

学生考试成绩管理系统

体系结构设计说明书

1.0

2015.12.19

孔丹晨

修订记录

日期	版本	作者	修改描述
2015-12-19	1.0	孔丹晨	第一版

文档批准

签名	姓名	题目	日期

修订记录.....II

文档批准.....II

1. 引言1

 1.1 编写目的.....1

 1.2 文档范围.....1

 1.3 定义、术语和缩写语.....1

 1.4 参考文献.....2

 1.5 文档概述.....2

2. 体系结构需求2

 2.1 产品概述.....2

 2.2 体系结构用例.....2

 2.3 各相关方对体系结构的要求.....3

 2.4 约束条件.....3

 2.5 非功能需求.....3

3. 设计方案4

 3.1 相关体系结构模式.....4

 3.1.1 C/S 体系架构.....4

 3.1.2 B/S 体系架构.....5

 3.1.3 两种体系结构模式的比较.....6

 3.2 体系结构概述.....7

 3.3 结构化视图.....7

 3.3.1 系统概念级体系结构设计.....7

 3.3.2 系统模块级体系结构设计.....8

 3.3.3 系统运行级体系结构设计.....10

 3.4 行为视图.....17

 3.4.1 逻辑视角.....18

 3.4.2 进程视角.....18

 3.4.3 实现视角.....21

 3.4.4 部署视角.....23

 3.4.5 用例视角.....23

 3.5 实现问题.....25

 3.5.1 用户增加.....25

 3.5.3 质量属性.....27

4. 系统的质量分析和评价27

 4.1 场景分析.....27

 4.1.1 用例场景.....28

 4.1.2 增长性场景.....28

 4.1.3 探索性场景.....29

表 1 定义与术语.....	1
表 2. 缩写语.....	1
表 3.体系结构用例.....	3
表 4. 质量或可信赖属性.....	4
表 5. 用户登录界面类.....	11
表 6. 用户权限操作类.....	11
表 7.用户权限数据库操作类.....	11
表 8.用户密码修改界面类.....	12
表 9.用户密码修改类.....	12
表 10.成绩录入界面类.....	13
表 11.成绩录入类.....	13
表 12.成绩操作数据库类.....	14
表 13.成绩查看界面类.....	14
表 14.成绩查看类.....	15
表 15.成绩修改界面类.....	15
表 16.成绩修改类.....	16
表 17. 管理用户界面类.....	17
表 18.管理用户类.....	17
表 19.用户数据库操作类.....	17
表 20.B/S 架构对质量属性的影响.....	27
表 21.用例场景.....	28

图 1. C/S 体系架构及其工作过程.....	4
图 2 . B/S 体系架构模式及其工作过程.....	5
图 3.系统总体架构图.....	8
图 4. 登录模块.....	9
图 5.成绩管理模块.....	9
图 6. 用户信息管理模块.....	9
图 7.登录功能消息序列图.....	10
图 8.修改密码消息序列图.....	12
图 9.录入成绩消息序列图.....	13
图 10.查看成绩消息界面图.....	14
图 11.修改成绩消息序列图.....	15
图 12.用户管理消息序列图.....	16
图 13."4+1"视图模型.....	17
图 14. 系统逻辑视角.....	18
图 15.学生进程.....	19
图 16.教师进程.....	20
图 17. 教务员进程.....	21
图 18. 架构开发视图.....	22
图 19. 系统总包图.....	22
图 20. 系统部署视图.....	23
图 21.学生用例视图.....	24
图 22. 教师用例视图.....	24
图 23. 教务员用例视图.....	25
图 24.响应时间与吞吐量的关系.....	26

1. 引言

1.1 编写目的

本说明书的编写目的是描述《学生考试成绩管理系统》的架构设计方案，包括系统的软件总体架构设计及使用的框架说明，以及基于该架构的开发流程，以便于软件设计人员、软件开发人员和整合测试人员了解软件体系结构、具体设计方案，为其系统功能实现提供构造体系，提供一定的直接性结构说明。

本文的预期读者为：

- 1. 设计人员：对需求进行分析，并设计出系统，包括数据库的设计。
- 2. 开发人员：了解系统功能，据此进行项目开发。
- 3. 用户：了解预期产品的功能和性能，并与分析人员一起对需求进行讨论和协商。
- 4. 测试人员：编写测试用例，对软件产品进行功能性测试和非功能性测试。
- 5. 老师：对此软件体系结构说明书的质量做出评价。

1.2 文档范围

本文档的名称为：学生考试成绩管理系统。

本系统仅适用于学校对学生成绩与基本信息进行管理，并不包含学生选课，学生学籍管理，考试安排等其他管理信息。其中学生成绩仅包括期末成绩，基本信息只包括用户的姓名与密码，并不包含其它信息。

凡是在《学生考试成绩管理系统软件体系结构说明书》中出现的内容，本文档只提供相应的引用，具体内容不再赘述。

1.3 定义、术语和缩写语

表 1 列出了本文档使用到的全部定义、术语，表 1 列出了本文档使用到的全部缩写语。

表 1 定义与术语

名称	名词解释
学生	使用本系统的在校学生
教师	使用本系统的在职教师
教务员	使用本系统的学校教务员

表 2. 缩写语

缩写	名词解释
MIS	Management Information System 管理信息系统 主要由计算机相关软硬件设备组成的能够供人们收集、存贮、传递和加工各种信息的系统。
E-R 图	Entity Relationship Diagram 实体-联系图 提供了表示实体类型、属性和联系的方法，用来描述现实世界的概念模型。
UML	Unified Modeling Language 统一建模语言或标准建模语言，为软件开发的所

	有阶段提供模型化和可视化支持，包括由需求分析到规格，到构造和配置。
SDL	Specification and Description Language 规格与描述性语言
B/S	Browser/Server 浏览器/服务器
C/S	Client/Server 客户端/服务器

1.4 参考文献

[1]王安生.软件工程化[M].北京:清华大学出版社.2014.p164-p211.

1.5 文档概述

本文主要针对学生考试成绩管理系统的软件结构进行设计，得出了学生考试成绩管理系统的体系结构说明书，主要内容包括四部分，具体如下：

第一部分主要介绍了本文档的编写目的、使用范围以及本文使用到的定义、术语与缩写词，同时也列出了本文的参考文献。

第二部分主要对本体系结构进行了总体的概述，简述了体系结构的需求以及用例，并对本系统的约束与假设和依赖做了简单陈述。

第三部分主要介绍了对本系统的体系结构进行了分级设计。对相关的体系结构进行了概述，并用结构化的视图描绘了概念级、模块级、以及运行级的体系结构设计，同时本系统还使用了“4+1”体系结构模型对体系结构系统进行了描绘。

第四部分对系统进行了质量分析与评价。主要使用了场景分析的方式，描绘了用例场景，增长性场景和探索性场景；并使用该场景进行手工测试。

2. 体系结构需求

2.1 产品概述

学生成绩信息管理主要是由教务处对学生成绩信息进行管理，以方便学生成绩信息的即时交互。为了方便用户的使用以及保证用户的隐私，本系统设定了教务员、教师、学生三种用户权限并加入了登录系统来避免用户的越权操作。基本信息由教务员维护，学生和教师登录系统后根据各自的相应权限进行操作。主要实现管理员管理功能，包括学生管理，教师管理，课程管理；以及成绩管理功能，教师录入成绩信息，经过教务员审核后发布给学生用户查看。

2.2 体系结构用例

本说明书的用例是从使用者的角度出发，描述用户所期望的与整个运行系统的交互[1]。根据《软件需求规格说明书》对本系统用户需求的分析以及对用例角色的陈述，本体系结构说明书将用户的需求与系统进行交互，设计体系结构用例如表 3. 详细用例图以及用例描述见本系统的《学生考试成绩管理系统软件需求规格说明书》。

表 3.体系结构用例

角色	描述
教务员	<ol style="list-style-type: none"> 1. 所有查询成绩功能。 2. 对某一科成绩进行统计。 3. 审核成绩，即审核教师提交的成绩，确认无误后进行发布。 4. 发布成绩。 5. 授权教师，允许教师对成绩进行修改。 6. 回收授权信息，不允许教师对成绩进行修改。 7. 用户管理，包括对学生，教师，科目信息的增删改。 8. 数据库管理。 9. 修改自己的密码。
教师	<ol style="list-style-type: none"> 1. 查询某一学生的所有成绩。 2. 查询某一班级的成绩。 3. 录入成绩和修改成绩。 4. 修改自己的密码。
学生	<ol style="list-style-type: none"> 1. 查询自己的成绩。 2. 修改自己的密码。

2.3 各相关方对体系结构的要求

目前，随着大规模的网络技术迅速的发展使用，在人们日常的工作、生活过程中，产生了海量的数据，为了能够让人们有效的利用这些数据，本地数据库已经无法满足人们的多种多样的需求，因此，在工作、学习和生活过程中，必须开发基于网络的分布式应用系统，提高海量数据信息的使用效益。分布式体系架构是实现分布式应用系统的基本，为了能够使得分布式应用系统实现高性能访问和处理，诸多科学工作者对其进行了研究。

2.4 约束条件

学校有 3 万学生，2000 位老师，每学期开出 500 门课，平均一门课有 100 个学生。学生的成绩仅包含期末成绩。

2.5 非功能需求

表 4 给出了质量和可信赖特征对体系结构的约束，该表中易用性要求决定了系统必须采用 Browser/Server 的结构，而不能是 Client/Server 的结构[1]。

表 4. 质量或可信赖属性

质量或可信赖性属性	体系结构需求
性能	对 90%的请求，系统必须在 3 秒以内做出响应。
资源管理	服务器部件的内存必须有 40%以上的预留量。
易用性	客户端必须支持浏览器，兼容 IE6.0，以保证远程客户不需要安装软件。
可伸缩性	系统在高峰时期能处理 30000 个并发用户。
可修改性	体系结构必须支持从当前的第四代语言向.NET 或 J2EE 环境的过度。
密安性	所有通信必须被授权，并使用认证过程进行加密。
可适用性	系统必须 24×365 运行，可使用性达到 99%以上。
可靠性	不允许丢失信息，所有信息必须在 30s 内完成。

3. 设计方案

3.1 相关体系结构模式

3.1.1 C/S 体系架构

C/S 体系架构是基于网络的分布式应用系统采用的第一个系统架构，其组成部分包括两个，分别是客户机(Client)和服务端(Server)。客户机为用户提供访问系统的接口，负责分布式应用系统的前台输入和输出，向服务器发送用户的逻辑业务请求，向用户反馈服务器响应的请求结果;服务器端由 Web 服务器和数据库服务器共同构成，其主要完成客户端发送过来的逻辑业务请求和数据业务请求，并且将二者的处理结果反馈给客户端浏览器，以使用户能够浏览。基于 C/S 体系架构的分布式应用系统将用户端和服务端分为独立的两个部分，不同部分完成各自应该完成的功能，二者之间的通信使用相关的接口和协议实现功能需求。因此，对于用户来讲，每一个请求都由二者有机配合，共同完成，对于用户每一个处理过程都是透明的，无感知度，即使是用户的需求需要在很多台机器上共同协作完成，用户也会感到仅仅是在一台机器操作是一样的。

