# 基于SSH的视频教学平台

[摘要] 在大数据时代，要跟上时代的潮流，不能止步于传统的课堂教学。随着网络的进一步发展，网络课堂已经成为必然趋势。本文介绍了使用面向对象设计的方法，实现教师添加课程并对课程进行管理、添加测试、上传课件等功能，同时设立管理员对教师和视频教学课程进行审核和管理。前端使用流行的Bootstrap前端框架来快速搭建页面，后台使用Struts2, Spring, Hibernate框架来实现视频教学平台，以实现教育资源的共享的目的。最终完成了使用面对象的方法和SSH框架来实现视频教学平台的基本功能。

[关键词] SSH框架；视频教学；教育资源共享；网络课堂

# Based on SSH Video Teaching Platform

[Abstract] In the age of big data, we must keep up with the trend of the time instead of sticking to traditional classroom teaching. With the further development of the network, the online classroom has become the inevitable trend. In this essay, by using object-oriented design method, teachers can add courses and manage them, add tests, upload teaching courseware, and many other functions. Meanwhile, administrators can be set up to audit and manage the teachers and their video teaching courses. With front end using popular Bootstrap front frame to quickly build a page, the background uses the Struts2, Spring, Hibernate framework to implement video teaching platform, to achieve the purpose of the education resource sharing. Ultimately this achieves the basic functions of video teaching platform by using object-oriented design method and SSH framework.

[Keyword] SSH framework; Video teaching; Education Resources Sharing; Network classroom

# 目录

[摘要 I](#_Toc482370010)

[Abstract II](#_Toc482370011)

[第1章 引言 1](#_Toc482370012)

[1.1论文研究背景与意义 1](#_Toc482370013)

[1.2 研究内容与目标 2](#_Toc482370014)

[第2章 开发环境及相关技术的介绍 3](#_Toc482370015)

[2.1 Java简介 3](#_Toc482370016)

[2.2 MVC简介 3](#_Toc482370017)

[2.3 SSH架构 4](#_Toc482370018)

[2.3.1 Struts 4](#_Toc482370019)

[2.3.2 Spring 5](#_Toc482370020)

[2.3.3 Hibernate 6](#_Toc482370021)

[2.4 Maven简介 7](#_Toc482370022)

[2.5 Bootstrap简介 7](#_Toc482370023)

[2.6 Tomcat服务器 8](#_Toc482370024)

[2.7 开发环境的搭建 8](#_Toc482370025)

[第3章 需求分析 9](#_Toc482370026)

[3.1 系统功能 9](#_Toc482370027)

[3.2 用例图 9](#_Toc482370028)

[3.3 用例规约 12](#_Toc482370029)

[第4章 概要设计 15](#_Toc482370030)

[4.1 系统框架设计 15](#_Toc482370031)

[4.2 类图 16](#_Toc482370032)

[4.3 顺序图 22](#_Toc482370033)

[4.4 系统活动图 22](#_Toc482370034)

[第5章 详细设计 24](#_Toc482370035)

[5.1数据关系模型 24](#_Toc482370036)

[5.2 实体-关系模型 24](#_Toc482370037)

[5.3 数据字典 26](#_Toc482370038)

[5.4 对象-关系映射 29](#_Toc482370039)

[第6章 系统实现 31](#_Toc482370040)

[6.1 系统代码 31](#_Toc482370041)

[6.2 系统截图 39](#_Toc482370042)

[结 论 43](#_Toc482370043)

[致 谢 44](#_Toc482370044)

[参 考 文 献 45](#_Toc482370045)

# 第1章 引言

## 1.1 论文研究背景与意义

随着我国教育事业和多媒体教学设施的快速发展，越来越多的人可以有机会接受教育，促使人们对教育的质量和要求就有了更高的要求。同时，教育也慢慢往公开方向发展，让人们有机会选择更适合自己的课程来学习提升自己，共享教育资源。视频教学平台是人们空闲时间学习和提升自己的重要场所，可以共享教育资源以提高利用率。现在我国的教育事业发展速度，大部分人都有机会可以接收到学校的教育。任何事物都是两面的，机会和选择的增多，也使得很多人不珍惜，现在很多高校存在学生上课不认真、课后不努力的现象就体现了这一点。

社会对人才的知识要求越来越高。80后、90后、乃至00后仍有很多人没有机会接收教育，部分原因是自己本身学不好不想上学、家庭条件不允许继续深造等。我国教育不像美国教育，出来社会后仍可以回校学习深造，以我国中学的教育理念和制度，出社会后很难再回校读书。

视频教学平台的出现，弥补了这块教育体系的空缺。不仅让学校的同学拥有学习其他相关专业的知识，实现教学资源的最大限度的利用，减少相关的资源浪费；学生课后可以对知识再复习巩固，减轻教师的教学负担和同学课后复习的困难，有利于推动学生的学习动力，促进学习氛围；同时在大数据时代，我们不能止步于传统的课堂教学，要跟上时代的潮流，随着网络的进一步发展，网络课堂已经成为必然趋势；社会上一些想学习的人即使没有机会上好学校，也可以在视频教学平台上学习到和学生一样的课程和知识。

开发视频教学平台，有利于学校的学科发展和建设，能够进一步提升学生的自学能力。有利于进一步加强学校的学术氛围和社会的学习氛围。在社会进行再教育，也极大地提升人们的知识文化水平，有利于社会的安定和整体文化实力的提升。同时也进一步地完善了我国的教育系统。

## 1.2 研究内容与目标

本毕业设计采用面向对象的方法，运用SSH框架、Oracle数据库等现有的技术，设计开发一个基于SSH框架的视频教学系统平台。主要实现教师添加课程并对课程进行管理、添加测试、上传课件等功能，同时设立管理员对教师和视频教学课程进行审核和管理。前端使用流行的Bootstrap前端框架来快速搭建页面，后台使用Struts2, Spring, Hibernate框架来实现视频教学平台，以实现网上教学和教育资源的共享的目标。

