学生体育成绩管理系统

体系结构文档

1.0

2017/12/25

邓琪

软件工程导论

2017 秋季

修订历史

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **描述** | **作者** | **备注** |
| 2017/12/25 | 版本1.0 | 邓琪 | 介绍与总体概述 |

文档批准

以下需求分析文档已经被以下机构人员批准并认可:

|  |  |  |  |
| --- | --- | --- | --- |
| **签名** | **打印签名** | **标题** | **日期** |
|  | 邓琪 | 学生体育成绩管理系统 | 2017/12/25 |
|  |  |  |  |
|  |  |  |  |

**目录**

1. 项目相关信息 1

1.1概述 1

1.2 范围 1

1.3主要功能 1

1.4 术语与缩写定义 1

1.5 引用 2

2. 体系结构需求 2

2.1 关键指标 2

2.2 用户特征 2

2.3体系结构用例 2

2.3.2用户角色1-学生 3

2.3.3用户角色2-体育老师 4

2.3.4用户角色3-体育教务长 4

2.3.5用户角色4-系统管理员 4

2.3.6用户角色5-档案 5

2.4 各相关方对体系结构的要求 5

2.6 非功能性需求 6

2.6.1质量需求 6

(1) 性能 6

(4)密安性 7

(6)可移植性 7

2.6.2 工程需求 7

(1) 逆向需求 7

(3) 逻辑数据库需求 7

3. 解决方案 8

3.1 相关的体系结构模式 8

3.1.1 共享数据仓库模式 8

3.1.2 分层的C/S模式 9

3.2 体系结构概述 9

3.3 结构化视图 10

3.3.1概念级体系结构视图 10

3.3.2模块级体系结构视图 10

3.3.3运行级体系结构 11

3.4 行为视图 11

4. 系统的质量分析与评价 12

4.1 场景分析 12

4.1.1用例场景 12

4.1.2增长性场景 12

4.1.3探索性场景 12

4.2 原型分析 13

4.3 风险 13

# 项目相关信息

## 1.1概述

学生体育成绩管理系统是一个web端软件系统，为了督促学生平常加强锻炼，提高身体素质，采取新型平时成绩管理规则；并方便老师录入及管理成绩，学生查询成绩而开发。

该项目由教务处统一管理，体育教务长、体育教师、学生可以登录系统进行对应操作，系统管理员负责分配使用权限。为了保证系统安全性，对体育教务长、体育教师和学生三种用户角色设定了不同的权限，系统管理员有最高权限。

该系统作为学校的学生管理系统的子系统。

## 1.2 范围

(1) 名称：学生体育成绩管理系统（*Students Physical Education Grades Administration System*）

(2) 该系统软件适用于大学，可作为学生管理系统的子模块。主要支持收集每个体育老师的评分规则、老师录入学生各项成绩、体育教务长核实发布成绩、学生查询成绩等功能模块；

不适用于体育以外科目成绩、学生学籍、选课等管理。

(3) 该系统旨在构建全面的体育评测模式。按照体育老师们不同的评分标准，将学生们每天或每周或其他频率的运动量算入平时成绩中，以提高学生们的身体素质；允许体育教务长核实、发布成绩，学生查询具体成绩，保证系统功能的全面性，提升学生体育成绩管理效率。

## 1.3主要功能

(1) 登录：系统管理员、体育教务长、体育老师、学生均可使用自己的账号密码登录SPEGAS，系统通过数据库对其验证；

(2) 权限控制：系统管理员拥有最高权限，其余体育教务长、体育老师、学生的权限依次降低，对不同权限的用户呈现不同的界面；

(3) 计算成绩：根据老师录入的数据计算出平时成绩，并求出总成绩；

(4)用户功能：

a. 系统管理员：管理体育教务长、体育老师、学生的使用权限；

b. 体育教务长：查询所有成绩，签名认定、发布成绩，当成绩有错时，授予体育更改成绩权限；

c. 体育老师：录入平时成绩计算规则，录入、修改并提交体育成绩；

d. 学生：传入平时成绩的数据，查询体育总成绩及其组成部分的成绩。

## 1.4 术语与缩写定义

|  |  |
| --- | --- |
| ****术语/缩写**** | ****释义**** |
| **账号** | **学生/职工账号** |
| SPEGAA | *Students Physical Education Grades Administration System* |
| C/S模式 | *Client/Server模式* |
| 用例图 | 用例图是指由参与者（Actor）、功能用例，以及他们之间的关系构成的图。其目的是描述系统功能的视图 |

表-1 术语与缩写定义表

## 1.5 引用

1. 王安生，《软件工程化》[M].北京：清华大学出版社，2014

# 2. 体系结构需求

## 2.1 关键指标

|  |  |
| --- | --- |
| 吞吐量 | 一秒钟内支持用户量不小于10000人 |
| 系统响应时间 | 正常情况：小于2秒  30000 |
| 最大并发数 | 5000人 |
| 最大用户数 | 100000人 |

## 2.2 用户特征

|  |  |
| --- | --- |
| 用户角色 | 特征 |
| 学生 | 拥有本学校学籍的学生， 可以查询自己的各项体育成绩，成绩若有误以申请修改 |
| 体育老师 | 本学校该学期任课的体育老师，可以制定体育平时成绩评定规则，录入学生的各项成绩，修改并提交学生成绩 |
| 体育教务长 | 本学校体育组教务长，管理体育老师，可以核实发布体育成绩 |
| 系统管理员 | 系统运行与管理的最高权限者，分配其他用户的权限 |

