|  |  |
| --- | --- |
| **项目编号** |  |
| **文档编号** |  |
| **密级** |  |

**作业管理系统详细设计**

**V4.0**

**广西民族大学**

评审日期： 2018年7月6日

目录

[1．导言 7](#_Toc30353)

[1.1 目的 7](#_Toc21112)

[1.2 范围 7](#_Toc15485)

[1.3 引用标准 7](#_Toc4588)

[1.4 参考资料 7](#_Toc28950)

[1.5 版本更新信息 8](#_Toc10084)

[2．详细设计简述 8](#_Toc29320)

[2.1 设计简介 8](#_Toc19110)

[2.2 模块简介 8](#_Toc19106)

[3、界面详细设计 9](#_Toc21210)

[3.1、页面设计说明 9](#_Toc32284)

[3.2、静态页面详细设计 9](#_Toc23365)

[4、客户端模块详细设计 9](#_Toc26349)

[4.1、公用模块的详细设计 9](#_Toc23231)

[4.1.1 表现层 10](#_Toc31324)

[4.1.2 控制层 10](#_Toc11255)

[4.1.3 数据持久层 11](#_Toc8385)

[4.2、登录模块 14](#_Toc13508)

[4.2.1 表现层 14](#_Toc13316)

[4.2.2 控制层 15](#_Toc3350)

[4.2.3 业务逻辑层 17](#_Toc9054)

[4.2.4 数据持久层 18](#_Toc19004)

[4.2.5 域模型层 20](#_Toc28144)

[4.3、成绩查询模块 21](#_Toc31135)

[4.3.1 表现层 22](#_Toc29470)

[4.3.2 控制层 24](#_Toc28203)

[4.3.3 业务逻辑层 26](#_Toc20190)

[4.3.4 数据持久层 30](#_Toc24039)

[4.3.5 域模型层 32](#_Toc26965)

[4.4、个人信息模块 34](#_Toc23668)

[4. 4.1 表现层 35](#_Toc13961)

[4.4.2 控制层 37](#_Toc10107)

[4.4.3 业务逻辑层 37](#_Toc579)

[4.4.4 数据持久层 38](#_Toc10117)

[4.4.5 域模型层 39](#_Toc7075)

[4.5、作业查询模块 40](#_Toc17805)

[4.5.1 表现层 41](#_Toc24369)

[4.5.2 控制层 42](#_Toc3472)

[4.5.3 业务逻辑层 45](#_Toc3138)

[4.5.4 数据持久层 49](#_Toc16304)

[4.5.5 域模型层 49](#_Toc28680)

[4.6、发布作业模块 50](#_Toc13193)

[4.6.1 表现层 51](#_Toc3610)

[4.6.2 控制层 52](#_Toc23102)

[4.6.3 业务逻辑层 54](#_Toc7644)

[4.6.4 数据持久层 55](#_Toc30075)

[4.6.5 域模型层 56](#_Toc23080)

[4.7、批改作业模块 57](#_Toc12472)

[4.7.1 表现层 58](#_Toc2480)

[4.7.2 控制层 59](#_Toc18394)

[4.7.3 业务逻辑层 61](#_Toc14744)

[4.7.4 数据持久层 62](#_Toc27496)

[4.7.5 域模型层 62](#_Toc30858)

[4.8、提交作业模块 63](#_Toc610)

[4.8.1 表现层 64](#_Toc18007)

[4.8.2 控制层 65](#_Toc20716)

[4.8.3 业务逻辑层 67](#_Toc9486)

[4.8.4 数据持久层 70](#_Toc21011)

[4.8.5 域模型层 71](#_Toc21046)

[5、管理端模块详细设计 72](#_Toc29160)

[5.1、删除用户模块详细设计 72](#_Toc20849)

[5.1.1 表现层 73](#_Toc23082)

[5.1.2 控制层 74](#_Toc26843)

[5.1.3 业务逻辑层 75](#_Toc18368)

[5.1.4 数据持久层 76](#_Toc9479)

[5.1.5 域模型层 77](#_Toc14523)

# 1．导言

## 1.1 目的

该文档是对作业管理系统概要设计进行的详细说明，是为了指导和规范作业管理系统开发而制定的详细开发设计文档

本文档的预期读者包括：

* 设计开发人员
* 项目管理人员
* 测试人员
* 用户

## 1.2 范围

该文档内容涵盖作业管理系统系统的所有功能模块的详细设计描述

## 1.3 引用标准

[1] 《软件工程案例教程 第2版》 韩万江等 机械工业出版社

## 1.4 参考资料

[1] 《软件项目管理案例教程 第3版》 韩万江等 机械工业出版社

## 1.5 版本更新信息

本文档的更新记录如下表所示。

表1-1 文档更新记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 修改编号 | 修改日期 | 修改后版本 | 修改位置 | 修改内容概述 |
| 001 | 2018.06.27 | 1.0 | 全部 | 初始发布版本 |
| 002 | 2018.06.29 | 2.0 | 操作流程 | 增加 |
| 003 | 2018.07.02 | 3.0 | 序列图 | 修改 |
| 004 | 2018.07.06 | 4.0 | 伪代码补充 | 修改 |

# 4、客户端模块详细设计

客户端模块主要包括登录、成绩查询个人信息、作业查询、发布作业、批改作业提交作业模块。

## 4.1、公用模块的详细设计

### 4.1.1 表现层

公用模块的表现层主要是一些页面出错时的跳转页面，显示当前错误的结果处理报告。具体见表

表 4-1

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 登录出错页面 | loginError.jsp | 登入时出现严重错误时跳转到该页面。 |
| 统一异常页面 | exceptionPage.jsp | 在系统运行时出现未知错误时跳转到该页面。 |

### 4.1.2 控制层

**public** class ActionSupport implements Action, Validateable, ValidationAware, TextProvider, LocaleProvider, Serializable {

