

# <学生运动成绩管理系统>

## 体系结构设计文档

2017.12.19

姓名：扎西南杰

学号：2015212108

班级：2015211505

# <学生运动成绩管理系统>

## 体系结构设计文档

2017.12.19

姓名：扎西南杰

学号：2015212108

班级：2015211505

|                      |           |
|----------------------|-----------|
| <b>1. 项目相关信息</b>     | <b>3</b>  |
| 1.1 概述               | 3         |
| 1.2 功能描述             | 4         |
| 1.3 使用人群和阅读建议        | 5         |
| 1.4 产品范围             | 6         |
| 1.5 参考文献             | 6         |
| <b>2. 体系结构需求</b>     | <b>6</b>  |
| 2.1 关键指标             | 6         |
| 2.2 体系结构用例           | 8         |
| 2.2.1 用户角色           | 8         |
| 2.2.2 用户功能用例图 #1     | 10        |
| 2.2.3 查询成绩功能用例图 #2   | 10        |
| 2.2.4 发布成绩用例图 #3     | 11        |
| 2.2.5 修改成绩用例图 #4     | 12        |
| 2.2.6 修改权限用例图 #5     | 12        |
| 2.2.6 上传运动记录用例图 #5   | 13        |
| 2.3 各相关方对体系结构的要求     | 13        |
| 2.4 约束条件             | 15        |
| 2.4.1 实现约束           | 15        |
| 2.4.2 设计约束           | 15        |
| 2.4.3 其他约束           | 16        |
| 2.5 非功能性需求           | 16        |
| 2.5.1 可靠性            | 16        |
| 2.5.2 可用性            | 17        |
| 2.5.3 安全性            | 17        |
| 2.5.4 可维护性           | 18        |
| 2.5.5 可移植性           | 18        |
| 2.5.6 性能             | 19        |
| <b>3. 解决方案</b>       | <b>20</b> |
| 3.1 相关的体系结构模式        | 20        |
| 3.2 体系结构概述           | 21        |
| 3.3 结构化视图            | 22        |
| 3.4 行为视图             | 25        |
| 3.5 实现问题             | 26        |
| <b>4. 系统的质量分析和评价</b> | <b>28</b> |
| 4.1 场景分析             | 28        |

|                     |    |
|---------------------|----|
| 4.1.1 用例场景分析 .....  | 28 |
| 4.1.2 增长性场景分析 ..... | 28 |
| 4.1.3 探索性场景分析 ..... | 29 |
| 4.2 原型分析 .....      | 30 |
| 4.3 风险 .....        | 30 |

# 1. 项目相关信息

## 1.1 概述

近年来，不断有媒体传出大学生体侧猝死的新闻，“体侧猝死”从一方面反映出大学生体质的不断衰弱。进入大学以来，大学生继承了高中以来错误的作息习惯，自律能力和习惯往往呈衰退状态，更多的学生选择“宅生活”，作息不规律，缺乏锻炼。且随着手机移动端的飞速发展，学生们的衣食住行得到了极大的方便，这些加助了这种“宅生活”的趋势。天长日久导致学生的整体体质下降。为了提高并督促学生日常锻炼避免“体侧猝死”事件的发生，同时为了丰富体育项目的检测项目，由于近些年来每日步数等一系列运动软件，老师开始关注学生的平时运动，想通过将学生每天的步数作为体育成绩的一部分，以此达到学生走出宿舍，促进学生体质的目的。

传统上，在学期期末，体育老师将会为学生进行体侧，例如：12 min 跑，立定跳远，50 m 加速跑，引体向上以及选修的体育项目等，但这些项目可能对一些体育方面薄弱的同学来说相对不太公平，同时这次学生的数目不断地在扩增。体育老师对学生的体侧成绩和平时成绩进行整合，然后提交到教务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，可能还希望获取到学生具体的各项体侧成绩、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，并且当学生核对成绩时不能得到准确的数据。对于体育老师也不能够随时监督学生的日常锻炼，

因此，基于上述出现的问题，学校决定开发一个学生平时运动成绩系统解决问题。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、平时运动记录和体育成绩管理体验。

## 1.2 功能描述

### (1) 学生

1. 学生可以登陆系统查看相关体育成绩以及自己平时的运动情况。
2. 学生可以每天上传自己的运动记录。

### (2) 体育老师

1. 体育老师可以登录系统，录入自己统计的体侧成绩。
2. 体育老师可以修改体评分函数。
3. 体育老师可以查看、并可以再授权的情况下修改学生成绩。

### (3) 教务

1. 教务员接收教师的申请，同意后经过审核，然后发布成绩。

2. 教务员发布成绩后会向教师发送答复信息。

3. 档案管理，将学生成绩存储到数据库中。

#### (4) 系统管理员

1. 用户权限修改，限定账户类型。账户类型有体育、学生、教务、系统管理员。并且将这些信息维护成一张用户权限表用作登录时使用。

2. 账号密码管理，维护所有用户的账号密码的数据表，可以按要求修改某个用户的密码或者告知某个用户的密码。

3. 增删用户，每当工作人员离职或者学生毕业就需要将数据库中的相应数据删除；若有新工作人员或者学生加入就相应的增加他们的账号密码。

这些可以调用学校师生数据库接口可以得到。

## 1.3 使用人群和阅读建议

本文档的主要内容共分为四部分，分别为：项目相关信息，体系结构需求，解决方案，系统的质量分析和评价。项目相关信息主要对系统的整体结构进行了大致的介绍；体系结构需求则描述了体系结构的相关信息；解决方案则提供了视图和解决的问题；系统的质量分析和评价则提供了场景和原型分析，以及本系统潜在的风险。

本文档面向多种读者对象：

- (1) 项目经理：项目经理可以根据该文档了解预期产品的功能，并据此进行系统设计、项目管理。

- (2) 设计员：对体系结构进行分析，并设计出系统，包括数据库的设计。
- (3) 程序员：配合《设计报告》，了解系统功能，编写《用户手册》。
- (4) 测试员：根据本文档编写测试用例，并对软件产品进行功能性测试和非功能性测试。
- (5) 用户：了解预期产品的功能和性能，并与分析人员一起对整个需求进行讨论和协商。
- (6) 其他人员：如学校领导、学生家长等可以据此了解产品的功能和性能。

