一个典型的工作流中，首先设计师会根据现存的业务需求以及应用程序所要实现的功能，制作一个静态的、单独的原型设计。然后设计师将设计内容交给开发人员评审。

开发人员负责确保所有要求实现的功能是可行的，并且符合公司的安全和隐私策略。如果开发人员没有发现任何问题，他们就会创建一个开发计划，其中包括一个API模型，然后再交给集成人员。如果原型出现问题，项目会回到设计人员手中，这一过程就再次从头开始。

准备好设计原型和API模型之后，集成工作就会开始。集成人员将研究原型设计，并将它转换成模板语言，如普通PHP、Smarty等。这使得设计内容可以处理动态数据，并且将设计集成到完全有集成人员负责的站点URL结构中。然后编写一个控制器组件，这一组件会将请求（来自于表格、URL参数、cookie等）在Web服务器和他们需要操作的模型信息之间代理分发。结果产生之后，获取视图并输出，并且控制器会提供所有动态内容的值。

最后，当API模型完全实现之后，就可以将应用程序交给测试人员测试了。如果集成人员根据规范来编写控制器，并且模型开发人员遵循API编写代码，那么一切都会顺利进行。

MVC架构除了可以在团队成员之间分配职责之外，它还提供了几个非常关键的功能。其中一个是只要表现层的设计不破坏与控制器的约定，它就允许在任何时间修改和更新表现层的设计。这意味着可以很容易地编辑副本或者变化主题，而不用改变应用程序的操作方法或者整理大量的PHP代码。

MVC架构还提供了一个有用的文件结构的分类方法。每个MVC框架都有一个不同于其他框架的结构和布局，但是他们全部为模型、视图和控制器提供了单独的文件，并且很少例外的情况。这一特性会减少代码冲突，并且会节省原本用于合并版本控制系统中的修改的时间。

1. **MVC应用程序布局**

每个框架的MVC应用程序的布局各不相同。不过，一些概念对于所有的MVC框架来说是通用的。本节介绍的是PHP中MVC模式的典型实现。

1. **从Web服务器开始**

与其他PHP应用程序不同的是，MVC站点通常会实现一个用来处理Web站点上所有请求的核心PHP脚本。所以，浏览器浏览的不是/path/to/somefile.php路径，而是/controller/action路径。不过，你无法找到一个名为controller的目录或者一个名为action的文件。相反，像mod\_rewrite这样的URL重写程序可以用来将所有的HTTP请求重定向到一个核心的脚本文件上，这个文件被称为引导文件。

引导文件负责初始化框架，其中包括加载文件，读取配置数据，将URL解析成可以被动作读取的信息，以及填充封装了请求信息的对象等操作。最后，它还负责初始化控制器程序。

1. **动作和控制器**

引导完成之后，一个名为前端控制器（FC）的类将会被实例化，并且接管程序的运行。前端控制器通常是一个负责解释请求变量的内置类，它会把代码的执行路由到一个用户自定义的类上，这个类被称为动作控制器（AC）。通常，动作控制器将实现一个标准的接口或者从一个抽象类继承，所以前端控制器和动作控制器可以使用一个公共的API来交互。

然后，前端控制器将会调用由URL指定的动作控制器上的一个方法。这个方法被称为动作，它的名称和动作控制器的名称一样，都是由URL决定的。动作负责实现所有的功能。在这个方法中，你将实例化模型类，解析视图模板，并输出结果。

1. **模型**

模型通常是最容易的部分。它们一般不遵循任何特定的结构，而只是存在于一个公共的位置中，所以它们可能会被框架自动加载。模型是提供了所需的数据操作和解析功能的简单的工具类。

1. **视图**

视图是模板，可以用任何模板语言来编写。使用视图的主要目的是不要在视图中包含任何类型的处理逻辑或者数据操作，并且指数初有控制器特别提供的数据。通常，这意味着在视图中操作时，只能显示数组，而不是数据对象。

