**学生运动成绩管理系统  
  
体系结构设计  
1.0**

**2017/12/25  
周振靖  
2015211986  
软件架构师**

**修订历史**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 描述 | 作者 | 备注 |
| 2017/12/25 | 版本 1.0 | 周振靖 | 第一版本文档编写 |
|  |  |  |  |
|  |  |  |  |

表格 1

**文档批准**

以下体系结构设计文档报告已经被以下机构人员批准并认可：

|  |  |  |  |
| --- | --- | --- | --- |
| 签名 | 打印姓名 | 职位 | 日期 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

表格 2

目录

[1 简介 5](#_Toc502419208)

[1.1 文档目的 5](#_Toc502419209)

[1.2 范围 5](#_Toc502419210)

[1.3 术语定义 6](#_Toc502419211)

[1.4 参考文献 6](#_Toc502419212)

[1.5 总览 6](#_Toc502419213)

[2 体系结构需求 7](#_Toc502419214)

[2.1 功能需求 7](#_Toc502419215)

[2.2 功能结构 9](#_Toc502419216)

[2.3 用例 10](#_Toc502419217)

[2.3.1 用户认证 10](#_Toc502419218)

[2.3.2 发布成绩 11](#_Toc502419219)

[2.3.3 修改成绩 12](#_Toc502419220)

[2.4 相关约束 12](#_Toc502419221)

[2.5 非质量需求 12](#_Toc502419222)

[2.5.1 性能 12](#_Toc502419223)

[2.5.2 可靠性 13](#_Toc502419224)

[2.5.3 易用性 13](#_Toc502419225)

[2.5.4 安全性 13](#_Toc502419226)

[2.5.5 可维护性 13](#_Toc502419227)

[2.5.6 可移植性 13](#_Toc502419228)

[3 解决方案 14](#_Toc502419229)

[3.1 概念级体系结构 14](#_Toc502419230)

[3.2 模块级体系结构 15](#_Toc502419231)

[3.2.1 横向模块划分 15](#_Toc502419232)

[3.2.2 纵向层次划分 16](#_Toc502419233)

[3.3 运行级体系结构 17](#_Toc502419234)

[4 系统的质量分析和评价 17](#_Toc502419235)

[4.1 正确性 17](#_Toc502419236)

[4.2 性能 17](#_Toc502419237)

[4.3 可靠性 18](#_Toc502419238)

[4.4 安全性 18](#_Toc502419239)

[4.5 可维护性 18](#_Toc502419240)

[5 修订管理 18](#_Toc502419241)

[A. 附录 18](#_Toc502419242)

# 简介

## 文档目的

本文档是学生运动成绩管理系统的体系结构设计文档，乃是根据学生运动成绩管理系统这一项目的需求分析报告进行体系结构设计得到的设计初稿。文档对项目从各个方面做出了详细的分析和设计，对功能需求和非功能需求均有考虑。本文档一方面准确、详细、无二义性地描述了当前的体系结构设计，一方面对开发人员、测试人员接下来的工作起到指导和参考作用，另一方面也面向所有设计人员征求优化、改进建议。同时，本文档将用于甲方追踪项目进度，提高工作效率和质量。

本文档面向的读者如下表。

|  |  |
| --- | --- |
| 读者 | 目的 |
| 项目经理 | 依据此文档了解整个项目的架构、各部分职责，对项目的工期、成本进行评估，以便于合理地监视、安排和管理项目的进度。 |
| 设计人员 | 评估此文档所述的项目体系结构，对当前设计的不足进行修正并修订此文档，并依据此文档进行各部分的详细设计。 |
| 开发人员 | 通过此文档了解整个项目的体系结构，并参照此文档进行实际开发工作。开发人员所开发出的产品应当严格符合本文档所描述的体系结构的各方面设计。 |
| 测试人员 | 按照此文档对项目按体系结构划分的各个部分进行分层、详细的测试，确保每一个模块的可靠性、逻辑的正确性，并结合需求分析文档，确认当前产出的产品符合客户的需要 |
| 甲方 | 根据本文档了解乙方当前做出的设计，并对其合理性和正确性做出评估，如果有问题及时向乙方提出，请求乙方对此作出修改。此外，甲方需要根据此文档追踪项目的进度。 |

