

学生期末考试成绩管理系统

体系结构设计报告

2.0

2015/12/19

苏畅

2013212042

体系结构设计工程师

修订历史

日期	描述	作者	备注
2015/12/13	版本 1.0	苏畅	体系基本框架设计
2015/12/19	版本 2.0	苏畅	完成体系结构文档

文档批准

以下需求分析报告已经被以下机构人员批准并认可：

签字	打印姓名	标题	日期

报告引言

该报告结合需求分析工程师苏畅于 2015/11/21 编写的《学生期末考试成绩管理系统-需求分析报告》，进一步分析设计了学生期末考试成绩管理系统的体系结构。在此感谢王安生老师所编著的《软件工程化》中的相关理论知识给该报告带来的帮助，同时感谢 Google 学术机构对相关文献的索引，以及傅滢、孔丹晨同学对一些问题的解答。

作业相关要求对应页码

作业准备：

- 1) 复习 11.1 节（体系结构分级，不写代码级）.....17
- 2) 建议读书的第 26 章 26.4-26.5 节
- 3) 可参考模板（图 10-6），封面格式（参照需求文档），交电子版和打印版

注意要回答出：

- 1) 选择体系结构的理由？.....17
- 2) 你为何会选择三层（多层）B/S 结构？.....17
- 3) 你的体系结构，对质量属性是如何影响的？.....12
- 4) 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？..... 29
- 5) 考虑其他可能的（增长性）场景，例如，用手机.....29

目录

修订历史.....	2
文档批准.....	2
报告引言.....	2
作业相关要求对应页码.....	2
1. 介绍.....	6
1.1 目的.....	6
1.2 项目相关信息.....	6
1.3 术语定义.....	6
1.4 参考文献.....	6
2. 体系结构需求.....	6
2.1 关键指标.....	7
2.2 体系结构用例.....	7
2.2.1 用户角色.....	7
2.2.2 用例图 1-学生.....	7
2.2.3 用例图 2-教师.....	8
2.2.4 用例图 3-教务员.....	9
2.2.5 用例图 4-档案.....	10
2.3 各相关方对体系结构的要求.....	11
2.4 约束条件.....	11
2.5 非功能需求.....	12
2.5.1 质量需求.....	12
2.5.1.1 性能.....	12
2.5.1.2 可靠性.....	12
2.5.1.3 可使用性.....	12
2.5.1.4 密安性.....	12
2.5.1.5 可维护性.....	12
2.5.1.6 可移植性.....	13
2.5.2 工程需求.....	13
2.5.2.1 设计约束.....	13
2.5.2.2 逻辑数据库需求.....	13
2.5.3 其它需求.....	14
2.5.3.1 良好的人机交互能力.....	14
2.5.3.2 界面需求.....	15
2.5.3.3 数据容量需求.....	15
2.6 风险.....	15
2.6.1 技术性风险.....	15
2.6.2 工程性风险.....	15
2.6.3 环境性风险.....	15
3. 解决方案.....	16
3.1 相关的体系结构模式.....	16
3.1.1 传统的信息管理系统开发模式(C/S).....	16
3.1.2 B/S 信息管理体统开发模式.....	16

3.1.3 体系结构选择.....	17
3.2 体系结构概述.....	17
3.3 结构化视图.....	17
3.3.1 概念级体系结构.....	17
3.3.2 模块级体系结构.....	18
3.3.2.1 功能分解结构图.....	19
3.3.2.1.1 学生交互模块.....	21
3.3.2.1.2 教师交互模块.....	21
3.3.2.1.3 教务员交互模块.....	21
3.3.2.2 系统分层结构图.....	21
3.4 行为视图.....	22
3.4.1 运行级体系结构.....	22
3.4.2 各子功能模块的消息序列图.....	23
3.4.2.1 学生交互模块.....	23
3.4.2.2 教师交互模块.....	25
3.4.2.3 教务员交互模块.....	27
3.5 实现问题.....	29
4. 质量的分析与评价.....	30
4.1 场景分析.....	30
4.1.1 场景设计.....	30
4.1.2 用例设计.....	30
4.2 原型分析.....	31
5. 同行评审.....	33
A. 附录.....	34
A.1 附录 1.....	34

表目录

表 1-1 术语表	6
表 2-1 关键指标分类	7
表 2-2 相关方对体系结构的要求	11
表 2-3 项目主要约束条件	11
表 2-4 项目设计约束条件	13
表 3-1 SGM 部件划分	18
表 4-1 场景设计	30
表 4-2 用例设计	30
表 5-1 同行评审意见表	33

图目录

图 2-1 学生用例图	8
图 2-2 教师用例图	9
图 2-3 教务员用例图	10
图 2-4 档案用例图	10
图 2-5 数据库 E-R 图	14
图 3-1 三层 B/S 模式的系统体系结构模型	17
图 3-2 SGM 系统的概念级体系结构	18
图 3-3 SGM 功能结构图	20
图 3-4 SGM 系统的分层结构	22
图 3-5 SGM 系统的运行体系结构	23
图 3-6 学生模块消息序列图	24
图 3-7 教师模块消息序列图	27
图 3-8 教务员模块消息序列图	28
图 4-1 SGM 系统登录界面	31
图 4-2 学生界面	32
图 4-3 教师界面	32
图 4-4 教务员界面	33
图 4-5 后台数据库界面	33

1. 介绍

本文档是一个学生考试成绩管理系统的体系结构分析报告。本文档主要通过对学生期末考试成绩管理系统的质量需求进行分析，选择 B/S 结构为系统的体系结构模式，设计出该系统的总体体系结构，并进行分级。

1.1 目的

本文档通过对学生考试成绩管理系统的需求进行分析，提供了一份从逻辑视角出发的体系结构文档。目的是为了使用户了解该软件系统的主要功能以及层次结构，便于用户与技术开发人员的理解与交流，熟练最终用户对系统的使用。

1.2 项目相关信息

本文档所分析的软件系统需求范围如下：

(1) 该软件系统的名字为：学生考试成绩管理系统(Student Grades Management System, SGM)。

(2) 该系统不适用于以下情况：

a) 由于该系统只是学生管理系统的一个子模块，所以该系统不包含的功能有：学生学籍管理，学校课程与专业管理。

b) 该系统属于在 Web 平台下的软件系统，因此在没有计算机网络的环境下，不能提供使用。

(3) 该系统只供学校的教务处使用。系统应用的目的是为了更方便学生在参加完期末考试之后可以及时查看个人成绩。同时，也可以方便教务处对学生的成绩进行存档。

1.3 术语定义

表 1-1 术语表

术语	代表含义
SGM	学生考试成绩管理系统
IMS	信息管理系统

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，2014:135-138.
- [2] 王国华，薛福冰.《基于 UML 的高校科研管理系统分析与设计》[J]. 今日科技：2012(06):99-102.

2. 体系结构需求

通过对该系统的质量和约束条件分析，该文档设计出以下的体系结构需求。

2.1 关键指标

软件体系结构为软件系统提供了一个结构、行为和属性的高级抽象，由构成系统的元素描述、元素相互作用、指导元素集成的模式以及这些模式的约束组成。一个良性的软件体系结构应该有以下五个质量要素：

- (1) 体系结构应是适宜的。
- (2) 体系结构应是概念完整的。
- (3) 体系结构应是易于维护和升级的。
- (4) 体系结构应是便于移植的。
- (5) 体系结构应是理性化的。

这五个质量要素体现了体系结构作为早期设计决策对系统需求的支持、实现的约束，管理的组织。

这五个质量要素是比较抽象的，我们从需求、开发、项目管理三个角度细化体系结构的指标，可以划分出以下的表格：

表 2-1 关键指标分类

角度	包含指标
需求角度	功能性 可靠性 可用性 安全性
开发角度	可维护性 可扩展性 可移植性 可测试性 可重用性
项目管理角度	部件无关 风险性 可度量性

在 2.5 节中，将针对具体的指标进行需求分析。

2.2 体系结构用例

2.2.1 用户角色

SGM 系统共有四种用户角色：学生、教师、教务员、档案。下面本文将针对四种用户画出它们的用例图。

2.2.2 用例图 1-学生

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：
 - a) 用户登录校验。
 - b) 用户信息管理。

c) 成绩查询。

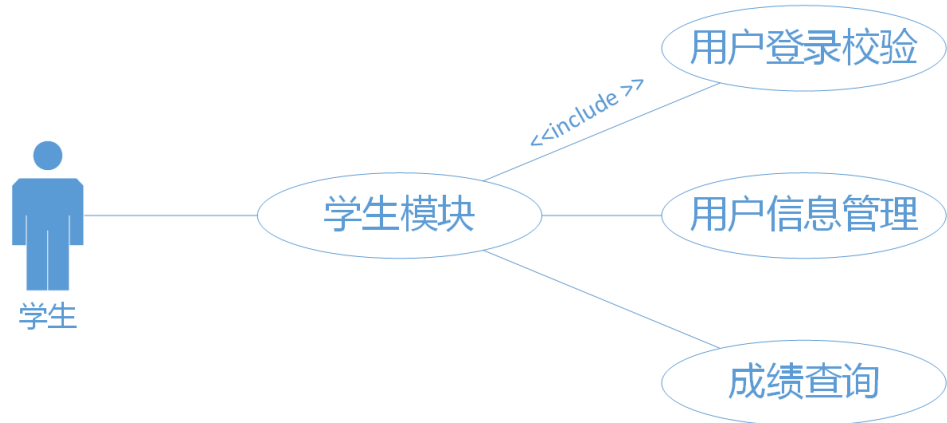


图 2-1 学生用例图

图例说明：

图形	含义
	参与者
	用例

2.2.3 用例图 2-教师

设计论述：

- (1) 教师用户负责的功能是教师模块。
- (2) 教师模块包含的子功能有：
 - a) 用户登录校验。
 - b) 用户信息管理。
 - c) 成绩查询。
 - d) 成绩录入，教师将成绩录入后发送给教务员审核。
 - e) 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。