模板语言的选择在你自己，这取决于它是否能喝目标框架集成在一起。我个人认为，在MVC环境中编程时，普通的PHP是最佳的模板语言。FastTemplate和Smarty是值得考虑的第二选择。

1. **选择MVC框架的标准**

每种MVC框架都具有特定的优势和劣势。选择框架时，应该考虑五个关键因素，即架构、文档、社区、支持和灵活性。建议使用这些标准给考虑要用到的所有框架打分，然后选择一个对你来说最合适的框架。

1. **MVC框架的架构**

开发人员各不相同。一名开发人员认为合理的东西对另外一名开发人员来说可能完全不符合逻辑。应该选择一个自己可以理解的框架。有很多开发人员由于使用了复杂的MVC框架而遇到困难，这是由于他们曾经听说过它可能可以提供所有功能。有一点是肯定的，如果对框架的架构没有完整的了解，那么它将会妨碍程序的开发，而不是有助于开发。所以，选择自己最了解的框架即可。

还应该考虑是使用一个遵循线性函数和回调风格的框架，还是使用一个面向对象风格的框架。可能有一些开发人员比较喜欢后者，这是因为封装的概念对使用模型类来说是非常合适的。

你将需要理解框架的技术要求和文件结构，以便确保它有效地和部署流程一起运作。例如，某些框架具有很好的后向兼容的API，而另外一些则甚至不去尝试维持后向兼容性。这两种方法都有合理的一面。选择一个适合你的系统的即可。

1. **MVC框架文档**

所有的框架都带有文档，但相似点仅此而已。文档可以简单到只有一个安装文件，也可以复杂到有正式的培训课程。

当查看某个框架的文档时，我会问三个关键问题。

* 它是否是最新的？如果它的内容不是当前要使用的版本的内容，那么这个文档是没有用处的，确保下载的版本是文档中说明的版本。
* 它是否清晰？文档的编写是否清晰并且容易裂解？如果不能理解它试图说明的内容，就跳过它。
* 它是否完整？文档的内容是否全面？它是否覆盖了框架中的所有内容？还是只介绍了基本内容？你经常会获得组合文档，它是由很好的介绍文档和机器生成的高级文档组合而成的。

好的文档至少要符合这些条件，并且成本不高。

1. **MVC框架的社区**

支持MVC框架的社区的存在是非常关键的。你应该关注社区成员的参与度，以及社区成员对bug做出响应并修复的时间。

社区也是经理们需要考虑的首要因素，这是因为它会对雇用熟悉这个框架的新员工的能力产生很大影响。

你应该能够与使用框架的开发人员直接交流，以便报告问题、建议特性以及获取问题修复等。但是，不应该把框架开发人员当作技术支持，技术支持是另外一个需要单独讨论的问题，虽然它和社区也是相关联的。

发布版本的时间为你提供了项目的活跃程度的线索。这通常会很好地预示未来更新的频率。你将需要求出业务变化的速度，以便合理地适应变化中的框架。在维持API的稳定性和实现新特性之间，需要作出一些权衡，这也取决你自己的态度。

1. **MVC框架的支持**

在前一节提到过，框架的开发人员并非技术支持。如果针对一个可忽略的问题多次向他们发同一封邮件，就不太可能得到只有他们才能提供的进一步协助。在对待开发团队的问题上，应考虑到开发人员都是比较傲气的。

那么，在这种情况下，当有问题需要支持时，应该问谁呢？有些MVC框架的支持工作通过一种支持-收入模型来进行，也就是说，框架是免费的，但获得技术支持则需要付费。另外一些框架不提供任何支持。如果是这种情况，就需要寻找具有相关经验的第三方组织了。

邮件列表和用于技术支持的聊天频道将为特定框架提供免费支持。这些资源非常有用，但与付费的支持相比有很高的不确定性。为了从这些资源中获取最佳结果，必须采取共享的态度，也就是说付出最多的人也将得到最多的帮助。