根据上述内容，分布式应用系统的 C/S 体系架构如图 1。

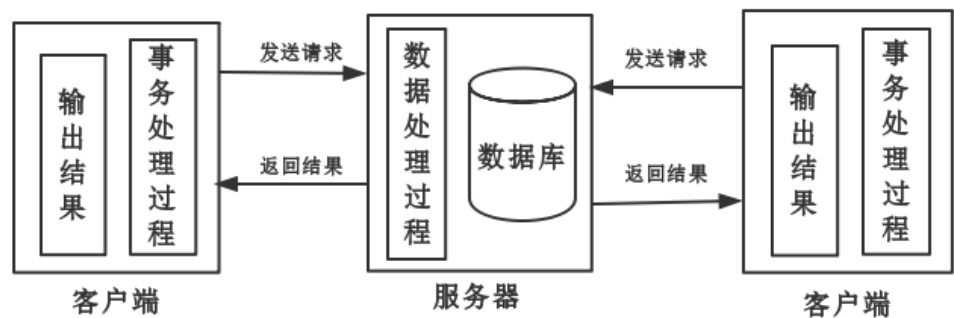


图 1. C/S 体系架构及其工作过程

3.1.2 B/S 体系架构

随着计算机技术、网络技术和数据库技术的迅猛发展，使用网络进行工作、生活和学习的人群越来越多，由于 C/S 体系架构设计的分布式应用系统安装负责、承载服务量较低，导致大规模的人群访问同一个系统时延时越来越长，满意度直线下降。为了满足海量人群应用系统的需求出现了 B/S 架构。B/S 体系架构由 Web 浏览器层(表示层)、Web 服务器(逻辑业务服务器层)和数据库服务器层有机构成，每一个层次都能够独立的完成相关工作，如果需要下层支撑，其将请求发送给下一层，如果不需要，直接反馈结果给上一层即可。鉴于 B/S 体系架构的客户端配置简单，升级优化也较为容易，非常受人们的欢迎，其是计算机知识相对较少的人们，已经在分布式应用系统中广泛的使用。B/S 体系架构如图 2-2 所示。

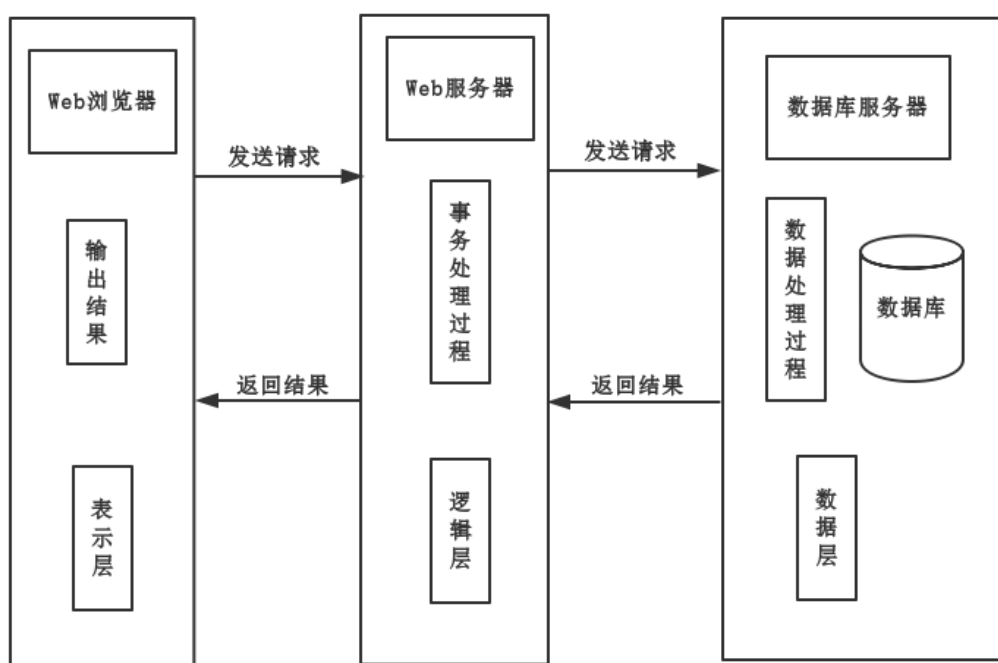


图 2. B/S 体系架构模式及其工作过程

B/S 体系架构的工作过程如图 2-2 所示，B/S 体系架构包括表示层、逻辑业务处理层和数据层，层次之间采用相关的通信接口和协议发送和反馈相关需求内容。

1. 表示层

表示层位于用户客户端，该层的相关功能是接受用户从客户端浏览器发送的业务请求，将相关请求发送到逻辑业务处理层，并将逻辑业务处理层反馈的处理结果输出到浏览器上，供用户浏览。B/S 体系架构中的客户端使用人们常用的 Web 浏览器作为载体，输入系统服务器所在网址，自动弹出下载插件的通知，将其装在浏览器中即可供用户使用，无需下载安装程序，减少了客户端应用程序安装、运营更新的需求，大大的降低了网络管理员的工作量。

2. 逻辑业务处理层(Web 服务器层)

逻辑业务处理层位于 B/S 体系架构中的 Web 服务器上，也被很多场合成为 Web 服务器层。逻辑业务处理层的相关功能是完成系统客户端发送过来的用户请求，由 Web 服务

器分配处理任务，执行用户请求的逻辑业务，同时并根据用户的需求检测其是否需要数据处理，如果需要数据处理，则将处理结果的请求发送到数据库服务器上，由数据库服务器完成余下工作，同时将数据库服务器处理的结果反馈到客户端，供用户浏览。

3. 数据层(数据库服务器层)

数据层位于数据库服务器上，通常又被称为数据库服务器层，其相关功能是处理逻辑业务处理层发送过来的数据业务请求，该层可以使用数据库管理系统的相关操作，实现据请求的添加、删除、修改等操作，同时将数据处理结果及时的反馈到逻辑业务处理层。

3.1.3 两种体系结构模式的比较

随着网络规模的增大，B/S 体系架构应用的越来越广泛，与 C/S 体系架构相比较，B/S 体系架构具有以下三个方面的优势：

➤ B/S 体系架构在系统性能方面表现更好。

在分布式应用系统开发过程中，C/S 体系架构通常采用内部一致的标准即可，由于其应用系统主要是基于局域网开发，与其相比，B/S 体系架构更具开放性，基于 B/S 体系架构开发的系统主要是在互联网上进行应用，因此，该体系架构标准是经过标准化组织制定，行业共同遵循的。因此，B/S 体系架构具有通用性和跨平台性，同时，B/S 体系架构开发的应用系统快捷方便、灵活性非常强。用户只要能够通过浏览器上网，就可以顺利的使用基于 B/S 体系架构开发的分布式应用系统。基于 B/S 体系架构开发的分布式系统客户端比较简单，只需要完成一些简单的浏览、数据输入等功能，使得用户使用时更加方便。总之，综合比较各个方面，B/S 体系架构的综合性能是高于 C/S 体系架构的。

➤ 在系统运行维护和升级方面。

基于 B/s 体系架构开发的应用系统在运行维护、升级方面具有较低的人力物力成本。B/s 体系架构开发的系统在升级时，仅仅需要下载插件，在浏览器上安装就行，无需对客户端做任何改动，而基于 C/S 体系架构开发的系统则不然，在系统进行升级时，需要开发专门的客户端应用软件与其相互匹配，而且需要在每一台终端上都需要安装新的客户端程序，无论是财力、人力都有极大的耗费，同时也极大的浪费时间。同时，B/S 体系架构模式具有三个独立的层次，层与层之间具有较好的独立性，因此，任何一个模块改变时都不会对其他模块造成过多的影响，因此，在升级系统时，只需要对其中相关的模块升级即可，无需改变其他模块，但是在 C/S 体系架构中，各个模块之间的耦合性较强，升级时无法独立进行，影响全局。

➤ 在系统开发方面。

B/S 体系架构是随着网络用户规模的增多发展起来的一种改进的 C/S 架构模式，属于新兴的技术。随着在分布式应用系统开发过程中的推广使用，已经得到了广泛的应用，因此，B/S 体系架构具有更好的发展前景和拓展性，使得其在分布式应用系统开发过程越来越抢手。并且，基于 B/S 模式的应用系统开发简单易学，使得开发人员很快就可以掌握，所以，系统开发人员也倾向于使用 B/S 体系架构模式。

总而言之，与 C/S 体系架构相比，B/S 体系架构应用范围更加广泛，其响应速度和安全性也比较高，系统升级维护过程简单，不繁琐，同时可以降低系统运行维护和升级成本，因此，在证券销售管理和业务统计系统开发过程中，由于许使用此系统的操作人员不是计算机专业人员，包括很多经济学、金融专业人员，因此，为了便于安装、运行和升级，同时提高可操作性，我们选择 B/S 体系架构作为系统开发架构。

3.2 体系结构概述

学生成绩信息管理系统是一个非常复杂的分布式应用系统，从《软件需求规格说明书》中我们得知，该系统涉及的组成部分包括成绩信息管理子系统、登录子系统和信息管理子系统等多个子系统，这些子系统涉及的部门非常广泛，不但包括教务处各个院系的相关部门，涉及的用户包括教师、学生和学校的行政管理人员等，因此，每一个子系统之间，都存在着多种关系，信息流互相通过对方，因此，成绩信息管理系统通过系统之间存在的服务，有机构成了的子系统共同实现了系统的功能。

3.3 结构化视图

3.3.1 系统概念级体系结构设计

本系统采用了分层的设计模式对系统进行设计，图 3 即为系统的总体架构图设计。

客户层:即为需要安装 Web 浏览器的客户端位置，用户在该层实现各自的功能操作，本系统中的操作用户包括: 教务员用户、教师用户、学生用户和系统管理员用户。

表示层:即是为教师用户等系统用户提供学生成绩管理系统的功能操作后显示出来的操作界面，教师用户等系统用户可以在该层的操作界面中输入要查询管理学生的考试成绩信息以及相关的内容，能够实现对成绩的录入、成绩的维护以及成绩信息的查询操作等管理功能。