# 第2章 开发环境及相关技术的介绍

## 2.1 Java简介

[Java](http://baike.baidu.com/item/Java/85979)是[面向对象](http://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1)的开发编程语言，具有功能强大和简单易用两个特征。Java具有[健壮性](http://baike.baidu.com/item/%E5%81%A5%E5%A3%AE%E6%80%A7)、面向对象、[多线程](http://baike.baidu.com/item/%E5%A4%9A%E7%BA%BF%E7%A8%8B)、[分布式](http://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F/19276232)、动态性、[安全性](http://baike.baidu.com/item/%E5%AE%89%E5%85%A8%E6%80%A7)、简单性、平台独立与可移植性等特点[1]。Java可以编写各种各样的应用程序，如[Web应用程序](http://baike.baidu.com/item/Web%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F)、[分布式系统](http://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F%E7%B3%BB%E7%BB%9F)和[嵌入式系统](http://baike.baidu.com/item/%E5%B5%8C%E5%85%A5%E5%BC%8F%E7%B3%BB%E7%BB%9F)应用程序、[桌面应用程序](http://baike.baidu.com/item/%E6%A1%8C%E9%9D%A2%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F)等。它在设计时就被承诺：Write Once, Run Anywhere[2].

Java上世纪九十年代设计出来到现在已经有二十多年，有生命力的东西才能够经住历史的考验，现在仍然作为一门最流行的编程语言之一，深受程序员的喜爱，可见java语言的强大与易用。

## 2.2 MVC简介

在以前Model1模式下，做出来的系统几乎全部是由JSP页面组成的，JSP页面不仅要接收和处理用户的请求，还要对请求进行处理后直接做出响应。只用少量的JavaBean来连接数据库的访问等操作。虽然这种模式实现简单，很适合极快地开发一些小型的项目。但因为把显示和控制的任务都交给JSP页面来处理，导致代码重用底，后期也难以扩展和维护[3]。

Model2模式已经是以MVC架构为基础的设计模式，MVC思想[4]将一个应用分成三个部分：Model(模型)、View（视图）、Controller(控制器)，这三部分各司其职，有分工也有合作，可以提高系统的可维护性和可扩展性，如图2-1。事件的发生由控制器来进行负责，控制器根据事件的类型来决定要调用什么样的视图或模型。



图2-1 MVC模式的Java Web运行流程

## 2.3 SSH架构

SSH框架是框架Struts2、Spring框架、Hibernate框架的集成框架，是开源的比较典型的基于MVC思想的框架。其中Struts负责进行对系统进行MVC分离，模型部分负责控制业务跳转；Hibernate提供面向对象的方法对持久化的数据进行存取，使程序员可以采用面向对象的方法来操作数据库；Spring负责整合和管理Struts2框架和Hibernate框架。

2.3.1 Struts

Struts2是一个以MVC设计模式为基础的Web应用型框架[5]。在MVC的设计模式下，模型和视图之间的数据交流传递是通过Struts2作为控制器(Controller)来建立的。Struts2框架在Struts1框架的基础上，去掉Struts1不合理的部分，再整合WebWork优秀部分，其实和Struts1差别很大。

Struts2的最前面的控制器是FilterDispatcher，它能最先接收和过滤用户的请求。FilterDispatcher的本质其实是一个Servlet。现在的Struts中已经把FilterDispatcher改名为StrutsPrepareAndExecuteFilter，这个命名更加清晰和明确。模型是对事物本身建立的一个模型，所有的业务逻辑、数据的存储都应该放在模型中。视图就是看得到的东西，即上网看到的页面都可以称为视图。在系统启动的时候Web容器会先进入web.xml，所以程序员要先在web.xml中指定FilterDispatcher来指定用户的什么请求需要过滤器过滤，然后让struts执行。Struts2的原理图见图2-2。

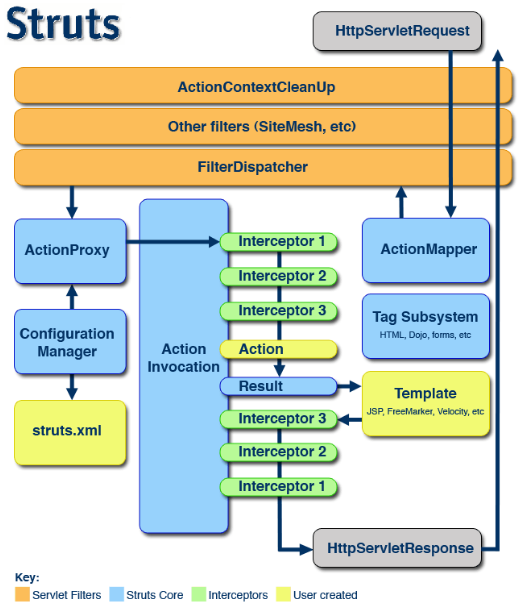


图2-2 Struts2的原理图

2.3.2 Spring

Spring是一个开源的框架。Spring开发流行的原因就是因为企业应用开的过程比较复杂，而Spring则可以使用基本的JavaBean来代替EJB，不仅如此它还提供很多的企业功能。Spring是一个轻量级的容器框架，控制反转(IoC)和面向切面(AOP)是它的两个重要特征。Spring的核心是轻量级(Lightweigt)的容器，它是实现IoC(Inversion of Control)容器和非侵入式(No intrusive)的框架，它提供AOP(Aspect-orientedprogramming)概念的实现方式；提供对持久层(Persistence)，事务(Transaction)的支持，提供MVC Web框架的实现，并对一些常用的企业级API提供一致的模型封装，是一个全方面的应用程序框架[6]。还有就是，现在的各种框架(如：Struts, Hibernate等)，Spring也提供了和它们无缝整合的方法。这也说明了它不仅专注于某一层的解决方案，可以说Spring是企业级应用开发的“一站式”选择，Spring贯穿表现层，业务层，持久层，Spring并不是想取代这些已有的框架，而是对它们进行无缝整合。从图2-3可以看到，使用Spring框架时，必须使用Spring Core Container(即Spring核心容器)，主要由org.springframework.core、org.springframework.beans、org.springframework.context和org.springframework.expression四个包和它的子包构成，以提供Spring IoC容器的支持。

