



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

教师职称 教 授

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
前 言.....	5
1. 绪论.....	6
1.1 研究背景.....	6
1.2 设计目标.....	8
1.3 本文结构.....	9
2. 系统开发环境与技术.....	9
2.1 系统开发环境.....	9
2.1.1 Apache 服务器介绍.....	10
2.1.2 Linux 服务器介绍.....	11
2.1.3 MySQL 服务器介绍.....	12
2.1.4 Git 软件版本控制管理.....	13
2.2 系统开发技术.....	14
2.2.1 PHP 脚本语言.....	14
2.2.2 ThinkPHP 开发框架.....	15
2.2.3 MVC 模式.....	15
2.2.4 JavaScript 与 ExtJS 库.....	16
2.2.5 新浪微博应用授权与开发流程.....	18
3. 系统需求分析与前台设计.....	26
3.1 需求分析.....	26
3.1.1 系统前端操作平台设计概述.....	27
3.1.2 系统用例图.....	28
3.2 系统设计.....	30
3.2.1 系统层次划分.....	30
3.2.2 数据库设计.....	31
3.2.3 系统时序图.....	38
4. 系统详细设计与实现.....	39
4.1 系统项目的文件结构.....	39
4.2 设计思路.....	40
4.3 核心功能实现代码及运行效果.....	40
5. 总结与展望.....	58
5.1 课题总结.....	58
5.2 进一步开发的展望.....	58
6. 参考文献.....	59
7. 致谢.....	59

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。

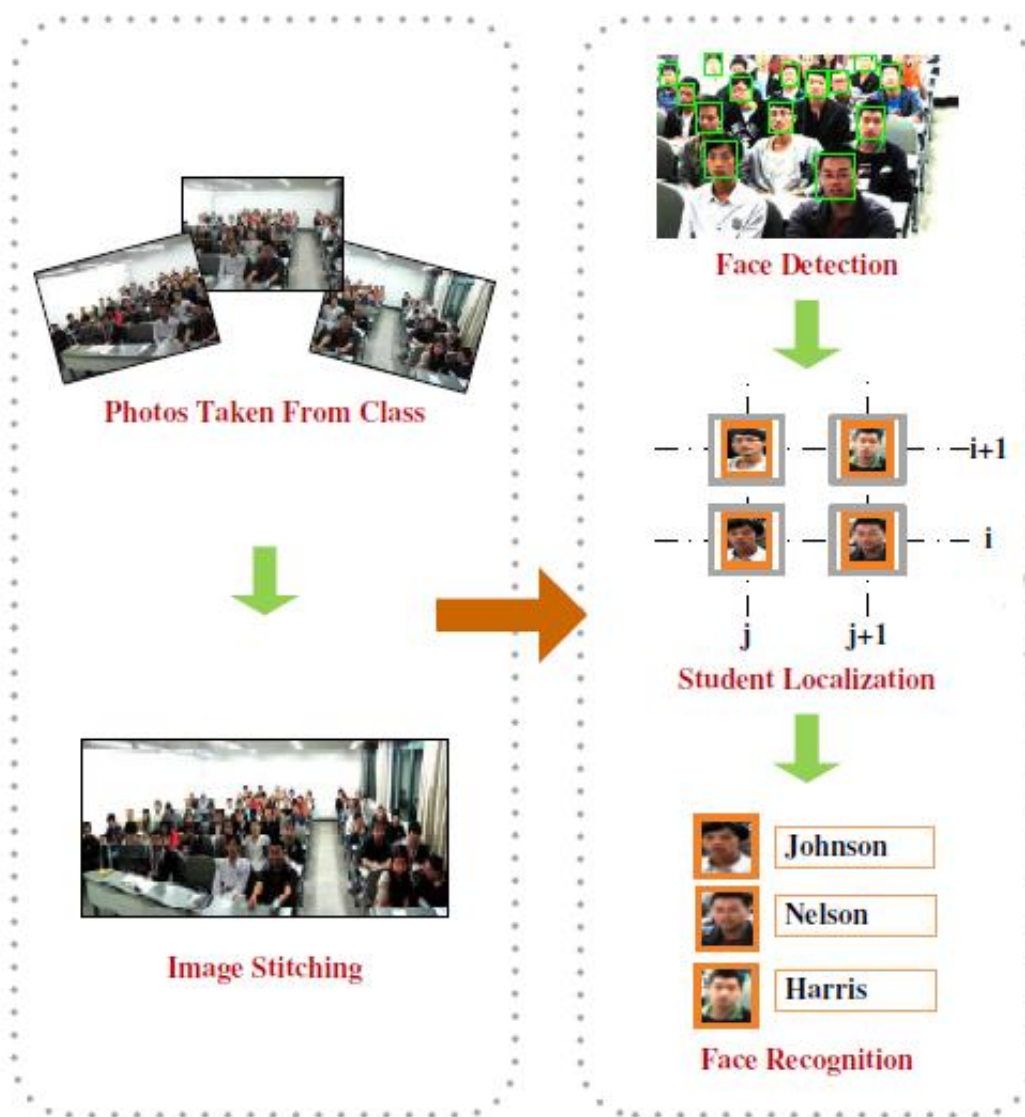
第四章是系统的详细设计与功能实现。

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。

1. 绪论

1.1 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。本系统通过对课堂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。



(图 1.1)

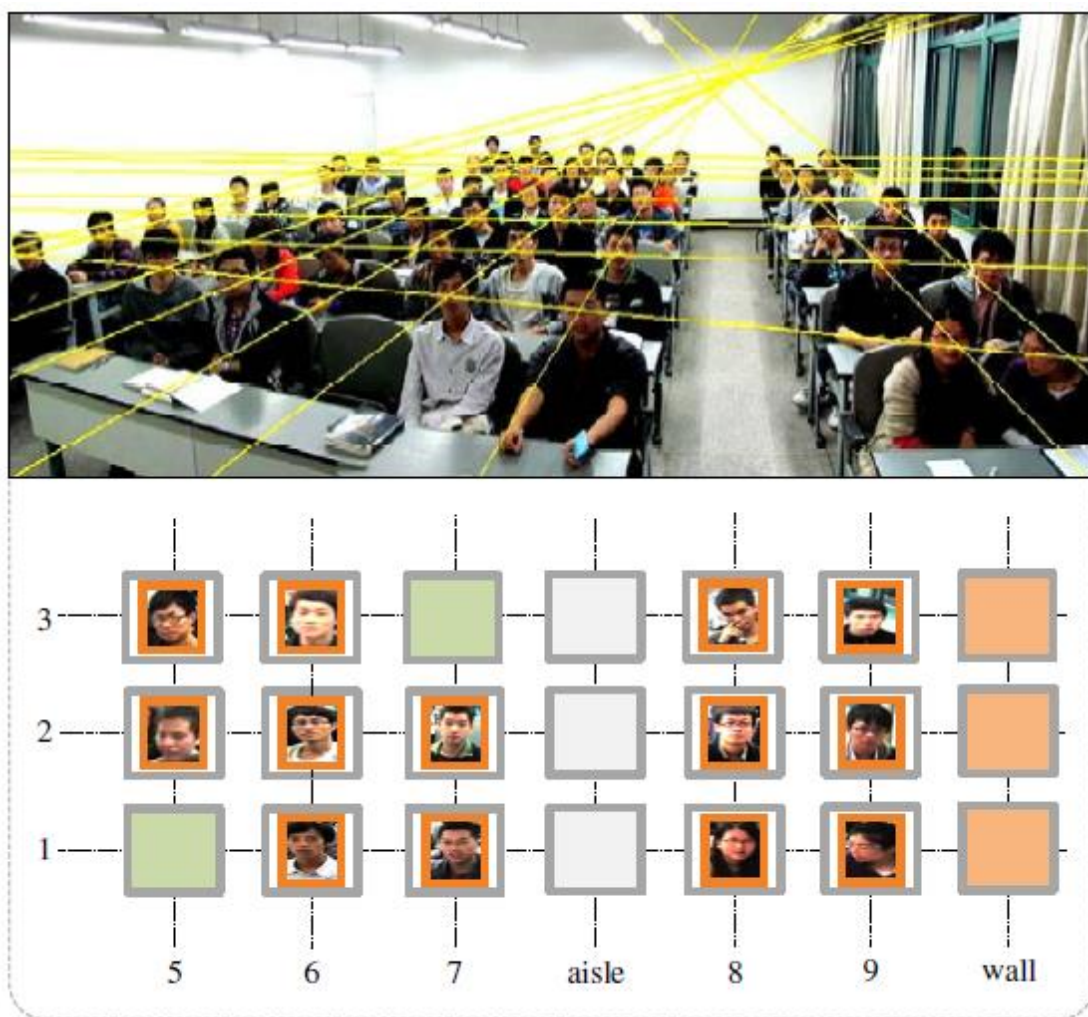
对于摄像机拍好的照片,根据实际情况进行人脸识别处理。首先固定好各自的座位,取出照片上各人的脸部,放入二维表中,与正常情况下进行对比。图片进行拆分和处理主要如下面两图所示。