表格 3

## 范围

本项目名称为：学生运动成绩管理系统。

该系统用于管理学生的体育课成绩和实时录入学生步行数据，并提供根据平时成绩、体育测试项目成绩计算并记录学生的最终体育成绩的功能，目的是方便学生的查询和节省教师的工作量。本系统不能用于管理学生的学籍，不能用于管理其它课程的信息。本系统需要通过现有的学籍管理系统的API来导入学生的个人信息。

本系统分为两部分，一部分是Web端的管理系统，另一部分为手机客户端。手机客户端用于记录学生的步行步数并同步到管理系统中，其余功能均实现在Web端系统上。在后期的迭代中，可在学生用的手机客户端加入一些查询功能，方便学生的使用。

## 术语定义

|  |  |
| --- | --- |
| 术语 | 解释 |
| 学生运动成绩管理系统 | 即该系统的名称，特质主要的Web部分。 |
| 系统 | 为方便叙述，用“系统”来代指学生运动成绩管理系统。 |
| 客户端 | 供学生使用的安装在手机上的客户端软件，是本系统的一个组成部分，但是独立运行。 |
| 学生 | 使用学生运动成绩管理系统的学生。 |
| 体育教师 | 使用学生运动成绩管理系统的体育教师。 |
| 体育教务长 | 使用学生运动成绩管理系统的体育教务长。 |
| 系统管理员 | 管理本系统的管理员。 |
| 平时运动成绩 | 基于学生的不行步数计算得来的成绩，学生的体育总成绩的其中一部分。 |
| 体育课成绩 | 学生在体育课得到的成绩，学生的体育总成绩的其中一部分。 |
| 总成绩 | 学生的体育总成绩，由平时运动成绩、体育课成绩两部分成绩汇总得到。 |
| 成绩 | 以上三种成绩的合称。 |

表格 4

## 参考文献

[1] 王安生.《软件工程化》[M]. 北京：清华大学出版社，2014.

## 总览

本文档为学生运动成绩管理系统的项目体系结构设计文档。本文档将在接下来的四个章节从不同角度描述当前的体系结构设计。

* 第一章（即本章）简要描述了本文档的撰写目的、适用范围、术语定义等，提供本文档的一个总览。
* 第二章系统的叙述了该项目的体系结构的需求。这部分内容大致地决定了后续的设计思路，并一定程度上解释了为何采用此种设计方案。
* 第三章系统地描述了我们对该项目当前的设计。
* 第四章对第三章所述的设计方案进行了一个全面的评估，对当前的体系结构设计方案的优缺点等方面进行了详细的分析，对今后的设计细化和修订工作有一定的指导作用。
* 第五章对本文档后续的修订工作的注意事项和操作流程进行了一些约定，以保证工作高效可靠。

# 体系结构需求

## 功能需求

1. 用户认证：  
   基本的用户认证系统。包括：
   1. 登录
   2. 注销
   3. 修改密码
   4. 设定个人资料

不提供新用户注册功能。

1. 学生/查询成绩：  
   学生可以查询自己的各项成绩。
2. 学生/记录步行数据：  
   学生安装手机客户端，手机客户端自动在后台记录学生的步行数据，并将数据同步到服务器，作为评定自己成绩的一个方面。
3. 体育教师/设定平时成绩计算规则：  
   体育教师可能希望根据学生的步行数据，使用较为复杂的自定义计算规则得到学生的平时成绩。这些规则至少包括：
   1. 分数为关于数值X的线性函数
   2. 分数为一固定值
   3. 根据条件进行分支判断

其中可用作分支判断的条件至少包括：

1. 数值X大于规定值
2. 数值X大于等于规定值
3. 数值X小于规定值
4. 数值X小于等于规定值
5. 条件A或条件B
6. 条件A且条件B
7. 条件A不满足