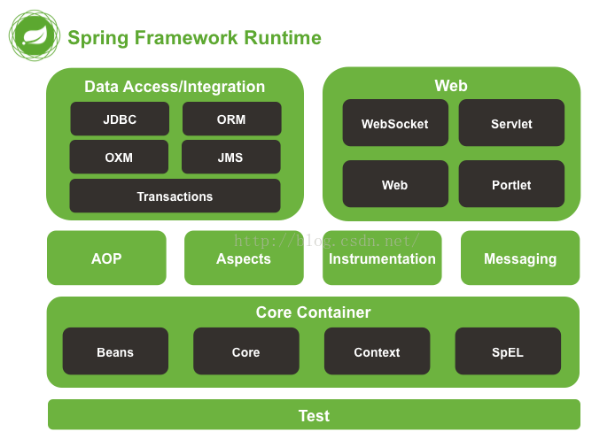


图2-3 Spring框架的组成结构图

2.3.3 Hibernate

Hibernate是一个开源的对象和数据库表的关系映射的框架。JDBC被Hibernate进行了十分轻量级的封装，Hibernate将普通的java类与数据库表建立起映射关系。Hibernate可以自动生成SQL语句，自动执行，使得Java程序员可以随心所欲的使用对象编程思维来操纵数据库[7]。Hibernate将数据库表和程序的普通Java类进行映射处理，对数据库的数据进行的操作进行了对象封装，使程序员不需要精通数据库知识，只要会面向对象编程就可以对数据进行存取。Hibernate不仅管理普通Java类到数据库表的映射关系，还提供数据查询和存储数据的方法，可以大大减少开发人员使用SQL语句和JDBC来处理数据问题的时间。

Hibernate是优秀的ORM框架，是因为它有以下优势：

* 开源、免费的License，可以对源码进行修改定制。
* 轻量级的封装，不会引入其他的复杂问题，方便调试。
* 扩展性强，API对外开放，需要时可以自己进行扩展。
* 开发者活跃，框架有稳定的发展和维护保障。

## 2.4 Maven简介

Maven是一个项目管理的工具，它管理项目的策略是“约定优于配置（CoC）”。它不仅用于把源代码构建成可发布的项目（包括编译、打包、测试和分发），还可以生成报告、生成Web站点等[8]。Maven有如下特点：

* Maven能够自动化地进行构建，比如：清理、编译、测试、生成报告、打包和部署。
* Maven极大地清除重复的构建。
* Maven抽取构建生命周期的模型，它吸收了很多其他的构建脚本和构建工具的精华，根据很多项目的实际需求来进行思考总结。如果参照这个模型，则可以避免很多没必要的错误。
* Maven的插件，Maven为几乎全部的构建任务提供了已经实现的插件，而且还有很多其他稳定的插件来完成任务，它还可以自己来实现插件。
* Maven统一标准化构建过程。

## 2.5 Bootstrap简介

Bootstrap是由Twitter的设计师Mark Otto和Jacob Thornton两个人一起合作开发的，是现在极受开发者喜欢的前端框架。Bootstrap是以[HTML5](http://baike.baidu.com/item/HTML5)和[CSS3](http://baike.baidu.com/item/CSS3)为基础开发的，它在[jQuery](http://baike.baidu.com/item/jQuery)的基础上进行了更好的封闭和发展，在这个过程中慢慢形成一种自己特有的网站风格，而且兼容大部分jQuery插件[9]。Bootstrap提供了漂亮合理的HTML和CSS的规范，它是由动态CSS语言和[Less](http://baike.baidu.com/item/Less)写成。有了bootstrap，程序员也可以快速地写出漂亮的前端页面，而不再完全需要美工的支持。同时它对移动端的支持也是它的一大特点，能够能页面在移动端和PC端都有很好的展示。

## 2.6 Tomcat服务器

Tomcat是Java Web项目中最著名的开源的Web容器，不仅简单容易使用，而且比较稳定，既可以拿它来自己学习，也可以用它对商业新产品进行发布。Tomcat的强大是因为它不仅有Web容器的基本功能，还提供JAAS和JNDI的绑定等。Tomcat和Java语言一样，可以在任何操作系统上运行，具有平台无关性的特点[10]。很多的程序员都喜欢Tomcat，因为使用它运行时占用的系统资源小，它的扩展性很好，支持邮件服务、负载平衡等应用系统常用到的功能；更重要的是它还在不断的完善和发展中，任何一个程序员只要你有兴趣都可以修改它或者在它里面加入新的功能。Tomcat 是一个轻量级的小型的Web服务器，适合在中小型系统和并发访问的用户不非常多的情况下使用，是开发和调试Web应用程序的第一选择。

## 2.7 开发环境的搭建

* 下载并安装JDK7.0(https://www.oracle.com)
* 下载并安装eclipse-jee-kepler-SR2-win32-x86\_64（https://www.eclipse.org）
* 下载Maven并在Eclipse中安装配置好（http://maven.apache.org）
* 下载并安装Oracle（[https://www.oracle.com](https://www.oracle.com/index.html)）
* 在Eclipse工具中，安装Struts2, Spring, Hibernate插件。

# 第3章 需求分析

## 3.1 系统功能

视频教学平台的用户类型分为普通用户、教师用户和管理员用户。教师用户除了具有普通用户所有的属性和操作外，还具有添加课程、管理课程、上传课件、添加测试、回复留言等操作；管理员用户可以添加教师和审核管理教师课程。