业务逻辑层与数据层:该层主要负责提供数据信息的交换以及共享功能，能够对不同数据格式以及信息实现共享和交换。

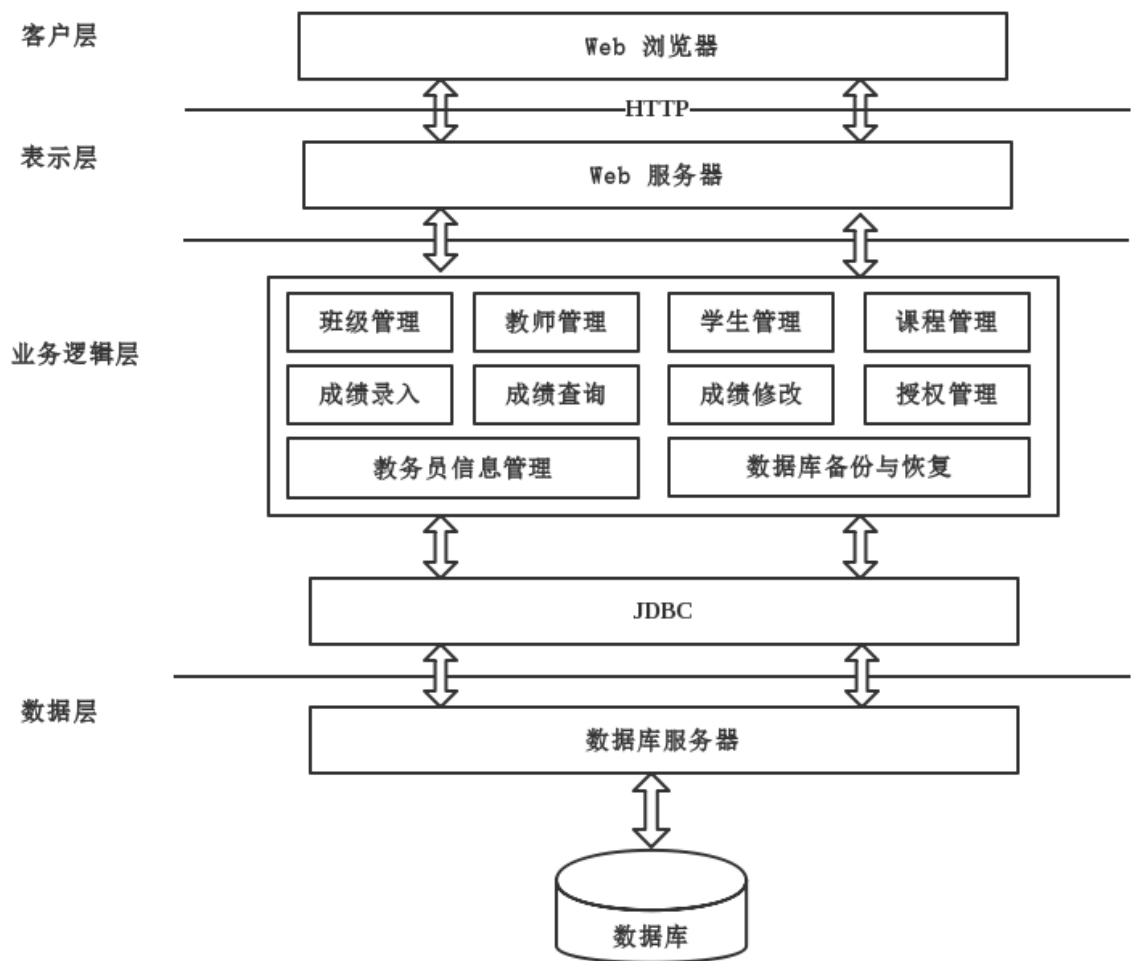


图 3.系统总体架构图

3.3.2 系统模块级体系结构设计

系统的模块化设计是系统进行复用的关键。模块级体系结构反映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。模块化表达有两种方式：

- 将系统按功能从逻辑上分解为系统、模块以及程序单元；
- 按系统的层次进行划分。

本说明书根据《学生成绩管理系统需求规格说明书》将系统按功能从逻辑上分解，细化功能结构模块。

3.3.2.1 登录模块

登录系统主要为学生、教师、教务员用户提供登录功能，功能结构图如图 4 所示，用户在浏览器中输入账号和密码之后，浏览器会读取用户输入的信息，并将数据信息传送到服务器。服务器将输入数据与后台数据库内容进行比对来查看用户名与密码是否匹配，将结果传送给浏览器向用户输出。

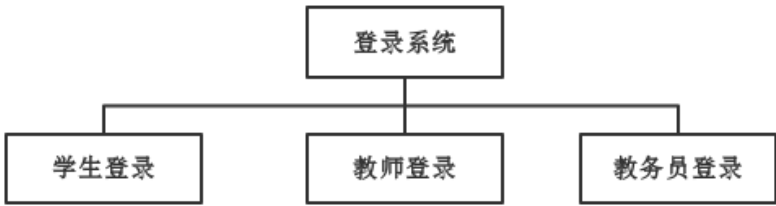


图 4. 登录模块

3.3.2.2 成绩管理系统模块

成绩管理功能主要完成系统管理员、教师和对成绩信息的管理工作，功能结构图如图 5 所示。用户选择要查询的信息，浏览器获取用户选择，并请求服务器响应。服务器将处理数据，并将结果传送给浏览器，浏览器通过该数据响应用户请求。



图 5. 成绩管理模块

3.3.2.3 用户信息管理模块

信息管理模块主要是保存学生、教师的个人信息。该模块的主要功能是实现用户信息的修改，功能结构图如图 6 所示。浏览器读入用户要修改的信息并将数据与请求发送给服务器，服务器根据浏览器传送的信息更新数据库，并返回结果。

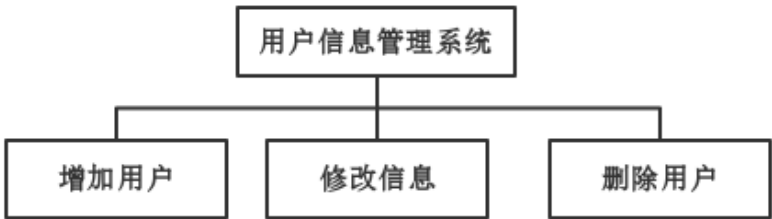


图 6. 用户信息管理模块

3.3.2.4 模块之间关系分析

➤ 登录模块和成绩管理模块的关系。

成绩管理模块中的功能一定在权限管理模块认可的情况下才能执行。即只有当用户成功登陆系统后，才能使用成绩管理模块中提供的功能。

➤ 登录模块和用户信息管理模块的关系。

用户信息管理模块中的功能一定在登录模块认可的情况下才能执行。即只有当用户成功登录系统后，才能使用用户信息管理模块中提供的功能。

➤ 用户信息管理模块和成绩管理模块的关系。

成绩管理模块中的成绩信息一定是在用户信息管理功能提供学生信息的情况下才能执行。

3.3.3 系统运行级体系结构设计

在运行级体系结构视图中，将模块指派到特定的运行映像，说明模块怎样组合为特定的产品。运行级体系结构是描述系统的动态结构。它主要用于系统的性能和可调度性进行分析，系统的静态和动态配置管理，向不同执行环境的移植等。UML 类图不能很好地表示系统的动态行为[1]。

因此本部分针对运行级的体系结构使用的结构化视图为：

- 用 UML 消息序列图表示配置项的动态行为或配置项之间的迁移。
- 用 UML 类图表示静态配置。

由于在《学生考试成绩管理系统需求规格说明书》中已经对本系统的类结构进行了概念级的设计，因此本部分不进行赘述，直接对消息序列图中使用到的类进行运行级的详细描述。

3.3.3.1 登录功能消息序列图

登录功能主要负责判定用户是否有权进入系统，因此登录功能的消息序列图如图 7.

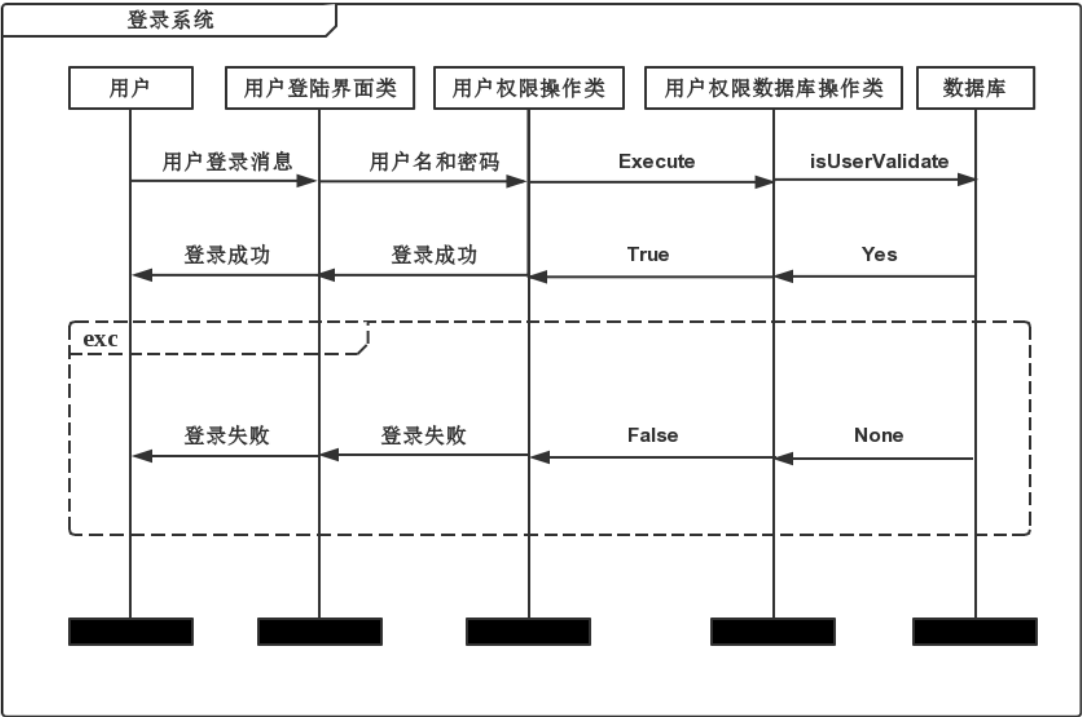


图 7.登录功能消息序列图

其中该功能使用到的类的详细描述见表 5-7:

表 5. 用户登录界面类

类名	用户登录界面类
种类	界面类
方法:	
1.	提供用户输入用户名和密码。
2.	如果用户名或密码错误, 给出提示信息。

表 6. 用户权限操作类

类名	用户权限操作类
种类	操作类
方法:	
ActionForward execute(ActionMapping mapping, ActionForm form, HripServletRequest request, HttpServletResponse response)[检验用户是否具有进入系统的权限]	

表 7. 用户权限数据库操作类

类名	用户权限数据库操作类
种类	操作类
方法:	
Boolean isUserValidate(String id, String pwd, int userRight)[检验用户合法]	
DBUser getUser(String id)[根据用户 ID 取得用户信息]	
Boolean updateUserPwd(String id, String pwd, int userRight) [更新用户密码信息]	
int getUsertype(String id) [取得用户的类型]	

3.3.3.2 修改密码消息序列图

修改密码的消息序列图如图 8，该功能主要进行用户的密码修改。

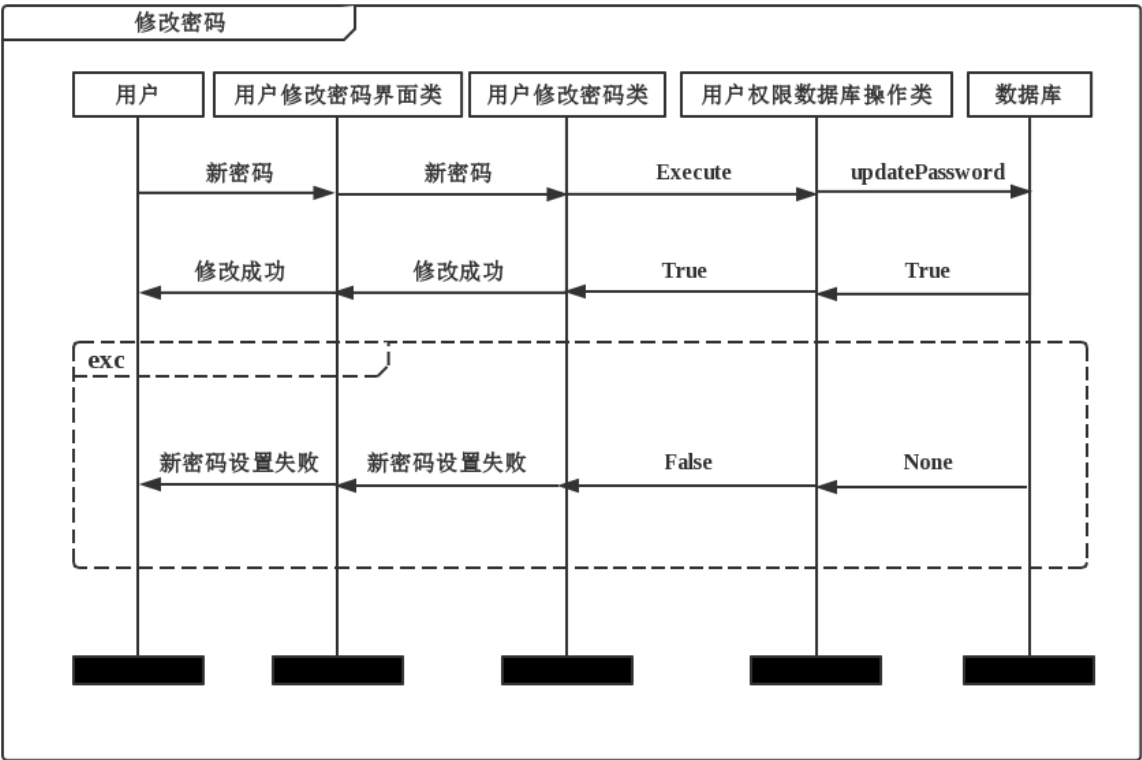


图 8.修改密码消息序列图

其中该功能使用到的类的详细描述见表 8-9：

表 8.用户密码修改界面类

类名	用户密码修改界面类
种类	界面类
方法：	要求用户输入密码与输入确认密码。

表 9.用户密码修改类

类名	用户密码修改类
种类	操作类
方法：	ActionForward execute(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)[检验用户是否成功设置密码]

