

学生考试成绩管理系统

体系结构设计规范

V1.0

2016.12.4

李璐尧 廖凌飞

北京邮电大学软件学院

学号：2014212087 2014211984

班级：2014211504 2014211501

软件工程导论

2016.秋

修订记录

日期	版本描述	作者	评价
2016.12.4	V1.0	李璐尧、廖凌飞	基本完成了对体系结构的设计和分析

文档审批

本体系结构设计规范已经通过以下审批：

签名	姓名	标题	日期

修订记录	1
文档审批	1
1 引言	3
1.1 编写目的	3
1.2 适用范围	3
1.3 缩略语记号及含义	3
1.4 参考文献	4
1.5 文档概述	4
2 体系结构需求	4
2.1 关键指标	4
2.2 体系结构用例	5
2.2.1 教师用例	5
2.2.2 学生用例	5
2.2.3 教务员用例	6
2.2.4 教务长用例	7
2.2.5 系统管理员用例	7
2.3 约束条件	8
3 解决方案	9
3.1 相关的体系结构模式	9
3.2 体系结构概述	10
3.3 结构化视图	14
3.4 实现问题	19
4 系统的质量分析和评价	20
4.1 场景分析	20
4.2 原型分析	22
4.3 效用树分析	23
4.4 风险	23

1 引言

1.1 编写目的

本系统结构设计规范是学生考试成绩管理系统的体系结构设计说明书，主要描述了系统的功能构件、构件之间的连接件、构件受到的系统约束、能够理解系统行为的语义模型。

体系结构会直接影响到系统质量属性。本文档是相关各方(客户、项目经理、开发人员、测试者和维护人员等)进行沟通的基础和共同语言，也是早期决策的基础，同时是编写测试用例和进行集成测试的主要依据。这就是本文档编写的目的。

文档预期读者包括：

- ✓ 客户。了解项目开发进程和经费预算；
- ✓ 项目经理。通过体系结构设计分割开发任务；
- ✓ 开发人员。关心体系结构如何实现；
- ✓ 测试和维护人员。以体系结构为依据，编写测试用例；
- ✓ 最终用户。审核系统设计是否满足了预期功能和实际需求。

1.2 适用范围

本文档适用于学生考试成绩管理系统。系统主要使用者和功能概述如下：

- 学生。选择学期，查询整个学期内所修读课程的分數；
- 教师。按课程考核指标在合法时间段内录入分数，将成绩单提交至教务长。提交后需要修改成绩，需要提交申请，经教务长允许后可修改；
- 教务长。对教师提交的成绩单进行签字认定后，交给教务员发布。另外，控制教师修改成绩的权限；
- 教务员。发布成绩；
- 系统管理员。管理系统用户的权限。

1.3 缩略语记号及含义

文档中使用的缩略语记号及含义见下表。

记号	含义
SGMS	学术考试成绩管理系统的缩写

表 1 缩略语记号及其含义

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

1.5 文档概述

本文第一章，介绍了文中引用的术语和编写目的；第二章介绍系统体系结构需求和约束；第三章对系统体系结构的设计方案进行详细描述；第四章给出对系统体系结构质量进行分析和评价的方法。

2 体系结构需求

2.1 关键指标

SGMS 系统的关键指标及指标描述见下表。

质量属性指标	描述
性能	请求被响应的速度
易用性	用户使用系统的难易程度
可伸缩性	系统是否是可伸缩的
可修改性	系统修改的难易程度
可使用性	系统在不同环境下可以正常使用的程度
密安性	系统受到攻击而存活概率的高低
可靠性	系统信息保存能力的强弱

表 2 SGMS 系统关键指标及描述

2.2 体系结构用例

2.2.1 教师用例

教师用例图如下。

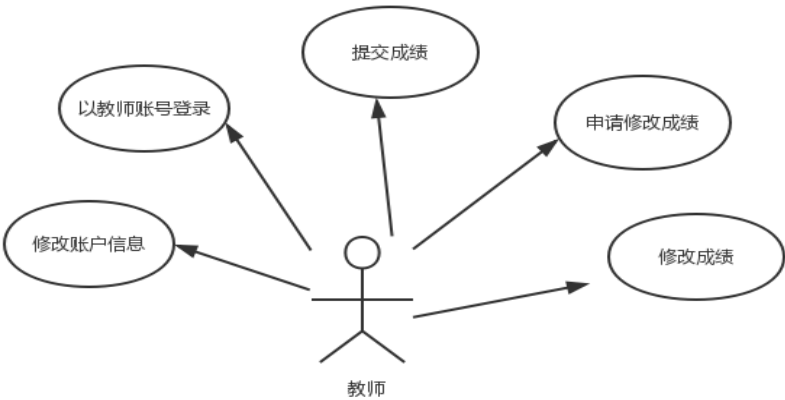


图 1 教师用例

教师用例对系统的体系结构需求如下表。

体系结构需求	描述
登录	系统应当允许教师登录
修改账户	系统应当允许教师修改账户信息
提交成绩	系统应当允许教师提交成绩
修改成绩	系统应当允许教师提交成绩
申请权限	当教师发现成绩有问题时，系统应当允许教师申请修改成绩的权限