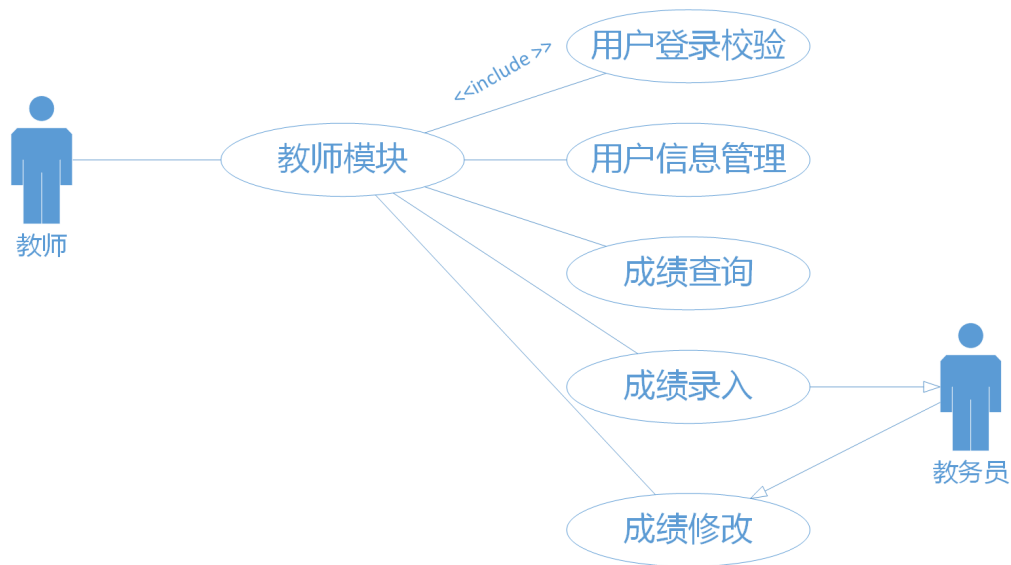


图 2-2 教师用例图

图例说明：

图形	含义
	参与者
	用例

2.2.4 用例图 3-教务员

设计论述：

- (1) 教务员用户负责的功能是教务员模块。
- (2) 教务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将告知用户。
 - d) 成绩审定，包含两个子功能：如果审定无误，成绩发布，需要归档；如果有错误，将授权教师修改成绩。

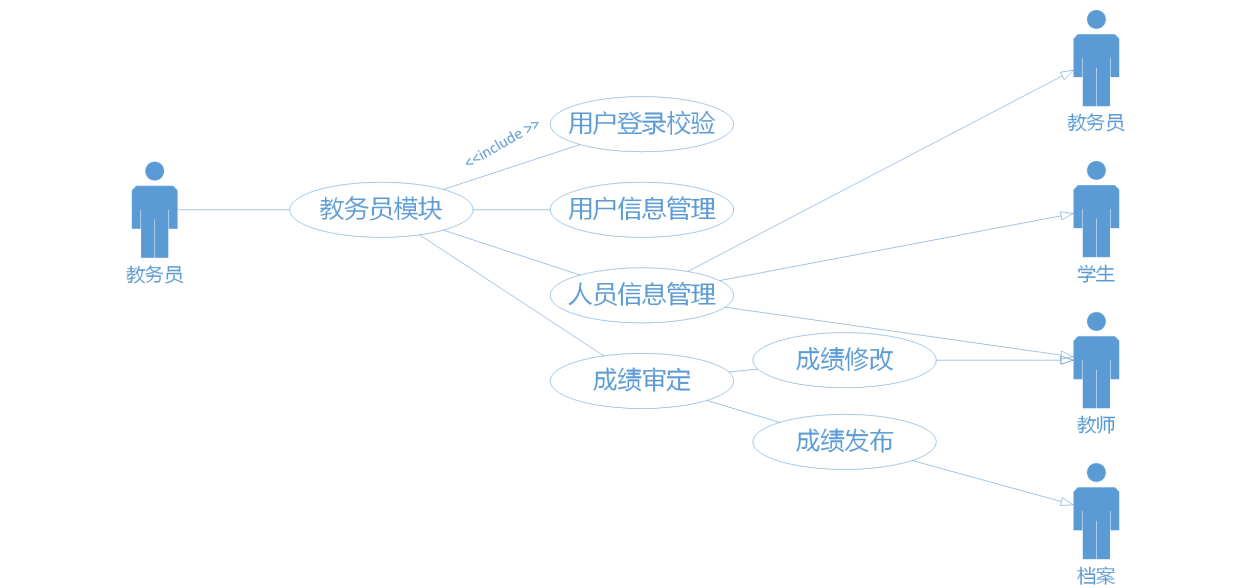


图 2-3 教务员用例图

图例说明：

图形	含义
	参与者
	用例

2.2.5 用例图 4-档案

- 设计论述：
- (1) 档案用户负责的功能是档案模块。
 - (2) 档案模块包含的子功能有：
 - a) 归档，由教务员传送归档信息。
 - b) 读档。

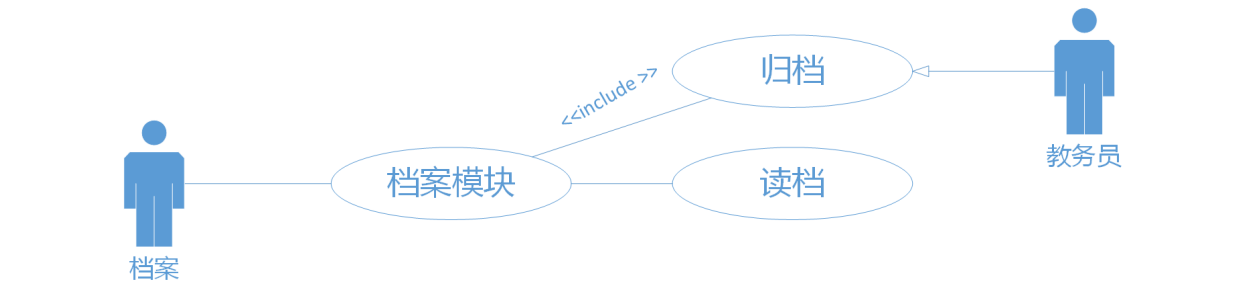




图 2-4 档案用例图

图例说明：

图形	含义
	参与者
	用例

2.3 各相关方对体系结构的要求

在 SGM 系统的开发的过程中，存在的相关方有如下 5 类：

- (1) 最终用户、设计者，其关注点是功能
- (2) 程序员，其关注点是软件管理
- (3) 用户、分析人员和测试人员，其关注点是特性
- (4) 系统集成人员，其关注点是性能、伸缩性、吞吐量
- (5) 系统工程师，其关注点是系统拓扑；交付、安装、交流

根据以上的分析，可以看出：

最终用户对体系结构的要求是：功能清晰、完善；特性完备；

程序员对体系结构的要求是：层次清晰，易于软件的管理；

分析人员和测试人员对体系结构的要求是：测试模块鲁棒性高，体现出软件测试要考虑到的问题；

系统集成人员对体系结构的要求是：考虑到系统性能和约束条件对体系结构的影响；

系统工程师对体系结构的要求是：体系结构清晰，系统拓扑结构逻辑正确；包含系统最终交付安装以及跟踪的设计。

可以将以上要求描述为如下的表格：

表 2-2 相关方对体系结构的要求

相关方	要求
最终用户	功能清晰、完善；特性完备
程序员	层次清晰，易于软件的管理
分析人员和测试人员	测试模块鲁棒性高，体现出软件测试要考虑到的问题
系统集成人员	考虑到系统性能和约束条件对体系结构的影响
系统工程师	体系结构清晰，系统拓扑结构逻辑正确；包含系统最终交付安装以及跟踪的设计

