<学生考试成绩管理系统>

体系结构设计文档

2016.12.5

**赵亚楠2014211987**

**目录**

[1. 项目相关信息 2](#_Toc26731)

[2. 体系结构需求 3](#_Toc9835)

[2.1 关键指标 3](#_Toc31506)

[2.2 体系结构用例 3](#_Toc16919)

[2.3 各相关方对体系结构的要求 4](#_Toc2892)

[2.4 约束条件 5](#_Toc19794)

[2.4.1实现约束 5](#_Toc1279)

[2.4.2设计约束 5](#_Toc8364)

[2.4.3 其他约束 6](#_Toc4535)

[2.5 非功能性需求 6](#_Toc3540)

[2.5.1可靠性 6](#_Toc6510)

[2.5.2可用性 6](#_Toc28093)

[2.5.3安全性 7](#_Toc26282)

[2.5.4可维护性 7](#_Toc18632)

[2.5.5可移植性 7](#_Toc28783)

[2.6 风险 8](#_Toc12651)

[3. 解决方案 8](#_Toc8403)

[3.1 相关的体系结构模式 8](#_Toc27889)

[3.2 体系结构概述 8](#_Toc25922)

[3.3 结构化视图 9](#_Toc9024)

[3.4 行为视图 10](#_Toc32166)

[3.5 实现问题 10](#_Toc21960)

[4. 系统的质量分析和评价 11](#_Toc27154)

[4.1 场景分析 11](#_Toc7954)

[4.1.1 用例场景分析 11](#_Toc4780)

[4.1.2 增长性场景分析 11](#_Toc1916)

[4.1.3 探索性场景分析 11](#_Toc25568)

[4.2 原型分析 12](#_Toc24874)

[4.3 风险 12](#_Toc10526)

# 

# 1. 项目相关信息

随着教育体制的改革和学术水平的进步，越来越多的学校开始进行学生人数的扩招，这在为各个高校引进人才的同时也带了学生管理的问题。其中一个比较主要的问题就是如何进行学生考试成绩的管理。传统上，在学期期末，教师会对学生的考试成绩分数进行整合，由教务处进行手动进行成绩的统计排序。在教务处完成成绩的整合之后，再以成绩单或者其他的通知方式公布给全体学生。

这种传统方式固然有它的优点，但是在信息容量和效率上面存在着非常大的问题。首先，无论是对于学生还是学校，单单获得到一个成绩信息是不够的，可能还希望获取到一些个人平均成绩、学院平均成绩、某教师某门课程的平均成绩信息等，这些成绩最好可以用表格或者统计图的方式进行显示。其次，传统工作流程效率慢，教务处需要进行整合的数据信息太多，并且学生家长在获取学生成绩信息要依照邮寄成绩单的方式进行成绩信息的获取。

因此，基于上述两种问题，决定开发一个学生考试成绩管理系统解决问题。本系统采取B/S架构方式。前端采取HTML5语言进行用户网页界面的编写，后端使用MySQL数据库管理系统进行数据信息的存储和访问，前端后端借助Web Server进行连接沟通，给使用者带来方便快捷的成绩查询和成绩管理体验。

# 2. 体系结构需求

## 2.1 关键指标

本系统的非功能性能指标可以分为时间性能和空间性能指标两大类。

1. 时间指标：

* 系统的一般操作相应时间应该不超过2秒，系统涉及数据连接的操作相应时间应该不超过5秒。
* 在有300个以内用户同时登录进行成绩查询时，本系统应该不会有超过6秒的信息反馈延迟；当用户数量超过500人时，本系统应该不会有超过8秒的信息反馈延迟。
* 在系统因为访问人数过多而发生宕机时，本系统应该在10分钟之内可以进行恢复。

2. 空间指标：

* 本系统应该存储至少2万人的信息的记录，用于不同身份背景的人进行系统的使用。
* 本系统的磁盘容量应该至少为200GB，其中应该有至少30GB的空闲空间，用于信息的扩充。

## 2.2 体系结构用例

进行成绩管理系统的设计时，将用户的用例模型进行了相应的简化。因为原有的家长角色和导员角色功能过于相似，因此进行了相应的角色合并。教务长和教务员角色功能进行相应的互补，变成了一个教务组角色，具体体系结构用例图如图2-1所示。