普通用户的基本操作：

（1）用户的注册、登录，没有登陆则不能观看视频。

（2）维护个人信息，包括个人信息的修改和登录密码的修改。

（3）下载课程资料。

（4）进行课程测试。

（5）给教师留言和查看留言信息。

教师用户的基本操作：

（1）维护个人信息，包括个人信息的修改和登录密码的修改。

（2）添加、查看、删除课程

（3）上传课件

（4）添加测试

（5）回复留言

管理员用户的基本操作：

（1）维护个人信息，包括个人信息的修改和登录密码的修改。

（2）审核和管理教师课程

（3）添加教师

## 3.2 用例图

用例图[11]用来表示系统做了哪些事情的，可以帮助分析系统有哪些功能，以及明确系统内部和系统外部的交互。从图3-1到图3-3中，可以看到系统具体的功能，同时可以清楚的知道哪些用户具有哪些功能。



图3-1 学生用户用例图



图3-2 教师用户用例图



图3-3 管理员用户用例图

## 3.3 用例规约

用例图只能在总体上大概地说明了系统的用户和功能服务，简洁明了。所以，用例模型不仅要包括用例图，还包括了用例规约[12]。

用例编号：UC01

用例名称：用户学习课程

简要说明：用户选择课程进行学习

参与者：用户

涉众及其关注点：教师希望用户可以在课程中学到自己想要的东西

前置条件：登陆

后置条件：无

扩展点：无

触发事件：用户点击观看视频

基本流：

1．用户浏览页面，选择相应的课程

2．用户登陆成功

3．用户点击课程进行学习

备选流：

\*a. 在基本流的任何一步骤用户退出系统或关闭浏览器

2a. 系统提示用户登陆后才能学习课程

成功场景：学习课程成功

失败场景：

1．用户登陆失败

2．系统异常

特殊需求：无

未解决问题：无

用例编号：UC02

用例名称：教师上传课程

简要说明：教师上传课程和课程相关的信息

参与者：教师

涉众及其关注点：教师希望可以成功上传课程，管理员希望教师上传的课程合理。

前置条件：教师登陆成功

后置条件：课程上传成功后需等待管理员审核

扩展点：无

触发事件：教师提交课程

基本流：

1．教师点击“添加课程”菜单

2．填写课程的基本信息并提交

3．教师选择添加章节和视频，为课程继续添加信息

4. 教师选择视频文件

5. 提交信息

备选流：

\*a. 在基本流的任何一步骤用户退出系统或关闭浏览器

2a. 填写的信息不符合规范

4a. 选择的视频文件与允许上传的文件格式不符合，或视频文件大小过大

成功场景：基本流，添加课程成功

失败场景：

1．教师登陆失败

2．填写的课程信息有误

3．视频格式不正确或大小过大

特殊需求：无

未解决问题：无

# 第4章 概要设计

## 4.1 系统框架设计

该系统为前端页面展示和后台业务处理两部分。前端使用Bootstrap框架，页面简单处理即可获取优美的展现效果。后台使用流行的SSH框架，Struts2负责分层和过滤用户请求，并控制页面的跳转；Hibernate负责所有的持久化操作；Spring负责管理Struts2和Hibernate。

后台具体可划分为4层。

* Action层（控制器层)：接收前端页面数据的传输，调用业务层，负责页面的跳转。
* Service层（业务逻辑层)：处理系统的业务。
* Dao层（持久层）：负责所有对象的数据库操作，如增、删、改、查。
* Entity层(实体层)：普通的Java实体类。

通过以上的分层，可以提高应用程序的扩展性、维护性、伸缩性、重用性、管理性。Action层获取前端数据，调用业务处理层进行处理，如果需要持久化处理则业务层调用持久层来进行数据的存取，最后由Action控制层来决定返回哪个逻辑视图显示给用户。具体的系统框架如图4-1所示：



图4-1 系统框架图

## 4.2 类图

根据需求分析里面的系统功能和用例规约，可以抽取出以下类图：管理员、课程附件、章节、评论、课程、留言、留言子类、教师、测试、用户、视频，如表4-1到表4-7所示：

表4-1 User类图

|  |
| --- |
| 用户类：User |
| - userID 用户ID  - username 用户名  - password 密码  - sex 性别  - email 邮箱  - address 地址  - phone 电话  - vacation 职业  - status 状态  - intro 简介  - school 学校  - img 头像 |
| + updatePassword()  + updateUser()  + jump()  + logout() |

表4-2 Admin类模型

|  |
| --- |
| 管理员类：Admin |
| - userID 用户ID  - username 用户名  - password 密码  - sex 性别  - email 邮箱  - address 地址  - phone 电话  - vacation 职业  - status 状态  - intro 简介  - school 学校  - img 头像 |
| + updatePassword()  + updateUser()  + findAllTeacher()  + deleteTeacher()  + adminAddTeacher()  + jump()  + logout() |

表4-3 Teacher类图

|  |
| --- |
| 教师类：Teacher |
| - userID 用户ID  - username 用户名  - password 密码  - sex 性别  - email 邮箱  - address 地址  - phone 电话  - vacation 职业  - status 状态  - intro 简介  - school 学校  - unit 单位  - img 头像 |
| + updatePassword()  + updateUser()  + jump()  + logout() |

表4-4 Comment类图

|  |
| --- |
| 课程评论类：Comment |
| - commentId 评论ID  - content 评论内容  - commentDate 评论时间  - User 用户  - Course 课程 |
| + submitComment()  + courseComment() |

表4-5 Annex类图

|  |
| --- |
| 课件类：Annex |
| - annexId 课件ID  - path 课件路径  - uploadContentType 课件类型  - uploadFileName 课件名称  - Course 课程 |
| + download()  + downloadPage()  + uploadAnnexPage()  + uploadAnnex()  + uploadFile() |

表4-6 Course类图

|  |
| --- |
| 课程类：Course |
| - userID 课程ID  - courseName 课程名称  - intro 课程简介  - period 课程学时  - status 课程状态  - classify 分类  - uploadDate 课程创建时间  - remark 备注  - Teacher 教师（对象） |