表 3 教师用例体系结构需求

2.2.2 学生用例

学生用例图如下。

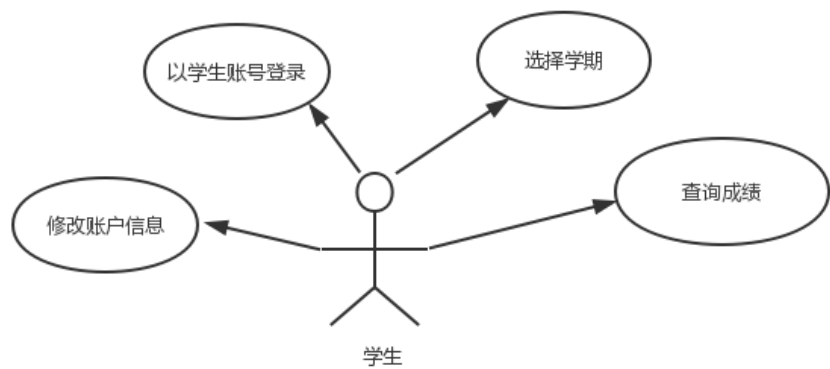


图 2 学生用例

学生用例对系统的体系结构需求见下表。

体系结构需求	描述
登录	系统应当允许学生登陆
修改账户	系统应当允许学生修改账户
查询成绩	系统应当允许学生查询指定成绩

表 4 学生用例体系结构需求

2.2.3 教务员用例

教务员用例图如下。

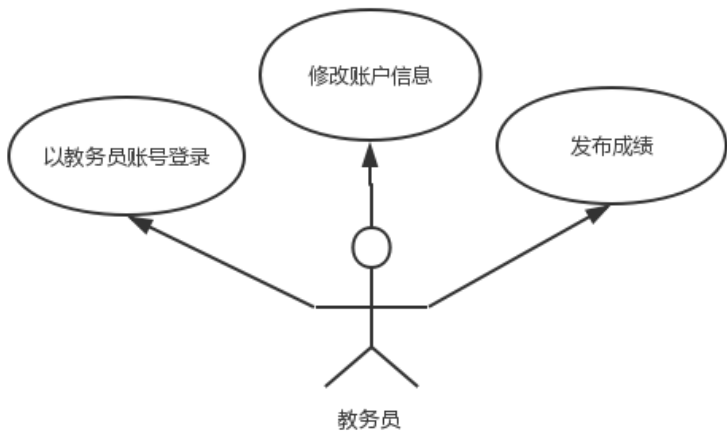


图 3 教务员用例

教务员对系统体系结构需求如下表。

体系结构需求	描述

登录	系统应当允许教务员登陆
修改账户	系统应当允许教务员修改账户
查看和发布成绩	系统应当允许教务员查看待发布成绩并发布

表 5 教务员用例体系结构需求

2.2.4 教务长用例

教务长用例如下图。

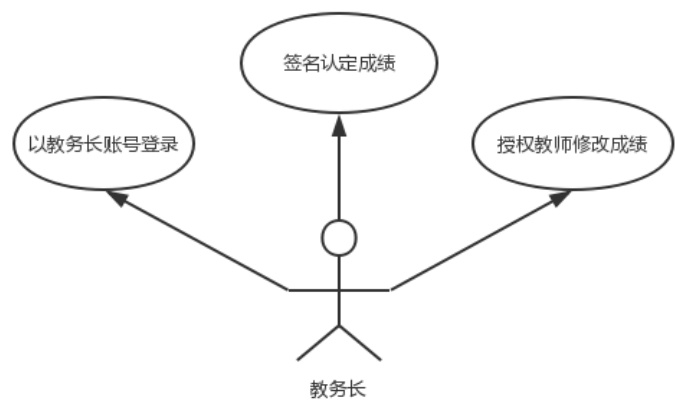


图 4 教务长用例

体系结构需求	描述
登录	系统应当允许教务长登陆
修改账户	系统应当允许教务长修改账户
认定成绩	系统应当允许教务长查看待认定成绩并认定

表 6 教务长体系结构需求

2.2.5 系统管理员用例

系统管理员用例如下图。

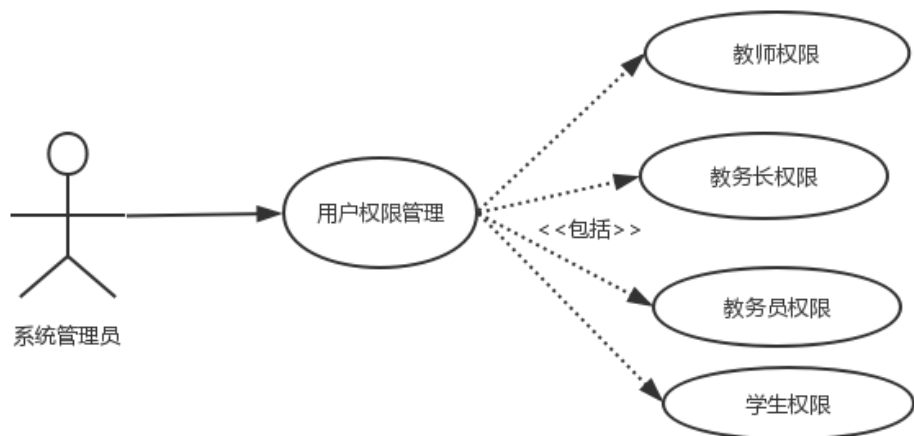


图 5 系统管理员用例

系统管理员用例对系统的体系结构需求如下表。

体系结构需求	描述
登录	系统应当允许系统管理员登陆
修改账户	系统应当允许系统管理员修改账户
管理系统用户	系统应当允许系统管理员创建新用户，查看现有系统用户，删除现有系统用户
修改用户权限	系统应当允许系统管理员修改各个用户身份的权限

表 7 系统管理员体系结构需求

2.3 约束条件

SGMS 系统约束条件如下表。

质量属性	系统约束描述
性能	对 95% 的请求，系统必须在 3 秒以内做出相应
易用性	客户端必须支持浏览器，兼容 IE 6.0，以保证远程客户不需要安装软件
可伸缩性	系统在高峰时期能够至少处理 1000 个并发用户的请求
资源管理	服务器部件的内存必须具有 40% 以上的预留量
可修改性	体系结构必须支持从当前的第 4 代语言向 .NET 或 J2EE 环境的过滤

可使用性	系统宕机频率不高于 1 次/月，每次宕机恢复时间不得高于 5min
密安性	用户在登录认证过程中，与系统进行通讯的信息均需要加密
可靠性	所有事务处理必须在 30 秒内完成

表 8 SGMS 的系统约束

3 解决方案

3.1 相关的体系结构模式

3.1.1 C/S 体系结构

C/S 体系结构是一种常用的网络体系架构，组成部分包括客户端(client)和服务端(Server)。客户端封装了一套访问服务器和接受服务端反馈的接口，是用户唯一可见的系统终端。客户端将用户的逻辑业务请求发送给服务器，并向用户反馈服务器响应的请求结果。服务器由 web 服务器和数据库服务器组成，主要处理客户端提交的请求，并返回处理结果。客户端和服务端二者独立运行，通过相关接口和协议通信，完成一套完整的业务。