(图 1.2)



(图 1.3)



(图 1.4)

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。

1.2 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。

1.3 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。

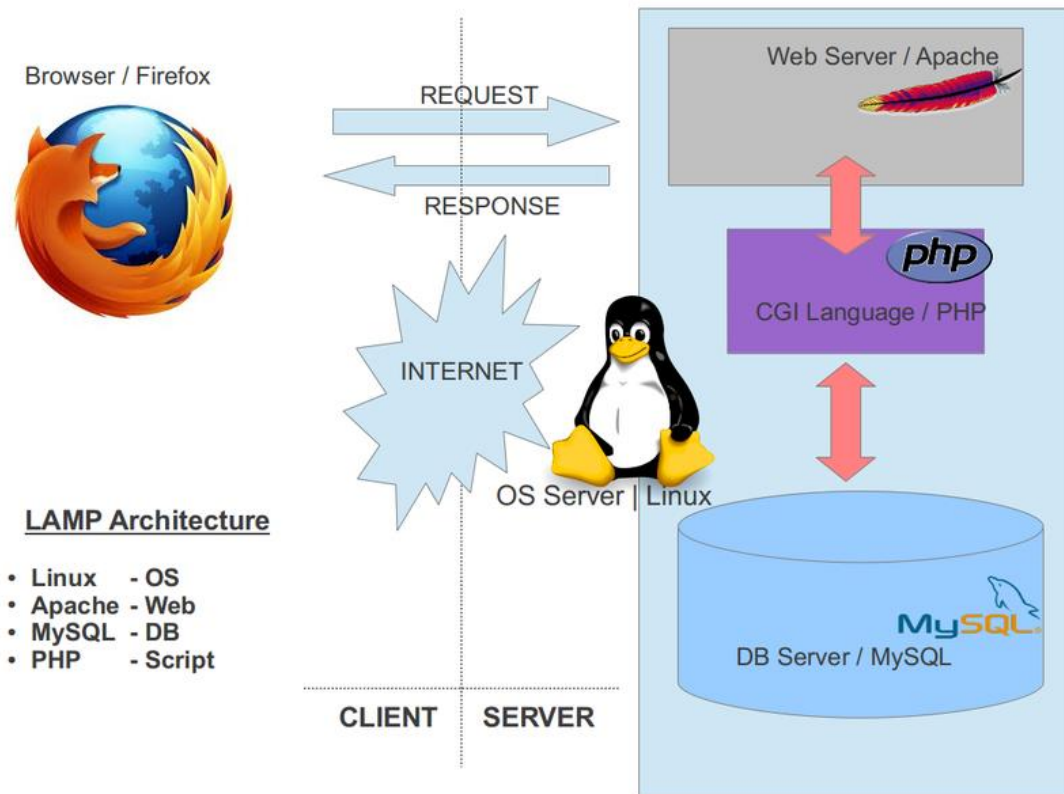
第四章是系统实现，给出系统后台的具体实现，包括实现效果和核心代码说明。

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。

2. 系统开发环境与技术

2.1 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。



(图 2.1)

2.1.1 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 and 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。

2.1.2 Linux 服务器介绍

Linux(i/'lɪnəks/ 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端使用的 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。

2.1.3 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。

2.1.4 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。

2.2 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。

2.2.1 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。

2.2.2 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。

2.2.3 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。

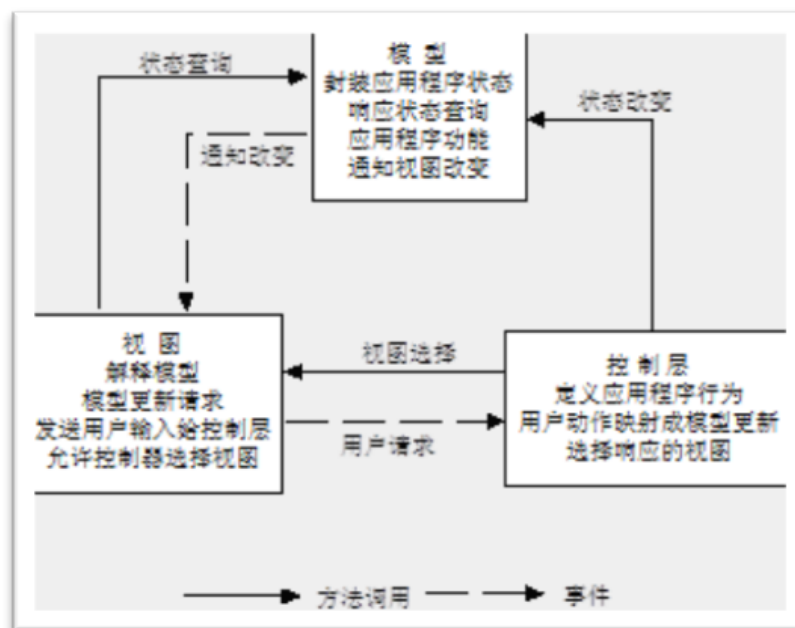
模型（Model）层：也就是软件业务的实现部分。一个软件有什么功能，它如何实现功能，这些功能之间的联系都是在模型层进行处理的。可以这样说，模型层是一个软

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所示。



(图 2.2)