续表4-6

|  |
| --- |
| + videoPlay()  + classify()  + courseScan()  + addCourse()  + findAllCourses()  + findAllCoursesByStatus()  + checkCourse()  + findAllCoursesByTeacherId()  + chapterDetail()  + adminChapterDetail()  + videoDetail()  + adminVideo()  + editCourse()  + updateCourse()  + editChapter()  + updateChapter()  + editVideo()  + updateVideo()  + deleteCourse()  + deleteChapter()  + deleteVideo()  + uploadAnnex()  + uploadVideo()  + addVideo()  + addChapter()  + scanMyCourse() |

表4-7 Message类图

|  |
| --- |
| 留言类：Message |
| - messageId 留言ID  - content 留言内容  - messageDate 留言时间  - status 留言状态  - fromWho 谁留言  - toWho 给谁留言 |
| + studentScanSubMessage()  + revertMessage()  + subMessageDetail()  + teacherScanMessage()  + teacherRevert()  + deleteMessage()  + sendMessageToTeacher() |

表4-7 Test类图

|  |
| --- |
| 测试类：Test |
| - testId 测试ID  - testName 测试题目  - itemA 选项A  - itemB 选项B  - itemC 选项C  - itemD 选项D  - answer 答案  - detail 答案详细  - Course 课程 |
| + addTestPage()  + saveTest()  + testPage() |

## 4.3 顺序图

用户所有操作都要先经过Action，当用户发送一个请求时，Action获取到用户请求的数据，必要时把数据封装成对象，再调用业务层对数据进行处理，若需要到数据库的操作，则由业务层来调用持久层来完成，最后由Action来返回页面给用户。具体的顺序图模型如图4-2所示：



图4-2 用户请求顺序图

## 4.4 系统活动图

当用户浏览页面时，系统处于没有登录的状态。当用户登录成功后，系统处于已经登录的状态，可以进行各种各样的操作。若未进行登录，则不能进行任何有关数据的操作。如果用户退出登陆后，系统处于没有登录的状态，可以重新进行登录，或退出应用程序。具体的系统活动图[13]，如图4-3所示：



图4-3 系统活动图

# 第5章 详细设计

## 5.1 数据关系模型

根据需求分析和用例分析，此系统的数据库可以由如下实体构成：

（1） 用户表User（用户编号，用户名，密码，性别，邮箱，地址，电话，职业，状态，简介，学校）

（2） 管理员表Admin（管理员编号，用户表）

（3） 教师表Teacher（教师编号，单位，用户表）

（4） 课程表Course（课程编号，课程名，课程简介，学时，状态，分类，上传时间，备注，教师）

（5） 课件表Annex（课件编号，课件路径，课件类型，课件名，课程表）

（6） 章节表Chapter（章节编号，章节名，课程表）

（7） 评论表Comment（评论编号，评论内容，评论时间，用户表，课程表）

（8） 留言表Message（留言编号，留言内容，留言时间，状态，谁留言，给谁留言）

（9） 评论表Comment（评论编号，评论内容，评论时间，用户表，课程表）

（10）测试表Test（测试编号，测试内容，选项A，选项B，选项C，选项D，答案，答案详情）

（11） 视频表Video（视频编号，视频标题，视频路径，上传时间，视频名称）

## 5.2 实体-关系模型

系统的实体-关系（Entity-Relationship，ER）模型[14]，如图5-1和图5-2所示：



图5-1 数据库概念结构图



图5-2 数据库物理结构图

## 5.3 数据字典

系统的数据字典如表5-1到表5-7所示：

表5-1 用户表（User）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 主键 | 外键 |
| 用户ID | userId | Number | 255 | TRUE | FALSE |
| 用户名 | userName | Varchar2 | 255 | FALSE | FALSE |
| 密码 | Password | Varchar2 | 255 | FALSE | FALSE |
| 性别 | Sex | Varchar2 | 255 | FALSE | FALSE |
| 邮箱 | email | Varchar2 | 255 | FALSE | FALSE |
| 地址 | address | Varchar2 | 255 | FALSE | FALSE |
| 电话 | phone | Varchar2 | 255 | FALSE | FALSE |
| 职业 | Vocation | Varchar2 | 255 | FALSE | FALSE |
| 状态 | Status | Varchar2 | 255 | FALSE | FALSE |
| 简介 | intro | Varchar2 | 255 | FALSE | FALSE |
| 学校 | school | Varchar2 | 255 | FALSE | FALSE |

表5-2 管理员表（Admin）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 主键 | 外键 |
| 管理员ID | adminId | Number | 10 | TRUE | FALSE |
| 用户ID | userId | Number | 10 | FALSE | TRUE |

表5-3 教师表（Teacher）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 主键 | 外键 |
| 教师ID | teacherId | Number | 10 | TRUE | FALSE |
| 教师单位 | unit | Varchar2 | 255 | FALSE | FALSE |
| 用户ID | userId | Number | 10 | FALSE | TRUE |

表5-4 课程表（Course）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 主键 | 外键 |
| 课程ID | courseId | Number | 10 | TRUE | FALSE |
| 课程名称 | courseName | Varchar2 | 255 | FALSE | FALSE |
| 课程简介 | Intro | Varchar2 | 255 | FALSE | FALSE |
| 学时 | period | Varchar2 | 255 | FALSE | FALSE |
| 状态 | Status | Varchar2 | 255 | TRUE | FALSE |
| 分类 | classify | Varchar2 | 255 | FALSE | FALSE |
| 上传时间 | uploadDate | Date |  | FALSE | FALSE |
| 备注 | Remark | Varchar2 | 255 | FALSE | FALSE |
| 教师ID | teacherId | Number | 10 | FALSE | TRUE |

表5-5 留言表（Message）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 主键 | 外键 |
| 留言ID | messageId | Number | 10 | TRUE | FALSE |
| 留言内容 | Content | Varchar2 | 255 | FALSE | FALSE |
| 留言时间 | messageDate | Date |  | FALSE | FALSE |
| 状态 | Status | Varchar2 | 255 | FALSE | FALSE |
| 谁留言 | fromWho | Varchar2 | 255 | FALSE | FALSE |
| 给谁留言 | toWho | Varchar2 | 255 | FALSE | FALSE |

