学生考试成绩管理系统

软件需求说明书

版本1

2018/1/1

杨正昱

学号：2015212012

体系结构师

软件工程导论

2017 冬

# 版本历史

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **描述** | **作者** | **内容** |
| 2018.1.1 | 版本1.0 | 杨正昱 | 体系结构文档编写 |
|  |  |  |  |

# 文档批准

以下需求分析报告已经被以下机构人员批准并认可：

|  |  |  |  |
| --- | --- | --- | --- |
| **签字** | **打印姓名** | **标题** | **日期** |
|  |  | . |  |
|  |  |  |  |
|  |  |  |  |

**内容目录**

版本历史 1

文档批准 2

1. 项目相关信息 5

2. 体系结构需求 5

2.1 关键指标 5

2.2 体系结构用例 6

2.2.1 用例图 6

2.2.2 用例分析 7

2.3 各相关方对体系结构的要求 8

2.4约束条件 9

2.4.1 良好的人机交互能力 9

2.4.2 界面需求 9

2.4.3 设计约束 9

2.4.4逻辑数据库要求 10

2.4.5. 外部接口需求 10

2.4.5.1 用户接口 10

2.4.5.2 硬件接口 10

2.4.5.3 软件接口 10

2.4.5.4 通信接口 11

2.5 非功能需求 11

2.5.1 质量要求 11

2.5.1.1性能 11

2.5.1.2 可用性 11

2.5.1.3 安全性 11

2.5.1.4 可维护性 12

2.5.1.5 可移植性 12

2.6 风险 12

3.解决方案 13

3.1 相关的体系结构模式 13

3.1.1 体系结构分级 14

3.2 体系结构概述 16

3.3 结构化视图 19

3.4 行为视图 21

3.5 实现问题 21

4.系统的质量分析和评价 21

4.1 场景分析 21

4.1.2 增长性场景 23

4.1.3 探索性场景 23

4.2原型分析 24

4.3 风险 25

4.3.1 规模风险 25

4.3.2 技术风险 25

# 1. 项目相关信息

本文档是一个学生运动成绩管理系统的体系结构分析报告文档。文档较为全面。系统共与五种角色进行交互：学生、体育老师、体育教务长，系统管理员和档案。同时使用了传统的图示化建模方法，通过系统周境图、数据流图、功能结构图、系统实体-关系图来由浅入深地分析了该系统的结构与功能与用户之间的关系。同时初步提出了系统的功能性、非功能性需求。希望通过本文档能给该系统的开发提供便利。

# 2. 体系结构需求

## 2.1 关键指标

2.1.1 功能需求：

①系统要求任何用户在进入系统之前输入用户名和密码，进行用户验证；

②系统应该能够实现让学生查看各门课成绩的功能；

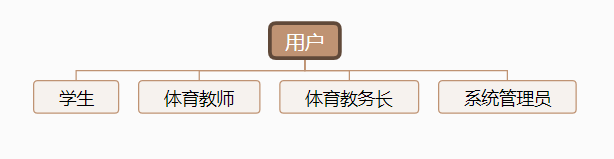
③系统应该能够实现让体育教师能够录入，修改成绩以及制定平时分规则的功能；

④系统应该能够实现让体育教务长能够审查成绩，认定成绩后签名，若成绩有误，授权教师修改成绩的功能；

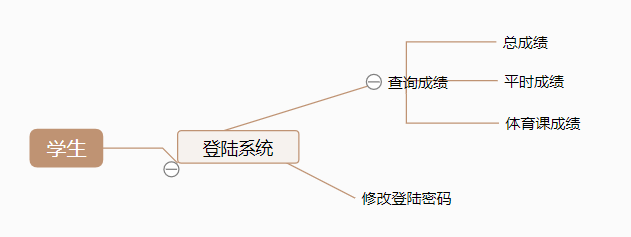
⑤系统应当能够实现让系统管理人员能够管理学生，体育教师，体育教务长的使用权限并对其信息进行增删改查的功能。

