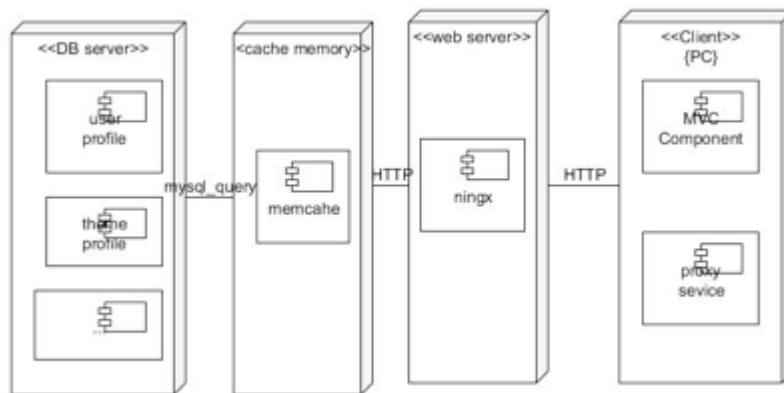


# 软件设计文档

我们的教学效果检车软件主要由教师 PC 端后台管理系统（网站），学生投票系统两部分组成。学生投票系统可以在 PC 端访问，也可以在手机端访问。

为了实现上面的功能，我们使用如下作为我们软件的部署图。



从图中左边开始，DB 服务器使用的是 mysql 数据库。

## 数据库-----mysql

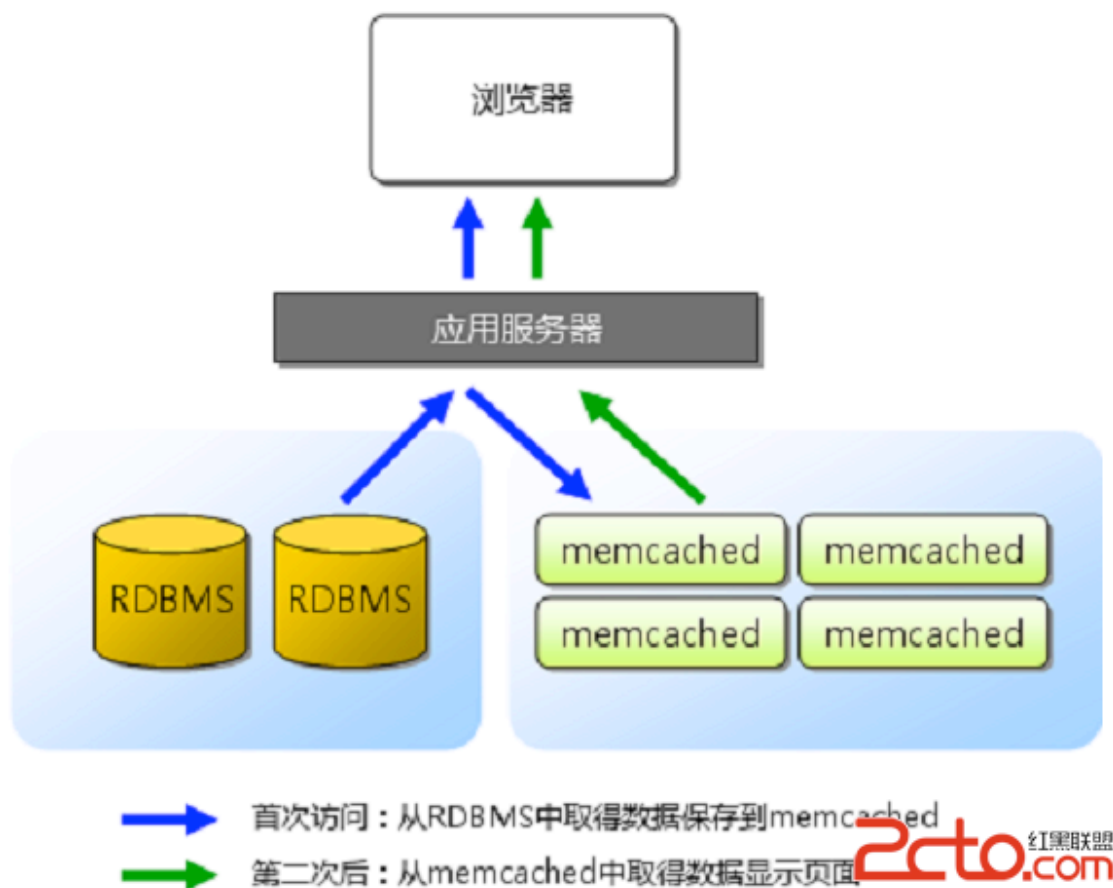
选用理由：

MySQL 是一个小型关系型数据库管理系统,MySQL 被广泛地应用在 Internet 上的中小型网站中。由于其体积小、速度快、总体拥有成本低，尤其开放源码这一特点，许多中小型网站为了降低网站总体拥有成本而选择了 MySQL 作为网站数据库。

1. 它使用的核心线程是完全多线程，支持多处理器。 速度快
2. 有多种列类型：1、2、3、4、和 8 字节长度自有符号 / 无符号整数、FLOAT、DOUBLE、CHAR、VARCHAR、TEXT、BLOB、DATE、TIME、DATETIME、TIMESTAMP、YEAR、和 ENUM 类型。
3. 它通过一个高度优化的类库实现 SQL 函数库并像他们能达到的一样快速，通常在查询初始化后不该有任何内存分配。没有内存漏洞。
4. 全面支持 SQL 的 GROUP BY 和 ORDER BY 子句，支持聚合函数(COUNT()、COUNT(DISTINCT)、AVG()、STD()、SUM()、MAX()和 MIN())。你可以在同一查询中混来自不同数据库的表。

## 缓存数据库-----memcache

MemCache 的工作流程如下：先检查客户端的请求数据是否在 memcached 中，如有，直接把请求数据返回，不再对数据库进行任何操作；如果请求的数据不在 memcached 中，就去查数据库，把从数据库中获取的数据返回给客户端，同时把数据缓存一份到 memcached 