消息流动和处理对于用户是透明的，就如同客户端独立完成整个操作一样。这也是 C/S 体系结构最大的优势。

C/S 体系结构如下图。

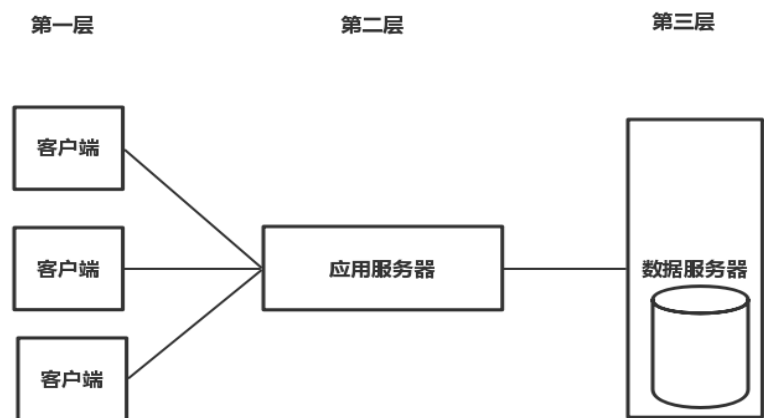


图 6 C/S 体系结构

3.1.2 B/S 体系结构

从某些意义上说，B/S 体系结构可以看作是 C/S 构架在新的技术支持下的产物。

B/S 构架同样采用三层模式，其显著特点在于，客户端舍弃了繁琐且性能低下的客户表现，采用广泛而通用的 www 浏览技术，使用通用的网络技术和网络协议实现原来客户端的所有功能。

B/S 体系结构如下图。

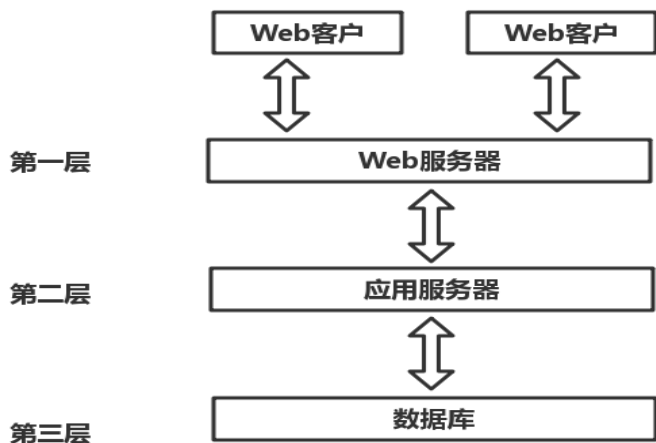


图 7 B/S 体系结构

由上图可知，B/S 架构由 Web 浏览器层(表示层)、Web 服务器(逻辑业务服务器层)和数据库服务器层构成。每层均能够独立完成工作，业务能够在本层得到处理或者向下一层发送请求。

B/S 体系结构的优势在于其配置简单、升级容易、可移植性良好、平台通用的特性。

3.1.3 MVC 设计模式

MVC 模式是一种业界流行的设计模式。模式用业务逻辑、数据、界面显示分离的方法来组织程序。当用户所需的应用数据发生改变时，能够自动且灵活得反映到用户界面上。这样一来，在改进和个性化定制用户界面的同时，省去重新编写应用逻辑的繁杂工作。

MVC 中 M(Model)是指业务模型，V(View)是指用户界面，C(Controllor)则是控制器。M 和 V 实现代码分离，从而使同一套应用逻辑可以使用不同的表现形式。C 的功能是确保 M 和 V 的同步，一旦 M 改变，V 应该同步更新。

3.2 体系结构概述

3.2.1 体系结构视角

引用下图中“4+1”视图模型，分别从逻辑、实现、进程和部署 4 个角度，外加用例角度，看待软件体系结构。

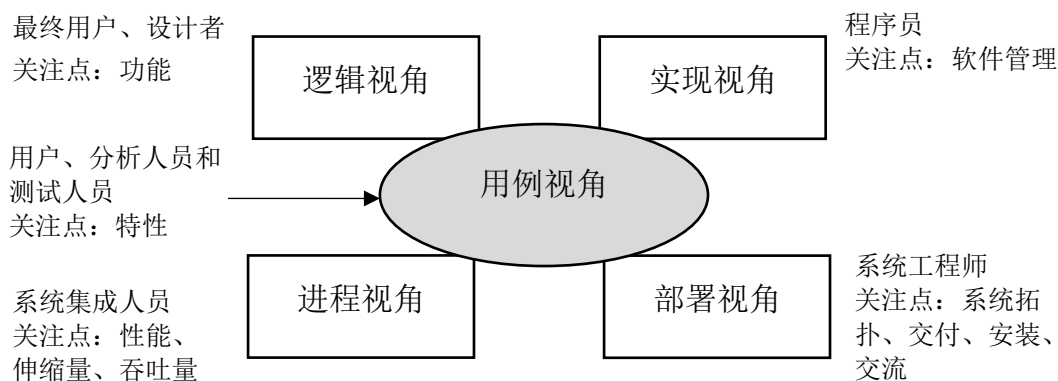


图 8 软件体系结构视角

3.2.1.1 逻辑视角

SGMS 的最终用户和设计者关注系统的功能。对于最终用户而言，系统需要简单易用，并且能够完成他们各自的业务。就易用性而言，浏览器是不二选择。设计者应该考虑到系统体系结构对性能的影响，使得系统能够稳定、高效地完成业务需求。

3.2.1.2 实现视角

从程序员的系统实现视角来看，轻量的体系结构是受到青睐的，这样实现过程会更加容易，软件管理也会更加容易。

3.2.1.3 进程视角

站在系统集成人员的角度，SGMS 系统应当对 95% 的用户请求在 3 秒内给出响应，系统的可伸缩性应当能够在高峰时期满足 1000 个用户的并发请求，并且当用户数目达到 10 万时，能够通过小于 2 个月的升级和维护，使得系统满足非功能需求。

3.2.1.4 部署视角

系统工程师重视系统的可移植性，希望系统能够在多个不同的平台和服务器上简单安装、顺利运行。从这个角度而言，系统的环境需求和环境依赖应当较少，并且使用包含多平台接口的通用中间件。