**protected** static Logger LOG = LoggerFactory.getLogger(ActionSupport.class);

**private** **final** ValidationAwareSupport validationAware = new ValidationAwareSupport();

**private** transient TextProvider textProvider;

**private** Container container;

**public** **void** setActionErrors(Collection<String> errorMessages) {  }

**public** Collection<**String**> getActionErrors() {  }

**public** void setActionMessages(Collection<String> messages) {  }

**public** Collection<**String**> getActionMessages() {  }

}

### 4.1.3 数据持久层

IBaseDAO.java代码结构：

**package** com.buptsse.spm.dao;

**import** java.io.Serializable;

**import** java.util.List;

/\*\*

\* 基础数据库操作类

\* @author hysd

\*/

**public interface** IBaseDAO<T> {

**public** Serializable save(T o); /\*\*保存一个对象\*/

**public void** delete(T o); /\*\* 删除一个对象\*/

**public void** update(T o); /\*更新一个对象\*/

**public void** saveOrUpdate(T o); /\*保存或更新对象\*/

**public List<T>** find(**String** hql); /\*查询\*/

**public List<T>** find(**String** hql, T[] param); /\*查询集合\*/

**public List<T>** find(**String** hql, **List<T>** param); /\*查询集合\*/

**public List<T>** find(**String** hql, **T[]** param, **Integer** page, **Integer** rows); /\*查询集合

\*/

p**ublic T** get(Class<T> c, **Serializable** id); /\*获得一个对象\*/

**public Integer** executeHql(**String** hql); /\*执行HQL语句\*/

**public Integer** executeHql(**String** hql, **List<T>** param); /\*执行HQL语句\*/

}

BaseDAOImpl.java对IBaseDAO.java实现：

**package** com.buptsse.spm.dao.impl;

/\*\*

\* 基础数据库操作类

\* @author hysd

\*/

*@Repository("baseDAO")*

*@SuppressWarnings("all")*

**public** class BaseDAOImpl<T> implements IBaseDAO<T> {

**private** SessionFactory*sessionFactory***;**

**public** Serializable save(T o); /\*\*保存一个对象\*/

**public** void delete(T o); /\*\* 删除一个对象\*/

**public** void update(T o); /\*更新一个对象\*/

**public** void saveOrUpdate(T o); /\*保存或更新对象\*/

**public** List<T> find(String hql); /\*查询\*/

**public** List<T> find(String hql, T[] param); /\*查询集合\*/

**public** List<T> find(String hql, List<T> param); /\*查询集合\*/

**public** List<T> find(String hql, T[] param, Integer page, Integer rows); /\*查询集合

\*/

**public** T get(Class<T> c, Serializable id); /\*获得一个对象\*/

**public** Integer executeHql(String hql); /\*执行HQL语句\*/

**public** Integer executeHql(String hql, List<T> param); /\*执行HQL语句\*/

}

## 4.2、登录模块

在登录模块时，系统内部的相应响应操作示意图如图4-2所示。



图 4-2 系统响应示意图

### 4.2.1 表现层

登录模块的表现层主要完成不同用户的登录功能，登录页面时要求用户输入账号密码的基本信息，确认后页面给出响应消息，提示登录成功或失败的提示。表现层对应的JSP页面列表见表4-3所示。

表 4-3 登入模块表现层JSP列表

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 登录页面 | login.jsp | 用户（教师、学生、管理员）登入功能，当登入出错时给出提示。 |

login.jsp的流程图如图4-3所示。

进入游首页

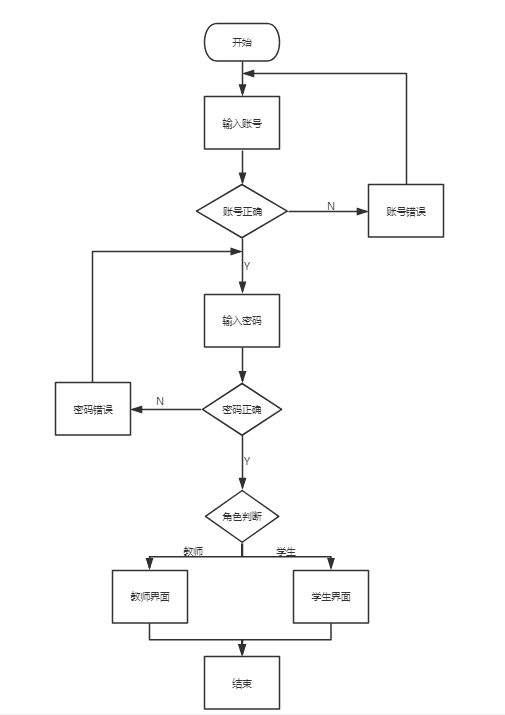


图 4-3 登录模块详细设计流程图

### 4.2.2 控制层

登录模块的控制层负责接受来自login.jsp的用户输入，同时调用登入模块的业务逻辑接口，将用户名与密码等用户关键信息传递到业务逻辑层进行判定。等到业务逻辑处理完成之后，将来自业务逻辑层的相应信息传到表现层，并决定显示页面。登入模块控制层列表见表4-4所示。

表 4-4 登入模块控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 登录 | LoginAction.java | SUCCESS | mainFrame.jsp—登入成功，显示提示窗口。 |
| ERROR | login.jsp---登入失败，显示提示窗口。 |

在控制层中LoginAction.java的描述如下所示：

**package** com.buptsse.spm.action;

/\*\*

\* *@description* 实现登录页面Action

\*/

**public** **class** LoginAction **extends** ActionSupport

{

**private** User user;

*@Resource*

**private** IUserService userService;

/\*实现各种属性的setter和getter方法\*/

/\*实现登入模块的控制层方法\*/

**public** String login(){}

}

### 4.2.3 业务逻辑层

登录模块的业务逻辑层主要完成对用户登录逻辑的判定，同时调用登录模块的业务逻辑接口。比如：用户到登录是输入的ID是否存在、密码是否正确，同时在对用户的身份进行判定。登录模块业务逻辑层列表如图4-5所示。

表 4-5登入模块业务逻辑层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 登录 | IUserService.java  UserServiceImpl.java | 调用IUserDao | 返回给Action |

在登入模块的业务逻辑层是调用了公用的IUserService接口，同时在实现该接口。

IUserService.java接口主要方法：

**package** com.buptsse.spm.service;

**public** **interface** IUserService {

/\*登入检测\*/

**public** User login(String userName,String password);

/\*查找用户\*/

**public** User findUser(String userName);

/\*删除用户\*/

**public** boolean deleteUser(String username);

/\*增加用户\*/

**public** boolean addUser(User user);

}

UserServiceImpl.java实现了IUserService.java接口， 同时需要调用数据持久层的UserDao.Java，利用@Resource来实现对数据持久层接口的调用。

UserServiceImpl.java主要实现属性与方法:

**package** com.buptsse.spm.service.impl;

*@Transactional*

*@Service*

**public** **class** UserServiceImpl **implements** IUserService {

*@Resource*

**private** IUserDao iUserDao;

*@Override*

**public** User login(String userName,String password);

*@Override*

**public** User findUser(String userName);

*@Override*

**public** boolean deleteUser(String username);

*@Override*

**public** boolean addUser(User user);

}

### 4.2.4 数据持久层

登录模块的是数据持久层能对用户数据进行增删改查。登入模块数据持久层列表见表4-6所示。

表 4-6 登入模块数据持久层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 登录 | IUserDao.java  UserDaoImpl.java | 调用 user.Java | 对用户信息进行增删改查操作 |

