

文章编号: 2096-1472(2017)-12-26-03

## 基于SSH框架的实践课程管理系统的设计与实现

杨佳绩, 孟艳红

(沈阳工业大学软件学院, 辽宁 沈阳 110023)

**摘要:** 本文针对各高校实践课程管理工作中存在的任务繁重、教学效果不佳、项目工程化培养程度不够等问题, 提出了一套实践课程管理系统的开发方案。该方案在对需求分析的基础上确定了系统所使用的开发框架, 设计了系统的数据结构模型, 并阐述了系统实现的功能。对如何实现课程报告的自动生成并打印提出了一种有效的解决途径, 并阐述了其利用Freemaker的实现过程。该系统操作方便, 有效地提高了高校实践课程的管理效率, 具有一定的使用价值。

**关键词:** 实践课程管理; SSH框架; Freemake模板引擎; 数据库

**中图分类号:** TP315 **文献标识码:** A

## Design and Implementation of the Practice Curriculum Management System Based on the SSH Framework

YANG Jiaji, MENG Yanhong

(School of Software, Shenyang University of Technology, Shenyang 110023, China)

**Abstract:** In view of the problems, such as heavy tasks, poor teaching effect and weak degree of project engineering training, this paper puts forward a development scheme of the practice curriculum management system. On the basis of requirement analysis, the scheme determines the system development framework, designs the system data structure model, and describes the system functions. The paper proposes effective solutions to the automatic generation and printing of curriculum reports, and expounds the implementation process via Freemake. The system is easy to operate and effectively improves the efficiency of college practical curriculum management, which is of great application value.

**Keywords:** practice curriculum management; SSH framework; Freemake template engine; database

### 1 引言(Introduction)

实践课程管理是各大学的主要日常管理工作之一, 涉及校、系、师、生的诸多方面。学生实践课程日常管理工作及保存管理日趋繁重、复杂。现存在的系统往往没有对学生的情况做更好的检测与评估, 没有很好地切合目前工程化培养的目标。无论是教师还是学生都迫切需要一套更加完善的管理系统, 以进行良好的实践课程管理。

对于教师来说, 随时了解学生实践课程的学习情况, 并对学生进行项目工程化培养是十分必要的。另一方面, 对于学生来说, 精准无误地掌握项目开发知识是学生在进行实践课程时所要完成的首要任务。基于以上存在的问题我们使用具有良好扩展性和耦合性的SSH框架对系统进行开发, 实现了学生、教师和管理员端的功能, 使用Maven工具与SSH框架结合为系统开发及维护提供了便利条件。采用Freemaker替换技术对word模板进行替换修改不仅解决了实验报告按模板自动生成并打印的难点, 也为后期图片的替换提供了一种合理方法。

### 2 系统开发框架及技术(System development framework and technology)

基于对系统需求上的深入分析, 其使用框架应该具有良好的可扩展性、解耦合性, 并且能够适应大规模用户同时在线。

系统使用Struts2-Spring-Hibernate三种框架整合开发, 利用Struts2将从页面接收到的请求发送给一个标准的过滤链, 包括(ActionContextCleanUp)过滤器。通过FilterDispatcher核心控制器调用ActionMapper配置集来确定请求哪个Action, 这种方式可以使系统更加健壮安全<sup>[1]</sup>。利用Spring去整合服务器对数据库的事务处理。对Action、Dao和数据库连接池进行配置<sup>[2]</sup>。利用Hibernate实现对表中数据的持久化处理, 利用映射技术, 直接将表转换成类, 并在类中编写对数据库进行处理的SQL语句, 方便系统对数据进行操作, 减少服务器连接数据库时产生的开销, 提高系统运行效率<sup>[3]</sup>。

采用Maven管理工具中横切的逻辑, 这些逻辑来自一组

共享的(或者自定义的)插件<sup>[4]</sup>。基于Maven的实践课程管理系统可以更加方便地在pom.xml中引入依赖,使得开发效率得以提高<sup>[5]</sup>。

系统使用三个常用的开发工具Easy-ui、Ztree和富文本编辑框Ueditor设计了用户界面,使用户在使用上更加方便。另外在对页面加载技术方面,系统采用了Ajax异步加载技术,使该系统不用重新加载整个页面便可实现部分刷新效果<sup>[6,7]</sup>。此外Struts2框架对Ajax提供支持,开发者只需要调用action中的getter方法就能获取到相应的值,使系统在对请求action的处理上更加灵活<sup>[8]</sup>。