中（memcached 客户端不负责，需要程序明确实现）；每次更新数据库的同时更新 memcached 中的数据，保证一致性；当分配给 memcached 内存空间用完之后，会使用 LRU(Least Recently Used，最近最少使用)策略加上到期失效策略，失效数据首先被替换，然后再替换掉最近未使用的数据。



## 服务器-----nginx

选择理由：

首先我们想选择的是 apache，但是后来经过研究，发现 nginx 有更好的特性。下面列举几个观点进行对比。

Nginx (“engine x”) 是一个性能的 HTTP 和 反向代理 服务器，也是一个 IMAP/POP3/SMTP 代理服务器。Nginx 是由 Igor Sysoev 为俄罗斯访问量第二的 Rambler.ru 站点开发的，第一个公开版本 0.1.0 发布于 2004 年 10 月 4 日。其将源代码以类 BSD 许可证的形式发布，因它的稳定性、丰富的功能集、示例配置文件和 低系统资源的消耗而闻名。

### 1、热部署

团队觉得这个很不错。在 master 管理进程与 worker 工作进程的分离设计，使的 Nginx 具有热部署的功能，那么在 7×24 小时不间断服务的前提下，升级 Nginx 的可执行文件。也可以在不 停止服务的情况下修改配置文件，更换日志文件等功能。

### 2、可以高并发连接

这是一个很重要的一个特性！在这一个互联网快速发展，互联网用户数量不断增加，一些大公司、网站都需要面对高并发请求，如果有一个能够在峰值顶住 10 万以上并发请求的 Server，肯定会得到大家的青睐。理论上，Nginx 支持的并发连接上限取决于你的内存，10 万远未封顶。nginx 采用 epoll，大大提高了并发的数量。

### 3、低的内存消耗

在一般的情况下，10000 个非活跃的 HTTP Keep-Alive 连接在 Nginx 中仅消耗 2.5M 的内存，这也是 Nginx 支持高并发连接的基础。