IUserDao.java定义了用户信息进行增、删、改、查的接口：

package com.buptsse.spm.dao;

/\*\*

\* @description 用户信息持久层接口定义，包括用户增加，保存，查询

\*/

**public** **interface** IUserDao {

**public** User findUser(User user); /\*查找用户信息\*/

**public** boolean insertUser(User user); /\*插入用户信息\*/

**public** List<User> searchUser(List<Object> choose); /\*查找用户信息\*/

**public** boolean deleteUser(String username); /\*删除用户信息\*/

**public** boolean addUser(User user); /\*增加用户信息\*/

}

UserDaoImpl.java是IUserDao.java接口的实现：

package com.buptsse.spm.dao;

/\*\*

\* @description 用户信息持久层接口定义，包括用户增加，保存，查询

\*/

@Repository

**public** **class** UserDaoImpl extends BaseDAOImpl implements IUserDao {

**public** User findUser(User user); /\*查找用户信息\*/

**public** boolean insertUser(User user); /\*插入用户信息\*/

**public** List<User> searchUser(List<Object> choose); /\*查找用户信息\*/

**public** boolean deleteUser(String username); /\*删除用户信息\*/

**public** boolean addUser(User user); /\*增加用户信息\*/

}

### 4.2.5 域模型层

登录模块用到域模型层中User.java，User.java是一个公用域模型，在涉及到用户信息查询等操作时，就会调用到该模型，登录模块域模型层列表如表4-7所示。

表 4-7登入模块域模型层列表

|  |  |
| --- | --- |
| 域模型 | 描述 |
| User.java | 对用户信息的增、删、改、查操作 |

User.java主要属性与方法：

**package** com.buptsse.spm.domain;

/\*\*

\* *@description* User表的信息记录

\*/

**public** **class** User **implements** Serializable{

**private** **String** id; /\*id识别\*/

**private** **String** userName; /\*用户名\*/

**private** **String** password; /\*密码 \*/

**private** **String** password1; /\*验证密码\*/

**private** **String** position; /\*身份信息 \*/

**private** **String** userId; /\*用户ID\*/

**private** **int** videoTime; /\*用户学习时间 \*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

## 4.3、成绩查询模块

在成绩查询模块时，系统内部的相应响应操作示意图如图4-2所示。



### 4.3.1 表现层

成绩查询模块的表现层主要完成用户查询学生成绩功能，成绩查询页面时要求用户输入学生学号，确认后页面给出响应消息，查询成功或失败的提示。表现层对应的JSP页面列表见表4-8所示。教师成绩查询页面要求用户登录成功，确认后页面给出响应消息，查询成功或失败的提示。表现层对应的JSP页面列表见表4-8所示。

表 4-8 成绩查询模块表现层JSP列表

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 查询成绩 | LookScore.JSP | 用户（教师、学生）登入功能，当查询出错时给出提示。 |
| 作业列表 | teaTaskCorrectList.jsp | 教师登录功能，进入到作业列表页面。 |
| 作业成绩列表 | teaSingleTaskList.jsp | 教师点击单份作业可查看到本次作业学生的成绩情况。 |

成绩查询模块的流程图如图4-9所示。

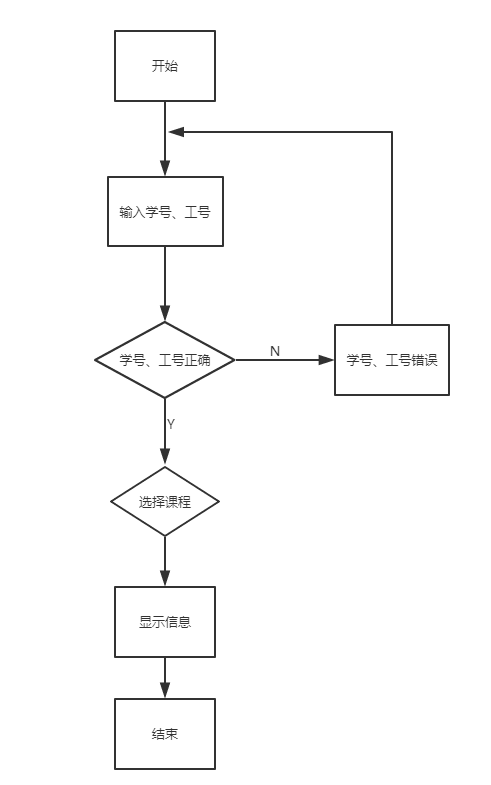


图 4-9 成绩查询模块详细设计流程图图

### 4.3.2 控制层

成绩查询模块的控制层负责接受来自 的用户输入，同时调用登入模块的业务逻辑接口，将学号传递到业务逻辑层进行判定。等到业务逻辑处理完成之后，将来自业务逻辑层的相应信息传到表现层，并决定显示页面。登入模块控制层列表见表4-10所示。

表 4-10成绩查询模块控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 查询 | LookScoreAction.java | SUCCESS | mainFrame.jsp—查询成功，显示提示窗口。 |
| ERROR | login.jsp---查询失败，显示提示窗口。 |
| 查询 | TeataskAction | SUCCESS | teaSingleTaskList.jsp—查询成功，显示提示窗口。 |
| ERROR | main.jsp---查询失败，显示提示窗口。 |

在控制层中LookScoreAction.java 的描述如下所示：

**package action;**

**import java.util.List;**

**import pojos.Stutask;**

**import pojos.User;**

**import service.StudentService;**

**import service.StutaskService;**

**public class LookScoreAction {**

**List<Stutask> stutask;**

**StutaskService stutaskService;**

**String userId;**

**List<User> users;**

**StudentService studentService;**

**public String execute(){**

**stutask=stutaskService.findListByStuId(userId);**

**users=studentService.findByUserId(userId);**

**return "Score";**

**}**

在控制层中TeataskAction的描述如下所示：

public class TeataskAction extends BaseAction implements Preparable,ModelDriven {

private Teatask teatask;

@Autowired

private ITeataskService teataskService;

@Autowired

private IStutaskService stutaskService;

private Result<Teatask> pageResult;

private Result<Stutask> pageResult2;

public String openCorrectList();//打开列表，一个老师发布的所有作业的情况

public String openSingleList();//打开列表，同一份作业，所有上交的作业，，可查看学生成绩情况

### 4.3.3 业务逻辑层

成绩查询模块的业务逻辑层主要完成对用户查询逻辑的判定，同时调用查询成绩模块的业务逻辑接口。比如：用户输入的ID是否存在，同时在对用户的身份进行判定。成绩查询模块业务逻辑层列表如图4-11所示。

