**<学生体育成绩管理系统>**

**体系结构设计文档**

**<1.0>**

**<2017.12.14>**

**薄德芳**

**2015212026**

**2015211502班**

**软件工程导论**

**2017秋**

# 修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **描述** | **作者** | **内容** |
| <2017-12-14> | <版本1> | <薄德芳> | <第一次修订> |

# 文档审批

以下的软件需求规格说明书已被接受并认可：

|  |  |  |  |
| --- | --- | --- | --- |
| **签名** | **打印版姓名** | **职位** | **日期** |
|  | <薄德芳> |  |  |

目录

[修订记录 2](#_Toc9731)

[文档审批 2](#_Toc1365)

[1. 介绍 5](#_Toc29728)

[1.1目的 5](#_Toc32491)

[1.2范围 6](#_Toc8368)

[1.3定义，首字母缩略字及缩写 6](#_Toc24034)

[1.4参考 8](#_Toc17406)

[1.5概述 9](#_Toc26885)

[2. 项目相关信息 9](#_Toc8845)

[2.1产品描述 9](#_Toc12303)

[2.2产品角色和功能 10](#_Toc32103)

[3. 体系结构需求 12](#_Toc1846)

[3.1体系结构定义及关键指标 12](#_Toc23030)

[3.2体系结构用例 15](#_Toc27)

[3.2.1学生用例 16](#_Toc9650)

[3.2.2体育老师用例 17](#_Toc21320)

[3.2.3体育教务长用例 18](#_Toc31780)

[3.2.4系统管理员用例 19](#_Toc1703)

[3.3各相关方对体系结构的要求 20](#_Toc18953)

[3.3.1系统开发、测试、维护人员 20](#_Toc16528)

[3.3.2 用户 21](#_Toc7844)

[3.4约束条件 22](#_Toc29792)

[3.5非功能需求 22](#_Toc17528)

[3.6.1性能表现 22](#_Toc15545)

[3.6.2可靠性 24](#_Toc17671)

[3.6.3可用性 24](#_Toc6365)

[3.6.4密安性 25](#_Toc22341)

[3.6.5可维护性 26](#_Toc11715)

[3.6.6可移植性 26](#_Toc30428)

[3.6设计需遵循的标准 27](#_Toc28874)

[3.6.1实用性原则 27](#_Toc4736)

[3.6.2稳定性原则 27](#_Toc4426)

[3.6.3复用性原则 27](#_Toc29147)

[3.6.4扩展性原则 28](#_Toc24391)

[3.6.5安全性原则  28](#_Toc14298)

[4. 解决方案 29](#_Toc8460)

[4.1相关的体系结构模式 29](#_Toc19555)

[4.1.1 三层B/S体系结构概述 29](#_Toc5771)

[4.1.2 三层B/S体系结构的优点 31](#_Toc4283)

[4.1.3 三层B/S体系结构对质量属性的影响 34](#_Toc8358)

[4.2体系结构概述 35](#_Toc7207)

[4.2.1概念级体系结构 35](#_Toc29645)

[4.2.2模块级体系结构 36](#_Toc27278)

[4.2.3运行级体系结构 37](#_Toc14494)

[4.3结构化视图 38](#_Toc29988)

[4.3.1体系结构视角 38](#_Toc21092)

[4.3.2逻辑视图 39](#_Toc11616)

[5. 系统的质量分析与评价 40](#_Toc678)

[5.1场景分析 40](#_Toc21718)

[5.1.1用例场景 40](#_Toc6701)

[5.1.2增长性场景 41](#_Toc16553)

[5.1.3探索性场景 41](#_Toc11039)

[5.2原型分析 42](#_Toc19883)

[5.3风险 42](#_Toc3373)

# 介绍

在这一部分，主要介绍了本体系结构文档编写的目的、范围、相关概念定义、以及所使用的参考文献。软件体系结构设计属于高层设计文档，是符合现代软件工程要求的概要设计。

## 1.1目的

本文档为体系结构设计文档。旨在从设计的角度对学生体育成绩管理系统的总体结构（逻辑设计、物理结构等）进行综合的描述，并给出了该软件系统设计的总体策略和相应技术解决方案，对软件系统的质量以及开发过程的风险进行了评估。方便了开发过程中不同人员间的交流，提高了软件系统开发的效率。

用户可通过本文档了解软件系统的总体功能和整体架构；开发人员可以通过本文档了解该软件系统预期的功能以及相应的体系架构，并以此为指导对系统进行设计和开发；测试人员可以通过此文档选择与该系统体系结构相适应的测试模型与流程，针对性地进行系统测试；维护人员可根据本文档知道如何对本系统进行维护并在实际应用场景下，进行相关参数的调整；项目管理人员则能根据本文档评估软件系统开发大致的时间流程并制定详尽的开发计划。

## 1.2范围

（1）软件系统的名称：学生体育成绩管理系统

（2）文档内容：本体系结构设计文档分析了学生体育成绩管理系统的功能、系统设计的总体结构以及相应的设计策略。本软件体系结构设计文档能够满足系统的功能和非功能性需求（主要为质量、可信赖性要求），并且分析了在未来的升级和维护中可能遇到的问题以及相应的解决方案。

## 1.3定义，首字母缩略字及缩写

（1）学生体育成绩管理系统：一个具有录入、自动计算、审核、认证、发布、查询成绩以及相关附属功能的信息管理系统。

（2）功能需求：功能需求是对未来软件的服务功能进行的描述。功能需求更关注系统的输入、加工（业务流程）直到输出的情况。

（3）非功能需求：非功能需求是指功能以外的需求描述，主要是系统的一些关键的特性，例如可靠性、响应时间、存储空间、I/O设备的吞吐量、接口的数据格式等。

（4）用户角色：用户角色是指按照一定参考体系划分的用户类型，是能够代表某种用户特征、便于统一描述的众多用户个体的集合。

（5）用例图：用例图是指由参与者（Actor）、用例（Use Case）以及它们之间的关系构成的用于描述系统功能的视图。用例图（User Case）是被称为参与者的外部用户所能观察到的系统功能的模型图，呈现了一些参与者和一些用例，以及它们之间的关系，主要用于对系统、子系统或类的功能行为进行建模。

（6）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。

（7）模块级体系结构：按两两正交模块的结构组织系统，即，功能分解和层次分解。

（8）运行时体系结构：描述系统的动态结构。

（9）代码级体系结构：描述开发环境中的源代码、二进制代码、开发库的组织形式。

（10）ATAM：（Architecture Tradeoff Analysis Method）体系结构折中分析方法。这是SEI开发的通过手工评估和确认体系结构的方法。ATAM将场景分为三种类型：

* 用例场景(use case scenarios)

用例场景描述系统的典型使用，可以启发出实际的应用情况；

* 增长性场景(growth scenarios)

增长性场景涵盖了对系统的预期更改的场景;

* 探索性场景(exploratory scenarios)。

探索性场景可以涵盖极端的修改情况，期望对系统进行“压力”确认。

（10）C/S模式：即Client/Server (客户机/服务器) 结构，是大家熟知的软件系统体系结构，通过将任务合理分配到Client端和Server端，降低了系统的通讯开销，需要安装客户端才可进行管理操作。

客户端和服务器端的程序不同，用户的程序主要在客户端，服务器端主要提供数据管理、数据共享、数据及系统维护和并发控制等，客户端程序主要完成用户的具体的业务。

开发比较容易，操作简便，但应用程序的升级和客户端程序的维护较为困难。

（11）B/S模式：即Browser/Server (浏览器/服务器) 结构，是随着Internet技术的兴起，对C/S结构的一种变化或者改进的结构。在这种结构下，用户界面完全通过WWW浏览器实现。

客户端基本上没有专门的应用程序，应用程序基本上都在服务器端。由于客户端没有程序，应用程序的升级和维护都可以在服务器端完成，升级维护方便。由于客户端使用浏览器，使得用户界面“丰富多彩”，但数据的打印输出等功能受到了限制。为了克服这个缺点，一般把利用浏览器方式实现困难的功能，单独开发成可以发布的控件，在客户端利用程序调用来完成。

## 1.4参考

[1] 王安生，《软件工程化》[M].北京:清华大学出版社，2014

[2] 李莹莹. 对软件体系结构评价指标的探讨[J]. 福建电脑, 2009, 25(10):73-73.