2.2.4 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：

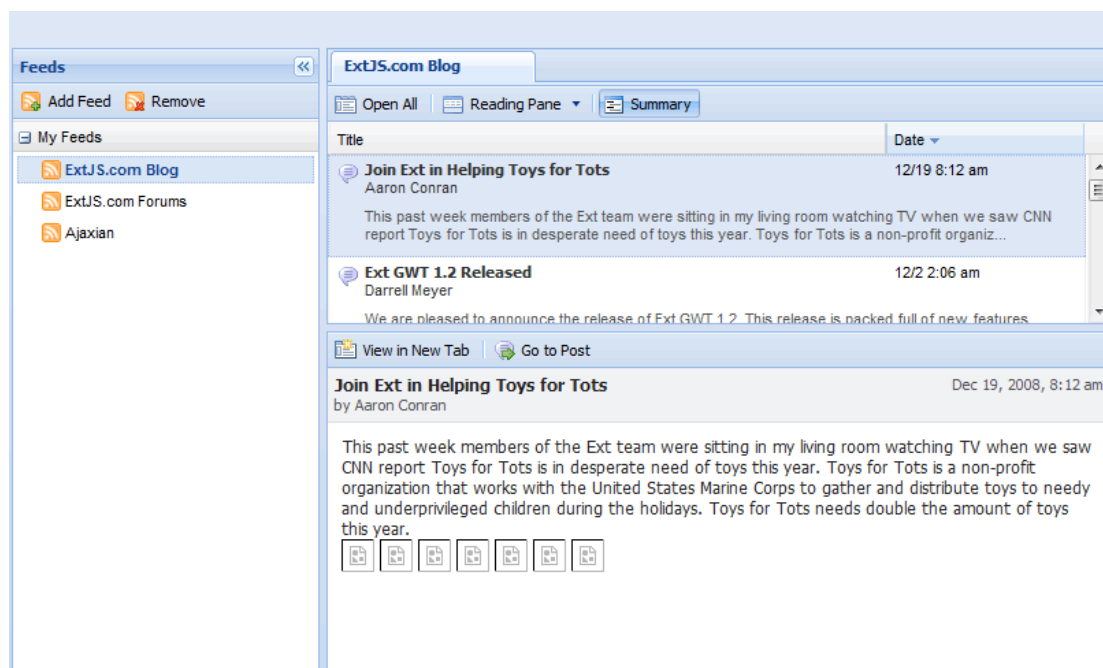
- ECMAScript，描述了该语言的语法和基本对象；
- 文档对象模型（DOM），描述处理网页内容的方法和接口；
- 浏览器对象模型（BOM），描述与浏览器进行交互的方法和接口。

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



（图 2.3）

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。

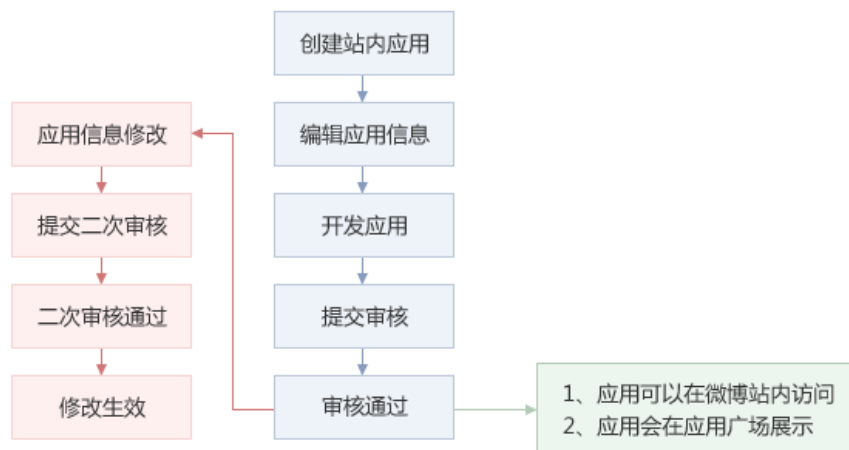
Ext 的 UI 组件模型和开发理念脱胎、成型于 Yahoo 组件库 YUI 和 Java 平台上 Swing 两者，并为开发者屏蔽了大量跨浏览器方面的处理。相对来说，EXT 要比开发者直接针对 DOM、W3C 对象模型开发 UI 组件轻松。

特点如下：

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

2.2.5 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



(图 2.4)

2) 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。



(图 2.5)

3) 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：



(图 2.6)

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。

编辑开发者信息

成为一名开发者需要完善您的信息，以便我们及时与您取得联系

开发者类型: ☒ 个人 ☐ 公司

开发者名称:

所在地区:

邮箱:

联系电话:

聊天工具:

网站:

开发者类型一经填写，不可修改

请填写您的个人姓名或公司名称

请填写您所在的城市

请务必填写常用邮箱，最主要联系方式

请填写您的联系电话，如13123456789或010-42676666

MSN/QQ/Gtalk至少填写一项

请填写您个人或公司网站网址

保存设置

放弃填写

(图 2.7)

填写完毕，请点击保存按钮进行保存。

4) 创建站内应用

在应用开发页面点击“创建应用”，会弹出浮层：



(图 2.8)

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：

(图 2.9)

- 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。

5) 编辑应用基本信息

1、 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。

(图 2.10)

2、 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。



(图 2.11)

3、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。



(图 2.12)

4、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。



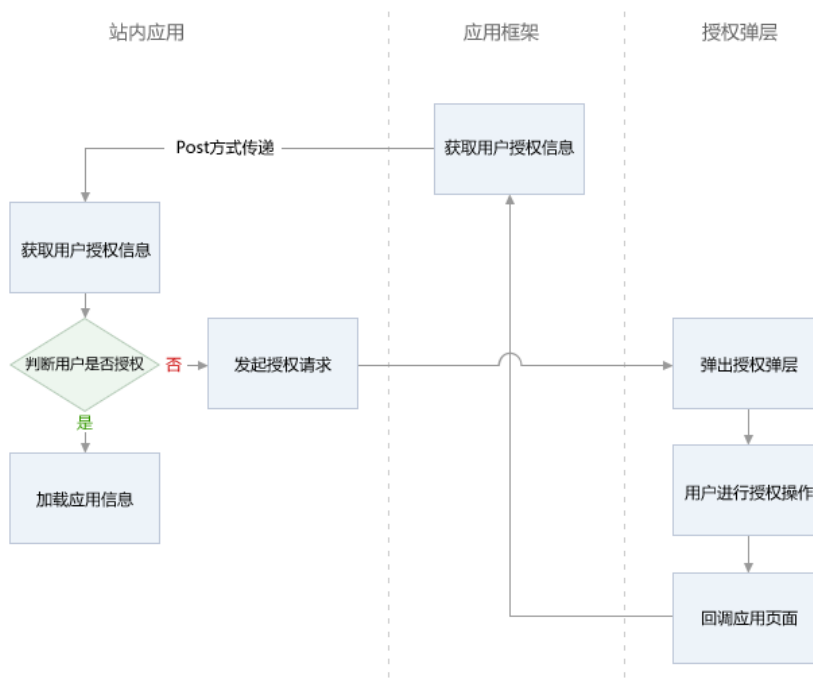
(图 2.13)

相关内容填写完毕后，请点击保存按钮进行保存。

6) 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，开发者需要妥善保管这个密钥，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：



(图 2.14)

站内应用框架会 POST 给您的应用以下信息：

A. 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

(表 2.1)

