华中科技大学网上选课系统

概要设计说明书

**V1.0**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 人员 | 时间 | 备注 |
| 编写 | 贾智勇、徐可辰 | 2023.4.22 |  |
| 审核 | 贾智勇、徐可辰 | 2023.4.22 |  |
|  |  |  |  |
|  |  |  |  |

**目录**

[1 概述 3](#_Toc22050)

[2 选课系统架构的作用和表示方式 3](#_Toc1251)

[2.1 软件架构的作用 3](#_Toc26115)

[2.2 软件架构的表示方式 3](#_Toc13999)

[3 选课系统架构的目标和约束 4](#_Toc29686)

[3.1 目标 4](#_Toc19437)

[3.2 约束 4](#_Toc31632)

[4 架构设计方案 5](#_Toc20796)

[4.1 架构分析 5](#_Toc16618)

[4.2 设计思想 5](#_Toc1148)

[4.3 架构体系 5](#_Toc5092)

[4.4 子系统和模块划分 6](#_Toc24964)

[4.4.1 视图层 7](#_Toc9864)

[4.4.2 业务逻辑层 8](#_Toc10628)

[4.4.3 数据访问层 8](#_Toc22380)

[5 用例视图 9](#_Toc25166)

[5.1 学生管理用例 9](#_Toc6218)

[5.2 课程管理用例 9](#_Toc29986)

[5.3 成绩管理用例 9](#_Toc6691)

[5.4 选课用例 9](#_Toc4279)

[6 逻辑视图 10](#_Toc29604)

[6.1 子系统与包 10](#_Toc20620)

[6.2 实体类设计 10](#_Toc23476)

[学生实体类 10](#_Toc31692)

[教师实体类 10](#_Toc8517)

[课程实体类 11](#_Toc73)

[选课申请实体类 11](#_Toc4835)

[成绩实体类 11](#_Toc12888)

[7 进程视图 11](#_Toc23451)

[8 部署视图 12](#_Toc24733)

[9 实施视图 12](#_Toc18473)

# **1 概述**

本文旨在介绍华中科技大学网上选课系统的架构设计方案，以满足华中科技大学管理编排课程、信息管理、网上选课等需求。在本文中，我们会详细介绍软件架构的作用和表示方式、软件架构的目标和约束、架构设计方案、用例视图、逻辑视图、进程视图、部署视图、实施视图等内容，以便读者全面了解该方案的设计和实现。

本文的目标读者包括但不限于本系统的开发工程师、信息管理人员、教务管理人员和其他相关技术人员。

# **2 选课系统架构的作用和表示方式**

**2.1 软件架构的作用**

软件架构指的是软件系统的组织结构，它是由一组相互关联的组件和连接这些组件的接口所构成。软件架构决定了软件系统的性能、可扩展性、可维护性、安全性等方面。因此，在软件开发过程中，架构设计尤为重要。

软件架构可以帮助开发者更好地组织代码和功能模块，使得系统具有更好的灵活性和可维护性。其次，软件架构可以有效地划分开发人员的工作范围，提高开发效率。另外，软件架构还可以使得软件系统更易于测试、部署和维护，从而降低了开发成本。

**2.2 软件架构的表示方式**

软件架构的表示方式通常通过不同的视图来呈现。在网上选课系统的架构设计中，我们采用了四种常用的视图：用例视图、逻辑视图、进程视图和部署视图。这些视图分别从不同的角度描述了系统的结构和行为。

1. 用例视图

用例视图描述系统的功能和行为，它是从用户的角度出发，对系统进行的逻辑分解。通过用例视图，我们可以清晰地了解系统的核心用例，以及这些用例之间的关系。

1. 逻辑视图

逻辑视图描述系统的静态结构，它包括了系统中各个模块的属性、接口和相互关系。逻辑视图可以帮助开发人员更好地理解系统的组成部分和其内部结构。

1. 进程视图

进程视图描述系统的并发和异步性质，它主要关注系统中各个进程和线程的交互关系。通过进程视图，我们可以更好地把握系统并发执行的特点和优化策略。

1. 部署视图

部署视图描述系统在不同硬件环境下的部署情况，它能够帮助开发人员更好地了解系统的部署需求和资源消耗，方便进行系统部署和维护。

# 3 选课系统架构的目标和约束

在进行网上选课系统的架构设计时，我们需要考虑以下目标和约束。这些目标和约束是在需求分析和技术选型的基础上得出的，旨在确保系统具有高可用性、可扩展性、可维护性和安全性。