当打算使用某种框架开始开发之前，确保了解可以获得支持的情况。当碰到问题时，如果不能迅速找打答案，付出的代价将是很高的。

1. **MVC框架的灵活性**

MVC框架的实现有多大的灵活性？你是否能够将现存的应用程序与这个框架集成，并确保它支持数据库？

某些框架的要求非常严格，会在编写和组织代码方面施加很多限制。这些框架可能会规定模板语言，也可能有非常独特的语法，从而导致很高的培训成本。

你应该寻找施加了尽可能少的限制的灵活框架。

1. **实现MVC框架**

一般来说，推荐你从可获得的公共框架中选择一种来使用，而不是自己去创建一个。再好的程序员也比不上社区的开发能力。不过，充分理解MVC的概念，有助于了解如何编写基本的框架，这正是我们目前要做的事情。

在这个例子中，你将遵循与Zend框架相似的布局，使用之前介绍的技术，并实现一个典型的前端控制器/操作路由系统。

在这个例子中，你将需要一个运行着Apache2、PHP5、mod\_rewrite的Web服务器，以及一个如PostgreSQL这样的数据库。这里是针对Debian Linux操作系统的，但是它们也可以在所有的支持PHP的平台上运行。

**说明** 如果没有具有必须配置的服务器，可以将所有必要的应用程序下载到桌面上，这些应用程序的安装并不复杂。稍微研究一下，便能够完成它们的安装了。

1. **设置一个虚拟主机**

首先，需要创建一个虚拟主机和一个新站点。步骤如下。

1. 执行一下命令，创建一个虚拟主机。

*> mkdir -p /usr/local/www/yourdomain.com/document\_root*

*> cd /etc/apache2/sites-available*

1. 编辑一个名为youdomain.com的文件。

*> pico –w yourdomain.com*

1. 将虚拟主机的定义放在yourdomain.com文件中。

*<VirtualHost \*>*

*DocumentRoot /usr/local/www/yourdomain.com/document\_root*

*ServerName yourdomain.com*

*</VirtualHost>*

1. 执行以下命令以激活站点。

*> a2ensite yourdomain.com*

*> /etc/init.d/apache2 reload*

1. 使浏览器可以访问你的站点。通过在服务器中指定这一域名系统（DNS）或者编辑hosts文件，便可以实现这一目的。要在Windows中编辑hosts文件，需要查找C:\WINDOWS\system32\drivers\etc\hosts文件。在文本编辑器打开它并添加一个新行。输入服务器的IP地址，按Tab键，然后在输入刚刚创建的文件名称（yourdomain.com）。最后，保存这个文件即可。
2. 打开新的浏览器实例（如果已经打开了一个浏览器，可能需要先关闭它，然后才能获得hosts文件的修改信息），然后浏览新站点。应该会看到一个空的目录索引。

现在，你已经准备开始开发你自己的MVC框架了。

1. **创建一个MVC框架**

现在，需要开始创建一些目录，推荐以下的布局：

*> cd /usr/local/www/yourdomain.com*

*> mkdir document\_root/images document\_root/styles*

*> mkdir –p application/models application/views \*

*> application/controllers*

*> find*

*.*

*./document\_root*

*./document\_root/images*

*./document\_root/styles*

*./application*

*./application/models*

*./application/views*

*./application/controllers*

下一步，为引导文件创建一个占位符，如下代码清单所示。

*Listing 14-1. Creating the Bootstrap index.php*

*> pico document\_root/index.php*

*Hello, World!*

然后，重新加载浏览器，你将会看到显示出的“Hello, World!”字样。浏览<http://yourdomain.com/nonexistent>这个地址。应该会碰到一个“404文件未找到”的错误信息。

为了让MVC框架起作用，你需要一个请求来触发这个index.php文件。要做到这一点，需要通过.htaccess文件获得mod\_rewrite的帮助，如下代码清单所示。