用户未授权时应用需要自动判断，并调用 OAuth 2.0 授权弹层。

B. 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

(表 2.1)

```
// 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:或

② 在当前窗口打开，刷新整个框架，地址栏改变

调用方法:

7) 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，会与您进行联络，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。



(图 2.15)

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使用，并且自动被推荐到应用广场。

3. 系统需求分析与前台设计

3.1 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。

3.1.1 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。

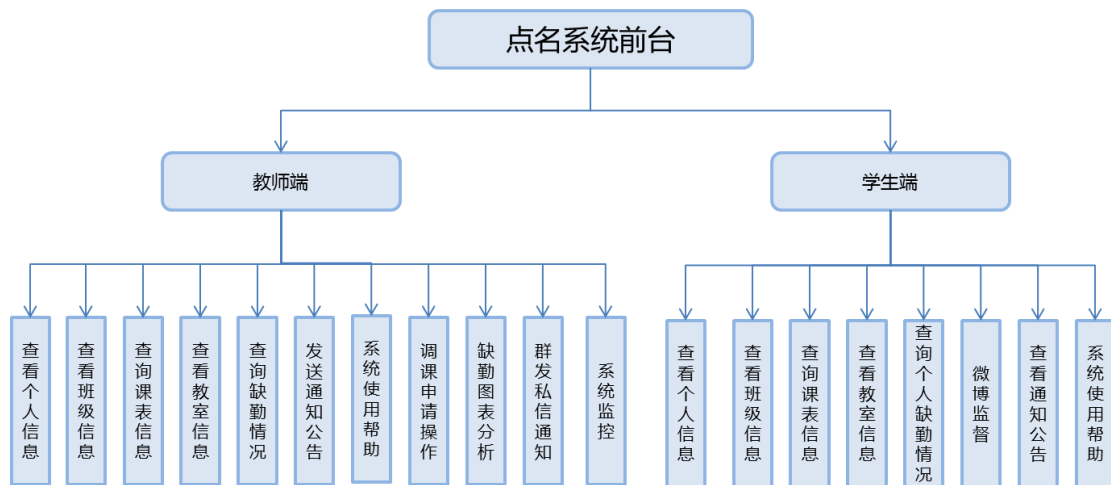
下面为“教师端”的功能说明：

- 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

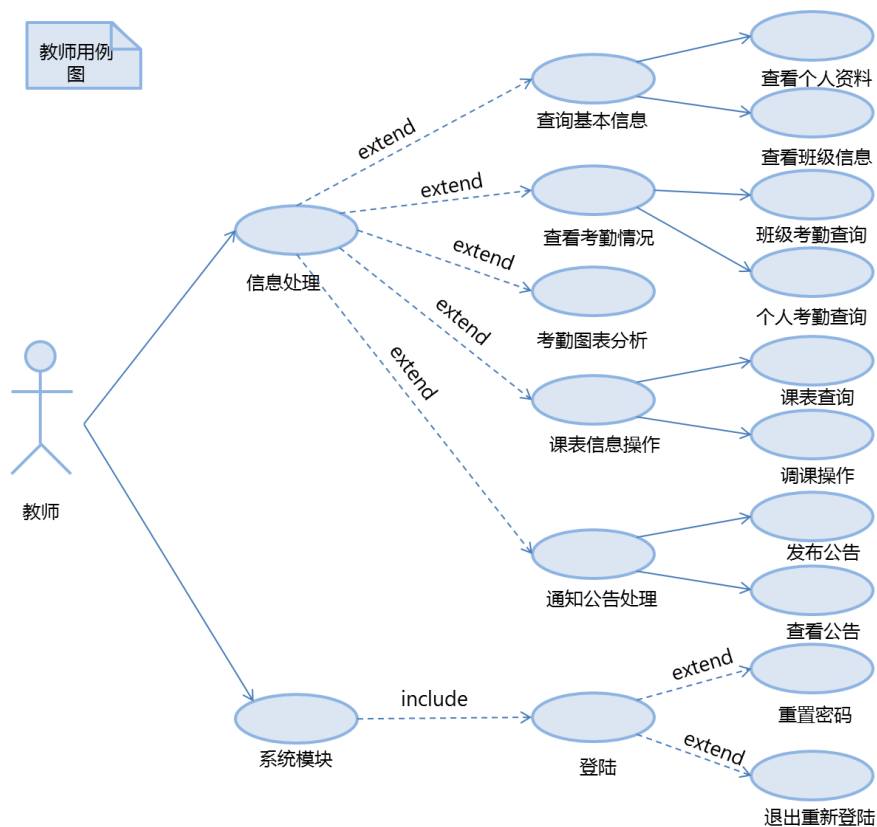
系统结构总图如下图所示：



(图 3.1)

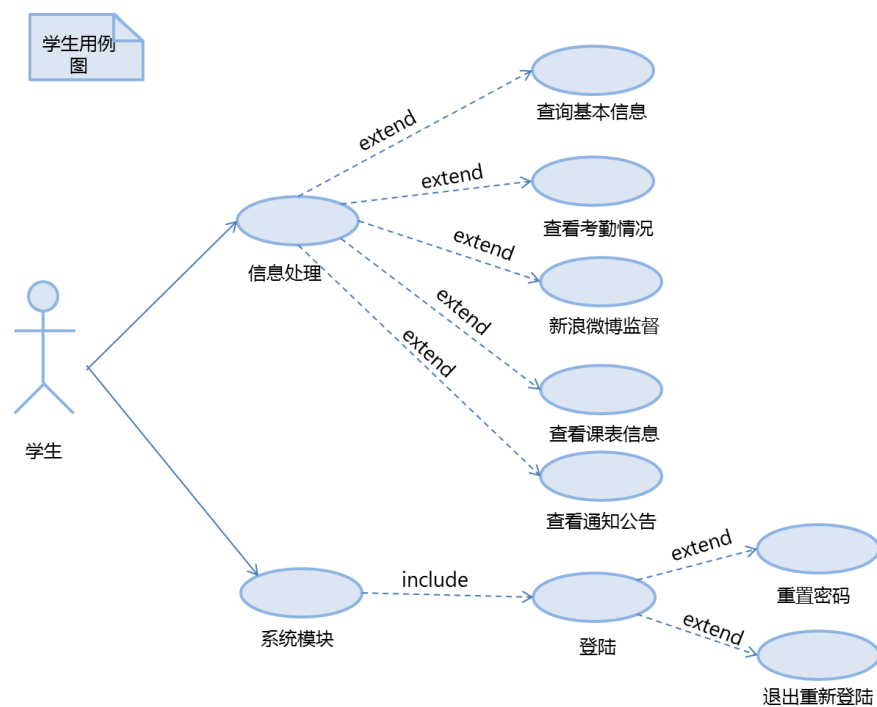
3.1.2 系统用例图

教师操作用例图如下图所示：



(图 3.2)

学生操作用例图如下图所示：

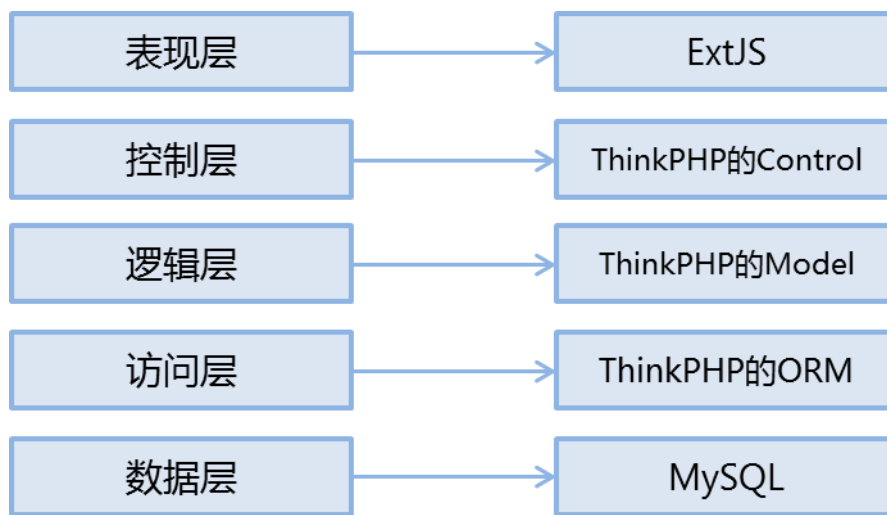


(图 3.3)

3.2 系统设计

3.2.1 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：



(图 3.4)

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。

3.2.2 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。如果设计不当，查询起来就非常吃力，程序的性能也会受到影响。

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

2) 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

3) 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

4) 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

5) 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

6) 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

7) 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

8) 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

9) 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

10) 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

11) 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

12) 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

13) 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

14) 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

15) 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

16) 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

17) 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	

(表 3.1)

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	

(表 3.2)

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	

(表 3.3)

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	

(表 3.4)