### 3 系统需求分析和功能设计(System requirement analysis and function design)

#### 3.1 系统功能设计

系统的整体设计如下:

(1)用户登录设计:将用户按角色分为学生、教师和管理员。用户以用户名与密码进行登录。如果用户名或密码错误,给予提示。用户信息统一由管理员添加。用户不能自行注册。

##### (2)学生端功能设计

①上传实践报告:学生将编写好的实践报告通过系统上传界面上传给教师。

②填写实践报告:学生根据提供的实验报告模板进行填写,根据填写信息自动生成实验报告文档。

③签到:学生在系统中进行本次课程的签到,系统自行对签到时间及学生信息进行记录,教师通过系统查看学生的签到情况。

④在线测试:学生在系统中进行在线测试,测试题目由教师发布,学生提交的答案将由教师评阅并给出最终分数。

⑤成绩查询:学生通过系统对在线测试的成绩进行查询。

##### (3)教师端功能设计

①发布课程任务:根据相应科目发布新的实践课程任务,并可对发布的实践课程任务进行添加、查看、删除操作。

②发布在线测试:教师对学生端的测试内容进行修改及发布。

③成绩评判:教师可查看到学生对在线测试做出的答案,给出成绩的评判并打分,成绩可被学生查看。

④查看签到:教师可查询所有人的签到信息及签到时间,并可进行打印。

⑤下载报告:教师对学生上传的实践报告进行下载并可打印。

##### (4)管理员功能设计

管理员可通过正确录入老师信息(姓名、密码等)添加教师用户,其录入值不能为空。管理员可删除数据库中所有该老

师的关联信息。管理员可通过系统,对学生信息进行录入,学生无法登录也无法自行注册。管理员可对学生信息进行删除。系统功能结构图如图1所示。

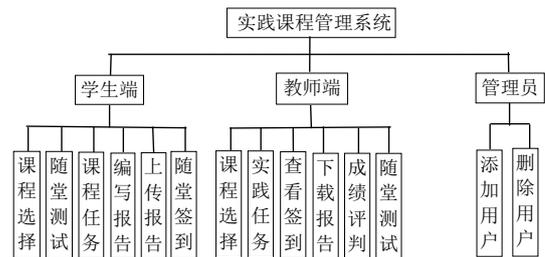


图1 系统功能结构图

Fig.1 Structure diagram of system functions

#### 3.2 系统数据库表结构

数据结构是系统功能实现的关键核心。数据结构是否符合业务要求,是否存在冗余直接影响到系统功能优化程度。系统对其功能需求进行深入分析,避免了数据冗余,并根据各实体间存在的联系建立了如下数据库表。

系统数据结构包括权限表(pow)、用户表(user)、功能表(able)、教授表(teach)、课程表(class)、题目表(title)、测试表(test)、任务表(task)、任务内容表(content)。

(1)用户表(user)其主要字段有账号、用户名、密码、权限身份,该表包含该系统三个角色所有用户信息数据。

(2)权限表(pow)其主要字段有权限、权限名称。权限表与用户表之间组成一对多的关系,实现不同用户不同权限划分。

(3)功能表(able)其主要字段有父节点、子节点、跳转页面、功能名称、权限。功能表与权限表组成一对一的关系,实现不同权限不同功能。登录时系统须根据权限确定用户功能,即每个用户对应一个权限,每个权限对应多个用户,每个权限对应多个功能,每个功能对应一个权限。

(4)教授表(teach)其主要字段有编号、用户账号、课程编号。

(5)课程表(class)其主要字段有课程编号、课程名称、用户表、教授表。课程表三表实现用户分班管理的结构。系统须根据班级管理用户信息,每个用户对应一个班级,教师教授多个班级,每个班级对应多个用户。

(6)题目表(title)其主要字段有编号、用户账号、课程名称、测试题目、测试时间、截止时间、发布人。

(7)测试表(test)其主要字段有题号、题目内容、a选项、b选项、c选项、d选项、试题编号。题目表与测试表之间存在一对多的关系,从而实现测试题成组出,测试功能测试题目对应多个测试题,每个测试题对应一个题目。