表 4-11成绩查询模块业务逻辑层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 查询 | IStutaskService.java  StutaskServiceImpl.java  ITeataskService.java  TeataskServiceImpl.java | 调用StutaskDAO  &&TeataskDAO | 返回给StutaskAction  和TeataskAction |

在成绩查询模块的业务逻辑层是调用了公用的ITeataskServic和IStutaskService，同时在实现该接口。

ITeataskServic.Java接口主要方法：

IStutaskServic.Java接口主要方法：

public interface ITeataskService {

/\*\*

\* 根据property的value查询

\* @param property

\* @param value

\* @return SysCount

\*/

public Teatask findByProperty(String property,Object value);

/\*\*

\* @return\_type: Result<TeaTask>

\* @file\_name: 分页查询用户列表

\*/

public Result<Teatask> find(Teatask teaTask,String teacherId,int page, int row);

}

IStutaskServic.Java接口主要方法：

public interface IStutaskService {

/\*\*

\* @return\_type: Result<TeaTask>

\* @file\_name: 分页查询同一班级用户列表

\*/

public Result<Stutask> findByClasses(int teataskId,int page, int row);

}

TeataskServiceImpl.java实现了ITeataskService.java接口， 同时需要调用数据持久层的TeataskDao.Java，利用@Service来实现对数据持久层接口的调用。

TeataskServiceImpl.java主要实现属性与方法:

@Service("teataskService")

public class TeataskServiceImpl implements ITeataskService{

@Autowired

private TeataskDAO teataskDAO;

@Override

public Teatask findByProperty(String property, Object value) ;

/\*\*

\* 分页,一个老师的所有发布作业

\*/

@Override

public Result<Teatask> find(Teatask teatask,String teacherId, int page, int row) ;

}

StutaskServiceImpl.java实现了IStutaskService.java接口， 同时需要调用数据持久层的StutaskDao.Java，利用@Service来实现对数据持久层接口的调用。

StutaskServiceImpl.java主要实现属性与方法:

@Service("stutaskService")

public class StutaskServiceImpl implements IStutaskService{

@Autowired

private StutaskDAO stutaskDAO;

/\*\*

\* 分页,同一班级

\*/

@Override

public Result<Stutask> findByClasses(int teataskId, int page, int row)；

}

### 4.3.4 数据持久层

成绩查询模块的是数据持久层能对作业数据进行查询。成绩查询模块数据持久层列表见表4-12所示。

表 4-12 成绩查询模块数据持久层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 查询成绩 | StutaskDAO.java  TeataskDAO.java | 调用Stutask.Java和Teatask.java | 对作业数据进行查询操作 |

StutaskDAO..java定义了对作业数据进行查询的接口：

@Repository("stutaskDAO")

public class StutaskDAO extends BaseDaoImpl<Stutask> {

/\*\*

\* 分页,同一班级的学生作业

\* @return\_type: Result<SysMarjors>

\*/

@SuppressWarnings("unchecked")

public Result<Stutask> findByClasses(int teataskId, int page, int size)；

TeataskDAO..java定义了对作业数据进行查询的接口：

@Repository("teataskDAO")

public class TeataskDAO extends BaseDaoImpl<Teatask> {

/\*\*

\* 分页,单个老师发布的全部作业

\* @return\_type: Result<SysMarjors>

\*/

@SuppressWarnings("unchecked")

public Result<Teatask> find(Teatask teaTask,String teacherId, int page, int size)；

/\*\*

\*

\* 根据字段值查询 (实体)

\*/

@SuppressWarnings("unchecked")

public Teatask findByProperty(String property, Object value) ；

### 4.3.5 域模型层

成绩查询模块用到域模型层中StuTask.java，StuTask.java是一个公用域模型，在涉及到学生作业信息查询等操作时，就会调用到该模型，登录模块域模型层列表如表4-13所示。

表 4-13成绩查询模块域模型层列表

|  |  |
| --- | --- |
| 域模型 | 描述 |
| StuTask.java  Teatask.java | 对学生作业信息的查询操作 |

StuTask.java主要属性与方法：

**package Stu;**

**import java.sql.Timestamp;**

**public class StuTask {**

**int stutask\_id;**

**String student\_id;**

**Timestamp sub\_time;**

**String grade;**

**int is\_doing;**

**int teatask\_id;**

**String content\_answer;**

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

Teatask.java主要属性与方法：

public class Teatask implements java.io.Serializable {

// Fields

private Integer teataskId;

private String teacherId;

private Integer taskNum;

private Integer doingNum;

private Timestamp startTime;

private Timestamp finishTime;

private String isFinish;

private String collegeId;

private String classesId;

private String courseId;

private String contentTheme;

private String contentQuestion;

}

## 4.4、个人信息模块

个人信息模块的系统响应如图4-14所示。

图 4-14 个人信息系统响应示意图

### 4.1 表现层

个人信息模块的表现层主要完成用户查看或修改个人信息，个人信息页面修改密码时要求用户输入初始密码，修改密码以及再次输入修改密码，确认后页面给出响应消息，修改成功或输入错误提示。表现层对应的JSP页面列表见表4-15所示。

表 4-15个人信息模块表现层JSP列表

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 个人信息 | Personal.jsp  teaInfo.jsp  teaInfoUpdate.jsp | 用户（教师、学生）查询和修改个人信息功能，当输入错误时给出提示。 |

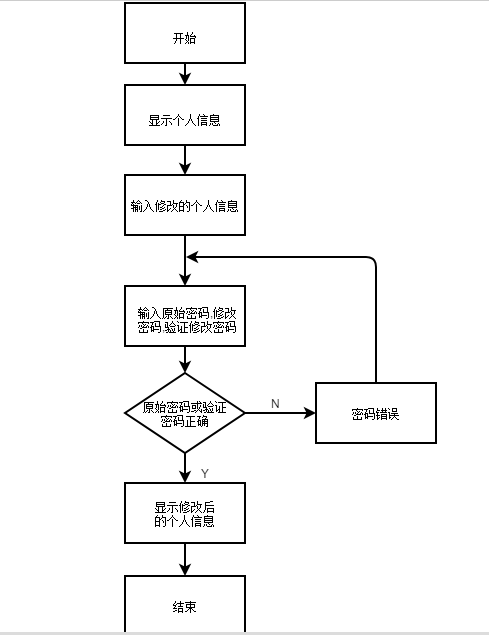
个人信息模块的流程图如图4-16所示。

图 4-16 个人信息模块详细设计流程图图

### 4.4.2 控制层

个人信息模块的控制层负责接受来自 的用户输入，同时调用个人查询的业务逻辑接口，将密码传递到业务逻辑层进行判定。等到业务逻辑处理完成之后，将来自业务逻辑层的相应信息传到表现层，并决定显示页面。个人信息模块控制层列表见表4-17所示。