3.2.1.5 用例视角

系统的不同用例场景要求系统有着不同的特性。SGMS 系统用例包括学生、教师、教务长、教务员、系统管理员用例。

- 学生用例要求系统能够允许学生登录、修改账户信息、查询指定学期的所修读课程的分

- 教师用例要求系统能够允许教师登录、修改账户信息、在合法时间段录入相应考核指标的学生分数、提交分数、提交成绩单后能够向教务长提交修改申请，得到授权后能修改学生分数；
- 教务长用例要求系统能允许教务长登录、修改账户信息、对教师提交的成绩单进行签字认定、授权教师修改已提交的学生成绩；
- 教务员用例要求系统能允许教务员登录、修改账户信息、发布经过认定的成绩；
- 系统管理员用例要求系统能允许系统管理员登录、修改账户信息、管理用户权限。

3.2.2 体系结构设计

基于上述分析，SGMS 系统采用 B/S 体系结构。理由如下：

- C/S 体系结构通常采用内部一致的标准，其主要应用于基于局域网的项目开发。而 B/S 标准是经过标准化组织制定，行业共同遵循。因此，B/S 体系架构具有通用性和跨平台性，更具开放性。
- 基于 C/S 架构开发的系统则需要独立的表现形式，用户访问形式单一，灵活性差，且开发过程繁琐。相比之下，基于 B/S 架构所开发的系统适用于互联网上的用户广泛访问，只需要浏览器便可以使用系统。灵活性高，且综合性能优于前者。
- 由于没有客户端的系统访问方式限制，对于用户而言，省去了一系列繁杂的操作，使用便捷；对于开发人员而言，省去了一系列的开发工作，节约了人力成本。
- 基于 B/S 架构开发的系统更新和维护容易，更新时只需要更新服务端返回给表现层的界面，只要用户访问的地址不变即可，更新和维护，对用户是透明的。而基于 C/S 构架开发的系统，更新时需更新客户端，维护成本同样也高。

综上所述，本系统采用 B/S 体系结构设计。系统体系结构见下图。

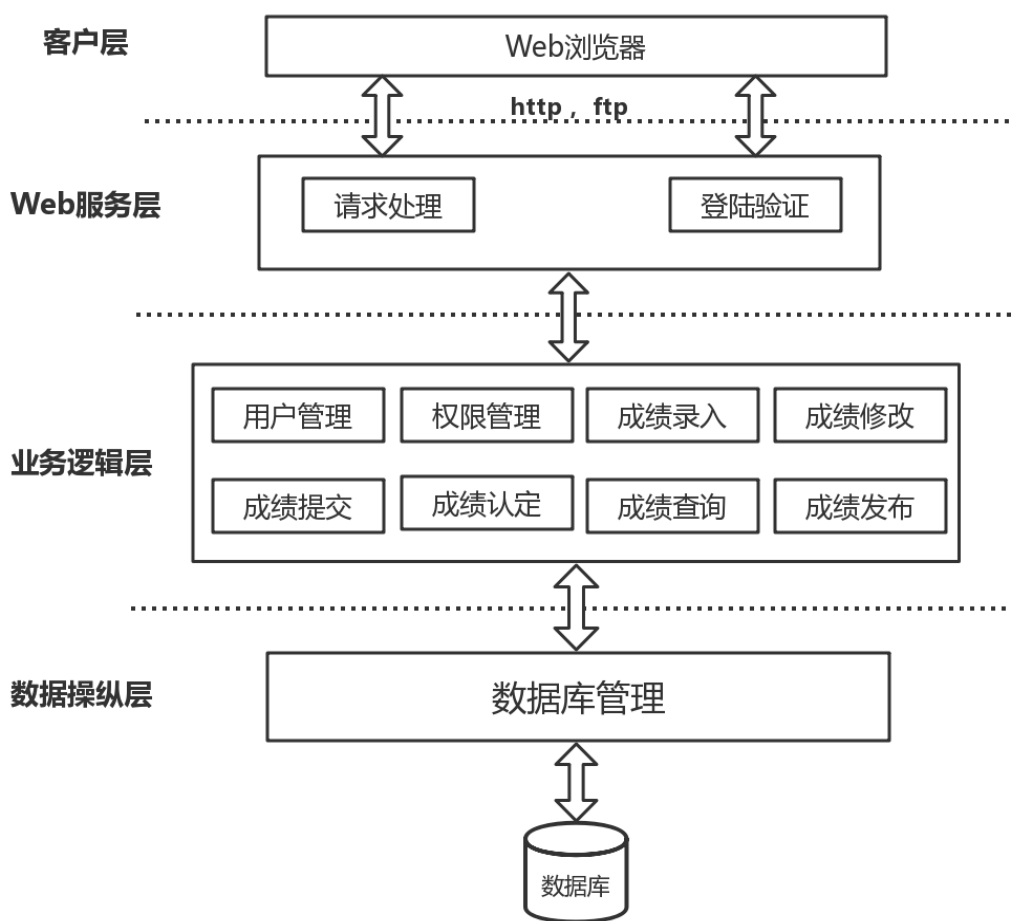


图9 SGMS 系统体系结构

系统所采用的此 3+1 层 B/S 架构设计，使得各个层之间的操作相互独立，不同的请求采用不同的处理模块进行业务处理，使得功能之间界限清晰，各个模块协作流程，内聚程度高，耦合程度低，使得系统容易拓展新的模块功能，可拓展性良好。

- 客户层。即客户访问系统所用的浏览器，用户在此层完成信息查看，请求提交功能，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，用户需要在这个层完成登录页面，操作页面的表示；客户层需要能够提交请求，获取服务器回复，并在相关页面上进行表示。
- Web 服务层：通过用户上传的账号和密码和数据库已有用户进行比对，能够验证用户登陆，根据用户身份的不同，如老师或者学生或者教务员，反馈回不同的成功登陆界面或失败登陆界面；能够处理不同用户的请求，如成绩查询，成绩修改，成绩提交等，接受用户请求，发送给业务逻辑层，在业务逻辑层处理完成后，获得业务层处理结果，并将处理结果反馈给客户层。
- 业务逻辑层：能够将服务层传递进的用户请求过滤分类处理，调用不同的模块完成不同的请求，如学生查询成绩的请求，要调用成绩查询模块，教务员发布成绩，要先调用成绩查询模块，获得待发布成绩，再调用成绩发布模块，发布成绩。