nginx 内存分配将内存需求分成了两种：

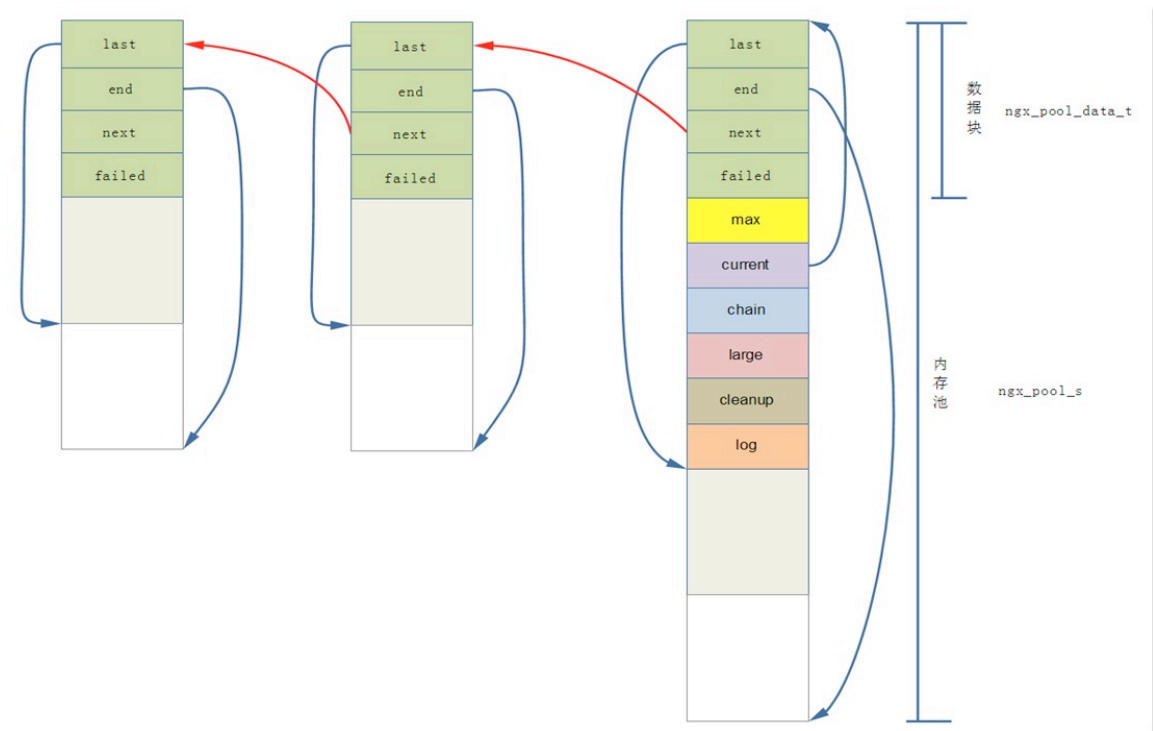
（1）大块内存

（2）小内存。

内存大小的判定依据是申请的内存是否比同页大小与 pool 的 size 两者都小。

对于大块内存，单独利用 malloc 来申请，并且使用单向链表管理起来。

对于小块内存，则从已有的 pool 数据区中划分出一部分出来，这里的内存划分出去后没有特殊的结构来保存，而是等到申请对象生命周期结束后一起释放。小块内存的存储方式非常类似于 sk\_buffer，通过 tail，end 指针来表示多少内存已经被分配出去。、处理响应请求很快在正常的情况下，单次请求会得到更快的响应。在高峰期，Nginx 可以比其他的 Web 服务器更快的响应请求。



## 5、具有很高的可靠性

Nginx 是一个高可靠性的 Web 服务器，这也是我们为什么选择 Nginx 的基本条件，现在很多的网站都在使用 Nginx，足以说明 Nginx 的可靠性。高可靠性来自其核心框架代码的优秀设计、模块设计的简单性；并且这些模块都非常的稳定。

## Nginx Web 服务器优点

Nginx 是一个很牛的高性能 Web 和反向代理服务器，它具有有很多非常优越的特性：

在高连接并发的情况下，Nginx 是 Apache 服务器不错的替代品：Nginx 在美国是做虚拟主机生意的老板们经常选择的软件平台之一。能够支持高达 50,000 个并发连接数的响应，感谢 Nginx 为我们选择了 `epoll` and `kqueue` 作为开发模型。

Nginx 作为负载均衡服务器：Nginx 既可以在内部直接支持 Rails 和 PHP 程序对外进行服务，也可以支持作为 HTTP 代理服务器对外进行服务。Nginx 采用 C 进行编写，不论是系统资源开销还是 CPU 使用效率都比 Perlbal 要好很多。