表 4-17个人信息模块控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 修改 | PersonalAction.java  UserAction.java | SUCCESS | teaInfoUpdate.jsp—查询成功，显示提示窗口。 |
| ERROR | error.jsp---查询失败，显示提示窗口。 |

在控制层中 PersonalAction.java 的描述如下所示：

public class UserAction extends BaseAction implements Preparable,ModelDriven {

private User user;

@Autowired

private ILoginService userService;

@SuppressWarnings({ "rawtypes", "unchecked" })

public String update();

### 4.4.3 业务逻辑层

个人信息模块的业务逻辑层主要完成对用户信息修改逻辑的判定，同时调用个人信息模块的业务逻辑接口。比如：用户输入的密码是否正确，同时在对用户的身份进行判定。个人信息模块业务逻辑层列表如图4-18所示。

表 4-18个人信息模块业务逻辑层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 查询 | IUserService.java  UserServiceImpl.java | 调用IUserDao | 返回给Action |

在个人信息模块的业务逻辑层是调用了公用的IUserService接口，同时在实现该接口。

IUserService.java接口主要方法：

public interface IUserService {

public void update (User user);

}

}

UserServiceImpl.java实现了IUserService.java接口， 同时需要调用数据持久层的UserDao.Java，利用@ Service来实现对数据持久层接口的调用。

UserServiceImpl.java主要实现属性与方法:

@Service("userService")

public class UserServiceImpl implements IUserService {

@Autowired

private UserDAO userDAO;

public void update(User user)

### 4.4.4 数据持久层

个人信息模块的是数据持久层能对用户信息进行查询,修改。个人信息模块数据持久层列表见表4-19所示。

表 4-19 个人信息模块数据持久层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 查询,修改个人信息 | UserDAO.java | 调用 User.Java | 对个人信息进行查询或修改操作 |

.java定义了对个人信息进行查询或修改的接口：

@Repository("userDAO")

public class UserDAO extends BaseDaoImpl<User>{

@SuppressWarnings("unchecked")

public User findByProperty(String property, String value); //查询

//修改，调用hql封装的update（）方法

}

### 4.4.5 域模型层

个人信息模块用到域模型层中User.java是一个公用域模型，在涉及到用户信息查询,修改等操作时，就会调用到该模型，登录模块域模型层列表如表4-20所示。

表 4-20个人信息模块域模型层列表

|  |  |
| --- | --- |
| 域模型 | 描述 |
| User.java | 对个人信息的查询,修改操作 |

User.java主要属性与方法：

**package** com.buptsse.spm.domain;

/\*\*

\* *@description* User表的信息记录

\*/

**public** **class** User **implements** Serializable{

**private** **String** id; /\*id识别\*/

**private** **String** userName; /\*用户名\*/

**private** **String** password; /\*密码 \*/

**private** **String** password1; /\*验证密码\*/

**private** **String** position; /\*身份信息 \*/

**private** **String** userId; /\*用户ID\*/

**private** **int** videoTime; /\*用户学习时间 \*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

## 4.5、作业查询模块

作业查询模块的系统响应如图4-21所示。



图 4-21 作业查询系统响应示意图

### 4.5.1 表现层

作业查询模块的表现层主要完成用户查看作业信息，作业查询页面查询作业时要求用户输入课程，班级，确认后页面给出响应消息，修改成功或输入错误提示。表现层对应的JSP页面列表见表4-22所示。

表 4-22作业查询模块表现层JSP列表

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 作业查询 | StutaskQuery.JSP | 用户（教师、学生）查询作业信息功能，当输入错误时给出提示。 |

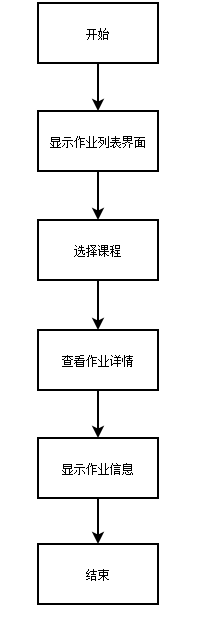


图 4-24 作业查询模块详细设计流程图图

### 4.5.2 控制层

作业查询模块的控制层负责接受来自浏览器端的用户输入，同时调用作业查询的业务逻辑接口，将密码传递到业务逻辑层进行判定。等到业务逻辑处理完成之后，将来自业务逻辑层的相应信息传到表现层，并决定显示页面。作业查询模块控制层列表见表4-24所示。

表 4-24 作业查询模块控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 查询作业 | StutaskQueryAction.java | SUCCESS | .jsp—查询成功，显示提示窗口。 |
| ERROR | .jsp---查询失败，显示提示窗口。 |

在控制层中 StutaskQueryAction.java 的描述如下所示：

**package action;**

**import java.util.List;**

**import pojos.Stutask;**

**import pojos.User;**

**import service.StudentService;**

**import service.StutaskService;**

**public class StutaskQueryAction {**

**List<Stutask> stutask;**

**String studentId;**

**StutaskService stutaskService;**

**List<User> users;**

**StudentService studentService;**

**public String execute(){**

**stutask=stutaskService.findListByStuId(studentId);**

**users=studentService.findByUserId(studentId);**

**return "stutasklist";**

/\*编写对上述各个属性的*getter*与*setter*方法\*/

**}**

### 4.5.3 业务逻辑层

作业查询模块的业务逻辑层主要完成对用户信息修改逻辑的判定，同时调用成绩查询模块的业务逻辑接口。比如：用户输入的课程或教师是否正确。作业查询模块业务逻辑层列表如图4-25所示。

表 4-25作业查询模块业务逻辑层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 作业查询 | IStutaskService.java  StutaskServiceImpl.java | 调用 | 返回给Action |

在作业查询模块的业务逻辑层是调用了公用的 接口，同时在实现该接口。接口主要方法：

**@Service("stutaskService")**

**public class StutaskServiceImpl implements IStutaskService{**

**@Autowired**

**private StutaskDAO stutaskDAO;**

**@Override**

**public Stutask findByProperty(String property, int value) {**

**return stutaskDAO.findByProperty(property, value);**

**}**

**/\*\***

**\* 分页**

**\*/**

**@Override**

**public Result<Stutask> find(Stutask stutask,String studentId, int page, int row) {**

**return stutaskDAO.find(stutask,studentId, page, row);**

**}**

**/\*\***

**\* 分页,同一班级**

**\*/**

**@Override**

**public Result<Stutask> findByClasses(int teataskId, int page, int row) {**

**return stutaskDAO.findByClasses(teataskId, page, row);**

**}**

**/\*\***

**\* add**

**\*/**

**@Override**

**public Stutask add(Stutask stutask) {**

**System.out.println(stutask.toString());**

**return stutaskDAO.save(stutask);**

**}**

**@Override**

**public void delete(int id) {**

**stutaskDAO.remove(Stutask.class,id);**

**}**

**@Override**

**public void update(Stutask stuTask) {**

**}**

**@Override**

**public void updatetablebyproperty(String table,String field1,String field2,String property,int value,String fieldvalue1,String fieldvalue2) {**