## 2.2 体系结构用例

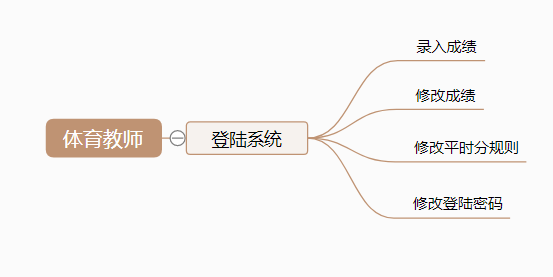
### 2.2.1 用例图



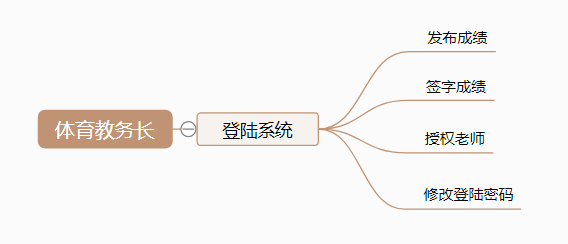
2.2.1.1 用户用例图



2.2.1.2 学生用例图



2.2.1.3 体育教师用例图



2.2.1.4 体育教务长用例图



2.2.1.5 系统管理员用例图

### 用例分析

#### 2.2.2.1学生用例

（1）查询成绩

学生可以查询自己所选课程的成绩。

#### 2.2.2.2 体育教师用例

1. 录入学生成绩

教师可以将自己所教课程的考试成绩为学生录入。

1. 修改学生成绩

若成绩有误，经授权可对数据库中成绩信息进行修改。

1. 制定平时分规则

可以制定平时分的占比比重以及如何计算平时分。

#### 2.2.2.3体育教务长用例

* 1. 签字成绩

在教师将学生期末成绩录入数据库之后，对该成绩信息进行签字，表明认同此成绩的合法性。

* 1. 发布成绩

在签字成绩之后，体育教务长可以选择发布成绩，发布成绩之后同学可以查询该学科的成绩。

* 1. 授权教师修改成绩

若发现某成绩有误，则授权教师对出错信息进行修改。

#### 2.2.2.4 系统管理员用例

修改权限和信息

①学生职位变更或信息变更

更新数据库中学生的信息，同时修改权限。

②体育教师职位变更或信息变更

更新数据库中教师的信息，同时修改权限。

③体育教务长职位变更或信息变更

更新数据库中教务长的信息，同时修改权限。

## 2.3 各相关方对体系结构的要求

1.用户：

1. 学生：查询平时成绩、体育课成绩、总成绩。
2. 教师：录入、修改学生成绩，制定平时分规则。
3. 体育教务长：签字、发布成绩并授权老师修改错误成绩。
4. 系统管理员：修改普通用户的权限和信息。