## 1.4 产品范围

该产品是在积累了丰富业务经验的基础上进行开发的，在体系结构上，充分考虑了具体用户的实际情况。本产品将主要适用于学生考试成绩管理，主要完成学生运动成绩查询、录入、审定、发布、修改等业务，也可作为学校学生成绩管理系统的一个子模块。

## 1.5 参考文献

《软件工程化》，王安生，P175-P177

# 2. 体系结构需求

## 2.1 关键指标

本系统主要面对学校学生、体育老师和体育教务，可以明显简化对学生成绩录入、查询、修改等繁琐又易于出错的工作。同时，因为所存储的成绩信息属于比较重要的信息，所以要求系统的稳定性和安全性得达到一定的高度，出错率降

低到一定的水平。才能保证软件正常、快速、高效地使用，所以本系统在性能上应当具备以下关键指标：

1> 查询数据响应时间：

在同时有 30000 名用户查询成绩的情况下，系统响应时间不得大于 9 s。

在同时有 10000 名用户查询成绩的情况下，系统响应时间不得大于 7 s。

在同时有 5000 名用户查询成绩的情况下，系统响应时间不得大于 5 s。

2> 插入数据响应时间：

在同时有 30000 名用户上传数据的情况下，系统响应时间不得大于 6 s。

在同时有 10000 名用户上传数据的情况下，系统响应时间不得大于 4 s。

在同时有 5000 名用户上传数据的情况下，系统响应时间不得大于 3 s。

3> 其他所有交互功能反应速度：

在系统的用户数目为 8000 的情况下，响应时间不超过 3 s。

4> 对处理数据量的要求：

对数据的处理量在 G 级别时，系统的响应时间应当在 5s 之内。

所以本系统在可靠性上应当：

1) 出错率

在累计 100000 次服务请求的情况下，其出错率不得高于 3 次。

在累计 1000000 次服务请求的情况下，其出错率不得高于 60 次。

2) 稳定性

在 10000 用户的并发操作下，连续 5 个小时工作的情况下，系统不会出现停止服务的情况。

3) 易恢复性



在停机进行恢复之后损失的数据不得大于 1 h。

## 2.2 体系结构用例

### 2.2.1 用户角色

| 角色    | 职责描述                             |
|-------|----------------------------------|
| 学生    | 查看自己的体育成绩、查询历史运动记录、查询平时运动成绩      |
| 老师    | 录入成绩、申请修改成绩、申请提交成绩               |
| 教务    | 审定和发布成绩、申请发布成绩、批准成绩修改申请、批准发布成绩申请 |
| 系统管理员 | 管理用户权限表                          |

进行平时运动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。

教务长和教务员角色功能进行了相应的互补，变成了了一个教务部门的角色，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查询体育总成绩、每日步数上传、查询历史步数以及查看平时运动成绩。通过查看对存在质疑的地方提出在线申请，主要的成绩质疑问题在于分数和自己的上传的每日步数所合成的分户偏差过大等。

体育老师可以查看学生每日步数、学生体育总成绩、上传学生体育理论成绩、体侧成绩以及上传合算总成绩的换算函数。在考试结束之后额一段时间体育老师被授权允许体育老师对成绩进行一定的修改，体育老师可能收到教务发过来的挂科警告信息，该消息可以通知该体育老师他所授课的班级学生的挂科率太高，体育老师会通过修改评分函数的方式来解决这一问题。当老师通过数据发现某一学生的步数长时间达不到每日步数标准时，会通过邮件的方式和学生进行交流并督促他加紧日常体育锻炼。