3.3.3.2 录入成绩消息序列图

录入成绩的消息序列图如图 9，该功能主要进行学生成绩的录入。

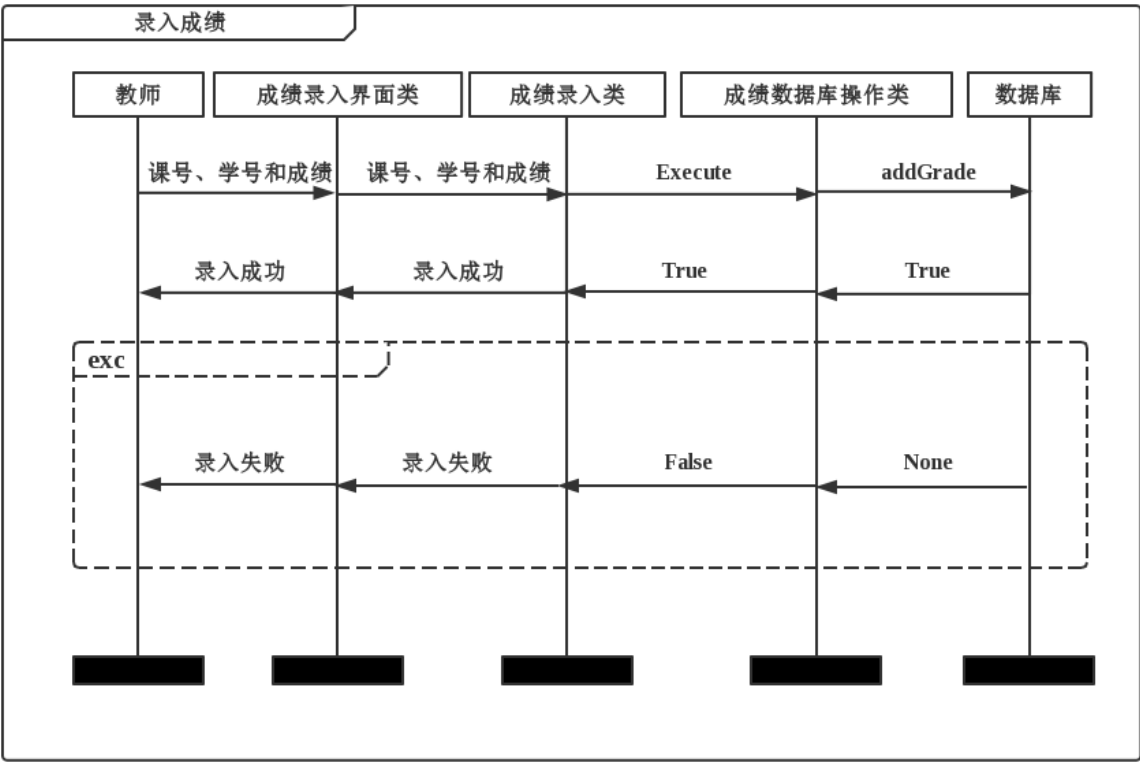


图 9.录入成绩消息序列图

其中该功能使用到的类的详细描述见表 10-12.

表 10.成绩录入界面类

类名	成绩录入界面类
种类	界面类
方法:	
1.	向老师以列表的方式列出所有可以选择的课程
2.	老师可以选择某一个课程，填写该课程的成绩
3.	提供“课程成绩提交”按钮

表 11.成绩录入类

类名	成绩录入类
种类	操作类
方法:	
ActionForward excute(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)[增加学生课程成绩]	

表 12.成绩操作数据库类

类名	成绩数据库操作类
种类	操作类
方法:	
	boolean addGrade(IntelForm Intel) [在数据库中增加成绩信息]
	boolean updateGrade(IntelForm Intel) [在数据库中更新成绩信息]
	IntelForm getGrade(String id) [根据 ID 在数据库中取得对应的成绩信息]

3.3.3.3 查看成绩消息序列图

查看成绩的消息序列图如图 9，该功能主要进行学生成绩的查询。

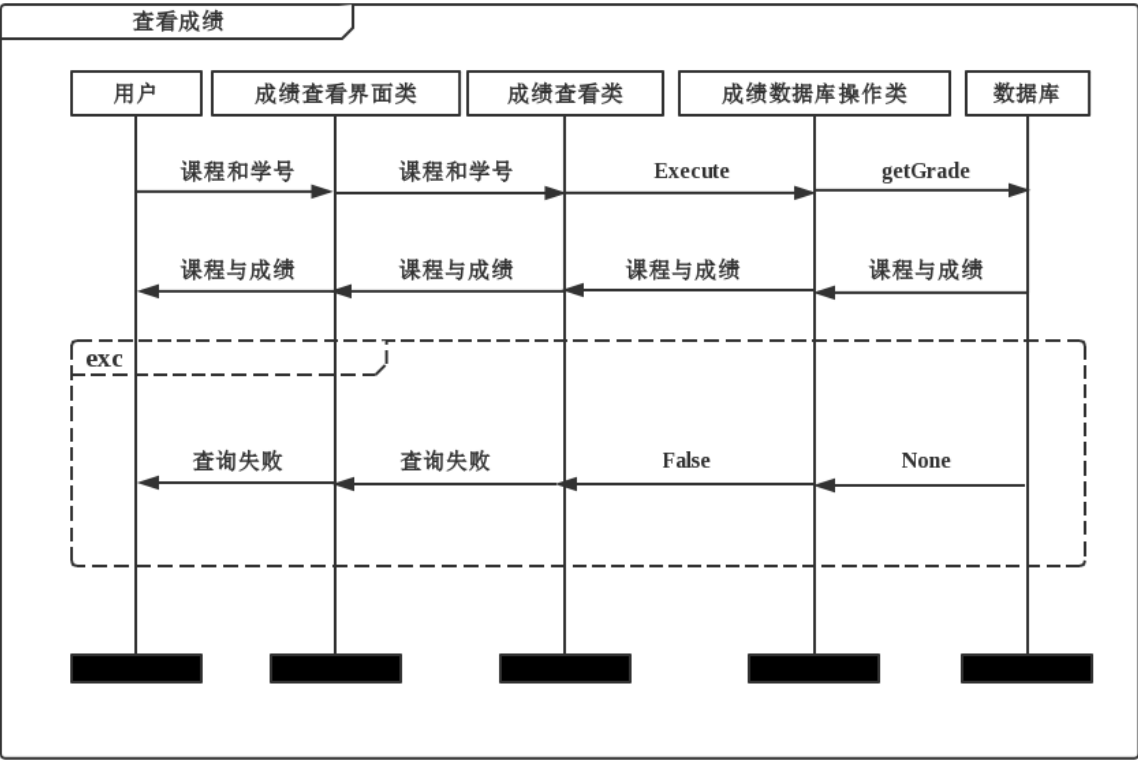


图 10.查看成绩消息界面图

其中该功能使用到的类的详细描述见表 13-14.

表 13.成绩查看界面类

类名	成绩查看界面类
种类	界面类
方法:	
	以列表的方式查看成绩。

表 14.成绩查看类

类名	成绩查看类
种类	操作类
方法:	ActionForward excute(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)[查看学生课程成绩]

3.3.3.4 修改成绩消息序列图

修改成绩的消息序列图如图 9，该功能主要进行学生成绩的修改。

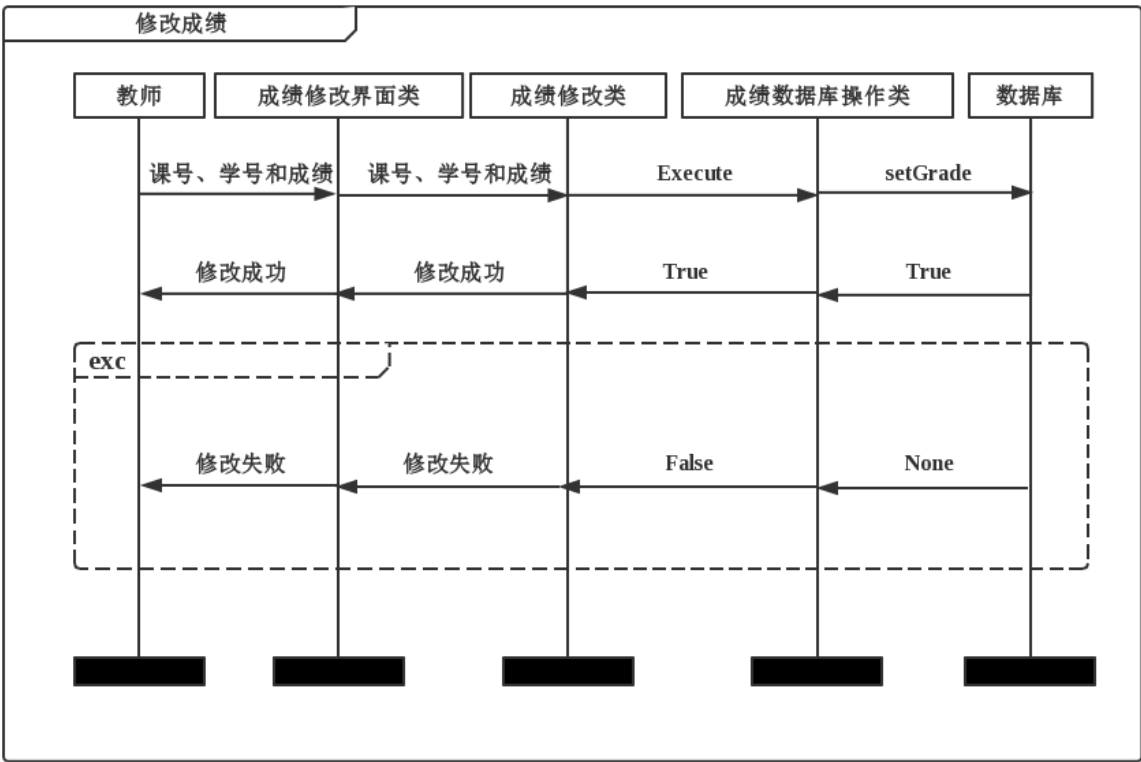


图 11.修改成绩消息序列图

其中该功能使用到的类的详细描述见表 15-16.