2.4 约束条件

针对项目的基本要素进行分析，可以得到下面的主要约束表格。

表 2-3 项目主要约束条件

基本要素	主要约束
项目运行范围	完成项目的开发与测试

项目开发时间	1 个月以内
项目开发成本	1000-2000 元

项目基本限制：使用本项目的学校有 3 万学生，2000 个老师，每学期开出 500 门课，平均一门课有 100 个学生。本项目只考虑学生的期末考试成绩。

2.5 非功能需求

所谓非功能性需求，是指软件产品为满足用户业务需求而必须具有且除功能需求以外的特性^[2]。非功能性需求在需求分析阶段常常被忽略或没有被足够重视。软件产品的非功能性需求包括系统的质量需求和工程需求。对于本系统，主要通过以下三个方面对本系统的非功能性需求进行描述。

2.5.1 质量需求

2.5.1.1 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与课程总数，系统的响应时间应该达到如下标准比较合理：

系统响应登录时间最大=3 秒

单个用户每天操作最大次数=1500 次

预计系统平均响应时间=1.3 秒

系统最大并发用户数=120000 人

2.5.1.2 可靠性

可靠性是指产品在规定的条件下，规定的时间内，完成规定功能的能力。该系统的可靠性需求具体体现在系统能较长时间下稳定运行。同时，该系统需要具备一定的故障恢复能力，即有一定的容错能力。当用户的操作不当引起某些故障时，或者是由于操作系统或者网络发生故障时，系统需要具备一定的故障恢复能力。

2.5.1.3 可使用性

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。

2.5.1.4 密安性

通常来讲，实际使用的管理系统，必须具备相应的安全性能。该系统各级用户有各自的权限设置，例如学生之间不可以互相查看或修改其他人的成绩。同时该系统应该通过设置防火墙确保数据传输的安全。确保系统在一个安全可靠的环境中运行。

2.5.1.5 可维护性

软件的可维护性是指改进软件的难易程度。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。同时，该系统有良好的测试和诊断系统错误的功能。当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。

2.5.1.6 可移植性

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。

2.5.2 工程需求

2.5.2.1 设计约束

针对项目的基本要素进行分析，可以得到下面的设计约束表格。

表 2-4 项目设计约束条件

设计要素	主要约束
运行环境软件	操作系统
	数据库
	Web 服务器
用户端 PC 软件	操作系统
	浏览器
开发环境支持	操作系统
	开发工具
	Web 服务器
	CPU
	内存

2.5.2.2 逻辑数据库需求

数据库需求设计分为两部分：概念结构设计和逻辑结构设计。概念结构设计指的是画出 E-R 模型。将概念结构进一步转化为某一 DBMS 所支持的数据模型，然后根据逻辑设计的准则、数据的语义约束、规范化理论等对数据模型进行适当的调整和优化，形成合理的全局逻辑结构，并设计出用户子模式。这就是数据库逻辑设计所要完成的任务。

1. E-R 图设计：

设计论述：

(1) 学生实体和教师实体是多对多关系。一位教师会教授多位学生，一位学生也有可能在上多位教师的课。

(2) 学生实体和教务员实体是多对一关系。一位教务员可以管理多名学生，一位学生只会由一名教务员管理。

(3) 教师实体和教务员实体是多对一关系。一位教务员可以管理多名教师，一位教师只会由一名教务员管理。

(4) 教师实体和课程实体是多对多关系。一位教师可以开设多门课程，一门课程也可以由多位老师共同授课。

(5) 学生实体和课程实体是多对多关系。一位学生可以选择多门课程，一门课程也可以被多位学生选择。

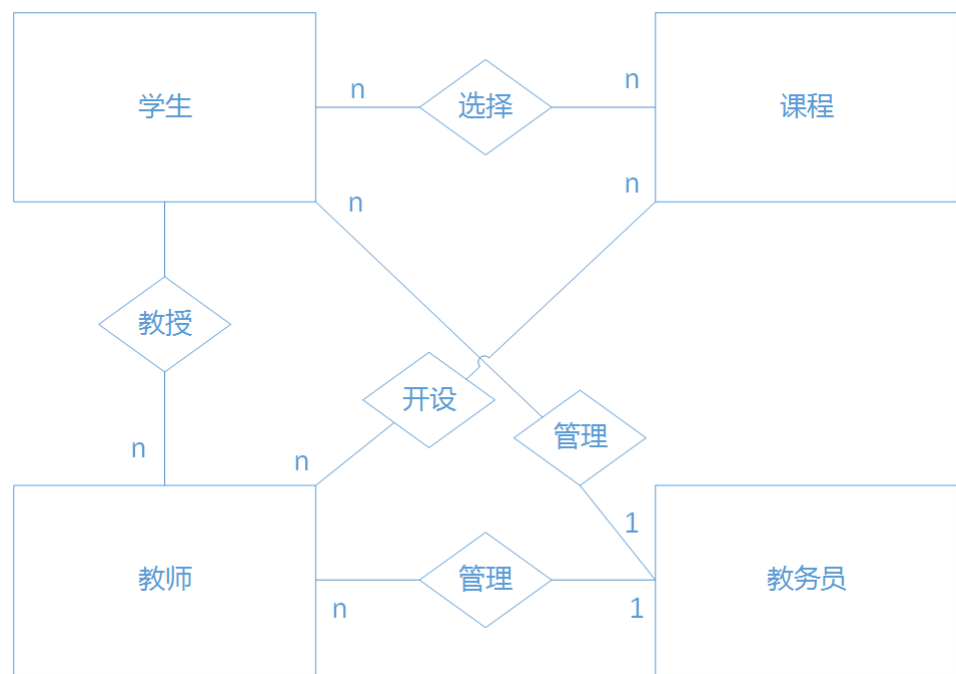


图 2-5 数据库 E-R 图

图例说明：

图形	含义
	实体
	关系

2. 数据库逻辑设计

根据 E-R 图可以建立的数据库逻辑结构如下：其中，“_____”位主键，“_____”为外键。

学生(学号，用户名，密码，受管理的教员工号)

教师(教师工号，用户名，密码，权限，受管理的教员工号)

教务员(教员工号，用户名，密码)

课程(课程 ID，课程名称)

开设课程(课程编号，授课教师工号)，(课程编号，授课教师工号)作为主键。

选择课程(课程编号，上课学生学号，考试成绩)，(课程编号，上课学生学号)作为主键。

2.5.3 其它需求

2.5.3.1 良好的人机交互能力

由于系统的用户较多，且操作习惯、年龄阶段、接受事物能力都各不相同，所以应要求该系统具备良好的人机交互能力。系统提供的各种功能能够便于用户理解，操作简单。

2.5.3.2 界面需求

系统应该具有可以适用于多平台、多浏览器的界面。

2.5.3.3 数据容量需求

由于用户量较大，该系统应该能够支持大数据分析处理。

2.6 风险

SGM 系统可能面临的主要风险有 3 类：技术性风险、工程性风险和环境性风险。下面将针对三中风险进行分析。

2.6.1 技术性风险

技术风险的种类很多，其主要类型是技术不足风险、技术开发风险、技术保护风险、技术使用风险。

1. 技术不足风险

SGM 系统需要用到的体系结构，涉及到 ASP，.NET 等相关技术以及 Java，C++ 等程序设计语言，在开发人员没有熟练掌握工程开发的技术以及相关设计模式的情况下，有可能会导致最终项目开发的缓慢延期，甚至停滞。

