



**网络数据库技术**

**课程设计报告**

**学 院**：计算机学院

**专 业：** 网络工程

**指导老师：** 梁庆中

**题 目**： 公共考试统一报名系统

**小组成员：**姜瑞、简敏勇、王辉

2017年06月27日

前言

随着社会的快速发展，体力不再是我们唯一的生存方式了，人们也越来越注重自身的文化素养，随之而来的也有许多成人考试，为已经步入社会的人提供一个再学习的机会。

众所周知，作为学生，考试是我们必经的过程，选考也会非常容易。甚至在初高中时我们不需要选择考试，而是在学校的安排下直接参加考试。而对于已经步入社会的成人来说，他们选考就会比较的麻烦。一方面他们面临着工作的压力，另一方面工作之余还会有许多别的社会活动。为了不让选考成为广大人群的负担，我们小组一起设计编写了公共考试报名管理系统，目的就在于让所有人都有一个方便、公平的选考平台。不管是学生还是成人，都可以通过注册自己的信息，登录公共考试管理系统选择考试。

而一个考试报名管理系统，它所要面对的对象不仅仅是广大考生群体，还需要管理员来进行管理整个考试报名系统的运作。而考生和管理员作为不同的人员类型，当然具有不同的权限，并且具有不同的页面。考生的权限包括用户账号注册，修改自身注册信息，查询考试列表以及报名需要的考试，同时可以查看自己已报名的考试以及退选。管理员分为两种，第一种是超级管理员，它拥有此系统的最高权限，包括考生管理（添加考生，以及对已注册的考生进行管理包括编辑信息，和删除考生），考试管理（添加考试即发布考试，以及删除考试），管理员管理（添加、编辑管理员信息以及删除管理员）和系统设置（注册界面，考生登录界面的公告栏的编辑）；另一种是招生人员，它和超级管理员之间的差别就在于缺少对管理员的管理。

为使网页模块紧凑，功能完善，需对每一个模块都要进行检测，如在输入错误数据后，会给出相应错误提示；而在进行正确操作后，则进入相应的界面。

在本次设计中，通过“发现问题，分析问题，最终解决问题”三步。对系统作了认真的分析和测试，用实际操作实现了知识的积累和运用。并在开发过程中初步掌握PHP技术，MySQL数据库技术，git的使用方法以及协同编程的过程。通过设计，使我们对基础知识有了更深入的了解，并培养了团队的合作精神。

目录

[第一章 系统分析 3](#_Toc486883755)

[1.1 需求分析 3](#_Toc486883756)

[1.2 可行性分析 3](#_Toc486883757)

[第二章 总体设计 4](#_Toc486883758)

[2.1 项目规划 4](#_Toc486883759)

[2.2 系统功能结构图 4](#_Toc486883760)

[第三章 系统设计 7](#_Toc486883761)

[3.1 使用语言 7](#_Toc486883762)

[3.2 使用框架 8](#_Toc486883763)

[3.3 服务器 8](#_Toc486883764)

[3.4 MYSQL数据库 9](#_Toc486883765)

[3.5 源代码管理 10](#_Toc486883766)

[第四章 视图（view）主要模块详细设计 11](#_Toc486883767)

[4.1 视图文件总体架构 11](#_Toc486883768)

[4.2视图主要模块详细设计 11](#_Toc486883769)

[4.2.1 登录界面 11](#_Toc486883770)

[4.2.2 注册界面 12](#_Toc486883771)

[4.2.3 添加考生界面 12](#_Toc486883772)

[4.2.4 考生界面 13](#_Toc486883773)

[4.2.5 考试列表界面 15](#_Toc486883774)

[4.2.6 管理员界面 16](#_Toc486883775)

[第五章 模型（Models）主要模块详细设计 18](#_Toc486883776)

[5.1 数据库配置 18](#_Toc486883777)

[5.2 自定义模型类 19](#_Toc486883778)

[第六章 控制器（controllers）主要模块设计 20](#_Toc486883779)

[6.1 控制器文件总体架构 20](#_Toc486883780)

[6.2 控制器主要模块详细设计 20](#_Toc486883781)

[6.2.1 登录模块 20](#_Toc486883782)

[6.2.2 添加和编辑信息模块 21](#_Toc486883783)

[6.2.3 列表显示模块 21](#_Toc486883784)

[6.2.4 删除模块 22](#_Toc486883785)

[第七章 个人总结 23](#_Toc486883786)

# 第一章 系统分析

## **1.1 需求分析**

* 1. 实现公告栏的告示（包括注册公告栏和考生登录公告栏）
* 2. 实现用户的注册和登录
* 3. 实现用户对注册信息进行修改
* 4. 实现用户查看系统中已发布考试
* 5. 实现用户报名考试
* 6. 实现用户查看自己已报名的考试
* 7. 实现用户对自身已报名的考试进行退选
* 8. 实现管理员对考生进行添加
* 9. 实现管理员对考生进行管理（编辑，删除）
* 10. 实现管理员对考试进行添加即发布考试
* 11. 实现管理员查看已发布的考试并可以对相应考试进行删除
* 12. 实现管理员查看某门考试的已报名该考试的考生列表
* 13. 实现超级管理员对管理员进行添加，以及编辑管理员信息，删除管理员
* 14. 实现管理员进行公告栏的修改及发布
* 15. 系统运行稳定，具有强大的数据处理功能

## **1.2 可行性分析**

* 操作方便

1. 考生界面设计简洁，功能操作较为方便
2. 在界面切换的过渡时均有提示，使用户的操作更为明了
3. 当用户输入错误信息时，有提示弹窗对用户进行提示
4. 系统的各种设置较为简单

* 适用性广

1. 本系统并不是专为学校里的学生报名而做的，它是一个公共考试报名系统，意味着它还面向社会大众
2. 而本系统中的报名方式简单而不随便，适用于各种考试的报名

* 统一管理

1. 管理员可以对考生及考试进行统一管理
2. 统一管理使得管理员的工作变得更为方便，管理工作更为简单

# 第二章 总体设计

## **2.1 项目规划**

公共考试报名管理系统是一个典型的数据库开发应用程序，由前台考生报名管理区和后台管理员管理区组成，规划系统功能模块如下：

**前台考生报名管理区主要功能**：

修改考生自身注册信息，查看可报名的考试，对目的考试进行报名，查看已报名的考试，退选已报名的考试。

**后台管理员管理区主要功能**：

超级管理员：考生管理（添加，编辑，删除，查看列表），考试管理（添加，删除，查看某种考试的已报名学生，查看所有已发布考试），管理员管理（添加，编辑，删除），系统设置（编辑状态栏信息）。