2.开发技术人员：

系统有明确的开发需求，客户不增添和本系统无关的任意功能，不添加不可能实现的任意功能，不在职责之外添加任务，保证规定时间之内任务能够完成。

3.客户：

项目需要保证在规定的合同约定的时间内能定时定量地完成进度，保证进度不拖拉，同时在财政预算的方面没有多余的预算。

4.项目经理：

保证任务能够及时并合理地进行分解，分发给每一个开发团队独立工作。保证每个团队都能按质按量按时地完成任务。

## 2.4约束条件

### 2.4.1 良好的人机交互能力

由于系统的用户较多，且操作习惯，年龄阶段，接受事物能力都各不相同，所以应要求该系统具备良好的人机交互能力。系统提供的各种功能能够便于用户理解，操作简单。

### 2.4.2 界面需求

系统应该具有可以适用于多平台，多浏览器的界面。例如Linux操作系统、MAC OS操作系统等。

### 2.4.3 设计约束

（1）数据命名1. 变量名、参数名全部由小写字母组成； 2. 函数名首字母为大写字母，其余为小写字母。

（2）变量范围1. 体育教师数目大于等于200；

2. 学生数目大于等于 30000；

3. 课程名称不能大于64字节；4. 所有文本框都应该有合法输入。

（3）其他约束1. 系统的相应时间应当小于0.5秒；2. 系统崩溃后的重启时间应当小于60秒；

3. 采用合适的编程语言保证系统可移植性与可用性。

### 2.4.4逻辑数据库要求

在整个数据库中，包含：开设成绩表，并与学生信息表和体育教师信息表相关联，此外还包括体育教务员长信息表，系统管理员信息表。

1. 学生信息表中，应包括的学生信息有：学生ID、密码、各种成绩、权限。
2. 体育教师信息表中，应包括的学生信息有：学生ID、密码、平时分规则、权限。
3. 体育教务长信息表中，应包括教务长信息有：教务员ID、教务员密码，权限。
4. 管理员信息表中，应包括管理员信息有：管理员ID，管理员密码，权限。

当生成的表后，可以根据所属角色权限设定来保证安全性，也可以通过学生学号进行检索。

# 2.4.5. 外部接口需求

## 2.4.5.1 用户接口

本系统采用B/S架构，所有界面使用WEB风格。

## 2.4.5.2 硬件接口

服务器端建议使用专用服务器。

## 2.4.5.3 软件接口

1. mysql-5.5.20-win32数据库软件；
2. Windows10操作系统/Linux操作系统；
3. myeclipse-9.0M1-win32/Eclipse；

## 2.4.5.4 通信接口

无特殊需求。

## 非功能需求

### 2.5.1 质量要求

### 2.5.1.1性能

1. 时间性能：

（1）本学生体育成绩管理系统，客户端一般响应时间（除数据导入）应不超过0.5秒。

（2）系统崩溃后，恢复时间不应超过60s。

（3）检索文件的反应时间不能超过0.5s。

（4）当多用户同时下载文件时，不会出现服务器宕机的情况。根据该校实际情况，此人数定为30000。

（5）当多用户同时上传文件时，不会出现服务器崩溃的情况。此处人数设定为30000。

1. 空间性能：
2. 数据库容量应能够存储至少5万名学生，500名教师的信息（考虑到扩招的情况）。
3. 数据库容量能够存储至少30000条学生成绩记录
4. 系统有足够大的缓存空间，保障系统运行流畅。

### 2.5.1.2 可用性

1. 方便操作，操作流程合理，尽量从用户角度出发，以方便使用本产品。
2. 支持没有计算机使用经验、计算机使用经验较少及有较多计算机使用经验的用户均能方便地使用本系统。
3. 控制必录入项，本系统能够对必须录入的项目进行控制，使用户能够确保信息录入的完整。同时对必录入项进行有效的统一的提示。
4. 操作完成时有统一规范的提示信息
5. 联机帮助与操作指南。

### 2.5.1.3 安全性

1. 权限控制

根据不同用户角色，设置相应权限，用户的重要操作都做相应的日志记录以备查看，没有权限的用户禁止使用系统。学生只可查看自已的学习成绩，教师只可查看所教课程的学生成绩等等。

1. 重要数据加密

本系统对一些重要的数据按一定的算法进行加密，如用户口令、重 要参数等。

1. 数据备份

允许用户进行数据的备份和恢复，以弥补数据的破坏和丢失。

1. 归档记录

本系统应该能够记录系统运行时所发生的所有错误，包括本机错误和网络错误。这些错误记录便于查找错误的原因。日志同时记录用户的关键性操作信息。

### 2.5.1.4 可维护性

1. 本系统的构建结构十分合理
2. 本系统[可进行测试性](http://baike.baidu.com/view/2194552.htm" \t "_blank)
3. 本系统可进行修改

### 2.5.1.5 可移植性

（1）系统软件接口易改造，因此转置到其它计算机上的能力很强。又由于本系统可维护性强，也方便了系统的移植。

（2）系统分为服务端和客户端，服务端要求支持在 Linux 系统上运行，客户端要求拥有良好的可移植性，至少能够支持 Window 系统和 Mac 系统，最好能够支持 android、iOS 等手机系统。