2. 技术开发风险

SGM 系统所用到的数据量是很庞大的，在开发过程中如果没有进行及时的备份以及一致性处理，将使最终用户不能有效地查询到相关信息。

3. 技术使用风险

SGM 系统要应用到网络平台，所以系统的数据安全也是项目需要关注的重点。如果用户在使用过程中不慎泄露相关安全数据，将会对系统造成严重伤害。

2.6.2 工程性风险

工程性风险主要包括：工程组织是否规范、工程管理是否混乱、在工程中是否使用了不称职的管理人员或工程技术人员。

2.6.3 环境性风险

环境性风险主要包括一些外界环境因素：

(1)SGM 的开发技术应该做到与时俱进。在必要的时候，我们应该更新系统开发所使用的技术和环境，使其使用最先进，最科学的开发平台与技术。

(2)同时在开发过程中，我们应该考虑到社会的法律、政策、道德规范等因素，比如用户隐私的保护。

(3)组织管理是否实现科学化，编写信息系统的程序员的文化知识素养有没有达到建设信息系统的需要，也是我们要考虑的环境因素。

(4)在系统开发过程中，我们应该考虑到一些可能存在的环境增长因素，比如增添了手机用户，有 10 万学生和教师同时使用该系统时可能存在的问题。

(5)最后，我们应该保证系统开发的一些外界影响因素，比如开发时间，开发预算等因素。只有这样，我们才能更高效更科学的开发这一系统。

3. 解决方案

3.1 相关的体系结构模式

本系统采用分层结构模式。分层系统体系结构有一下有点：

(1) 支持基于抽象程度递增的系统设计。这允许设计者可以将一个复杂系统设计按递增的步骤进行分解。

(2) 支持扩充。因为每层之多和与之相邻的上层和下层交互，所以，改变某层的功能最多只会影响与之相邻的其它两层。

(3) 支持重用。与抽象数据类型一样，只要对相邻层提供同样的接口，每层可以有很多不同的实现方法，并且这些方法可以相互替代。

下面将对一些典型的层次体系结构模式进行介绍和分析。

3.1.1 传统的信息管理系统开发模式(C/S)

传统的基于客户机/服务器(Client/Server)模式的管理信息系统产生于上世纪 70 年代并创立了一种分布式应用标准。到目前为止大多数的企业仍然在使用此种模式的信息管理系统，它为企业信息管理系统的共享集成和分布式应用做出了巨大的贡献，但是传统信息管理系统的缺点也是比较明显的：

(1) 安装、升级、维护工作量大。每个客户机安装一套应用软件，一旦出现一点微小的修改或版本升级就需要对每台机重装一次，这对客户端较多的大型系统来说，既费时有提高了软件成本。

(2) 数据一致性差。在 C/S 结构软件的解决方案中，不同地域的用户需要安装区域服务器，然后再进行数据同步。局部网络故障或者人为因素都有可能造成个别数据库不能同步，即使同步上来，各服务器也不是一个时点上的数据，数据很难做到一致性。

(3) 系统生命周期短，移植困难，升级麻烦。

3.1.2 B/S 信息管理体制开发模式

B/S 结构，即浏览器/服务器(Browser/Server)结构，是随着 Internet 技术的发展，对 C/S 结构的改进。在 B/S 结构中，客户端只需要安装 Web 浏览器软件，主要事务逻辑由服务器实现，用户通过浏览器向分布在网络上的服务器发出请求，服务器负责对客户端的请求进行分析、处理，并将反馈信息返回到浏览器。如下图所示。

因此可以看出，B/S 体系结构简化了客户端的工作，在该模式中，客户端只需要装上操作系统和浏览器，就可以在服务器上进行所需要的软件开发、维护等工作，极大地提高了工作效率。

与传统的 C/S 两层结构相比，B/S 应用体系的主要优势包括：

(1) 升级维护方便。

(2) 客户端负载轻。在客户端只需要标准的 Web 浏览器作为接入方式，而在各种平台上均有专门厂商提供的浏览器，从而简化了客户端配置。

(3) 资源访问简单。系统信息和资源以 HTML 标准进行组织，通过统一资源定位(URL)方式进行访问，并且访问点单一，允许在不同的地方访问数据库。

(4) 安全性高，应用逻辑和数据库由服务器实现，对客户端是非透明的，保证了系统的安全可靠性。

从上述对 C/S 和 B/S 开发模式的分析可以看出，基于 B/S 模式的 IMS 系统将会成为 IMS 发展的必然趋势。

3.1.3 体系结构选择

综合以上对 C/S 和 B/S 两种开发模式的分析，可以看出 B/S 模式具有适用范围广、异构和开放性强、平台技术稳定的特点。学生期末考试成绩管理系统需要网上办公和对信息进行高效管理，结合以上分析，本系统决定采用三层 B/S 结构进行开发。

3.2 体系结构概述

图 3-1 给出了一个简单的 B/S 结构的 IMS 系统所应包括的几个部分：浏览器、Web 服务器和数据库服务器。浏览器是用户输入数据和显示结果的交互界面，用户在浏览器表单中输入数据，然后将表单中的数据提交并发送给 Web 服务器，Web 服务器应用程序接收并处理用户的数据，并从数据库中查询用户数据或把用户数据录入数据库。最后 Web 服务器把返回的结果插入 HTML 页面，传送到客户端，在浏览器中显示出来。对于较为复杂的 IMS，可以将业务逻辑组件发布到一个单独的应用服务器来处理用户的数据。在本系统中，Web 服务器、应用服务器和数据库服务器配置在同一台计算机上。

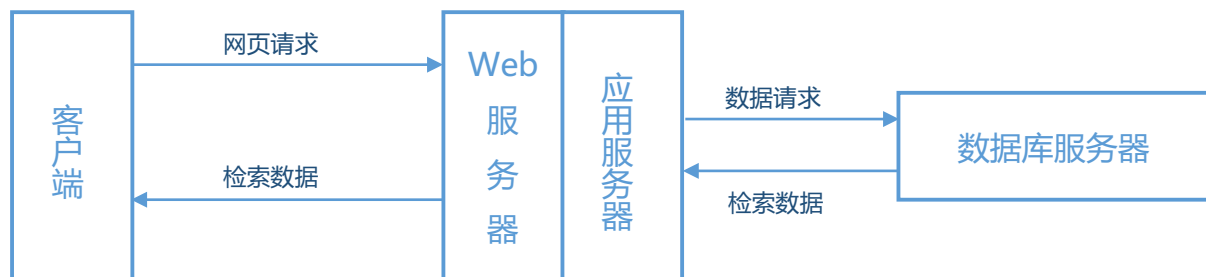


图 3-1 三层 B/S 模式的系统体系结构模型

3.3 结构化视图

本章节将体系结构进行分级，共分成了以下 3 类：

- (1) 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。
- (2) 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。
- (3) 运行级体系结构：描述系统的动态结构。

其中，(1)和(2)的体系结构描述采用结构化视图进行描述。(3)的描述采用行为视图进行描述。

3.3.1 概念级体系结构

概念级体系结构是从需求工程阶段给出的系统模型直接进化过来的。在需求阶段描述所期望的系统，概念级体系结构则说明所设计的元素以及元素领域的特定关系。针对 SGM 系统，可以设计出如下的概念级体系结构，该结构是由系统周境图和图 3-1 结合之后得出的，SGM 系统的部件根据不同的层次可以有如下划分：

表 3-1 SGM 部件划分

层次	所含部件
客户端层次	学生、教师、教务员、客户
Web 服务器层次	显示服务
数据库服务器层次	数据库管理服务

根据以上部件和应该提供的接口和服务，可以设计出如下的概念级体系结构：

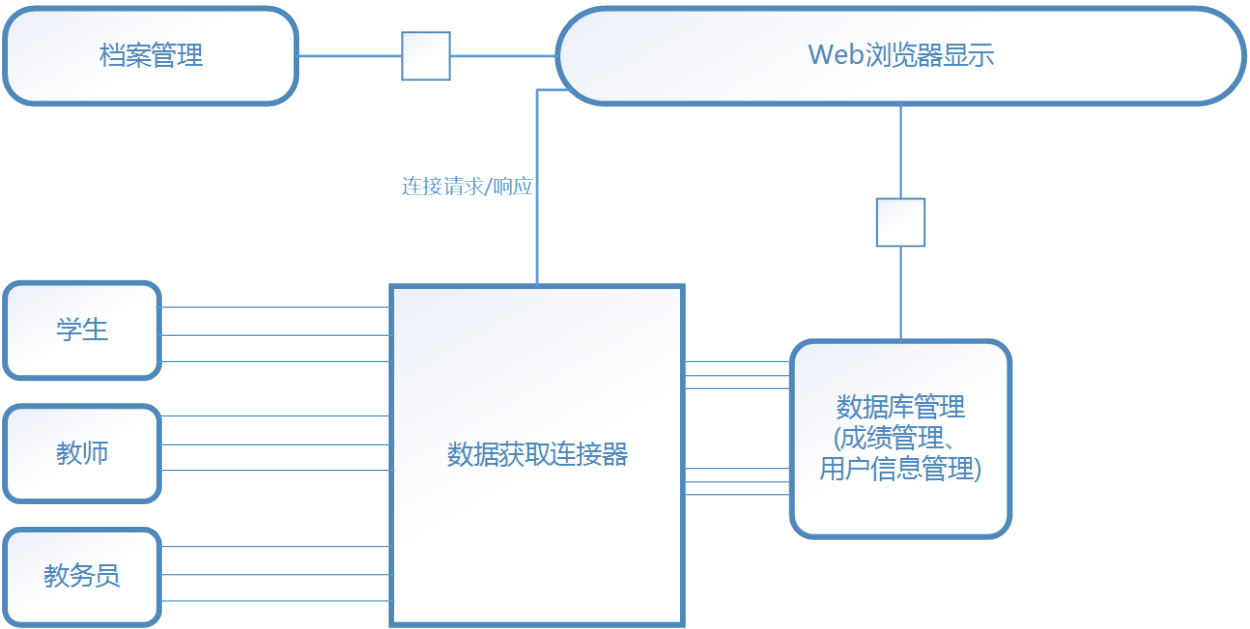


图 3-2 SGM 系统的概念级体系结构

图例如下所示：

图形	含义
	部件
	连接器

3.3.2 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；(1) 将系统按功能从逻辑上分解为系统、模块以及程序单元；(2) 按系统的层次进行划分。模块化分解方式很容易区分出概要设计和详细设计两个阶段。