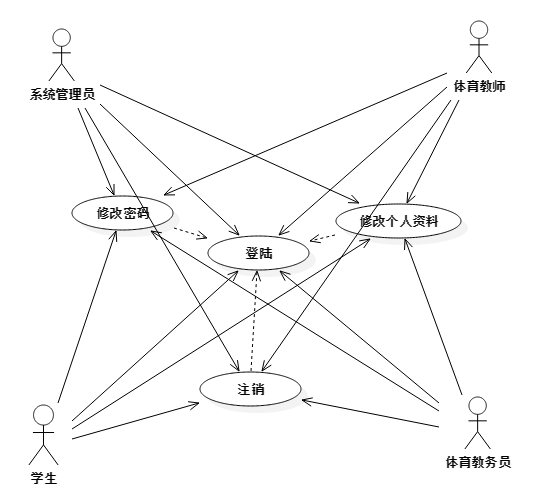
其中这里可用的数值X至少包括：

1. 步数大于某一规定值的天数
2. 总步数大于等于某一规定值的周数
3. 平均每天步数大于等于某一规定值的周数
4. 总步数大于等于某一规定值的月数
5. 平均每天步数大于等于某一规定值的月数
6. 期间内总步数
7. 期间内平均步数
8. 体育教师/录入学生成绩：  
   体育教师批量录入学生的体育课成绩。
9. 体育教师/修改学生成绩：  
   特殊情况下，体育教师可以单独修改某一学生的成绩
10. 体育教师/统计成绩：  
    根据学生的步数记录和体育课成绩，统计出所有学生的成绩。
11. 体育教师/查询所有学生的成绩：  
    体育教师有权限查看该教师所管理的所有学生的成绩。
12. 体育教师/申请认定成绩：  
    体育教师在统计完成绩、处理完需要单独修改的记录、确认无误之后，可以提交成绩单给教务长申请认定。
13. 体育教务长/查阅成绩：  
    体育教务长查看他所管辖班级的全部成绩。
14. 体育教务长/签名认定成绩：  
    在体育教师提交申请认定之后，审阅成绩单确认无误后即可签名认定成绩。
15. 体育教务长/发布成绩：  
    签名认定后的成绩需要发布。发布之后，成绩将公开给学生查询（仅可查询自己的成绩），系统将会把成绩数据同步到教务系统。
16. 体育教务长/控制体育教师的权限：  
    仅在体育教务长授权的情况下，体育教师可以进行修改、录入成绩等操作。
17. 系统管理员/全权管理整个系统：  
    以上所有功能，系统管理员均有权限进行，包括创建教务长用户。这对于系统管理员维护系统的运行是有必要的。
18. 系统管理员/设定系统设置：  
    系统管理员可以配置系统设置。

