

山东大学

毕业论文(设计)

论文（设计）题目：

在线辅助教学系统的研究与开发

姓 名 朱泽兵
学 号 201100300196
学 院 山东大学软件学院
专 业 软件工程
年 级 2011 级
指导教师 黎峰

2015 年 6 月 4 日

目录

摘 要.....	1
ABSTRACT.....	2
第 1 章 绪论.....	4
1.1 在线辅助教学系统开发背景.....	4
1.2 国内外研究现状.....	5
1.3 解决的主要问题.....	6
1.4 本文的主要工作.....	6
1.5 论文的组织结构.....	7
第 2 章 在线辅助教学系统需求分析.....	8
2.1 系统概述.....	8
2.1.1 项目背景.....	8
2.1.2 项目说明.....	8
2.1.3 系统整体系统概述.....	9
2.2 目标和解决的问题.....	10
2.3 在线辅助教学系统需求问题描述.....	11
2.3.1 功能性需求.....	11
2.3.2 需求建模.....	13
2.3.3 非功能性需求.....	14
第 3 章 在线辅助教学系统架构设计.....	16
3.1 系统设计目标和原则.....	16
3.2 系统技术架构设计.....	16
3.3 在线辅助教学系统功能架构.....	18
3.3.1 系统功能组成.....	18
第 4 章 在线辅助教学系统详细设计.....	20
4.1 系统建模.....	20
4.2 数据库设计.....	20
4.2.1 数据库设计原则.....	20
4.2.2 数据库标识符和状态.....	21
4.3 数据库逻辑设计.....	21
4.4 详细功能设计.....	22
第 5 章 在线辅助教学系统实现与测试.....	31
5.1 系统总体实现.....	31
第 6 章 结论.....	35
6.1 项目总结.....	35
6.2 系统存在的问题及展望.....	35
致谢.....	36
参考文献.....	37
附录 1 英文原文.....	39
附录 2 译文.....	43

在线辅助教学系统的研究与开发

摘 要

随着计算机技术和网络技术的不断发展，人们对教育水平、效率的需求不断增加以及我国教学资源的利用相对不统一。同时也为了适应由应试教育向素质教育改革的现状，基于网络的教学系统的研究与推广已经变得越来越普遍，也显得尤为重要。

目前，有相当一部分的辅助教学系统还是采用 C/S 结构的，它增加了客户端的管理和维护，也不利于移动办公。在此，设计一款基于 SSH2 框架的在线辅助教学系统。系统采用 B/S 模式，无需安装客户端。运行效率更高，使用更加方便。在线辅助教学系统的研究推广，有利于推动教育教学的改革，推动社会全面进步。

本论文创新点在：（1）采用高性能的 SSH（Struts+Spring+Hibernate）框架，充分利用 IoC（Inverse of Control）技术实现依赖注入，充分利用代理，降低服务器的工作负载，提高处理能力

（2）实现文档、影音等学习资料的上传以及在线学习。教学试题在线编辑。

（3）同时使用 JSP 技术进行基本页面的设计与功能实现。操作简单，界面友好，高效率，充分体现了网络对现代教育的意义。

（4）代码低耦合度，充分利用了 java 语言的反射技术和 ClassLoader 加载机制，使用动态代理，对容易变化的业务使用代理结构完全抽取出来，采用事务注入，使可维护性大大提高，降低硬编码的污染。

关键字：在线辅助教学系统； SSH； B/S； 可维护性

The Research and Implementation of Online Class Platform based on SSH

ABSTRACT

As the development of Internet and technology, the demand on the education level and efficiency is increasing quickly. The use of teaching resources in China are relatively unified. At the same time, in order to adapt to the change from the exam-oriented education to quality education, the research and Extension of teaching system based on network is increasingly common and particularly important.

At present, there is a considerable part of the auxiliary teaching system is the C/S structure, it increases the client management and maintenance, and it is not conducive to the mobile office. In this design, the online auxiliary teaching system is based on a SSH2 framework. The system adopts B/S mode, no need to install the client. More efficient, more convenient to use. Study on the promotion of online auxiliary teaching system will help to promote the reform of education and teaching, and promote all-round social progress.

Innovation in this paper:

(1)Active use of high-performance frameworkSSH(Struts+Spring+Hibernate), and make full use of IoC(Inverse of Control) technology to multi-user demand and implements Dependency Injection,Make full use of the agent which reduces the server workload, to a certain extent, improve the processing ability.

(2)Implementation of video, documents and other learning materials uploading and teaching questions online editing.

(3)To the basic page design and function using JSP technology to achieve. Simple operation, friendly interface, high efficiency, fully embodies the significance of modern network education.

(4)Low coupling code and making full use of the Java language reflection

technique and ClassLoader loading mechanism. The use of dynamic proxy and the completely extraction of agent structure makes it easy to change the business . The affairs of injection makes the maintainability fo business often changed greatly improved, and it can also reduce the hard coded pollution.

Keyword: The online auxiliary teaching system; SSH; B/S; maintainability

第 1 章 绪论

1.1 在线辅助教学系统开发背景

随着互联网的迅猛发展, Internet 已然成为最流行和主要的信息服务方式, 它正在以其庞大的用户量还有惊人的增长速度显示旺盛鲜活的生命力。为了适应从应试教育向素质教育的教育改革, 学校需要在传统的教学模式以外寻找新的学习模式。在线辅助教学是一种适应新环境、新变化的教育形式。在线辅助教学系统主要服务于学校内的所有教师和学生, 为教师提供更加快捷方便的教学平台, 为学生很好的提供更多元化的学习环境、可以更方便复习, 预习和自查自测的学习渠道。在线辅助教学系统是一个正在兴起的应用领域, 主要利用 WEB 技术、数据库技术等实现网上辅助教学的交互性、实时性、动态性。目前对于网上辅助教学的模式以及其所涉及的各种技术仍然有许多问题需要探讨和解决。WWW 已成为 Internet 上最重要和最具潜力的信息发布、检索和交互方式, 它可以把各种类型的信息资源(如文本、数据、静态图像、音频、视频等)有机的结合起来, 使用户能够在 Internet 上共享。它被广泛地用于资源共享和技术共享, 而且这些技术(Web、多媒体技术、数据库技术以及分布对象技术等)在教育方面的重要性和潜力也变得越来越明显, 尤其在开放的和远程的辅助教学方面。在线辅助教学是利用计算机网络技术、多媒体技术、数据库技术等现代信息技术手段结合发展起来的一种新型教学模式。它使得学校教师以及广大的受教育者突破传统教育的在教育资源(教材、演示、实验等)和教育方法(集中教学、进度统一、单向教授等)方面的限制, 实现了优秀的教育资源和教育方法不再受到时间和空间等的约束。学生能够根据自己的个人学业水平和个人时间情况来安排自己的学习计划, 实现了以往的传统教育无法做到的“个性化定制教育”。

本在线辅助教学系统突破传统教学限制的思想, 目的是能够尽量将现实的活生生的学习环境和方法应用到网络在线辅助教学中, 能够有效提高学生学习的兴趣, 从而实现真正提高网上学习质量的目的。同时本在线辅助教学系统是基于 SSH2(Struts2+Spring+Hibernate)框架的 B/S 结构系统, 便于开发和维护,

有着更好的安全性，在用户机器上不需要安装额外的应用程序，总体框架是建立通用的模块化的网上辅助教学平台。本系统作为传统教学的延伸，网络信息化在线辅助教学的作用正在被人们重视，随着网络技术、数据库技术以及多媒体技术的进步，网络信息化在线辅助教学系统将还会有更快的发展速度和更大的发展空间。

1.2 国内外研究现状

国外在线辅助教学系统的应用比较广泛，应用效果明显

在许多发达国家，人们的生活节奏不一样，很多人的工作不固定，多以许多人更愿意选择半工半读或者在线学习的方式。在线辅助教学系统的出现打破了时间、地点等的限制，从而满足了这部分人的学习要求，同时也使很多学校的教学更加便捷、高效，学生的学习兴趣更高，学习计划和进度也更加自主化。更好的实现教学资源的共享与师生的实时交互。特别是 Internet 技术、多媒体技术和数据库技术等融合，很好的满足了网络教育的需求。另一方面，这些国家政府也是十分重视现代信息技术在教育中的应用。

国内在线辅助教学工作起步晚，发展快速

我国开展计算机辅助教学工作起步较晚，但我国政府十分重视网络教育。在 1999 年国务院转发了教育部制订的“面向 21 世纪教育振兴行动计划”，其中明确提出了实施现代远程教育工程，其后又在全国教育工作第三次会议上提出，到 2010 年初步实现高等教育大众化的目标，建立现代远程教育网络，构建终身教育体系。现在我国的多所试点院校初步摸索出一套网上教育的办学模式，开发出一批教学资源 and 课程，初步形成了校内基于校园网的网络教学与校外远程教学同时进行并能相互融合的开放式办学模式。

我国计算机辅助教学工作已经取得了一大批成果。80 年代初期，我国就有少数科研、教学人员着手开展这项工作，80 年代后期，进行了广泛的学术活动，先后成立了全国计算机辅助教学学会和地方性学会。1992 年国家教委对全国高等学校开展 CAI 的情况进行调查。下面提供一些数据供大家参考。以工科为例，全国现有 288 所普通高等工业学校，在 122 所学校的反馈信息中，117 所学校(本