招生人员：考生管理（添加，编辑，删除，查看列表），考试管理（添加，删除，查看某种考试的已报名学生，查看所有已发布考试），系统设置（编辑状态栏信息）。

注：招生人员与超级管理员是两类管理员。超级管理员拥有最高权限，而它与招生人员的主要区别就是它可以进行管理员管理，而招生人员是不具备这种功能的。

## **2.2 系统功能结构图**

网上报名系统功能结构图，前台报名管理模块如图1所示。

考生报名流程

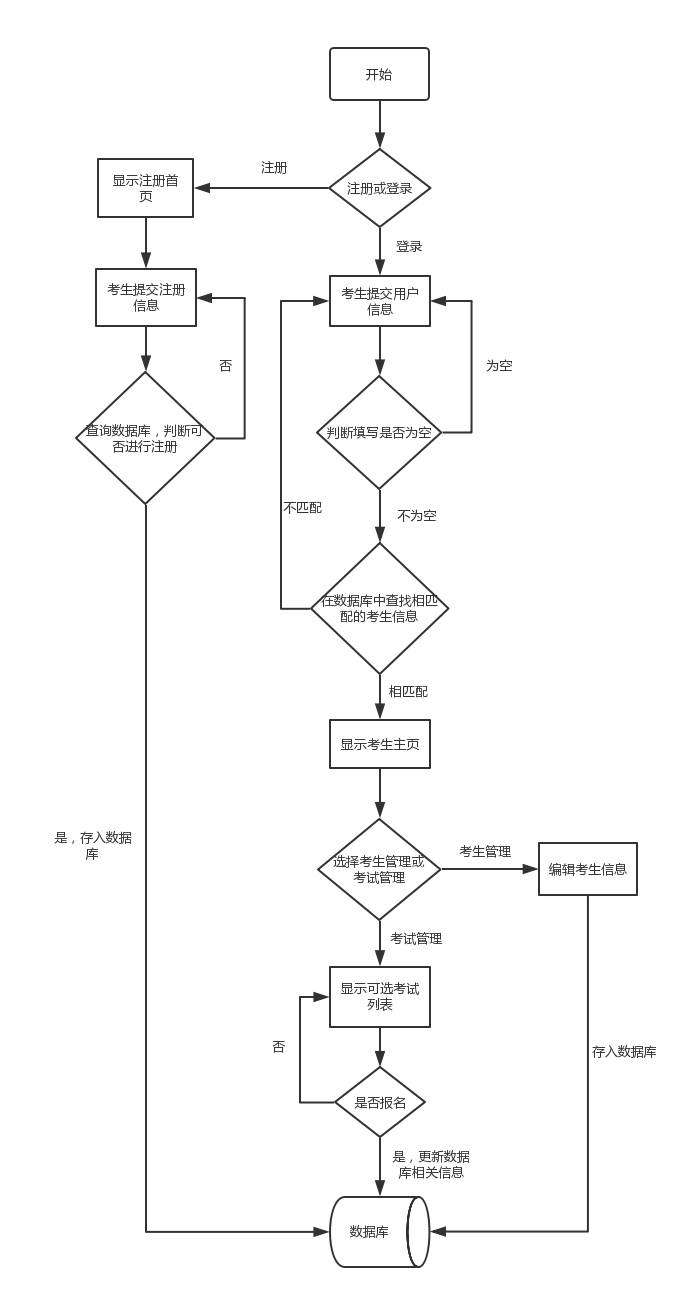


图1 考生报名流程

后台功能模块如图2所示：

管理员流程图

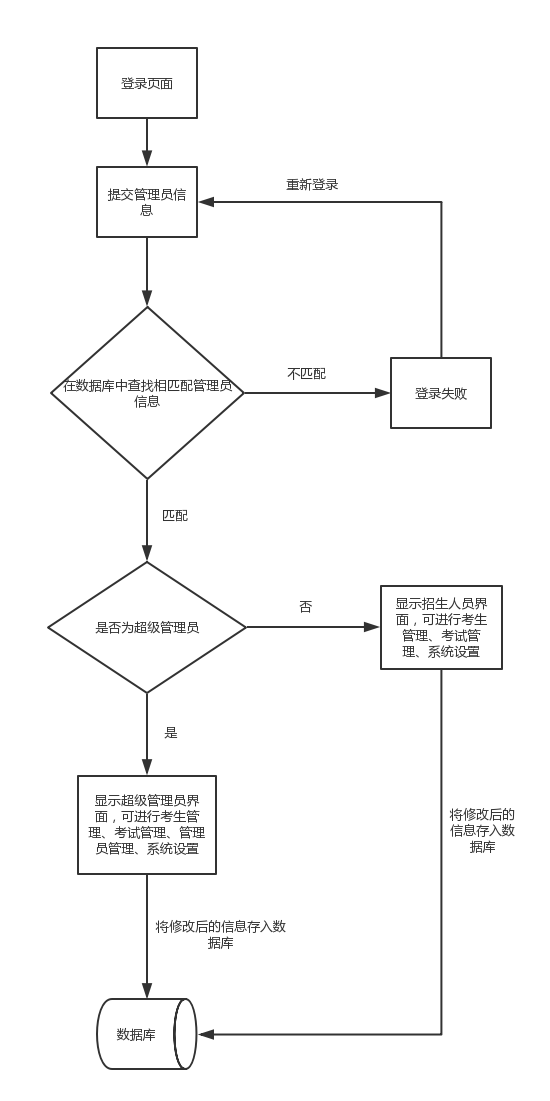


图2 管理员登陆图

# 第三章 系统设计

## 3.1 使用语言

在本次课程设计对公共考试报名管理系统的设计中，我们采用PHP脚本语言并嵌入HTML脚本来进行网站的设计。

PHP是一种[开源的](https://zh.wikipedia.org/wiki/%E5%BC%80%E6%BA%90)通用[计算机](https://zh.wikipedia.org/wiki/%E8%AE%A1%E7%AE%97%E6%9C%BA)[脚本语言](https://zh.wikipedia.org/wiki/%E8%84%9A%E6%9C%AC%E8%AF%AD%E8%A8%80)，尤其适用于[网络开发](https://zh.wikipedia.org/wiki/%E7%BD%91%E7%BB%9C%E5%BC%80%E5%8F%91)并可嵌入[HTML](https://zh.wikipedia.org/wiki/HTML)中使用。PHP的语法借鉴吸收[C语言](https://zh.wikipedia.org/wiki/C%E8%AF%AD%E8%A8%80)、[Java](https://zh.wikipedia.org/wiki/Java)和[Perl](https://zh.wikipedia.org/wiki/Perl)等流行计算机语言的特点，易于学习。PHP的主要目标是允许网络开发人员快速编写[动态页面](https://zh.wikipedia.org/wiki/%E5%8A%A8%E6%80%81%E9%A1%B5%E9%9D%A2)。

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。

使用PHP进行WEB应用程序开发，具备如下的优点：

        ◆  解释与编译PHP是性能优越的编译程序，又具备解释过程的优点。

        ◆  开发效率高，函数语言简洁明了。

        ◆  输出控制灵活，可在HTML中内嵌PHP代码，也可以由PHP输出HTML运行，也可以在命令行下执行，将结果输出到其他设备。

        ◆  可实现模板化，实现程序逻辑与用户界面分离。

        ◆  跨平台可运行在Win32或UNIX/Linux/Macintosh/FreeBSD/OS2等平台上。

        ◆  与多个WEB服务器兼容，如Apache、MS IIS、Netscape Server等。

        ◆  完全支持面向对象开发，并向下兼容，支持过程与面向对象两种风格的开发。

        ◆  内嵌Zend加速引擎，性能稳定快速。

        ◆  PHP编写容易，内置函数丰富，几乎涵盖了WEB开发的所有方面。

        ◆  组件化开啊，提供MySQL、Oracle、MS SQL等多种数据库的访问接口，支持ODBC。

        ◆  扩展性好，支持访问Win32系统的COM对象。

        ◆  支持正则表达式，内置POSIX与Perl兼容两类的正则表达式支持。

        ◆  开发成本低，开发工具多，且有众多使用PHP开发的开放源代码项目供我们参考和二次开发。

        ◆  完全支持企业级开发，PHP并非只能做网站，在国内外有金融行业、能源行业、打得政府型采购系统采用PHP开发系统平台，并且近年来IBM、Adobe、Intel等行业巨头在为PHP以及其商业公司Zend提供了巨大的资本投入，并得到Oracle、Adobe、微软等公司的支持，另外以PHP软件开发商如NuSphere、Borland等公司均在为PHP的企业化助力驰援。