*Listing 14-2. Using mod\_rewrite in an .htaccess File*

*> pico document\_root/.htaccess*

*RewriteEngine on*

*RewriteRule !\.(js|gif|jpg|png|css)$ index.php*

重新加载浏览器，你将会看到“Hello, World!”字样。如果没有看到，需要确保服务器上安装了mod\_rewrite，并且没有使用非标准的allowOverrides指令来限制.htaccess文件的重写。

1. 引导

这个过程的下一步是在index.php文件中创建一个引导脚本。这个文件将负责包含文件并初始化前端控制器。大多数MVC控制器将使用SPL自动加载机制来加载文件，但是为简单起见，现在我们只使用require\_once。创建的代码清单如下所示。

*Listing 14-3. The Bootstrap Loader*

*> pico –w document\_root/index.php*

*<?php*

*//Require Components*

*require\_once('../application/models/front.php');*

*require\_once('../application/models/icontroller.php');*

*require\_once('../application/models/view.php');*

*//Require Controllers*

*require\_once('../application/controllers/index.php');*

*//Initialize the FrontController*

*$front = FrontController::getInstance();*

*$front->route();*

*echo $front->getBody();*

说明 上面代码清单不需要使用结束标签?>，这是因为文件的结束字符会自动关闭标签。由于文件末尾的新行会导致过早的输出，并且可能会给发送HTTP头带来问题，所以不推荐大家使用结束标签。

1. 前端控制器

前端控制器的任务是解析URL，实例化控制器，并调用实现操作的方法。这是一个单例类，它负责控制应用程序的程序流。

实现操作的原理是URL遵循/controller/action/key1/value1/…格式。解析了URL之后，使用反射API调用IController实现类的实现操作的方法。

首先，需要创建如下代码清单所示的FrontController类。

*Listing 14-4. The FrontController Class*

*> pico –w application/models/front.php*

*<?php*

*class FrontController {*

*protected $\_controller, $\_action, $\_params, $\_body;*

*static $\_instance;*

*public static function getInstance() {*

*if( ! (self::$\_instance instanceof self) ) {*

*self::$\_instance = new self();*

*}*

*return self::$\_instance;*

*}*

*private function \_\_construct() {*

*$request = $\_SERVER['REQUEST\_URI'];*

*$splits = explode('/', trim($request,'/'));*

*$this->\_controller = !empty($splits[0])?$splits[0]:'index';*

*$this->\_action = !empty($splits[1])?$splits[1]:'index';*

*if(!empty($splits[2])) {*

*$keys = $values = array();*

*for($idx=2, $cnt = count($splits); $idx<$cnt; $idx++) {*

*if($idx % 2 == 0) {*

*//Is even, is key*

*$keys[] = $splits[$idx];*

*} else {*

*//Is odd, is value;*

*$values[] = $splits[$idx];*

*}*

*}*

*$this->\_params = array\_combine($keys, $values);*

*}*

*}*

*public function route() {*

*if(class\_exists($this->getController())) {*

*$rc = new ReflectionClass($this->getController());*

*if($rc->implementsInterface('IController')) {*

*if($rc->hasMethod($this->getAction())) {*

*$controller = $rc->newInstance();*

*$method = $rc->getMethod($this->getAction());*

*$method->invoke($controller);*

*} else {*

*throw new Exception("Action");*

*}*

*} else {*

*throw new Exception("Interface");*

*}*

*} else {*

*throw new Exception("Controller");*

*}*

*}*

*public function getParams() {*

*return $this->\_params;*

*}*

*public function getController() {*

*return $this->\_controller;*

*}*

*public function getAction() {*

*return $this->\_action;*

*}*

*public function getBody() {*

*return $this->\_body;*

*}*

*public function setBody($body) {*

*$this->\_body = $body;*

*}*

*}*

1. 控制器

控制器是创建应用程序代码的地方。除了需要实现不包含任何方法的IController“标记”接口之外，它只是一个普通的类。创建如下代码清单所示的IController接口。