表 15.成绩修改界面类

类名	成绩修改界面类
种类	界面类
方法:	<div>1. 向老师以列表的方式列出所有可以修改的课程</div> <div>2. 老师可以选择某一个课程，修改该课程的成绩</div> <div>3. 提供“提交修改”按钮</div>

表 16.成绩修改类

类名	成绩修改类
种类	操作类
方法： ActionForward excute(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)[修改学生课程成绩]	

3.3.3.5 用户管理消息序列图

用户的消息序列图如图 9，该功能主要进行教务员对用户的消息管理。

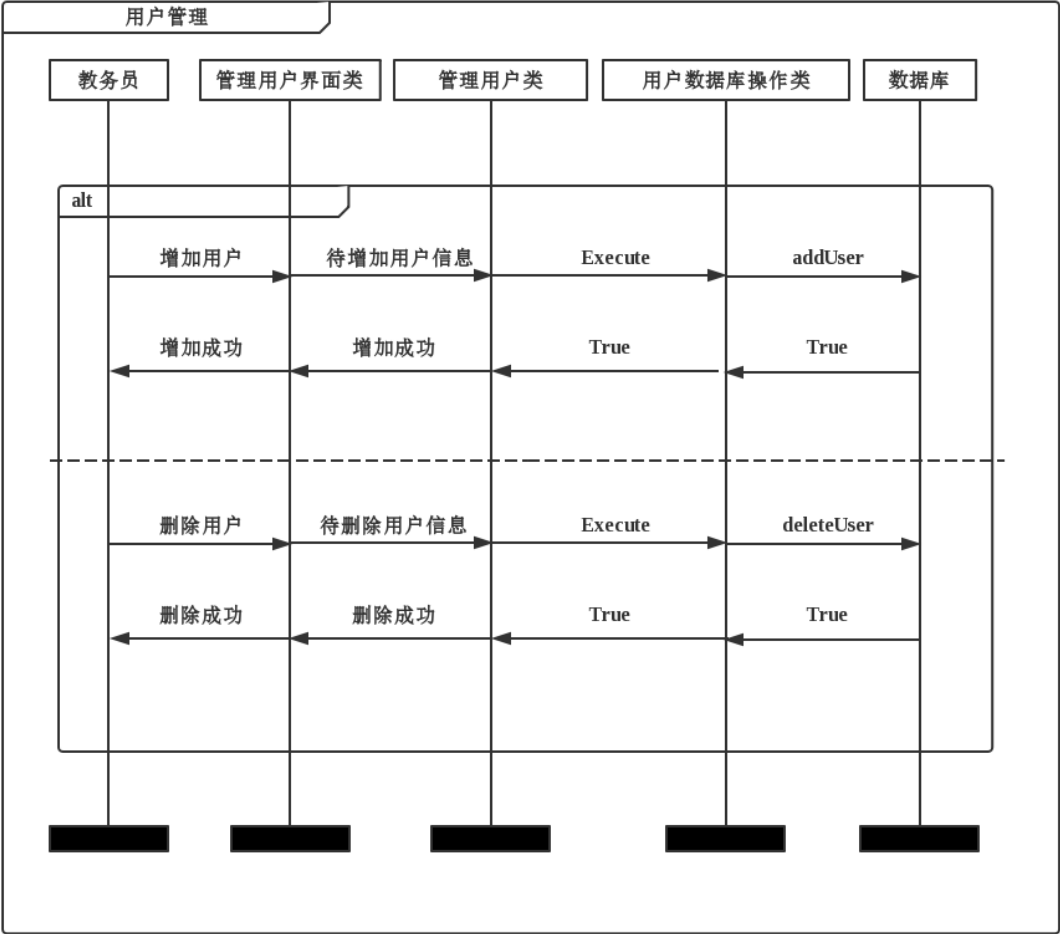


图 12.用户管理消息序列图

其中该功能使用到的类的详细描述见表 17-19.

表 17. 管理用户界面类

类名	管理用户界面类
种类	界面类
方法:	
	1. 向教员以列表的方式列出要增加还是删除用户。
	2. 教员可以选择某一个, 填写该用户信息。
	3. 提供“提交修改”按钮

表 18.管理用户类

类名	管理用户类
种类	操作类
方法:	
	ActionForward excute(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)[修改用户信息]

表 19.用户数据库操作类

类名	用户数据库操作类
种类	操作类
方法:	
	boolean addUser(CourseForm course)[在数据库中增加用户]
	boolean deleteUser (String id)[在数据库中删除用户]

3.4 行为视图

本说明书使用 Philippe Kruchten 提出的从不同角度勾画系统的蓝图, 即“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进行观察, 分析其影响系统的上下文和商业目标情况[1]。

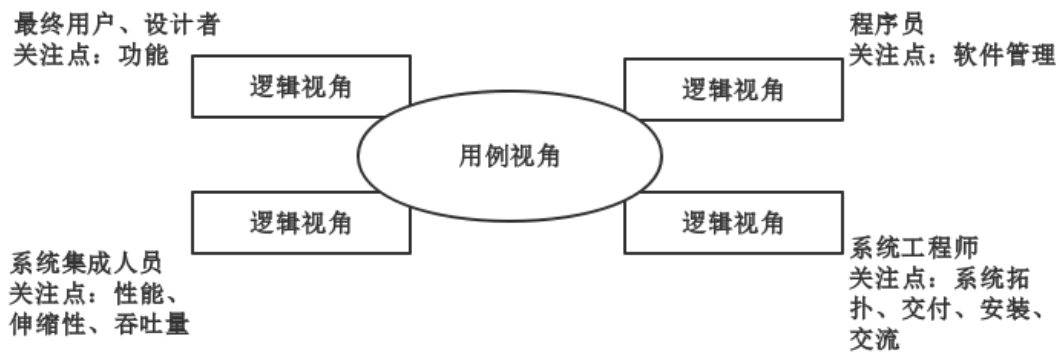


图 13."4+1"视图模型

3.4.1 逻辑视角

本系统在软件逻辑架构的设计工作上采用的是轻量级 Struts 框架，该框架是基于 MVC 模式的一种架构。图 14 为本文系统的软件逻辑架构示意图。

在 Struts 框架技术中，Action 和 Actionservlet 具有控制器的功能，而 JavaBean 则是用来充当系统的控制模型的角色。具体实现的过程为:当用户提出操作请求之后，第一步会由 Tomcat 服务器负责接收用户发出的操作请求的信息，在将此信息请求传送到控制器 Actionservlet 中，接下来由 Action 来负责接收操作指令，然后从数据库 MySQL 中找到的对应信息(像班级管理、学生管理、教师管理以及成绩查询等数据信息)，之后再由数据访问结果返回给视图呈现给用户。

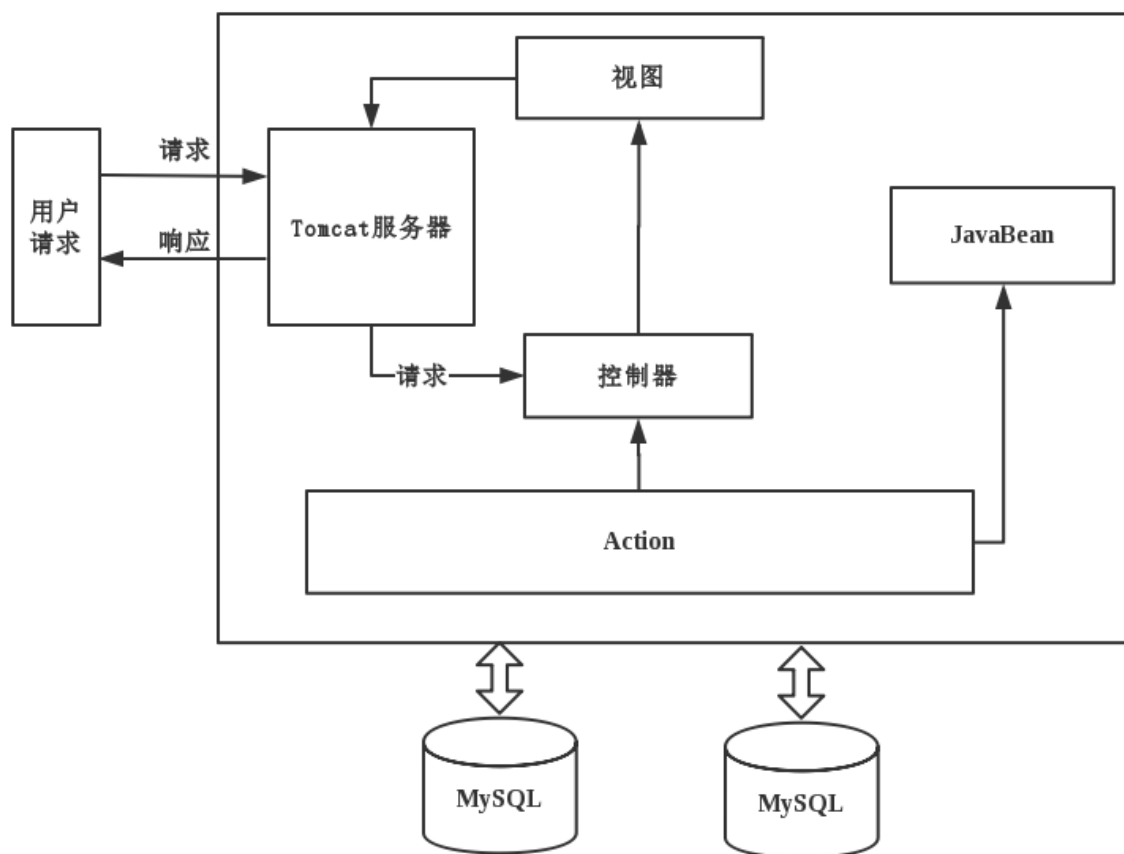


图 14. 系统逻辑视角

3.4.2 进程视角

本说明书使用 SDL 语言对针对进程视角进行描述。主要描述在概念级的软件体系结构下，系统运行态的情况。描述系统在执行时，包括哪些进程（包括线程、进程、进程组），以及它们之间是如何进行通信的、如何进行消息传递、接口如何。并且来说明如何进行组织。

Process 学生进程视图如图 15 所示。学生登录系统并选择自己要进行的操作，即进行查询成绩或修改密码，若选择查询成绩，则进入成绩管理进程查询自己的成绩；若选择修改密码，则进入密码修改进程修改自己的密码。成绩管理与修改密码对应的进程本文已经在运行级的体系结构中进行了详细的消息序列图的描述，因此此处不再赘述。