- 数据操纵层：业务逻辑层涉及数据库操作的任务时，会发送给数据操纵层，完成数据读取或写入,并将结果返回给业务逻辑层，比如学生的成绩查询请求，需要业务逻辑层向数据操纵层发送读取数据的请求，等待数据操纵层返回查询结果后，才能继续完成业务。

3.3 结构化视图

3.3.1 概念级体系结构

系统的概念级体系结构如下图。

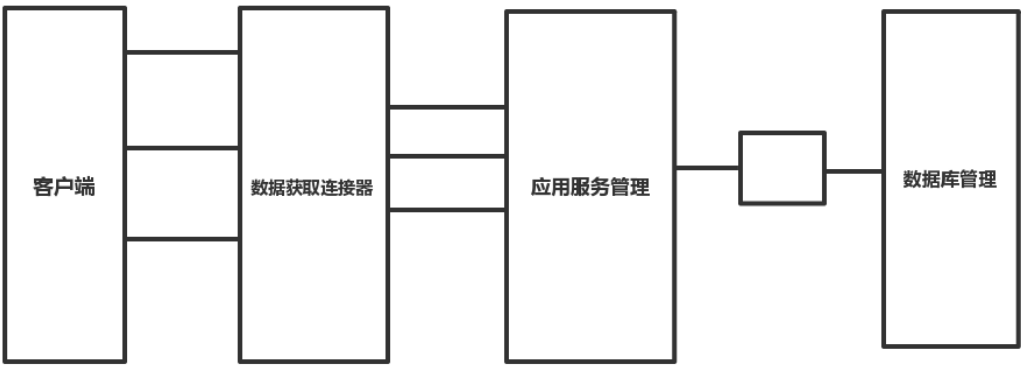


图 10 SGMS 系统概念级体系结构

上图所示的概念图中，各个部件和连接器设计如下表。

部件或连接器	功能与约束	选用领域部件	分析评估	影响因素
客户端	功能： 表现界面，支持用户登陆和数据请求服务 约束： 简易，不需要安装，能够多平台运行，表现界面优良，功能全面	火狐浏览器（内核 IE7 以上），flash 插件，HTML5	性能： 能够在 1 秒内渲染完成页面 安全性： 页面不会加载其他脚本 可靠性： 99% 的页面加载不会崩溃	实时操作系统，内存容量，显卡版本

数据获取连接 器	功能: 传输数据 约束: 数据传输安全性要强, 数据流动速度快, 数据传输需要有保障, 数据在服务器和客户端间的流动速度不低于 2 秒。	HTTP,FTP 协议	性能: 能够在 1 秒内将信息发送出去 安全性: 传输过程不能保证不可被截取 可靠性: 有可能会丢失	网卡, 网卡驱动, 网络环境
应用服务器管 理	功能: 处理请求, 反馈结果 约束: 用户的请求需要精确的进行处理, 服务器需要支持用户同时进行访问, 其响应速度不小于 1000 位每秒; 服务器应当能够处理完成用户请求, 在 5 秒内。	Tomcat 服务器, struts 开发框架	性能: 可以支持用户同时访问 安全性: 拒绝非法用户访问 可靠性: 不会丢失用户提交请求	Tomcat 版本, cpu 线程数, 操作系统环境
数据库管理	功能: 保存数据, 反馈数据 约束: 数据库服务需要将数据保存在数据库中, 应该保证数据的安全, 应该能够支持数据的快速查询。	Mysql 数据库	性能: 能够快速写入和访问数据 安全性: 只允许有权限用户访问 可靠性: 数据被损坏能够恢复	数据库版本, 硬盘质量

表 9 概念级体系结构说明

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

本系统模块间有业务上关联, 相互组合完成一项业务, 因此本将系统按功能从逻辑上分解, 细化功能结构模块。

SGMS 系统一级和二级模块体系结构如下图。

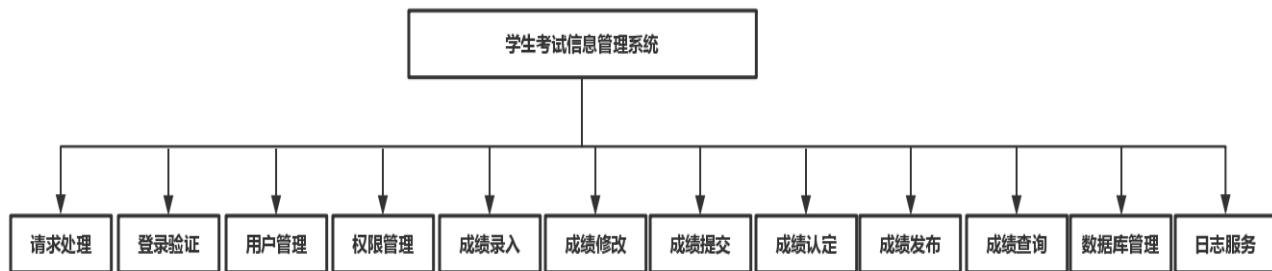


图 11 一级模块体系结构

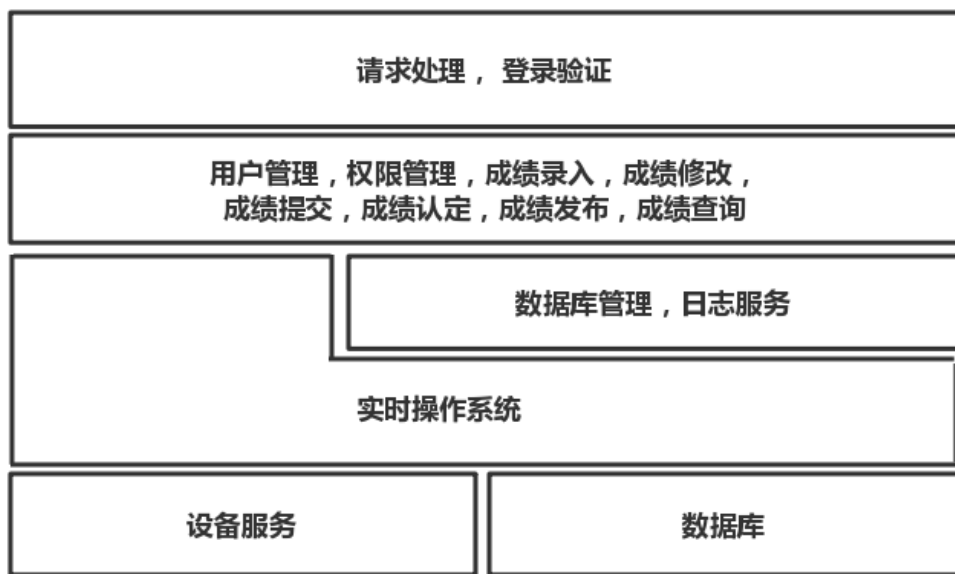


图 12 二级模块体系结构

3.3.2.1 登录认证子系统

登录认证子系统主要为各个用户提供登录服务，用户在浏览器输入账号密码并提交后，登录子系统获取用户账号密码，并传到服务器，服务器查询数据库，查看用户是否合法，如果合法返回登录成功界限，非法返回失败界面，并将结果传回用户浏览器，登录认证子系统功能结构图如下图所示。