Nginx Web 服务器是一个安装非常的简单，配置文件非常简洁（还能够支持 perl 语法），Bugs 非常少的服务器：Nginx 启动特别容易，并且几乎可以做到 7\*24 不间断运行，即使运行数个月也不需要重新启动。你还能够在不间断服务的情况下进行软件版本的升级。

## 编程语言-----PHP

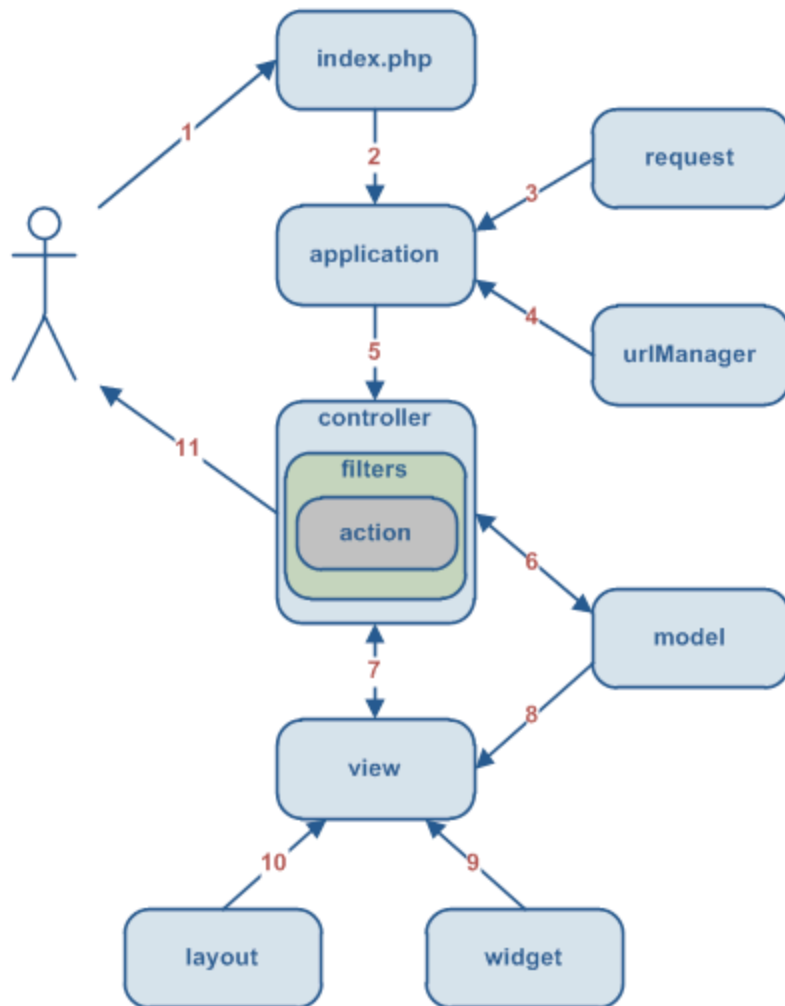
PC 端使用的语言为 PHP，使用的框架是 YII 框架。反向代理也是由 nignx 完成的。

## PHP

1. 跨平台，性能优越，跟 Linux/Unix 结合别跟 Windows 结合性能强 45%，并且和很多免费的平台结合非常省钱，比如 LAMP(Linux /Apache/Mysql/PHP)
2. 语法简单，如果有学习 C 和 Perl 的很容易上手，并且跟 ASP 有部分类似。有成熟的开发工具，比如 NuPHPed，或者 Zend Studio 等等
3. 目前主流技术都支持，比如 Webservice、Ajax、XML 等等。
4. 有比较完整的支持，比如使用 ADODB 或者 PEAR::DB 做数据库抽象层，用 Smarty 或者 smart template 做模板层，如果是 PHP 5.1 的话，还能够使用 PDO(PHP Data Object)来访问数据库。
5. 有很多成熟的框架，比如支持 MVC 的框架：phpMVC，支持类似 ASP.net 的事件驱动的框架：Prado，支持类似 Ruby On Rails 的快速开发的框架：Cake 等等，足够满足你的应用需求。
6. PHP 5 已经有成熟的面向对象体系，能够适应基本的面向对象要求。适合开发大型项目。
7. 有成熟的社区来支持 PHP 的开发。
8. 目前已经很多大型应用都是使用 PHP，比如淘宝网、Yahoo、163、Sina 等等大型门户，很多选用 PHP 来作为他们的开发语言，所以大型门户都能够选用它。