*Listing 14-5. The IController Interface*

*> pico –w application/models/icontroller.php*

*<?php*

*interface IController {}*

下一步，需要创建用于封装程序逻辑的类。这个类将负责创建视图的实例，将变量绑定到视图上，然后调用提交方法，最后，这个方法将返回解析后的结果。

完成视图输出之后，控制器将设置FrontController的内容，并返回。创建如下代码清单所示的控制器类。

*Listing 14-6. A Default Controller*

*> pico –w application/controllers/index.php*

*<?php*

*class index implements IController {*

*public function index() {*

*$view = new View();*

*$view->name = "Kevin";*

*$result = $view->render('../views/index.php');*

*$fc = FrontController::getInstance();*

*$fc->setBody($result);*

*}*

*}*

1. 视图

视图模型是MVC框架中比较复杂的类之一。它的作用是为变量创建作用域，并且包含了返回解析结果的文件。

通过在方法内部包含输出结果，从而限制那个方法的作用域，并使用PHP的输出缓存特性，就可以实现这个功能。实际的属性支持是有SPL类ArrayObject提供的，它使用的是起始为空的数组和ARRAY\_AS\_PROPS特殊标记，这样便允许轻松地创建属性重载对象了。

创建如下代码清单所示的视图模型。

说明 这个类被称为View，它是一个模型。它提供了视图输出的功能。“视图”实际上是之后将要定义的模板，模板文件将会放在views目录中。

*Listing 14-7. The View Model Class*

*> pico –w application/models/view.php*

*<?php*

*class View extends ArrayObject {*

*public function \_\_construct() {*

*parent::\_\_construct(array(), ArrayObject::ARRAY\_AS\_PROPS);*

*}*

*public function render($file) {*

*ob\_start();*

*include(dirname(\_\_FILE\_\_) . '/' . $file);*

*return ob\_get\_clean();*

*}*

*}*

下一步，需要创建一个默认的视图模板，如下代码清单所示。

*Listing 14-8. The Default View*

*> pico application/views/index.php*

*Hello, <?php echo $this->name; ?>!*

重新加载浏览器，你将看到“Hello, Kevin!”字样。

1. URL参数

你可能不叫Kevin，所以让我们将你的名字与这个脚本集成在一起。由于类封装的关系，要实现这一点，需要修改控制器类。

将application/controller/index.php文件的内容替换为如下代码清单内容。

*Listing 14-9. Using URL Parameters*

*> pico –w application/controllers/index.php*

*<?php*

*class index implements IController {*

*public function index() {*

*$fc = FrontController::getInstance();*

*$params = $fc->getParams();*

*$view = new View();*

*$view->name = $params['name'];*

*$result = $view->render('../views/index.php');*

*$fc->setBody($result);*

*}*

*}*

现在，将URL修改为*http://youdomain.com/index/index/name/youname=*，将会看到“Hello, yourname!”字样。

你可以看到，创建一个MVC框架实际上非常简单。不过，这个示例的框架太简单，以至于几乎没有提供任何功能。

市场上的框架具有大量的内置特性，例如数据库抽象、数据过滤和验证等。

1. **小结**

MVC模式是一种设计模式，它将应用程序分隔为三个部分：模型、视图和控制器。MVC的作用主要在于为Web应用程序开发工作流提供帮助，通过定义特定的角色，使得团队成员更加高效地一起工作。这些角色是开发人员、设计师和集成人员。开发人员负责实现模型，设计师负责设计站点的外观，集成人员负责将它们集成在一起。

大多数MVC框架遵循相似的概念，它们都提供了控制器类和用于实现操作的方法。视图是模板，可以使用普通PHP代码来创建，也可以使用如Smarty这样的模板语言来创建。

选择适合项目的框架时，需要考虑许多因素，其中五个关键的因素是：架构、文档、社区、支持和灵活性。