表5-6 章节表（Chapter）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 主键 | 外键 |
| 章节ID | chapterId | Number | 10 | TRUE | FALSE |
| 章节名称 | chapterName | Varchar2 | 255 | FALSE | FALSE |
| 课程ID | courseId | Number | 10 | FALSE | TRUE |

表5-7 视频表（Video）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 主键 | 外键 |
| 视频ID | videoId | Number | 10 | TRUE | FALSE |
| 视频标题 | title | Varchar2 | 255 | FALSE | FALSE |
| 视频路径 | path | Varchar2 | 255 | FALSE | FALSE |
| 视频类型 | uploadContentType | Varchar2 | 255 | FALSE | FALSE |
| 视频名称 | uploadFileName | Varchar2 | 255 | FALSE | FALSE |
| 章节ID | chapterId | Number | 10 | FALSE | TRUE |

表5-8 评论表（Comment）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 主键 | 外键 |
| 评论ID | commentId | Number | 10 | TRUE | FALSE |
| 评论内容 | Content | Varchar2 | 255 | FALSE | FALSE |
| 评论时间 | commentDate | Date |  | FALSE | FALSE |
| 课程ID | courseId | Varchar2 | 10 | FALSE | TRUE |
| 用户ID | userId | Varchar2 | 10 | FALSE | TRUE |

表5-9 课件表（Annex）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 主键 | 外键 |
| 课件Id | courseId | Number | 10 | TRUE | FALSE |
| 课件路径 | Path | Varchar2 | 255 | FALSE | FALSE |
| 课件类型 | uploadcontentType | Varchar2 | 255 | FALSE | FALSE |
| 课件名称 | uploadFileName | Varchar2 | 255 | FALSE | FALSE |
| 课程Id | courseId | Nubmer | 10 | FALSE | TRUE |

表5-10 测试表（Test）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 主键 | 外键 |
| 测试Id | testId | Number | 10 | TRUE | FALSE |
| 题目 | testName | Varchar2 | 255 | FALSE | FALSE |
| 选项A | itemA | Varchar2 | 255 | FALSE | FALSE |
| 选项B | itemB | Varchar2 | 255 | FALSE | FALSE |
| 选项C | ItemC | Varchar2 | 255 | FALSE | FALSE |
| 选项D | itemD | Varchar2 | 255 | FALSE | FALSE |
| 答案 | Answer | Varchar2 | 255 | FALSE | FALSE |
| 答案详情 | detail | Varchar2 | 255 | FALSE | FALSE |
| 课程Id | courseId | Number | 10 | FALSE | TRUE |

## 5.4 对象-关系映射

对象-关系映射（Object-Relation Mapping，ORM）[15]是说明对象与数据库的表映射关系，将Java系统中的对象自动映射到关系数据库中。Hibernate提供了实现这种映射的模式，它使用映射元数据的方法来说明对象-关系的映射细节，使得Hibernate中间件能在任何一个Java应用的业务逻辑层和数据库层之间充当桥梁。如课程表的映射配置文件如下：

<hibernate-mapping>

<class name="com.jmu.video.entity.Course" table="T\_COURSE">

<id name="courseId" type="java.lang.Integer">

<column name="COURSEID" />

<generator class="increment" />

</id>

<property name="courseName" type="java.lang.String">

<column name="COURSENAME" />

</property>

<property name="intro" type="java.lang.String">

<column name="INTRO" />

</property>

<property name="period" type="java.lang.String">

<column name="PERIOD" />

</property>

<property name="status" type="java.lang.String">

<column name="STATUS" />

</property>

<property name="classify" type="java.lang.String">

<column name="CLASSIFY" />

</property>

<property name="uploadDate" type="java.util.Date">

<column name="UPLOADDATE" />

</property>

<property name="remark" type="java.lang.String">

<column name="REMARK" />

</property>

<many-to-one name="teacher" class="com.jmu.video.entity.Teacher">

<column name="TEACHERID" />

</many-to-one>

</class>

</hibernate-mapping>

Class指定对应实体类的全类名，Table指定实体类对应数据库表的表名，id指定实体和数据库中的主键，column对应数据库表的列，generator说明了主健的生成策略，property对应其他普通的列，many-to-one对应表与表的关联，如Course表与Teacher表关联。

# 第6章 系统实现

## 6.1 系统代码

（1）Spring管理的配置文件（部分）

<?xml version="1.0" encoding="UTF-8"?>

<beans

<bean id="loginAction" class="com.jmu.video.action.LoginAction" scope="prototype">

<property name="userService" ref="userService"></property>

</bean>

<bean id="userAction" class="com.jmu.video.action.UserAction" scope="prototype">

<property name="userService" ref="userService"></property>

</bean>

<!-- 配置业务层的类 -->

<bean id="userService" class="com.jmu.video.service.impl.UserServiceImpl">

<property name="userDao" ref="userDao"></property>

</bean>

<bean id="courseService" class="com.jmu.video.service.impl.CourseServiceImpl">

<property name="courseDao" ref="courseDao"></property>

</bean>

<!-- 配置dao的类 -->

<bean id="userDao" class="com.jmu.video.dao.UserDao">

<property name="sessionFactory" ref="sessionFactory"></property>

</bean>

<bean id="courseDao" class="com.jmu.video.dao.CourseDao">

<property name="sessionFactory" ref="sessionFactory"></property>

</bean>

<!-- 引入外部的属性文件 -->

<context:property-placeholder location="classpath:jdbc.properties" />

<!-- 配置连接池 -->

<bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource">

<property name="driverClass" value="${jdbc.driverClass}"></property>

<property name="jdbcUrl" value="${jdbc.url}"></property>

<property name="user" value="${jdbc.username}"></property>

<property name="password" value="${jdbc.password}"></property>

</bean>

<!-- 配置Hibernate的相关属性，需要spring-orm.jar包 -->

<bean id="sessionFactory"