根据上述两种方式，分别设计了 SGM 系统的功能分解体系结构图和层次体系结构图。

3.3.2.1 功能分解结构图

由于每个用户角色应具备的功能并不完全相同，所以将系统按照功能模块划分归类会造成数据流的错综复杂。因此该系统的第一层功能是根据用户角色进行划分的。分为四个模块：学生交互模块、教师交互模块、教务员交互模块和档案模块。



图 3-3 SGM 功能结构图

3.3.2.1.1 学生交互模块

学生交互模块是针对学生用户设计的。主要设计思想如下：

- (1) 系统要求学生用户在进入系统之前输入用户名和密码进行信息的验证。
- (2) 系统要求当密码校验失败时，会提示错误，如果超过系统允许的最大连续出错值，那么会对该用户进行锁定。
- (3) 系统应该能够实现让学生可以根据自己的喜好来修改自己的用户名称和密码的功能。
- (4) 系统应该能够实现让学生可以查看自己的各科期末考试成绩的功能。

3.3.2.1.2 教师交互模块

教师交互模块是针对教师用户设计的。主要设计思想如下：

- (1) 系统要求教师用户在进入系统之前输入用户名和密码进行信息的验证。
- (2) 系统要求当密码校验失败时，会提示错误，如果超过系统允许的最大连续出错值，那么会对该用户进行锁定。
- (3) 系统应该能够实现让老师在经教务员授权的情况下录入成绩，并修改错误成绩的功能。
- (4) 系统应该能够实现让老师查看选择了本人所授课程的学生的成绩的功能。

3.3.2.1.3 教务员交互模块

教务员交互模块是针对教务员用户设计的。主要设计思想如下：

- (1) 系统要求教务员用户在进入系统之前输入用户名和密码进行信息的验证。
- (2) 系统要求当密码校验失败时，会提示错误，如果超过系统允许的最大连续出错值，那么会对该用户进行锁定。
- (3) 系统应该能够实现让教务员对系统中人员信息的增删改查功能，人员包括：学生、教师、教务员。
- (4) 系统应该能够实现让教务员对教师录入成绩审定的功能。如果审定无误，系统应当实现让教务员对成绩进行发布，并给档案进行归档的功能。如果审定中出现错误，系统可以让教务员对教师授权进行修改。

3.3.2.2 系统分层结构图

针对本系统的实际管理情况以及图 3-1 的分析，设计该系统的分层结构如下图所示，其中最上层的是应用层(客户端层)，接下来是 Web 服务器层和应用服务器层，最下层是数据库服务器层。学生、教师、教务员等用户在客户端层可以进行成绩信息检索、用户信息管理服务；Web 服务器在 HTML 页面显示可视化的返回结果；数据库服务器内部进行数据的管理和处理工作。每层之间的数据交流与图 3-1 一致，在此省略。

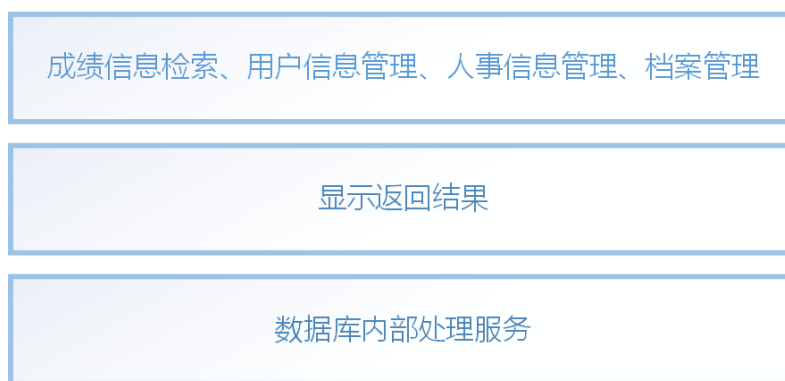


图 3-4 SGM 系统的分层结构

3.4 行为视图

行为视图主要包括以下两部分：(1) SGM 的运行级体系结构 (2)各子系统的消息序列图。

3.4.1 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，要尽可能描述运行元素、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、可移植性等元素。

图 3-5 是 SGM 系统的运行体系结构。成绩信息检索、用户信息管理等每个应用模块都被分配为操作系统的一个任务，每个模块可以具有多个运行实例。例如，每个学生都可以分配到一个进行成绩信息检索的线程或进程。

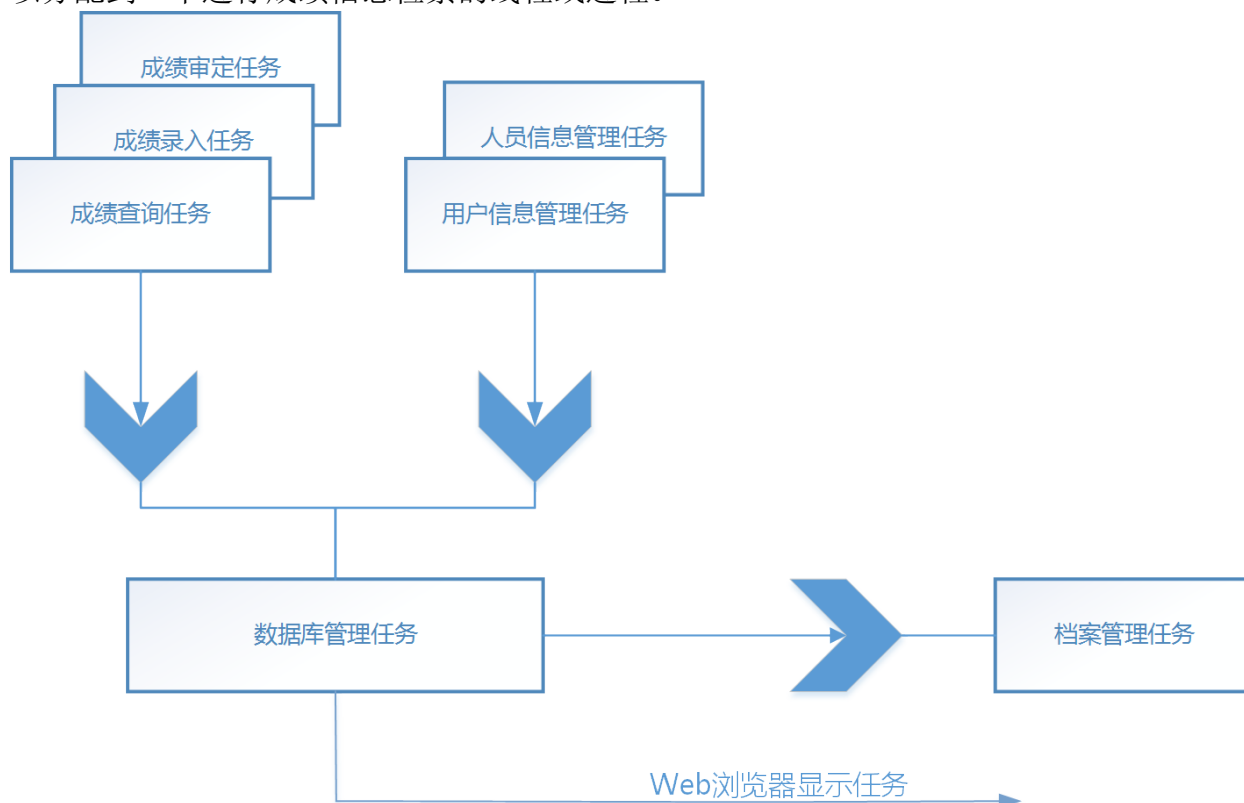




图 3-5 SGM 系统的运行体系结构

图例如下所示：

图形	含义
	操作系统的一个任务
	队列

3.4.2 各子功能模块的消息序列图

3.4.2.1 学生交互模块

学生用户和 Web 服务器以及数据库的交互过程论述如下：

- (1) 学生调用 Web 服务器，发送用户登录消息(此处包含用户信息的修改)
 - (2) Web 服务器收到学生的用户登录消息，并将相应的数据传送给数据库
 - (3) 数据库根据收到的数据和数据库中的数据进行对比，返回结果
 - (4) Web 服务器收到数据库返回的结果并在页面上显示，反馈给学生用户
 - (5) 学生发送成绩查询消息
 - (6) Web 服务器收到学生的成绩查询消息，并将相应的数据传送给数据库
 - (7) 数据库根据收到的数据进行查询，返回结果
 - (8) Web 服务器收到数据库返回的结果并在页面上显示，反馈给学生用户
- 如上过程可以描述为以下的消息序列图：