[3] 李书杰, 李志刚. B/S三层体系结构模式[J]. 河北联合大学学报(自然科学版), 2002, 24(s1):25-28.

[4] 徐建, 荆文娟, 严悍,等. 一种软件体系结构风险评估方法[J]. 南京理工大学学报(自然科学版), 2010, 34(5):680-685.

## 1.5概述

本文档将依次介绍：项目相关信息、体系结构需求、解决方案、系统质量分析和评价等。

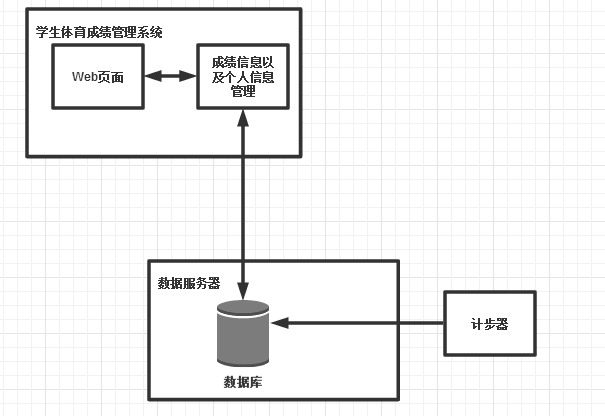
## 项目相关信息

## 2.1产品描述

此学生体育成绩管理系统，实质上是个基于web服务的信息管理系统，目的是提高教职工作人员的工作效率，方便他们对学生体育成绩的录入、修改和管理以及使得血红色呢个可以方便快捷地查询自己的体育成绩。系统的用户分为四类：学生、体育老师、体育教务长、系统管理员。学生可查询自己各个学期的体育平时成绩、项目测试成绩、理论成绩及体育总成绩。体育老师可录入项目测试成绩和理论成绩，可输入平时成绩计算规则，使系统可根据此规则和计步器上传的学生的平时步数计算出平时成绩，从而计算出体育总成绩。体育老师可在授权的情况下修改成绩。体育教务长可以审定和发布成绩，查询所有学生的所有体育成绩，当发现成绩出错时，可授权体育老师修改成绩的权限。系统管理员对学生、体育老师、体育教务长的个人信息和使用权限进行管理。

基本的业务流程为：在进行理论考试和项目测试后，体育老师录入学生的考试成绩，同时输入平时成绩计算规则。计算机将根据此规则和计步器上传的学生的平时步数计算出平时成绩，从而计算出体育总成绩。体育教务长对成绩进行审批后，若成绩无误则发布，否则授权相应的体育老师对成绩进行修改，修改后再次进入审核环节。

系统管理员能管理学生、体育老师、体育教务长的权限和账户信息。



2-1系统块图

## 2.2产品角色和功能

1. 学生

* 对个人的账户信息进行管理。
* 按照学期查询自己的体育理论成绩、项目测试成绩、平时成绩。

1. 体育老师

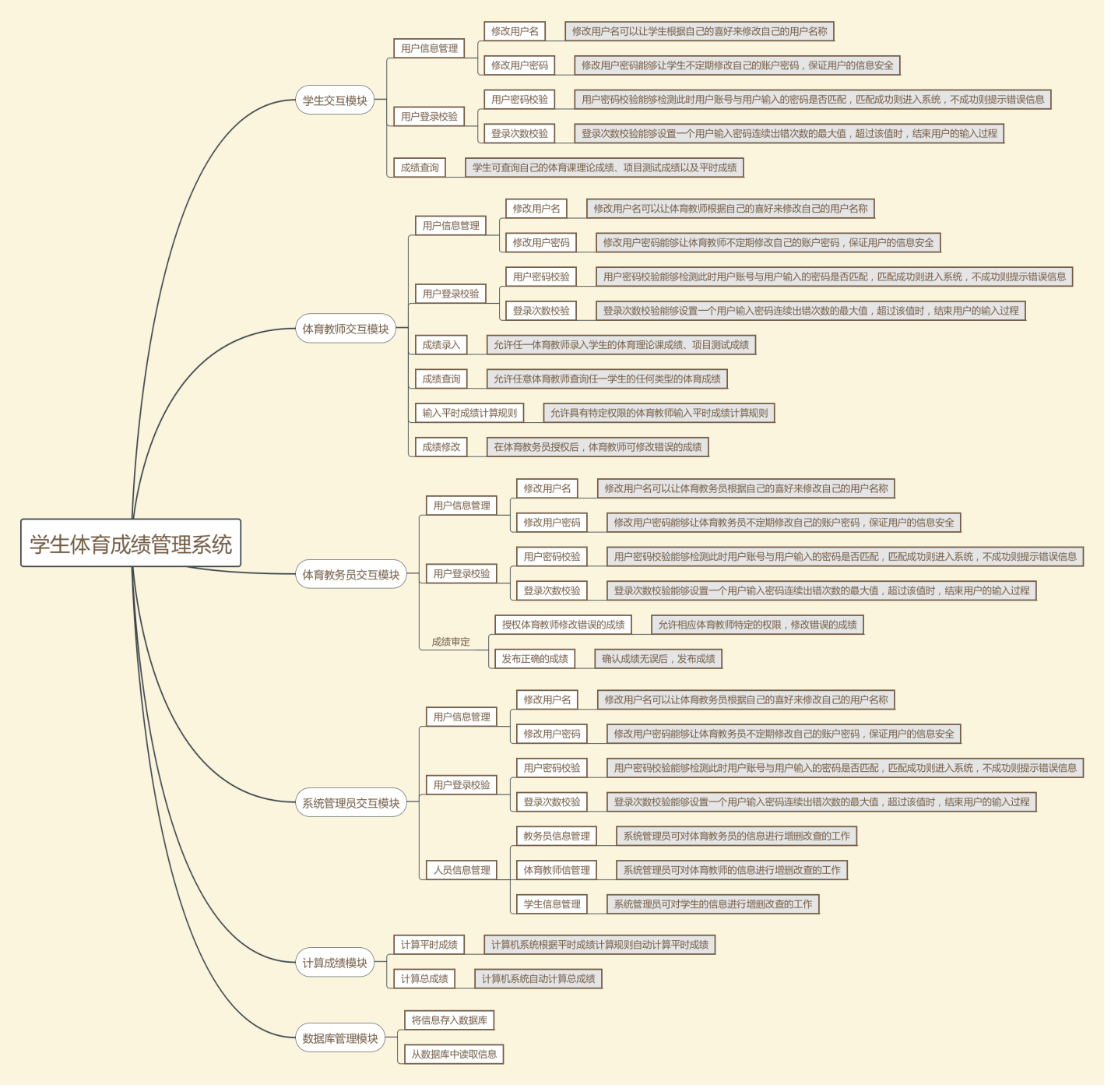
* 对个人的账户信息进行管理。
* 录入所教科目的所有学生的理论成绩和项目测试成绩，之后向教务员提交审核。
* 按规定的格式录入平时成绩的计算规则。
* 可以查看所有自己所教课程的学生的成绩，并且可以按照班级查看所教课程的各项统计数据，包括平均分、最高分、最低分、及格率等。
* 若发现成绩有误，可以向教务长申请权限修改已录入的学生成绩。