科 100 所、专科 17 所)开展了这项工作, 5 所尚未开展。除广泛信息交流外, 我国还在“七五”、“八五”期间, 将许多 CAI 课题列入国家重点科技攻关项目, 增加大量投资, 并有计划地组织科研机关、企业、学校的技术力量开发大量的教学软件。我们已经建立了教学软件评审、管理机构和一整套的教育软件登录、管理、评审、推广、发行的制度与方法, 先后制定了《教育软件登录评审方法》、《教育软件评审标准》、《教育软件脚本编写原则》、《教育软件开发、出版、发行暂行办法》, 已经进行了四次教育软件评审, 共通过 300 多套并由出版社正式出版发行。1993 年第 4 季度成立了全国高等工业学校 CAI 协作组, 1994 年 5 月, 理科 CAI 协作组相继成立, 1995 年初“农科协作组”成立, 同时国家教委决定分期分批制订各科 CAI 和试题库研究规划, 对各学科 CAI 和试题进行立项研究, 按项目进行审批立项、拨款资助、成果鉴定和推广使用管理。

1.3 解决的主要问题

系统主要面向教育部门、学校、教师、学生、家长及其他教育工作者, 这些不同身份的人群, 可以在同一个平台上, 根据权限去完成不同的工作。降低教学成本, 共享教学资源。真正为学生及教育工作者带来极大的便利。

在需求分析方面, 要在充分理解在线辅助教学系统实际需求的基础上, 获得业务流程, 需求包括功能需求和非功能需求。通过分析总结出合理的设计和开发思路。

在设计方面, 要考虑辅助教育平台的网络架构和业务特点, 设计要满足系统可维护性, 灵活性和可扩展性。

在实现方面, 在实现前文设计基础上, 采用 SSH 框架进行实现。

1.4 本文的主要工作

本文在现有辅助教学系统基础上, 分析了在线辅助教学的现实需求和业务流程, 并基于 SSH 框架思想, 设计和实现了在线辅助教学系统。

1.5 论文的组织结构

第一章绪论，主要描述在线辅助教学系统的开发背景、国内外现状，本文解决的主要问题和完成的工作。

第二章在线辅助教学系统需求分析，主要概述在线辅助教学系统项目背景、说明，系统目标和解决的问题以及需求问题描述。

第三章在线辅助教学系统架构设计，主要分析在线辅助教学系统设计目标和原则，描述其物理架构和逻辑架构以及其功能架构

第四章在线辅助教学系统详细设计，主要进行在线辅助教学系统建模以及模块设计。

第五章在线辅助教学系统实现与测试，主要描述在线辅助教学系统的总体实现和模块实现。

第六章结论，对在线辅助教学系统的开发与设计进行总结，描述所得结论以及所得收获。

最后向指导教师致谢，说明参考文献，附上英文原文以及译文。

第 2 章 在线辅助教学系统需求分析

2.1 系统概述

在线辅助教学系统是主要面向教育部门、学校、教师、学生、家长及其他教育工作者，这些不同身份的人群，可以在同一个平台上，根据权限去完成不同的工作。降低教学成本，共享教学资源。真正为学生及教育工作者带来极大的便利。

2.1.1 项目背景

网络作为教学的辅助手段，在各高等院校中日益受到教师和学生的青睐，已经逐步成为教学过程的重要支撑。开发和使用在线辅助教学系统必然会成为我院教学工作的重要组成部分。作为学院的教学平台，应该充分利用已有的校园网，为教学创建一个教学环境，使学生可以在这个应用环境中访问到所有的课程资源，并能通过聊天室、留言簿等形式与教师、同学进行交流、讨论;教师不用专门考虑课程的技术实现，而将精力集中于课程内容、教学过程本身。鉴于对在线辅助教学系统的认识，将该系统定位于以培养自主学习能力为特色的在线辅助教学系统。

2.1.2 项目说明

在本系统中将电子教案、多媒体课件等有效地集成为一体，方便学习者进行自主学习、协作讨论。该系统不仅为学生提供了丰富的学习资源，还提供了便利的网络交流空间，使学生在课程学习的过程中，学会合理的规划网络学习、高效地利用网络;同时，该系统也满足教师能及时解决学生提出的问题，加强与学生的沟通。在线辅助教学系统是主要面向教育部门、学校、教师、学生、家长及其他教育工作者，这些不同身份的人群，可以在同一个平台上，根据权限去完成不同的工作。降低教学成本，共享教学资源。真正为学生及教育工作者带来极大的便利。

2.1.3 系统整体系统概述

在线辅助教学系统整体系统采用 Web-B/S 结构，在 JavaEE 应用系统开发中，整合 Struts，Spring 和 Hibernate 三大主流开源框架，用他们分别架构表示层，业务层和持久层并协同工作，可增强应用系统的可维护性和可扩展性，提高代码复用度和开发效率。本文在提出 SSH 整合架构，分析其体系结构和整合原理的基础上，设计了一个跨平台、可伸缩、适合于多种层面、多种对象和多种网络环境的教学支持系统（OnlineClass），该系统实现了教务管理网上教学、网上学习、在线论坛、在线评议和作业管理等功能。实际应用表面 SSH 整合架构为开放、企业级和可伸缩的 JavaEE 应用开发提供了有效支持，提高了开发效率，具有广阔的应用前景。

SSH 体系结构是基于 Struts-Spring-Hibernate 的整合架构，如图 2-1 所示。SSH 是一个由客户端表示层(client presentation layer, CPL)、服务器端表示层(server presentation layer, SPL)、业务逻辑层(business logic layer, BLL)、数据持久层(data persistence layer, DPL)和域模型层(domain mode layer, DML)构成的四层体系结构，其中 SPL 用 Struts 框架描述 VC(model-view-controller, 模型-视图-控制器)模型；BLL 由 Spring 框架的控制反转(inversion of control, IoC)容器协助完成；DPL 用 Hibernate 框架的 ORM(object relation mapping) 完成与后台数据库的交互；此外，SPL 与 BLL 通过 DML 的值对象(value object, VO)进行数据传递；BLL 与 DPL 用 DML 的 VO-PO(persistence object)来描述对象之间的关系。

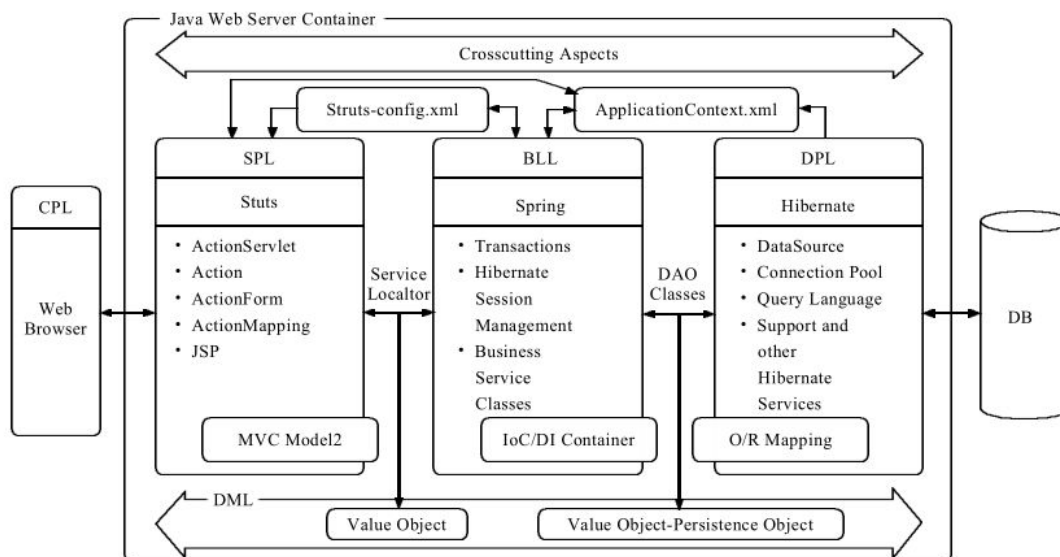


图 2-1 SSH 体系结构

SSH 架构中各层的主要职责如下：

(1)SPL 中 Struts 框架的主要职责： 对客户端表单的输入进行验证；管理请求与响应；提供控制器来完成页面流转和向 BLL 的委托； 返回到客户端页面显示。

(2)BLL 中 Spring 框架的主要职责： 为 SPL 及 DPL 与 BLL 之间提供松散耦合； 处理真实的企业级应用； 事务管理的选择；协调各种逻辑对象之间的依赖关系； 实现 DPL 的业务逻辑。

(3)DPL 中 Hibernate 框架的主要职责： 对数据库进行查询，获得持久化对象 PO； 以 PO 完成对数据记录的增、 删、 改操作。

(4) DML 中 VO 的主要职责： 为各层之间数据交互服务； 在持久层部分描述一个实体并与 PO 进行转换。

2.2 目标和解决的问题

在线辅助教学系统的主要目标是充分利用网络优势,设计一套适合网络教学模式和教学环境;有效管理和使用网络教学资源,更好地指导学生进行学习和实践;形成一套完整的课程教学体系和教学质量评价体系。

2.3 在线辅助教学系统需求问题描述

2.3.1 功能性需求

在线辅助教学是一种适应新环境、新变化的教育形式。在线辅助教学系统主要服务于学校内的所有教师和学生，为教师提供更加快捷方便的教学平台，为学生很好的提供更多元化的学习环境、可以更方便复习、预习和自查自测的学习渠道。根据调研得出该系统的功能性需求如表 2-1 所示。