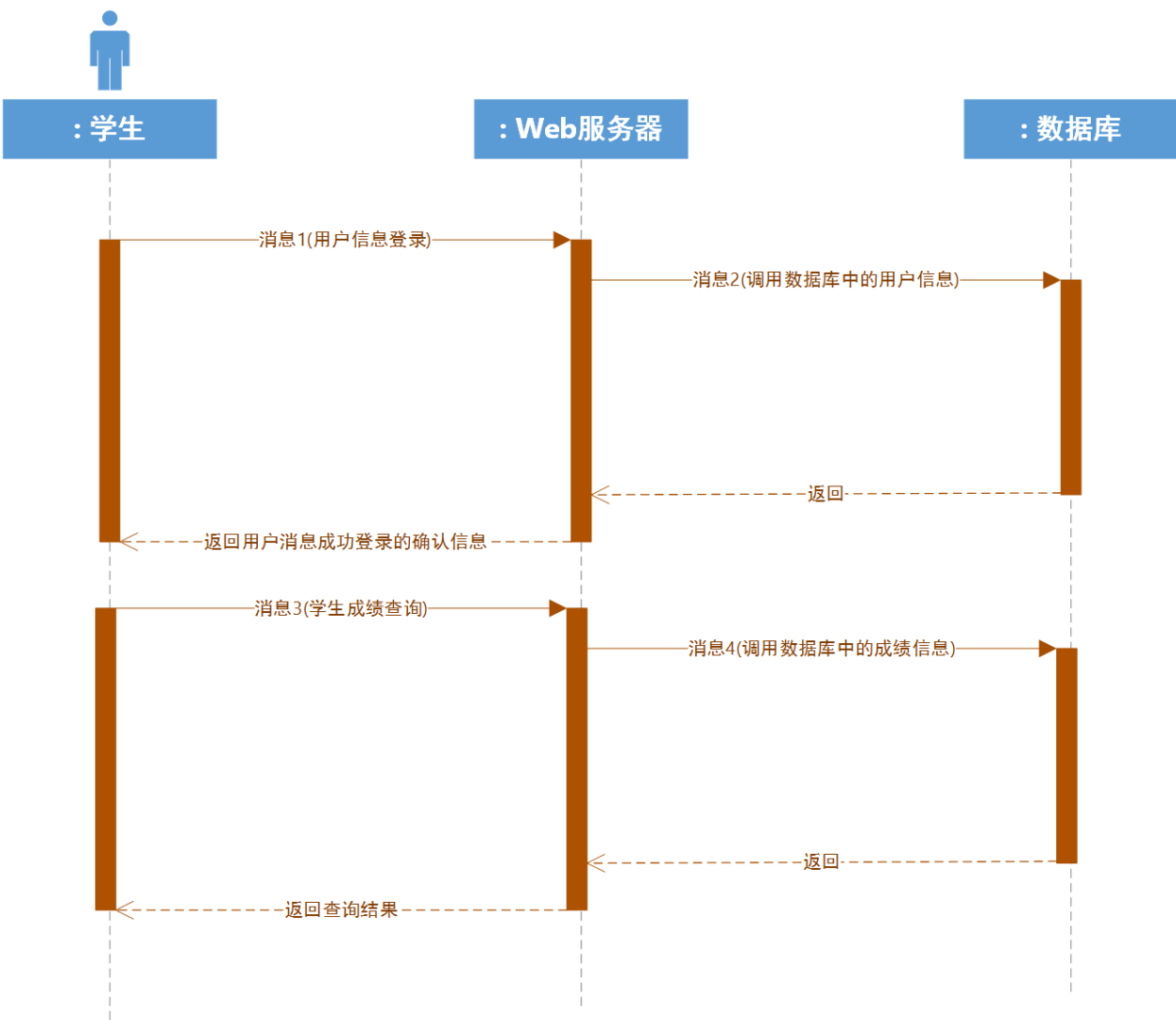
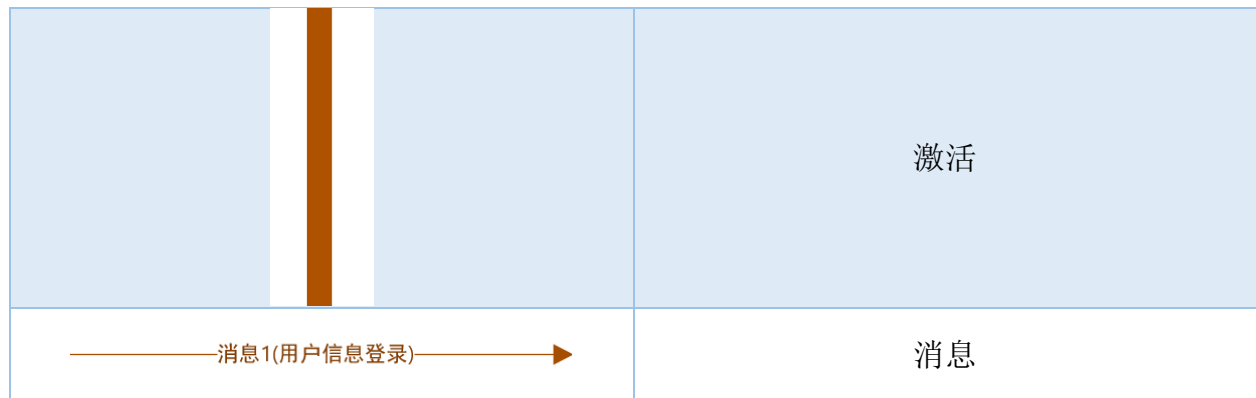


图 3-6 学生模块消息序列图

图例如下所示：

图形		含义	
			参与者
			对象



3.4.2.2 教师交互模块

教师用户和 Web 服务器以及数据库的交互过程论述如下：

- (1) 教师调用 Web 服务器，发送用户登录消息(此处包含用户信息的修改)
 - (2) Web 服务器收到教师的用户登录消息，并将相应的数据传送给数据库
 - (3) 数据库根据收到的数据和数据库中的数据进行对比，返回结果
 - (4) Web 服务器收到数据库返回的结果并在页面上显示，反馈给教师用户
 - (5) 教师发送成绩录入消息
 - (6) Web 服务器收到教师的成绩录入消息，并将相应的数据传送给数据库
 - (7) 数据库根据收到的数据进行数据更新，并将相应的结果返回给教务员
 - (8) 教务员对教师录入的消息进行审核，判定消息是否有误，并将判断结果返回给 Web 服务器
 - (9) Web 服务器将返回的结果返回给教师，如果成绩无误，则此次操作结束
 - (10) 如果成绩有误，教师需要修改成绩，并重复(5)-(9)过程，直到成绩无误
 - (11) 教师发送成绩查询消息
 - (12) Web 服务器收到教师的成绩查询消息，并将相应的数据传送给数据库
 - (13) 数据库根据收到的数据进行查询，返回结果
 - (14) Web 服务器收到数据库返回的结果并在页面上显示，反馈给教师用户
- 如上过程可以描述为以下的消息序列图：