## 2.3体系结构用例

**2.3.1用户角色**

该系统有五种用户角色（参与者）：学生、体育老师、体育教务长、系统管理员和档案，图-1展示了各个用例和参与者的关系，以及用例，参与者本身之间的联系。

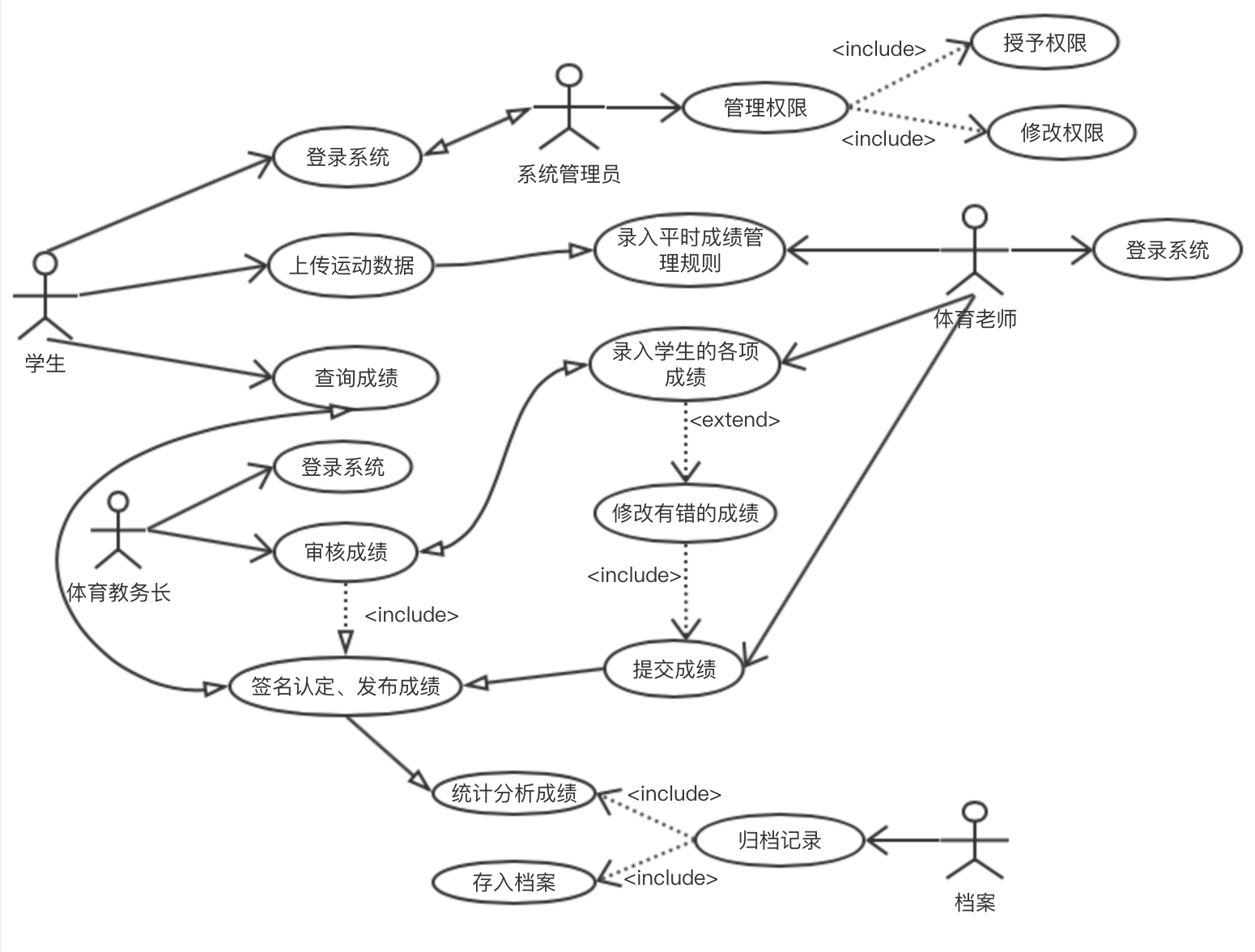


图-1 用户角色图

### 2.3.2用户角色1-学生

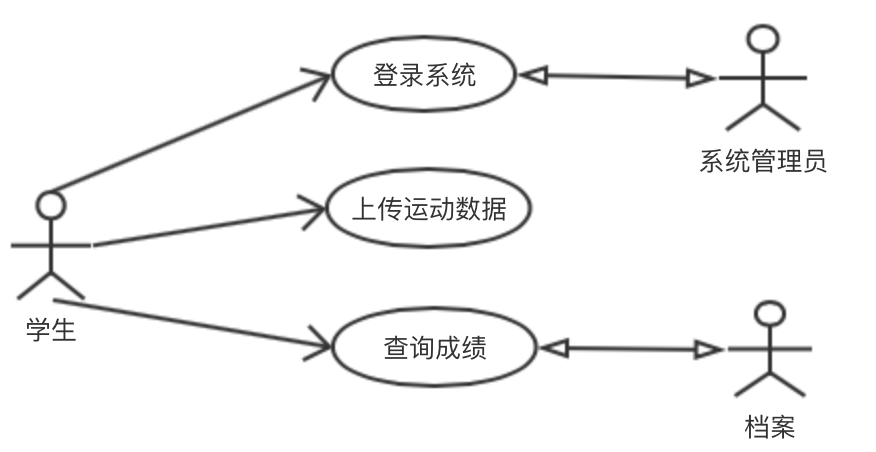


图-2 用户角色1-学生图

### 2.3.3用户角色2-体育老师

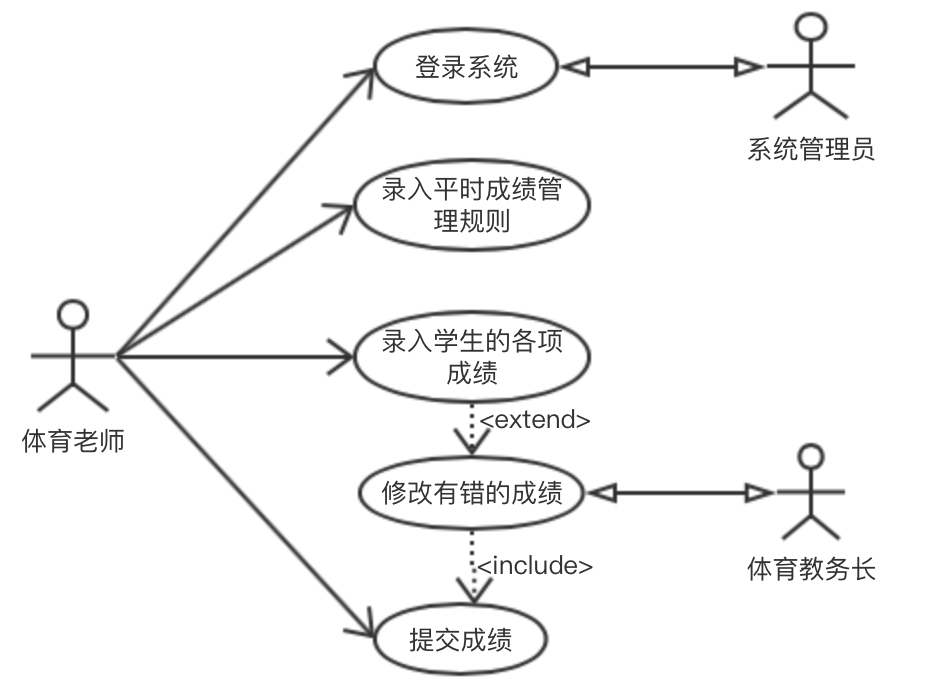


图-3 用户角色2-体育老师图

### 2.3.4用户角色3-体育教务长

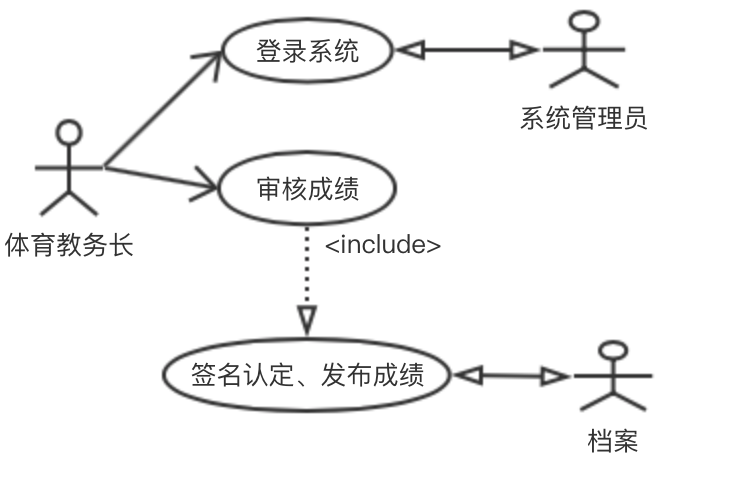


图-4 用户角色3-体育教务长图

### 2.3.5用户角色4-系统管理员

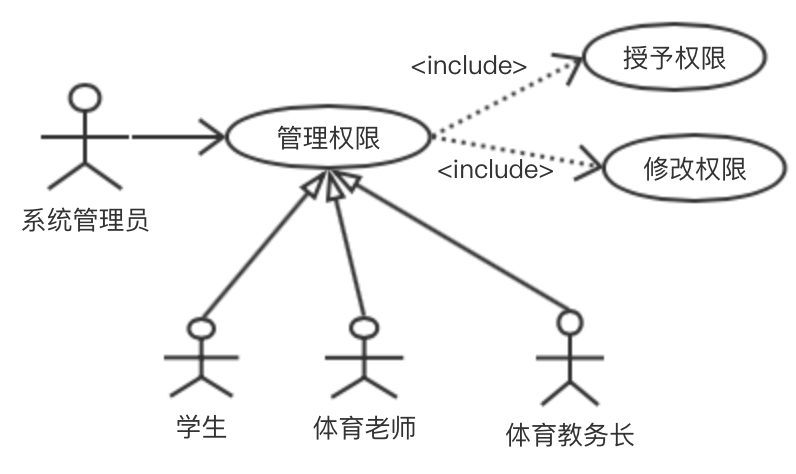


图-5 用户角色4-系统管理员图

### 2.3.6用户角色5-档案

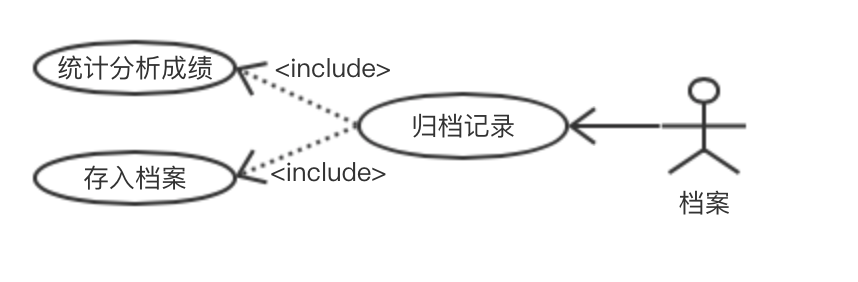


图-6 用户角色5-档案图

## 2.4 各相关方对体系结构的要求

最终用户： 要求每一个角色对应的功能齐全，不同角色功能的特性要明显，并且逻辑合理，便于使用；