        ◆  PHP5的下一版本6.0将全面支持多线程以及Unicode-16、il8n、字符集整理(Collation)、亚洲语言翻译(str\_transliterate)等国际化及其他新技术。

        ◆  支持桌面级系统开发

              PHP不仅能够开发动态网站系统，还能够开发Win32/X-window桌面级（PHP GTK、Winbinder）应用程序，以及Shell或命令行下运行的deamon守护脚本以及服务器端管理程序。

        ◆  支持加密分发代码

              在使用Java、.net这些虚拟机字节码的语言，在某些时候非常容易被反编译，导致一些安全问题，使用Zend Optimizer不仅可以使PHP实现“编译”运行，不仅实现速度的飞跃，而且可以实现PHP源代码的完全加密，从而保护作者的利益以及软件版权。这是PHP的最大特点之一。

## 3.2 使用框架

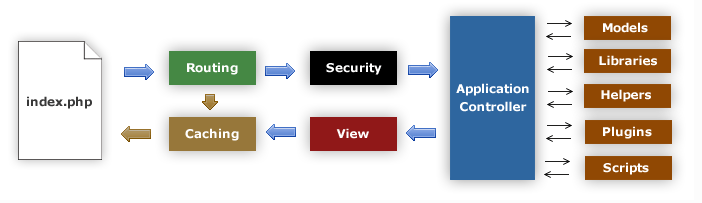
在本次程序设计中，为了方便进行web开发，我们使用了CI框架来进行网站开发。

**CodeIgniter**是一套给[PHP](https://zh.wikipedia.org/wiki/PHP)网站开发者使用的应用程序开发框架和工具包。它提供一套丰富的标准库以及简单的接口和逻辑结构，它可以为开发者们建立功能完善的 Web 应用程序，其目的是使开发人员更快速地进行项目开发。

CodeIgniter 的开发基于 MVC（模型-视图-控制器）设计模式。MVC 是一种用于将应用程序的逻辑层和表现层分离出来的软件方法。

* **模型（Model）** 用于封装与应用程序的业务逻辑相关的数据以及对数据的处理方法。“ Model ”有对数据直接访问的权力，例如对数据库的访问。“Model”不依赖“View”和“Controller”，也就是说， Model 不关心它会被如何显示或是如何被操作。但是 Model 中数据的变化一般会通过一种刷新机制被公布。为了实现这种机制，那些用于监视此 Model 的 View 必须事先在此 Model 上注册，从而，View 可以了解在数据 Model 上发生的改变。（比较：[观察者模式](https://zh.wikipedia.org/wiki/%E8%A7%82%E5%AF%9F%E8%80%85%E6%A8%A1%E5%BC%8F)（[软件设计模式](https://zh.wikipedia.org/wiki/%E8%BD%AF%E4%BB%B6%E8%AE%BE%E8%AE%A1%E6%A8%A1%E5%BC%8F)））
* **视图（View）**能够实现数据有目的的显示（理论上，这不是必需的）。在 View 中一般没有程序上的逻辑。为了实现 View 上的刷新功能，View 需要访问它监视的数据模型（Model），因此应该事先在被它监视的数据那里注册。
* **控制器（Controller）**起到不同层面间的组织作用，用于控制应用程序的流程。它处理事件并作出响应。“事件”包括用户的行为和数据 Model 上的改变。

以下是CI框架定义的应用程序流程：



1. index.php 文件作为前端控制器，初始化运行 CodeIgniter 所需的基本资源；
2. Router 检查 HTTP 请求，以确定如何处理该请求；
3. 如果存在缓存文件，将直接输出到浏览器，不用走下面正常的系统流程；
4. 在加载应用程序控制器之前，对 HTTP 请求以及任何用户提交的数据进行安全检查；
5. 控制器加载模型、核心类库、辅助函数以及其他所有处理请求所需的资源；
6. 最后一步，渲染视图并发送至浏览器，如果开启了缓存，视图被会先缓存起来用于 后续的请求。

## 3.3 服务器

在本次程序设计中我们选用的服务器为Apache，当然我们使用的是xampp，一个Apache，MySQL等的集成软件。

将Apache的配置文件中的路径DocumentRoot改为我们的项目所在的路径，即搭好了我们的服务器。

## 3.4 MYSQL数据库

由于我们采用的是xampp，一个Apache、MySQL等的集成软件。

它里面存在一个对MYSQL的管理工具PHPMyAdmin。因此我们首先写好一个创建数据库，数据表的SQL语句的.sql文件，然后将其导入该管理工具。

本系统创建的数据库名称为management。数据库中共包含五张表，分别为student，exam，admin\_core，student\_exam，info。



**Student表**：用于存储学生信息。主键为考生号，即id，并定义考生号必须为11位。

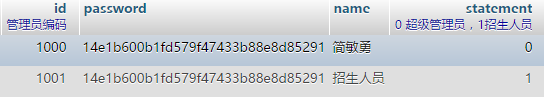


**注**：此处插入的数据使用的密码均为123456。而插入数据时使用了两次md5加密，所以上面显示的一样。

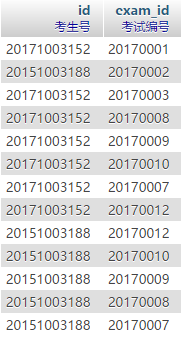
**Exam表**：用于存储考试信息。主键为考试编号，即id。



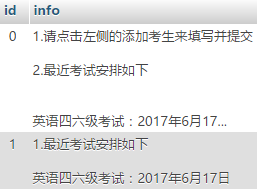
Admin\_core表：用于存储管理员信息。主键为管理员编号，即id。在这里将管理员编号定义为四位，若不是四位则提示错误。



**Student\_exam表**：用于存储考生的报名信息，每当考生报名某门考试，则将考生的id和考试的id一并存入作为一条数据。



**Info表**：用于存储公告栏的信息。主键为id，0表示注册界面的公告栏，1表示考生登录界面的公告栏。

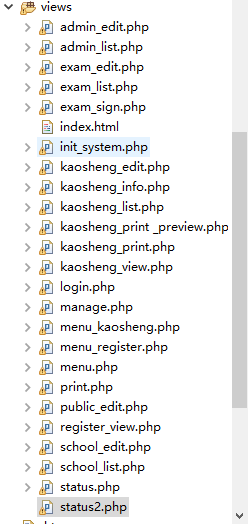


## 3.5 源代码管理

使用国内源代码托管服务coding来进行源代码及版本的管理通过增加SSH公钥的方式来给予组内成员推送的权限。

# 第四章 视图（view）主要模块详细设计

## 4.1 视图文件总体架构



## 4.2视图主要模块详细设计

### 4.2.1 登录界面

在登录界面创建了三个文本框，分别是用来进行输入用户名，密码，以及验证码。而输入验证码的文本框右侧有一张显示验证码的图片，它的来源是index控制器的random函数。