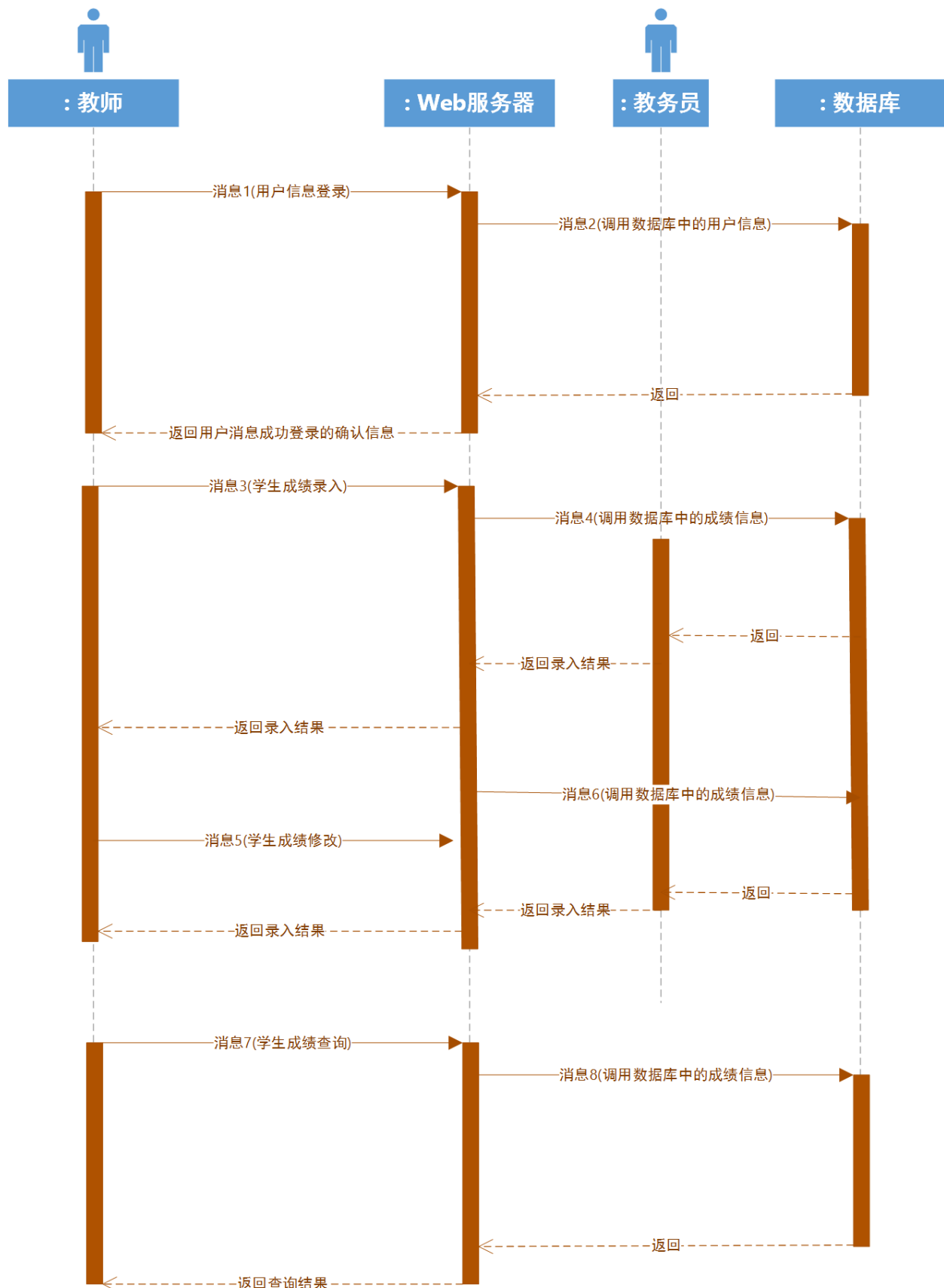






图 3-7 教师模块消息序列图

图例如下所示：

图形	含义
	参与者
	对象
	激活
	消息

3.4.2.3 教务员交互模块

教务员用户和 Web 服务器以及数据库的交互过程论述如下：

- (1) 教务员调用 Web 服务器，发送用户登录消息(此处包含用户信息的修改)
- (2) Web 服务器收到教务员的用户登录消息，并将相应的数据传送给数据库
- (3) 数据库根据收到的数据和数据库中的数据进行对比，返回结果
- (4) Web 服务器收到数据库返回的结果并在页面上显示，反馈给教务员用户
- (5) 教务员用户收到成绩录入消息的返回结果，进行成绩审定后将结果发送给 Web 服务器
- (6) Web 服务器将返回的结果返回给教师，如果成绩无误，则此次操作结束
- (7) 如果成绩有误，教师需要修改成绩，并重复(5)-(9)过程，直到成绩无误
- (8) 教务员发送成绩查询消息
- (9) Web 服务器收到教师的成绩查询消息，并将相应的数据传送给数据库
- (10) 数据库根据收到的数据进行查询，返回结果
- (11) Web 服务器收到数据库返回的结果并在页面上显示，反馈给教务员用户
- (12) 教务员对人事信息进行修改
- (13) Web 服务器收到教务员的人事信息，并将相应的数据传送给数据库

(14) 数据库根据收到的数据对数据库进行更新，返回结果

(15) Web 服务器收到数据库返回的结果并在页面上显示，反馈给教务员用户
如上过程可以描述为以下的消息序列图：

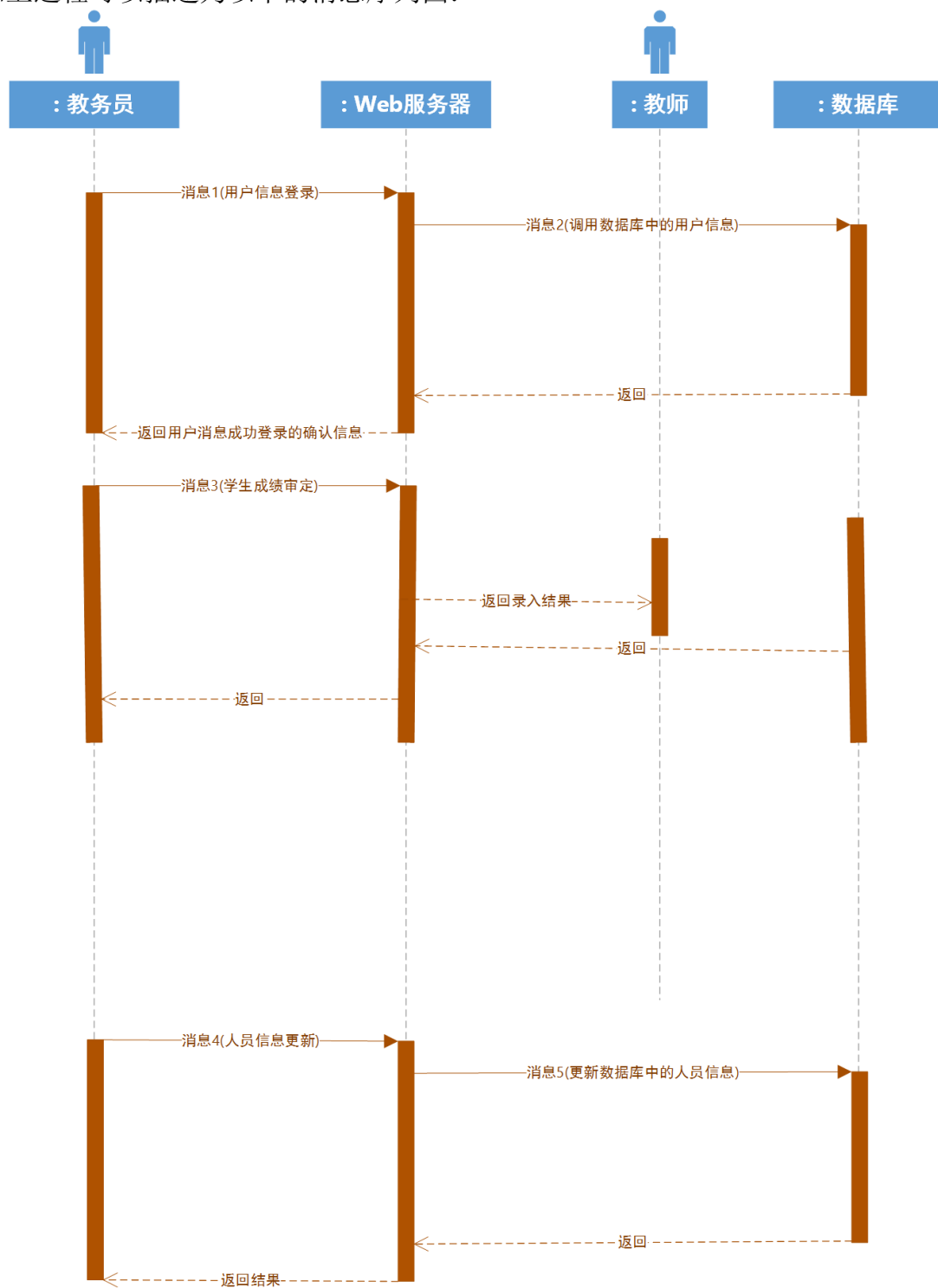
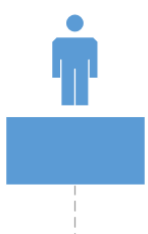