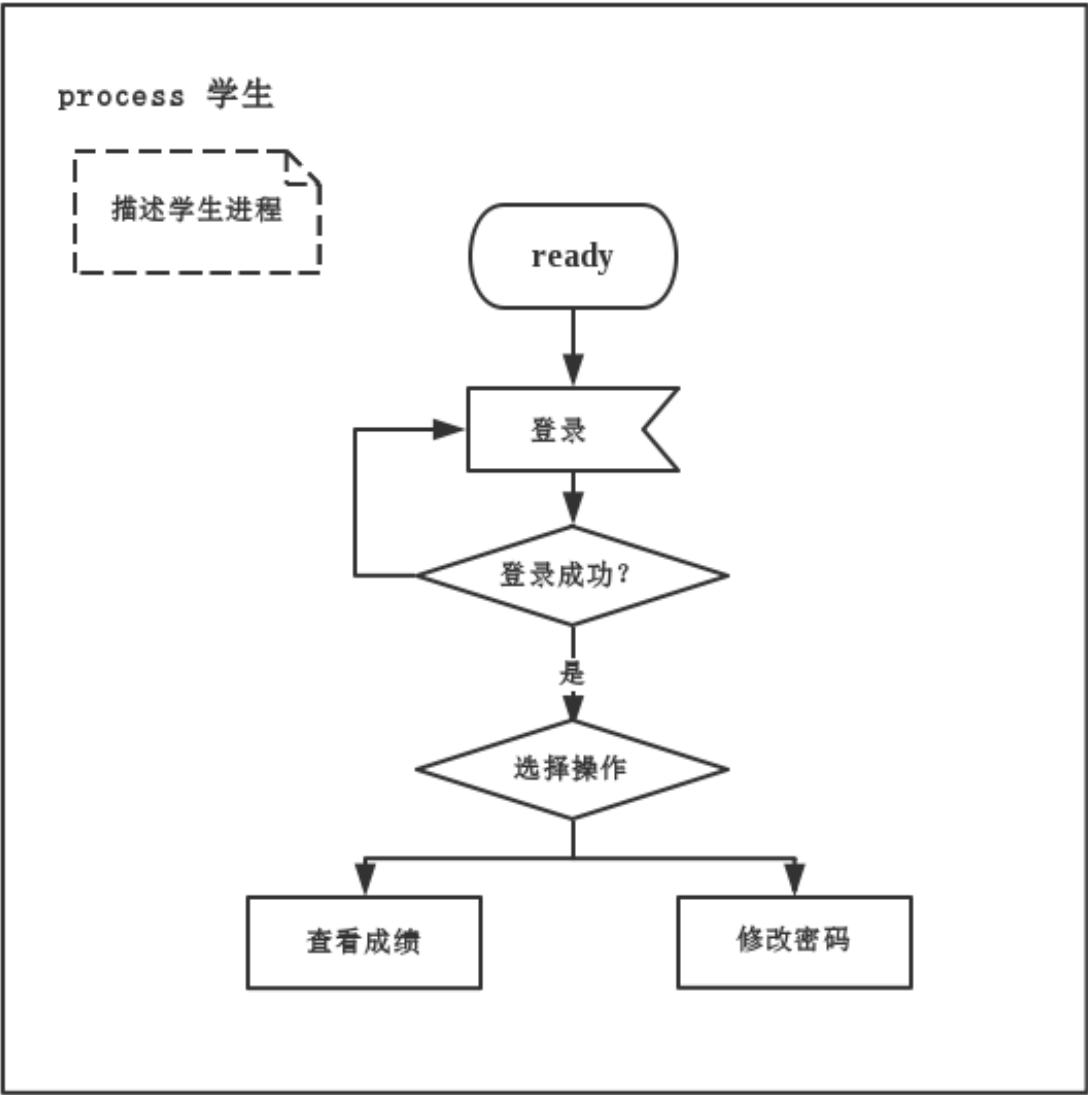


图 15.学生进程

Process 教师进程视图如图 16 所示。教师登录系统并选择自己要进行的操作，即进行成绩管理或修改自己的密码，若选择成绩管理则进入成绩管理进程，进行查询、修改或录入成绩；若选择修改密码则进入相应的修改密码进程修改密码。成绩管理与修改密码对应的进程本文已经在运行级的体系结构中进行了详细的消息序列图的描述，因此此处不再赘述。

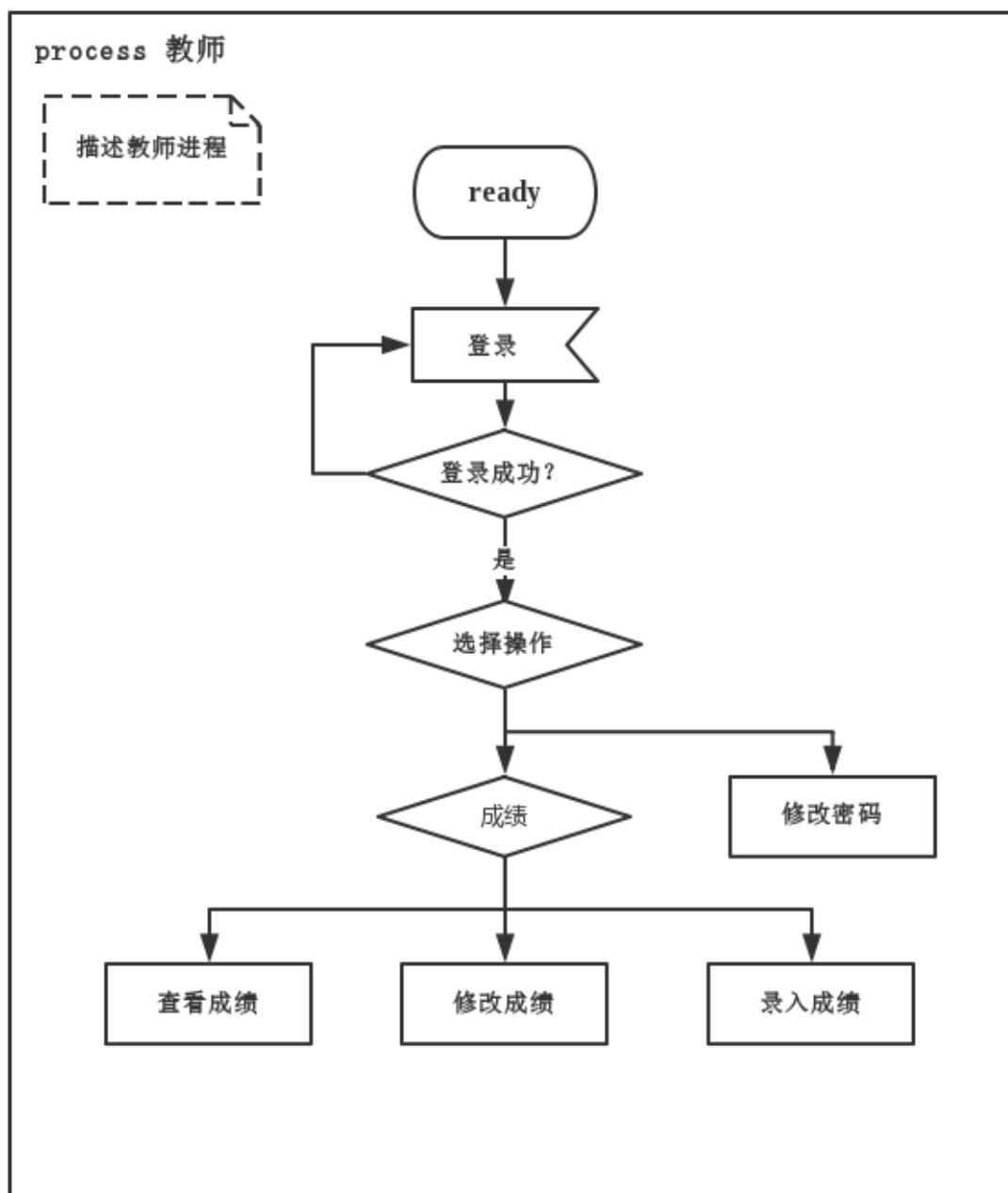


图 16.教师进程

Process 教务员进程视图如图 17 所示。教务员登录系统并选择自己要进行的操作，即进行成绩管理、用户管理或修改自己的密码，若选择成绩管理则进入成绩管理进程，进行查询、修改或录入成绩；若选择用户管理，则进入用户管理进程，进行增加或删除用户；若选择修改密码，则进入修改密码相应的进程。成绩管理、用户管理与修改密码对应的进程本文已经在运行级的体系结构中进行了详细的消息序列图的描述，因此此处不再赘述。

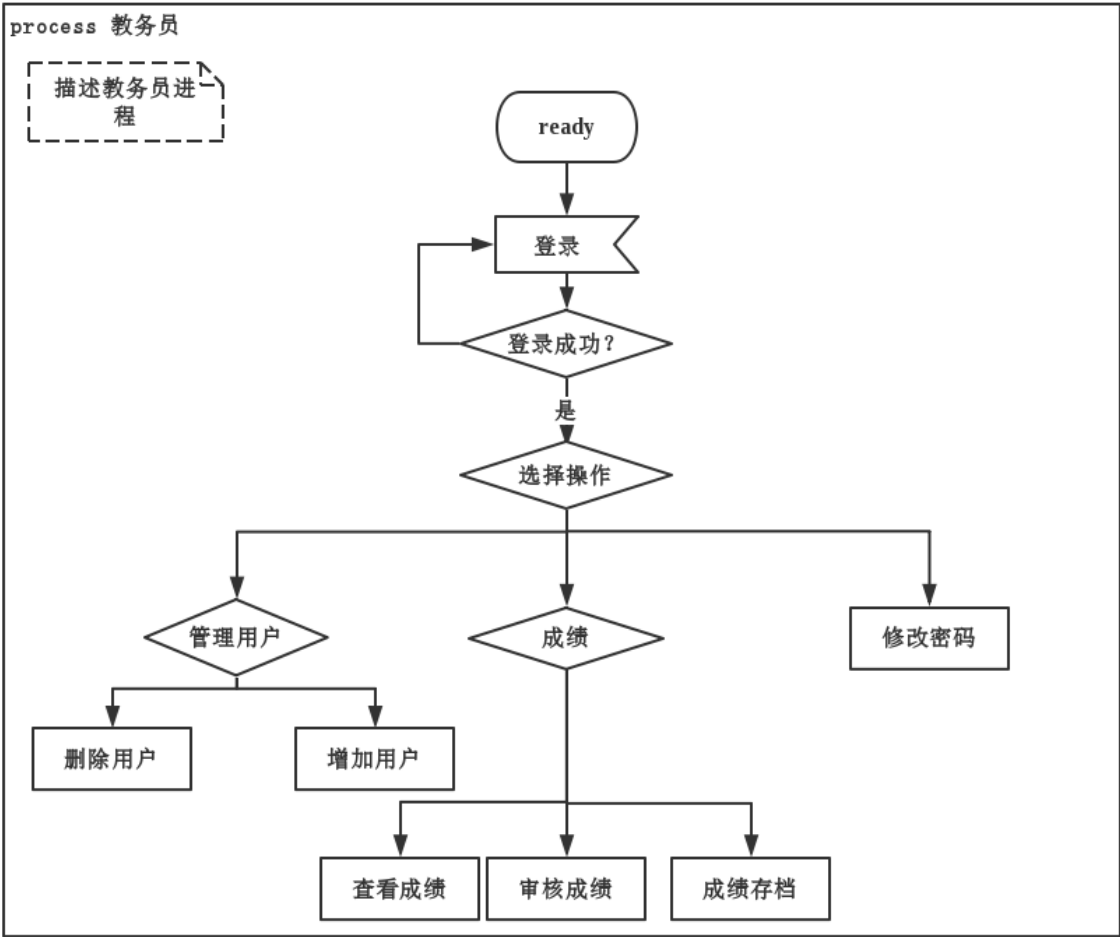


图 17. 教务员进程

3.4.3 实现视角

实现视角，又叫开发视图（Development View），描述了在开发环境中软件的静态组织结构，即关注软件开发环境下实际模块的组织，服务于软件编程人员。将软件打包成小的程序块（程序库或子系统），它们可以由一位或几位开发人员来开发。子系统可以组织成分层结构，每个层为上一层提供良好定义的接口。因此系统的架构开发视图如 18。