**3.1 目标**

1. 高可用性：系统需要具备高可用性，在面对高并发访问和突发流量时，能够保证系统的稳定运行，能够及时响应用户的请求。
2. 可扩展性：系统需要具备良好的可扩展性，随着业务规模的增长，系统能够快速地进行水平或垂直扩展，以满足业务需求。
3. 易维护性：系统需要具备良好的易维护性，代码的复杂度要尽可能低，结构要清晰明了，方便维护人员进行代码的修改、测试和调试。
4. 安全性：系统需要具备良好的安全性，如防止SQL注入、 XSS攻击等安全问题。

**3.2 约束**

1. 技术选型：系统采用的技术选型需要符合企业技术栈和技术规划，具有可靠性、稳定性、安全性等特点。
2. 可扩展性：系统的架构需要具备良好的可扩展性，以便于后期的业务扩展。
3. 性能优化：系统需要通过各种手段对性能进行优化，如缓存、负载均衡、异步处理等。
4. 易用性：系统需要考虑到用户体验，操作简单易懂、界面美观大方。
5. 成本控制：系统的设计需要考虑成本的因素，需要设计出高效低成本的架构。

# **4 架构设计方案**

**4.1 架构分析**

网上选课系统是一个基于互联网的课程管理平台，在满足用户选课需求的同时，需要保证系统的安全性、可靠性和稳定性。考虑到系统的规模和复杂性，我们采用了分层架构的设计思想，并在每一层中采用了不同的技术和框架进行实现。

**4.2 设计思想**

我们的设计思想主要是基于MVC（Model-View-Controller）模式，将系统的功能划分为三个部分：视图层、业务逻辑层和数据访问层。其中，视图层负责与用户的交互，业务逻辑层负责处理具体的业务流程，数据访问层负责管理数据的存储和持久化。

**4.3 架构体系**

在系统架构方面，选课系统项目采用Web系统常用的MVC的模式开发，使用SpringMVC作为项目的主要架构。MVC模式是软件工程中的一种软件架构模式，把软件系统分为三个基本部分：模型（Model）、视图（View）和控制器（Controller），即MVC。下图给出了一个通用MVC的三层架构图。模型和视图有着业务层面的业务数据紧密耦合关系，控制器的核心工作就是业务逻辑处理。这种架构的最大特点就是用户界面容易随着需求的变更而改变，可以包容需求上的变化导致修改用户界面的程序而不变影响软件的核心功能代码。不难想到，鉴于MVC架构的特点，可以较为完善的解决我们的需求。

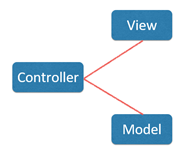
[](https://img2020.cnblogs.com/blog/2164317/202012/2164317-20201228105930800-1827133184.png)