学生登录系统可以进行成绩分数和相关成绩图表查看，对存在质疑的成绩提出在线申请，主要的成绩质疑问题为分数和自己预期偏差过大、缺少分数等。

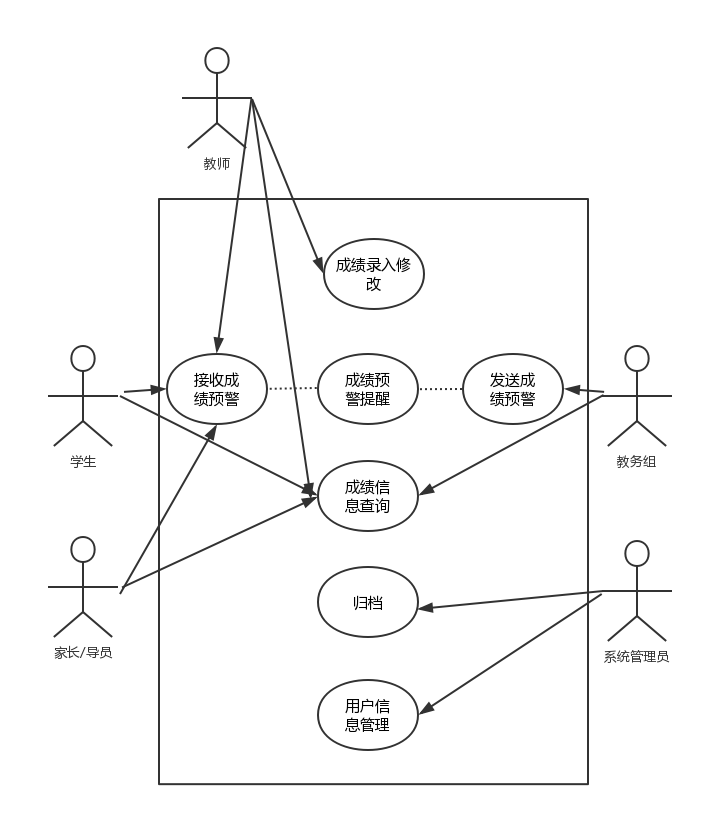
教师可以济宁成绩分数和相关成绩图表查看，在考试结束之后的一段时间会被授权，允许教师进行成绩的修改，教师可能收到教务长发送过来的“挂科警告消息”，该消息可以通知教师所教授课程挂科人数过多，教师会自行采取解决措施。

教务组除了查看成绩之外，可以进行成绩审批和在线发布。还可以进行打印成绩信息，即以成绩单的形式进行成绩的展示以及相应数据展示图表的生成，教务组负责选择什么区段的数据生成什么类型的图形信息，以此来进行成绩信息更进一步的展示。当教务组发现某一学生“挂科”过于频发的时候，会给该学生以及他的负责人发送“挂科警告”，通知他们采取相应的应对措施。

系统管理员进行账户信息管理，增加账户或者删除现有账户。同时可以进行权限管理。

系统管理员维护着账户和相应角色相对应，即用户在登录时便可以知道该用户是属于哪一种角色。管理员进行数据的定期备份，防止因为服务器宕机而发生历史数据的丢失。

导员和家长进行成绩分数和相关成绩图表查看，并且处理“挂科警告”。在每一位同学的成绩不好时，导员会收到“挂科警告”，导员可以和学生或者学生家长进行沟通，防止学生因为成绩过差而留级甚至退学。