class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">

<!-- 注入连接池 -->

<property name="dataSource" ref="dataSource"></property>

<!-- 配置Hibernate的属性,配置hibernate 的基本信息 -->

<property name="hibernateProperties">

<props>

<!-- hibernate 所使用的数据库方言 -->

<prop key="hibernate.dialect">org.hibernate.dialect.Oracle10gDialect</prop>

<!-- 执行操作时是否在控制台打印SQL -->

<prop key="hibernate.show\_sql">true</prop>

<!-- 是否对SQL 进行格式化 -->

<prop key="hibernate.format\_sql">true</prop>

<!-- 指定自动生成数据表的策略。update:数据库中有该表就使用，没表就创建 -->

<prop key="hibernate.hbm2ddl.auto">update</prop>

</props>

</property>

<!-- 加载Hibernate中的映射文件 -->

<property name="mappingResources">

<list>

<!-- .hbm.xml的全路径，包名用"/"隔开而不是用"." -->

<value>com/jmu/video/entity/User.hbm.xml</value>

</list>

</property>

</bean>

<!-- 配置事务管理器 -->

<bean id="transactionManager"

class="org.springframework.orm.hibernate4.HibernateTransactionManager">

<property name="sessionFactory" ref="sessionFactory"></property>

</bean>

<!-- 开启注解事务 -->

<tx:annotation-driven transaction-manager="transactionManager" />

<!-- 配置事务的传播特性 -->

<tx:advice id="txAdvice" transaction-manager="transactionManager">

<tx:attributes>

<tx:method name="add\*" propagation="REQUIRED" />

<tx:method name="del\*" propagation="REQUIRED" />

<tx:method name="modify\*" propagation="REQUIRED" />

<tx:method name="save\*" rollback-for="Exception" propagation="REQUIRED" />

<tx:method name="\*" read-only="false" />

</tx:attributes>

</tx:advice>

</beans>

（2）Struts2配置文件(部分)

<?xml version="1.0" encoding="UTF-8" ?>

<struts>

<!-- 添加上传文件错误提示的国际化资源 -->

<constant name="struts.custom.i18n.resources" value="app"></constant>

<!-- 设置项目允许上传最大的文件 -->

<constant name="struts.multipart.maxSize" value="900000000"/>

<package name="default" namespace="/" extends="struts-default">

<!-- 首页的action，加载所有课程 -->

<action name="index" class="indexAction" method="execute">

<result name="index">/WEB-INF/pages/index.jsp</result>

<result name="success">/WEB-INF/pages/index.jsp</result>

<result name="input">/WEB-INF/pages/index.jsp</result>

</action>

<!-- 文件下载 -->

<param name="inputPath">/annex</param>

<result name="success" type="stream">

<param name="contentType">application/ectet-stream</param>

<param name="inputName">targetFile</param>

<!-- 以附件的形式下载 -->

<param name="contentDisposition">attachment;filename="${fileName}"</param>

<param name="bufferSize">1024</param>

</result>

<result name="input">/WEB-INF/pages/register.jsp</result>

</action>

<action name="loginAction" class="loginAction" method="execute">

<result name="success" type="chain">index</result>

<result name="fail">/WEB-INF/pages/fail.jsp</result>

<result name="input">/WEB-INF/pages/register.jsp</result>

</action>

<action name="registerAction" class="registerAction" method="execute">

<result name="success" type="chain">index</result>

<result name="input">/WEB-INF/pages/register.jsp</result>

</action>

<action name="user\_\*" class="userAction" method="{1}">

<result name="student">/WEB-INF/pages/student.jsp</result>

<result name="teacher">/WEB-INF/pages/teacher.jsp</result>

<result name="admin">/WEB-INF/pages/admin.jsp</result>

<result name="success" type="chain">index</result>

<result name="updateTeacherSuccess">/WEB-INF/pages/teacher.jsp</result>

<result name="updateAdminSuccess">/WEB-INF/pages/admin.jsp</result>

<result name="updateStudentSuccess">/WEB-INF/pages/student.jsp</result>

<result name="input" type="chain">index</result>

</action>

<action name="course\_\*" class="courseAction" method="{1}">

<result name="success">/WEB-INF/pages/index.jsp</result>

<result name="input">/WEB-INF/pages/index.jsp</result>

<result name="courseScanSuccess">/WEB-INF/pages/courseScan.jsp</result>

<result name="classify">/WEB-INF/pages/classify.jsp</result>

<result name="videoPlay">/WEB-INF/pages/videoPlay.jsp</result>

<result name="input" type="chain">index</result>

</action>

<!-- 视频上传 -->

<action name="upload" class="courseAction" method="upload">

<interceptor-ref name="fileUpload">

<param name="allowedTypes">video/mp4,image/png,image/gif,image/jpeg</param>

<!-- 允许上传500M -->

<param name ="maximumSize">1024\*1024\*500</param>

</interceptor-ref>

<interceptor-ref name="defaultStack"></interceptor-ref>

<param name="savePath">/video</param>

<result name="uploadSuccess">/WEB-INF/pages/teacher.jsp</result>

<result name="uploadFail">/WEB-INF/pages/error.jsp</result>

<result name="input">/WEB-INF/pages/teacher.jsp</result>

</action>

<!-- 课件上传 -->

<action name="uploadAnnex" class="annexAction" method="uploadAnnex">

<interceptor-ref name="fileUpload">

<!-- <param name="allowedTypes">video/mp4</param> -->

<param name="allowedExtensions">

png,bmp,jpg,doc,xls,ppt,zip,rar

</param>

<!-- 允许上传500M -->

<param name ="maximumSize">1024\*1024\*500</param>

</interceptor-ref>

<interceptor-ref name="defaultStack"></interceptor-ref>

<param name="savePath">/annex</param>

<result name="uploadSuccess">/WEB-INF/pages/teacher.jsp</result>

<result name="uploadFail" type="chain">course\_findAllCourses</result>

<result name="input" type="chain">course\_findAllCourses</result>

</action>

<!-- 页面跳转 -->

<action name="\*">

<result>/WEB-INF/pages/{1}.jsp</result>

</action>

<!-- 验证码 -->

<action name="createImageAction" class="com.jmu.video.action.CreateImageAction"

method="execute">

<result name="success" type="stream">

<param name="contentType">image/jpeg</param>

<param name="inputName">inputStream</param>

</result>

</action>

</package>

</struts>

（3）课程Action的部分代码

ActionContext context = ActionContext.getContext();