表 2-1 系统功能性需求

模块名称	子模块名称	功能名称、标识符	描述
课程中心	课程学习	根据目录查询课程	查看目录，点击相应的目录，即可看到该目录及其子目录下的所有课程
		按课程名称查询课程	在课程名称输入框中输入课程名称中任意的关键字，即可进行模糊查询，找到相应的课程
		根据有无学习权限查询课程	选择有权限，可以看到用户有课程学习权限的全部课程；选择无权限，可以看到用户无课程学习权限的全部课程；选择全部，可以看到全部课程
		按发布单位查询课程	在发布单位输入框中输入课程的发布单位，即可找到相应的课程
		根据课程类型查询课程	可以分别只选择多媒体类型的课件、文档类型的课件或者 AICC 标准课件
		对检索到的课程进行排序	可以分别按照课程名称、类型、类别、发布时间、发布单位、学习量、下载量等对查询到的课程进行升序或降序排序
	学习报告	查询学习课程	按照输入的开始和结束日期，查询介于此时间段之间的课程，以及其相对应的考试
学员自助	修改信息	修改信息	修改自己的姓名、登录名、办公电话、手机号码、邮箱、描述等。
	修改密码	修改密码	提供旧密码，完成设置新密码。
	收回权限	收回权限	管理员把管理权限赋予给某个用户后，可以通过此功能收回管理员

				权限
课程管理	课件管理	上传课件	上传课件	输入课件名称，课件作者，选择课件类型，输入学习地址，下载地址，选择课件附件，输入课件描述，进行课件上传
		课件维护	课件查询	输入课程名称中任意关键字，即可进行模糊查询；输入作者名称中任意关键字，即可进行模糊查询；选择课件类型，共享状态，输入创建的起始和终止时间，即可进行查询
			编辑课件信息	选中某一个课件，编辑课件的信息，包括课件名称，索引文件名，课件作者，共享状态，描述信息。
			删除课件	可以全选或者选择某一个课件进行删除
			查看课件信息	点击课件名称超链接，查看课件的详细信息
	课程管理	根据课程名称，是否公开，考试科目，排序类别查询课程信息		输入课程名称中任意关键字，即可进行模糊查询；选择是否公开，查询课程信息； 选择考试科目，查询课程信息； 选择排序类别，以及排序类型，对结果进行排序查询
		增加课程		输入课程名称，作者，课程简介，选择课程类别，相关考试，是否公开，输入有效时间段增加课程，进行下一步操作，根据所选择的课件类型，增加相应的课件。
		编辑课程		编辑课程名称，作者，课程简介，修改课程类别，相关考试，是否公开，修改有效时间段增加课程，进行下一步操作，根据所选择的课件类型，修改相应的课件。
		删除课程		删除课程
	系统管理	授权		赋予某人管理员权限
分类管理		课程分类	增加课程分类	输入课程名称，描述，排序号信息，新建课程分类。
			编辑课程分类信息	修改课程分类的名称，描述，排序号
			删除课程分类	删除某一个或者所有的课程分类
	用户管理		查询用户	输入用户名，登陆名，创建人中的任意关键字，即可进行模糊查询，选择所属组织，创建组织，输入创

			建时间进行查询
		增加用户	输入用户的基本信息增加用户，包括用户名，登录名，登录密码，办公电话，手机邮件等信息
		删除用户	选定某个用户进行删除
	组织管理	查询组织	输入机构 ID，名称，创建人，选择父机构，机构类型，创建时间，创建组织查询组织
		增加组织	输入组织 ID，组织名称，选择父组织，机构类别，排序序号，描述信息进行创建组织
		修改组织信息	修改组织 ID，组织名称，排序号，描述
		删除组织	选定某个组织，然后删除
	设置管理员	给用户设置管理员权限	选定角色列表，选择要赋予角色的人员进行设置
首页	课程中心	课程列表	在首页点击课程中心 more 按钮，进入课程学习模块，得到课程列表
		课程信息	在首页点击课程名称超链接，查看课程的基本信息，学习课程
登陆	登陆	用户登陆	输入用户名、密码，若是用户名、密码相匹配则登陆成功。

2.3.2 需求建模

根据系统需求，建立如图 2-2、图 2-3、图 2-4 的用例模型。模型分为三类用户，分别为系统普通用户、系统培训管理员以及平台系统管理员。

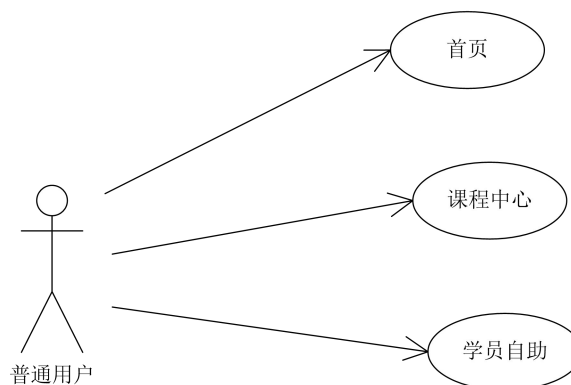


图 2-2 系统普通用户用例模型

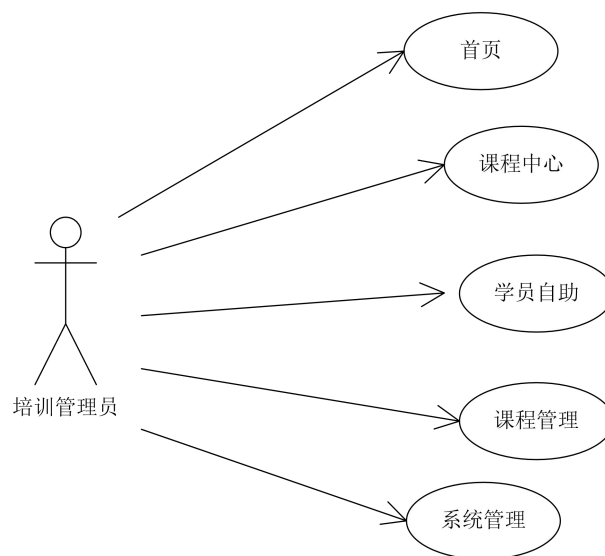


图 2-3 系统培训管理员用例模型

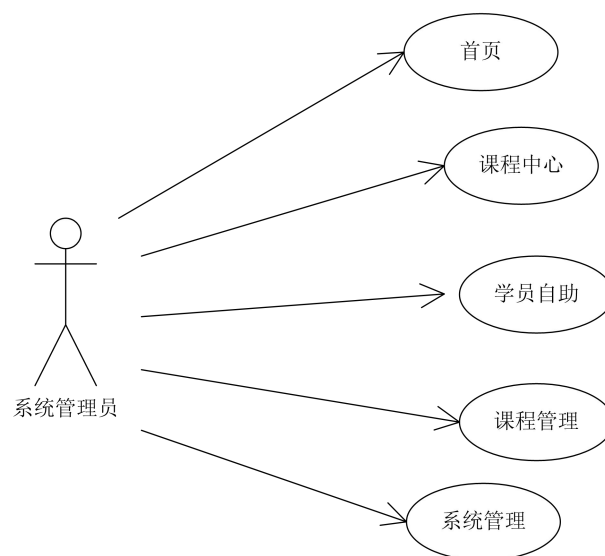


图 2-4 平台系统管理员用例模型

2.3.3 非功能性需求

1. 约束

- (1) 项目时间有限。要求半年内完成。

(2) 要求系统具有良好的可迁移性。

2. 为了提供更好地交互界面，让用户操作方便并且界面清新，适合教学场景，对于用户界面，有如表 2-2 的需求。

表 2-2 用户界面需求

需求名称	详细要求
页面色调	以蓝色和橘黄色为主为主，对比鲜明
页面复杂性	页面应该简单实用，便于操作
页面人性化设计	页面设计应该尽量达到人性化，在需要的地方给出人性化的页面信息提示

3. 同时，为了保证系统正常、稳定的运行，需要使系统满足一定的健壮性、可靠性、效率等质量属性，产品质量属性需求表如表 2-3 所示：

表 2-3 产品质量需求

主要质量属性	详细要求
正确性	能够实现需求中描述的功能模块
健壮性	要有足够的错误和异常处理机制来处理突发事件
可靠性	系统运行要稳定
性能，效率	系统的响应时间以及处理要在可接受的范围内
易用性	各项功能的执行操作简单
清晰性	功能模块的划分应该清晰
安全性	数据库应该要及时备份，接口确保安全
可扩展性	系统也要具有较好可扩展性和可维护性，便于后期开发
兼容性	应该能够兼容不同浏览器
可移植性	系统要与平台无关，便于移植

第 3 章 在线辅助教学系统架构设计

3.1 系统设计目标和原则

在线辅助教学系统的设计目标和原则主要体现在一体化思想、多业务模式和系统可维护性三个方面。

根据系统的需求分析，把在线辅助教学系统进行细化，形成如图 3-1 的在线辅助教学系统功能模块图。

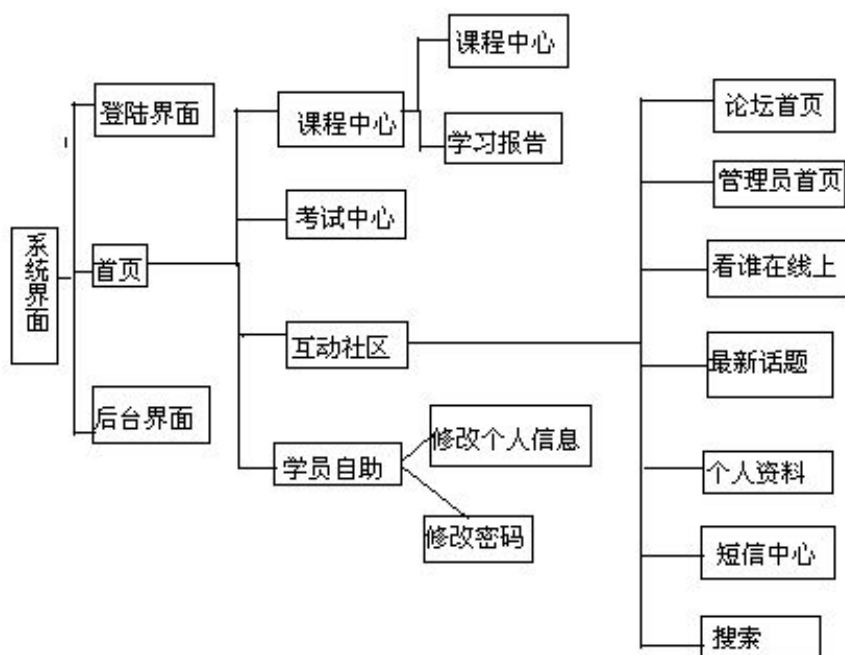


图 3-1 在线辅助教学系统功能模块图

3.2 系统技术架构设计

SSH 整合的基本原理是:

(1)Struts-Spring 的整合

当用户通过 JSP 页面提交请求时, Struts 的 ActionServlet 根据

Struts-config.xml 中的配置, 通过 Spring 的请求代理接口

DelegatingActionProxy 将所接收的用户请求转发给相应的 Struts 的 Action,

以此实现 Strus 与 Spring 的有机整合。下面是管理员登录的配置实例:

①在 struts-config.xml 中, 将 Struts 的 Action 映射到 Spring 的请求代理接

口 DelegatingActionProxy

```
<action path ="/AdminLog" name="AdminUser"
```

```
scope ="request" input ="/admin.jsp"
```

```
type = "org.springframework.web.struts. DelegatingAction-Proxy">
```

```
<forward name="login" path="/view/admincontent.jsp"/>
```

```
</action>
```

②在 Spring 的 beans.xml 中配置 Struts 的相应的 action

```
<bean name ="/AdminLog"
```

```
class="tts.action.AdminLogActoin" singleton="false">
```

```
<property name="adminlog">
```

```
<ref bean="adminlog"/>
```

```
</property>
```

```
</bean>
```

上述配置①与②的关键在于通过给 Spring 的 beans.xml 中 bean 的 name 属性与 Strus 的 struts-config.xml 中 Action 的 path 属性指定一个相同的属性, 从而使用 Spring 的代理接口 DelegatingActionProxy 能找到真正的 Struts 的 Action。

③在 Struts 中访问 Spring 的上下文环境

实现在 Struts 中访问 Spring 的上下文环境的方法是在 struts-config.xml 中将

Spring 的上下文文件以插件的形式加载到 Struts 中即可, 其配置如下:

```
<plug-in className= "org.springframework.web.struts.Conte-
```

```
xtLoaderPlugIn">
<set-property property= "contextConfigLocation" value ="/
WEB-INF/beans.xml"/>
</plug-in>
```

(2) Spring-Hibernate 的整合

Spring-Hibernate 的整合原理是：

①Spring 通过操作 Hibernate 中的 DAO 实现类的 DAO 接口来操作数据库，具体的实现由 Hibernate 来完成。

②让 Hibernate 的每个 DAO 实现类都继承 org.springframework.orm.

hibernate3.support.HibernateDaoSupport 类，通过 getHibernateTemplate()方法返回一个 hibernateTemplate 对象，根据一系列 O/R 映射文件以及数据源或连接池与数据库进行连接，实现“操作一个对象就是操作数据库的一行数据”的目的，从而完成 Spring 层对 DAO 接口的操作的响应。

此外，如图 3-2，域模型中的业务对象贯穿了整个应用的各个层次。在表示层 Struts 的 FormBean 类与业务对象对应；在业务逻辑层 Spring 的业务实现类使用 VO；在持久层 Hibernate 的 DAO 实现类使用 VO-PO。

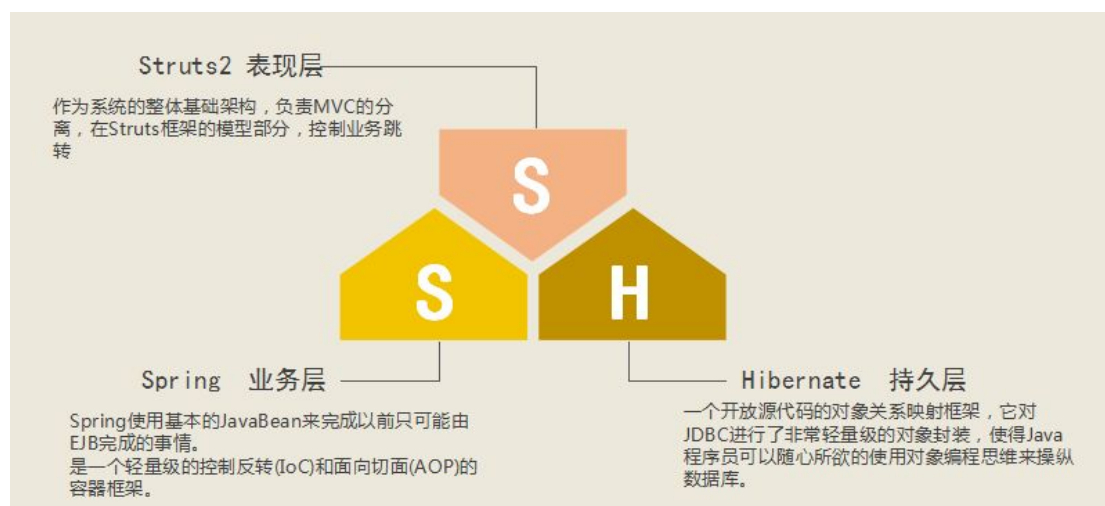


图 3-2 SSH 框架系统

3.3 在线辅助教学系统功能架构

3.3.1 系统功能组成

该在线辅助教学系统的意义在于，可以做到教学资源的规范管理、科学统计和在线实时交流。从而减少教师教学管理与学生学习方面的工作量。目前，我系仍采用传统的以教为主的教学模式，教师无法顾及所有学生对所学内容的理解情况，并且由于授课进度和时间的限制，使师生之间的交流很少，致使教师很难及时获得学生的反馈信息。由以上分析，我们获得系统的功能架构图，如图 3-3 所示

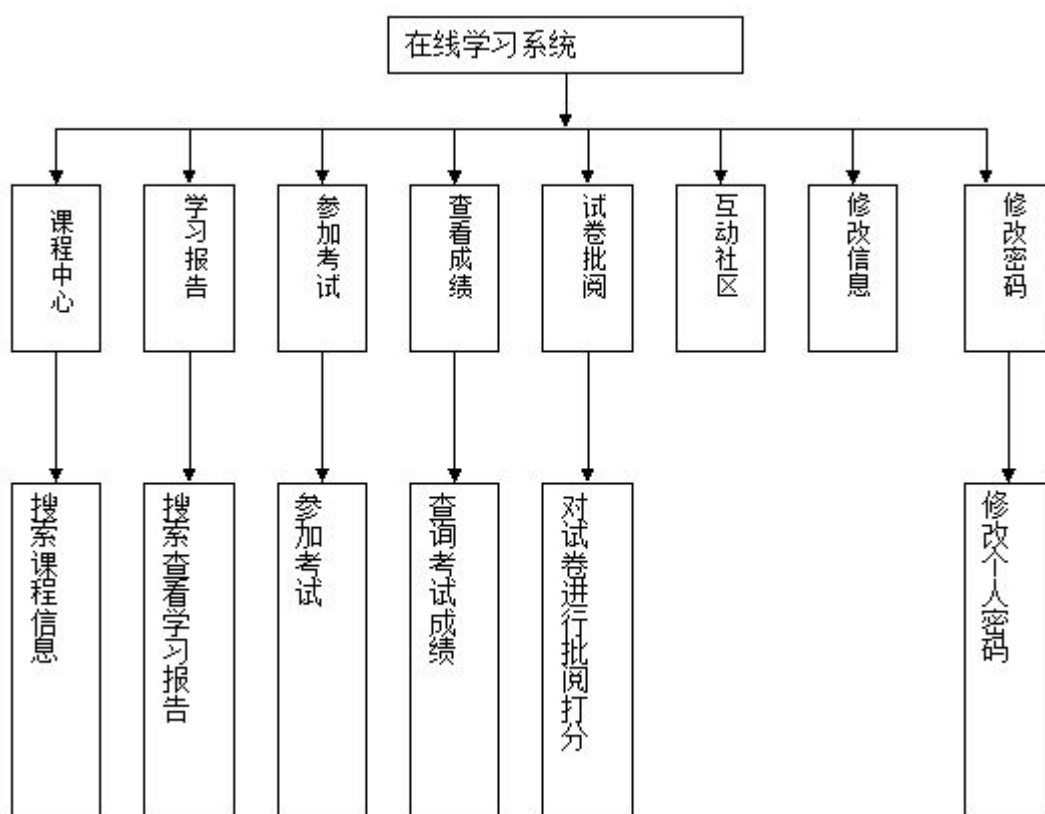


图 3-3 系统功能架构图

第4章 在线辅助教学系统详细设计

4.1 系统建模

经过需求分析和架构设计，我们了解了在线辅助教学系统的业务需求和架构流程。本章在此基础上，进一步分析系统的模型结构和数据库结构。该系统的整体模型结构采用 SSH 框架搭建，系统功能模块模型如图 4-1。