## 2.6 风险

1.组织

对该项目是否有足够的支持（包括管理人员、测试员、QA 和其他外部的相关各方）。

这是否是该组织尝试过的最大项目。

软件工程是否有明确定义的流程？需求记录和管理。

2.资金

完成项目所需的资金是否到位。

是否为培训和指导分配了资金。

是否有预算限制使得系统必须以固定的成本交付，否则将被取消。

成本估算是否准确

3.人员

是否可以获得足够的项目工作人员。

他们是否具备合适的技能和经验。

他们以前是否在一起工作过。

他们是否相信项目会成功。

是否可以找到用户代表来担任复审员。

是否可以找到领域专家。

4.时间

时间表制定得是否现实。

是否可以为了满足时间表而对功能进行规模管理。

对交付日期的要求有多严格。

是否有时间“把工作做好”。

# 3.解决方案

## 3.1 相关的体系结构模式

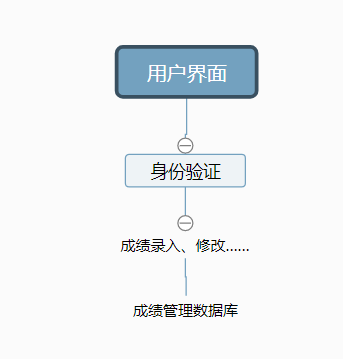
本系统将采用分层体系结构模式，分层系统体系结构有以下几个优点：

1. 支持基于抽象程度递增的系统设计。这允许设计者可以将一个复杂系统设计按递增的步骤进行分解。

2）支持扩充。因为每层至多和与之相邻的上层和下层交互，所以，改变某层的功能 最多只会影响与之相邻的其它两层。

3）支持重用。与抽象数据类型一样，只要对相邻层提供同样的接口，每层可以有很多不同的可相互替代的实现方法。 因此，可能出现对于标准的层接口的定义可以有不同的实现方法。

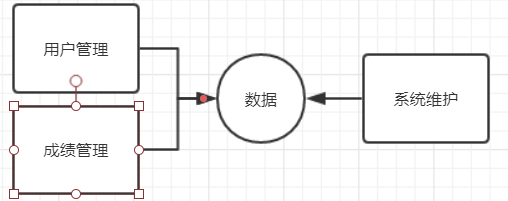
分层图如下所示：（由学生、体育教师、体育教务长、系统管理员界面，身份验证，成绩录入、修改、发布，成绩管理数据库四层组成）