在此界面还包括两个按钮——注册和登录。注册按钮是一个超链接，连接到index控制器的register函数。登录按钮是提交按钮，当按下这个按钮，form表单中的数据则将传送到由action定义的index控制器的login函数中。

在这个文件中使用JavaScript定义了一个函数，当用户在文本框中填入的数据为空的时候直接进行弹框显示。

以下是登录界面的截图：



### 4.2.2 注册界面

注册界面由考生须知和添加考生组成。考生须知是注册界面的首页，这里主要是一个公告栏显示对注册考生的提示信息，而公告栏中的数据是由控制器index中的register函数通过查询数据库传入本视图的。添加考生视图在下一节进行解释。



### 4.2.3 添加考生界面

添加考生界面是一个被多角色多任务复用的视图。多角色说的是它同时被考生和管理员使用，多任务是说它进行多种页面显示。

添加考生界面同时也是编辑考生界面，它在控制器传进来不同的参数时显示不同的标题。控制器传给此视图一个list数组，当这个数组中的action键中存储的值是‘add’时，显示的标题为“添加考生”，而当list数组中的action键中存储的值是‘edit’时，显示的标题是编辑考生。这也就是进行页面的复用。

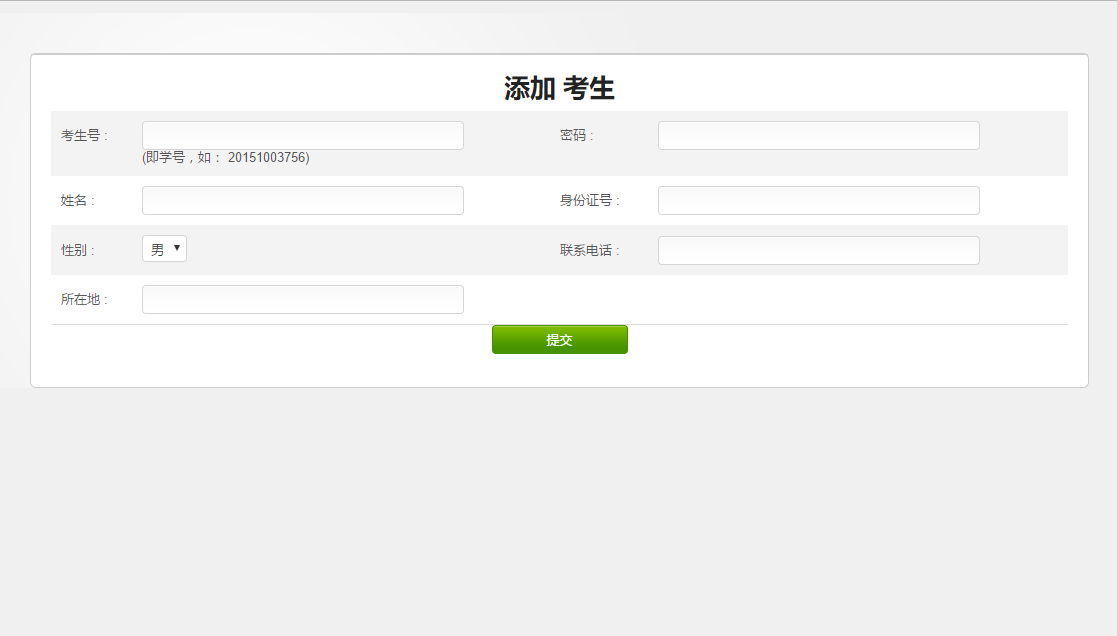
而在添加考生和编辑考生的情况中，表单提交的目的地是不一样的。因此在form表单的action定义时，也需要进行对list数组中action键的值的判断，然后将action定义为不同的值。

接下来说一下这个页面的主要组成部分，这个页面包括六个文本框和一个选择框，文本框用于对考生基本信息的输入，选择框用于选择性别。提交按钮则是用来提交表单中的数据。

在这里，由于添加和编辑考生任务的不同，需要对页面进行一些必要的处理。例如在编辑考生的时候，我希望它是可以将原来考生的信息输出到文本框中的，这有利于我对它进行编辑。而出于安全考虑，在编辑考生时我们没有将它的密码显示出来，当考生在密码的文本框中填入了新密码时，我们将它一并提交，并进行数据库的更新，而当考生不需要改密码，则可以不进行填写，这里就提交一个空的密码到控制器。而控制器判断到密码为空时，则进行数据库的查询，将该考生原来的密码再重新更新至数据库。

注：考生号是不允许编辑的，因为在控制器端更新数据库时，是以它来作为查询的关键码。

以下是添加考生界面截图：



### 4.2.4 考生界面

当考生进行登录后，进入到考生界面。考生界面由考生管理和考试管理组成。考生管理包括考生须知，考生信息，以及修改信息。考生须知是考生界面的主页，和前面注册界面的考生须知一样，只不过这两个界面的公告栏是不同的，管理员可以对这两个公告栏里的信息进行编辑显示。考生信息界面则是显示考生基本信息的页面，在控制器中使用该考生的考生号查询数据库，将得到的数据分别给list数组中的不同键，然后将list数组传进本页面，通过不同的键值进行输出即可。修改信息界面也就是编辑考生界面，在上一节中已经介绍过。

考试管理包括报名考试和考试列表两个界面。

**报名考试界面**显示所有已发布考试的具体信息，以及可以进行考试的报名操作。控制器中查询考试表得到考试表中所有的信息，将它存在一个list数组中的list键中，并将list数组传入此页面，然后经过for循环将list数组送list键中的数据进行一行一行的输出。

而在输出每一行考试信息的末端，有一个报名的超链接，当按下这个超链接则将连接到exam控制器的exam\_signdo的函数中，并将考试的考试编号一并传过去。在控制器端通过Session得到考生的考生号然后再讲考生号和考试编号一并插入student\_exam表中。

在这个页面中，当考生报名了某门考试时，我们想要的是之前报名的超链接被“已报名”取代。实现这个的方法是在这个视图中通过session得到考生id然后与考试id一起作为条件来进行查询student\_exam表，当查询到表中存在某行数据包括这两个ID，则表示考生已报名该门考试，则输出“已报名”三个字。若是不存在该数据，则输出“报名”的超链接。

该段代码如下：

<td>

<!-- 如果发现已经报名。则显示已报名 -->

<?php

$this->load->model ( 'Data\_model' );

$data ['query'] = $this->Data\_model->get\_exists\_data ( **array** (

'id' => $this->session->userdata('id'),

'exam\_id' => $item['id'],

), 'student\_exam' );

**if**($data['query']>0){

**echo** *已报名*;

}**else**{?>

<!-- 未报名则可以报名 -->

<a href="<?php **echo** *site\_url*("exam/exam\_signdo/".$item['id']);?>"

title=*"报名"*>报名</a>&nbsp;&nbsp;