体育教务除了审核体育成绩以外，可以授权成绩的修改和发布。还可以进行成绩的展示以及相应数据展示图的生成，当教务发现某以班级的学生挂科率异常

时，教务会和对应的体育老师联系，通过更改换算公式来达到普遍学生成绩的合格。

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。

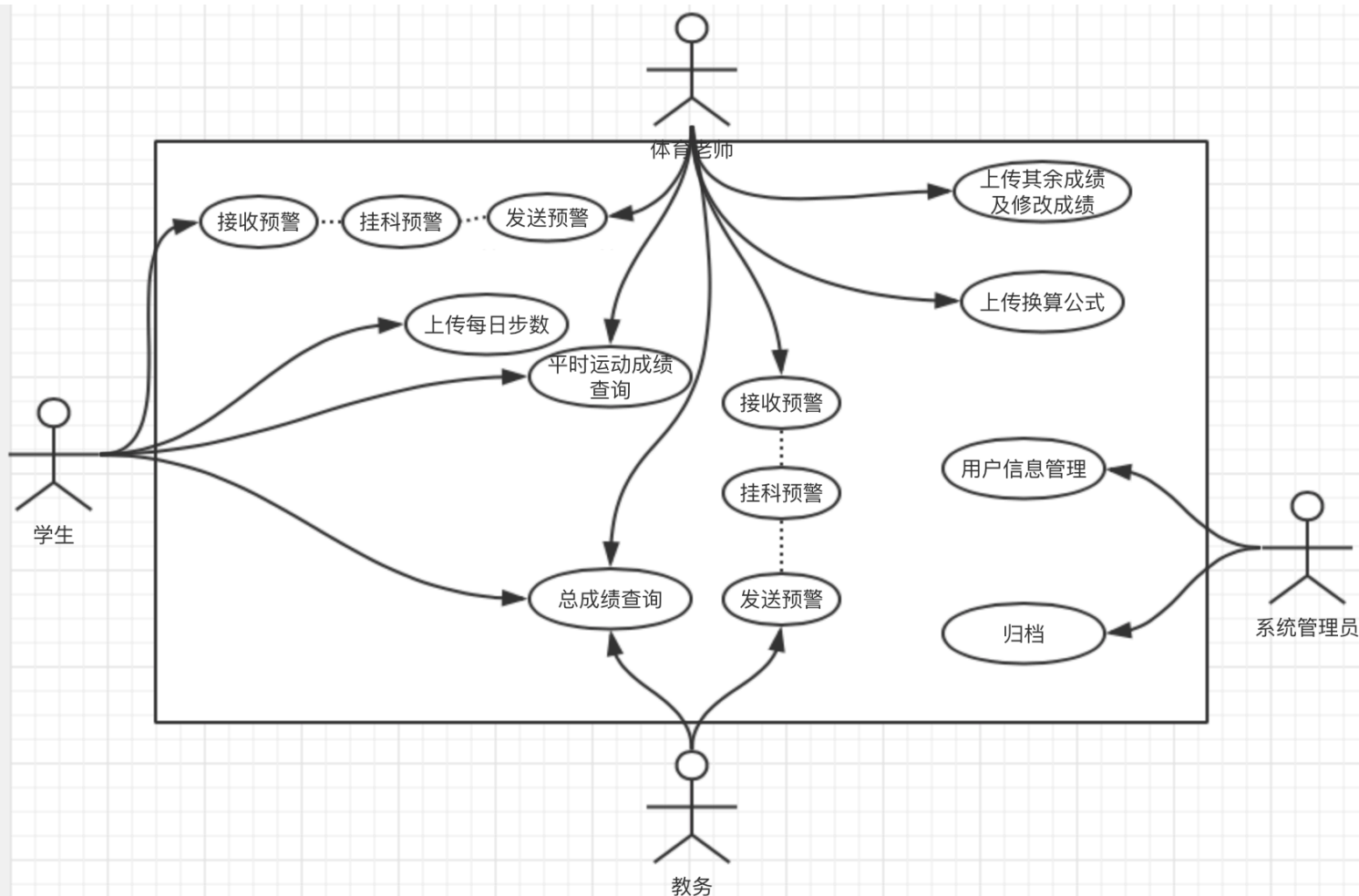


图 2-1 体系结构用例图

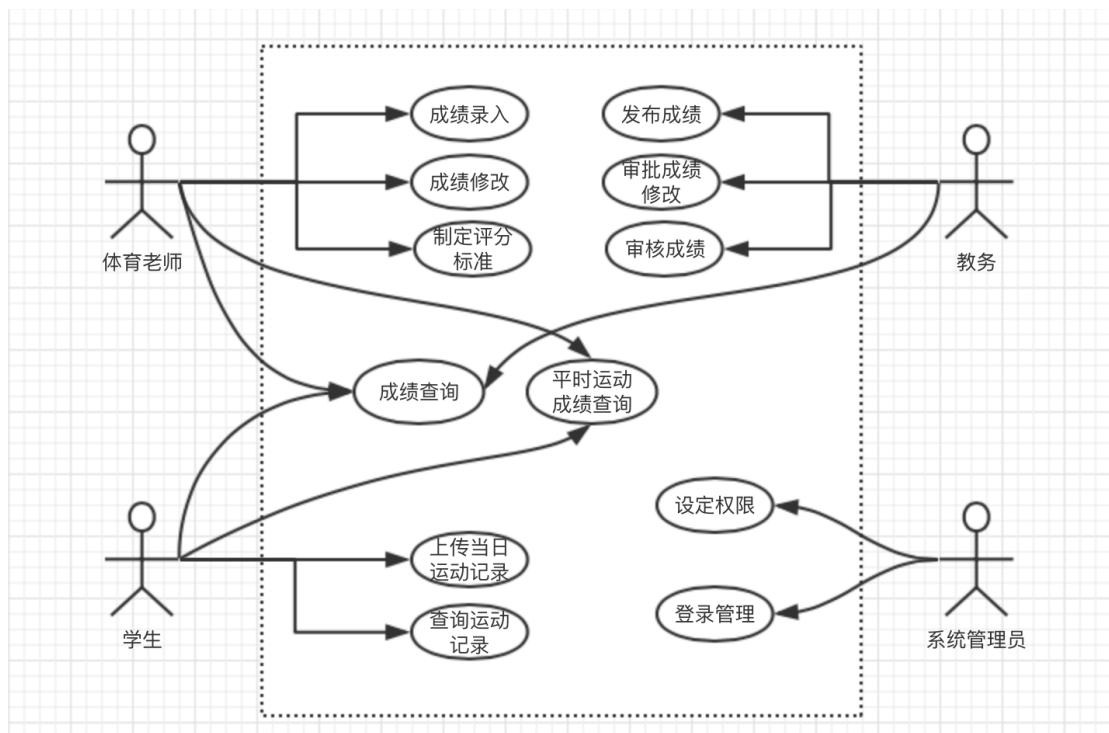
图 2-1 体系结构用例图

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

其中用户界面首先获取用户输入输入的相关账号密码等信息：用户身份验证

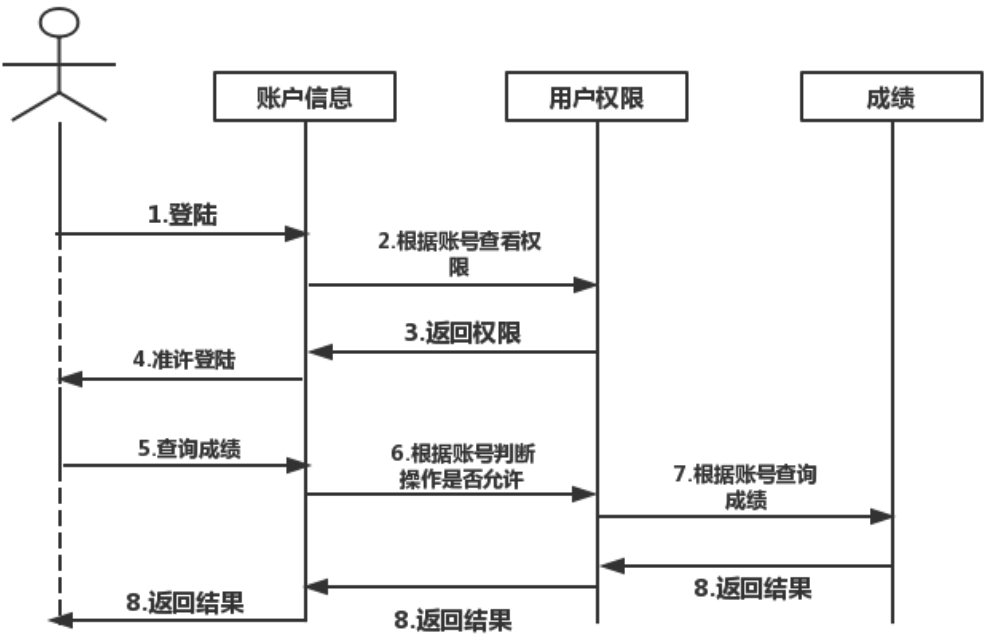
层获取这些信息，进行相关的身份验证，判断用户属于哪一种角色类型，并为当前用户分配相应的使用权限；当用户的账号密码匹配之后就进入了信息检索和修改层，在这一层用户可以管理个人信息，进行相关成绩信息的查询；用户发起一个查询请求之后进入数据库层，系统在数据库之中检索满足条件的相关信息返回给用户；数据信息最终又以页面的方式展示在用户界面层。（实现这部分可以调用学校校园网密码服务的 API）

