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

**作业批改系统详细设计**

**V1.0**

**广西民族大学**

评审日期： 2016年2月28日

目录

[1．导言 4](#_Toc444248412)

[1.1 目的 4](#_Toc444248413)

[1.2 范围 4](#_Toc444248414)

[1.3 引用标准 5](#_Toc444248415)

[1.4 参考资料 5](#_Toc444248416)

[1.5 版本更新信息 5](#_Toc444248417)

[2．详细设计简述 6](#_Toc444248418)

[2.1 设计简介 6](#_Toc444248419)

[2.2 模块简介 6](#_Toc444248420)

[3、界面详细设计 7](#_Toc444248421)

[3.1、页面设计说明 7](#_Toc444248422)

[3.2、静态页面详细设计 7](#_Toc444248423)

[4、客户端模块详细设计 7](#_Toc444248424)

[4.1、公用模块的详细设计 7](#_Toc444248425)

[4.1.1 表现层 8](#_Toc444248426)

[4.1.2 控制层 8](#_Toc444248427)

[4.1.3 数据持久层 9](#_Toc444248428)

[4.2、登录模块 12](#_Toc444248429)

[4.2.1 表现层 12](#_Toc444248430)

[4.2.2 控制层 14](#_Toc444248431)

[4.2.3 业务逻辑层 15](#_Toc444248432)

[4.2.4 数据持久层 17](#_Toc444248433)

[4.2.5 域模型层 19](#_Toc444248434)

[4.n、XXXX模块 20](#_Toc444248435)

[4.n.1 表现层 20](#_Toc444248436)

[4.n.2 控制层 21](#_Toc444248437)

[4.n.3 业务逻辑层 21](#_Toc444248438)

[4.n.4 数据持久层 21](#_Toc444248439)

[4.n.5 域模型层 21](#_Toc444248440)

[5、管理端模块详细设计 21](#_Toc444248441)

[5.1、XXXXX模块详细设计 21](#_Toc444248442)

[5.1.1 表现层 21](#_Toc444248443)

[5.1.2 控制层 21](#_Toc444248444)

[5.1.3 业务逻辑层 21](#_Toc444248445)

[5.1.4 数据持久层 22](#_Toc444248446)

[5.1.5 域模型层 22](#_Toc444248447)

[5.n、XXXXX模块详细设计 22](#_Toc444248448)

[5.n.1 表现层 22](#_Toc444248449)

[5.n.2 控制层 22](#_Toc444248450)

[5.n.3 业务逻辑层 23](#_Toc444248451)

[5.n.4 数据持久层 23](#_Toc444248452)

[5.n.5 域模型层 23](#_Toc444248453)

[6、公共部分模块详细设计 23](#_Toc444248454)

[6.1 公共页面 23](#_Toc444248455)

[6.2 安全模块的详细设计 23](#_Toc444248456)

[6.n XXXXX 23](#_Toc444248457)

# 1．导言

## 1.1 目的

该文档是对作业批改系统概要设计进行的详细说明，是为了指导和规范作业批改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批改系统系统的所有功能模块的详细设计描述，根据本系统的《概要设计》进行详细设计。

## 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 | 2019.6.28 | 1.0 | 全部 | 初始发布版本 |
| 002 | 2019.7.2 | 2.0 | 4、5章节 | 增加 |
| 003 | 2015.4.16 | 2.0 | 4章节 | 修改 |
| 004 | 2015.4.23 | 3.0 | 4、6章节 | 修改 |
| 005 | 2015.4.25 | 4.0 | 6、7章节 | 修改、增加 |
| 006 | 2015.4.28 | 5.0 | 5、7章节 | 修改、增加 |
| 007 | 2015.6.3 | 6.0 | 4、6章节 | 修改 |
| 008 | 2015.6.8 | 7.0 | 5、7章节 | 修改 |
| 009 | 2015.12.17 | 8．0 | 全部 | 修改，增加 |