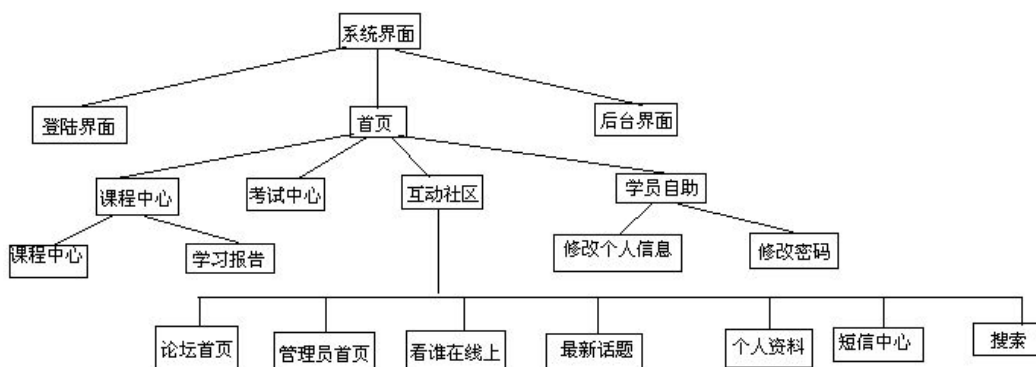


图 4-1 系统功能模块模型

4.2 数据库设计

4.2.1 数据库设计原则

1、尽可能减少数据重复和冗余，进行规范化设计

数据库表的合理设计应该是在满足需求的前提下，让数据的重复量降到最小，尽量按关系数据库系统的方法去设计数据库，以尽量减少数据的冗余。

2、操作设计和结构设计相结合

在数据库结构设计中，有时为了思路清晰以及编程实现的简捷，故意增加一些冗余的数据。这样做虽然有悖于传统的关系数据库方法理论，但是考虑到如果增加的冗余不会明显降低处理效率或者增加存储空间，并且能够对编程实现较大帮助的话，可以采用这种方法。

3、数据结构应该具有相对稳定性

无论采用何种手段处理日常事务，都应能完成对应功能，决不会因采用了计算机而丢失了某些必要功能，或者是彻底改变了原有工作模式和流程。

4.2.2 数据库标识符和状态

基本描述：每个数据表最多只能有一个逻辑主键，即该主键在同一表中必须唯一。本系统中有唯一标识的见表 4-1。

表 4-1 有唯一标识的表

序号	数据表名称	标识符名称	标识符名称属性
1	t_admin	id	管理员 ID
2	t_course	id	课程 ID
3	t_course_category	id	类别 ID
4	t_exam	id	考试 ID
5	t_message	id	消息 ID
6	t_multiplechoice	id	多选 ID
7	t_retopic	id	回帖 ID
8	t_shortanswer	id	简答 ID
9	t_singlechoice	id	单选 ID
10	t_topic	id	帖子 ID
11	t_user_collection	id	用户收藏 ID
12	t_user_course	id	用户课程 ID
13	t_user_exam	id	用户考试 ID
14	t_usermultiplechoice	id	用户多选答案 ID
15	t_users	id	用户 ID
16	t_usersshortanswer	id	用户简单答案 ID
17	t_usersinglechoice	id	用户单选答案 ID

4.3 数据库逻辑设计

逻辑设计是系统中数据库设计的重要步骤之一，其中罗列了文档需求分析中各种功能实现之间的本质关系，通过 ER 图描述实例化了文字的抽象理念。以下就针对本系统加以阐述其逻辑设计。数据表中的基本数据结构 E-R 图如图 4-2

所示:

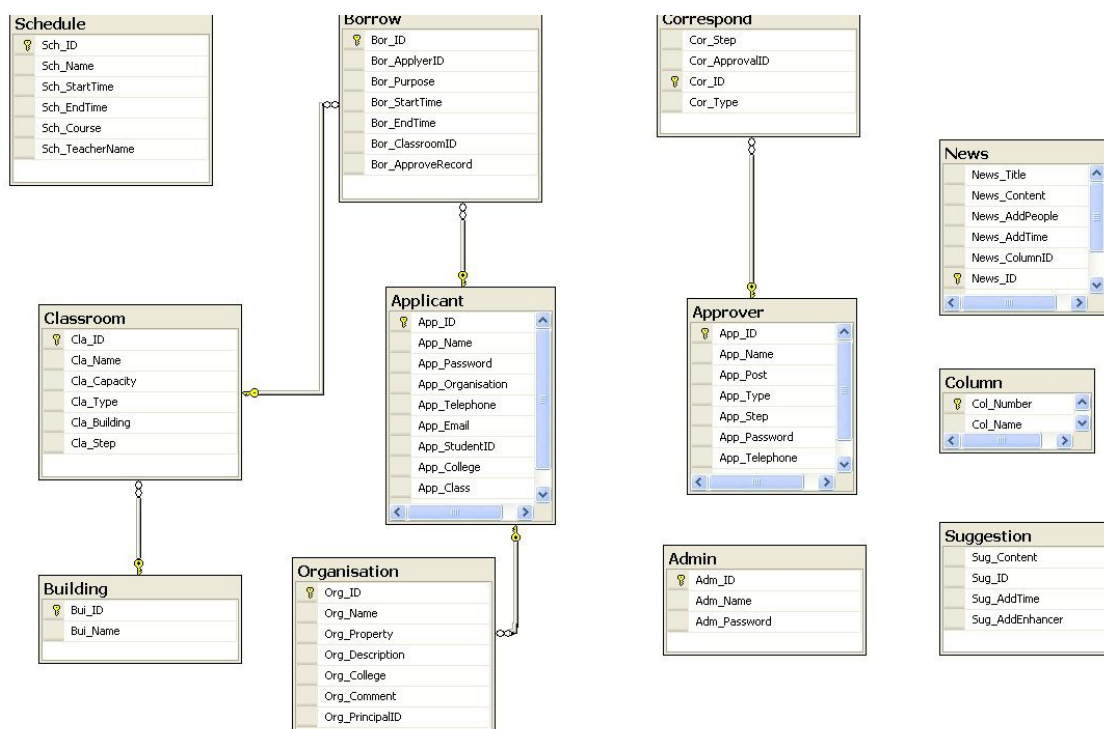


图 4-2 基本数据结构 E-R 图

4.4 详细功能设计

用户登录：输入用户名、密码，若是用户名、密码相匹配则登陆成功。输入内容为用户名和密码，输出内容为网站首页。

课程中心：

课程列表：用户进入系统，看到最新的课程名和课程发布时间。输入内容为进入系统或点击首页按钮。输出内容为课程中心课程学习模块。

查看更多课程列表：在首页点击课程中心 more 按钮，进入课程学习模块，得到课程列表输入内容为：点击课程中心 more 按钮。输出内容为：课程中心/课程学习模块。

课程信息：在首页点击课程名称超链接，查看课程的基本信息，学习课程。输入内容为点击课程名称。输出内容为课程信息。

注销：点注销按钮，退出系统

课程学习：

查询课程：根据输入的课程的相关信息查询，得到符合条件的课程信息。输入内容为课程名称、发布单位、排序类别、学习权限、课程类型、是否包含子类。输出内容为课程列表。课程名称，发布单位支持模糊查询，各字段组合查询；双击列表任意行，显示课程的详细信息

课程列表首页：课程列表实现分页功能，能够通过按钮回到列表首页。点课程列表首页，输出课程列表首页课程。

课程列表上一页：课程列表实现分页功能，能够通过按钮回到列表上一页。输入内容为点课程列表上一页。输出内容为课程列表上一页课程。

课程列表下一页：课程列表实现分页功能，能够通过按钮回到列表下一页。点课程列表下一页，输出课程列表下一页课程。

课程列表末页：课程列表实现分页功能，能够通过按钮回到列表末页。点课程列表末页，输出课程列表末页课程。

转到课程列表任意页：课程列表实现分页功能，能够通过按钮转到列表任意页。输入页号，输出输入页号的课程列表。

课程信息：显示课程的基本信息，课件和附件信息，考试信息。双击课程列表任意一行，输出相应课程信息。

课程学习：点击学习按钮，输出课程视频，只有具有学习权限的课程才允许学习。

学习报告：

查询学习课程：查询某段时间的课程学习情况。输入起止时间，输出学习课程列表。

学习课程列表首页：学习课程列表实现分页功能，能够通过按钮回到列表首页。点课程列表首页，输出课程列表首页课程。

学习课程列表上一页：学习课程列表实现分页功能，能够通过按钮回到列表上一页，点课程列表上一页，输出课程列表上一页课程。

学习课程列表下一页：学习课程列表实现分页功能，能够通过按钮回到列表下一页，点课程列表下一页，输出课程列表下一页课程。

学习课程列表末页：学习课程列表实现分页功能，能够通过按钮回到列表末页。点学习课程列表末页，课程列表末页课程。

转到学习课程列表任意页：习课程列表实现分页功能，能够通过按钮转到列表任意页。输入页号，输出号码页课程列表。

学员自助：

修改个人信息：修改自己的姓名、办公电话、手机号码、邮箱、描述。输入姓名、办公电话、手机号码、邮箱、描述，输出修改成功提示。

返回：点击返回按钮，回到系统首页。

修改密码：提供旧密码，完成设置新密码。输入旧密码、新密码。输出修改成功提示。

课件管理：

上传课件：把制作好的课件，上传到 WEB 服务器。输入课件名称，课件作者，课件类型，学习地址，课件附件，课件描述，输出课件上传成功提示。

返回：点击返回按钮，回到系统首页。

查询课件：根据输入的课件的相关信息查询，得到符合条件的课件列表。输入课件名称，课件作者，课件类型，共享状态，创建的起始终止时间。输出课件列表。支持课件名称，课件作者模糊查询，支持各字段组合查询。