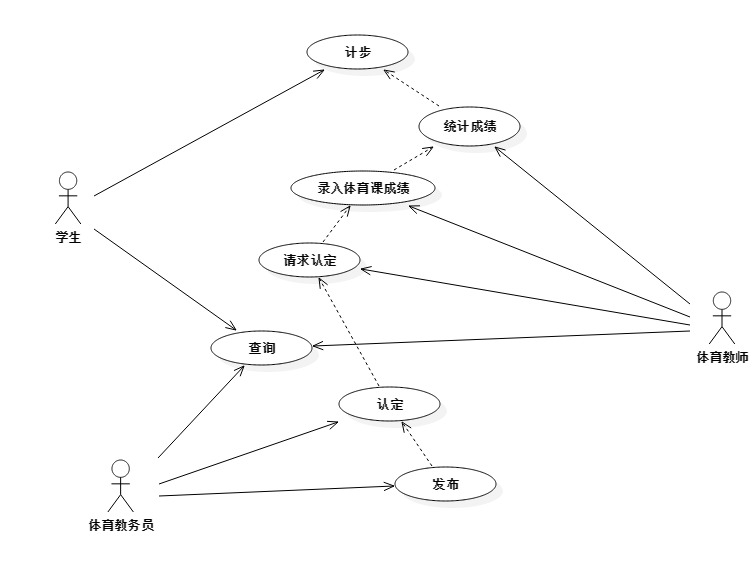
## 功能结构

## 用例

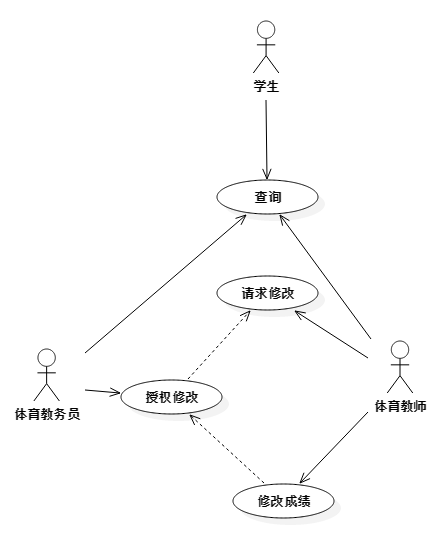
### 用户认证



### 发布成绩



### 修改成绩



## 相关约束

* + - 1. 项目主体应当应用B/S 架构，用户一般访问进行查询成绩等操作时不需要在自己的设备上安装特殊的程序。
      2. 计步功能的实现这一部分采用C/S架构（需要使用手机客户端）
      3. 学籍信息支持通过与现有学籍管理系统或教务系统对接来获取。

## 非质量需求

### 性能

本系统将会有30000名学生、200名教师使用，而且由于计步过程中要经常同步到服务器，所以存在巨大的被动的访问量。因此，本系统对性能要求很高。

1. 本系统应当能支持（M:12000 D:28000 B:35000）个用户的计步客户端同时进行步数上传操作。
2. 本系统应当能支持（M:5000 D:10000 B:15000）个用户同时流畅地进行查询操作。
3. 本系统应当能支持（M:100 D:150 B:200）个教师同时流畅进行批量录入体育课成绩操作。

### 可靠性

本系统关系到30000多名学生的成绩，所以在可靠性方面，首先要尽可能避免运行过程中因为故障而产生错误数据。其次本系统实现上应当支持多活容灾，避免因意外情况导致服务停止运行。在此基础之上，尽量避免可能导致服务崩溃的缺陷。

此外，对于用户的错误输入需要考虑全面。尤其是在用户人数众多的情况下，处理用户错误输入不够全面导致的BUG更容易显现出来。因而，保证本系统的高容错性和发生问题时的稳定性是非常重要的。

### 易用性

本系统的设计应建立在用户不具备专业计算机知识的假设之上。界面应当设计合理，符合一般人使用电脑的操作习惯，不需要专业计算机知识即可食用。这一点要求在大多数情况下都很容易得到保证，只有一点比较复杂，也就是体育教师设置根据步数计算成绩的规则的界面。因而在设计这一部分时应当特别注意。

### 安全性

本系统的安全性十分重要。为了保证系统的安全性，在这里提出如下要求：

1. 严格保证未经认证的用户不能访问到关键信息。
2. 严格保证未经授权的操作不能被以外地、或通过不正常途径执行。

除此之外，对实现方面也提出如下要求：

1. 用户的密码不得直接保存在数据库中，应当加盐取Hash再保存。
2. 尽量避免在业务代码中直接使用SQL访问数据库，应当使用ORM来代替。
3. 服务器除了80（HTTP）、443（HTTPS）不得对公网开放其它端口。
4. 所有表单加入CSRF验证
5. 多次登陆失败之后再次登陆要求输入验证码。

### 可维护性

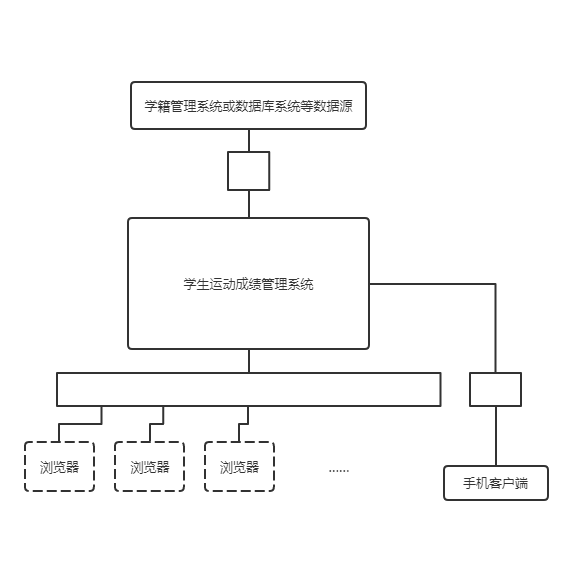
代码应当遵从面向对象设计原则，设计上要做到易扩展、易维护，以减少以后的维护和二次开发所需的成本，保证代码的质量。