### 2.2.2 用户功能用例图 #1



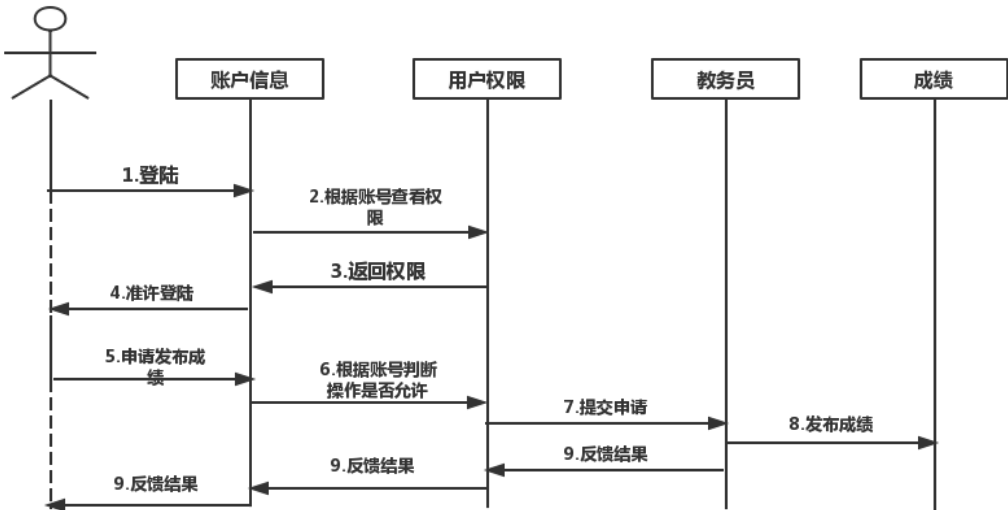
### 2.2.3 查询成绩功能用例图 #2

教师和学生查询成绩的用例



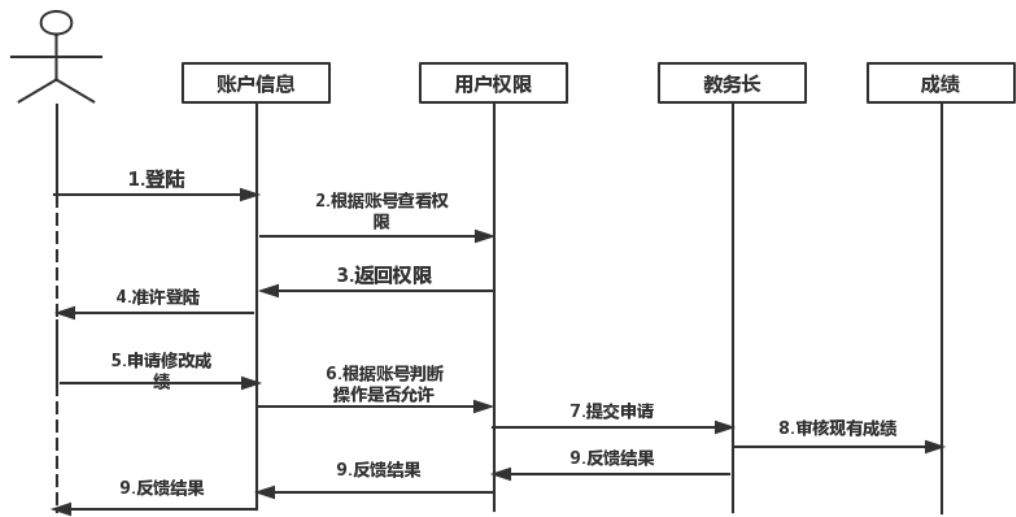
2.2.4 发布成绩用例图 #3

教师申请发布成绩的用例图



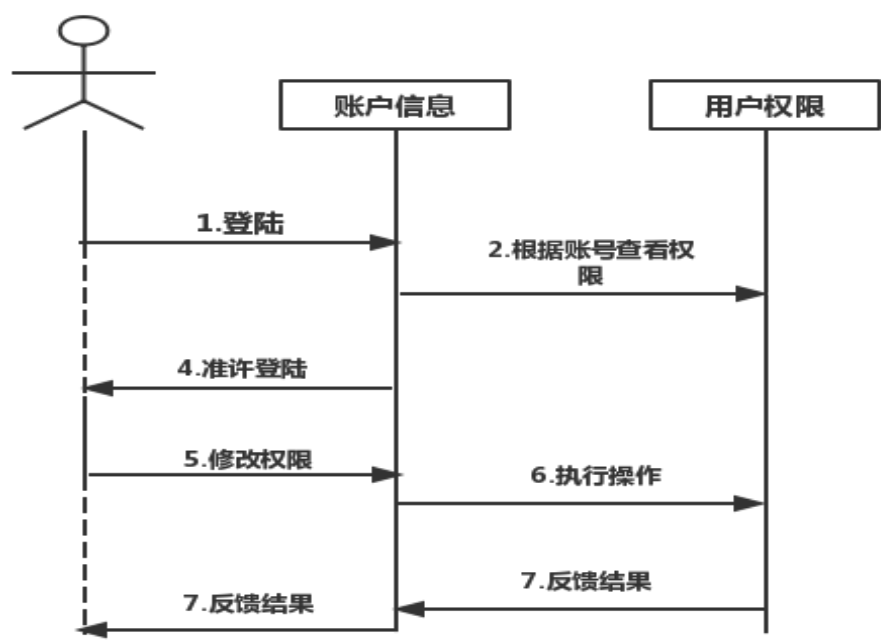
2.2.5 修改成绩用例图 #4

教师修改成绩的用例图

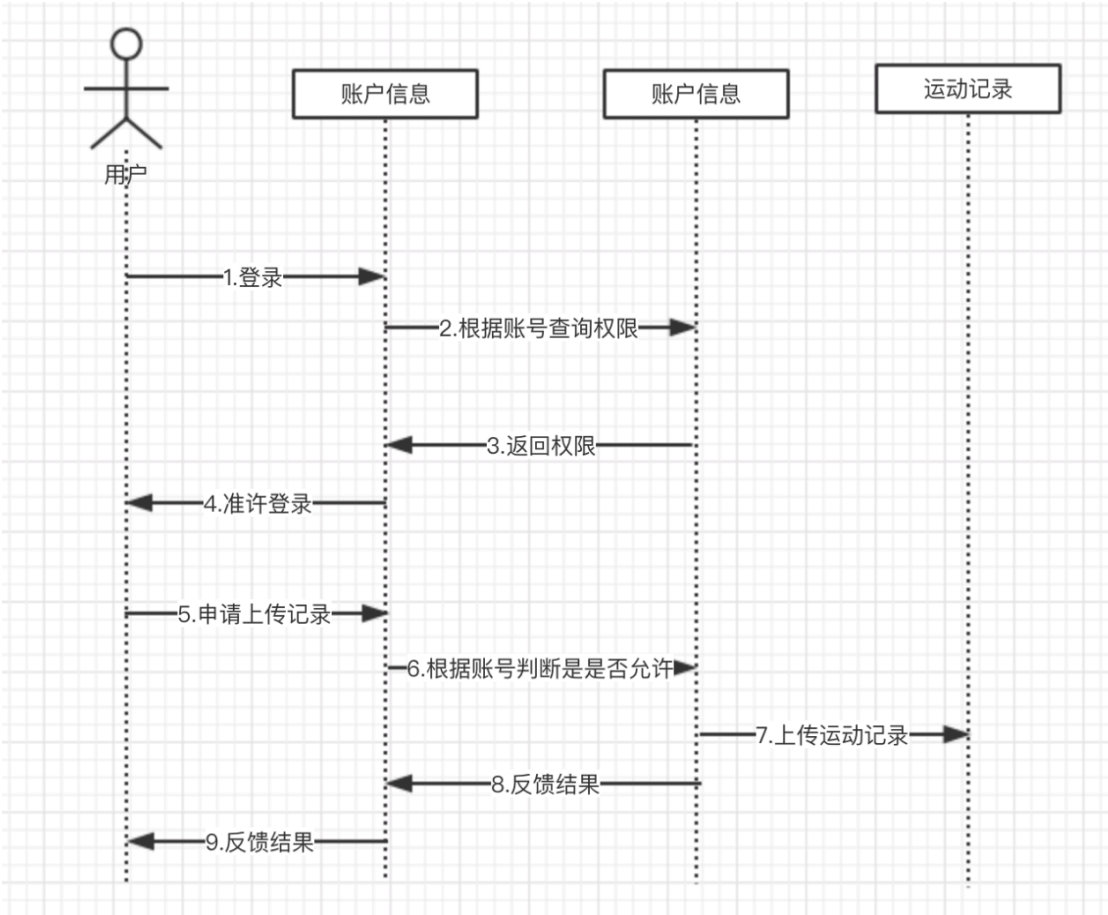


2.2.6 修改权限用例图 #5

系统管理员修改权限



2.2.6 上传运动记录用例图 #5



2.3 各相关方对体系结构的要求