图4-1

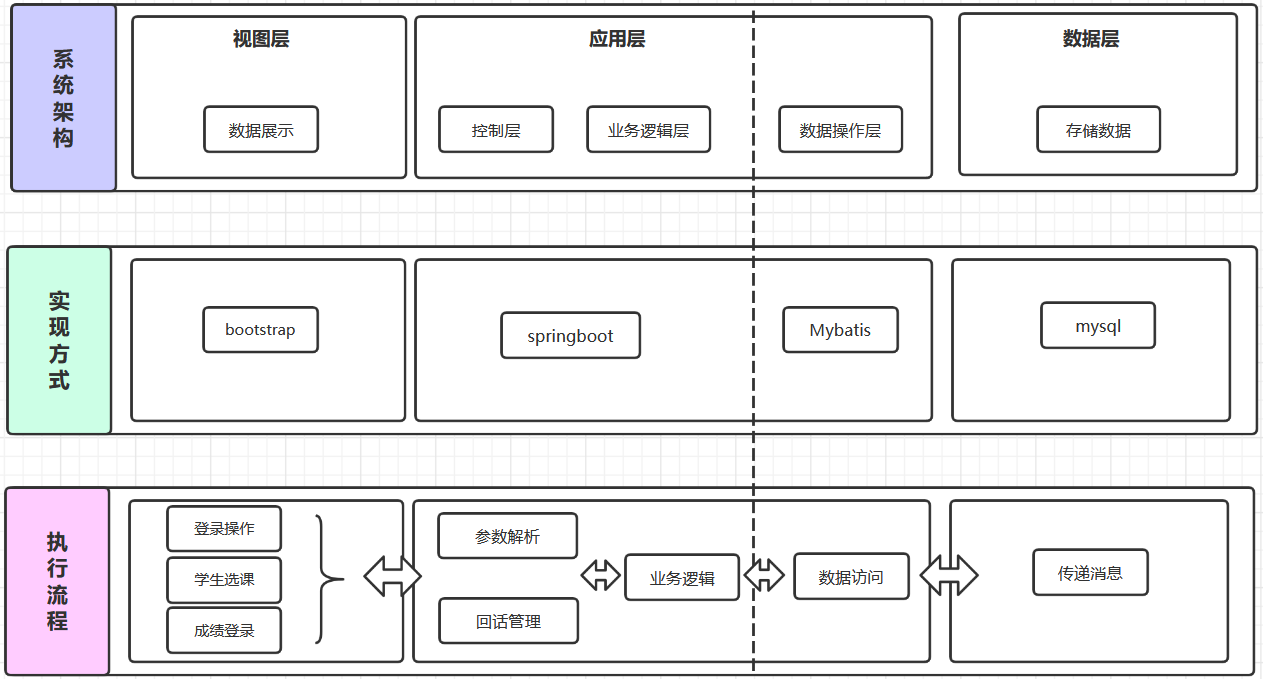
结合MVC和选课系统的业务特点，我们将系统的数据流通过的过程分为三层：视图层、应用层和数据层。视图层主要负责数据的展现；应用层主要负责业务逻辑处理；数据库层负责数据的读写以及数据的持久化。系统架构图、实现方案如下图所示。由SpringMVC+MyBatis为主要框架，前端采用bootsrap框架，后端使用mysql8.0配置主从复制实现读写分离。

