**技术路线及实现方案（S2C）**

目录

[1.系统整体技术解决方案 1](#_Toc10454)

[2.前端展示技术 2](#_Toc25110)

[2.1前端UI框架 3](#_Toc346)

[2.2前端JavaScript框架 3](#_Toc13106)

[2.3页面布局 3](#_Toc23731)

[2.4前后端数据交互 3](#_Toc6473)

[3.后端技术方案 3](#_Toc5548)

[3.1 MVC 3](#_Toc12734)

[3.2 J2EE 4](#_Toc29376)

# 

# 1.系统整体技术解决方案

此部分主要介绍系统在进入实施过程中的技术方案。本系统业务逻辑较为复杂，包含有教师管理、学生管理、课程管理和系统管理多个核心业务模块，因此要将开发的主要精力投入到业务逻辑开发中来，避免重复的开发工作，提升开发效率，除了根据开发人员的开发习惯选择主流的开发框架作为基础框架之外，也要注意针对本系统进行公用模块格模块的开发积累，形成一套针对本系统的、可维护性和扩展性较强的代码系统。

本系统采用经典的B/S框架开发实现方式，用户可通过浏览网页的形成使用本系统，根据系统的实现方式，可将具体的开发技术方案分为前端技术方案和后端技术两个模块。前端技术方案负责实现web端的页面展示和用户交互功能。在实际开发过程中，前端技术方案主要包括前端UI框架、JavaScript框架、页面布局、前后端数据交互。后端技术方案主要基于J2EE平台实现，根据Spring MVC思想，将后端技术方案分为表示层、业务层逻辑、持久层、控制层四个技术层面，每一层都有面向上一层的接口，实现了不同层之间的“高內聚-低耦合”。

面向对象的架构设计：通过抽象、封装、模块化和层次化，从架构设计上确保软件实现的结构化、灵活性和可复用。

关注分离的架构设计：通过分层、面向方面和容器等设计模式，将需求和架构分析结果中交织、混沌的软件元素关系梳理清楚，使性质不同的关注面被分割而独立，进而实现分而治之。

通过面向对象和关注分离的设计思想和相关实现技术，可以尽可能促使技术架构及其支撑的企业级行业应用的具有良好、清晰的层次结构，能够封装变化，具有足够的灵活性和可复用能力。

应用开发框架的技术主要包括：

前台展现技术：JSP、jQuery、JSTL

后台业务逻辑技术：Spring、HttpClient、Dubbo/osgi、Mybatis、Log4j、shiro

# 2.前端展示技术

随着互联网时代的到来，人们对于web应用的交互需求和体验要求越来越高，良好的页面展示和交互设计往往能大大提升用户体验，吸引更多用户。因为根据测评系统的应用需求和用户使用特点，我们选择了以下技术方案：

### 2.1前端UI框架

随着CSS和HTML的流行，web页面不仅需要合理的流程设计，更需要酷炫的页面风格和良好的用户体验，前端UI框架为用户提供了一整套风格一致，浏览器兼容性和的封装组件，在美化页面的同时，也大大提高了开发效率。本系统选择了Bootstrap作为前端UI框架。

### 2.2前端JavaScript框架

实现网页和用户之间实时、动态的、交互的关系。将其中一些借口暴露给开发人员，降低代码的重复率，提升开发效率。本系统选择JQuery作为前端JavaScript框架。

### 2.3页面布局

目前页面布局采用标准的DIV+CSS来实现。因为本系统主要针对高校师生，且应用功能强于展示功能，因此采用简单明朗、且大众普遍熟悉和接收的T字型网页布局。

### 2.4前后端数据交互

在传输方式上选择AJAX,它允许网页在无刷新的情况下异步请求服务端数据，大大提升用户的使用体验。数据传输格式选择JSON。

# 3.后端技术方案

### 3.1 MVC

本系统分为三个部分：

1. 模型：用来封装业务数据、处理业务逻辑，提供数据访问接口。
2. 试图：负责用户与web界面的交互和页面的展示，它一方面接收模型层传输来的数据并将在页面中展示这些数据，一方面将用户的请求和输入的数据转发给控制层和模型层。
3. 控制层：它接收视图层传来的用户请求，按照请求内容将请求数据传输给对应的数据模型或业务模型，在模型完成逻辑处理后，将处理后的结果返回给相应的视图。

### 3.2 J2EE

根据J2EE平台的MVC思想在本系统的应用，本系统采用项目架构分层的思想，将系统开发架构分为七个层面：表示层、控制层、服务层、业务逻辑层、原子服务API层、数据访问层、数据储存层。表示层采用JQuery+Bootstrap+FreeMarker，负责展示页面、与用户交互；控制层采用Struts2，负责请求调度和数据转发；业务逻辑层采用Spring框架，负责业务模块的处理和整个项目的规划调度；原子服务层：抽象的公共原子服务，可在多个系统间共享，采用Dubbo/OSGI的方式对服务进行管理和使用；数据访问层：负责数据库数据的读写。采用Mybatis技术实现数据访问层功能。层与层之间的采用持久层对象和实体对象来进行数据传输。

1. 表示层

采用FFreeMarker，在HTML语言的基础上创建出独立的数据展示形式，配合JQuery、CSS、Bootstrap共同实现了系统的前端知识功能。

1. 控制层

本层的实现采用了Struts2框架，通过对拦截器等处理请求组件的实现，清晰的实现MVC思想。XWork2为命令模式框架，能控制反转容器，表达式语言，数据类型转换和验证配置。

1. **服务层**

本系统选用MyBatis来实现系统的持久层构建，它支持SQL语句查询、高级映射和数据存储。MyBatis具有三层功能结构：API接口层向上层代码提供处理接口；数据处理层负责具体的SQL查找、解析、执行和执行结果映射处理；基础支撑层，将数据库连接、加载机制、事务管理、缓存策略等抽取成基础组件。

1. **业务逻辑层**

业务逻辑层是本系统的核心功能，它负责处理业务需求，实现系统的价值型操作。在处理具体业务逻辑时会涉及到功能代码模块的粘连和复用，因此对具体业务代码进行抽离、封装和管理对于项目今后的维护和扩展显得尤为重要。本系统采用了Spring框架来实现此功能。

1. **数据储存层**

数据库拆分，把生产数据库和查询数据库分离；通过读写分离技术提高数 据库扩展能力；采用高效的网络文件共享策略，采用图片服务器来实现页面的图片存储。