图 3-8 教务员模块消息序列图

图例如下所示：

图形	含义
	参与者
	对象
	激活
	消息

3.5 实现问题

针对 2.6 节中的风险分析，得出该体系结构可能存在的实现问题如下：

- (1) 相关人员应该对该系统所使用的开发结构、平台以及相关技术十分熟练
- (2) 当学校人数上升时，系统本身可以并发响应的用户有可能达不到此时实际的并发用户。我们不可能无上限地设置并发用户。因此，当发生这种情况时，我们应该设置相应的优先级队列进行处理，比如先到先处理，同时，对有一段时间没有响应处理的线程任务，我们应当将其暂时挂起。
- (3) 可能存在部分用户通过手机进行访问，这个时候应该考虑到要对相应的一些手机系统的浏览器进行接口完善，防止无法访问的情况发生。比如，在 QQ 中打开某一网址时，QQ 会自动在网址后面添加一些参数，如果系统本身对参数没有处理，有可能导致手机用户无法访问。
- (4) 在系统开发的过程中，要对开发周期和开发预算有严格的限制，在必要的情况下可以对这些环境因素进行调整，但是并不能无节制的调整和延期。

4. 质量的分析与评价

4.1 场景分析

场景分析通过分析软件应用的场景，从用户的角度出发，从场景的角度来设计测试用例，是一种面向用户的测试用例设计方法。

4.1.1 场景设计

通过对需求分析报告中的 DFD 图的分析，可以得出在 SGM 系统中，不同场景下的数据流走向，从而得出，SGM 系统的场景共可以分为以下几种：

表 4-1 场景设计

场景编号	场景描述
场景 1	学生用户登录系统，查询成绩
场景 2	教师用户登录系统，查询成绩
场景 3	教师用户登录系统，录入成绩成功
场景 4	教师用户登录系统，录入成绩失败
场景 5	教师用户修改成绩
场景 6	教务员用户登录系统，审定成绩
场景 7	教务员用户登录系统，修改人员信息

4.1.2 用例设计

针对表 4-1 中的 7 中不同场景，可以设计出以下几种测试用例，在输入数据部分，T 代表输入正确数据，F 代表输入错误数据，空代表不输入数据。

表 4-2 用例设计

所属场景 编号	用例编号	输入数据				预期结果
		用户名	用户密码	课程名称	课程成绩	
场景 1	用例 1	T	T	T		学生登录成功， 返回成绩
	用例 2	T	F			学生登录失败， 返回错误信息
	用例 3	F	T			学生登录失败， 返回错误信息
场景 2	用例 4	T	T	T		教师登录成功， 返回成绩
	用例 5	T	F			教师登录失败， 返回错误信息
	用例 6	F	T			教师登录失败， 返回错误信息
场景 3	用例 7	T	T	T	T	教师登录成功， 录入成绩正确

场景 4	用例 8	T	T	T	F	教师登录成功， 录入成绩错误
场景 5	用例 9	T	T	T	T	教师登录成功， 修改成绩成功
场景 6	用例 10	T	T	T		教务员登录成功， 审定成绩
	用例 11	T	F			教务员登录失败
	用例 12	F	T			教务员登录失败
场景 7	用例 13	T	T			教务员修改人员 信息

4.2 原型分析

原型法（Prototyping）的理念是指在获取一组基本需求之后，快速地构造出一个能够反映用户需求的初始系统原型。让用户看到未来系统的概貌，以便判断哪些功能是符合要求的，哪些方面还需要改进，然后不断地对这些需求进一步补充、细化和修改。依次类推，反复进行，直到用户满意为止并由此开发出完整的系统。简单的说，原型法就是不断地运行系统的“原型”来进行揭示、判断、修改和完善需求的分析方法。

（1）快速分析，弄清用户的基本信息需求。

SGM 系统中，用户的基本信息需求有：

- ① 系统应该保证用户能够正常登录系统，查询相关信息。
- ② 系统应该保证用户的隐私信息，如密码不会泄露。
- ③ 系统应该对用户设置权限。

（2）构造原型，开发初始原型系统。

采用了 Balsamiq Mockups 设计出了该项目的初始原型。

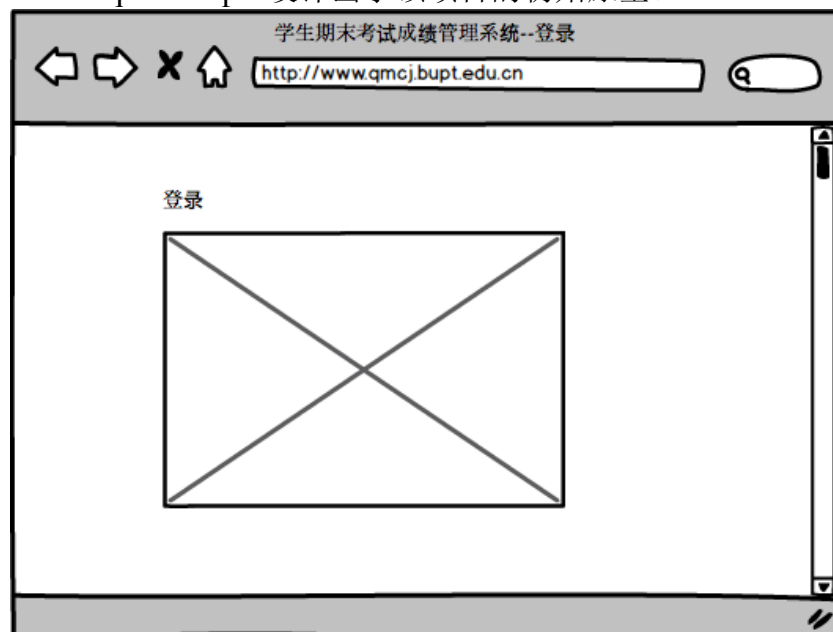


图 4-1 SGM 系统登录界面



图 4-2 学生界面



图 4-3 教师界面



图 4-4 教员界面

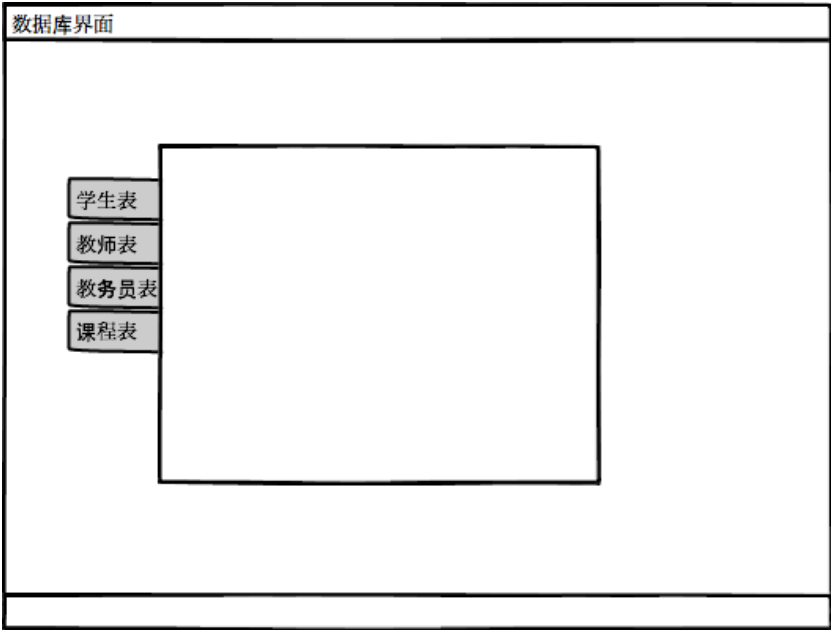


图 4-5 后台数据库界面

5. 同行评审

表 5-1 同行评审意见表

评审员	评审意见
何利莉	该文档让阅读的人能够直观全面的认识到该系统的结构，无需深刻的专业知识也能了解到系统的各方面。
冯宇晖	该文档对系统的需求分析较为全面，对系统结构的分析还算准确，但是对系统结构的分层有些模糊，不够清晰。

A. 附录

A.1 附录 1

某校有 3 万学生，2000 个老师，每学期开出 500 门课，平均一门课有 100 个学生。

编写一份“学生考试成绩管理系统”需求分析报告，起码包括如下功能：

角色：学生、教师、教务员

成绩只包括：期末成绩（为了简单）

学生：能查询各门课的成绩

教师：能录入成绩

教务员：可以审定和发布成绩

教务员：可以给老师授权修改成绩，一旦发现成绩有错。

编写该系统的体系结构文档

作业准备：

1) 复习 11.1 节（体系结构分级，不写代码级）.....17

2) 建议读书的第 26 章 26.4-26.5 节

3) 可参考模板（图 10-6），封面格式（参照需求文档），交电子版和打印版

注意要回答出：

1) 选择体系结构的理由？.....17

2) 你为何会选择三层（多层）B/S 结构？.....17

3) 你的体系结构，对质量属性是如何影响的？.....12

4) 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？..... 29

5) 考虑其他可能的（增长性）场景，例如，用手机.....29