图4-2

**4.4 子系统和模块划分**

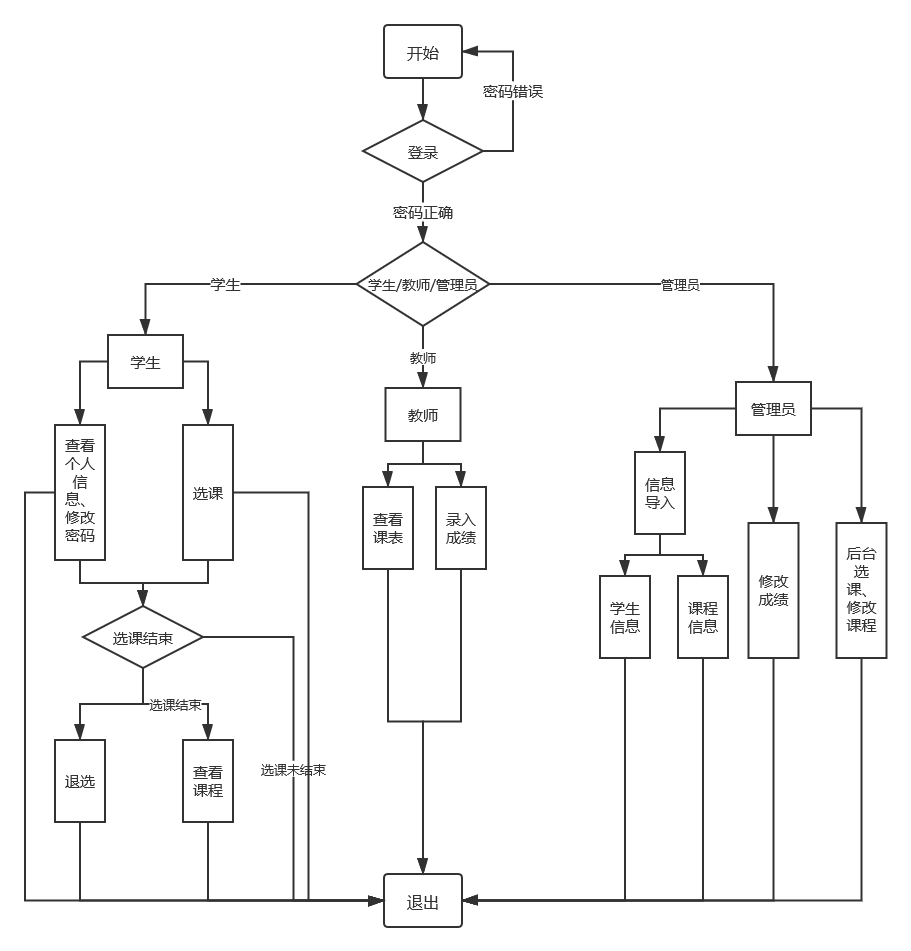


图4-3

**4.4.1 视图层**

#### 模块描述

视图层主要包括登录、注册、选课、退课、查看课表等操作的页面展示和交互设计。在实现过程中，我们使用HTML、CSS、JavaScript、jQuery等前端技术，以及Bootstrap等UI框架进行页面的布局和美化。

#### 模块接口设计

* 用户登录：用户输入账号和密码，通过Ajax技术向后台发送请求，后台对用户身份进行验证，如果验证通过，则返回用户信息并跳转至主界面；否则，返回错误提示信息。
* 用户注册：用户输入账号、密码、姓名、学号等信息，通过Ajax技术向后台发送请求，后台进行信息校验，如果信息格式正确，则将用户信息保存至数据库，并返回成功信息；否则，返回错误提示信息。
* 选课和退课：用户点击课程列表中的选课或退课按钮，通过Ajax技术向后台发送请求，后台对用户身份进行验证，如果验证通过，则执行相应的操作，并返回成功信息；否则，返回错误提示信息。

### 4.4.2 业务逻辑层

#### 模块描述

业务逻辑层主要包括学生信息管理、课程信息管理、选课退课管理、成绩查询等操作的具体实现。在实现过程中，我们使用Java语言进行编程，使用Spring、SpringMVC、MyBatis等开源框架进行搭建。

#### 模块接口设计

* 学生信息管理：提供学生信息的增删改查功能，包括添加学生、删除学生、修改学生信息、查询学生信息等操作。
* 课程信息管理：提供课程信息的增删改查功能，包括添加课程、删除课程、修改课程信息、查询课程信息等操作。
* 选课和退课管理：提供选课和退课相关的服务，包括选课、退课、查看选课情况等操作。
* 成绩查询：提供成绩查询服务，包括查看个人成绩、查看整个课程的成绩排名等操作。

### 4.4.3 数据访问层

#### 模块描述

数据访问层主要负责管理数据的存储和持久化，采用MySQL作为数据库的存储系统。在实现过程中，我们使用DAO（Data Access Object）和POJO（Plain Old Java Object）设计模式进行编程，以实现对数据访问的封装和隔离。

#### 模块接口设计

* 学生信息管理DAO：提供对学生信息的增删改查等基本操作，包括添加学生、删除学生、修改学生信息、查询学生信息等操作。
* 课程信息管理DAO：提供对课程信息的增删改查等基本操作，包括添加课程、删除课程、修改课程信息、查询课程信息等操作。
* 选课和退课管理DAO：提供对选课和退课操作的支持，包括选课、退课、查看选课情况等操作。
* 成绩查询DAO：提供成绩查询服务，包括查看个人成绩、查看整个课程的成绩排名等操作。

# **5 用例视图**

根据网上选课系统的需求分析，在用例视图中，主要包括以下几个用例：

**5.1 学生管理用例**

学生管理用例主要包括添加学生、修改学生信息、删除学生、查看学生信息等基本功能。当教务管理员需要对学生信息进行管理时，就会触发这些用例。

**5.2 课程管理用例**

课程管理用例主要包括添加课程、修改课程信息、删除课程、查看课程信息等基本功能。当教务管理员需要对课程信息进行管理时，就会触发这些用例。

**5.3 成绩管理用例**

成绩管理用例主要包括录入成绩、查询成绩等功能。当教师需要管理课程成绩或者学生需要查询自己的成绩时，就会触发这些用例。

**5.4 选课用例**

**概要**

该用例描述了学生在网上选课系统中进行选课和退课的操作流程，包括选课申请、审核、审核通过与否反馈等过程。

**触发条件**

当学生需要进行选课或者退课操作时，就会触发该用例。

**前置条件**

* 学生需要提前注册并登录到网上选课系统中。
* 学生需要选择待选课程并填写选课申请单。

**基本流程**

1. 学生在网上选课系统中进行选课或退课操作。
2. 系统接收到学生的选课或退课请求后，将其提交给教务管理员进行审核。
3. 教务管理员审核学生的选课或退课申请，如果通过审核，则将其反馈给学生。如果未通过审核，则需要说明原因，并拒绝该申请。
4. 学生根据审核结果进行相应的操作，完成选课或退课流程。