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	

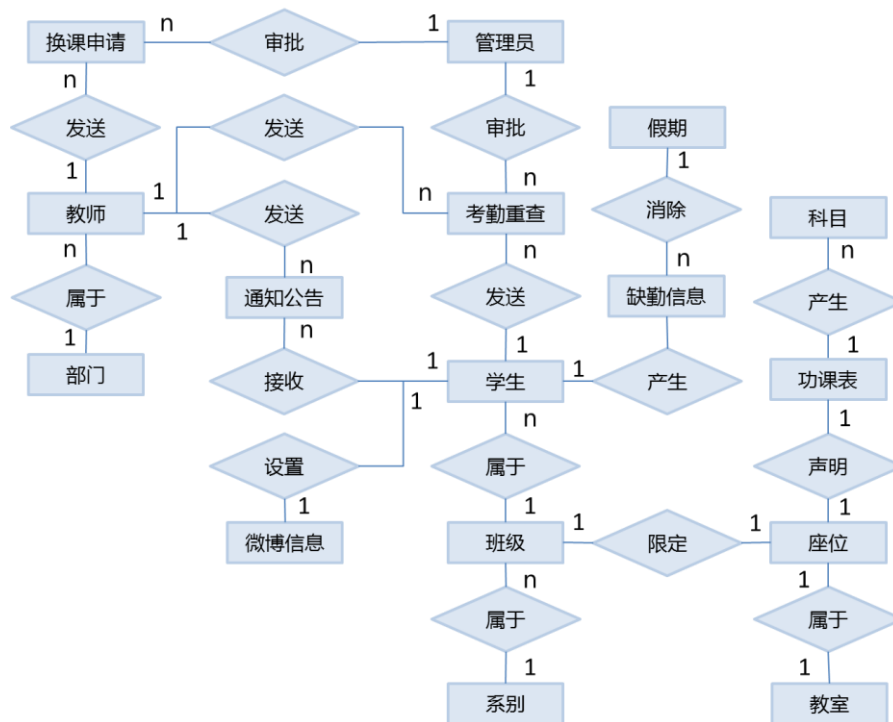
(表 3.5)

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	

(表 3.6)

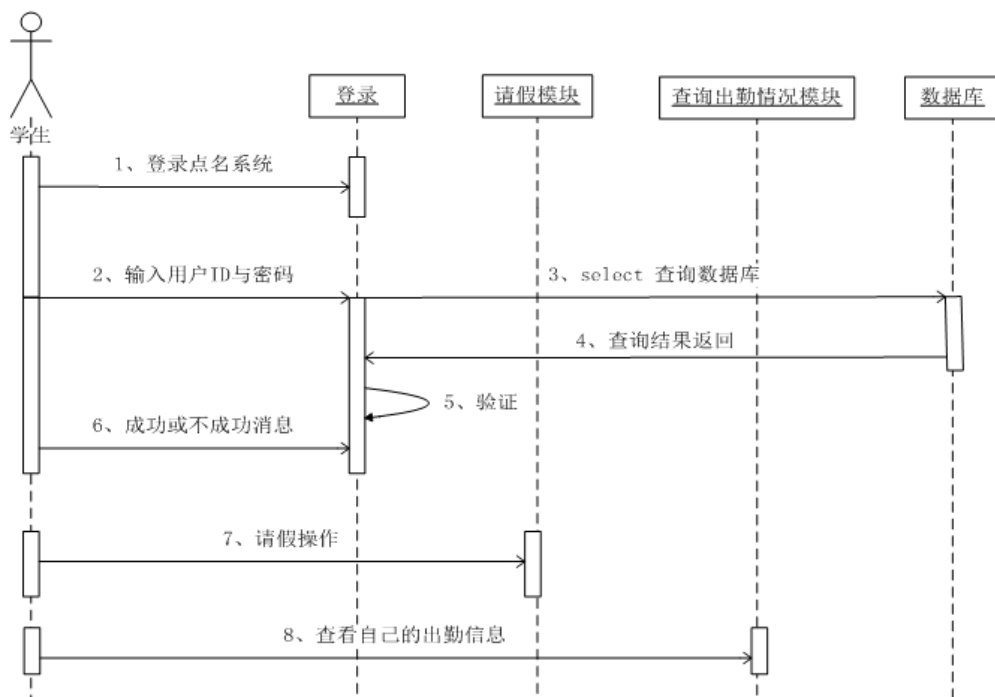
综合以上分析和数据结构情况，构建数据库 E-R 图，如下图所示：



(图 3.5)

3.2.3 系统时序图

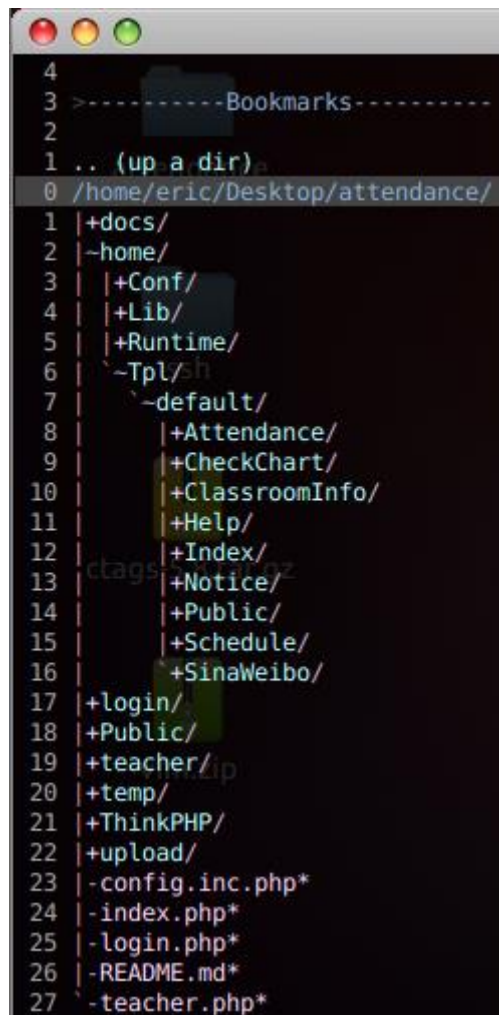
对于学生，在登录以后，点击查询出勤情况，将向数据库发出一个请求，连接成功以后，返回数据请求的结果。学生操作时时序图如下图所示：



(图 3.6)

4. 系统详细设计与实现

4.1 系统项目的文件结构



(图 4.1)

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

4.2 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。

4.3 核心功能实现代码及运行效果

1. 系统登陆界面:

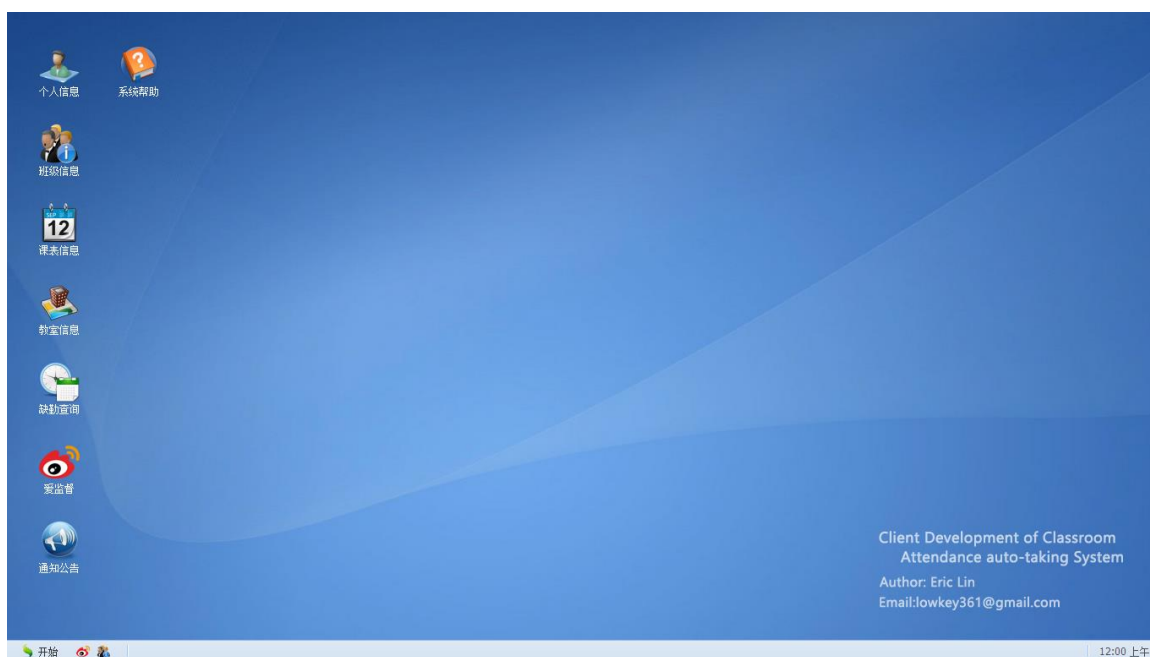
- 1) 输入学生学号或者教师教工号, 选择对应的身份角色, 敲击“登陆”可以登陆系统。



(图 4.2)

2. 系统主界面:

- 1) 点击桌面对应图标, 可以选择相应的功能;
- 2) 可以通过鼠标右键、菜单栏等方式, 自定义桌面背景及其他设置。



(图 4.3)

3. 班级人员查询界面:

- 1) 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;

- 2) 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;
- 3) 如果不输入搜索条件, 则返回用户权限范围内的全部记录。

学号	姓名	性别	班级	系别	邮箱	电话	监护人
091402116	eric	女	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402117	55eric	男	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402118	eric	男	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402120	eric	女	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402122	eric	男	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402123	eric	男	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402124	eric	女	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402129	eric	男	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402131	eric	女	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402132	eric	男	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402133	eric	男	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402135	eric	男	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402136	eric	女	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402137	eric	男	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402140	eric	男	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402141	eric	男	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402144	eric	男	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402145	eric	男	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402146	eric	男	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father
091402147	eric	男	网络工程1班	计算机科学系	lowkey361@gmail.com	13790766810	erics father

(图 4.4)

部分代码:

```

public function getClassmate() {
    $start = isset($_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
    $studentModel = M('Student');
    if(isset($_GET['filterId'])) {
        $filterIdArr = split(',', $_GET['filterId']);
        $classModel = M('Class');
        $classinfo = $classModel->field("`id`,`name`")->where("`name` = '". $filterIdArr[0] .
        "").->select();
        if(empty($query)) {
            $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 {
        $classinfo = $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`=$classinfo 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($total) . '}';
    }
} else {
    if(empty($query)) {
        $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 {
        $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) . '}';
    }
}
}
}

```

4. 班级座位表查询

- 1) 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；
- 2) 只能查询某个班级在某节课的座位。



(图 4.5)

5. 课表查询

- 1) 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；
- 2) 可以查询某个班级一段时间内的课程安排。



(图 4.6)

部分代码:

```
public function getSchedule() {
    $query = $_GET['scheduleQueryObj'];
    $queryArr = split(',', $query);

    $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();
        $count = 0;
        for($i=0; $i<11; $i++) {
            for($j=0; $j<7; $j++) {
                $has = 'no';
                $k = 0;
                for($k; $k<count($info); $k++) {
                    if($info[$k][day]==$j && $info[$k][time]==$i) {
                        $has = 'yes';
                        break 1;
                    } else {
                        continue;
                    }
                }
                if($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';
                }
                if(fmod($count, 7) == 0) {
                    $scheduleArray[$count][turn] = 'yes';
                } else {
                    $scheduleArray[$count][turn] = 'no';
                }
            }
        }
    }
}
```

```

        $count++;
    }
}
$this->assign('info', $scheduleArray);
$this->display();
} else {
    $this->display('Schedule:noSchedule');
}
}
}

```

6. 调课申请

- 1) 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；
- 2) 调课申请需后台管理员批准后生效；
- 3) 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。
- 4) 该功能只供教师申请，学生端没有权限。

(图 4.7)

部分代码：

```

public function applySchedule() {
    $applyScheduleOriginClass = isset($_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'] : "";
    $applyScheduleContent = isset($_POST['applyScheduleContent']) ?
$_POST['applyScheduleContent'] : "";              // 申请说明内容
    $teacherinfo = $modifyclassModel->query("SELECT `teacher`.`id` FROM `teacher` WHERE
`teacher`.`number`=$id");
    $info = $modifyclassModel->query("INSERT INTO `modifyclass` (`scheduleid`, `classroomid`,
`day`, `time`, `status`, `teacherid`, `title`, `content`) VALUES ($applyScheduleOriginSubject,
$applyScheduleAfterClassroom, $day, $time, 0, $teacherid, '$applyScheduleTitle',
'$applyScheduleContent')");
}

```

7. 教室信息

1) 可以查看全校所有教室的基本情况，供调课等情况使用。

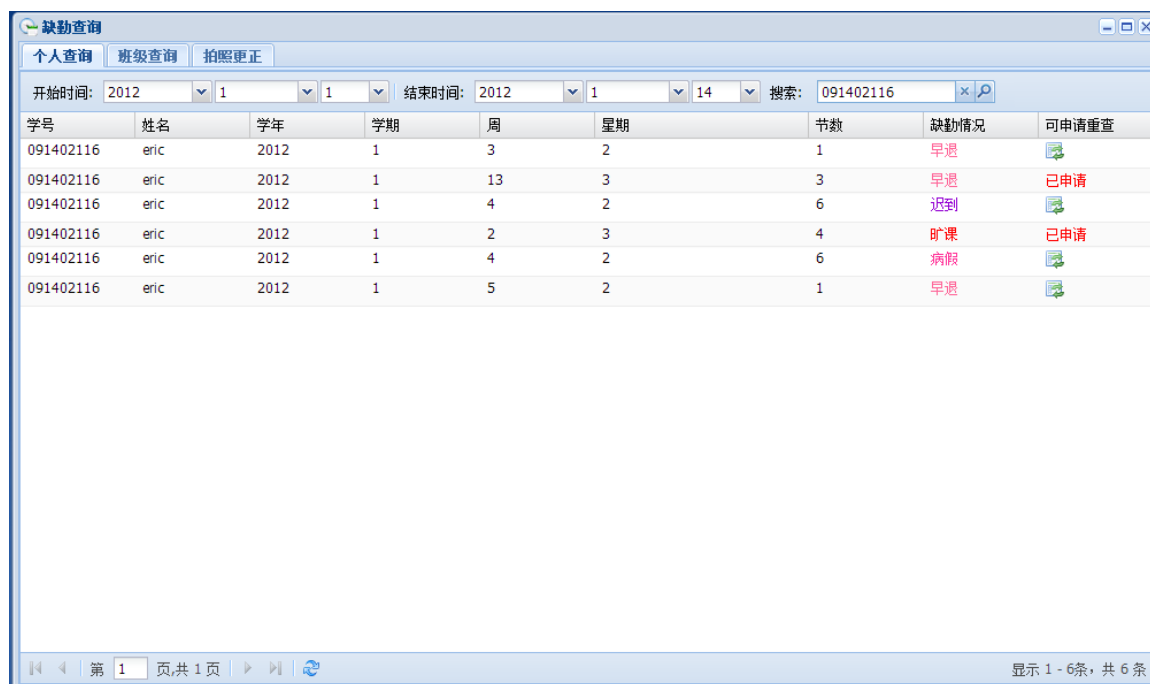
教室名称	是否多媒体	容纳学生数	一排座位数	一列座位数
1-101	是	150	10	15
1-102	是	72	6	12
1-103	是	120	10	12
1-104	是	120	10	12
1-105	是	80	8	10
1-106	否	80	8	10
1-107	否	80	8	10
1-108	是	80	10	8
1-109	是	60	10	6
1-110	是	160	10	16
1-111	是	160	10	16
1-201	是	70	7	10
1-205	是	70	7	10
1-206	是	70	7	10
1-207	是	70	7	10
1-310	是	100	10	10
1-311	是	30	6	5
1-312	否	30	6	5
5-501	是	80	10	8
5-503	是	80	10	8

第 1 页, 共 2 页 显示 1 - 20条, 共 24 条

(图 4.8)

8. 缺勤个人查询

- 1) 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，点击“查询”即可返回相应的结果。



(图 4.9)

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 {
        $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) . '>';
        }
    }
}
}
}