1. 体育教务长

* 对个人的账户信息进行管理。
* 可查看成绩信息。体育教务员可查看所有学生的体育成绩，包括班级的平均分、及格率、挂科人数。
* 审定成绩。体育老师录入成绩和平时成绩计算规则且计算机计算完毕后，体育教务员对成绩信息进行审核，如果发现错误，则授权相应的老师修改成绩。（修改成绩的权限仅限于24小时内，之后体育老师若想修改需再次申请）
* 发布成绩。在确认体育老师录入的成绩无误后，发布成绩供学生和体育老师进行查阅。
* 归档记录。相应的成绩信息、用户信息、错误记录进行整理归档。

1. 系统管理员

* 对个人的账户信息进行管理。
* 管理账户信息。系统管理员可以对所有类型的用户（学生、教师、教务长）进行权限的授予撤回以及账户的增删和信息的修改。



2-1系统的结构功能图

# 体系结构需求

## 3.1体系结构定义及关键指标

**1.体系结构的定义：** 软件体系结构是具有一定形式的结构化元素，即构件的集合，包括处理构件、数据构件和连接构件。处理构件负责对数据进行加工，数据构件是被加工的信息，连接构件把体系结构的不同部分组合连接起来。这一定义注重区分处理构件、数据构件和连接构件，这一方法在其他的定义和方法中基本上得到保持。

1. **关键指标说明**

**2.1从软件需求角度**

* **功能性：**体系结构体现系统所期望需求功能的能力。
* **性能：**是指按该体系结构实施后的系统的响应能力。影响性能的体系结构因素包括计算在组件上怎样分配。组件(内部进程和交互进程)间的通信模式等。
* **可靠性：**可靠性是按该体系结构实施下的软件系统在系统发生错误。或在意外或错误使用的情况下．采用体系结构策略维持软件系统功能特性的基本能力。
* **可用性：**是指按该体系结构实施的系统能够正常运行的时间比例。影响该指标的因素包括冗余构件和故障检测构件。
* **安全性：**是指按该体系结构实施的系统在向合法用户提供服务的同时能够阻止非授权用户使用的企图或拒绝服务的能力。
* **扩充性：**系统的体系结构应满足系统生存期间额外的要求．允许增加新的构件或结构以满足系统新的需求的能力。
* **死锁性：**在体系结构的设计中。是否能够避免构件间的交互被”卡住”．不能继续向前的状况。
* **适应性：**体系结构是否有适应用户变化而变化的能力。

**2.2从软件实现角度**

* **可扩展性：**是指该体系结构支持用新特性来扩展软件系统．允许开发人员在不影响客户的情况下替换构件以及支持把新构件集成到现有的体系结构中的能力。
* **结构重组：**是指该体系结构根据开发需要重新组织软件系统的构件及构件问的关系的能力。
* **移植性：**是指该体系结构实施下．软件系统适用于多种硬件平台、用户界面、操作系统、编程语言或编译器的能力。
* **可变性：**是指体系结构经扩充或变更而成为新体系结构的能力。
* **可集成性：**是指按此体系结构构成的系统能与其他部件协作的容易程度。
* **互操作性：**是指该体系结构实施下。系统与其他系统或自身环境相互作用的能力。
* **可测试性：**是指软件体系结构可以为错误探测和改正。以及调试代码和组件的I临时集成给予支持的能力。
* **可理解性：**是指软件体系结构的概念和描述能够全面表达和被深刻理解的程度．反映了该体系结构支持软件设计者之间、设计者与用户之间可以快速方便地交流知识、经验和新设计思想的能力。
* **可重用性：**是指体系结构的框架或映射到体系结构中的构件可以重用体系结构库的结构或构件库的构件的程度。
* **效率性：**是指按该体系结构实施的系统在运行时存储空间、运行时间、吞吐量等对系统的影响程度。
* **可维护性：**按该体系结构实施的系统在错误发生后”修复”软件系统的能力。

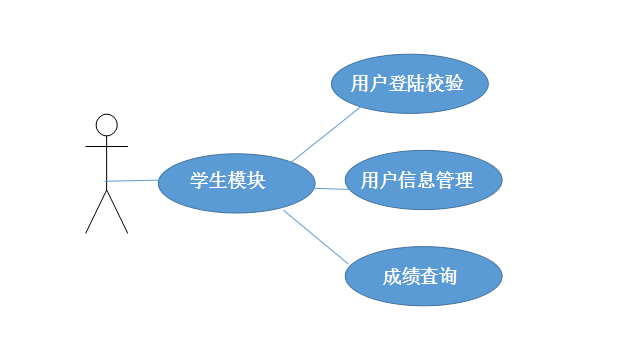
**2.3从软件工程管理的角度**

* **部件无关：**是指体系结构中部件无需了解其它部件的存 在就能独立工作的特性。
* **风险性**：按此体系结构实施的技术和技能的风险度的高低。
* **复用性：**反映该体系结构的抽象性和通用性的程度。
* **可度量性：**按该体系结构实施，系统的开发进度，人力、资金、资源调配可以估计的能力。
* **正交性：**是指反映体系结构中同一层次的构件之间不存在相互调用的状况。