**stutaskDAO.updatetablebyproperty(table, field1,field2, property, value, fieldvalue1,fieldvalue2);**

**}**

**@Override**

**public List<Stutask> findListByProperty(String property, String value) {**

**// TODO Auto-generated method stub**

**return null;**

**}**

**public List<Stutask> findListById(){**

**return stutaskDAO.findListById();**

**}**

**@Override**

**public List<Stutask> findAllList() {**

**return stutaskDAO.getAll(Stutask.class);**

**}**

**}**

### 4.5.4 数据持久层

作业查询模块的是数据持久层能对用户信息进行查询,修改。成绩查询模块数据持久层列表见表4-26所示。

表 4-26作业查询模块数据持久层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 查询作业信息 | stutaskDAO.java | 调用stutask .Java | 对作业信息进行查询操作 |

.java定义了对作业信息进行查询的接口：

@Repository("stutaskDAO")

public class StutaskDAO extends BaseDaoImpl<Stutask> {

@SuppressWarnings("unchecked")

public void updatetablebyproperty(String table,String field1,String field2,String property,int value,String fieldvalue1,String fieldvalue2) ;

}

### 4.5.5 域模型层

作业查询模块用到域模型层中StuTask.java，StuTask.java是一个公用域模型，在涉及到作业信息查询操作时，就会调用到该模型，作业查询模块域模型层列表如表4-27所示。

表 4-27作业查询模块域模型层列表

|  |  |
| --- | --- |
| 域模型 | 描述 |
| StuTask.java | 对作业查询的查询操作 |

StuTask.java主要属性与方法：

**package Stu;**

**import java.sql.Timestamp;**

**public class StuTask {**

**int stutask\_id;**

**String student\_id;**

**Timestamp sub\_time;**

**String grade;**

**int is\_doing;**

**int teatask\_id;**

**String content\_answer;**

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

## 4.6、发布作业模块

发布作业模块的系统响应图如图4-28所示。



图 4-29 发布作业系统响应图

### 4.6.1 表现层

发布作业模块的表现层主要完成用户查看作业信息，作业查询页面查询作业时要求用户输入作业信息，确认后页面给出响应消息，发布成功或输入错误提示。表现层对应的JSP页面列表见表4-29所示。

表 4-29发布作业模块表现层JSP列表

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 发布作业 | teaTaskAdd.Jsp | 教师发布作业信息功能，当发布失效时给出提示。 |

### 发布作业模块的流程图如图4-30所示。

图 4-30 发布作业模块详细设计流程图图

### 4.6.2 控制层

发布作业模块的控制层负责接受来自浏览器端的用户输入，同时调用发布作业的业务逻辑接口，将作业内容传递到业务逻辑层进行判定。等到业务逻辑处理完成之后，将来自业务逻辑层的相应信息传到表现层，并决定显示页面。发布作业模块控制层列表见表4-31所示。

表 4-31 发布作业模块控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 发布作业 | TeataskAction.java | SUCCESS | teaTaskList.jsp—发布成功，显示提示窗口。 |
| ERROR | error.jsp---发布失败，显示提示窗口。 |

在控制层中TeataskAction的描述如下所示：

public class TeataskAction extends BaseAction implements Preparable,ModelDriven {

private Teatask teatask;

@Autowired

private ITeataskService teataskService;

private Stutask stutask;

@Autowired

private IStutaskService stutaskService;

private Result<Teatask> pageResult;

private Result<Stutask> pageResult2;

private List<Teatask> teataskAllList;

@Autowired

private ICollegeService collegeService;

List<College> collegeList;

List<Classes> classesList;

List<Course> courseList;

public String openAdd();//打开发布作业界面

public String add();//发布作业

}

### 4.6.3 业务逻辑层

发布作业模块的业务逻辑层主要完成对作业信息修改逻辑的判定，同时调用发布作业模块的业务逻辑接口。比如：用户输入作业内容并选择发布班级。发布作业模块业务逻辑层列表如图4-32所示。

表 4-32发布作业模块业务逻辑层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 发布作业 | ITeataskService.java  TeataskServiceImpl.java | 调用TeataskDAO | 返回给Action |

在发布作业模块的业务逻辑层是调用了公用的ITeataskService接口，同时在实现该接口。接口主要方法：

public interface ITeataskService {

public Teatask add(Teatask teaTask);

}

TeataskServiceImpl.java实现了ITeataskService.java接口， 同时需要调用数据持久层的TeataskDAO.Java，利用@Service来实现对数据持久层接口的调用。

TeataskServiceImpl.java主要实现属性与方法:

@Service("teataskService")

public class TeataskServiceImpl implements ITeataskService{

public Teatask add(Teatask teatask) ;

}

### 4.6.4 数据持久层

发布作业模块的是数据持久层能对作业内容进行增加。发布作业模块数据持久层列表见表4-33所示。

表 4-33发布作业模块数据持久层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 发布作业 | TeataskDAO.java | 调用Teatask .Java | 对作业内用进行增加操作 |

TeataskDAO.java定义了对作业内用进行增加询的接口：

@Repository("teataskDAO")

public class TeataskDAO extends BaseDaoImpl<Teatask> {

//调用hql的save()方法

}

### 4.6.5 域模型层

发布作业模块用到域模型层中Teatask.java是一个公用域模型，在涉及到作业内容的添加操作时，就会调用到该模型，发布作业模块域模型层列表如表4-34所示。

表 4-34发布作业模块域模型层列表

|  |  |
| --- | --- |
| 域模型 | 描述 |
| Teatask.java | 对内容进行添加操作 |

Teatask.java主要属性与方法：

public class Teatask implements java.io.Serializable {

private Integer teataskId;

private String teacherId;

private Integer taskNum;

private Integer doingNum;

private Timestamp startTime;

private Timestamp finishTime;

private String isFinish;

private String collegeId;

private String classesId;

private String courseId;

private String contentTheme;

private String contentQuestion;