课件列表首页：课件列表实现分页功能，能够通过按钮回到列表首页。点击课件列表首页，输出课件列表首页。

课件列表上一页：课件列表实现分页功能，能够通过按钮回到列表上一页。点课件列表上一页，输出课件列表上一页。

课件列表下一页：课件列表实现分页功能，能够通过按钮回到列表下一页。点课件列表下一页，输出课件列表下一页。

转到课件列表任意页：课件列表实现分页功能，能够通过按钮转到列表任意页，输入页号，输出输入号码页课件列表。

编辑课件：选下某个课件，修改课件的信息。输入修改课件名称，课件作者，共享状态，描述。输出课件列表。

查看相关课程：点击查看相关课程，输出相关课程信息。

关闭：关闭课程信息页面。点击关闭按钮，关闭页面。

编辑索引文件名：修改课件的索引文件名。点击编辑索引文件名，返回到编辑模块。

取消：取消修改课件信息，返回课件维护模块。点击取消按钮，返回到课件维护模块。

删除课件：删除选定的某个或者多个课件，输入选定待删除的课件，输出删除成功，已经被课程引用的课件不允许被删除。

全选：选定所有的课件。点全选按钮，选定所有课件，选择框打勾。

取消全选：不选定任何课件。点击取消全选按钮，不选定任何课件，选择框为空。

查看课件详细信息：查看课件详细信息，包括课件名称，课件作者，课件类型，共享状态，附件数目，描述，相关课程数。点击课程列表中课件名称。输出课件信息。

观看课件：观看课件播放效果。点击观看课件，播放课件。

查看相关课程：点查看相关课程，输出相关课程信息。

关闭：关闭课程信息页面，点击关闭按钮，关闭页面。

返回：点击返回按钮，回到课件维护模块。

课程管理：

查询课程：根据输入的课程的相关信息查询，得到符合条件的课程列表。输入试题名称，作者，创建时间，输出课程列表。

课程列表首页：课程列表实现分页功能，能够通过按钮回到列表首页，点课程列表首页，输出课程列表首页课程。

课程列表上一页：课程列表实现分页功能，能够通过按钮回到列表上一页。点课程列表上一页。输出课程列表上一页课程。

课程列表末页：课程列表实现分页功能，能够通过按钮回到列表末页。点击课程列表末页，输出课程列表末页课程。

转到课程列表任意页：课程列表实现分页功能，能够通过按钮转到列表任意页。输入页号，输出输入号码页课程列表。

增加课程：

新建课程：增加课程信息。输入课程名称，作者，课程简介，课程类别，

相关考试，是否公开，有效时段。输出相关课件类型。

重置课程信息：清空所有输入框内容，重新输入课程信息。点击重置按钮，清空输入框内容。

上一步：回到新建课程的上一层菜单。点击上一步按钮，输出课程管理首页。

下一步：继续新建课程，增加课件信息。点击下一步按钮，输出相关课件类型。

相关课件类型：根据选择的课件类型，得到现有的课件信息。输入课件类型，输出课件列表。

保存课件信息：选中课程对应的课件，保存课件信息。输入选定课程，点保存按钮。输出添加课程成功提示。

上一步：回到相关课件类型的上一层。点上一步按钮，新建课程。

编辑课程：

编辑课程信息：输入课程名称，作者，课程简介，课程类别，相关考试，是否公开，有效时段。输出相关课件类型。

重置课程信息：恢复输入框的课程信息为修改前的数据，点重置按钮，恢复输入框数据。

上一步：回到编辑课程的上一层菜单，点击上一步按钮。返回课程管理首页。

下一步：继续编辑课程，增加课件信息，点击下一步按钮，输出相关课件类型。

增加课件：继续增加课程对应的课件。点击增加按钮，输出课件列表。

保存课件信息：选中课程对应的课件，保存课件信息。输出修改课程成功提示。

上一步：回到相关课件类型的上一层。点击上一步按钮，编辑课程信息。

取消：取消对课件信息修改。点击取消按钮，输出课程原有课件信息。

删除课程：删除选定的某个或者多个课程。选中课程，点击删除按钮，删除课程成功提示。

课程学习情况：

查询课程学习情况：输入学习人，学习部门，选择时间，输出课程学习情

况列表。

课程学习情况列表首页：课程学习情况列表实现分页功能，能够通过按钮回到列表首页。

课程学习情况列表上一页：课程学习情况列表实现分页功能，能够通过按钮回到列表上一页。点击课程学习情况列表上一页，输出课程学习情况列表上一页课程。

课程学习情况列表下一页：课程学习情况列表实现分页功能，能够通过按钮回到列表下一页。点击课程学习情况列表下一页，输出课程学习情况列表下一页课程。

课程学习情况列表末页：课程学习情况列表实现分页功能，能够通过按钮回到列表末页。点课程学习情况列表末页，输出课程学习情况列表末页课程。

转到课程学习情况列表任意页：课程学习情况列表实现分页功能，能够通过按钮转到列表任意页。输入页号，输出输入号码页课程学习情况列表。

导出课程学习情况：导出课程学习情况，包括学习时间，打印日期，员工编号，学习人，部门，学习次数，下载次数。点导出按钮，导出课程学习况报表 excel 文档。

全选：选定所有的课程，取消选定所有课程。点全选按钮，选定所有课件，选择框打勾。取消选定所有课程，选择框勾去掉。

系统管理：

查询用户：根据输入的用户的相关信息查询，得到符合条件的用户列表。输入用户名称，输出用户列表。

授权：把系统管理员权限赋予给某个成员。选中某个用户，输出授权成功提示。

分类管理：

增加课程分类：输入名称，描述，排序号，输出增加课程类别成功提示。

编辑课程分类：编辑课程分类信息，包括名称，描述，排序号。点分类名称或者点编辑按钮，输出修改课程分类信息成功提示。

删除课程分类：删除某个或者多个课程分类信息。选定课程分类，输出删除成功提示。

全选：选定所有的课程分类，点击全选按钮，课程列表第一列打勾。

取消全选：取消选定所有的课程分类，点击取消全选按钮，课程列表第一列去掉打勾。

课程列表首页：课程列表实现分页功能，能够通过按钮回到列表首页。点课程列表首页，输出课程列表首页课程。

课程列表上一页：课程列表实现分页功能，能够通过按钮回到列表上一页。点课程列表上一页，输出课程列表上一页课程。

课程列表下一页：课程列表实现分页功能，能够通过按钮回到列表下一页。点课程列表下一页，输出课程列表下一页课程。

课程列表末页：课程列表实现分页功能，能够通过按钮回到列表末页。点课程列表末页，课程列表末页课程。

转到课程列表任意页：课程列表实现分页功能，能够通过按钮转到列表任意页。输入页号，输出输入号码页课程列表。

用户管理：

查询用户：根据输入的用户的相关信息查询，得到符合条件的用户列表。输入用户名，登陆名，所属组织，创建时间，创建人，创建组织，输出用户列表。

用户列表首页：用户列表实现分页功能，能够通过按钮回到列表首页。输入点用户列表首页。输出用户列表首页。

用户列表上一页：用户列表实现分页功能，能够通过按钮回到列表上一页。点用户列表上一页。输出用户列表上一页。

用户列表下一页：用户列表实现分页功能，能够通过按钮回到列表下一页。点用户列表下一页。输出用户列表下一页。

用户列表末页：用户列表实现分页功能，能够通过按钮回到列表末页。点用户列表末页输出用户列表末页。

转到用户列表任意页：用户列表实现分页功能，能够通过按钮转到列表任意页。输入页号。输出所选号码页用户列表。

增加用户：输入用户名，登录名，登录密码，所属组织，办公电话，手机，邮件，描述，用户类型。输出用户管理首页。

取消：取消增加新的用户。点击取消按钮，输出用户管理首页。

修改用户：修改用户信息。输入用户名，登录名，登录密码，所属组织，办公电话，手机，邮件，描述。输出用户管理首页。

取消：取消修改用户服务信息。点取消按钮，返回用户管理首页。

删除用户：删除选定的某个或者多个用户信息。点删除按钮，返回用户管理首页。

查看用户详细信息：查看用户的详细信息，包括用户名，登录名，登录密码，所属组织，办公电话，手机，邮件，描述，创建人，用户类型。点详细链接，输出用户详细信息。

全选：点全选选择框，选定所有用户，选择框打勾。再次点全选选择框，不选定任何用户，选择框为空。点选择框，全选或取消全选。

组织管理：

查询组织：根据输入的组织的相关信息查询，得到符合条件的组织列表。输入机构 ID，机构名称，父机构，机构类型，创建时间，创建人，创建组织。输出组织列表。

组织列表首页：组织列表实现分页功能，能够通过按钮回到列表首页。点组织列表首页，输出组织列表首页。

组织列表上一页：组织列表实现分页功能，能够通过按钮回到列表上一页。点组织列表上一页，输出组织列表上一页。

组织列表下一页：组织列表实现分页功能，能够通过按钮回到列表下一页。点组织列表下一页，输出组织列表下一页。

组织列表末页：组织列表实现分页功能，能够通过按钮回到列表末页。点组织列表末页，输出组织列表末页。

转到组织列表任意页：组织列表实现分页功能，能够通过按钮转到列表任意页。输入页号，输出输入号码页组织列表。

增加组织：增加新的用户。输入组织 id,组织名称，父组织，机构类别，创建人，创建部门，创建时间，排序，描述。输出组织管理首页。

取消：取消增加新的组织。点取消按钮，返回组织管理首页。

修改组织：修改组织信息。输入组织 id,组织名称，机构类别，排序，描述。输入用户管理首页。

取消：取消修改组织信息。点取消按钮，返回组织管理首页。删除组织：删除选定的某个或者多个组织信息。点删除按钮，输出组织管理首页。

设置管理员

选择对象：给选定的某个或者用户设置管理员权限。输入用户列表。输出管理员权限。

删除对象：删除选定的某个或者用户设置管理员权限。输入用户列表，输出管理员列表。

保存对象：保存选定的某个或者用户设置管理员权限。输入用户列表。输出管理员权限。

关闭：关闭设置管理员页面。点关闭按钮，输出系统首页。

第 5 章 在线辅助教学系统实现与测试

5.1 系统总体实现

系统实现后的登录界面中，三种用户（系统管理员、教师、学生）的登录页面类似，图 5-1 为管理员用户的登录界面：



图 5-1 管理员用户登录界面

系统完成后，学生用户与教师用户登录后进入的界面类似，区别在于，学生用户有参加考试的权限，而没有教师用户所拥有的试卷批阅权限。除此之外，均可在课程中心查询课程信息，在学习报告页面搜索查看学习报告，查询考试成绩、在互动论坛进行在线交流讨论、并在学员自助中修改个人信息。登录后的主页面如图 5-2 所示：