<?php }?>

</td>

考试列表界面由下一节来进行阐述。

考生主界面：



报名考试界面：



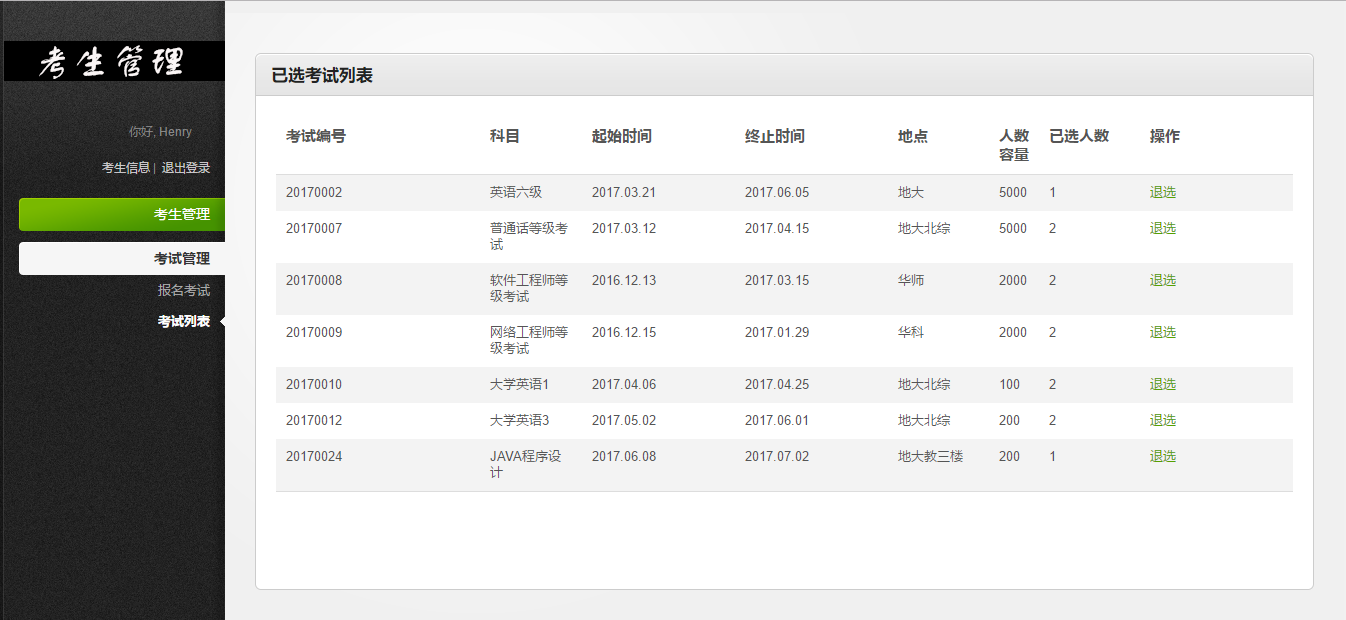
### 4.2.5 考试列表界面

考试列表界面是一个复用的界面，也就是考生和管理员复用的界面。考生端要显示的是自已选的考试列表，而管理员界面显示的考试列表要显示的是考试表里所有的考试。

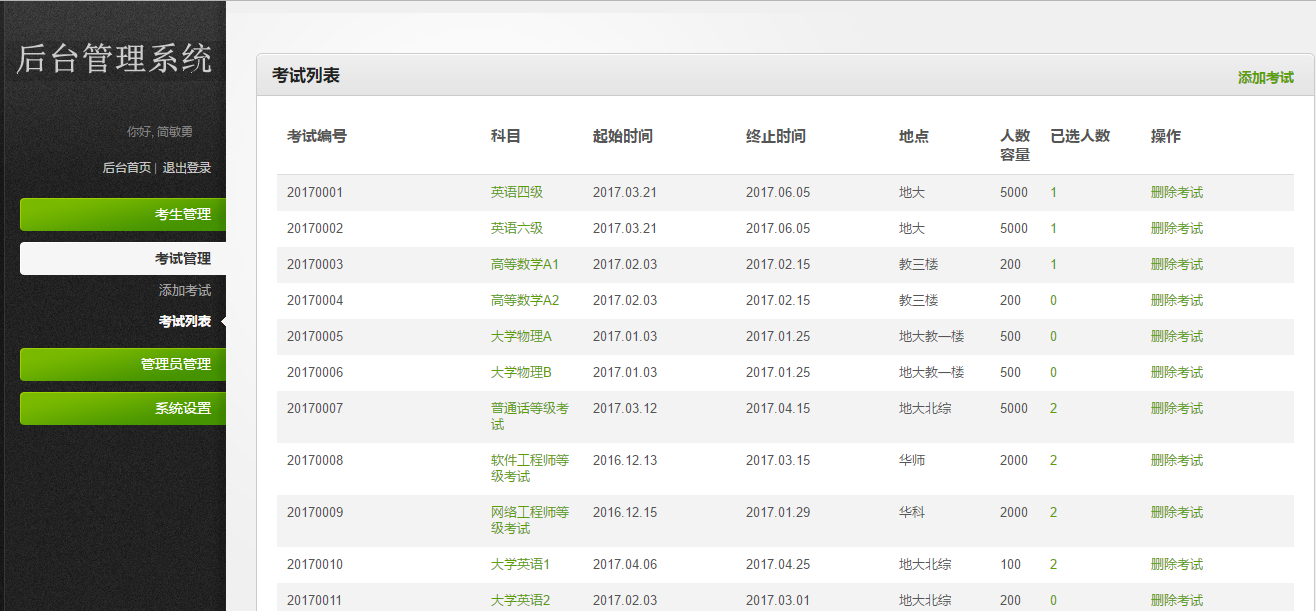
因此这个界面在不同角色登录时所要展示的页面使不同的，而这些不同信息的展示则是通过控制器加载此页面时所传过来的list数组来实现的。例如考生界面中考试列表的界面的标题是已选考试列表，管理员界面中显示的标题是考试列表，这个标题便是存在list数组的info键中。在这里只要将这个键中的值输出即可。

而在这个视图中，有些内容是在管理员中显示而在考生界面是不显示的，有些界面则是只显示在考生界面中。要实现这些则是通过session来实现，提取session中所存储的相应信息，得到这个用户的身份（这是因为在用户登录中，将这个用户的角色身份存进了session），然后经过判断其身份的不同来使得界面显示的不同。

考生界面的考试列表：



管理员界面的考试列表：



### 4.2.6 管理员界面

当管理员进行登录后，则进入管理员界面。管理员界面包括考生管理，考试管理，管理员管理和系统设置，其中管理员管理是超级管理员才有的权限，一般的管理员（招生人员）不具有该权限。

**考生管理**包括添加考生和考生列表，其中添加考生在4.2.3节中讲过，考生列表的基本原理和考试列表差不多，而考生列表界面也是一个复用界面，它在这里是显示所有考生的信息，并且可以进行删除和编辑考生。而在考试列表中点击某门考试的已选人数时，也会调用考生列表界面，在这个界面中显示的是该门考试所对应的所有考生的信息（也就是考生报名了该考试科目）。这两个界面和之前讲的考试列表的复用类似，就不再赘述。