3.1 体系结构模式

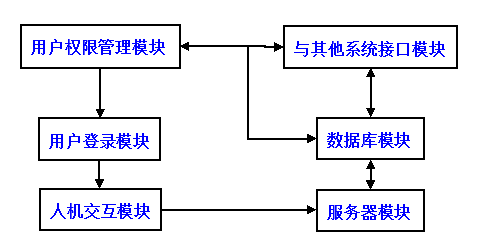
### 3.1.1 体系结构分级

1. 概念级体系结构：描述系统的主要设计元素和元素之间的关系。

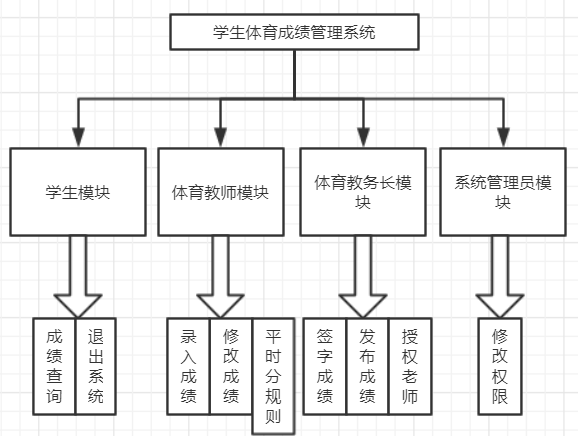


3.1.1.1（1） 概念级体系结构模式

1. 模块级体系结构：

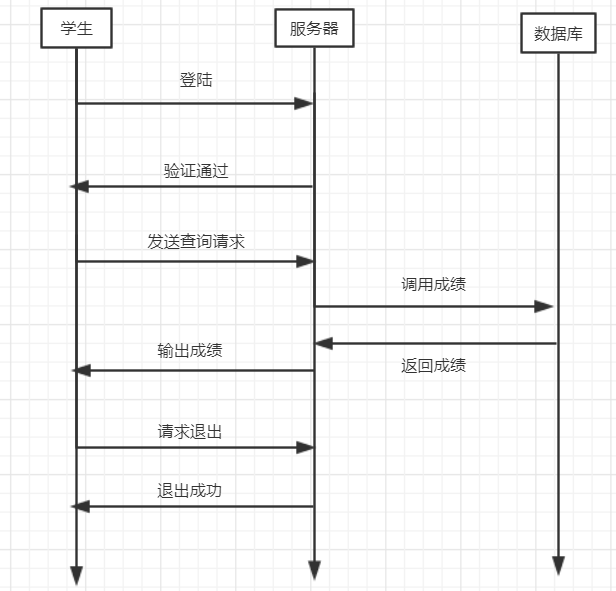


3.1.1.2（1） 模块级体系结构模式



3.1.1.2（2） 模块级体系结构模式

1. 运行级体系结构：



3.3.1.3 运行级体系结构模式

## 3.2 体系结构概述

3.2.1 B/ S 三层体系结构简介

B/ S 体系结构是由C/ S ( Client/ Server ) 结构,即客户端/ 服务器体系结构发展而来的。它是一种以Web 技术为基础的新型的系统平台模式,把传统C/ S 模式中的服务器部分分解为一个数据服务器与一个或多个应用服务器(Web 服务器) ,从而构成一个三层结构的客户服务器体系。Web 客户首先通过HTTP 协议与Web 服务器进行连接,再由Web 客户经浏览器向Web 服务器提交请求,经Web服务器处理后把结果输出至浏览器。

在B/ S 三层体系结构中,应用服务器的功能可由微软IIS ( Internet Information Server ) 和ASP(Active Server Pages) 技术实现。IIS 是一种功能强大的Web 服务器,使用超文本传输协议传输信息。ASP 是一种运行于服务器端的命令执行环境,虽然它只能运行于Windows 平台,但它可以轻松的结合HTML Web 页面、脚本( Script ) 程序和ActiveX组件建立和执行动态、交互式的Web 服务器应用程序。用ASP 开发的Web 应用程序面向对象,无需手动编译或链接程序,而且服务器端程序代码隐藏,在客户端仅可看到由ASP 输出的HTML文件。因此文中设计的系统采用了ASP 作为服务器端的命令执行环境。

3.2.2 体系结构设计

B/S结构有三层的模式,即客户端,中间逻辑层,数据库层。这种结构具有以下的优点:

1.容易维护。瘦客户端,不需要安装单独的客户端应用程序,只需要一个浏览器即可,对客户机所采用的平台无限制,管理及维护工作全部在服务器端进行,增加客户端的应用系统的管理、维护工作量基本不变。另外,统一的、抽象的用户界面使用户更有效地从统一数据源中存取数据,也便于数据库的管理和维护。

2.可扩展性好。系统对中间逻辑层的逻辑事务的增加、修改、删除可灵活地改变,扩展系统的功能容易在运行时自动升级。

3.可重用性好。当多个页面需要进行相同的事务处理时,只需调用相应的事务处理组件即可。

这种结构可更有效地运行。

NET平台四层体系结构描述

对于结构的系统的开发,中间逻辑层的设计,在平台下,可以采用四个逻辑层的体系结构,这四个层分别为WEB层,业务外观层,业务规则层,数据访问层。

1. web层

该层是能够给客户端浏览器提供对于应用程序的访问的一个层。它可以提供对客户的服务,例如:windows用户界面,web用户界面客户端应用程序,或者提供web服务项目。

1. 业务外观层

该层常用于向基础业务对象提供一致的接口,并将客户端同基础业务逻辑的更改隔离开。当它出现时,它或者处于web用户界面和业务规则层之间,或者处于web服务项目和业务规则层之间。