开发人员：关注软件的管理。使用Java EE编写，便于维护；

学校：系统能支持100000个用户，最大并发量为5000人，正常情况响应时间小于2秒，最大并发量下响应时间控制在5秒以内；为保证系统安全，只能在校园网内使用，不允许丢失信息；系统需要保证24x365正常运行。

**2.5约束条件**

|  |  |
| --- | --- |
| 约束 | 描述 |
| 业务与项目约束条件 | |
| 项目进度要求 | 产品第一版需在60天以内交付 |
| 系统开发成本 | 10000～15000元 |
| 项目开发要求 | 使用Java EE编写，便于维护 |
| 业务约束 | 系统作为学校学生信息管理系统的子系统，导入父系统的数据，最终结果可被父系统使用 |
| 质量与可信赖性约束 | |
| 性能 | 正常情况响应时间小于2秒，最大并发量下响应时间控制在5秒以内 |
| 资源管理 | 服务器部件的内存必须具有40%以上的预留量 |
| 可使用性 | 系统需要保证24x365正常运行，可使用性达到99%以上 |
| 易用性 | 系统是web应用，用户无需安装客户端 |
| 可靠性 | 只能在校园网内使用，不允许丢失信息 |
| 密安性 | 所有的修改请求必须被授权，并使用认证过程进行加密 |