**考试管理**包括添加考试和考试列表。添加考试和之前讲的添加考生界面类似，考试列表在4.2.5节中讲过。

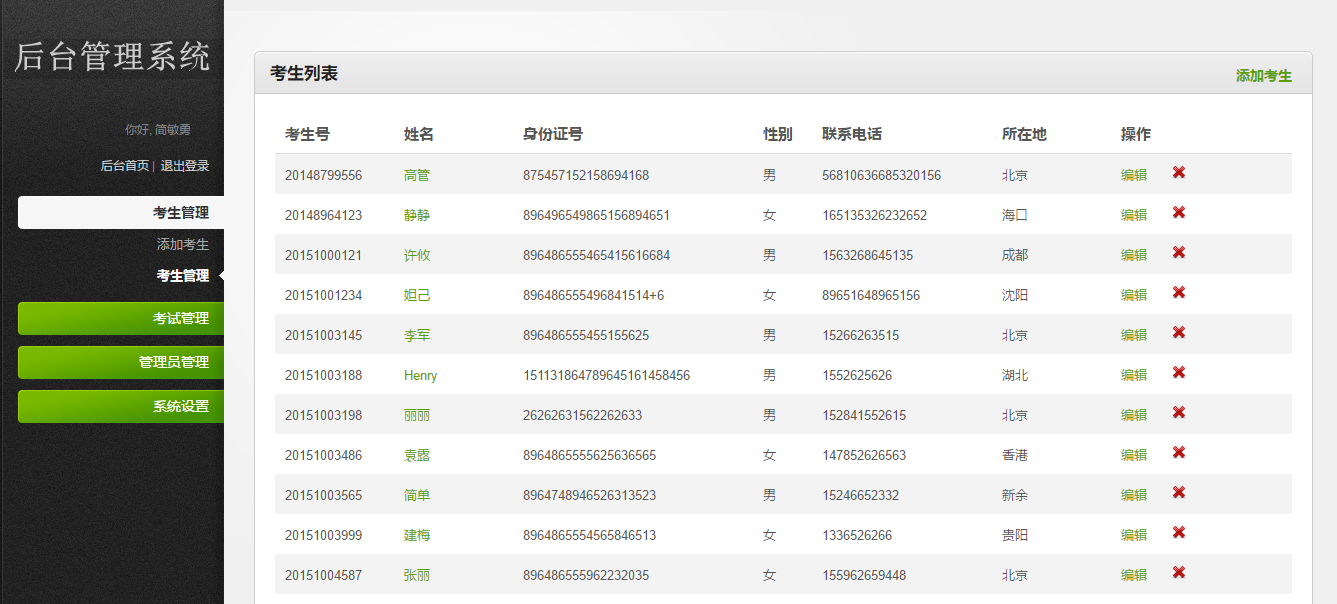
**管理员管理**包括添加管理员和管理员管理。添加管理员和之前的添加考生界面类似，管理员管理则是显示管理员表中的所有管理员信息，并可以进行编辑和删除，这和之前的列表界面类似。