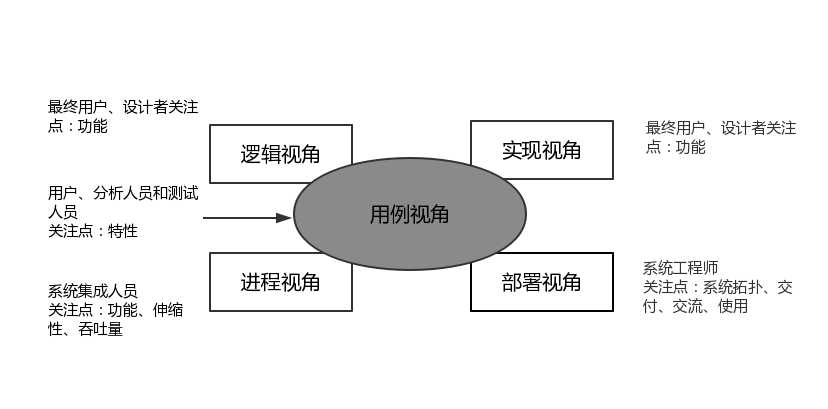
**图2-1 体系结构用例图**

学生考试成绩管理系统一共分为四层：用户界面层、用户身份验证层、信息检索和修改层、数据库层。在图2-2之中对这几层进行了相应的体系描述。

其中用户界面层首先获取用户输入的相关账号密码信息；用户身份验证层获取这些信息，进行相关的身份验证，判断用户属于哪一种角色类型，并为当前用户分配相应的使用权限；当用户的账号密码匹配之后就进入了信息检索和修改层，在这一层用户可以管理个人信息，进行相关成绩信息的查询；用户发起一个查询请求之后进入数据库层，系统在数据库之中检索满足条件的相关信息返还给用户；数据信息最终又以页面的方式展示在用户界面层。

## 2.3 各相关方对体系结构的要求

学生考试成绩管理系统建立了如图2-2的“4+1”视图模型，该模型从4个角度（逻辑、实现、进程和部署）指出不同的相关利益方关心的事情，外加从使用者的角度对用例做观察分析，分析其影响系统的上下文和商业目标情况。



**图2-2 “4+1”视图模型**

最终用户、设计者要求系统可以实现需求分析阶段的基本需求。要求本系统提供给学生、家长、教师、导员、教务处进行使用。通过限制不同角色的使用权限，从而达到可以安全、快速、便捷的成绩录入、成绩查看、成绩管理功能，在节省了人力资源的同时不失安全性和可靠性。

用户、分析人员和测试人员要求系统在最终使用的时候可以可靠完整的运行，数据展示得当，带给用户良好的视觉体验性。

系统集成人员主要着重于系统的性能方面。比如系统时间性能和空间系能上的要求，这些要求保证了系统的未来的可增长性。

系统工程师主要关注于系统的组织结构、交付时间、系统设计时和用户的交流以及在系统完成时对校方负责人进行相应的培训，以便其正确的使用系统。

## 2.4 约束条件

### 2.4.1实现约束

本系统在进行代码编写的时候要求给出详细的代码注释，每一个功能函数都要给出函数的执行功能，输入和返回值，便于后期进行代码的重用。系统要求应用Java语言进行编写，并且所有变量的命名规范符合Java语言命名规范。系统所有网页HTML语言都要符合XHTML 5.0标准。系统要求对数据信息进行加密，要求使用当下市面上已有系统的数据加密算法。

### 2.4.2设计约束

学生成绩管理系统服务于全体在校师生以及学生家长，因此在系统的最终数据呈现上，一定要求带给用户良好的视觉体验性。约束系统的数据展示清晰、图表美观、相应的功能组件设计得体。

这就要求系统最开始的时候从用户的需求出发，设计出相应的应用逻辑可以满足用户的需求，之后根据业务逻辑设计出相应的业务模型开始进行系统的构建，构建的过程之中根据从数据库中获取到的数生成相应视图呈现在网页界面上，呈现给最终用户，设计约束如图2-3所示。