学生平时运动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。

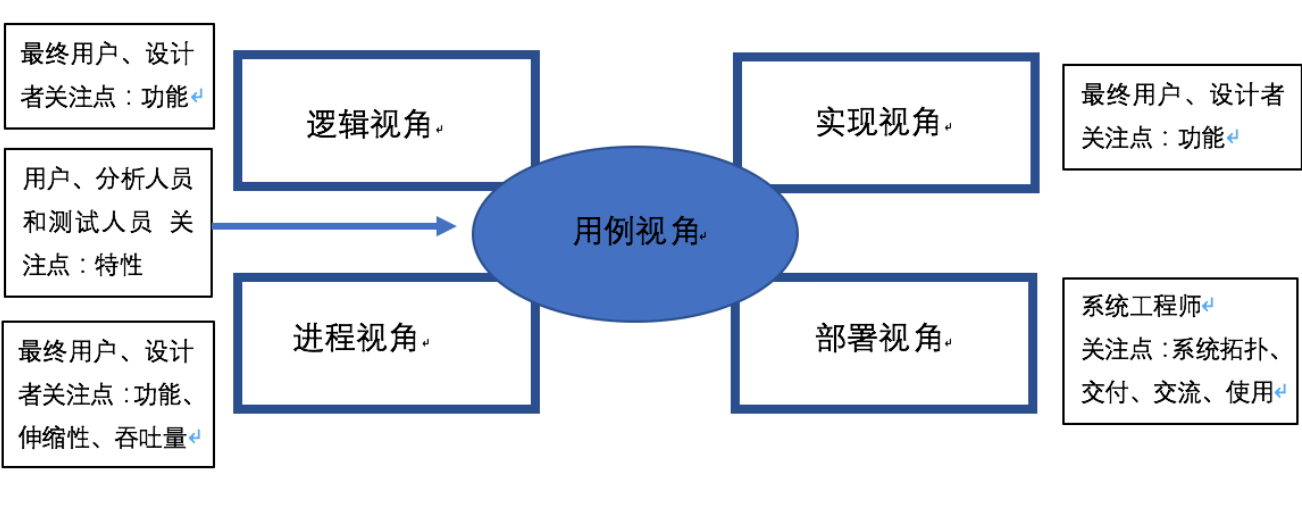


图 2-2 “4+1” 视图模型

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。要求本系统供给给学生、体育老师、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、便捷的学生每日步数上传，成绩录入、成绩查看、成绩管理功能、成绩审核功能，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性能上的要求，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