(8)任务表(task)其主要字段有编号、课程编号、任务号。

(9)任务内容表(content)其主要字段有任务编号、任务内容、题目编号。任务表与任务内容表存在一对多的关系,任

务编号作为连接两个表的码，从而实现成组管理。

数据库表之间的关系图如图2所示。

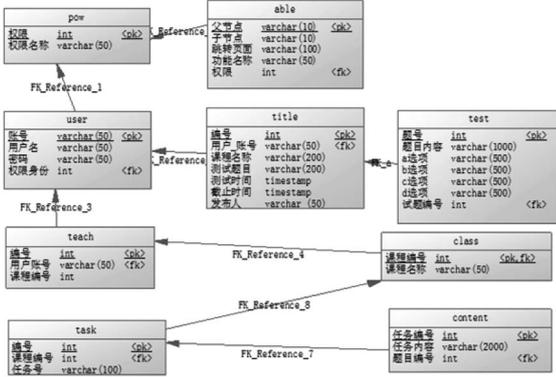


图2 数据库表之间的关系图

Fig.2 Diagram of the relationship among database tables

### 4 系统的实现(System implementation)

系统各功能运行时依次连接Action、Service、Dao三种类型的类实现数据通信，这样对于同一张表中的不同事务处理就需要每次先执行相同的连接操作。基于这种操作，设计BaseXXX抽象类，将重复代码放入抽象类中，进行自定义基础通用类DAO、Service为BaseDAO、BaseService，从而在运行时只需继承这些抽象类便可实现数据通信。同时对于不同表所执行的增删改查时，设计通用DAO组件，只编写一个DAO工具类就可完成对所有表的增删改查处理。通过反射和泛型技术使系统各模块功能都只需调用上述定义的函数就可实现。另外自定义拦截器LoginInterceptor，实现未登录用户不能访问Web-INF下所有页面和除登录之外的其他Action，并用拦截器传递错误信息。系统各功能模块设计包括如下：

#### (1)用户登录功能实现

登录时并不是跳转到不同界面，而是通过Ajax异步加载动态加载每种用户的功能部分，将数据库中根据权限存储好的功能表抽象成对象userDao，并通过userDAO.findById方法取回数据，再将数据压入值栈中以json输出，将得到的字符串赋值到利用zTree工具生成的树中，实现动态加载菜单的效果，减轻了系统接收请求的负担，增加了加载页面的效率。

#### (2)学生端功能实现

填写实践报告：学生根据提供的文本框及提示进行填写，最后提交到数据库中，并根据填写信息自动生成实验报告文档。学生填写的信息通过public void save保存到数据库中。

签到：将签到表抽象为对象userDAO.findById，对用户表的签到信息进行修改和存储。

在线测试：学生可在该界面对教师下发的任务进行回答，并由教师给出相关分数。该功能的实现与菜单功能基本类似，将测试题信息抽象为对象，再通过自定义函数对数据进行操作，将从数据库找回的数据存储为List，压入值栈，

转换为json字符串，然后在前端对字符串进行处理，从而实现在线测试题的显示。

成绩查询：通过数据库查询函数userDAO.findById，返回成绩信息并显示到前端页面上。

学生端运行界面如图3所示。



图3 学生端界面图

Fig.3 Student-End interface

#### (3)教师端功能实现

发布课程任务：该功能将填写的任务信息通过save方法存入数据库中。在学生端进行获取显示，实现发布任务的效果。

发布在线测试：教师可对学生端的测试内容进行修改及发布。与发布课程任务功能原理相同。

成绩评判：教师可查看到学生对在线测试做出的答案，给出成绩的评判并打分。将成绩直接存入用户表中对应id的成绩信息中。

查看签到：教师可看到所有人的签到信息及签到时间，将数据库中签到标记为“1”的信息取回屏显示，通过userDAO.findByNameQuery方法实现。

下载报告：教师可下载学生上传的实践报告。通过configuration.setClassForTemplateLoading方法加载文档模板，然后将存储的数据装载到定义出的map中，通过Freemaker模板引擎生成报告并下载。

教师端运行界面如图4所示。



图4 教师端界面图

Fig.4 Teacher-End interface

#### (4)实践报告模板模块实现

系统的开发难点在于如何实现Java与Word之间的模板式转换。Word是微软公司开发的字处理文件格式，以.doc或者.docx作为文件后缀名。使用Freemaeker模板引擎对已定义好的模板进行替换修改，从而实现实验报告的自动生成和下载上传。