## 3.2体系结构用例

为了更加详细的描述软件的用法，对用户进行分类，定义用户角色。

### 3.2.1学生用例



3-1学生用例图

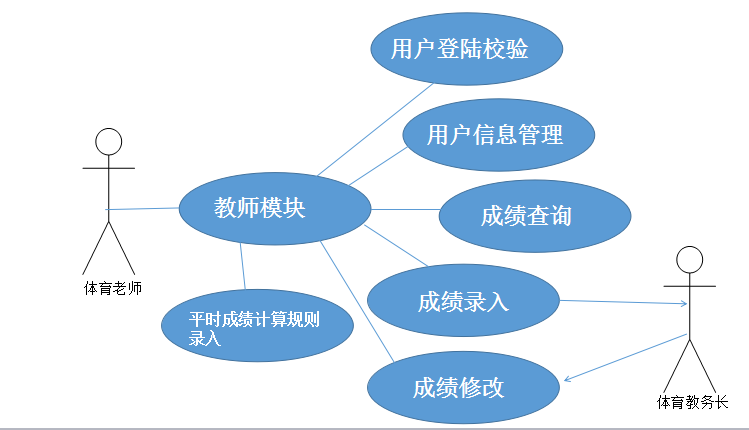
图例说明

|  |  |
| --- | --- |
| 图形 | 含义 |
|  | 参与者 |
|  | 用例 |

设计论述：

1. 学生用户负责的功能是学生模块。
2. 学生模块包含的子功能有：
   1. 用户登录校验。
   2. 用户信息管理。
   3. 成绩查询。

### 3.2.2体育老师用例



3-2体育老师用例图

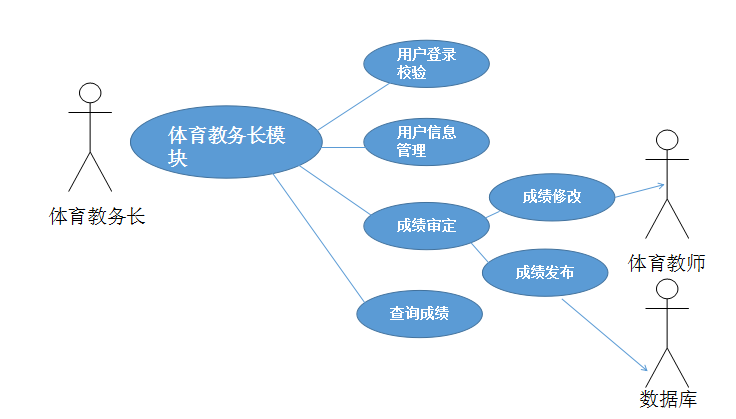
图例说明：

|  |  |
| --- | --- |
| 图形 | 含义 |
|  | 参与者 |
|  | 用例 |

设计论述：

1. 体育老师用户负责的功能是教师模块。
2. 体育老师模块包含的子功能有：
   1. 用户登录校验。
   2. 用户信息管理。
   3. 成绩查询。
   4. 成绩录入。教师将成绩录入后发送给教务员审核。
   5. 成绩修改。如果教务员审核出错误的成绩，将授权教师修改成绩。
   6. 平时成绩计算规则录入。教师将平时成绩计算规则录入，由计算机自动求得平时成绩。

### 3.2.3体育教务长用例



3-3体育教务长用例图

图例说明：

|  |  |
| --- | --- |
| 图形 | 含义 |
|  | 参与者 |
|  | 用例 |

设计论述：

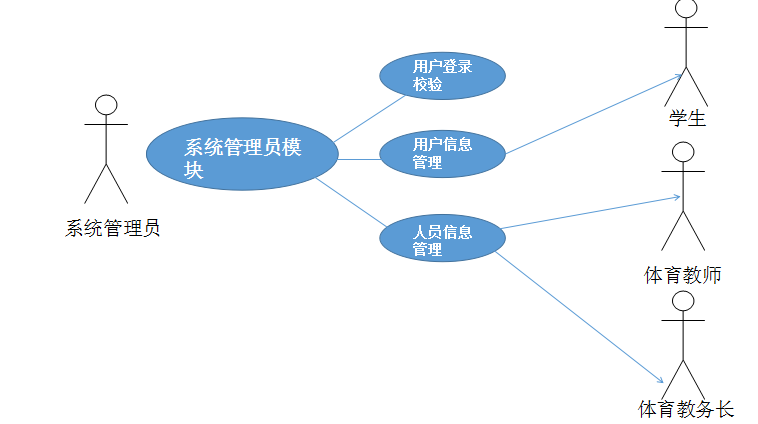
1. 体育教务长用户负责的功能是教务员模块。
2. 教务长模块包含的子功能有：
   1. 用户登录校验。
   2. 用户信息管理。
   3. 成绩审定，包含两个子功能：如果审定无误，成绩发布，需要录入数据库；如果有错误，将授权教师修改成绩。
   4. 成绩查询。
   5. 归档记录。

1.每学期成绩全部发布后，将相应的成绩信息归档。

2.删除用户前，将用户信息归档。

3.当系统发生故障时，将相关错误信息等系统日志进行归档。

### 3.2.4系统管理员用例



3-4 系统管理员用例图

图例说明：

|  |  |
| --- | --- |
| 图形 | 含义 |
|  | 参与者 |
|  | 用例 |

设计论述：

(1)系统管理员用户负责的功能是教务员模块。

（2）系统管理员模块包含的子功能有：

a)用户登录校验。

b)用户信息管理。

c)人员信息管理，更改后的人员信息管理需告知用户。

1. 增加用户。
2. 删除用户。
3. 更改用户信息。
4. 查询用户信息。

## 3.3各相关方对体系结构的要求

### 3.3.1系统开发、测试、维护人员

体系结构的设计应与部件无关、低风险性、高可复用性高、可度量性、高可移植性、高可维护性。

与部件无关：是指体系结构的部件可相互独立地工作，无需了解其他部件的具体实现原理

低风险性：按此体系结构实施的技术和技能的风险度较低。

可复用性：体系结构设计抽象，且具有高通用性

可度量性高：按该体系结构实施，系统的开发进度，人力、资 金、资源调配可以估计的能力。使项目管理人员能更好地进行项目进度的管理和安排

高移植性：统软件的接口易改造，可以比较容易地转移到其他计算机上使用；同时系统使用了跨平台的Java语言来编写并且使用了开源库和复用了一些开源项目的代码，因此可以比较方便地移植到不同平台上。

高可维护性：系统的构建结构合理，采用“高内聚、低耦合”的原则，可以比较方便地进行修改、升级、测试、维护。

### 3.3.2 用户

系统应具有高可用性

高可用性：

* 该具备容易操作的功能。

（1）提供针对不同用户的用户使用说明手册，方便用户学习使用。

（2）系统应提供在线帮助界面，方便用户学习操作。

* 由于操作该系统的人员有很多，且操作习惯、受教育程度、年龄阶段、接受 事物能力等都各不相同，这就要求系统具备良好的人机交互能力。系统提供的各 种功能便于用户理解，操作简单，用户很容易掌握。

（1)系统的界面设计应简洁明了，使用户能够自己学会使用本系统。

(2)系统应具有一定的美观性，可参考目前大部分网站的扁平化设计。

## 3.4约束条件

使用本学生体育成绩管理系统的学校中有 3 万学生，2000 位体育老师。将学生每天、每周、每月的步数作为体育课的平时成绩。平时成绩的计算规则由任课老师给出。本系统只负责处理学生的体育成绩。

## 3.5非功能需求

### 3.6.1性能表现

1. 时间性能：

(1)系统执行速度

系统执行速度需要满足以下条件：系统响应快。数据库单表操作时间不大于 0.5 秒。

(2)系统响应时间考虑到学校的学生人数，系统的响应时间应该达到如下标准比较合理：

* 系统应当可以查看用户的基本信息。其响应速度最佳为 b1 s,平均为 a1 s，不超过 m1 s。
* 系统应当可以查询体育成绩。其响应速度最佳为 b2s,平均为 a2s，不超过 m2 s。
* 系统应当能够更新体育成绩，其响应速度最佳为 b3s,平均为 a3 s，不超过 m3s。
* 系统应当能够计算体育成绩（包括计算平时成绩和总成绩），其响应速度最佳为 b4s,平均为 a4s，不超过 m4s。
* 系统应当能够赋予教师修改成绩权限，其响应速度最佳为 b5s,平均为 a5 s， 不超过 m5s。
* 当多用户同时访问系统时，不会出现服务器宕机的情况。根据该校总共由30000名教职工，此人数定为15000，保证系统在此时的响应速度最佳为 b6s,平均为 a6s，不超过 m6s。

1. 空间性能：
2. 数据库容量应能够存储至少30000名学生，2000名教师的信息。但考虑到可能的扩招，本系统的数据库容量应该能够存储10万名学生和老师的信息。
3. 数据库容量能够存储至少50000条学生成绩记录，但考虑到可能的课程的增加和学生的扩张，本系统的数据库容量设置为能够存储150000条学生成绩记录。
4. 操作占用内存的限制

* 系统应当能够查看四种用户信息，每次操作占用内存不超过 s1。
* 系统应当能够查询学生考试成绩，每次操作占用内存不超过 s2
* 系统应当能够更新学生考试成绩，每次操作占用内存不超过 s3。
* 系统应当能够更新学生考试成绩，每次操作占用内存不超过 s4。