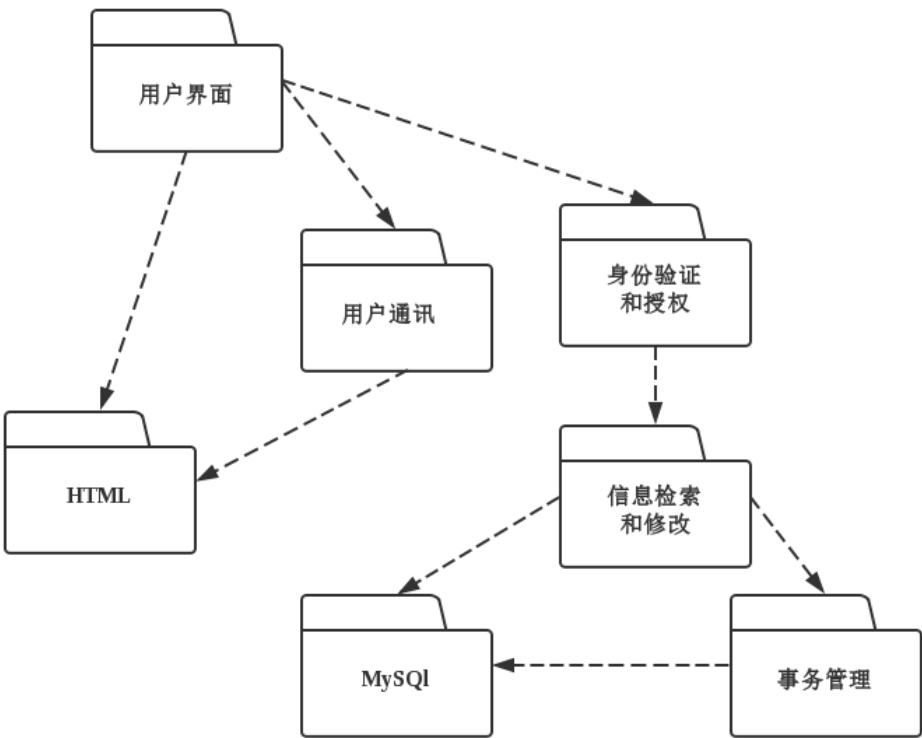


图 18. 架构开发视图

系统的实现视图用模块和子系统图来表达。完整的开发架构只有当所有软件元素被识别后才能加以描述。但是，可以列出控制开发架构的规则：分块、分组和可见性。因此本说明书设计系统总包图如 19。

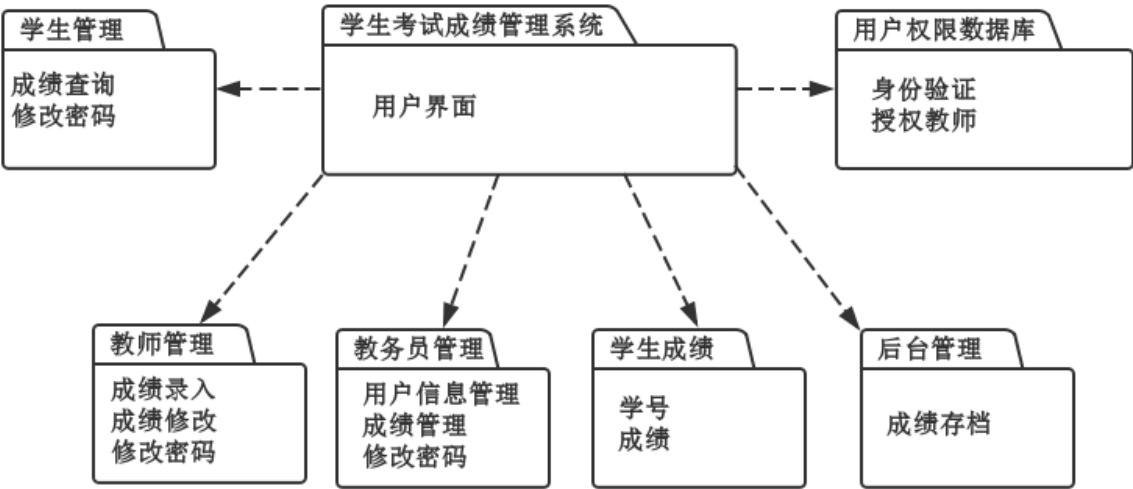


图 19. 系统总包图

3.4.4 部署视角

从系统软硬件物理配置的角度，描述系统的网络逻辑拓扑结构。模型包括各个物理节点的硬件与软件配置，网络的逻辑拓扑结构，节点间的交互和通讯关系等。同时还表达了进程视图中的各个进程具体分配到物理节点的映射关系。由于本系统采用的是 B/S 架构，结合现实的网络拓扑结构，本文设计部署视角如图 20.

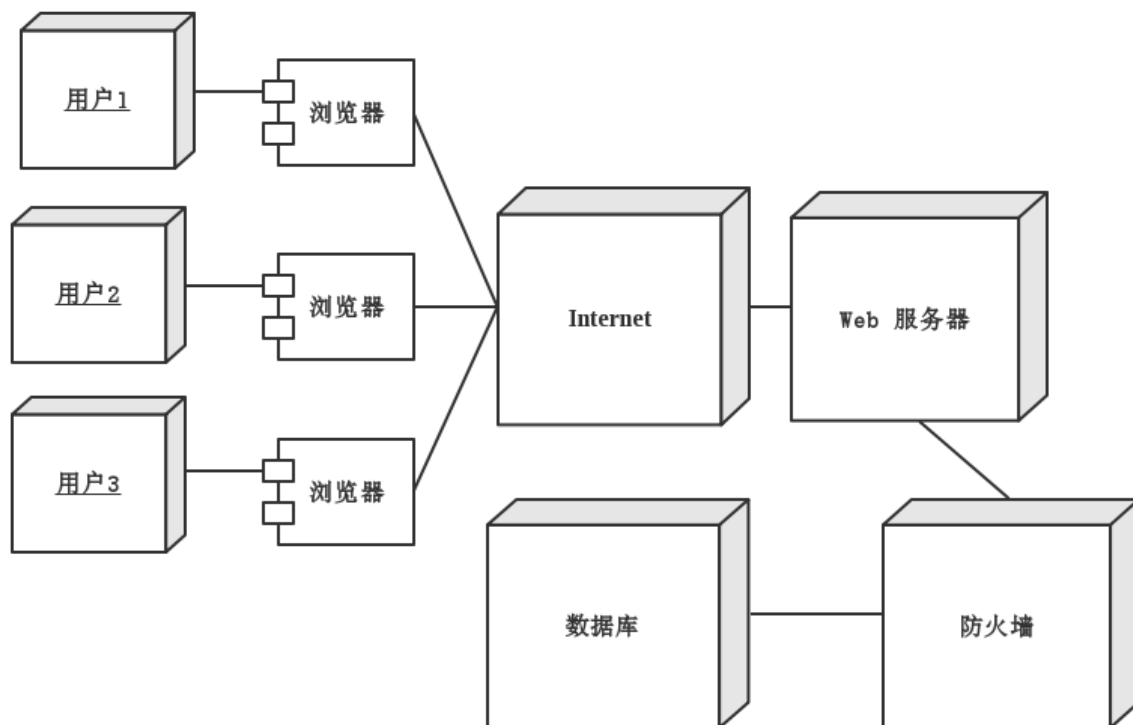


图 20. 系统部署视图

3.4.5 用例视角

用例视角，又称“场景视图”，它综合所有的视图。用于刻画构件之间的相互关系，将四个视图有机地联系起来。可以描述一个特定的视图内的构件关系，也可以描述不同视图间的构件关系。

四种视图的元素通过一组是用例进行无缝协同工作，用例场景描述相应对象之间和过程之间的交互序列。在某种意义上场景是最重要的需求抽象，它们的设计使用对象场景图和对象交互图来表示。

场景视图是其他视图的冗余（因此“+1”），但它起到了两个作用：

- 作为一项驱动因素来发现架构设计过程中的架构元素。
- 作为架构设计结束后的一项验证和说明功能，既以视图的角度来说明，又作为架构原型测试的出发点。

本文设计从“教务员”、“教师”和学生的角度设计用例图如。用例每一项的详细描述参见《学生考试成绩管理系统体系结构说明书》。

学生用例如图 21 所示，学生能够查询成绩，修改自己的密码。

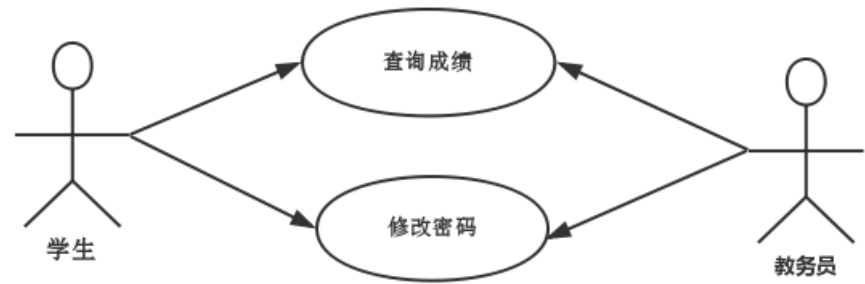


图 21.学生用例视图

教师用例如图 22 所示，教师能够查询成绩，录入成绩，修改成绩以及修改密码。

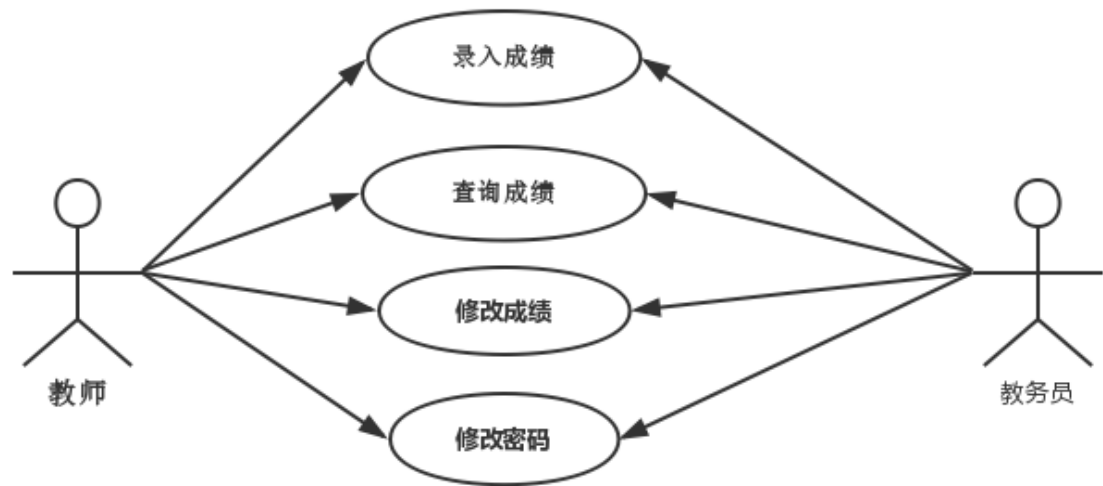


图 22. 教师用例视图

教务员用例如图 23 所示，教务员能够进行成绩管理，包括审核成绩、发布成绩和成绩存档；能够进行用户管理，包括增加、修改和删除用户；能够进行课程管理，包括增加、修改和删除课程信息；同时教务员还能够修改自己的密码。

