



课程管理系统设计文档

131250181 陈云龙



2016-3-25

目录

更新历史.....1

1. 引言.....2

 1.1. 目的.....2

 1.2. 范围.....2

2. 架构设计.....2

3. 类的设计.....2

 3.1. 业务逻辑层分解.....2

 3.1.1. 业务逻辑层模块职责2

 3.1.2. 业务逻辑层模块接口规范3

4. 数据库设计.....6

更新历史

修改人员	日期	变更原因	版本号
陈云龙	2016-03-25	课程设计	V1.0

1.引言

1.1. 目的

本文档详细描述了课程管理系统的设计，达到引导开发的作用，同时实现测试人员以及用户的沟通。

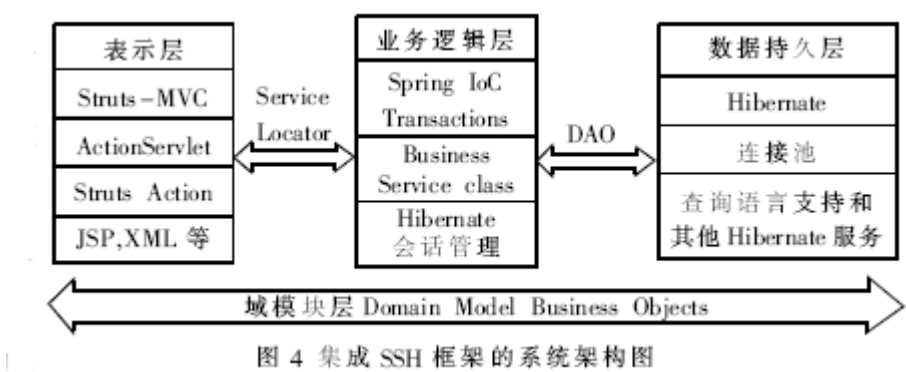
本文档面向开发人员，测试人员以及最终用户编写，是了解系统的导航。

1.2. 范围

课程管理系统是方便教学管理，有管理员对用户和课程的管理，学生可以方便的查找自己的课程，提交作业，查看课程作业。老师可以布置作业，安排助教批改作业，安排选课学生，安排课程助教。同时有教学管理人员对学生情况，课程情况进行统计。

2.架构设计

此项目采用 MVC 架构，使用 Spring+Struts+Hibernate 框架



3.类的设计

3.1. 业务逻辑层分解

3.1.1. 业务逻辑层模块职责

模块	职责
----	----

UserManage	负责用户模块所需要的服务
CourseManage	负责课程模块所需要的服务
AssignmentManage	负责作业模块所需要的服务

3.1.2. 业务逻辑层模块接口规范

3.1.2.1. userManage 模块

- 模块概述
userManage 模块负责用户的登陆注册，设置个人资料，获得全部用户列表，查找用户等功能。
- userManage 接口规范

提供的服务（供接口）		
userManage.validateUser	语法	public User validateUser(String userid, String password);
	前置条件	userid 和 password 不为空
	后置条件	验证用户名和密码是否匹配
userManage.registerUser	语法	public String registerUser(User user);
	前置条件	user 不为空
	后置条件	数据库中添加一条 User 用户
userManage.userList	语法	public List<User> userList();
	前置条件	无
	后置条件	返回全部用户列表
userManage.delete	语法	public void delete(User user);
	前置条件	user 不为空
	后置条件	删除此条 user 记录
userManage.update	语法	public void update(User user);
	前置条件	user 不为空
	后置条件	更新此条 user 记录
userManage.findUser	语法	public User findUser(String id);
	前置条件	id 不为空
	后置条件	返回此 id 的 user 记录

3.1.2.2. CourseManage 模块

➤ 模块概述

CourseManage 模块提供课程的添加删除修改，课程老师添加，添加课程助教，选课学生等功能

➤ CourseManage 接口规范

提供的服务（供接口）		
CourseManage. setCourse	语法	public String setCourse(Course course);
	前置条件	course 不为空
	后置条件	数据库中添加一条课程记录
CourseManage. courseList	语法	public List<Course> courseList();
	前置条件	无
	后置条件	返回全部课程的列表
CourseManage. delete	语法	public void delete(Course course);
	前置条件	course 不为空
	后置条件	删除数据库中的此课程数据
CourseManage. update	语法	public void update(Course course);
	前置条件	course 不为空
	后置条件	更新数据库中的此课程数据
CourseManage. findCourse	语法	public Course findCourse(String cid);
	前置条件	cid 不为空
	后置条件	返回符合条件 cid 的课程
CourseManage. addCourseAssistant	语法	public void addCourseAssistant(String courseid, String assid);
	前置条件	courseid 和 assid 分别有数据库中的数据, 且不能重复添加, assid 不能是选课学生
	后置条件	数据库中添加一条数据
CourseManage. findCourseAssistant	语法	public CourseAssistants findCourseAssistant(String assid, String courseid);
	前置条件	courseid 和 assid 分别有数据库中的数据
	后置条件	返回对应的猪脚信息
CourseManage. findAssistant	语法	public List<User> findAssistant(String cid);
	前置条件	cid 不为空
	后置条件	返回对应课程的全部猪脚
CourseManage. deleteAssistant	语法	public void deleteAssistant(String assid, String courseid);
	前置条件	courseid 和 assid 分别有数据库中的数据
	后置条件	删除数据库中的数据