### 可移植性

本系统一般来说不需要多平台部署，对可移植性没有太高的要求，但是也应当兼容同一系列操作系统的不同版本（如，如果选择使用Linux的话应当兼容不同版本的Linux）。具体选择何种平台可在开发过程中商定。

# 解决方案

## 概念级体系结构



如图所示。整体采用B/S 架构，但是其中部分（为了实现记录学生步行步数）采取了局部的C/S架构。实际上当今的软件项目中早已没有B/S和C/S的严格区分，往往一个服务既可以用浏览器访问，又可以使用客户端（大多是手机客户端）来使用。采用这样的设计基于如下的一些原因：

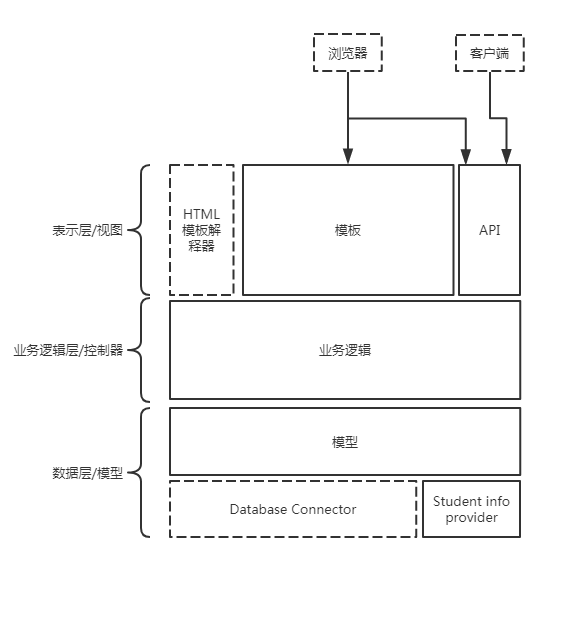
1. 为学生提供更好的用户体验
2. 系统主体及时更新

因为体育成绩而不得不安装一个手机APP这件事情，本身对学生来说，是体验非常差的一件事。这里手机客户端可以只有计步功能，这样功能简单而基本不需频繁更新，并保持客户端自身体积小巧，安装容易。大部分操作（如学生查询成绩，老师录入分数等）都支持在通过浏览器操作。这样如果系统有更新，用户不需要浪费时间更新自己安装的客户端，通过浏览器访问到的页面一定是最新版本。（而事实上，学生使用的计步用的客户端是功能固定的部分，不需要经常更新）