**系统设置**包括的就是公告栏的编辑界面，其中包括两个公告栏的编辑。每次按下提交按钮的时候，则将文本域中的内容提交至相应控制器，在控制器端更新相应数据库。

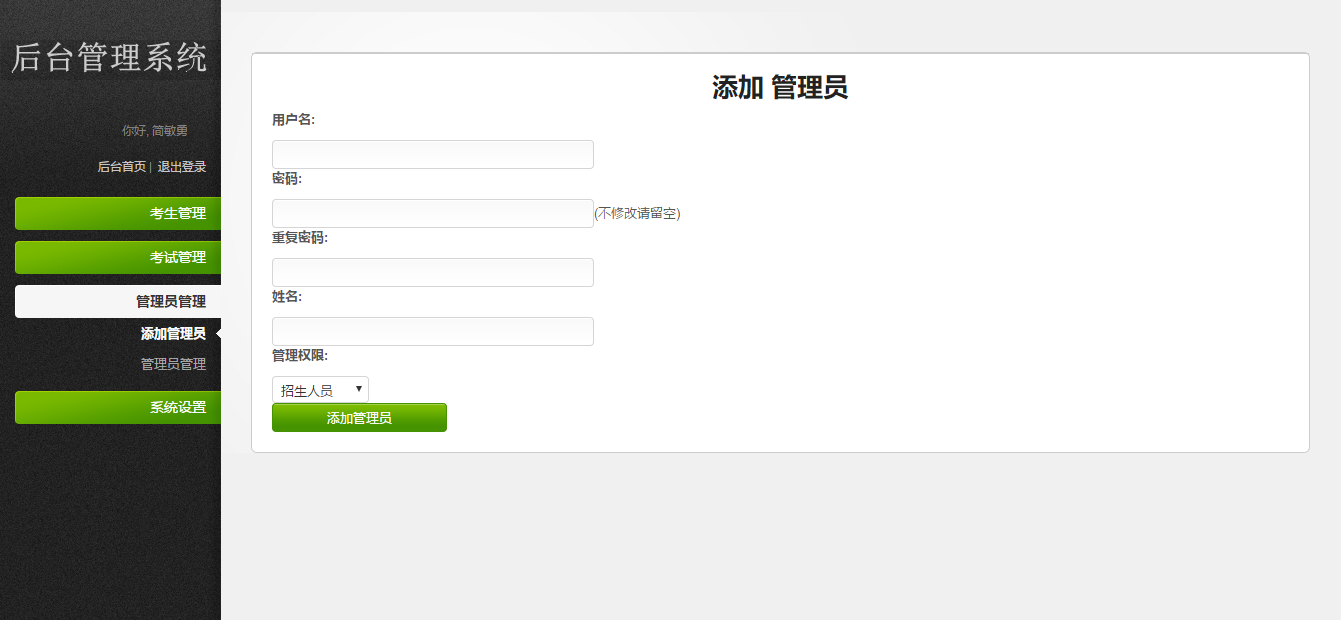
管理员主界面：



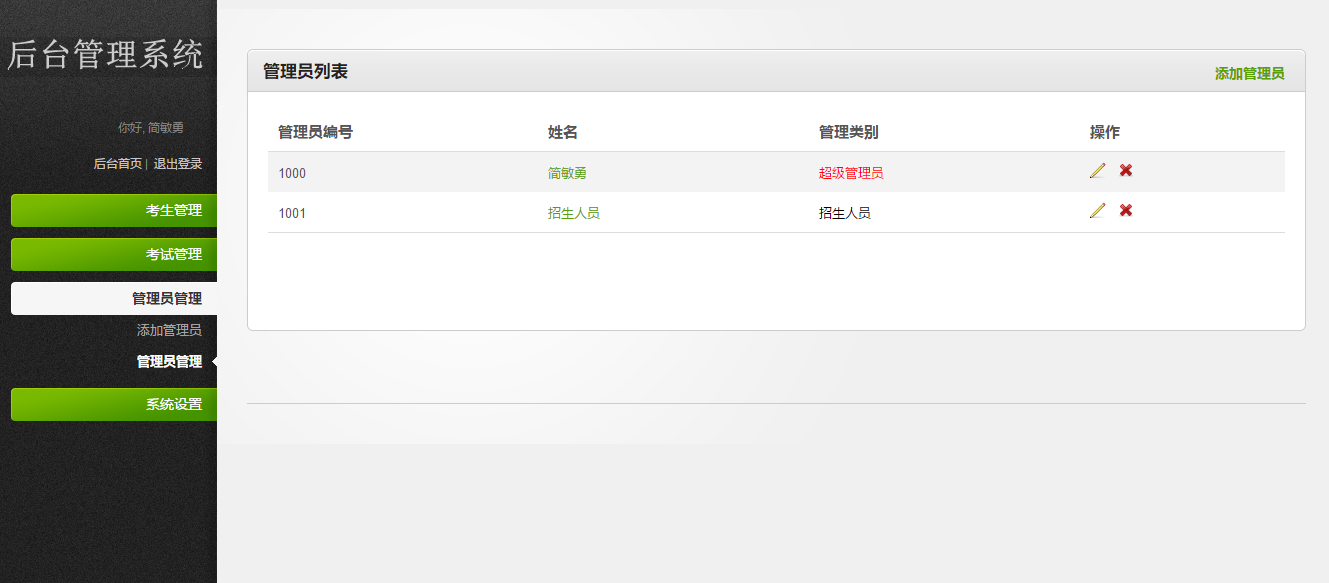
考生管理界面：



添加管理员界面：



管理员管理界面：



系统设置-修改公告栏界面



# 第五章 模型（Models）主要模块详细设计

## 5.1 数据库配置

通常来说，模型类将包含帮助我们对数据库进行增删改查的方法。在本节我就先讲讲数据库的配置。CodeIgniter 有一个配置文件用来保存数据库配置（用户名、密码、数据库名等等），这个配置文件位于 application/config/database.php。

配置文件中的语句如下：



## 5.2 自定义模型类

在模型类中已经包含对数据库进行增删改查的方法，而在这里我们将自定义一个类，让它继承模型类，并在这个类中自定义一些函数来方便我们的控制器的调用。

自定义的函数通常是通过传入不同的参数，然后调用模型类中已经有的对数据库进行增删改查的方法，来进行对数据库的操作，通常该函数都会将一个以对象形式返回的结果集返回。

下面这个是自定义的一个函数，它的作用是获取某个学生已报名的考试，代码如下：

**function** get\_alldata($id,$table1,$table2)

{//获取某个学生已报名的考试

$query2 = $this->db->get\_where($table1, **array**('id'=>$id)); //$some 类似数组 array('city\_name' => $city\_name)

$num=$query2->num\_rows(); //返回条数，

**if**($num>0){ $query1=$this->db->where('id',$id)->select('exam\_id')->get($table1)->result\_array();

$re=**array**();

**foreach**($query1 **as** $resp){

$re[]=$resp['exam\_id'];

} $query=$this->db->where\_in('id',$re)->get($table2)->result\_array();

**return** $query;

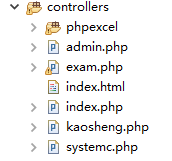
}**else**

**return** ' ';

}

# 第六章 控制器（controllers）主要模块设计

## 6.1 控制器文件总体架构



## 6.2 控制器主要模块详细设计

控制器中的函数众多，因此我就挑几个主要的模块来进行讲述。这几个主要的模块分别是登录模块（讲述考生和管理员的不同身份登录），添加和编辑信息模块（以添加考生和编辑考生为例进行讲述），列表显示模块（以考生界面的“已选考试列表”和管理员界面的“考试列表”为例进行讲述），删除模块（以删除考生作为重点讲述对象）。

### 6.2.1 登录模块

index.php 文件作为前端控制器，初始化运行 CodeIgniter 所需的基本资源。登录视图是由index.php所定义的运行程序时首次加载的页面。而登录视图中定义了将表单数据提交至index控制器的login函数中，因此在这个函数中我们将对表单中传入的数据进行处理。

首先用以下语句得到表单提交的数据，并赋给相应变量。

$manage\_name = *trim*(*htmlspecialchars* ( $this->input->post ( 'username' ) ));

$manage\_password = *trim*(*htmlspecialchars* ( $this->input->post ( 'password' ) ));

$yzm = *trim*(*htmlspecialchars* ( $this->input->post ( 'yzm' ) ));

由于在视图中form表单中定义的method为“post”，所以在这里使用post方法来得到表单提交的数据。Trim则是起到忽略原字符串中的空格的作用。

这样便得到了用户在文本框中填写的登录信息，我们首先就要判断其验证码填写是否正确，然后再判断用户名、密码填写是否为空，若是，则进行相应的弹窗显示。同时退出本函数。

接下来才是进行查询数据库来判断该用户能否登录。

由于考生和管理员是在同一个登录界面进行登录的，但是我们不能将考生和管理员归为一类来进行查询，因为他们在不同的表中。因此必须用一个判定条件来将管理员和考生区分开，然后进行不同表的查询。在本程序中我们定义的规矩是管理员的管理员编号（id）必须为四位，而考生号（id）则必须为11位。所以在这里，我们已经得到了用户登录时输入的用户名也就是id，因此首先判断它为多少位，若是4位，则说明他是管理员，若是11位则说明是考生，若二者都不是，则弹窗显示用户名输入错误。

在控制器中需要查询数据库时，首先加载相应的自定义的函数所在的模型，然后再调用这个模型中的函数。

当输入的为管理员账号，则现在需要判断的是他是否为管理员，能否进行登录，首先通过自定义函数通过输入的用户名和密码来进行查询管理员表，若是存在则首先得到该管理员的一部分信息，然后存进session。接下来进行页面跳转至管理员界面。

当输入的为考生账号，则进行查询考生表，若是考生表中存在该用户，则得到该考生的一部分信息，然后存进session。然后跳转考生界面。

在存入session的信息中包含有一个是确定用户的身份信息的，它的关键码是‘statement’，若是它的值为0，则表示超级管理员，若是它的值为1，则表示招生人员，若是它的值为2，则表示考生。这个在之前的视图模块中用到了。

### 6.2.2 添加和编辑信息模块

添加和编辑信息模块包括添加考生，添加考试，添加管理员，编辑考生信息，编辑管理员信息。这里就以添加考生和编辑考生为例进行讲述。

**添加考生**：在控制器端得到视图页面中所提交的表单数据，然后要做的当然是将这些信息插入考生表。但是在插入之前要做一些判断，首先需要确定填入的所有信息都不为空，然后需要判断其考生号填写的是否为11位，接下来则需要判断的是这个考生是否已经存在于数据库中，所以需要调用模型中的函数来进行操作。以上条件只要不符合条件就进行弹窗显示错误。若都符合条件，则进行插入操作。

**编辑考生**：在控制器端得到视图页面中所提交的表单数据，编辑考生的时候，若是用户没有对考生的密码进行修改，则控制器端得到的密码为空，这时候就需要在这里根据传过来的考生ID来进行查询该考生密码，得到之后再进行考生所有信息的更新（考生号（id）不允许编辑）。