1. 系统有足够大的缓存空间，保障系统运行流畅。

### 3.6.2可靠性

1. 系统的可连续运行时间在一年以上，并且每次发生故障或者宕机后，能够在半小时内恢复正常。
2. 故障恢复能力。每次发生故障或者宕机后，能够在半小时内恢复正常。
3. 容错能力。系统具有一定的容错和抗干扰能力，在非硬件故障或非通讯故障时，系统能够保证正常运行。如用户在系统中输入不规范时，不会引起系统的故障，且系统会给出提示信息，并限定不正确输入的次数。
4. 数据负载能力。选择数据库产品时，要考虑一定的数据负载能力。由于在处理员工信息、客 户信息、账户信息等信息时，系统需要做大量的数据统计和处理，因此要具备相应的数据负载能力。

### 3.6.3可用性

* 该系统的可使用性体现在它可以支持多操作系统多浏览器运行。同时该系统应该具备容易操作的功能。

（1）提供针对不同用户的用户使用说明手册，方便用户学习使用。

（2）系统应提供在线帮助界面，方便用户学习操作。

* 由于操作该系统的人员有很多，且操作习惯、受教育程度、年龄阶段、接受 事物能力等都各不相同，这就要求系统具备良好的人机交互能力。系统提供的各 种功能便于用户理解，操作简单，用户很容易掌握。

（1)系统的界面设计应简洁明了，使用户能够自己学会使用本系统。

(2)系统应具有一定的美观性，可参考目前大部分网站的扁平化设计。

### 3.6.4密安性

1. 权限控制。不同类型的用户有不同的使用权限，同时登陆后所看到的界面也略有不同。如学生之间不可以互相查看或修改其他人的成绩。
2. 数据加密。系统应保证用户信息不泄露，系统配置文件和数据库关键存储文件应当进行加密处理。
3. 数据备份。系统会定期对系统的数据进行备份。以防止可能的数据破坏和用于系统崩溃后的数据恢复。
4. 归档记录与日志记录。

在删除学生、教师用户前由系统管理员归档相关用户的信息（包括成绩信息、课程信息等），以用于可能的查询。除此之外，在系统发生故障后，也会由系统管理员归档相应的系统日志信息，以便于之后研究系统发生故障的根源。

### 3.6.5可维护性

1. 软件的可维护性是指改进软件的难易程度。系统部件遵守“高内聚，低耦合”原则。该系统的结构、接口、功能以及内部过程在开发以及跟踪阶段，容易被维护人员理解。
2. 该系统有良好的测试和诊断系统错误的功能。
3. 当该系统应用于不同学校的学生管理系统之下时，应该具备良好的适应性。不需要通过大幅度的接口与内部过程修改，就能使用户进行使用。

### 3.6.6可移植性

1. 易安装性：该系统能够跨平台移动运行，包括 Windows 服务器平台以及 Linux 平台。
2. 共存性：系统应当能够和其他软件共存于一个平台上，不存在冲突的软件。
3. 可替换性：系统可被容易地卸载，也易被高版本的系统替换。
4. 接口易改造：可方便地移植到不同的设备上。

## 3.6设计需遵循的标准

该软件系统必须遵循以下设计原则：

### 3.6.1实用性原则

在学生体育成绩管理系统的设计过程中，要充分考虑各个高校的实际情况进行业务逻辑的设计，以保证最终开发出来的软件系统切实可用，可以良好地运行。

### 3.6.2稳定性原则

学生体育成绩管理系统能24小时提供不间断的服务，要保证用户在任何时刻都能使用到我们的平台来他们所需的功能。因此，在设计时要考虑到任何可能导致服务中断的原因，为用户提供稳定的服务。

系统设计时应有系统故障的应急预案，当主系统发生故障时，应保证可从主系统平滑地切换到备份系统，尽可能使用户感知不到系统发生的故障，把故障的损失降到最低。

### 3.6.3复用性原则

组件复用是学生体育成绩管理系统设计中必须遵守的原则。对可重用的组件进行统一包装，包括系统级的应用组件和应用级的服务组件。其中，系统级的应用组件主要考虑应用服务器和数据库的广泛适用性，对各种企业级服务进行重新包装，以便于提高架构的平台独立性。以使后续能够为不同的高校开发出适合他们实际情况的学生体育成绩管理系统。

### 3.6.4扩展性原则

学生体育成绩管理系统必须具有良好地可扩展性。随着移动互联网的兴起，越来越多的用户选择在移动终端访问互联网。因此，学生体育成绩管理系统应能在未来扩展到能够适配到移动终端的访问，以满足用户的需求。学生体育成绩管理系统必须能顺应这种业务的急剧扩展而相应地进行开发。

### 3.6.5安全性原则

学生体育成绩管理系统架构的设计将在成熟稳定的硬件环境和应用软件基础上，通过网络、系统、应用、备份恢复、安全控制机制、运行管理监控等手段来保障系统的安全运行。

1.应充分、全面、完整地对系统的安全漏洞和安全威胁进行分析；

2.在网络发生被攻击、破坏事件的情况下，应尽可能地快速恢复网络信息中心的服务，减少损失。应建立安全防护机制、安全检测机制和安全恢复机制。

3.建立合理的实用安全性与用户需求评价与平衡体系。

4.系统是一个庞大的系统工程，其安全体系的设计必须遵循一系列的标准，这样才能确保各个分系统的一致性，使整个系统安全地互联互通、信息共享。

5.技术与管理应相结合原则。应将各种安全技术与运行管理机制、人员思想教育与技术培训安全规章制度建设相结合。

6.统筹规划，分步实施原则。

7.等级性原则。将系统进行分为不同的等级，包括对信息保密程度分级，对用户操作权限分级，对网络安全程度分级，对系统实现结构的分级，从而针对不同级别的安全对象，提供全面安全算法和安全体制。

8.动态发展原则

9.易操作性原则

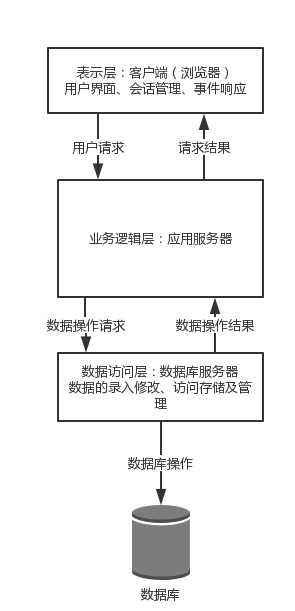
安全措施需要人为去完成，如果措施过于复杂，对人的要求过高，本身就降低了安全性。其次，措施的采用不能影响系统的正常运行。

# 解决方案

## 4.1相关的体系结构模式

### 4.1.1 三层B/S体系结构概述

B/S结构（Browser/Server，浏览器/服务器模式），是WEB兴起后的一种网络结构模式，是从传统的C/S发展起来的计算方式。C/S是松耦合的系统，通过消息传递机制进行对话，由客户端发出请求给服务器，服务器进行相应的处理后经传递机制送回给客户端；B/S模式则把C/S模式的服务器进一步深化，分解成应用服务器和多个数据库服务器，仅保留其表示功能，从而称为一种由表示层（Browser)、功能层（Web Server)、数据库服务层(DataBase Server)构成的三层分布式结构。如下图所示：