## 模块级体系结构

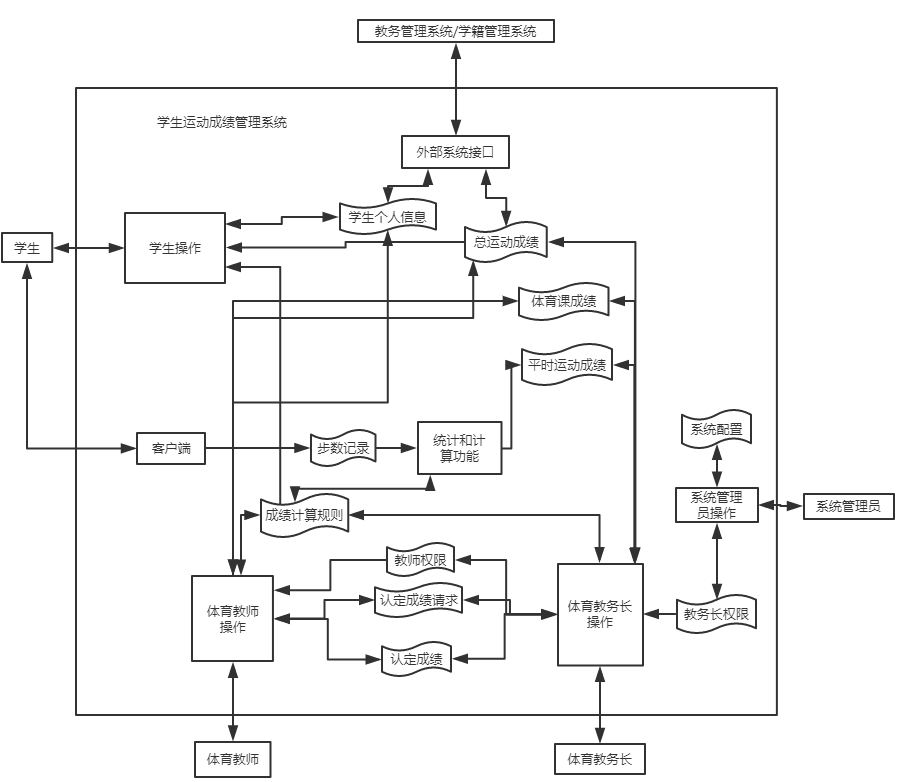
### 横向模块划分

### 纵向层次划分



纵向架构采用饱经历史考验的MVC三层架构。这有多方面的原因：其一，MVC架构在饱经历史的考验之后，如今依然广泛使用，正是得益于它提供的可维护性、可扩展性，比如，试图一层和业务逻辑解耦，因而可以非常方便地更换试图而无需关注下层的具体构造；其二，由于MVC架构应用相当广泛，因而对于我们当前的这个Web应用来说，有很多现有的MVC框架可以采用，大大节省了软件开发成本和减少了工期。上图中的HTML模板解释器和Database Connector通常在这些MVC框架中都会提供。

## 运行级体系结构



参考如上数据流图。

# 系统的质量分析和评价

## 正确性

由于本系统可以在现有的MVC框架基础上进行开发，那么可以认为，可能存在BUG的部分集中在业务逻辑中。而通常，业务逻辑部分很容易编写自动化的单元测试，并且，当今的MVC框架也会内置单元测试框架。因而，系统的质量可以得到很大程度的保证。

## 性能

在代码编写没有出问题的情况下，本系统的性能瓶颈主要可能在两个位置，其一是数据库读写，其二是网络IO。网络IO可以采用Nginx接受用户请求，然后进行负载均衡。数据库也可以通过分库分表或使用分布式解决方案。业务逻辑比较简单，因而不太可能称为性能瓶颈。

## 可靠性

本系统应当支持多活容灾，避免因为机器故障而导致数据丢失。实际上，这和提高性能这个目标有一个同样的途径，即使用分布式的部署方式。

## 安全性

首先，HTTP协议上的安全性（如CSRF验证、反XSS攻击等）基本可以由本项目中所用的MVC Web框架来提供。用户密码的储存方式也由Web框架提供（这些Web框架通常有完整的用户认证系统）。

本系统的权限管理系统比较复杂，没有现有的工具可以直接使用，需要自行实现。可以选择实现为业务逻辑的一部分或者实现为一个工具库（今后如有需要可以复用，但是开发成本较高）。这一点需要在经过讨论后再决定。

一般来说，考虑到安全性的话，应当抛弃HTTP而使用HTTPS。这不属于开发工作，需要客户自行购买和安装证书。

## 可维护性

由于本系统将采用MVC三层架构，并在现有MVC Web框架的基础之上开发，该架构可以保证系统的层次和逻辑清晰。另外，前述设计进行了清晰详细的模块划分。在开发过程中，我们将严格地进行代码review，控制代码质量。以上几点可以保证代码的可读性和可维护性。

# 修订管理

在本系统的开发过程中，难免在某些情况下发现体系结构设计的不合理之处。在这种情况下，如果不得不变更设计，那么本文档也应当跟进修订。

为了保证开发过程的流畅，避免在体系结构设计变更时严重耽搁开发进度，在这里对体系结构设计变更、本文档的修订以及其它相关的事宜做出如下约定：

1. 本文档修订时，要留下可追溯的旧版本的记录。
2. 本文档修订时，应当使用Word的修订功能。
3. 本文档修订时，有额外需要写明的信息应当附在本文档的附录内。
4. 开发过程中，迭代周期不能太长，保证在体系结构设计有变更时可以灵活应对。

# 附录