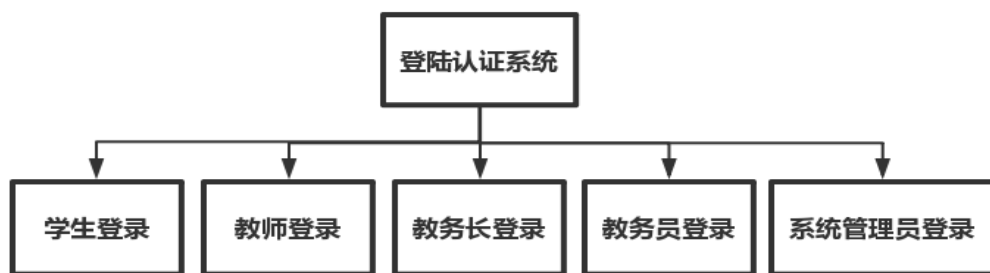


图 13 登陆认证子系统

3.3.2.2 请求处理子系统

请求处理子系统主要根据用户身份的不同，接受不同的请求类型，发送给业务逻辑层，将业务逻辑层的执行结果，反馈给用户，请求处理子系统如下图所示。

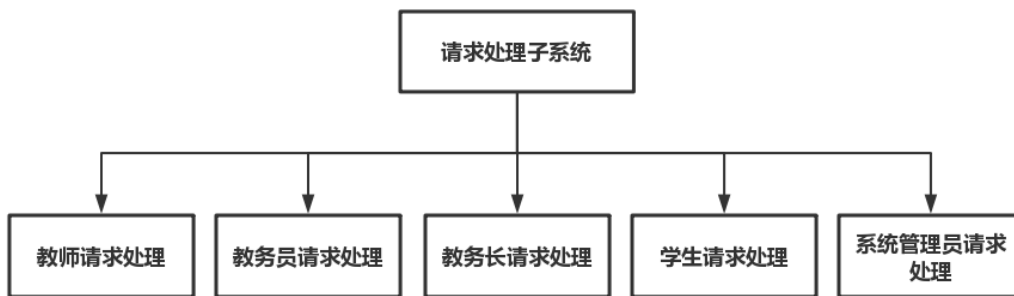


图 14 请求处理子系统

3.3.2.3 用户管理子系统

用户管理子系统主要用于管理用户密码，修改用户密码，查看操作记录等，用户管理子系统如下图所示。

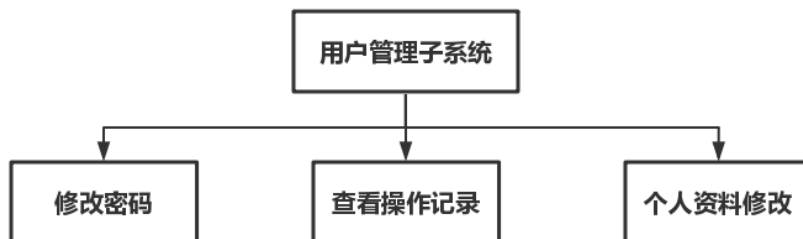


图 15 用户管理子系统

3.3.2.4 权限管理子系统

权限管理子系统主要用于创建用户，删除用户，查看系统用户等，权限管理子系统如下图所示。



图 16 权限管理子系统

3.3.2.5 成绩管理子系统

成绩管理系统主要对成绩信息的管理，针对不同用户对成绩信息的需求进行不同的业务工作，用户在浏览器提交针对成绩信息的操作，服务器接响应，并返回给用户，用户在浏览器接收服务器反馈结果，子系统模块结构如下图所示。



图 17 成绩管理子系统

3.3.2.6 成绩查询子系统

成绩查询子系统主要对成用户的不同身份，进行不同权限范围内的查询，用户在浏览器提交查询成绩，服务器根据身份相应，返回结果到用户的浏览器中，成绩查询子系统模块结构如下图所示。

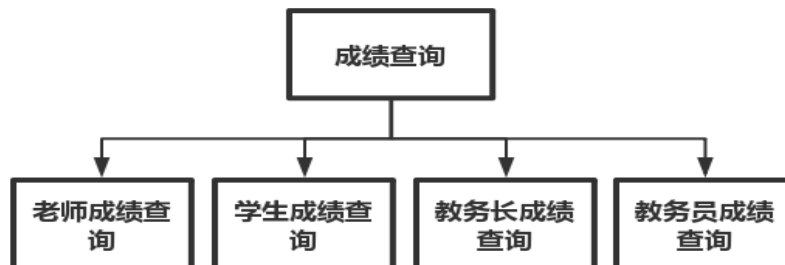


图 18 成绩查询子系统

3.3.2.7 其他模块

3.3.2.7.1 数据库管理模块

数据库管理模块主要对业务层提交的数据请求进行响应，反馈操作结果给

业务层。

3.3.2.7.2 日志管理模块

日志管理模块主要对用户的请求，登录信息进行记录。

3.3.3 运行级体系结构

系统的运行级体系结构视图描述了模块在运行时，被指派到的特定运行映像，运行级体系结构是系统的动态结构，主要用于系统的性能和可调度性分析，以及系统静态与动态配置的管理。系统的消息序列时序图如下。