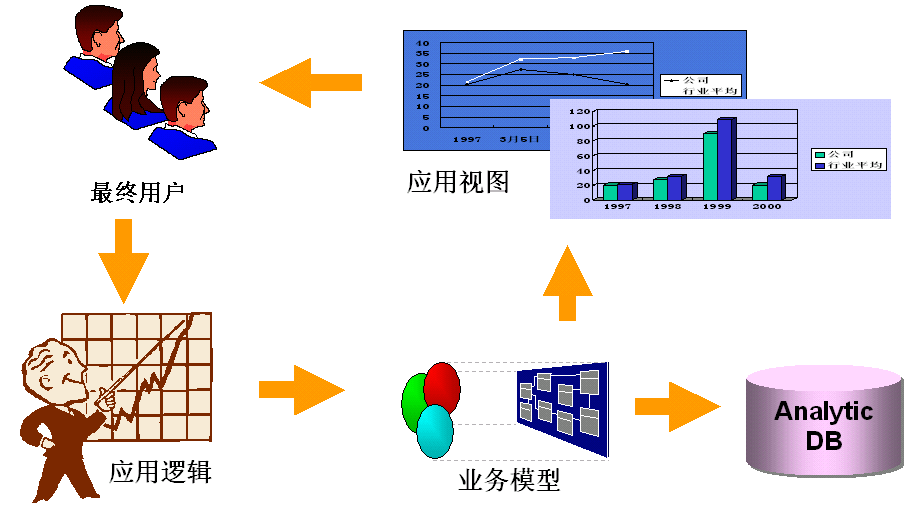


图2-3 设计约束图

### 2.4.3 其他约束

本系统的最后交付日期不能超过2017年7月1日；系统基于B/S架构开发，必须可以在Windows操作系统的IE浏览器上运行，数据库管理系统要求使用MySQL。系统保证在200人同时访问该系统时，不产生超过5秒的延迟。系统界面设计简单美观，让初次使用的人也可以在3分钟之内熟练掌握。

## 2.5 非功能性需求

### 2.5.1可靠性

* 本系统测试时间至少为100小时，测试参与用户至少为200人。本系统应该避免因为故障而导致的失效的能力，在进行测试的过程中，本系统应该尽可能遍历所有的可能故障情况，保证系统的成熟。
* 本系统在发生硬件或者软件异常时，应该具有依然具有服务能力。服务能力虽然下降，但不会达到系统崩溃的程度级别。
* 本系统在发生停机故障之后，应该保证十分钟之内可以修复并且正常运行。

### 2.5.2可用性

* 在接收了用户错误的数据输入之后，本系统能不死机、不崩溃，依然正常工作。本系统应该可以进行错误识别并且给出用户相应的提示信息。
* 系统界面美观，吸引使用者。本系统的界面设计应该在保证基本功能的基础上做到布局美观友好，系统界面和市场面上大部分管理系统界面类似，用户的体验性在本系统的界面设计上因该充分提高。
* 本系统易于学习。本系统应该保证新用户在初次使用时，在不超过3分钟的学习时间之内便可以明白整个系统的基本功能，在10分钟的学习时间之内该用户应该熟练掌握系统的全部功能。
* 用户使用本系统工作效率得到提高。本系统应该加快了学生的成绩查询效率和教师的成绩录入效率，并且本系统应该提高了教务管理人员的成绩数据保存和发布效率。
* 系统提供相应的操作指南文档。本系统提应该提供给使用者相应的文档使用手册，便于使用者获得系统功能的详细解读和使用。

### 2.5.3安全性

* 本系统应该对使用者的权限进行控制。本应同应该使用学生学号、教员教职工号、家长校方注册号进行系统的登录，不同的类型的使用者应该具有不同的系统使用权限。本系统通关系统管理员管理实现对于不同角色开放不同功能的行为，来防止不合法的操作的产生。
* 本系统对于重要数据应该进行数据加密。初次登录系统时，应该输入相应的用户名和密码，对于密码的存储应该采取一定的加密措施，防止因为管理员不适当的信息泄露而产生不可挽回的错误。
* 系统数据定期备份。为了本成绩管理系统的长远考虑，系统数据应该进行相应的备份，防止在系统停止工作时能保证进行数据恢复。

### 2.5.4可维护性

* 在系统运行的过程之中，系统容易诊断出缺陷和失效问题，并且给出行相应的运行时提示。
* 本系统的的代码和文档结构同样应该保证清晰易读，便于在系统主线错误之后进行快速的回复修改，能准确的找出系统错误的原因。
* 在进行系统升级的时候，应该进行版本的控制，减少不必要的功能升级，防止因为升级修改次数越多系统越不稳定现象的发生。
* 本系统耦合度低，便于完成后期系统测试。

### 2.5.5可移植性

* 本系统应该保证在市面上常用的系统环境下便于进行安装使用，不会因为环境的不同而导致系统的崩溃或者时系统功能的缺失。
* 本系统的接口设计合理，便于进行系统的改造和重用，应该方便被当作模板框架来进行相应其他系统的二次开发。
* 本系统在和其他系统在同一个平台下面进行运行的时候，不会产生冲突，各自有条不紊的运行。