**扩展流程**

1. 如果学生填写的选课申请单不符合相关规定，则需要让学生重新填写，否则无法进行下一步审核流程。

2. 如果教务管理员出现问题无法审核申请，则该申请需要重新提交到待审核状态。

# **6 逻辑视图**

**6.1 子系统与包**

在网上选课系统的逻辑视图中，可以将其分解为以下几个子系统：

* 用户管理子系统：主要职责是对系统用户进行管理，包括学生、教师和管理员等。
* 课程管理子系统：主要职责是对系统中的课程信息进行管理，包括添加、修改、删除和查询等操作。
* 选课管理子系统：主要职责是对选课和退课操作进行管理，包括选课审核、时间安排和成绩管理等。
* 统计报表子系统：主要职责是对系统中各种数据进行统计和汇总，生成各种报表和分析数据，为教学管理提供决策依据。

**6.2 实体类设计**

### 学生实体类

* 职责：保存学生的个人信息和选课信息，包括姓名、学号、性别、专业、已选课程等。
* 关系：与课程实体类之间存在多对多的关联关系。
* 操作：提供获取学生信息、选课和退课等功能接口。
* 属性：姓名、学号、性别、专业、已选课程等。

### 教师实体类

* 职责：保存教师的个人信息和教授课程信息，包括姓名、工号、性别、授课课程等。
* 关系：与课程实体类之间存在多对多的关联关系。
* 操作：提供获取教师信息和教授课程信息等功能接口。
* 属性：姓名、工号、性别、授课课程等。

### 课程实体类

* 职责：保存课程的基本信息，包括课程编号、名称、学分、授课教师等。
* 关系：与学生和教师实体类之间存在多对多的关联关系。
* 操作：提供添加、修改、查询和删除课程等操作接口。
* 属性：课程编号、名称、学分、授课教师等。

### 选课申请实体类

* 职责：保存学生的选课申请信息，包括选课时间、课程编号、学生编号等。
* 关系：与学生和课程实体类之间存在多对多的关联关系。
* 操作：提供提交选课申请和查询选课申请状态等操作接口。
* 属性：选课时间、课程编号、学生编号等。

### 成绩实体类

* 职责：保存学生成绩信息，包括课程编号、学生编号以及成绩等。
* 关系：与学生和课程实体类之间存在多对多的关联关系。
* 操作：提供录入成绩和查询成绩等操作接口。
* 属性：课程编号、学生编号、成绩等。

# **7 进程视图**

在网上选课系统的软件体系结构中，可以将其分解为以下几个进程：

1. 用户界面进程：负责处理用户的输入和输出，包括登录、注册、选课、退课等操作。该进程通常由多个线程组成，每个线程负责处理一个或多个用户请求。
2. 业务逻辑进程：负责处理用户操作所涉及的业务逻辑，包括权限验证、数据处理、状态转移等。该进程通常由多个线程组成，每个线程负责处理一个或多个业务操作。
3. 数据库管理进程：负责管理系统所使用的数据库，包括数据存储、索引管理、备份恢复等。该进程通常由多个线程组成，每个线程负责处理一个或多个数据库操作。

这三个进程之间通过进程间通信（IPC）机制进行通信。用户界面和业务逻辑进程之间通过消息队列进行通信，用户界面进程将用户请求发送到消息队列中，业务逻辑进程从消息队列中获取请求并进行业务处理，处理完成后将结果返回到消息队列中，用户界面进程再从消息队列中获取结果并反馈给用户。

业务逻辑进程和数据库管理进程之间通过共享内存进行通信，业务逻辑进程在需要访问数据库时，将请求发送到共享内存中，数据库管理进程从共享内存中获取请求并进行相应操作，处理完成后将结果写回共享内存中，业务逻辑进程再从共享内存中获取结果。