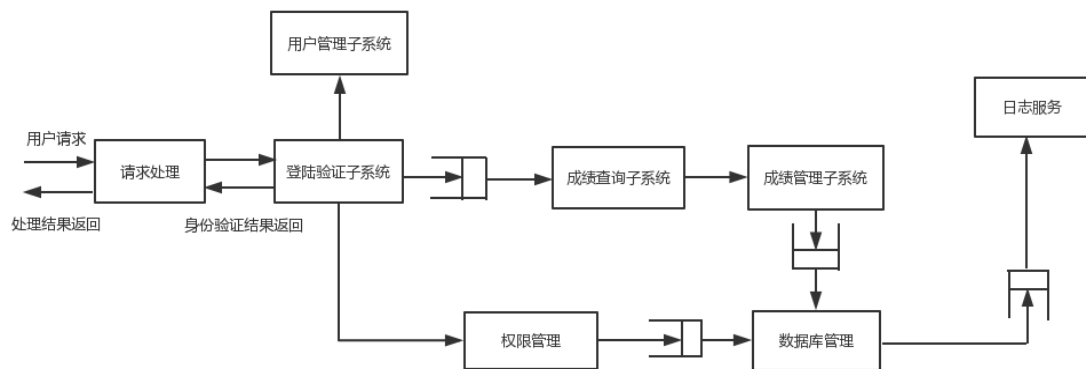


图 19 SGMS 系统运行级体系结构

3.4 实现问题

3.4.1 用户增加

3.4.1.1 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。

3.4.1.2 问题分析

3.4.1.2.1 产生原因

学校扩招，学生人数不断上升，老教师退休，新教师任职，其他学校合作，都会使得学校使用人数上升。

3.4.1.2.2 问题后果

服务器无法及时响应用户请求，可能会强制断开用户连接，影响体验。

3.4.1.3 解决方法

1.运行前预测最大容纳：

系统运行前，调动算法，根据学校总人数等其他条件，计算出饱和人数点，并在系统运行前，调整静态系统配置，让此人数点作为系统最大容纳人数。

2.运行时调整:

系统运行时动态检测人数,调整最大饱和点,对于服务极限以外的请求,予以拒绝。

3.增加硬件配置

购买支持更多线程数的 CPU,拓展成更大的服务内存。

3.4.2 质量属性

3.4.2.1 问题描述

体系结构对质量属性的影响。

3.4.1.3 问题回答

1 从性能上:

B/S 架构将表现层,业务层和逻辑层分开,功能模块界限清晰,解耦合性高,业务流水流畅,可以对请求进行快速响应。

2 从易用性上:

B/S 构架使得用户无需下载客户端,只需浏览器即可访问服务器。

2 从可伸缩性上:

B/S 架构采用的技术和配置,支持用户同时大量并发访问。

3 从密安性上:

B/S 架构中,所有数据均在服务器保存,客户只能根据权限的不同对服务器进行有限的操作。

4 系统的质量分析和评价

4.1 场景分析

4.1.1 用例场景

4.1.1.1 学生

场景 1: 用户期望在输入账号和密码后,系统能够在 3 秒之内给出响应。该场景代表了系统性能要求。

场景 2: 用户期望在成功登录系统后,输入查询成绩的学期,点击查询按钮后,系统能够在 5 秒之内给出响应。该场景代表了系统性能要求。

4.1.1.2 教师

场景 3: 用户期望在输入账号和密码后,系统能够在 3 秒之内给出响应。该场景代表了系统性能要求。

场景 4: 用户期望在成功登录系统后,点击查询按钮后,能够在 5 秒之内查询出

自己所授课程的学生名单以及课程考核指标。该场景代表了系统性能要求。

场景 5: 用户期望在得到学生待录入成绩单后, 在规定合理时间段内, 能够录入相应考核指标的学生分数。该场景代表了系统功能要求。

场景 6: 用户期望在完成学生成绩的录入后, 点击提交按钮, 系统能够在 5 秒之内成功提交给教务长。该场景代表了系统性能要求。

场景 7: 在成绩单提交后发现错误需要修改时, 用户期望能够提交修改申请, 并在点击按钮后 5 秒内得到响应。该场景代表了系统性能要求。

场景 8: 在修改申请得到授权许可后, 用户期望能够对错误的学生分数进行修改, 并且再次提交该分数。该场景代表了系统的功能需求。

4.1.1.3 教务长

场景 9: 用户期望在输入账号和密码后, 系统能够在 3 秒之内给出响应。该场景代表了系统性能要求。

场景 10: 用户期望在成功登录系统后, 点击查询按钮后, 能够在 5 秒之内查询出自己所要认定的成绩的名单及对应的学生名单。该场景代表了系统性能要求。

场景 11: 用户期望在得到学生待认定成绩单后, 在规定合理时间段内, 能够完成认定成绩, 并通知教务员发布认定的成绩。该场景代表了系统功能要求。

场景 12: 用户期望在完成学生成绩认定后, 点击确认按钮, 系统能够在 5 秒之内成功完成数据认定并同步给教务员。该场景代表了系统性能要求。

场景 13: 在成绩单提交后发现错误需要修改时, 用户期望能够选择老师, 进行授权, 通知其修改, 并在点击按钮后 5 秒内得到响应。该场景代表了系统性能要求。

场景 14: 在点击授权修改按钮后, 用户期望在点击按钮后 5 秒内得到响应。该场景代表了系统性能要求。

4.1.1.4 教务员

场景 15: 用户期望在输入账号和密码后, 系统能够在 3 秒之内给出响应。该场景代表了系统性能要求。

场景 16: 用户期望在成功登录系统后, 点击查询按钮后, 能够在 5 秒之内查询出自己所要发布成绩的名单。该场景代表了系统性能要求。

场景 17: 用户期望在得到学生待发布成绩单后, 在规定合理时间段内, 能够完成发布成绩, 该场景代表了系统功能要求。

场景 18: 用户期望在完成学生成绩发布后, 点击确认按钮, 系统能够在 5 秒之内成功完成数据更新。该场景代表了系统性能要求。

4.1.1.5 系统管理员

场景 19: 用户期望在输入账号和密码后, 系统能够在 3 秒之内给出响应。该场景代表了系统性能要求。

场景 20: 用户期望在成功登录系统后, 点击查询按钮后, 能够在 5 秒之内查询出系统用户名单。该场景代表了系统性能要求。

场景 21: 用户期望在得到系统用户名单后, 在规定合理时间段内, 能够完成删除

用户，该场景代表了系统功能要求。

场景 22: 用户期望在点击删除按钮后，能够在 5 秒之内，完成删除用户，该场景代表了系统性能要求。

场景 23: 用户期望在完成新用户信息填写后，在规定合理时间段内，能够完成用户建立。该场景代表了系统功能要求。

场景 24: 用户期望在点击提交按钮后，能够在 5 秒之内，完成用户建立。该场景代表了系统性能要求。

4.1.2 增长性场景

场景 1: 可以增加新的 web 服务器，将 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。

场景 2: 通过对数据库的表建立索引，把数据查询时间降低到 1 秒之内。

场景 3: 可以增加支持手机端 Android, ios 的访问，希望系统仅需增加一人周的工作量就能完成。

场景 4: 可以修改服务器的处理方式，将所支持的并发访问人数增长到 10000 人，希望系统仅需增加一人周的工作量就能完成。

4.1.3 探索性场景

场景 1: 用户可以使用 Email 或者短信接收系统消息通知。

- 学生所修读课程的成绩分数发布时收到提醒；
- 考核指标的相应合法录入时间段开放时，会通知教师进行成绩录入；
- 教师提交成绩单或者修改申请后，教务长会收到通知；
- 教务长认定成绩后，教务员会收到消息通知进行成绩发布。

场景 2: 正常情况下，当一半服务器宕机时，不影响整个系统的可使用性。

场景 3: 在成绩发布后一周是高峰时期，为保证至少对 1000 名学生的请求能够在 3 秒内相应，开启备用服务器进行并发工作。

4.2 原型分析

场景分析法能够对系统体系结构仅需有效的评价，但这些场景很难直接回答体系结构的问题，本系统采用原型分析法，对本系统体系结构进行测试，如下表。

功能	测试过程	测试结果
处理能力	调用 5000 次请求处理 API，计算处理完成时间	一共花费 3.7 秒，能够在 5 秒内完成 5000 次请求处理
数据库访问能力	调用 5000 次数据处理 API，计算处理完成时间	一共花费 4 秒，能在 5 秒内完成 5000 次数据读写

表 10 SGMS 原型分析

4.3 效用树分析

为了更加准确地描述 SGMS 系统的质量要求，对质量属性设置优先级别，以便于对系统体系结构的评价。构建效用树如下图。

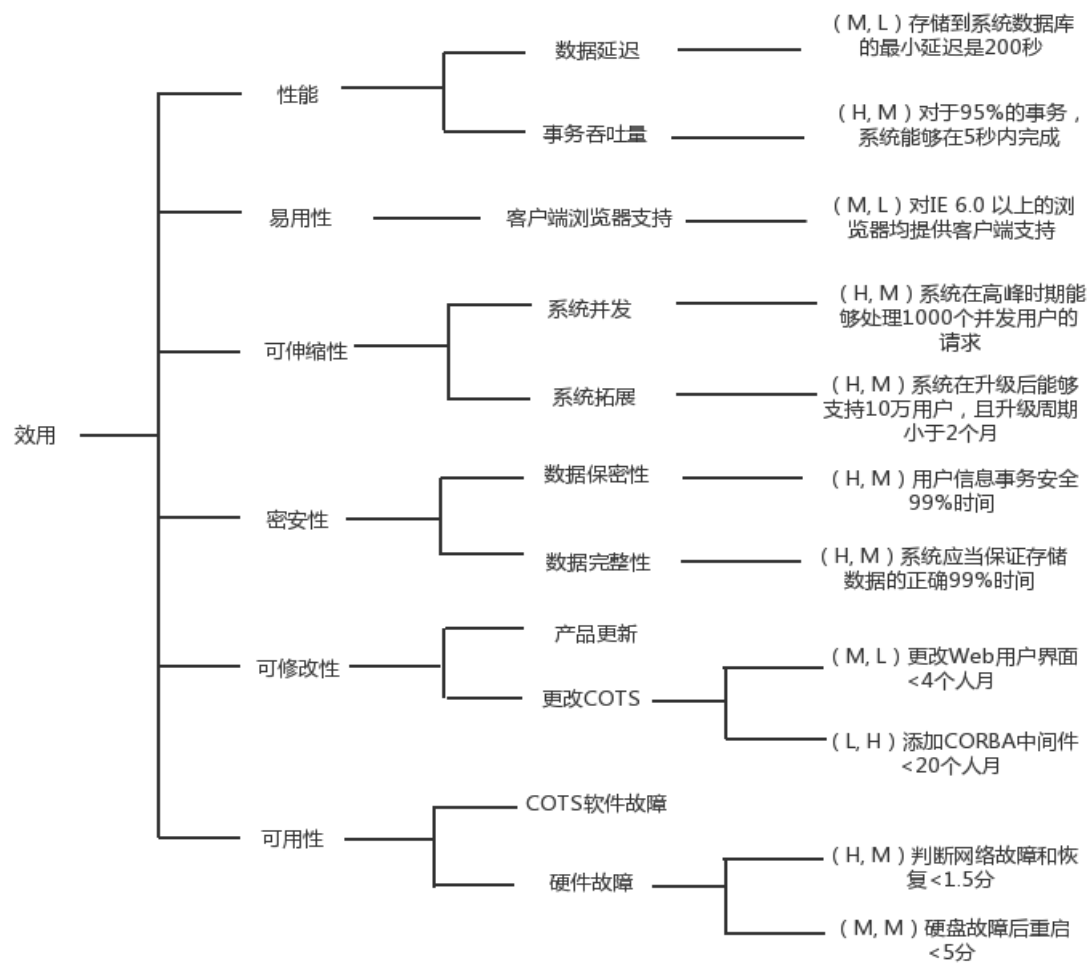


图 20 SGMS 系统的效用树

其中，图中 H、M、L 分别代表高、中、低的优先次序。优先权从两个维度看：每个节点的重要程度，以及实现该节点的要求所具有的风险。

4.4 风险

SGMS 系统在开发过程中可能遇到的风险以及控制办法如下表所示。

风险	控制办法
----	------

人员薄弱	招募顶级人才，培训，团队建设，项目开始前招聘或预定关键成员
不切实际的进度	通过协商制定一个更为合理的进度计划，任务并行化，关注关键路径

表 11 风险及控制策略