## 2.6 非功能性需求

## 2.6.1质量需求

### (1) 性能

a.响应时间：正常情况下，系统的响应时间应小于2秒；当有30000人同时使用时，应小于5秒；

b.吞吐量：一秒钟内支持用户量不小于10000人；

c.并发数：并发量不小于5000人；

d. 登录响应时间：正常情况下，时间应小于3秒；当有10000人同时登录时间应小于5秒。

**(2)可靠性**

a.在系统能够长时间稳定运行；

b.对于操作错误应给予提示并采取错误处理，具有基本的容错能力；

c.仍需一个备份，当系统故障时，用于恢复。

**(3)易用性**

a.易理解性：系统的界面、使用说明、功能设计应该能让80%以上用户理解；

b.易学性：对刚刚使用该系统的新用户，能在10分钟以内掌握整个系统的功能；

c.易操作性：系统的75%以上操作应由鼠标能单独完成；

d.吸引性：系统的颜色使用，图形化设计美观。

### (4)密安性

a.系统管理员应准确分配各用户的权限，及时修改与收回权限，避免影响成绩的公正性；

b.系统在登录时应增加验证码功能，防止用户的账号密码泄漏；

c.系统还需防火墙设置，在数据传输过程中避免丢失。

**(5)可维护性**

软件可被修改和维护的能力。在运行中，应当容易判断出系统的缺陷和失效原因。代码、设计和文档应当结构清晰，易于修改。同时保证系统的稳定性，避免多次修改造成代码混乱，文档不清晰。