学生平时运动成绩管理系统服务于学生、体育老师以及教务，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。



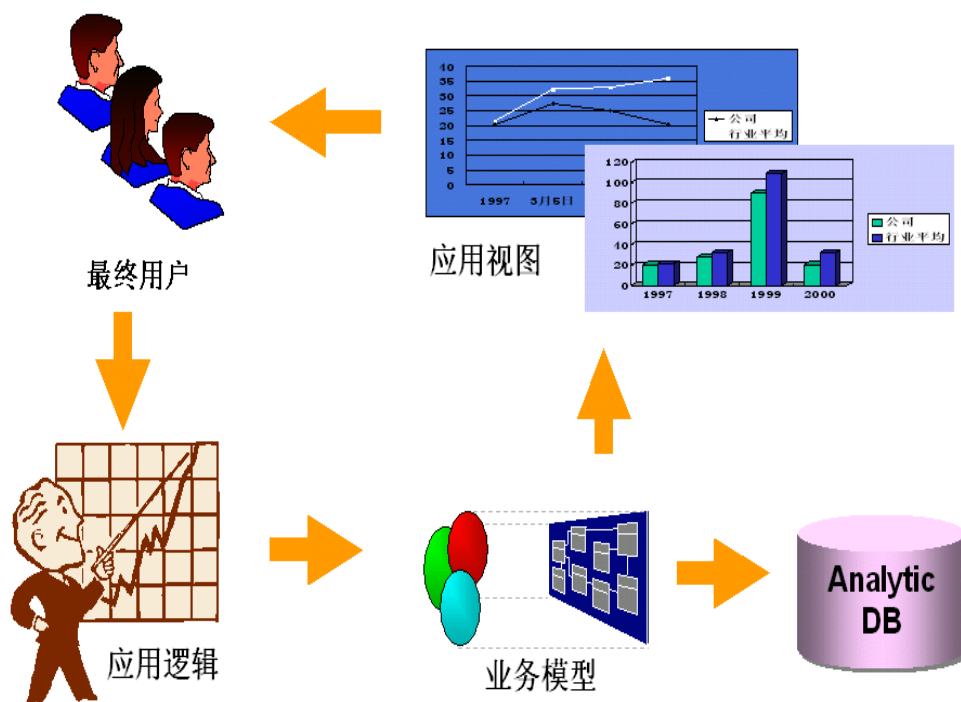


图 2-3 设计约束图

### 2.4.3 其他约束

本系统的最后交付日期不能超过 2018 年 9 月 16 日；系统基于 B/S 架构开发，必须可以在 Windows 操作系统的 IE 浏览器和 Mac OS 操作系统上的 Safari 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，并且给出相应的运行时提示。
- 本系统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。

## 2.5.6 性能

- 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。
- 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 10TB，其中应该有至少 300GB 的空闲空间，用于信息的扩充。

## 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。

## 3. 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。

本系统采用 B/S 架构的原因如下：

(1) B/S 最大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，由系统管理员分配一个用户名和密码就可以使用。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。B/S 这种优点非常符合本系统的要求，因此采用 B/S 架构进行系统的体系构建。

(2) B/S 架构建立在广域网之上的，不必是专门的网络硬件环境。例与电话上网，租用设备。信息自己管理，比 C/S 更强的适应范围，一般只要有操作系统和浏览器就行。因此本系统客户端采用浏览器进行访问的时候不用考虑网络因素的限制，保证了信息的反馈速度。

(3) 系统维护方便，构件组成方便构件个别的更换，实现系统的无缝升级。系统维护开销减到最小，用户从网上自己下载安装就可以实现升级。

(4) 建立在浏览器上，有更加丰富和生动的表现方式与用户交流，并且大部分难度减低,减低开发成本。

(5) 对安全以及访问速度的多重的考虑，建立在需要更加优化的基础之上。比 C/S 有更高的要求，B/S 结构的程序架构是发展的趋势，全面支持网络的构件搭建的系统，SUN 和 IBM 推的 JavaBean 构件技术等使 B/S 更加成熟。

3.3 结构化视图

MVC 体系架构主要体现在学生平时运动成绩管理系统，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端实现的。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、体育老师、体育教务教务等进行学生平时运动成绩管理查询使用，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（实现这部分可以调用学校校园网密码服务的 API）。

信息检索和修改层主要针对的是成绩查询、历史步数查询、平时运动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以及查询到自己历史步数信息。；体育老师可以查看到自己所教授的所有学生的考试成绩、学生平时运动的步数信息以及学生平时运动的成绩。