```

9. 缺勤班级查询

- 1) 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；
- 2) 该功能只限教师使用，学生端没有权限。

学号	姓名	性别	班级id	系别id	邮箱	电话	监护人	考勤情况	可申请重查
91402116	eric	女	网络工程1班	计算机科学系	lowkey361...	13790766810	erics father	早退	
91402117	55eric	男	网络工程2班	计算机科学系	lowkey361...	13790766810	erics father	早退	
91402116	eric	女	网络工程1班	计算机科学系	lowkey361...	13790766810	erics father	早退	已申请
91402116	eric	女	网络工程1班	计算机科学系	lowkey361...	13790766810	erics father	迟到	
91402116	eric	女	网络工程1班	计算机科学系	lowkey361...	13790766810	erics father	旷课	已申请
91402118	eric	男	网络工程1班	计算机科学系	lowkey361...	13790766810	erics father	旷课	
91402119	eric	男	中文2班	中文系	lowkey361...	13790766810	erics father	旷课	
91402117	55eric	男	网络工程1班	计算机科学系	lowkey361...	13790766810	erics father	迟到	
91402116	eric	女	网络工程1班	计算机科学系	lowkey361...	13790766810	erics father	病假	
91402119	eric	男	中文2班	中文系	lowkey361...	13790766810	erics father	早退	
91402117	55eric	男	网络工程1班	计算机科学系	lowkey361...	13790766810	erics father	早退	
91402116	eric	女	网络工程1班	计算机科学系	lowkey361...	13790766810	erics father	早退	
91402117	55eric	男	网络工程1班	计算机科学系	lowkey361...	13790766810	erics father	迟到	
91402118	eric	男	网络工程1班	计算机科学系	lowkey361...	13790766810	erics father	迟到	

(图 4.10)

部分代码:

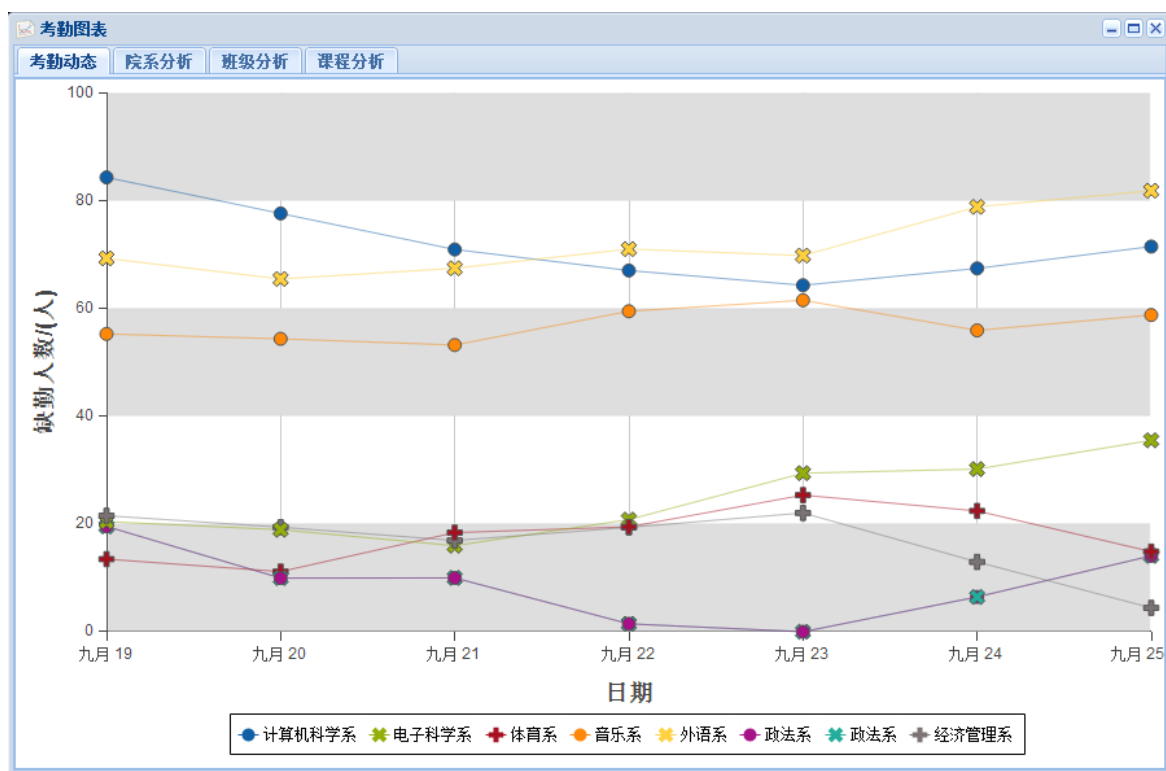
```
public function getClassAttendance() {
    // 获取学年、学期、年级、系别、班级、科目、上课时间、时间信息
    $query = $_GET['attendanceQueryObj'];

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

10. 考勤动态图表

- 1) 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;
- 2) 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;
- 3) 该功能仅教师端可以使用, 学生端没有权限。



(图 4.11)

11. 考勤院系分析图表

- 1) 系统展示不同系考勤状态的详细缺勤情况；
- 2) 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；
- 3) 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；
- 4) 该功能仅教师端可以使用，学生端没有权限。



(图 4.12)

部分代码:

```
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).'}';
    }
}

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`,
```

```

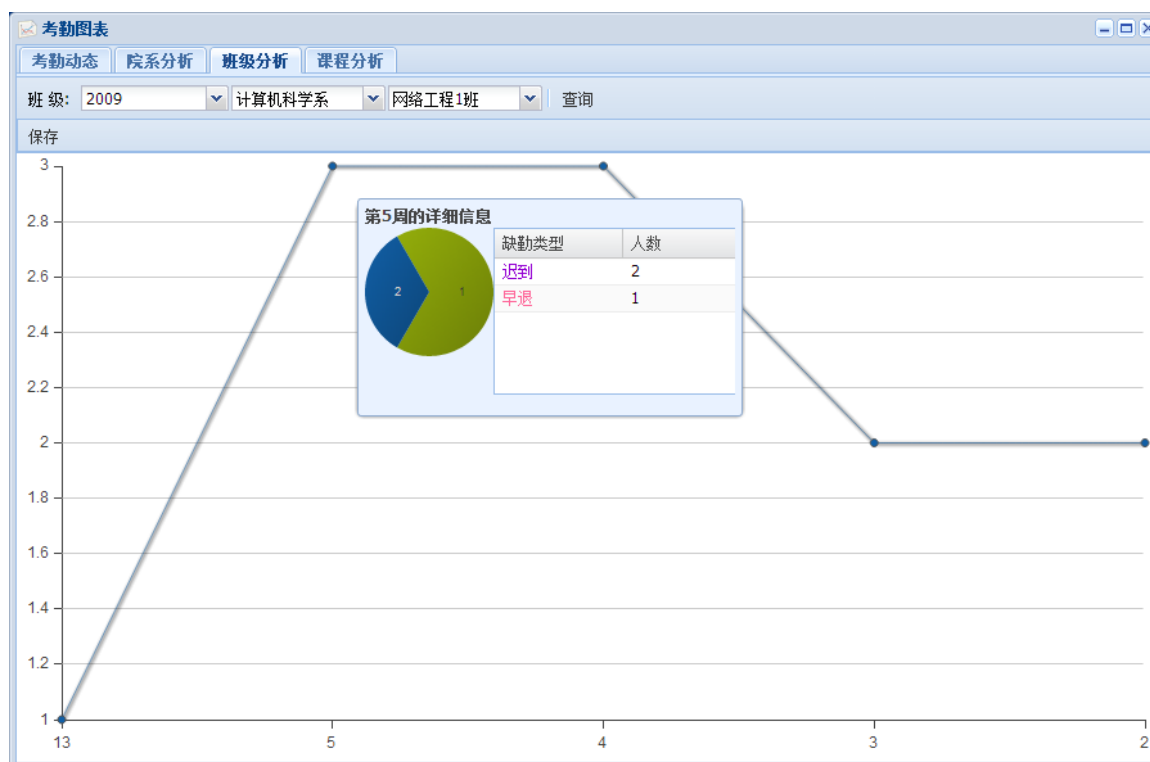
sum(case when `check`.`status`=2 then 1 else 0 end) as `chidao`,
sum(case when `check`.`status`=3 then 1 else 0 end) as `bingjia`,
sum(case when `check`.`status`=4 then 1 else 0 end) as `shijia`,
sum(case when `check`.`status`=5 then 1 else 0 end) as `gongjia`
from `check`
LEFT JOIN `schedule` ON `schedule`.`id`=`check`.`scheduleid`
LEFT JOIN `family` ON `family`.`id`=`schedule`.`familyid`
group by `schedule`.`familyid");

if($info) {
    echo '{"success":true,data:'.json_encode($info).'}';
}
}

```

12. 考勤班级分析图表

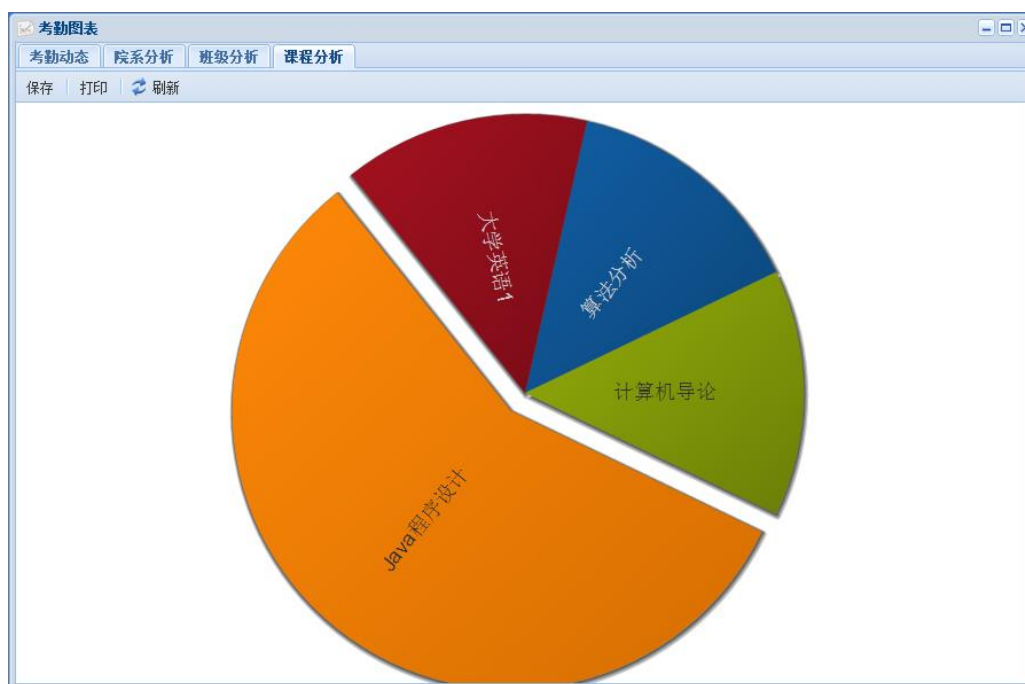
- 1) 在条件栏中输入相应的条件，敲击“查询”即可返回相应的结果；
- 2) 可以查询不同班级一个月内每天的缺勤人数；
- 3) 鼠标停留在某一天的对应节点上，可以以饼图的形式展示该天各个类型的缺勤人数；
- 4) 该功能只有教师端可以使用，学生端没有权限。



(图 4.13)

13. 考勤课程分析图表

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



(图 4. 14)

部分代码:

```
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).'}';
}
}

```

14. 发布通知公告

- 1) 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；
- 2) 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；
- 3) 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；
- 4) 该功能只有教师端可以使用，学生端没有权限。

(图 4. 15)

部分代码:

```

public function saveNotice() {
    $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
        $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
        $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");
        $noticeModel = M('Notice');
        for($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
        $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
        $noticeModel = M('Notice');
        $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}';
        }
    }
}

```



```
}  
}  
}
```

15. 微博应用爱监督

- 1) 首次进入该模块时，需要用户进行微博登陆授权该应用；
- 2) 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；
- 3) 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；
- 4) 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；
- 5) 该功能仅学生端可以使用，教师端没有权限。



(图 4.16)

5. 总结与展望

5.1 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，明白了“码农”与“程序员”的区别。
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。

5.2 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話，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，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。

4. 作为一个互联网产品,我认为要获得生命力,就需要开放平台。放眼各大互联网企业,无不打着开放平台的旗号,谷歌、百度、腾讯、新浪等等,均是如此。而我对此的一个分析是,一个人,一个团队的力量是有限的,而当我们的产品定位为一个平台之后,我们就不在是我们,而是更加广义的我们了。因为,世界各地的人,当然这里绝对不是单单指程序员,都可以参加我们这个平台的开发。所以,我觉得点名系统可以从教室点名出发,逐步走向这一步,开放自己的 API,最终成为一个校园平台。我认为,这是教室自动点名系统的最高价值,也是能够充分体现他的价值的一条路。

6. 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)

7. 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。