Map<String, Object> session = context.getSession();

Map<String, Object> request = (Map<String, Object>) context.get("request");

Teacher teacher = (Teacher) session.get("teacher");

public String videoPlay(){

HttpServletRequest httpServletRequest = ServletActionContext.getRequest();

String videoId = httpServletRequest.getParameter("videoId");

Video v = courseService.findVideoById(new Integer(videoId));

request.put("video", v);

return "videoPlay";

}

/\*\*

\* 添加课程

\* @return

\*/

public String addCourse(){

findCourse = courseService.findCourseByName(course.getCourseName());

if(findCourse != null){

System.out.println("已存在该课程！");

return "addCourse";

}else {

int CourseId = courseService.addCourse(course, teacher);

}

return "addCourseSuccess";

}

/\*\*

\* 根据教师的ID查找该教师的所有课程

\* @return List<Course>

\*/

public String findAllCoursesByTeacherId(){

List<Course> courses = courseService.findAllCoursesByTeacherId(teacher.getTeacherId());

request.put("courses", courses);

return "findAllCoursesByTeacherIdSuccess";

}

## 6.2 系统截图

系统截图见图6-1到图6-7。



图6-1 教学平台首页图



图6-2 课程章节详情图



图6-3 课程评论图



图6-4 课程测试图



图6-5教师信息图



图6-6 教师所有课程图



图6-7 管理员管理教师课程图

# 结 论

刚开始设计这个系统的时候，因为花在设计上的时间过少而直接写代码，导致后期出来很多不合理的问题。表结构的不合理，不仅仅影响进度，更重要的是影响自己思考的思路。最后不得不重新从头开始重新思考重新设计，合理的设计带来的效益是比较高的。

在实践中发现和解决问题。本次毕业设计遇到很多问题，刚开始时主要集中在框架上，后期主要集中在业务逻辑上。对框架的不熟悉，所以前期花了较多的时间去学习框架和处理框架的问题；后期更多的问题是业务逻辑的问题，思考不够全面，这些问题耐心花点时间调度就慢慢可以解决。

及时沟通。遇到问题想一两天都想不通时，最好的办法是去和别人沟通。在设计留言表结构的时候，要考虑的到回复时表的设计。自己想了很久都没想通怎么设计更合理，最后请教一个同学，问题很快就解决了。从这也学会了遇到问题先自己思考，自己想不通的时候要及时沟通，不然不仅问题没有解决，也会使自己一直陷入痛苦当中。

这次的毕业设计，不仅是学会了怎么去设计实现一个系统，还学会了遇到问题如何去解决，如何去查资料，如何去安排项目进度才更合理。

# 致 谢

在毕业设计中，首先要感谢我的指导老师曾华朴老师，感谢老师对我的支持、帮助和信任，让我按时顺利完成毕业设计；感谢家人对我的关心和照顾，是他们一直在我身后默默地支持、鼓励着我，让我有勇气和能力面对遇到的一切困难；感谢我的辅导员洪宗华老师，她给了我很大的鼓励和建议；感谢周遵荣同学、张秋杰同学、谢焕鹏同学，他们在我遇到问题时给予了大量的建议和帮助；感谢我的舍友翁建洪同学，是他的行动让我在设计过程中不敢懈怠。在集美大学计算机工程学院学习的时候，受到学院很多老师的关心、指导、帮助。特别是刚从文科转专业到计算机工程学院的时候，自己对理科的东西什么都不懂，谢谢老师们对我的理解和帮助，在这里衷心地感谢所有的老师支持和包容。

感谢所有关心、帮助过我的老师、朋友和同学，谢谢大家！

# 参 考 文 献

[1] Cay S.Horstmann,Gary Cornell著. 程峰、黄若波、章恒翀译．Java2核心技术 卷I（第六版）[M]. 北京:机械工业出版社．2005

[2] Bruce Eckel著. 陈昊鹏译. Thinking in Java[M]. 北京: 机械工业出版社．2007

[3] 李刚．轻量级Java EE企业应用实战: Struts2+Spring4+Hibernate整合开发[M]．北京:电子工业出版社．2014

[4] Praveen Gupta,Prof. M.C. Govil. MVC Design Pattern for the multi framework distributed applications using XML, spring and struts framework[J]. International Journal on Computer Science and Engineering,2010,2(4)

[5] 陆舟．Struts2技术内幕 深入解析Struts架构设计与实现原理[M]．北京: 机械工业出版社．2012

[6] Craig Walls著．张卫滨译．Spring实战（第4版）[M]. 北京：人民邮电出版社. 2016

[7] Christian Bauer, Gaving King, Gary Gregory著．蒲成译．Hibernate实战(第2版)[M]. 北京：清华大学出版社. 2016

[8] 许晓斌．Maven实战[M]. 北京：机械工业出版社. 2011

[9] Bootstrap中文网. Bootstrap2中文文档（v2.3.2）[EB/OL]. <http://www.bootcss.com/>. 2012

[10] Apache Tomcat®. Apache Tomcat官网 [EB/OL]. http://tomcat.apache.org/. 2017.05

[11] Sergey Diev. Use cases modeling and software estimation[J]. ACM SIGSOFT Software Engineering Notes,2006,31(6)

[12] 鄂大伟．软件工程[M]. 北京：清华大学出版社. 2010

[13] Criig Larman著.李洋等译．UML和模式应用[M]. 北京：机械工业出版社. 2006

[14] 郑阿奇．SQL Server 2008版[M]. 北京：电子工业出版社.2009

[15] Wen Qi,Jie Song. Study on Textual Data Sources Based Real-Time ETL[J]. Advanced Materials Research,2013,2534(756)