## YII 框架

- 1、Yii 框架优秀的代码风格和完备的开发文档，方便我们团队阅读其 php 代码，上手快。
- 2、Yii 的框架对扩展支持相当高，很方便扩展。
- 3、很重要的性能，Yii 的性能相对优秀，而且属于轻量级框架。spl\_autoload\_register 预注册，按需载入文件等对性能都有很大提高。
- 4、modules 设计是一个很优秀的东西，他理论上可以无限嵌套，当然不建议嵌套超过三层。
- 5、widget 窗口小部件的应用，可以更加方便的构造通用模块。当然我们还有更加方便构造通用模块的方法，后边说。
- 6、其实集成的 model，以及 CActiveRecord 模式 包含的验证机制，都是很好的东西，但是约束性也相对强一些。



## 前端布局框架-----bootstrap

其中前端使用 bootstrap 作为前端布局框架。

作为我们的业务需求是学生用手机浏览器进行访问，所以用 bootstrap 大大地方便了我们，可以让用户在各个平台使用我们的网站，比如安卓手机，iphone，旁边电脑，普通电脑。

并且它有如下几个我们认为比较重要的优点。

应用视觉效果一致性. 这个其实是很很难的，你希望你的链接、按钮、提醒都有统一的视觉效果，你可以为不同级别的提醒使用不同的颜色

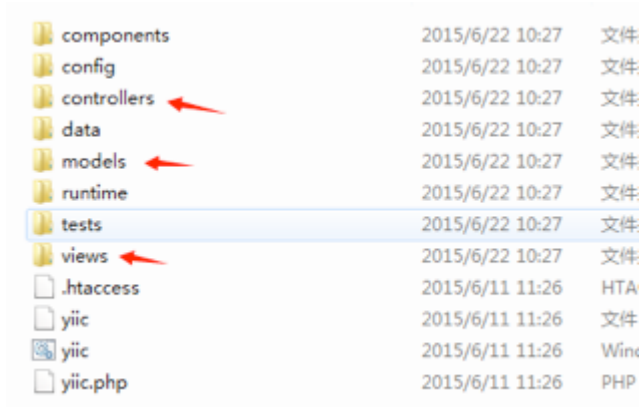
通过多个浏览器测试. 市面上主流的浏览器支持都没问题

完整的框架解决方案. 这是一个框架，你只需使用它而无需重新制作，这个框架专为 Web 应用设计，所有的元素都可以非常完美的在一起工作，可快速开发。

# 面向对象编程-----OOD 在我们项目中的应用

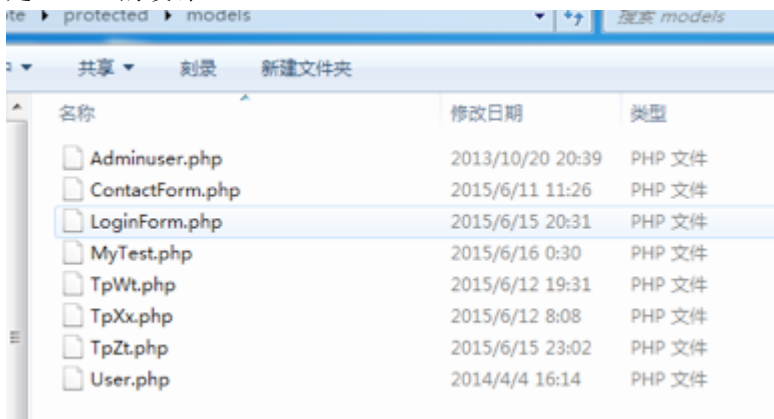
使用的软件设计技术为面向对象设计（Object-Oriented Design，OOD）方法。以服务器后台开发为例。