## 2.6 风险

1. 系统市场风险。本系统的市场风险主要存在于一下几个方面：系统对在校师生是否发挥了效率上的便利性；使用用户是否会接受并采用本系统提供的相关服务；是否存在其他比本系统更快、更好的学生考试成绩管理系统等。

2. 系统财务风险。本系统的财务风险来源于这几个方面：开发组织是否有能力承担这一系统项目的开发；项目关系人的财务预算是否合适，防止在系统开发的过程中出现预算超支的现象；该系统能否给学校带来相关利益等。

3. 技术风险。本系统的技术风险来源于：该项目在技术上是否可行、成熟；软件、硬件和网络功能是否合适；相应的技术是否满足项目目标；在产品生产出来之前，该技术是否会过时；在系统开发的过程之中过于热衷前沿的先进技术，从而使项目冒太大的技术风险。

4. 系统集成风险。本系统集成需要根据用户的具体信息，系统地设计和实现系统。本系统集成的关键是要将计算机信息领域内不同方面的技术揉合在一起，为用户提供一个出色的整体解决方案, 以减少用户在成绩查询、录入、管理方面的效率损失。

5. 系统管理风险。本系统要求在校方进行引入之后，安排相关人员进行后续管理维护。重点管理维护工作包括：无用数据清理、数据定期备份、用户信息加密管理。

# 解决方案

## 3.1 相关的体系结构模式

根据调查统计，当前市面上大部分相关的系统都是在抽象意义上采用B/S体系结构模式，在具体编程实现时采用MVC体系结构模式。

B/S体系结构的优点就是可以在任何地方进行操作而不用安装任何专门的软件。只要有一台能上网的电脑就能使用，客户端零维护。系统的扩展非常容易，只要能上网，再由系统管理员分配一个用户名和密码，就可以使用了。甚至可以在线申请，通过公司内部的安全认证（如CA证书）后，不需要人的参与，系统可以自动分配给用户一个账号进入系统。

MVC系统体系结构有以下优点：在目前用户需求的快速变化下，可能有多种方式访问应用的要求。减少了代码的复制，即减少了代码的维护量，一旦模型发生改变，也易于维护。 其次，由于模型返回的数据不带任何显示格式，因而这些模型也可直接应用于接口的使用。

再次，由于一个应用被分离为三层，因此有时改变其中的一层就能满足应用的改变。一个应用的业务流程或者业务规则的改变只需改动MVC的模型层。 控制层的概念也很有效，由于它把不同的模型和不同的视图组合在一起完成不同的请求，因此，控制层可以说是包含了用户请求权限的概念。最后，它还有利于软件工程化管理。由于不同的层各司其职，每一层不同的应用具有某些相同的特征，有利于通过工程化、工具化产生管理程序代码。

## 3.2 体系结构概述

本系统结合B/S体系结构模式和MVC体系结构模式设计而成。

B/S架构主要体现在使用者借助浏览器登录软件管理系统，开始进行相关数据的反馈，根据客户相关的业务逻辑，数据将会从数据库中取出同样显示在网页上面。MVC主要用于编写代码时的约束，使得设计思路清晰，便于未来系统的修改和重新使用。

## 3.3 结构化视图

MVC体系架构主要体现在学生考试成绩管理系统一共分为三层：表示层、应用层、服务层。这三层最终都是基于B/S架构的浏览器和服务器实现的。这三层主要用于编写代码时的约束，使得设计思路清晰，便于未来系统的修改和重新使用，表3-1展示了MVC和B/S组织模型，图3-1显示了MVC架构图。

|  |  |  |
| --- | --- | --- |
| MVC层次 | 理解性抽象 | 对应B/S架构 |
| 表示层（V） | 用户界面层 | 浏览器获取数据 |
| 应用层（C） | 用户身份验证层、信息检索和修改层 | 浏览器显示数据，服务器传送数据 |
| 服务层（M） | 数据库层 | 服务器计算数据 |