4-1三层B/S体系结构体示意图

图中，客户端只有Browser，一般没有应用程序，借助于Java applet、VBScript、JavaScript、ActiveX技术可处理一些简单的客户端处理逻辑，显示用户界面和WebServer端的运行结果。它向URL所制定的Web服务器提出服务申请，Web服务器对用户进行身份认证之后，用TCP/IP协议把所需的文件资料传送给用户，客户端只是接收文件资料，并显示在浏览器上。

服务器端由 W eb server 和DATABAsE server 构成，W eb server 负责接受远程或本 地的数据查询请求，然后运行服务器脚本，借助于CG I 、 ADO、 API 、 JDBC 等中间部件把数 据请求通过发送到DB server 上以获取相关数据，再把结果数据转化成HTML 及各种脚 本传回客户的B rowse 。

DB server 负责管理数据库，处理数据更新及完成查询要求、运行存储过程。可以是 集中式的也可以是分布式的。

由图中可以看出， Browse 与 Web Server 之间的关系就类似于主机／终端结构中两 者之间的关系，而 W eb Server 与DB Server 之间的关系就像Client／Server 结构中两者之间 的关系。 在三层结构中，数据计算与数据处理集中在中间层，即 Web Server 这一层，由于中间层的服务器的性能容易提升，所以在Internet 下的三层结构可以满足用户的需求。但这种结构对数据库服务器提出了高要求。从管理的角度来看，程序代码的维护、数据库的备 份虽然可以在服务器端执行，但这种三层结构对网络带宽是有一定要求的，因为客户端每 次要求获取的数据和反馈信息都要通过网络与服务器联系。

### 4.1.2 三层B/S体系结构的优点

1. **开放的标准**

Client／ Server 所采用的标准只要在内部统一就可，它的应用往往是专用的。Browser／ Server 所采用的TCP／ IP、 HTTP 等标准都是开放的、非专用的，是经过标准化组织所确定的而非单一厂商所制定，保证了其应用的通用性和跨平台性。同时，标准化使得B／ S模式 可直接接入Internet ，具有良好的扩展性、伸缩性，可从不同厂家选择设备和服务。

1. **分布计算的基础结构**

多层的B／ S 应用可以更充分的利用系统资源，在大型的联机应用中，数据库面临的客 户数量是非常庞大的，使用传统的客户／服务器模式可能根本无法胜任。例如，可能有上 千个客户机在同时运行，需要访问数据库。如果它们的请求都直接传递到数据库服务器 上， 就必须要有非常强大的硬件支持。通过中间层的缓冲，连接数据库的用户数大大减少。虽然增加了应用服务层，并不会使系统的性能和可靠性降低。因为在动态分布式计算系统中，客户端程序不必要确切指出应用服务的网络地址，如果应用服务器超负荷，通过统一的管理程序调度将请求转移到其他应用服务器上来消除瓶颈。

1. **较低的开发和维护成本**

Client／ Server 的应用必须开发出专用的客户端软件，无论是安装、配置还是升级都需要在所有的客户机上实施，极大地浪费了人力和物力。B rowser／ Server 的应用只需在客户端装有通用的浏览器即可，维护和升级工作都在服务器端进行，不需对客户端进行任何改 变，故而大大降低了开发和维护的成本。

1. **使用简单，界面友好**

Client／ Server 用户的界面是由客户端软件所决定的，其使用的方法和界面各不相同， 每推广一个Client／ Server 系统都要求用户从头学起，难以使用。B rowser／ Server 用户的界面都统一在浏览器上，浏览器易于使用、界面友好，不须再学习使用其它的软件，一劳永逸 的解决了用户的使用问题。

1. **系统灵活**

Client／ Server 系统的三部分模块中有一部分需改变就要关联到其它模块的变动，使 系统极难升级。B rowser／ Server 系统的三部分模块各自相对独立，其中一部分模块改变时， 其它模块不受影响，应用的增加、删减、更新不影响用户个数和执行环境，系统改进变 得非常容易，且可以用不同厂家的产品来组成性能更佳的系统。

1. **保障系统的安全性**

在C lient／ Server 系统中由于客户机直接与数据库服务器进行连接，用户可以很轻易的改变服务器上的数据，无法保证系统的安全性。B rowser／ Server 系统在客户机与数据库 服务器之间增加了一层 W eb 服务器，使两者不再直接相连，通过对中间层的用户编程可实现更加健全、灵活的安全机制。客户机无法直接对数据库操纵，有效地防止用户的非法 入侵。

1. **信息共享度高**

Client／ Server 系统使用专用的客户端软件，其数据格式为专用格式文件。Browser／ Server 系统使用HTML， HTML 是数据格式的一个开放标准，目前大多数流行的软件均 支持HTML，同时 MIME技术使得Browser可访问多种格式的文件。

1. **广域网支持**

Client／ Server 系统是基于局域网的，而B rowser／ Server 系统无论是PSTN、 DDN、帧中 继， X.25 、 ISDN，还是新出现的CATV、 ADSL， B／ S 结构均能透明的使用。

### 4.1.3 三层B/S体系结构对质量属性的影响

|  |  |
| --- | --- |
| **质量属性** | **对质量的影响** |
| 性能 | 这种结构的性能已经得到了证明。关键的问题是要考虑每个服务器支持的并发线程数量、各层之间的连接速度以及数据传递的速度。对于分布式系统来说，降低了为完成每个需求所需的层与层之间的调用时间。 |
| 可靠性 | 通过中间层的缓冲，连接数据库的用户数大大减少。虽然增加了应用服务层，并不会使系统的性能和可靠性降低。因为在动态分布式计算系统中，客户端程序不必要确切指出应用服务的网络地址，如果应用服务器超负荷，通过统一的管理程序调度将请求转移到其他应用服务器上来消除瓶颈。 |
| 可用性 | B rowser／ Server 用户的界面都统一在浏览器上，浏览器易于使用、界面友好，不须再学习使用其它的软件，一劳永逸 的解决了用户的使用问题。 |
| 密安性 | B rowser／ Server 系统在客户机与数据库 服务器之间增加了一层 W eb 服务器，使两者不再直接相连，通过对中间层的用户编程可实现更加健全、灵活的安全机制。客户机无法直接对数据库操纵，有效地防止用户的非法 入侵。 |
| 可维护性 | B rowser／ Server 系统的三部分模块各自相对独立，其中一部分模块改变时， 其它模块不受影响，应用的增加、删减、更新不影响用户个数和执行环境，系统改进变 得非常容易，且可以用不同厂家的产品来组成性能更佳的系统。 |
| 可移植性 | Browser／ Server 系统使用HTML， HTML 是数据格式的一个开放标准，目前大多数流行的软件均 支持HTML，同时 MIME技术使得Browser可访问多种格式的文件。 |
| 可伸缩性 | Browser／ Server 所采用的TCP／ IP、 HTTP 等标准都是开放的、非专用的，是经过标准化组织所确定的而非单一厂商所制定，保证了其应用的通用性和跨平台性。同时，标准化使得B／ S模式 可直接接入Internet ，具有良好的扩展性、伸缩性，可从不同厂家选择设备和服务。  多个服务器可以有备份，多个服务运行在同一个或多个不同的服务器上，使体系结构的规模可以得到很好的提升。在实际中，数据管理层往往成为系统能力的瓶颈。 |