CourseManage. addCourseStudent	语法	public void addCourseStudent(String courseid, String stuid);
	前置条件	courseid 和 stuid 分别有数据库中的数据, 且不能重复添加, 成为助教的学生不能选课
	后置条件	添加一条选课学生记录
CourseManage. findCourseStudent	语法	public CourseStudent findCourseStudent(String stuid, String courseid);
	前置条件	courseid 和 stuid 分别有数据库中的数据
	后置条件	返回一条选课学生记录
CourseManage. findStudent	语法	public List<User> findStudent(String cid);
	前置条件	cid 不能为空
	后置条件	返回对应课程的全部学生
CourseManage. deleteStudent	语法	public void deleteStudent(String uid, String courseid);
	前置条件	courseid 和 stuid 分别有数据库中的数据
	后置条件	删除数据库中的选课学生记录
CourseManage. myCourseList	语法	public List<Course> myCourseList(String userid);
	前置条件	userid 不为空
	后置条件	返回 userid 的全部课程

3.1.2.3. AssignmentManage 模块

➤ 模块概述

AssignmentManage 模块提供作业方面的服务, 包括老师布置作业, 修改作业, 删除作业, 学生提交作业等

➤ AssignmentManage 接口规范

提供的服务 (供接口)		
AssignmentManage. setAssignment	语法	public void setAssignment(Assignment assignment);
	前置条件	assignment 不为空
	后置条件	添加一条作业记录
AssignmentManage. findAssignment	语法	public List<Assignment> findAssignment(String cid);
	前置条件	cid 不为空
	后置条件	返回对应课程的全部作业
AssignmentManage. addStudentAssignment	语法	public void addStudentAssignment(StudentAssignment sa);
	前置条件	sa 不为空

	后置条件	添加一条学生提交作业记录
AssignmentManage. findStudentAssignment	语法	public StudentAssignment findStudentAssignment(String studentid, String assignmentid);
	前置条件	studentid, assignmentid 不为空, 且对应数据库中有数据
	后置条件	返回此条学生提交作业的记录
AssignmentManage. findAllStudentAssignment	语法	public List<StudentAssignment> findAllStudentAssignment(String cid);
	前置条件	cid 不为空
	后置条件	返回对应课程所有的学生提交的作业
AssignmentManage. updateStudentAssignment	语法	public void updateStudentAssignment(StudentAssignment sa);
	前置条件	sa 不为空
	后置条件	更新此条学生作业记录

4. 数据库设计

1. 老师布置作业表 assignment

ColumnName	DataType	PK	NotNull	Default
id	VARCHAR(255)	√	√	
correctDeadline	date			
deadline	date			
description	TEXT			NULL
filePath	VARCHAR(255)			NULL
format	VARCHAR(255)			NULL
score	INT			NULL
courseid	VARCHAR(255)			NULL
fileName	VARCHAR(255)			NULL

2. 课程表 course

ColumnName	DataType	PK	NotNull	Default
id	VARCHAR(255)	√	√	
description	TEXT			NULL
semester	date			NULL
instructor	VARCHAR(255)			NULL
cname	VARCHAR(255)			NULL
end	date			NULL

3. 课程助教表 courseassistants

ColumnName	DataType	PK	NotNull	Default
id	VARCHAR(255)	√	√	
assistantid	VARCHAR(255)			NULL
courseid	VARCHAR(255)			NULL

4. 选课学生表 coursestudent

ColumnName	DataType	PK	NotNull	Default
id	VARCHAR(255)	√	√	
courseid	VARCHAR(255)			NULL
studentid	VARCHAR(255)			NULL

5. 学生作业表 studentassignment

ColumnName	DataType	PK	NotNull	Default
id	VARCHAR(255)	√	√	
assignmentid	VARCHAR(255)			NULL
studentid	VARCHAR(255)			NULL
fileName	VARCHAR(255)			NULL

6. 用户表 user

ColumnName	DataType	PK	NotNull	Default
id	VARCHAR(255)	√	√	
iden	int			NULL
mail	VARCHAR(255)			NULL
name	VARCHAR(255)			NULL
password	VARCHAR(255)			NULL
phone	VARCHAR(255)			NULL
sex	int			NULL
userid	VARCHAR(255)			NULL
avatar	VARCHAR(255)			NULL