在实现这些进程时，需要考虑到其在系统中的具体应用场景和资源分配等因素，从而保证系统的高效性和稳定性。同时，还需要考虑到系统的可扩展性和可维护性，使得系统能够适应不断变化的需求，并且能够方便地进行更新和修复。

# **8 部署视图**

为了满足高峰使用期间需求，网上选课系统采用分布式架构进行部署，在多台服务器上运行。这些服务器须支持Java和MySQL数据库。在这些服务器上，需要安装Tomcat服务器、MySQL数据库和Apache HTTP Server等软件来支持系统的正常运行。

针对不同的业务需求，将网上选课系统划分为三个模块：前台展示模块、后台管理模块和数据接口模块。这些模块分别部署在不同的服务器上。

具体来讲，前台展示模块部署在Web服务器中。它包括用户登录、选课、查看成绩等功能。通过Web浏览器，用户可以访问Web服务器并使用这些功能。该模块与MySQL数据库交互以获取并显示数据。

后台管理模块是用于管理员管理网上选课系统的模块。它包括管理员登录、课程管理、学生管理等功能。它部署在专门的应用服务器上，并与MySQL数据库进行交互以获取和更新数据。

数据接口模块是网上选课系统的核心模块。它将前台展示模块和后台管理模块的请求传递给数据库，并将数据库的响应返回给相应的模块。该模块部署在一个单独的应用服务器上，并使用Java运行时环境（JRE）执行。

这些模块通过TCP/IP协议在本地网络环境中进行通信。它们通过RESTful API进行通信，使用JSON作为数据交换格式。

# **9 实施视图**

该软件系统的整体结构采用基于MVC设计模式的三层架构。它由表现层、业务逻辑层和数据访问层组成。

表现层是网上选课系统的用户界面。它使用HTML5、CSS3和JavaScript等前端技术，包括jQuery库和Bootstrap框架等第三方控件，自定义控件，以及一些公共样式和布局。表现层接收来自用户的请求，根据请求的类型和参数调用相应的服务，并将响应返回给用户。表现层可以在PC端和移动端访问。

业务逻辑层是网上选课系统的核心层。它包含各种业务逻辑和处理操作，以及所有的控制器和服务。它使用Java语言和Spring框架进行开发，并实现了CQRS设计模式和Event Sourcing模式。它采用ORM框架MyBatis与MySQL数据库进行交互，并通过Redis缓存提高系统性能。业务逻辑层对请求进行检查和验证，并将数据传递给数据访问层进行处理。业务逻辑层与表现层和数据访问层之间采用HTTP协议进行通信。

数据访问层是网上选课系统的数据存储层。它使用MySQL数据库进行数据存储，并与业务逻辑层进行交互。数据访问层实现了基本的CRUD操作和一些高级查询。它也使用了一些第三方控件和自定义控件，例如Druid数据库连接池和Mybatis Generator代码生成器等。

下图给出了选课系统的实现视图。回顾本项目的系统架构为MVC架构，下面就此架构说明项目的实现视图。

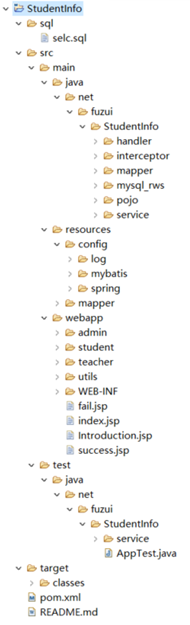
[](https://img2020.cnblogs.com/blog/2164317/202012/2164317-20201228110334898-905601880.png)

图9-1

模型部分由selc.sql文件和pojo文件下定义的类决定（如course、student、teacher类等），并在mapper文件夹中定义了这些类的访问接口。

视图部分由webapp文件夹决定，以webapp文件下的student为例，提供了各类交互方法，如退选课程、选择课程、修改个人信息等操作：

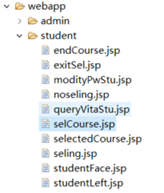
[](https://img2020.cnblogs.com/blog/2164317/202012/2164317-20201228110358724-1876392785.png)

图9-2

控制器部分，，其中实现了将视图层和模型层联系起来的众多方法，如在登录代码中，集成了教师、学生和管理员的登录和退出登录的操作，这些操作完成的时候，会将模型数据和页面显示结合起来，即实现了MVC架构的核心功能。