这一层完成的业务逻辑有:

1. 从web层(windows用户界面或web用户界面客户端应用程序)接收用户输入。
2. 如果请求需要对数据层进行访问,则请求使用数据访问层,如果需要对业务进行逻辑规则出来,则将请求传递到业务规则层。同时,它将响应从业务规则层和数据访问层返回到web层。
3. 在对业务规则层的调用之间维护临时状态。
4. 业务规则层

这一层包含业务对象本身以及应用于它们的规则。这一层完成的业务逻辑有:

1. 从业务外观层接受请求。
2. 根据编码的业务规则处理请求。
3. 使用数据访问层,响应数据访问层获取数据。
4. 将处理结果传递回业务外观层。
5. 数据访问层

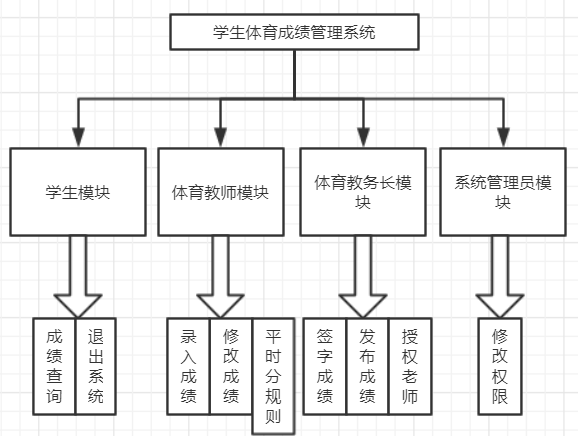
这一层执行从数据库或其他数据服务获取数据或向数据库发送数据的功能。在分布式应用程序结构中,相应功能使用ADO.NET数据适配器和服务器存储过程来完成。这一层完成的业务逻辑有:

1. 从业务规则层或业务外观层接收请求,从数据库或其他数据服务获取数据或向其发送数据。
2. 使用存储过程获取数据,并可选用ADO.NET向数据库发送数据。
3. 将数据库查询结果返回到业务规则层或业务外观层。

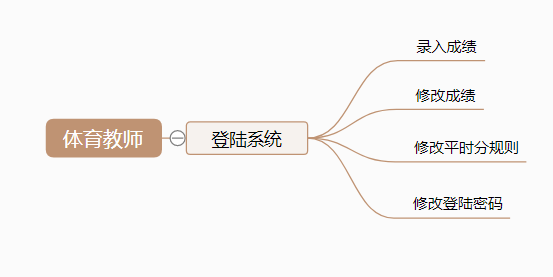
另外,中间逻辑层中还增加了一个通用层,它包含用于在各层间传递信息的对象类的属性。

将中间层分为四个层次的好处在于,通过按照这些原则对web应用程序进行分层,使用基于组件的编程技术,并充分利用.NET平台与Microsoft Windows操作系统的功能,开发人员可以生成具有高度可伸缩性和灵活性的应用程序。开发人员能够生成可在四个层的任何一个层上执行的业务逻辑。利用ASP.NET Web应用程序和XML Web Services, Web层能够以可视形式为应用程序提供丰富、灵活和交互的用户界面。