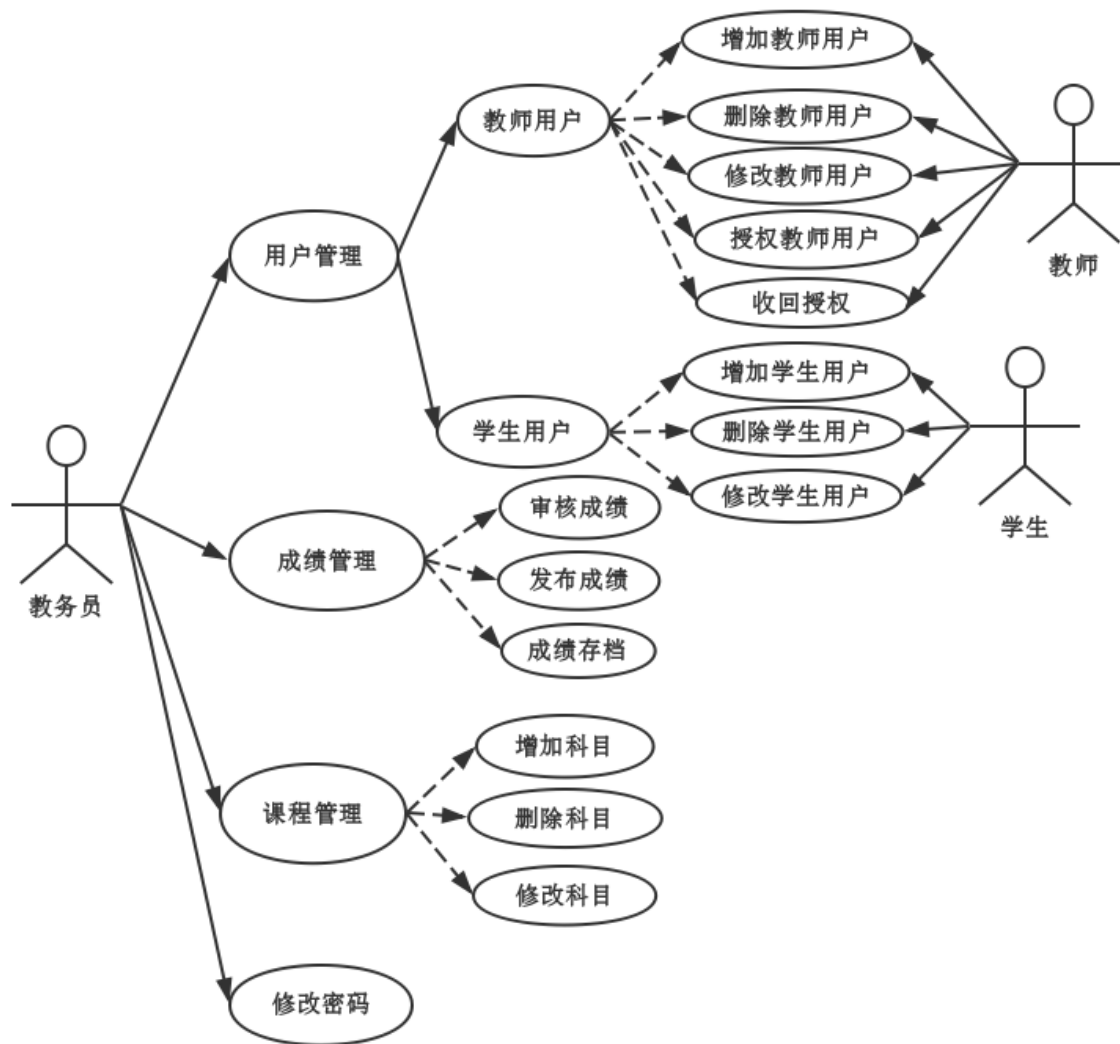


图 23. 教务员用例视图

3.5 实现问题

3.5.1 用户增加

3.5.1.1 问题描述

当学校人数上升时（例如，10 万学生和老师）同时使用时，会出现哪些问题？软件的结构上如何解决？

3.5.1.2 问题分析

➤ 问题产生原因

由于不需要安装客户端，客户端易扩展，还有就是外部用户也可以访问，所以 B/S 面对大量的不可知用户。

➤ 大量用户并发的后果

用户增多时，系统的吞吐量增大，并发数增加，导致服务器响应速度慢。同时页面需要不断地动态刷新，尤其是用户增多，网速慢得情况，很费时。服务器承担着重要的责任，数据负荷较重。一旦发生服务器宕机等问题，后果不堪设想。

3.5.1.3 解决方案

➤ 系统设计方面

1. 监控在线用户数量

图 24 是响应时间、用户并发量与吞吐量的关系图。从图中可以看出，整体上，用户数量的增加会导致系统响应时间的增加，但增加的程度又不尽相同。在点 A 之前，系统响应时间增长相对较缓，在点 A 之后，响应时间急剧上升，已经严重影响到了用于体验。

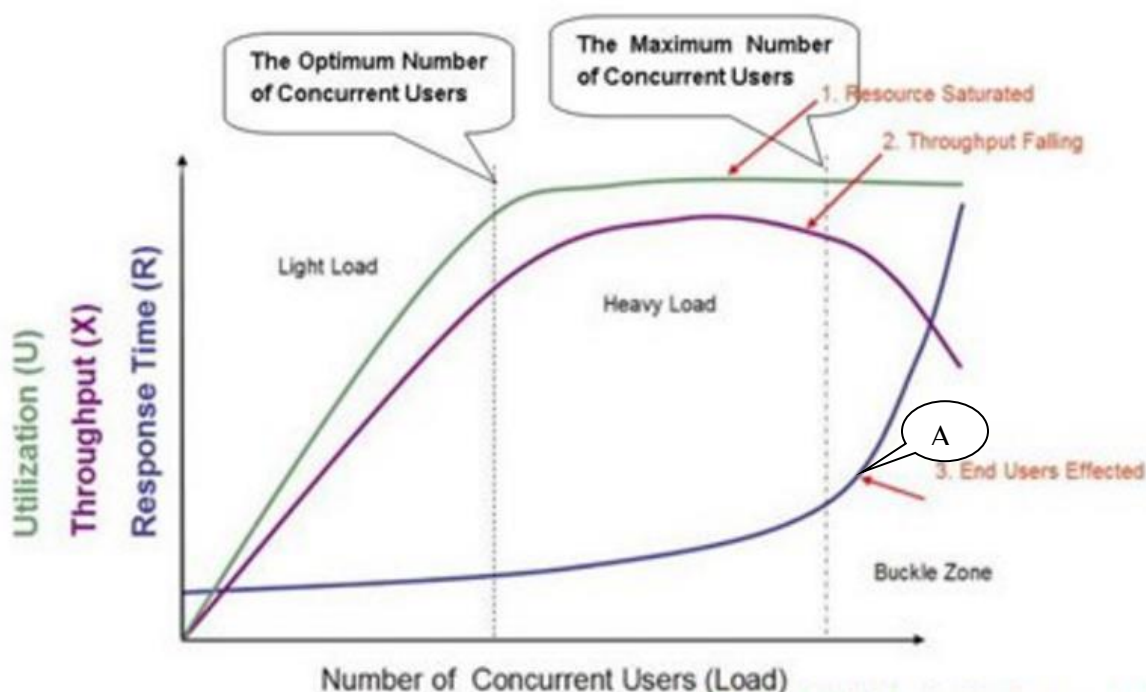


图 24. 响应时间与吞吐量的关系

因此，在不增加任何硬件配置的情况下，在系统启用前要进行用户量测试，找到图中的 A 点，以此时用户数量作为系统容许的最大用户并发数量，以此时间作为最大响应时间。系统启动后，可以监控在线用户数，到达最大容纳数之后，向用户返回系统忙碌等相关信息。

2. HTML 静态化

效率最高、消耗最小的就是纯静态化的 html 页面，因此我们的网站上的页面采用静态页面来实现，这个最简单的方法其实也是最有效的方法。

3. 数据库集群和库表散列

在面对大量访问的时候，这时一台数据库将很快无法满足应用，于是我们需要使用数据库集群或者库表散列。在数据库集群方面，本系统使用的 MySQL 提供的 Master/Slave 也是提供了解决方案。

➤ 系统硬件方面

在面对大量用户访问、高并发请求方面，基本的解决方案集中在这样几个环节：使用高性能的服务器、高性能的数据库、高效率的编程语言、还有高性能的 Web 容器。但是除了这几个方面，只能相对减轻，但不能从根本上解决网站面临的高负载和高并发问题。

3.5.3 质量属性

3.5.3.1 问题描述

你的体系结构，对质量属性是如何影响的？

3.5.3.2 B/S 架构对质量属性的影响

表 20.B/S 架构对质量属性的影响

质量属性	B/S 架构对质量属性的影响
性能	应用都集中于总部服务器上,各应用结点并没有任何程序,一个地方更新则全部应用程序更新,可以做到快速服务响应。
易用性	客户端无需安装，有 Web 浏览器即可。
可伸缩性	在上文论述中可以得出，B/S 架构既可以支持少量用户访问，也可以支持大量用户并发访问。
可修改性	体系结构必须支持从当前的第四代语言向.NET 或 J2EE 环境的过度。
密安性	数据集中存放于总部的数据库服务器,客户端不保存任何业务数据和数据库连接信息,也无需进行数据同步,密安性高
可适用性	C/S 结构软件仅适用于局域网内部用户或宽带用户; 而 B/S 结构软件可以适用于任何网络结构(包括 28.8K 拨号入网方式),特别适于宽带不能到达的地方。

4. 系统的质量分析和评价

4.1 场景分析

系统的使用场景与质量属性是密切相关的，也是决定体系结构的重要依据。通过场景可以很好的评估体系结构，本文使用 SEI 开发的体系结构折中分析方法(Architecture Tradeoff Analysis Method).

4.1.1 用例场景

表 21.用例场景

序号	动作	期望结果	测试结果
1	用户输入用户名与密码登录	弹出登录成功	
2	用户选择查询成绩	跳转到查询成绩界面	
3	用户输入查询成绩信息	可以看到查询结果	
4	教师录入成绩	可输入的成绩信息符合实际，教师点击提交后弹出录入成功	
5	教师修改成绩	可输入的成绩信息符合实际，教师点击提交后弹出修改成功	
6	教务员点击授权教师	弹出可授权教师列表	
7	教务员增加用户	可输入的用户信息符合实际，教务员点击提交后弹出增加成功	
8	教务员删除用户	可输入的用户信息符合实际，教务员点击提交后弹出删除成功	
9	教务员增加课程	可输入的用户信息符合实际，教务员点击提交后弹出增加成功	
10	教务员删除课程	可输入的用户信息符合实际，教务员点击提交后弹出删除成功	
11	教务员修改课程	可输入的用户信息符合实际，教务员点击提交后弹出修改成功	
12	教务员审核成绩	教务员可以查看到待审核的成绩信息	
13	教务员发布成绩	教务员将审核通过的成绩点击发布，完成发布	
14	成绩存档	教务员点击成绩存档后能在本地数据库看到存档信息	
15	用户修改密码	用户输入旧密码，输入两边新密码后提示修改成功。	

4.1.2 增长性场景

4.1.2.1 问题描述

考虑其他可能的（增长性）场景，例如用手机。

4.1.2.2 问题分析

由于 B/S 架构第一层是浏览器，只有简单的输入输出功能，处理极少部分的事务逻辑。它不受任何平台的限制，只要有浏览器就能上网浏览就可以使用。浏览器先向 Web 服务器发送用户请求信息，Web 服务器统一请求后会向数据库服务器发送访问数据库的请求。当数据库服务器收到了 Web 服务器的请求后，会对 SQL 语句进行处理，并将返回的结果发送给 Web 服务器，接下来，Web 服务器将收到的数据结果转换为 HTML 文本形式发送给浏览器，也就是我们打开浏览器看到的界面。

4.1.2.3 问题结论

在问题分析中我们可以看出，不论用户使用的终端设备是电脑，手机，或者平板电脑，对本系统采用的 B/S 的架构来说都是没有影响的。只要用户的终端设备上能够使用浏览器，本系统都可以正常用行。

4.1.3 探索性场景

暂无