### (6)可移植性

系统应具有一定的跨平台和环境的能力，作为学校教务管理的子系统，应能够与其在同一平台上运行，而不发生冲突。

## 2.6.2 工程需求

### (1) 逆向需求

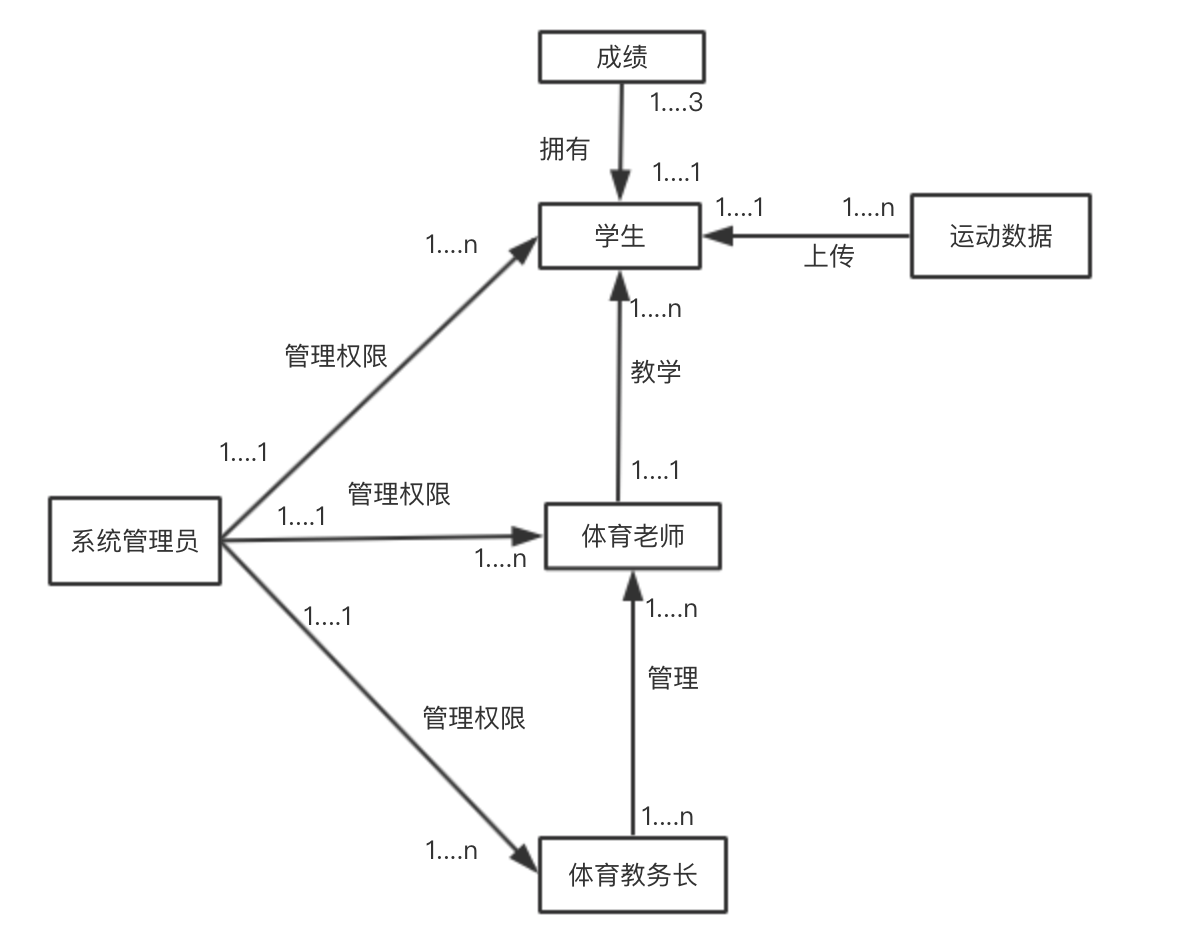
更新系统时，要求分析旧系统，在旧系统的基础上增加新功能，在添加新功能时，

尽量无需修改其他功能。

**(2) 设计约束**

|  |  |  |
| --- | --- | --- |
| 设计要素 | | 主要约束 |
| 运行环境 | 操作系统 | Windows 7/Linux 10及以上 |
| 数据库 | MySQL 14.0及以上 |
| Web服务器 | WebLogic |
| 用户端PC软件 | 操作系统 | Windows/OSX/Linux/Android/iOS |
| 浏览器 | Chrome/Edge/Firefox/Safari |
| 开发环境支持 | 操作系统 | Linux Ubuntu 16.04 |
| 开发工具 | myeclipse |
| Web服务器 | WebLogic |
| CPU | 3.4 GHz Intel Core i7 |
| 内存 | 16GB |

## (3) 逻辑数据库需求



**2.7 风险**

**2.7.1 风险一：系统并发量超过最大容忍度导致系统崩溃；**

**2.7.2 风险二：在学生上传运动数据、体育老师或教务长提交成绩时，出现网络故障导致传输过程中信息丢失；**

**2.7.3 风险三：服务器的数据库异常，造成用户登录失败；**

# 3. 解决方案

## 3.1 相关的体系结构模式

### 3.1.1 共享数据仓库模式

•在共享仓库模式中，有一个部件作为数据仓储，被其它的独立部件访问。这样的仓库可以被多个部件同时访问；

•对共享仓库模式，必须提供一些机制，例如信息安全机制；

–有些系统提供更高层的访问机制，例如，数据库管理系统(DBMS)的SQL查询语言实现对数据库表的并行访问；

•共享仓库是对顺序性体系结构（例如分层模式和管道过虑器模式）的一种替代，优势在于实现了数据共享，以及不相邻的两个部件之间的信息交流。如果将其用到管道过滤器中，可以实现不同过滤器之间的数据共享；

–在共享仓库中，所有的客户端都是独立的部件，很像是C/S结构，数据仓储扮演了服务器的角色。也可以认为是两层的C/S结构；

### 3.1.2 分层的C/S模式

1）分割：

–表现层、业务层和数据处理逻辑被分割到不同的层面；

2）各层之间异步通信：

–层与层之间的通信是异步的“请求-响应(request-reply)”。请求是单方向的，从客户层开始，经过Web和业务逻辑层，再到数据管理层。每层都等待其它层的处理响应。