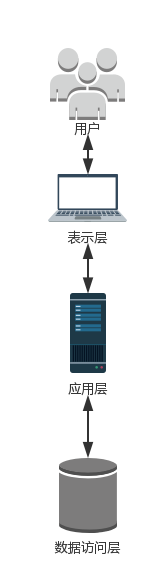
## 4.2体系结构概述

通过4.1部分的分析，可以知道三层B/S体系结构具有许多传统C/S体系结构不具备的优点。因此该体育成绩管理系统选择基于三层B/S结构设计的开发。

表现层在最外面，展现管理系统的界面，用户可以通过表现层直接与系统交互，进行一系列的操作，包括查询成绩、录入成绩等等；中间是业务逻辑层，联系前端与后端；最内部是数据库访问层，用来进行数据库的相关操作。

### 4.2.1概念级体系结构

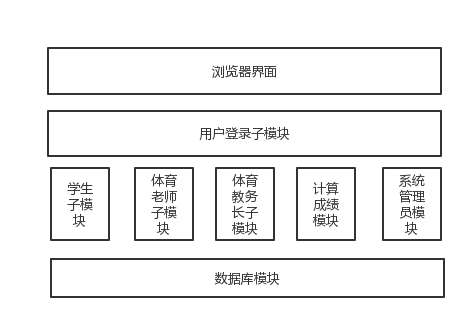
在概念级体系结构中，表现层负责向用户展示系统的页面及功能，业务逻辑层负责处理用户的请求并返回相应的数据，持久化层负责将用户的信息和数据保存在数据库中，并做必要的数据备份和恢复。



4-2概念级体系结构示意图

### 4.2.2模块级体系结构

系统的模块化设计是系统进行复用的关键。模块级体系结构反应了对软件代码实现时的期望。特别是对于程序规模较大的系统。下图是按层次方法划分的软件系统结构。体系结构的层次表达了每层的上一层提供的功能和接口，以及需要使用的下一层的功能。

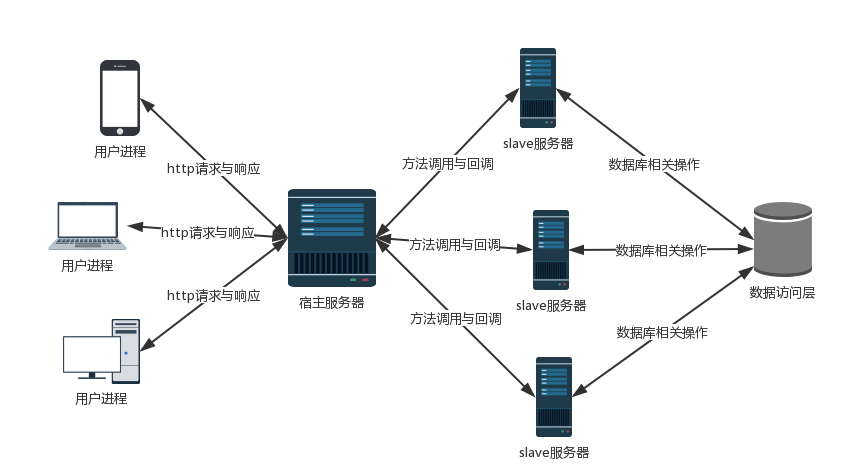


4-3 模块级体系结构示意图

### 4.2.3运行级体系结构

运行级的体系结构用来描述系统的动态结构，描述上要尽可能用运行元素(例如，操作系统的任务、进程、地址空间等)、通信机制、运行元素分配的功能、以及资源分配等。同时，还要考虑到系统运行元素的位置分布、可移植性等因素。

* 每个用户都是一个独立的进程，可以独立地访问服务器发送请求并查看结果。
* 服务器为分布式架构，一台服务器作为Master服务器，管理Slave服务器的资源配置与负载均衡。
* 所有的Slave均访问同一个数据库，便于维护数据的一致性，同时定期对服务器进行备份，防止可能发送的故障和奔溃。



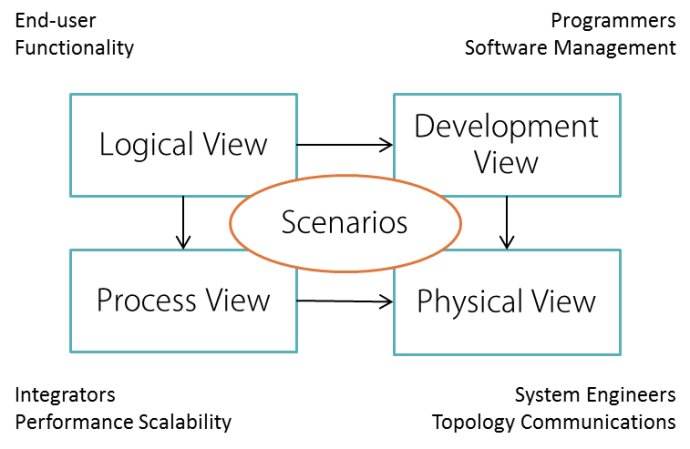
4-4 运行级体系结构示意图

## 4.3结构化视图

### 4.3.1体系结构视角

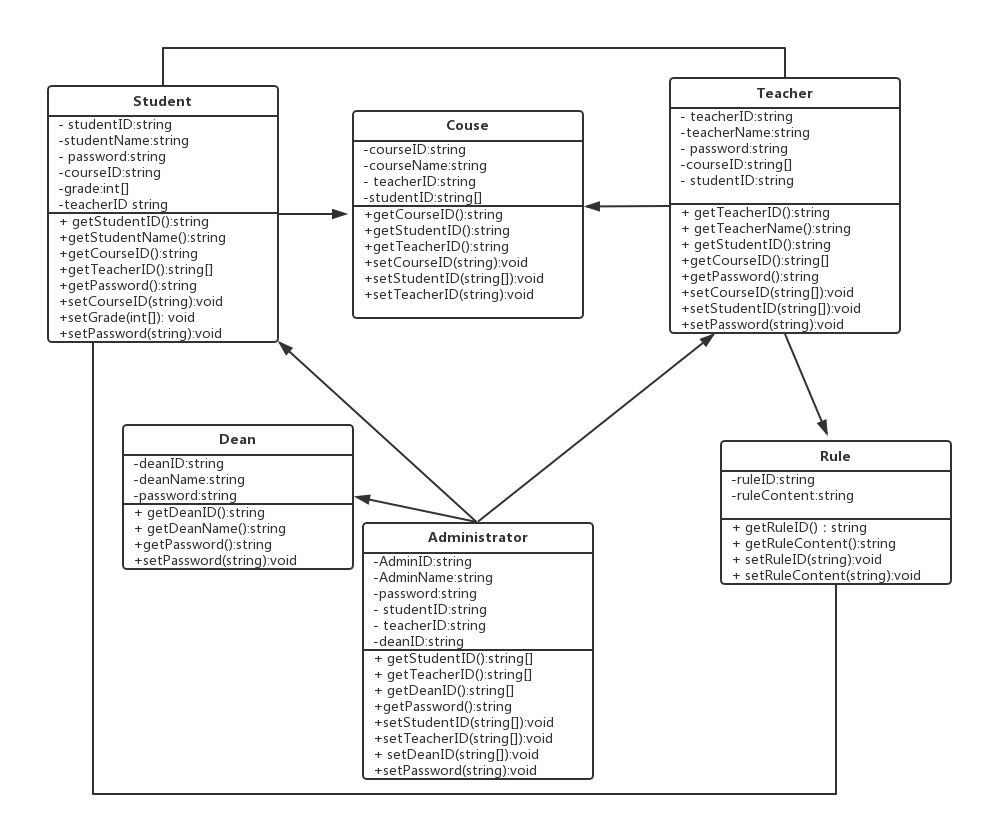
Philippe Kruchten提出从不同的角度勾画系统的蓝图，建立“4+1”视图模型。该模型从4个角度（逻辑、实现、进程和部署）指出不同的相关利益方关心的事情，外加从使用者的角度对用例观察，分析其影响系统的上下文和商业目标情况。