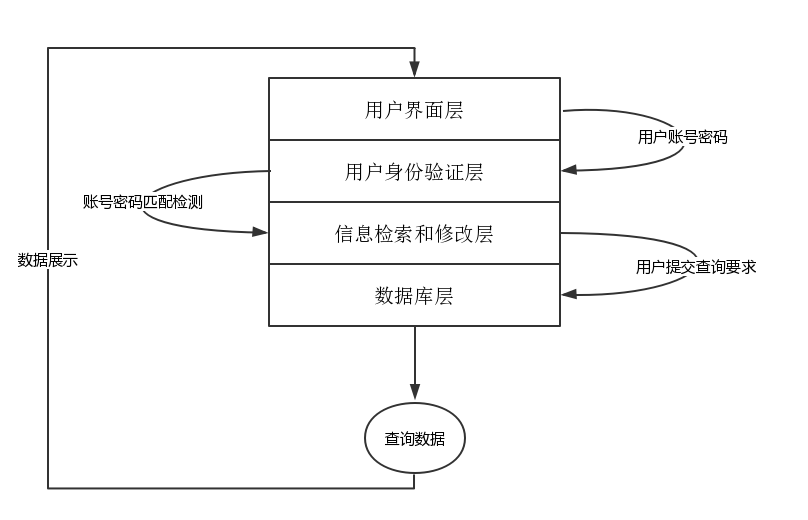
**表3-1 MVC和B/S组织模型**

用户界面层是和用户最直接接触的一个层次。这个层次通过网页界面进行相应的展示，用户可以通过自己的浏览器界面进行相应的访问和查看。

用户身份验证层是进行用户身份类型的检查。本系统面向在校学生、学生家长、教师、教务处等进行成绩管理查询使用，对于不同身份的人系统提供的功能是不同的。在实际中，采用识别出相应的学号工号来完成身份的验证，因此用户身份验证层是不可缺少的。

信息检索和修改层主要针对的是成绩查询和成绩修改功能。在这一层根据上一层获得的身份类型返回不同的查询结果，比如学生只能查询到学生本人的成绩信息；教师可以查看到自己所教授科目的所有学生的考试成绩。

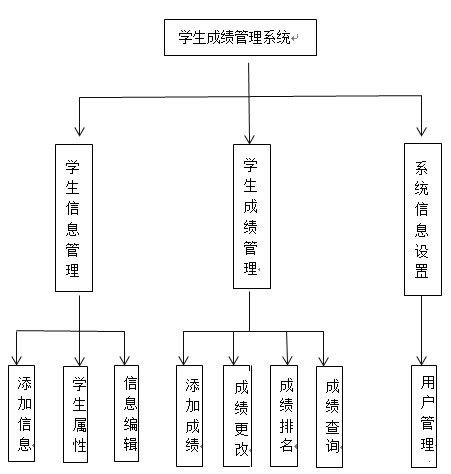
数据库层主要完成数据信息的存储和备份，根据上一层获取的条件返回相应的查询结果，并且对数据进行相应的权限管理，防止非管理人员进行相应的数据访问。



**图3-1 MVC架构图**

## 3.4 行为视图

本系统具有三种主要的行为：学生成绩管理行为，学生信息管理行为，系统信息设置行为。这三种行为之下又可以进行细分出许多其他行为，具体行为视图如图3-2所示。



**图3-2 系统行为图**

## 3.5 实现问题

1. 由于本系统是采用B/S架构实现的，这种情况下的数据库查询信息是一件非常困难的事情，很多的不确定因素会发生在查询的过程中。比如：查询结果在网页的显示上受到编码方式和页面排班布局的制约、查询之中的关键字是无法识别的且无法返回一个正确的结果、数据库在信息反馈上的时间延迟导致页面刷新时间比较长等。本系统不仅仅要求返回相关的成绩数值数据更要求可以返回给用户相关的图形绘制数据，这些数据的不准确必将影响到用户的使用体验程度。基于上述可能存在的问题，反复检查修改逻辑结果，用以来保证网站中的查询功能够基本的达到预期的设计结果。

2. 系统基在实现之后，可能会产生对于不同类型浏览器，系统界面不兼容的问题。这种问题具体体现为：页面排版布局错乱、信息显示丢失、无法显示页面等常见的浏览器不兼容问题。基于这种问题采用限制浏览器的方式解决，给出用户相应的使用要求说明文档，建议用户使用IE7版本以上的浏览器。