}

## 4.7、批改作业模块

批改作业模块的系统响应图如图4-35所示。



图 4-35 批改作业系统响应图

### 4.7.1 表现层

批改作业模块的表现层主要完成用户修改学生作业信息，批改作业页面批改作业时要求用户输入学生作业成绩，确认后页面给出响应消息，输入成功或输入错误提示。表现层对应的JSP页面列表见表4-36所示。

表 4-36批改作业模块表现层JSP列表

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 批改作业 | stuTaskAddLater.jsp | 教师批改作业功能，当输入成绩无效给出提示。 |

### 批改作业模块的流程图如图4-37所示。

图 4-37 批改作业模块详细设计流程图图

### 4.7.2 控制层

批改作业模块的控制层负责接受来自浏览器端的用户输入，同时调用批改作业的业务逻辑接口，将作业成绩传递到业务逻辑层进行判定。等到业务逻辑处理完成之后，将来自业务逻辑层的相应信息传到表现层，并决定显示页面。批改作业模块控制层列表见表4-38所示。

表 4-38 发布作业模块控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 批改作业 | StutaskAction.Java | SUCCESS | teaTaskCorrectList.jsp—批改成功，显示提示窗口。 |
| ERROR | error.jsp---批改失败，显示提示窗口。 |

在控制层中StutaskAction的描述如下所示：

public class StutaskAction extends BaseAction implements Preparable,ModelDriven {

private Stutask stutask;

@Autowired

private IStutaskService stutaskService;

private Teatask teatask;

@Autowired

private ITeataskService teataskService;

private Result<Stutask> pageResult;

private Result<Teatask> pageResult2;

private List<Stutask> stutaskAllList;

private String collegeId;

private String classesId;

private Map<Integer,Teatask> teataskMap = new HashMap<Integer, Teatask>();

public String update();

}

### 4.7.3 业务逻辑层

批改作业模块的业务逻辑层主要完成对学生作业信息修改逻辑的判定，同时调用批改作业模块的业务逻辑接口。比如：用户选择学生作业并添加作业成绩。批改作业模块业务逻辑层列表如图4-39所示。

表 4-39批改作业模块业务逻辑层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 批改作业 | IStutaskService.java  StutaskServiceImpl.java | 调用Stutask | 返回给Action |

在批改作业模块的业务逻辑层是调用了公用的IStutaskService接口，同时在实现该接口。接口主要方法：

public interface IStutaskService {

public void updatetablebyproperty(String table,String field1, String field2, String property, int value, String fieldvalue1,String fieldvalue2);

}

StutaskServiceImpl.java实现了IStutaskService.java接口， 同时需要调用数据持久层的StutaskDao.Java，利用@Resource来实现对数据持久层接口的调用。

StutaskServiceImpl.java主要实现属性与方法:

@Service("stutaskService")

public class StutaskServiceImpl implements IStutaskService{

@Autowired

private StutaskDAO stutaskDAO;

public void updatetablebyproperty(String table,String field1,String field2,String property,int value,String fieldvalue1,String fieldvalue2);

}

### 4.7.4 数据持久层

批改作业模块的是数据持久层能对作业内容进行增加。批改作业模块数据持久层列表见表4-40所示。

表 4-40批改作业模块数据持久层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 批改作业 | StutaskDAO.java | 调用 stutask.Java | 对学生作业信息进行修改操作 |

StutaskDAO.java定义了对学生作业信息进行修改的接口：

@Repository("stutaskDAO")

public class StutaskDAO extends BaseDaoImpl<Stutask> {

@SuppressWarnings("unchecked")

public void updatetablebyproperty(String table,String field1,String field2,String property,int value,String fieldvalue1,String fieldvalue2) ;

}

### 4.7.5 域模型层

批改作业模块用到域模型层中StuTask.java，StuTask.java是一个公用域模型，在涉及到作业内容的添加修改等操作时，就会调用到该模型，批改作业模块域模型层列表如表4-41所示。

表 4-41批改作业模块域模型层列表

|  |  |
| --- | --- |
| 域模型 | 描述 |
| StuTask.java | 对学生作业内容进行添加,修改操作 |

StuTask.java主要属性与方法：

**package Stu;**

**import java.sql.Timestamp;**

**public class StuTask {**

**int stutask\_id;**

**String student\_id;**

**Timestamp sub\_time;**

**String grade;**

**int is\_doing;**

**int teatask\_id;**

**String content\_answer;**

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

## 4.8、提交作业模块

提交作业模块的系统响应图如图4-42所示。



图 4-42提交作业系统响应图

### 4.8.1 表现层

提交作业模块的表现层主要完成用户添加学生作业信息，提交作业页面提交作业时要求用户提交作业文件，确认后页面给出响应消息，上传成功或上传错误提示。表现层对应的JSP页面列表见表4-43所示。

表 4-43提交作业模块表现层JSP列表

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 提交作业 | SubmitHomeWork.JSP | 学生提交作业功能，当上传作业无效给出提示。 |

### 提交作业模块的流程图如图4-44所示。

图 4-44 提交作业模块详细设计流程图图

### 4.8.2 控制层

提交作业模块的控制层负责接受来自 的用户输入，同时调用提交作业的业务逻辑接口，将作业内容传递到业务逻辑层进行判定。等到业务逻辑处理完成之后，将来自业务逻辑层的相应信息传到表现层，并决定显示页面。提交作业模块控制层列表见表4-45所示。

表 4-45 提交作业模块控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 提交作业 | SubmitHomeWorkAction.java | SUCCESS | .jsp—提交成功，显示提示窗口。 |
| ERROR | .jsp---提交失败，显示提示窗口。 |

在控制层中 SubmitHomeWorkAction.java 的描述如下所示：

**package action;**

**import pojos.User;**

**import service.StutaskService;**

**public class SubmitHomeWorkAction {**

**String contentAnswer;**

**StutaskService stutaskService;**

**String stutaskId;**

**public String execute(){**

**stutaskService.UpDataTask(contentAnswer,stutaskId);**

**return "Submit";**

/\*编写对上述各个属性的*getter*与*setter*方法\*/

**}**

### 4.8.3 业务逻辑层

提交作业模块的业务逻辑层主要完成对学生作业信息修改逻辑的判定，同时调用提交作业模块的业务逻辑接口。比如：用户选择作业并完成或者上传作业文件。提交作业模块业务逻辑层列表如图4-46所示。

表 4-46提交作业模块业务逻辑层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 提交作业 | IStutaskService.java  StutaskServiceImpl.java | 调用 | 返回给Action |