3）部署灵活：

–分层结构不限制多层应用的部署方式。所有层可以运行在一台机器上，或者，每层部署到一台独立的机器上。

4）标准化：

–分层可以实现各层的标准化，例如，在Web应用中，客户端可以使用不同厂家的浏览器，兼容地对不同厂家Web服务器访问。

采用分层分布式系统框架的C/S结构示意图：



## 3.2 体系结构概述

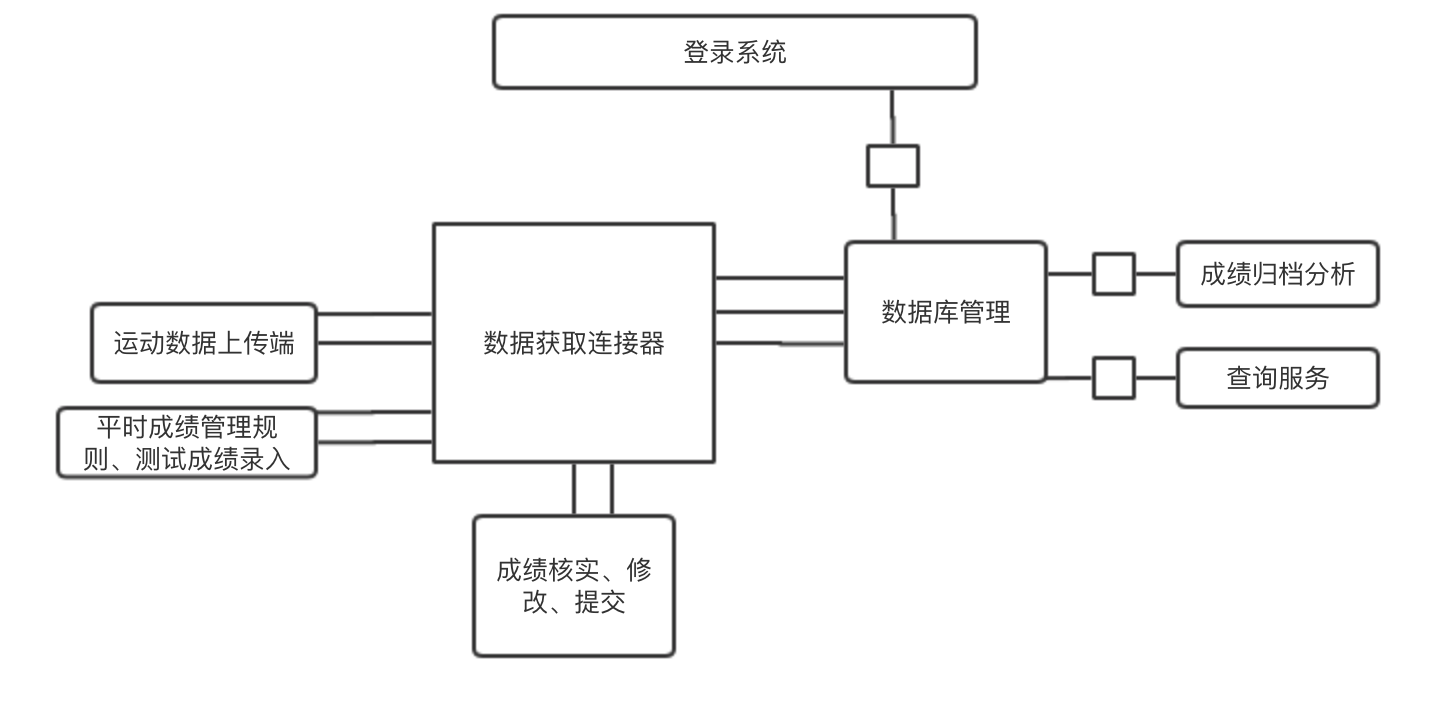
学生体育成绩管理系统核心在于学生可以上传自己的运动数据，体育老师可以制定平时成绩管理规则以督促学生加强锻炼，提高身体素质；同时还要支持一般成绩管理系统的成绩查询、核实、修改、提交功能。

在系统运行过程中，各个角色均访问同一类数据，为了实现数据共享以及不同角色的部件之间信息交流，采取共享数据仓库模式。

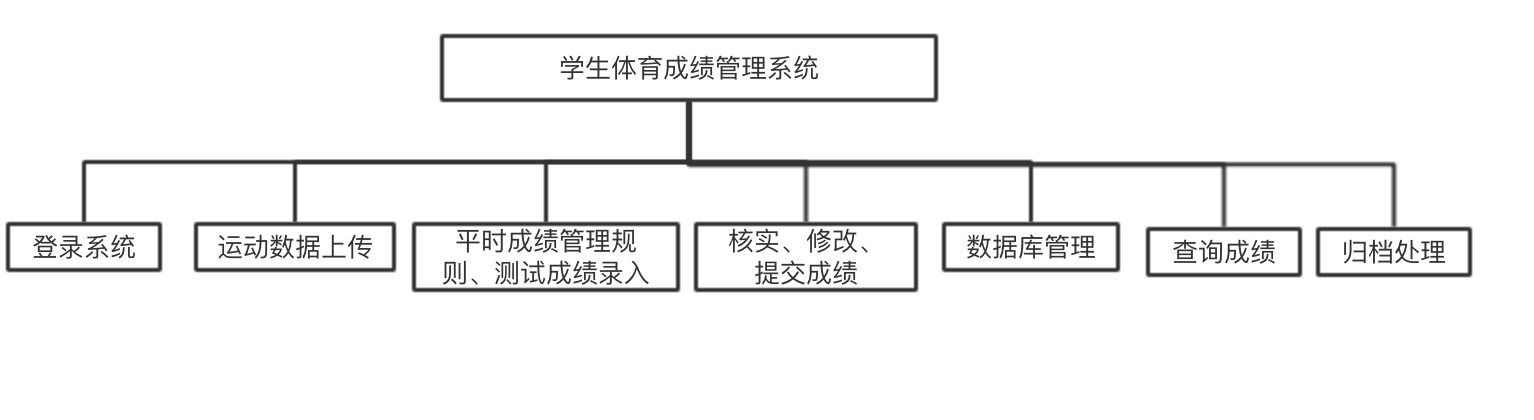
选用分层的C/S模式的原因： 系统需实现学生实时上传运动数据的功能，需要读取手机的运动数据并上传，面临手机操作系统不同，软件版本控制较麻烦，使用客户端不容易统一，不利于成绩管理。使用分层的C/S模式可以通过浏览器调取手机运动数据上传，学生也可使用浏览器查询成绩。其他角色用户的功能同样适合分层的C/S模式开发，于是统一各个功能的结构模式，同时节约了开发成本。