3. 密安性和安全性是本系统建设非常重要的一环，要求用户可以查看基于自己身份类型授权获取的相关信息，但是不可查看以及修改他人的资料信息。并且用户的信息不可近被泄露，在系统宕机的情况下可以进行快速的恢复。上述问题通过设置网站管理员入口链接，防范黑客的非法侵犯。同时，考虑到对网站的安全造成一定的安全隐患会发生在管理员的密码被泄露时，在设计系统中，将管理员界面放在标题为ADMIN的目录下，则进入该目录的必须有管理员的权限，相应的网页才能被打开，所以能进入该系统的只能是凭有正确的管理员账户及密码的人，系统的安全性大大的提升。

# 系统的质量分析和评价

## 4.1 场景分析

### 4.1.1 用例场景分析

**场景1**：使用者可以根据自己的身份类型查询到相应的数据信息和系统自动生成的相关数据统计表，而不用局限于时间因素。这个场景代表了用户期望的易用性。

**场景2**：该系统界面美观，布局得体，相应的功能组件直观便于学习。这个场景代表了用户期望的易用性。

**场景3：**使用者只能查看到自己的信息，他人的成绩信息无法被查看。这个场景代表了系统的密安性。

**场景4：**该系统在发生故障之后可以进行相应的数据恢复，使得用户相关信息数据不丢失。这个场景代表了系统的安全性和可靠性。

**场景5：**系统的一般操作相应时间不超过2秒，系统涉及数据连接的操作相应时间不超过5秒。这个场景代表了系统的性能要求。

**场景6：**在有300个以内用户同时登录进行成绩查询时，本系统不会有超过6秒的信息反馈延迟；当用户数量超过500人时，本系统不会有超过8秒的信息反馈延迟。这个场景代表了系统的性能要求。

### 4.1.2 增长性场景分析

**增长性场景1：**通过扩充数据库的规模，把数据检索时间降低到1秒之内。

**增长性场景2：**通过增加一台服务器，把支持同时在线访问的人数扩充到1000人，并且系统相应时间保证在5秒之内。

### 4.1.3 探索性场景分析

**探索性场景1：**系统可以应用于手机平台和平板笔记本电脑平台，支持IOS系统和Android系统。

**探索性场景2：**改进系统的可用性，提升到99%。

**探索性场景3：**添加日历、笔记本、天气小部件功能，方便使用者使用。

**探索性场景4：**在IE5以上的浏览器上都支持系统的运行访问。

## 4.2 原型分析

本系统要求一般操作相应时间应该不超过2秒，系统涉及数据连接的操作相应时间应该不超过5秒；在有300个以内用户同时登录进行成绩查询时，本系统应该不会有超过6秒的信息反馈延迟；当用户数量超过500人时，本系统应该不会有超过8秒的信息反馈延迟；在系统因为访问人数过多而发生宕机时，本系统应该在10分钟之内可以进行恢复。

因此本系统结合B/S体系结构模式和MVC体系结构模式设计而成。B/S架构主要体现在使用者借助浏览器登录软件管理系统，开始进行相关数据的反馈，根据客户相关的业务逻辑，数据将会从数据库中取出同样显示在网页上面。MVC主要用于编写代码时的约束，使得设计思路清晰，便于未来系统的修改和重新使用。

在进行该原型测试时可以采用相应的概念证明本系体结构是否可以满足上述的要求，同时也可以采用技术进行相应的证明，主要技术手段为编写一个程序，模拟多人并发同时访问本系统，记录系统的信息反馈情况和数据库的数据查询返回情况。

## 4.3 风险

本系统测试时间和测试参与用户有限，可能有的故障情况没有遍历到，本系统应该尽可能遍历所有的可能故障情况，保证系统的成熟。

本系统在发生硬件或者软件异常时，虽然具有依然具有服务能力，但是服务能力下降，不会达到系统崩溃的程度级别，但会影响到用户的使用体验性。

本系统在发生并发性故障之后，需要等待10分钟甚至更长时间才能恢复。这一段时间内，所有用户都不能进行系统的登录使用，用户的使用体验性大大下降。