图 5-2 学生及教师用户登录后主页面

系统管理员完成系统管理、用户管理、课程管理、考试管理、论坛管理以及个人信息管理。其中，系统管理包括添加角色信息和查看角色信息；用户管理包括添加用户信息和查看用户信息；课程管理包括添加类别信息、查看类别信息、添加课程和查看课程；考试管理包括题库信息管理、试卷信息管理、考试信息管理以及成绩信息管理；论坛管理包括添加板块信息、查看板块信息、查看主题信息和查看消息信息；个人信息管理包括查看个人信息和修改个人信息。系统管理员的所有功能在实现中全部体现在左侧的功能菜单树中。系统管理员登录后的主界面如图 5-3 所示：



图 5-3 系统管理员界面

第 6 章 结论

6.1 项目总结

在教育领域中，计算机的发展正逐渐成为备受关注的教学媒体和教育管理工具，它使当今的教学观念、观念和形式以及课堂的教学结构，教育理论和思想发生了巨大变革。该辅助教学系统的开发采用了 SSH

（Struts-Spring-Hibernate）框架，选择三种框架的组合，充分利用了它们各自的优点。在该辅助教学系统中，Struts 作为表现层，通过标签库的形式实现了页面信息的显示，配合了 Struts 控制部分。为了让系统具备简单、可测试性以及松耦合的特点，系统服务层的框架采用的是 Spring,降低了系统开发的复杂性。Hibernate 通过持久层将数据库存储操作从服务层中分离出来，实现了对象与数据库之间的映射。

本辅助教学系统的优势:采用表示层、业务逻辑层、数据访问层以及数据层四层架构的形式;系统是基于互联网的 B/S 结构，用户只要用浏览器就可以使用该系统的全部功能;操作简单方便，操作人员不需要特殊的计算机技术基础，只需要按照菜单的提示进行操作，教师、学生、管理员便可各自工作;管理和维护比较简单，有较强的可扩展性;采用 SSH 组合架构，模块化设计。

6.2 系统存在的问题及展望

由于本人技术水平以及时间都比较有限，本系统暂时还存在一些没有解决的问题。

(1)系统容错性不足，有些模块还没有对数据进行验证，有待在以后进行改进。

(2)缺少对网上教学的评价以及质量的跟踪。有了跟踪还有评价能更好提高教学水平，促进教师提升专业素养。

在线辅助教学系统提供给学生和教师一个方便及时的在线交流学习的平台，在线辅助教育是目前教学发展的趋势，还需要更多的研究，也需要对系统做进一步的完善。

致谢

在本文即将完成之际，回想本科阶段的学习生活，我衷心感谢我的导师黎峰。在本次论文撰写过程中，得到了黎峰老师的悉心指导与帮助，并对我的论文初稿提出了宝贵的修改意见。论文的工作是在导师的悉心指导和帮助下完成的，在论文选题、研究和写作阶段，导师都给予本人诸多指导和无私的帮助。顾老师敏锐的洞察力，严谨的治学态度和感人的敬业精神令本人受益匪浅。

最后，感谢在我成长的过程中给予我帮助的所有老师、朋友和同学们。

参考文献

- [1] Bruce Eckel. Thinking in Java[M]. Upper Saddle River, New Jersey, USA: Prentice Hall, 2006
- [2] 陈道鑫,宋绍云,袁中旺,等. ExtJS 框架在 Web 软件开发中的应用[J]. 电脑知 识与技 术 2011, 07(9): 2044-2047
- [3] 林信良. Spring2.0 技术手册[M]. 北京: 电子工业出版社, 2005
- [4] (美)威尔德, 斯尼德等著, 赵利通译. SpringFramework 2 入门经典[M]. 北 京: 清 华大学出版社, 2009
- [5] 林寒超,张南平. Hibernate 技术的研究[J].计算机技术与发展,2006, 16(11): 112-113,116
- [6] 田珂, 谢世波, 方马.J2EE 数据持久层的解决方案[J].计算机工程, 2003, 29(22): 93-95
- [7]林士敏, 《计算机辅助教学基础教程》, 浦东电子出版社, 2001 年 1 月
- [8]王逢贤, 《学与教的原理》, 高等教育出版社, 2000 年 7 月
- [9]张华, 《课程与教学论》, 上海教育出版社, 2000 年 11 月
- [10] Ian Smmerville, ((Software Engineering)), 机械工业出版社, 2003 年 1 月
- [11] Philip Metzger&John Boddie, ((Managing a Programming Project)), 电子工业 出版社, 2002 年 8 月
- [12]马林艺, 张喜英, 钱春升编著:《软件工程》, 机械工业出版社, 2006 年 1 月
- [13]赵子江编著:《多媒体技术应用教程》, 机械工业出版社第 4 版, 2005 年
- [14] Struts[EB/OL].<http://Jakarta.apache.org/struts/index.html>
- [15] Spring.<http://www.springframework.org/>[OL].
- [16] Hibernate.<http://www.hibernate.org/>[OL].
- [17].计文柯:《Spring 技术内幕--深入解析 Spring 架构与设计原理(第二版)》, 机械工业出版社 2012 年
- [18].李刚:《轻量级 Java EE 企业应用实战:Struts2+Spring3+Hibernate 整合开发 (第 3 版)》, 电子工业出版社 2012 年

[19].明日科技: 《实战突击: Java Web 项目整合开发》 电子工业出版社 2011 年

附录 1 英文原文

The Spring Framework

Beginning POJOs

Lightweight Java Web Development Using Plain Old Java Objects in Spring, Hibernate, and Tapestry (Beginning from Novice to Professional)

The Spring Framework is an open source Dependency Injection (DI) and Aspect Oriented Programming (AOP) lightweight container and full-stack Java EE application framework leading the pack in the movement towards making enterprise Java development simpler, faster, less error-prone, and more enjoyable. Spring helps you assemble components declaratively into a working system. Spring is based on the principle that good object-oriented design should come first and not be hindered by the choice of technology being used. Spring grew from the experiences of a seasoned and dedicated team of developers led by Rod Johnson dealing with real applications on a daily basis and looking for a way to address end-to-end application requirements in a consistent fashion across application tiers. The Spring Framework brings back some of the good object-oriented aspects that were missing from the previous versions of Java EE platform, such as the use of POJOs, programming by contract using interfaces, and clean, layered architecture. Spring eliminates the need to deal with most middle-tier glue as opposed to heavyweight component models like EJB 2.X, allowing you to build truly portable, object-oriented Java SE and Java EE applications. The Spring Framework is licensed under the terms of the Apache License, Version 2.0.

Coding to Interfaces

The first aspect of the Spring Framework that you'll notice is that it makes your Java code cleaner and easier to test by promoting the practice of coding to

interfaces, a practice well-known in object-oriented programming. Exposing a public contract (the interface) and being able to provide a suitable implementation of that interface seems like a no-brainer to most Java programmers, but unfortunately, previous versions of the Java EE framework, particularly EJB, made this very difficult with its heavy and bloated container contracts. The side effect of the pre-3.0 EJB APIs was that although they promoted a component-oriented development style, their particular implementation disregarded good object-oriented practices.

Developers had to go through hoops implementing their own frameworks to test their components in isolation outside of the J2EE container. Most developers who came from building good object-oriented applications, when thrown under the wheels of the EJB bus, decided to take the path of most resistance, using EJB in an inverted fashion by first building robust, interface-driven “by contract” , object-oriented applications and using EJB as a layer to leverage multithreading, object pooling, and other container services. After all, there is much more to J2EE than just EJB, but unfortunately, the previous ill-designed versions of the EJB specification took the limelight from the rest of the specification, especially from the real gems of the specification—in my humble opinion, JNDI and JMS. Those developers who took the path of least resistance ended up with so-called components that were not truly reusable, were hard to test, and contained more plumbing code than actual application code of value to the business.

Frameworks like Spring emerged from the philosophy of using the best parts of the specification, masking those parts that are cumbersome, and simplifying the APIs, all the while making good, object-oriented design priority number one. As you’ ll soon learn, Spring lets you approach the design of your application using interfaces in a loosely coupled way without compromising good, object-oriented principles.