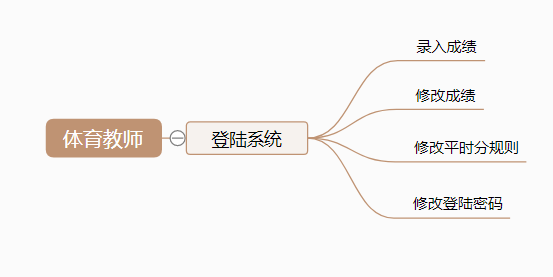
## 3.3 结构化视图



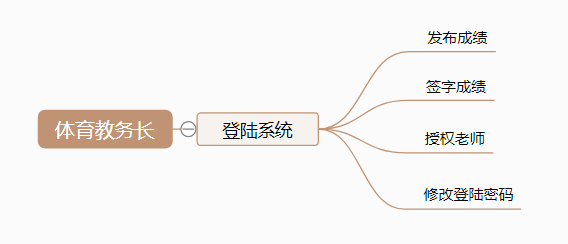
3.3.1 功能结构图



3.3.2 学生的功能结构图



3.3.3 体育教师结构功能图

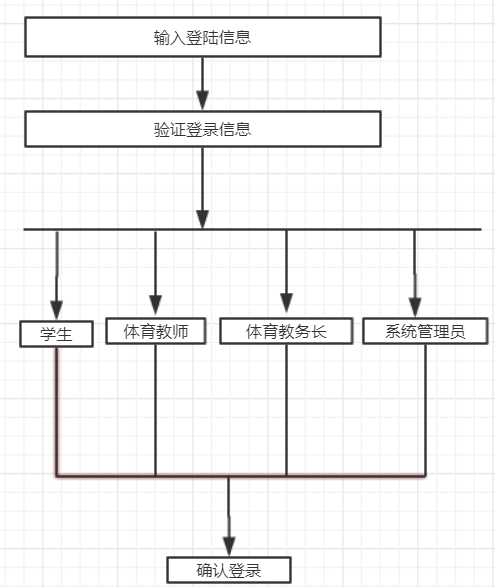


3.3.4 体育教务长结构功能图



3.3.5 系统管理员结构功能图

## 3.4 行为视图



3.4.1 行为视图

## 3.5 实现问题

当学校人数上升时同时使用系统不会出现服务器崩溃现象。当访问量过大时，服务器的负载也过大，则系统有可能进不去，这时可以优化数据库的访问。

# 4.系统的质量分析和评价

## 4.1 场景分析

4.1.1 用例场景

用例场景是从使用着（学生、体育教师、体育教务长、系统管理员）角度出发，描述用户所期望的与整个运行系统的交互。

**场景1**：当发生数据异常时，系统要通知到所有在线用户，并在屏幕上用红色的字体显示出来。该场景代表了用户期望的可靠性。

**场景2**：当用户登录系统时需要进行身份验证，而不是任何人都可以登录。该场景代表了用户所期望的安全性。

**场景3**：系统管理员期望授予一类用户以权限，例如授予所有的学生以查询成绩的权限等。该场景代表了用户期望的易用性。

**场景4**：当用户将图形界面放大时，屏幕要在1秒内重新显示出来。该场景代表了系统性能要求。

4.1.2 增长性场景

增长性场景是指预期未来系统修改时可能发生的场景。

**增长性场景1**：通过扩充现有数据库表的规模，把检索时间降低到平均0.5s以内。

**增长性场景2**：当系统不再局限于web，能用于手机app。

4.1.3 探索性场景

探索性场景是推动系统封装和降低工作压力的场景。

**探索性场景1**：系统能够从Windows平台更换到Linux平台。

**探索性场景2**：改进系统的可使用性，使其从98%提升到99%。

**探索性场景3**：正常情况下，当一半服务器宕机时，不影响整个系统的可使用性。

## 4.2 原型分析

## 风险

风险是指在项目进行过程中可能发生的事件，这些事件将会对项目按预期时间，资源和预算完成产生重大影响。

* + 1. 规模风险

①需求是否相当稳定并得到了充分的了解。

②项目规模是固定不变还是在不断扩展。

③项目开发的时间范围是否太短、不够灵活。

* + 1. 技术风险

①异常处理是否得当。

②构件可能要在若干次发布后才能变得稳定，以致无需重大变更即可复用。

③需求中的事务量是否合理。

④数据量是否合理？当前可用的框架是否能够保存这些数据。

⑤对于与其他系统（包括企业以外的系统）的接口是否存在外部依赖性？是否存在必需的接口或必须创建它们 。

⑥是否存在极不灵活的可用性和安全性需求（例如“系统必须永远不出现故障”）。

⑦系统的用户是否对正在开发的系统类型没有经验。

⑧应用程序的大小或复杂性，或者技术的新颖性是否导致了风险的增加。

⑨是否存在对国家语言支持的需求。