## 3.3 结构化视图

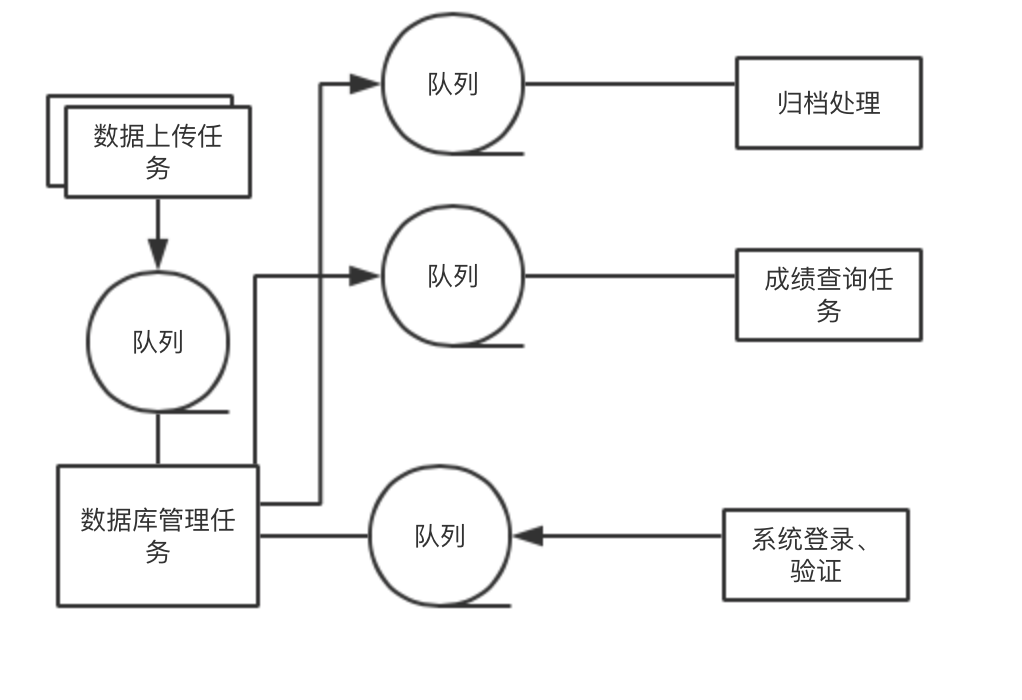
### 概念级体系结构视图



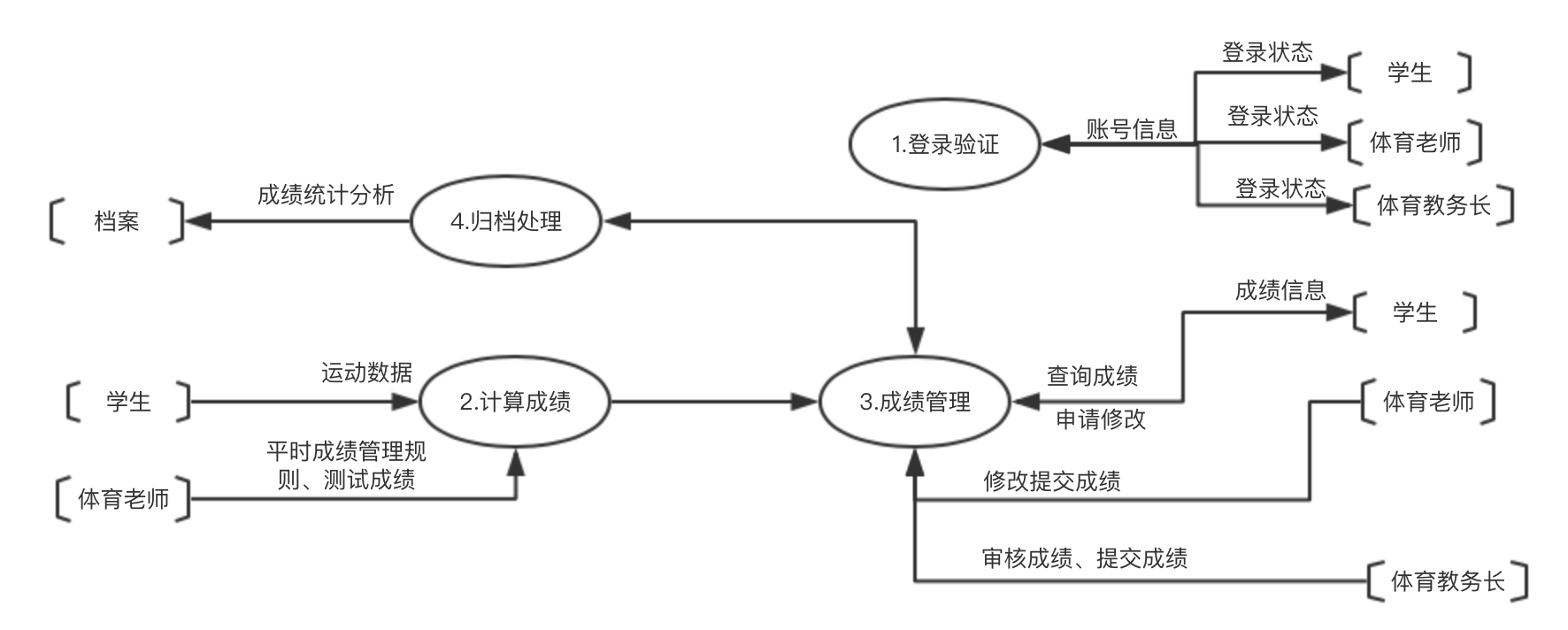
### 3.3.2模块级体系结构视图



### 3.3.3运行级体系结构



## 3.4 行为视图



**3.5 实现问题**

（1）软硬件条件的限制；

（2）软件系统设计人员对系统理解不到位；

（3）软件开发人员编程水平有限；

（4）时间限制。

# 4. 系统的质量分析与评价

## 4.1 场景分析

### 用例场景

4.1.1.1场景1：学生登录系统后，可以随时提交自己的运动数据，系统可以调用手机的接 口读取历史运动数据上传到服务器，再结合对应的体育老师的平时成绩管理规则计算平时成绩，而不用手动计算平时成绩。该场景代表了学生用户期待系统易于使用，即易用性；

* + - 1. 场景2：发生数据异常时，系统要通知系统管理员，并提示修改。该场景代表了用户期望的可靠性；
      2. 场景3：学生用户在手机端浏览器使用系统，体育老师与教务长在电脑端使用系统。该场景代表了用户期望的可使用性；
      3. 场景4：用户在登录系统时，期望填写验证码防止账号密码泄漏。该场景代表了用户期望的可靠性；
      4. 场景5: 用户通过web获得数据报告的时间小于5秒。该场景代表了系统的性能要求；
      5. 场景6: 当处理器发生故障时，缓存系统要能在1秒中内从一个处理器切换到另一个处理器，该场景代表了系统可靠性要求。

### 增长性场景

4.1.2.1增长性场景1: 可以增加新的数据服务器，将用户通过Web访问时间从5秒降低到2.5秒，希望系统仅需增加一人周的工作量就能完成对系统的调整；

4.1.2.2增长性场景2:通过扩充现有数据库表的规模，把检索时间降低到平均1秒之内；

4.1.2.3增长性场景3:学生使用手机端浏览器访问系统时在一段时间内可以记住密码，更方便地访问查询。