Dependency Injection

Spring is primarily known as a Dependency Injection container. Dependency Injection is the more appropriate term coined by Martin Fowler, Rod Johnson, and the PicoContainer team in late 2003 for what was once known as “Inversion of

Control.” An alternative view is that DI is a strategy or flavor of IoC. Most explanations of DI also referred to it as the Hollywood Principle: “Don’ t call us, we’ ll call you.”

Early in my forays into DI containers and the Spring Framework, I found some of the explanations of IoC/DI overly complex. The best explanation I can give for DI is that it is a technique by which you externalize the choice of which implementation of a given interface you’ ll use at runtime. In Spring, a system is designed using interfaces, and the Spring framework plugs or injects the right implementations at runtime. In Spring you typically inject specific implementations of an interface into other beans via an XML configuration file referred to as an application context.

Therefore in a Spring application your POJOs or beans do not have their dependencies explicitly constructed in the code. Instead, dependencies are “injected” by the container at runtime. For example, in a typical Spring Web application using the DAO pattern, you’ ll have interfaces defining the DAOs and interfaces defining the services. In the implementation of a service, a property for a DAO is defined (using the interface) along with a setter or a constructor that takes the DAO as a parameter. The Spring application context XML configuration contains a definition of the specific implementation of the DAO interface that will be injected into the service class (using the setter or the constructor provided). At runtime, the Spring container using the application context will correctly wire the beans and the collaborators needed to fulfill its functions. This is contrary to the lookup-based approached of EJB 2.X, in which the service locator pattern along with factories and straight object construction is used to wire all collaborators in a system. In a typical EJB 2.X system, a service locator is used as a way to pull the dependencies (typically from JNDI) into a component (all this is done programmatically from within the component itself). This doesn’ t mean that JNDI isn’ t a powerful abstraction for resource management, but the implementation in EJB 2.X forced this “pull” usage of JNDI and a dependence on the JNDI API (making it harder to test without having JNDI available and therefore in most cases a container) and requiring a lot of code to perform lookups. As we saw in Chapter 5,

the EJB3 specification allows for JNDI-bound resources to be injected into a component in a loosely coupled fashion via Java annotations. Similarly, Spring allows for the injection of JNDI-bound resources.

Dependency Injection moves the responsibility of configuring application objects and resource lookups to the container in a declarative fashion. The main advantages of DI are that it removes the requirement of business objects to implement any container-specific interfaces and eliminates redundant lookup code or the need for service locator implementations. By removing container-specific dependencies, objects can be tested without a container, and legacy code can also be easily adapted to work in new environments.

I mentioned before that the implementations to be injected with a resource need to provide either a setter or a constructor that is used by the container to inject the given resource. This brings us to the different types of dependency injection: constructor-based injection and setter-based injection. Both methods provide advantages and disadvantages. Renowned technologist Martin Fowler provides the explanation of IoC/DI in an online article at www.martinfowler.com/articles/injection.html.

In this book and in the Spring implementation of the TechConf application, I used setter-based DI, mostly because it makes it easier in the scenarios when multiple resources are being injected and prevents the proliferation of too many constructors, which I found is the result of trying to account for the possible combinations or required parameters. Of course, by providing constructors that take the required resources (and without a no-arguments constructor) you effectively guarantee that the object being constructed will be in a valid state upon creation.

In combination with the practice of coding to interfaces, DI provides for a very testable environment that makes it easier to take a test-first approach to development, providing reduced development effort and increasing test coverage and overall quality. Using a different application context during testing, one could, for example, inject mock implementations of certain collaborators to test a component in isolation.

附录 2 译文

The Spring Framework

Beginning POJOs

Lightweight Java Web Development Using Plain Old Java Objects in Spring, Hibernate, and Tapestry (Beginning from Novice to Professional)

Spring 框架是一个开放源代码的依赖注入（DI）和面向切面编程（AOP）的编程轻量级容器栈。Java EE 应用框架使企业 Java 对开发更简单快，更不容易出错，开发过程也更愉快。Spring 可以帮助你形成一个工作系统部件组装说明。Spring 的原理是基于良好的面向对象设计的，不受选择技术的限制。Spring 来自一个经验丰富的开发团队，该团队由 Rod 约翰逊带领，他们寻找一种方式来解决在应用层端到端的要求一致的方式。Spring 框架带来一些好的面向对象方面，从 Java EE 平台的以前版本的缺失，如 POJO 编程使用，合同的使用接口，和清洁，分层架构。Spring 消除了需要处理的最中间层与重量级的组件模型和 EJB 2.x，允许你建立真正的便携式，面向对象的 Java SE 和 Java EE 应用。Spring 框架的许可 Apache 许可证的 2.0 版。

面向接口编程

Spring 框架的第一个方面，你会注意到的是，它会让你的 Java 代码更干净、更容易促进编码接口的实践检验，实践面向对象编程中的知名。公开一个公共接口，提供一个合适的实现方法，对大部分大部分 java 程序员来说会觉得不明智，原来版本的 JavaEE 框架，特别是 EJB，使容器冗杂臃肿。EJB pre-3.0 API 的副作用是他们促进构件发展模式好的面向对象的特定实现，忽视了良好的面相对象的实践。开发商费尽了力气实施自己的框架在 J2EE 容器外部隔离元件测试。大多数建立良好的面向对象的应用程序的开发商，在 EJB 的车轮下，决定采取最小阻力的途径，首先通过建立强大的以倒置方式使用 EJB，接口驱动的“契约”，面向对象的应用程序，使用 EJB 层利用多线程，对象池，和其他集装箱服务。毕竟，J2EE 不是 EJB，但不幸的是，以前的设计不良的版本的 EJB

规范模仿了其余的规范，特别是从规范真正的宝石 JNDI, JMS——这只是我的愚见。采用最小的路径的这些开发商结束了所谓的不能真正重复使用的部分，这很难测试，并含有更多的管道代码，业务价值，实际的应用程序代码。

像 Spring 这样的框架的出现是使用规范的最好的部分哲学，隐藏繁琐的部分，简化 API，所有涉及以封装、面向对象设计为第一要务。你会很快学会，Spring 让你以松耦合和面向对象的方式使用你的涉及程序。

依赖注入

Spring 是个依赖注入容器。依赖注入是由 Martin Fowler, Rod Johnson, 2003 年底的 PicoContainer 团队提出的恰当的词。曾经被称为“控制反转。另一种观点是，DI 是一个 IoC 的策略。DI 的大多数情况下解释为好莱坞原则：“不用给我们，我们会找你。”

在我研究 DI 和 Spring 框架的早期，我发现一些关于 Ioc 或者 DI 的解释过于复杂。我能为 DI 给出的最好的解释是它是一种将你的运行时的接口所选择的具体化的一种的技术 在 Spring 中，系统用接口来设计，Spring 框架在运行时恰当的应用其中。在 Spring 中，你通常注入接口为其他 bean 类通过 XML 配置文件进行具体实现。

因而在一个 Spring 的应用程序中你的 POJO 和 beans 没有构造依赖。代替依赖关系的是运行时容器的“注入”。例如，在一个典型的 Spring Web 应用中，使用 DAO 模式，你必须定义 DAO 接口和接口定义的服务。在服务的实现中，属性定义（使用接口）以及一个 setter 或构造函数作为参数。Spring 应用程序上下文的 XML 配置包含一个 DAO 接口的，将被注入到服务类的具体实现定义（使用 setter 或构造函数提供）。在运行时，使用应用程序上下文的 Spring 容器将正确的链接 beans 和需求以满足功能需求。这与基于查找的 EJB2.X 相反，在 EJB2.X 中伴随着工厂和直接对象构造服务定位器模式是用来连接所有的合作者在一个系统中。在一个典型的 EJB2.X 系统中，服务定位器被用来作为一种拉动的依赖关系（通常从 JNDI）到一个组件（这一切都是从组件本身内编程完成）。这并不意味着 JNDI 是不是一个强大的抽象资源管理，但在 EJB2.x 中

的强制执行该“拉”JNDI的使用，并在JNDI API的依赖（使其难以测试，而无需使用JNDI因此在大多数情况下，一个容器），并需要大量的执行查找的代码。EJB3规范允许JNDI绑定的资源被注入到一个组件通过Java注解松散耦合的方式。同样，Spring允许用于注射JNDI绑定的资源。

依赖注入配置移动应用程序对象和资源查找到容器中以声明的方式。DI的主要优点是，它消除了业务对象的要求实现任何特定容器的接口和消除冗余的查找代码或需要服务定位器实现。通过移除容器特定的依赖关系，对象可以在没有容器进行测试，并遗留代码也可以容易地适应于新的环境中工作。

我之前提到过，要与资源被注入的实现需要提供任一设定器或所使用的容器的给定资源注入一个构造。这给我们带来了不同类型的依赖注入：基于构造函数的注入和基于setter方法注入。这两种方法都提供了优点和缺点。知名技术专家马丁·福勒提供的IoC/DI的解

www.martinfowler.com/articles/injection.html。

在这本书中，以及在Spring实现TechConf的应用程序，我用基于setter的DI，主要是因为它使得它在场景更容易当多个资源被注入，防止过多的构造函数的扩散，这是我发现的是结果试图解释可能的组合或所需的参数。当然，通过提供采取必要的资源构造（并且没有一个无参数的构造函数），你有效地保证了所构造的对象将是在建立一个有效的状态。

与编码到接口的做法相结合，DI提供了一个相当可测试的环境，使得它更容易采取一个测试第一的发展方针，提供降低开发难度，提高测试覆盖率和综合素质。在测试过程中使用不同的应用方面，例如人们可以独立注入某mock一个组件的测试。