在提交作业模块的业务逻辑层是调用了公用的 接口，同时在实现该接口。接口主要方法：

**@Service("stutaskService")**

**public class StutaskServiceImpl implements IStutaskService{**

**@Autowired**

**private StutaskDAO stutaskDAO;**

**@Override**

**public Stutask findByProperty(String property, int value) {**

**return stutaskDAO.findByProperty(property, value);**

**}**

**/\*\***

**\* 分页**

**\*/**

**@Override**

**public Result<Stutask> find(Stutask stutask,String studentId, int page, int row) {**

**return stutaskDAO.find(stutask,studentId, page, row);**

**}**

**/\*\***

**\* 分页,同一班级**

**\*/**

**@Override**

**public Result<Stutask> findByClasses(int teataskId, int page, int row) {**

**return stutaskDAO.findByClasses(teataskId, page, row);**

**}**

**/\*\***

**\* add**

**\*/**

**@Override**

**public Stutask add(Stutask stutask) {**

**System.out.println(stutask.toString());**

**return stutaskDAO.save(stutask);**

**}**

**@Override**

**public void delete(int id) {**

**stutaskDAO.remove(Stutask.class,id);**

**}**

**@Override**

**public void update(Stutask stuTask) {**

**}**

**@Override**

**public void updatetablebyproperty(String table,String field1,String field2,String property,int value,String fieldvalue1,String fieldvalue2) {**

**stutaskDAO.updatetablebyproperty(table, field1,field2, property, value, fieldvalue1,fieldvalue2);**

**}**

**@Override**

**public List<Stutask> findListByProperty(String property, String value) {**

**// TODO Auto-generated method stub**

**return null;**

**}**

**public List<Stutask> findListById(){**

**return stutaskDAO.findListById();**

**}**

**@Override**

**public List<Stutask> findAllList() {**

**return stutaskDAO.getAll(Stutask.class);**

**}**

**}**

### 4.8.4 数据持久层

提交作业模块的是数据持久层能对作业内容进行增加。提交作业模块数据持久层列表见表4-47所示。

表 4-47提交作业模块数据持久层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 提交作业 | StutaskDAO.java | 调用Stutask .Java | 对学生作业信息进行增加操作 |

StutaskDAO.java定义了对学生作业信息进行增加的接口：

@Repository("stutaskDAO")

public class StutaskDAO extends BaseDaoImpl<Stutask> {

@SuppressWarnings("unchecked")

public void updatetablebyproperty(String table,String field1,String field2,String property,int value,String fieldvalue1,String fieldvalue2) ;

}

### 4.8.5 域模型层

提交作业模块用到域模型层中StuTask.java，StuTask.java是一个公用域模型，在涉及到作业内容的添加操作时，就会调用到该模型，提交作业模块域模型层列表如表4-48所示。

表 4-48提交作业模块域模型层列表

|  |  |
| --- | --- |
| 域模型 | 描述 |
| StuTask.java | 对学生作业内容进行添加操作 |

StuTask.java主要属性与方法：

**package Stu;**

**import java.sql.Timestamp;**

**public class StuTask {**

**int stutask\_id;**

**String student\_id;**

**Timestamp sub\_time;**

**String grade;**

**int is\_doing;**

**int teatask\_id;**

**String content\_answer;**

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

# 5、管理端模块详细设计

## 5.1、删除用户模块详细设计

提交作业模块的系统响应图如图5-1所示。

图 5-1 删除用户系统响应图

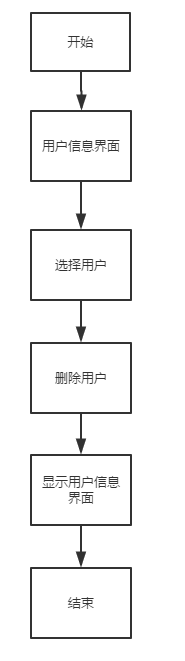
### 5.1.1 表现层

删除用户模块的表现层主要完成管理员删除用户信息，删除用户页面删除用户时要求管理员选择用户，确认后页面给出响应消息，删除成功或删除错误提示。表现层对应的JSP页面列表见表5-2所示。

表 5-2删除用户模块表现层JSP列表

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 删除用户 | Deleteuser.jsp | 管理员删除用户功能，当用户不存在给出提示。 |

### 删除用户模块的流程图如图5-3所示。



### 5.1.2 控制层

删除用户模块的控制层负责接受来自 的用户输入，同时调用删除用户的业务逻辑接口，将用户信息传递到业务逻辑层进行判定。等到业务逻辑处理完成之后，将来自业务逻辑层的相应信息传到表现层，并决定显示页面。删除用户模块控制层列表见表5-4所示。

表 5-4 删除用户模块控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 删除用户 | deleteuserAction.java | SUCCESS | userList.jsp—删除成功，显示提示窗口。 |
| ERROR | error.jsp---删除失败，显示提示窗口。 |

在控制层中Action的描述如下所示：

### 5.1.3 业务逻辑层

删除用户模块的业务逻辑层主要完成对用户信息删除逻辑的判定，同时调用删除用户模块的业务逻辑接口。比如：管理员选择用户并删除。删除用户模块业务逻辑层列表如图5-5所示。

表 5-5删除用户模块业务逻辑层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 删除用户 | IUserService.java  UserServiceImpl.java | 调用UserDAO | 返回给Action |

在删除用户模块的业务逻辑层是调用了公用的IUserService接口，同时在实现该接口。接口主要方法：

public interface IUserService {

public void delete(int id);

}

UserServiceImpl.java实现了IUserService.java接口， 同时需要调用数据持久层的UserDao.Java，利用@Resource来实现对数据持久层接口的调用。

UserServiceImpl.java主要实现属性与方法:

@Service("userService")

public class UserServiceImpl implements IUserService {

@Autowired

private UserDAO userDAO;

public void delete(int id);

}

### 5.1.4 数据持久层

删除用户模块的是数据持久层能对用户信息进行删除。删除用户模块数据持久层列表见表5-6所示。

表 5-6删除用户模块数据持久层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 删除用户 | UserDAO.java | 调用 user.Java | 对用户信息进行删除操作 |

UserDAO.java定义了对用户信息进行删除的接口：

@Repository("userDAO")

public class UserDAO extends BaseDaoImpl<User>{

//调用hql的delete（）方法

}

### 5.1.5 域模型层

删除用户模块用到域模型层中.java，.java是一个公用域模型，在涉及到用户信息的删除操作时，就会调用到该模型，删除用户模块域模型层列表如表4-48所示。

表 4-48删除用户模块域模型层列表

|  |  |
| --- | --- |
| 域模型 | 描述 |
| user.java | 对用户信息进行删除操作 |

user.java主要属性与方法：

public class User implements java.io.Serializable {

// Fields

private Integer id;

private String userId;

private String userName;

private String userPass;

private String userSex;

private String collegeId;

private String classesId;

private Integer userRole;

}