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。



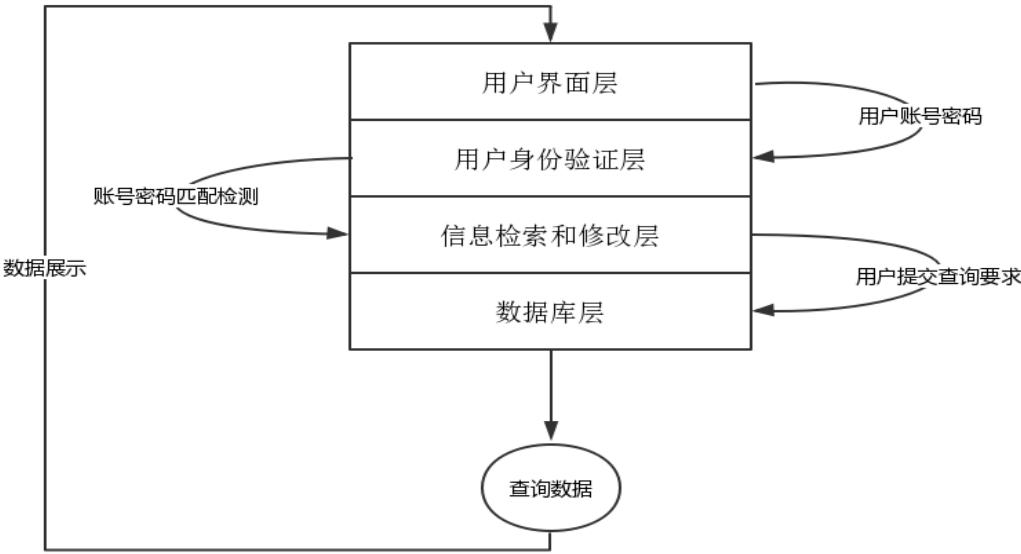
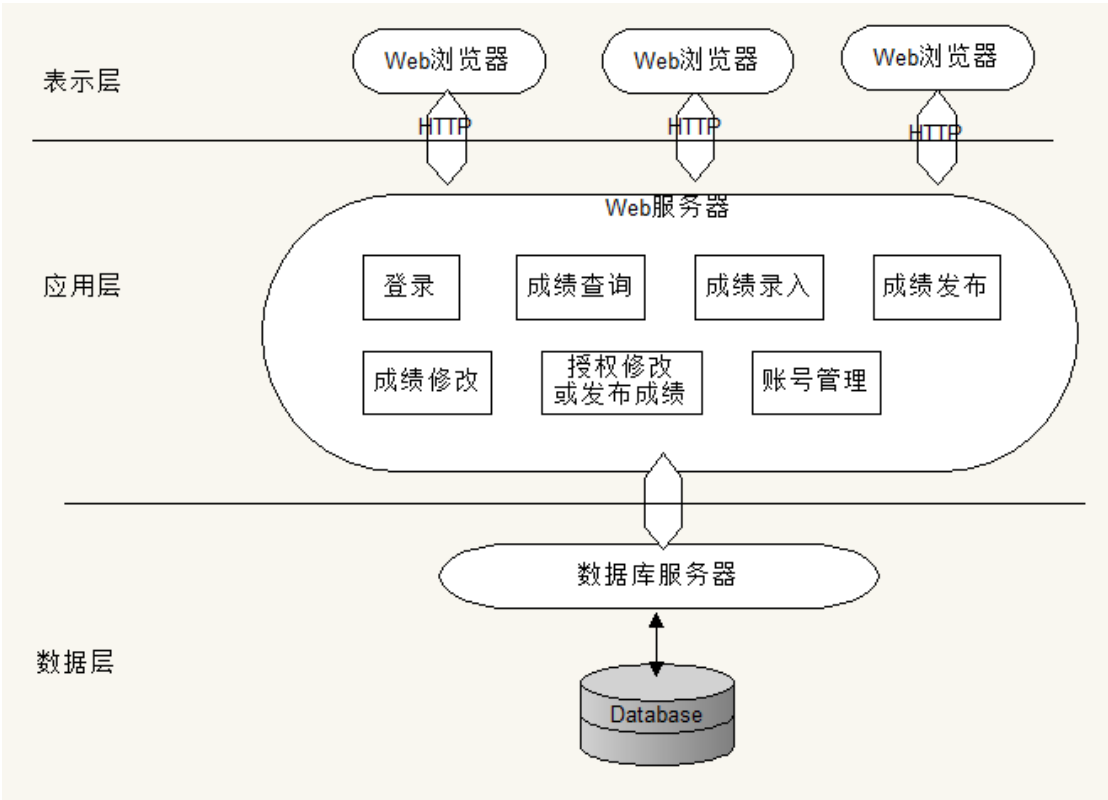
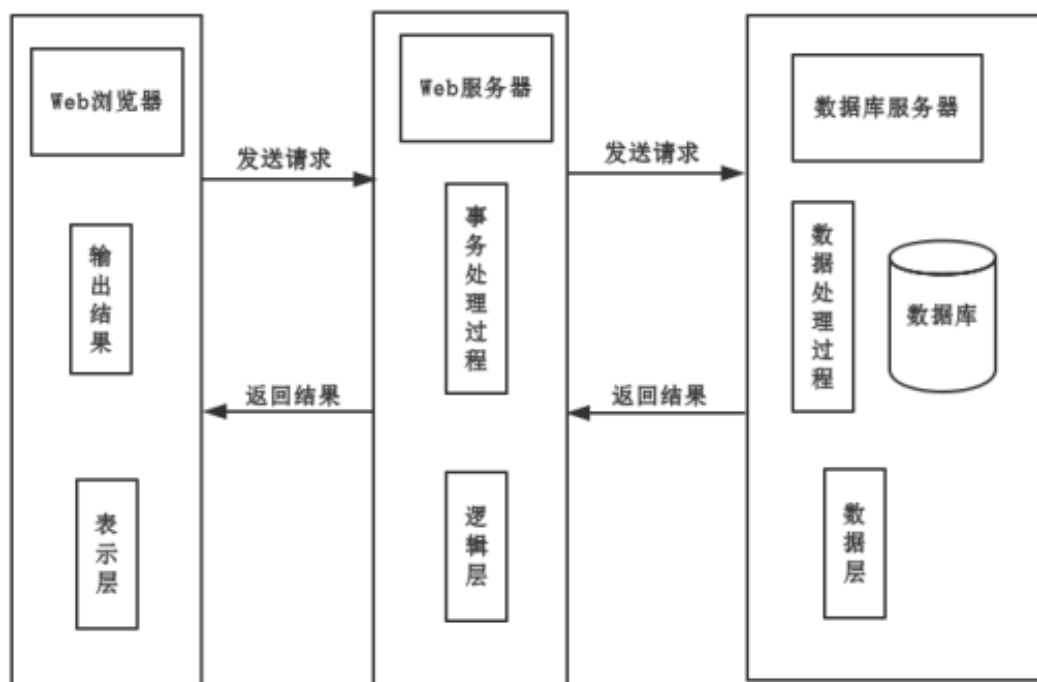