1，采用模型-视图-控制器（MVC）设计模式：在后台设计中严格执行这种设计模式，从下图可以看出。



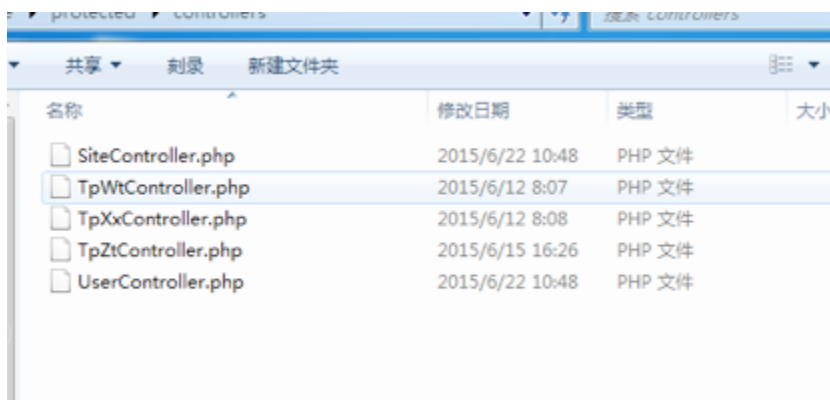
components	2015/6/22 10:27	文件
config	2015/6/22 10:27	文件
controllers	2015/6/22 10:27	文件
data	2015/6/22 10:27	文件
models	2015/6/22 10:27	文件
runtime	2015/6/22 10:27	文件
tests	2015/6/22 10:27	文件
views	2015/6/22 10:27	文件
.htaccess	2015/6/11 11:26	HTA
yiic	2015/6/11 11:26	文件
yiic	2015/6/11 11:26	Win
yiic.php	2015/6/11 11:26	PHP

以下是 model 的设计



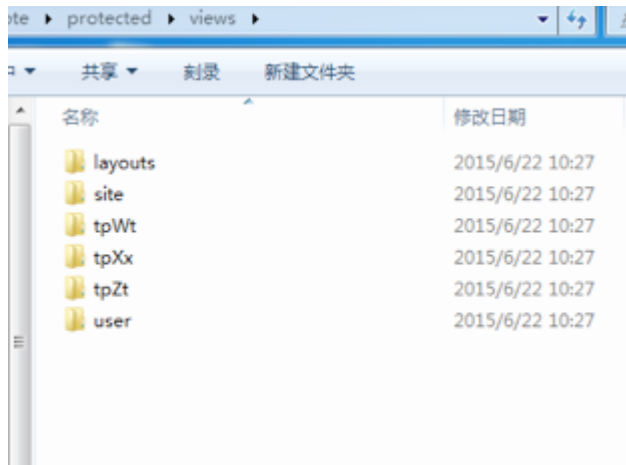
名称	修改日期	类型
Adminuser.php	2013/10/20 20:39	PHP 文件
ContactForm.php	2015/6/11 11:26	PHP 文件
LoginForm.php	2015/6/15 20:31	PHP 文件
MyTest.php	2015/6/16 0:30	PHP 文件
TpWt.php	2015/6/12 19:31	PHP 文件
TpXx.php	2015/6/12 8:08	PHP 文件
TpZt.php	2015/6/15 23:02	PHP 文件
User.php	2014/4/4 16:14	PHP 文件

以下是 controller 的设计



名称	修改日期	类型	大小
SiteController.php	2015/6/22 10:48	PHP 文件	
TpWtController.php	2015/6/12 8:07	PHP 文件	
TpXxController.php	2015/6/12 8:08	PHP 文件	
TpZtController.php	2015/6/15 16:26	PHP 文件	
UserController.php	2015/6/22 10:48	PHP 文件	

以下两张图片是 view 的设计



完全面向对象技术：坚持严格的面向对象编程范式。没有定义任何全局函数或变量。而且，定义类层次结构允许最大的可重用性和定制。

分层缓存方案：支持数据缓存，页面缓存，片段缓存和动态内容。缓存的存储介质，可以轻松更改而不触及应用程序代码。

比如登陆的代码使用缓存技术。其中逻辑是，当用户提交用户名和密码的时候，不直接去访问数据库，而是先到 `memcache` 去查找有没有缓存，如果找到缓存并且用户名和密码符合，那么验证通过。否则进入数据库进行查询，如果成功在数据库中查找到此用户，那么将此用户的用户名和密码保存至缓存。方便下一次快速查询。

可以对照下图代码。

表单输入和验证：Yii 使得收集表单输入非常容易和安全。Yii 拥有一套确保数据的有效性的验证器，它也有辅助方法和部件，显示验证失败时的错误。