注：插入密码时将密码经过两次md5加密后再进行插入。

### 6.2.3 列表显示模块

列表显示包括考生界面的考试列表和已选考试列表，管理员界面的考试列表，考生列表，管理员列表等等，在这里就以考生界面的“已选考试列表”和管理员界面的“考试列表”为例进行讲述。

因为这两个视图都是提交给同一个控制器端的函数进行处理，所以在这个函数中需要判断登录用户的身份，通过其登录时存在session中的一个statement的值来判断。若判断后，该用户为考生的话，则在这个函数中我要得到的是该考生已报名的考试。这时候调用模型中一个专为这个功能封装的函数，即可得到这个考生所选的考试的所有信息。而模型中的这个函数的实现的过程如下：首先判断该考生是否有报名的考试，即用这个考生的id去查询‘student\_exam’表，若是查询到的结果不为空则进行接下来的操作。接下来再通过考生的id去查询同一个表，选择‘exam\_id’属性，也就得到了该考生所有已报名的考试。但这只是得到了考试的id，接下来则用考试id去查询‘exam’表，再将得到的结果返回。控制器将返回的结果集赋给list数组的一个键，最后在加载相应页面时将list数组传进去即可。

而管理员界面的考试列表则是直接显示‘exam’表中的所有考试的所有信息。也就是直接将‘exam’表中数据全部返回即可。控制器将返回的结果集赋给list数组的一个键，最后在加载相应页面时将list数组传进去即可。

### 6.2.4 删除模块

删除模块包括考生删除，考试删除，管理员删除，某一条报名信息的删除。这里以考生删除作为例子来讲述删除的步骤。

当用户在视图端摁下删除考生时，会传入一个考生的id进入控制器kaosheng的kaosheng\_del函数中，函数接收到参数之后则要进行删除工作。

而当一个考生被删除，也就意味着它的报名考试的记录也应该被删除，而它所报名的考试的报名人数也应该减1。这就是这个函数的工作。

首先得到该考生所选的所有考试的具体信息选择其中的‘id’，‘number’属性，之后将得到的数据进行减1，然后在for循环中通过id将‘exam’表中的‘number’进行更新。然后在‘student\_exam’表中通过考生的id将与其相关的所有信息删除。最后将‘student’表中的该考生删除。然后弹窗显示“删除成功”。

关键代码如下：

$this->load->model ( 'Data\_model' );

$que=$this->Data\_model->get\_alldata ( $id, 'student\_exam','exam' );

**foreach**($que **as** $resp){

$rea=$resp['id'];

$re=$resp['number']-1;

//更新考试的已选人数

$this->Data\_model->update\_data( $rea,**array**('number'=>$re), 'exam' );

}

//删除学生\_考试表中的该考生所选的所有考试

$data ['query2'] = $this->Data\_model->delete\_data( $id, 'student\_exam' );

//删除学生表中的信息

$data ['query'] = $this->Data\_model->delete\_data ( $id, 'student' );

# 第七章 个人总结

**小组成员：姜瑞，简敏勇，王辉**

**姜瑞**：后端的流程处理和前端的显示安排

**总结**：在这次网络数据库的课程中，让我第一次如此接近地了解到了一个网站开发，初步了解了php的运行方式，在协同工作中能更好地调配自己与其他同伴之间的工作安排。我在项目中主要负责的是后端的流程处理和前端的显示安排。

第一部分：首先说说学了什么，大概从上个学期开始就开始接触git这样的源代码管理工具，从基本不会使用到了解了各种分支以及合并冲突的操作，了解了密码的认证方式，也就是ssh-keygen(ssh)的使用并且记录到了博客。在原本稍微了解了一点HTML的内容之后开始了进行php的学习。使用php来进行后端的流程控制，使用框架来满足mvc模式的需要。在JavaScript学会了简单的使用判断。

第二部分：确定了下一步的目标，JavaScript这个门非常火的语言现在很多地方都已经用到了前端到后端的使用，而且github上面近70%的项目都是JavaScript的项目，非常的强大。如果在接下来的学习时间里面还有空闲的时间将会把JavaScript放在第一位的学习方向。

第三部分：感想和体会，php在调侃中总被提到是“世界上最好的语言”，说明了它的学习成本比较低，在之前看完了php的基本教程之后就直接上手开始通过HTML写好的模板开始进行网站的开发，因为之前java的课设比较熟悉http协议的方式，过程中并没有什么阻碍的感觉，这门语言在网站的开发方面确实比较方便，不过这样的一个报名系统也并没有涉及太复杂的操作。在学习的过程中的一个非常深刻的体会是如果有了C/C++的基础再去学习一门语言简单的使用确实比较快(深入使用可能还是需要比较长的时间)。在这次的项目完成之后了解了原本看似神秘的网站大概的开发方式，对自己在的贡献和成果还是比较满意的，同时也认识到了一个大型项目的开发和小型自己的程序之间差异性，合作很重要。

**简敏勇**：所有有关数据库的操作

**总结**：这一次网络数据库的课程设计做一个网站，而写网页在本学期的Java课程中也进行学习过。当时在做Java的课程设计时，我也曾利用Java连接数据库来进行一些操作。当时使用的数据库是微软的Access，而在Java中的加载驱动，连接数据库也还算简单，所以对数据库这方面我还算是有一定的了解。因此本次课程设计，我选择负责有关数据库的一些操作。

这次课程设计，我们使用的语言是PHP，其实刚开始组长决定使用PHP的时候，我是有一些异议的，因为这门语言我几乎都没接触过，更何况来用它进行网页开发。但经过了解，我决定先学习学习，当我大概的了解这门语言，又了解到CI框架，我才发现开发一个大型网站原来如此简单。所以说，我们大学生确实不能够只满足于课堂上所教授的一些知识，更应该去探索更多课堂外有用的东西。

而本次课程设计进行的还算顺利，小组成员间的合作显得很默契。有一些bug，我们前前后后改了十几遍，提交了好几个版本，但最终呈现出来的结果让人满意。相信如果只有我一个人的话，那是做不出这个东西的，因此团队合作显得尤为重要。

**王辉**：材料模板的收集。细节规划的建议。后期测试审核。

**总结**：网络数据库结束了，我们组做了公共考试管理系统，虽然做的比较简陋，但是这让我感觉到我们也是可以做出产品出来的。我感觉对以后的学习充满了信心，从对PHP的一无所知，到去学习、使用并最终做出成品出来，这是一个充满挑战的过程。我们也遇到过许多困难，这时我们就需要不断的查找资料，不断的尝试，直到把它解决。在这个过程中，我们学到了很多东西，首先是面对难题的勇气，其次就是团队合作的必要。有时一个小小的问题可能卡住我们一个下午，甚至是晚上加班，但是我们没有放弃，坚持把这个东西给做出来了。

不管是在查找资料还是写代码方面，我们小组都非常团结，但是由于我个人能力的问题，可能给团队的帮助不是很大，但是团队给我的帮助是巨大的，它让我明白了很多道理。每当我想要放弃的时候，队员都会拉我一把，他们会告诉我，不要放弃，还有他们，所以感谢我的队友。然后还要感谢老师，教授了我们网络数据库的知识，让我们在做自己的项目的时候不会茫然。

总之，做出自己的东西是一个艰辛的过程，特别是从我们一无所知开始，但是幸好我有队友，让我在遇到困难的时候不会绝望。