### 探索性场景

* + - 1. 探索性场景1：改进系统的可使用性，从98% 提升到到 99.999%；
      2. 探索性场景2：当全校师生10000人同时使用该系统时，系统可承受并且反应时间控制在5秒内；
      3. 探索性场景3：对学生用户上传的运动数据进行分析，及时反应运动情况，督促运动少的用户加强锻炼；
      4. 探索性场景4：将系统做成一个微信小程序，方便手机端用户上传数据与查询。

## 原型分析

使用原型法进行需求分析的流程如下：

1. 快速分析，列出用户的基本信息需求：

学生体育成绩管理系统的用户可分为学生、体育教师和体育教务长三种角色，其中学生的需求是能够安全登录账户、上传运动数据、及时查询成绩、保证成绩的正确性等；教师的需求是安全登录账户、录入平时成绩管理规则、及时录入成绩、及时修改并提交成绩；体育教务长的需求是安全登录账户、及时审定发布成绩，授权老师修改成绩和管理系统信息等；

1. 构造原型，开发初始原型系统：

原型系统先考虑原型系统应必备的待评价特性，暂时不考虑系统的安全性、健壮性，建立基本满足学生、体育老师、体育教务长使用要求的系统；

1. 用户和开发人员共同评价原型：

本步骤要求系统的使用者（学生、教师和教务员）使用原型进行初步的评价，还要求开发人员对原型系统进行最基本的测试工作，从而找出系统需要改进的地方并进一步进行完善。

## 风险

* 风险1: 学生用户在上传运动数据时，出现网络障碍，数据上传不完全却无法修改，导致平时成绩受影响；
* 风险2: 学生用户在该学期手机数据丢失或者更换手机，将导致运动数据丢失、无法上传；
* 风险3: 大量学生用户查询体育成绩导致系统响应变慢甚至崩溃；
* 风险4: 学生用户在查询成绩成绩的同时其体育老师正在修改该班级的体育成绩，导致修改不成功或者查询失败；
* 风险5: 学生体育成绩管理系统的父系统——学生信息管理系统中信息变动，则可能会导致子系统信息未及时更新，成绩出差错。