IMG_256

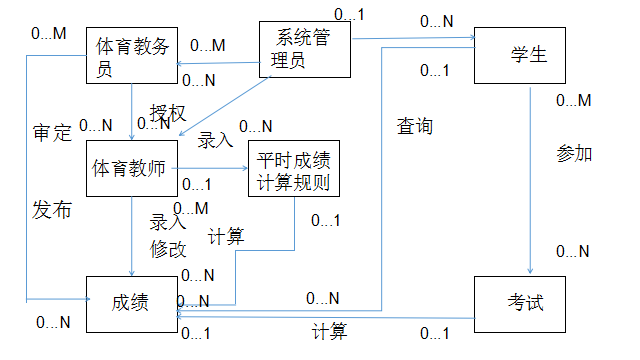
IMG_256

4-5“4+1”视图模型

### 4.3.2逻辑视图



4-6逻辑视图



4-6实体-关系图

# 系统的质量分析与评价

## 5.1场景分析

根据ATAM方法来对学生成绩管理系统的体系架构进行分析。

### 5.1.1用例场景

1. 系统应该能够支持30000名用户的并发操作。此场景代表了用户所期望的系统性能和效率。
2. 用户登录系统的响应时间不超过2秒，用户对数据的查询、增加、删除和更新的系统响应时间不超过1秒。此场景代表了用户所期望的系统性能和效率。
3. 计算机系统应在用户输入平时成绩计算规则后，在0.2秒内完成对平时成绩的计算，并将成绩存入缓存。此场景代表了用户所期望的系统性能和效率。
4. 当服务器故障发生时，系统能够在1分钟内切换到备用服务器上继续工作，该场景代表了用户期望的可靠性。
5. 当数据库访问异常时或者用户进行错误操作时，均应有相应的错误提示，该场景代表了用户期望的可靠性。
6. 允许15000个用户同时访问系统，系统能稳定地提供服务，3.此场景代表了用户所期望的系统性能和效率。
7. 用户可方便快捷地使用系统，系统界面简洁美观，具有很好的人机交互性。此场景代表了用户所期望的系统易用性。

### 5.1.2增长性场景

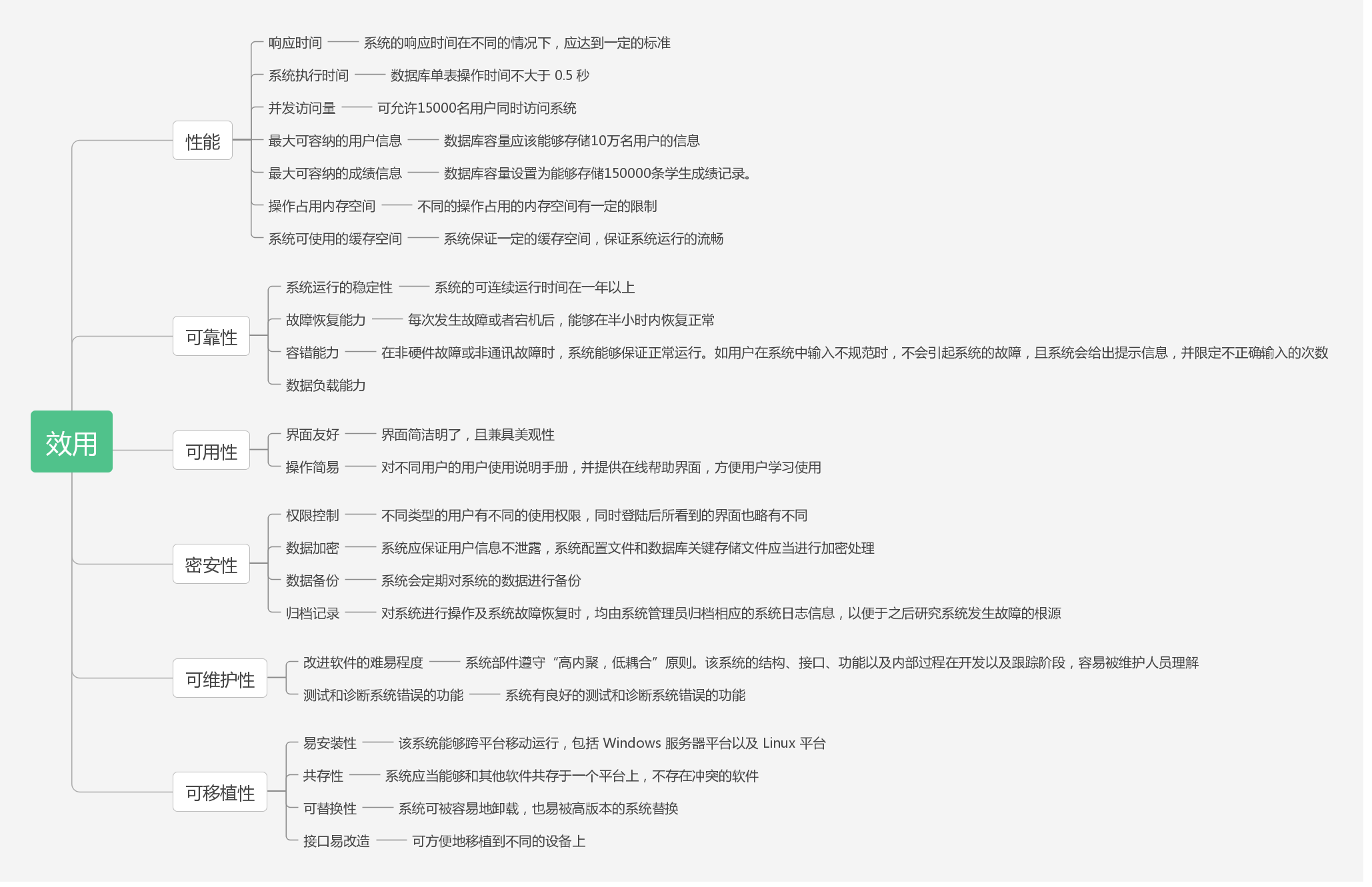
1. 可增加新的数据库服务器、扩充现有的数据规模，降低远程用户的访问时间；
2. 通过对软件系统适配手机界面，使得用户可以在移动设备终端上访问学生成绩管理系统；
3. 考虑使用分布式集群进行计算，均衡单个服务器的负载量，从而降低用户使用该软件系统的响应时间。

### 5.1.3探索性场景

1. 系统可以从windows平台切换到Android平台；
2. 增加系统的可使用性，使其从98%提升到99%;
3. 在正常情况下，当一半的服务器宕机时，可启动备用的服务器以致使用户感知不到服务器故障的发生，提升系统的可靠性；
4. 优化算法，改进系统的性能，降低系统的响应时间；
5. 提到系统对高并发访问的处理能力。

## 5.2原型分析

使用效用树法进行原型分析，分析结果如下图所示



5-1 效用树分析图

## 5.3风险

1. 开发费用预估不足，导致项目资金短缺，减缓了工程进度；
2. 需求分析不足，导致设计的体系结构并不满足项目的实际需求，在后期进行软件项目质量审核和评估时，进行大规模的返工。
3. 系统的重要构件、模块不符合设计的要求，导致系统功能的缺陷和项目的失败。
4. 系统的安全性设计存在漏洞，系统易受攻击且难以维护。