图 3-1 MVC 架构图





### 3.4 行为视图

本系统具有三种主要的行为：学生体育成绩成绩管理行为，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。

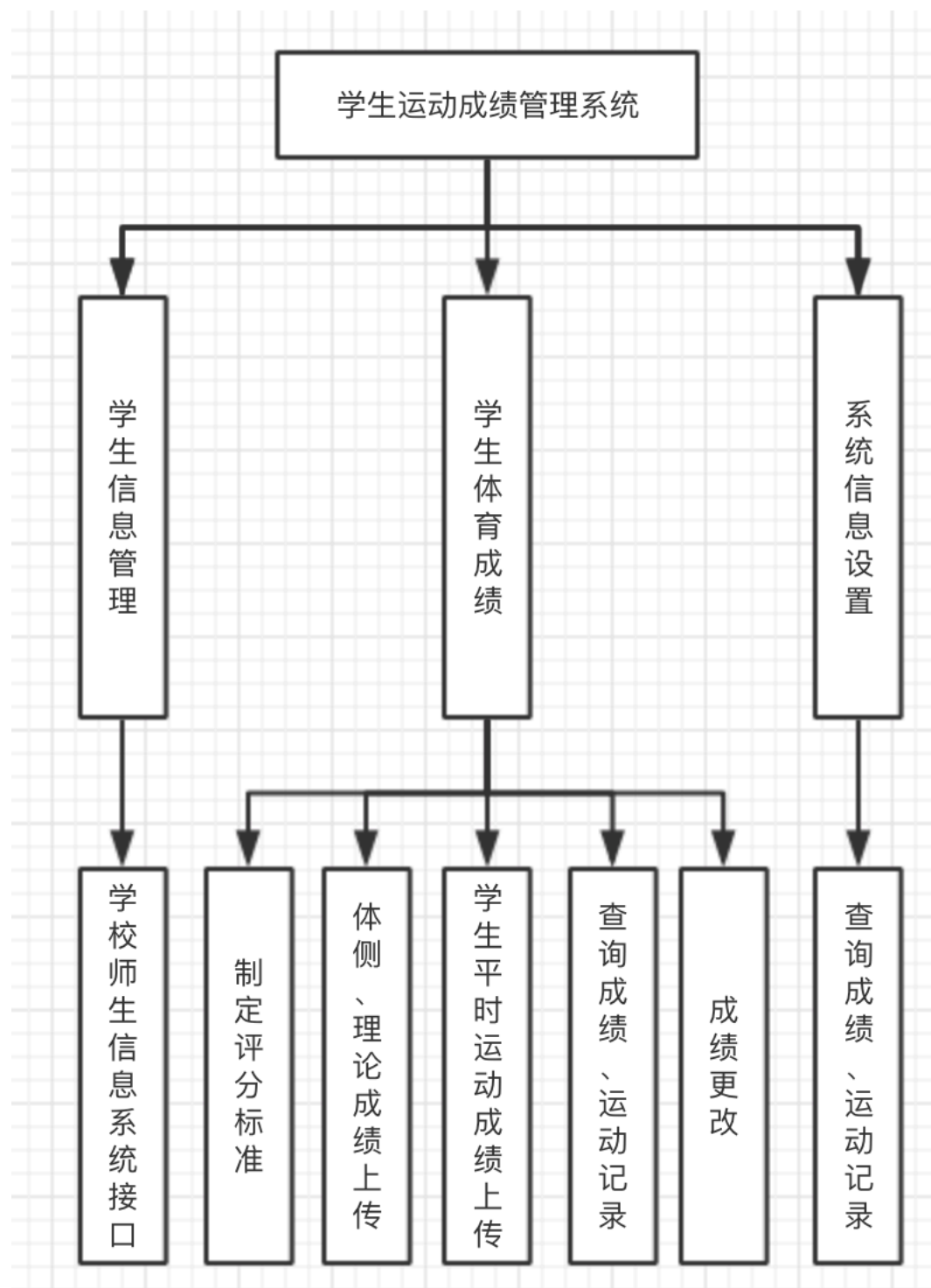


图 3-2 系统行为图

### 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版布局的制约、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。

## 4. 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：**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、Microsoft Phone、BlackBerry 系统。

**增长性场景 2：**改进系统的可用性，提升到 99%。

**增长性场景 3：**添加日历、笔记本、天气小部件功能，方便使用者使用。

**增长性场景 4：**在 IE5 以上的浏览器上都支持系统的运行访问。

**增长性场景 5：**未来可能会需要更改用户的 UI 界面，这只需要更改表示层的逻辑即可，而不需要更改业务逻辑层和数据层的代码，希望系统仅需增加 2 人周的工作量就能完成修改。

**增长性场景 6：**未来可能会增加新的数据服务器，这将会降低用户的 Web 访问时间，希望系统仅需增加 1 人周的工作量就能完成修改。

**增长性场景 7：**通过扩充现有数据库表的规模和在特定的列上创建索引，可以降低平均检索时间到 1s 以内。

### 4.1.3 探索性场景分析

**探索性场景 1：**系统可以和学生成绩管理系统对接，自动植入学生已选课程，并且自动在老师界面显示所有选课学生。不超过 5 个人月的工作量。

**探索性场景 2：**系统可以和学生考勤系统对接，自动生成对应课程的考勤分数。不超过 5 个人月的工作量。

**探索性场景 3：**正常情况下，当服务器有一半宕机时，不影响整个系统的可用性。

**探索性场景 4：**当系统的数据库出现问题时，可以通过日志文件进行恢复，损失的数据不超过 5 分钟。

**探索性场景 5：**当系统的数据库出现问题时，可以通过调用在学生成绩管理系统中的体育成绩备份数据库。

## 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统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。

本系统可能含有的风险如下：

- 学生查看自己成绩时，若体育同时更改该学生的成绩，可能导致成绩更改不成功或产生行为冲突。
- 大批量录入人员信息或学生成绩时，可能导致系统因响应时间过长而崩溃。
- 数据库和服务器信息可能不一致。
- 教务员修改或删除正处在登录状态的人员的信息时，可能导致操作不成功或产生行为冲突。
- 批量导入学生成绩时，学生查看的信息可能与数据库中成绩不对应。
- 大量用户同时登陆系统时，系统可能因为延迟时间过长而崩溃。
- 两个教务同时更改或删除某一人员或同一批次成员的信息时，可能导致操作不成功或产生行为冲突。
- 两个教务同时发布同一批次成绩时，可能导致操作不成功或产生行为冲突。