# 2．详细设计简述

## 2.1 设计简介

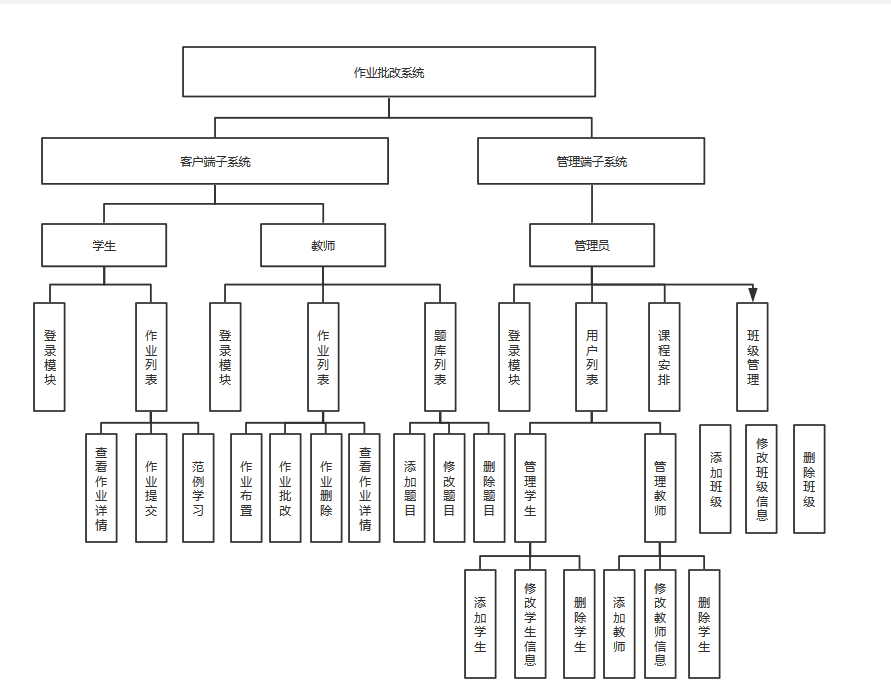
。。。。。。。。。分层结构图如图2-1所示。

图 2-1 XXXX示意图

。。。。。。。。。

## 2.2 模块简介

系统功能模块图如图？-？所示。



[对某个具体模块的详细设计可以采用流程图或者伪代码形式]

# 3、界面详细设计

[界面详细设计 包括界面设计的技术，以及静态页面设计]

## 3.1、页面设计说明

。。。。。。。。

## 3.2、静态页面详细设计

作业批改系统静态页面见表？-？

表3-1 页面表现层设计

|  |  |  |
| --- | --- | --- |
| 界面 | Vue | 功能描述 |
| 学生、教师登录界面 | login.vue | 提供给学生、教师输入账号、密码并登录 |
| 管理员登录界面 | adminLogin.vue | 提供给管理员输入账号、密码并登录 |
| 管理员教师信息管理界面 | TeacherInfo.vue | 提供管理员教师管理功能 |
| 管理员学生信息管理界面 | StudentInfo.vue | 提供管理员学生管理功能 |
| 管理员班级信息管理界面 | ClassInfo.vue | 提供管理员班级管理功能 |
| 管理员课程安排界面 | Course.vue | 提供管理员课程安排功能 |
| 学生作业主界面 | StudentIndex.vue | 显示学生作业列表，点击查看作业详情、查看作业答案 |
| 学生作业详细界面 | WorkDetail.vue | 提供给学生显示作业详情（完整题目、答案） |
| 教师作业列表界面 | TeacherWork.vue | 提供给教师显示发布作业列表，删除作业、已发布作业详情查看功能 |
| 教师发布作业界面 | workPublish.vue | 提供给教师从题库选择题目、自定义题目并发布 |
| 教师作业详情界面 | WorkDetailList.vue | 提供给教师查看已发布作业学生提交情况、查看分数统计、批改学生作业、 |
| 教师题库管理界面 | BankManagement.vue | 提供给教师查看题库题目、删除题库题目、添加题库题目 |
| 修改密码界面 | passwordModify | 提供给教师、学生修改密码 |

。。。。。。。。。。

# 4、客户端模块详细设计

客户端模块主要包括公用模块，登录模块，作业列表模块，作业提交模块，范例学习模块，作业布置模块，作业批改模块，作业删除模块 ，查看作业模块，添加题目模块，修改题目模块，删除题目模块。

## 4.1、公用模块的详细设计

[公用模块是指在系统在运行过程中将会被其他模块调用的基础资源]。

### 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 公用模块表现层

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 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

[公用模块的控制层中需要用到的是各种控制器的父类，例如ActionSupport.java。此类中实现了很多的实用接口，提供了很多默认方法，这些默认方法包括国际化信息的方法、默认的处理用户请求的方法等，这样可以大大的简化Acion的开发。ActionSupport.java的部分代码框架(由于该类是struts包中提供，且属性与方法较多，这里只选取部分展示说明)]

**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\*/

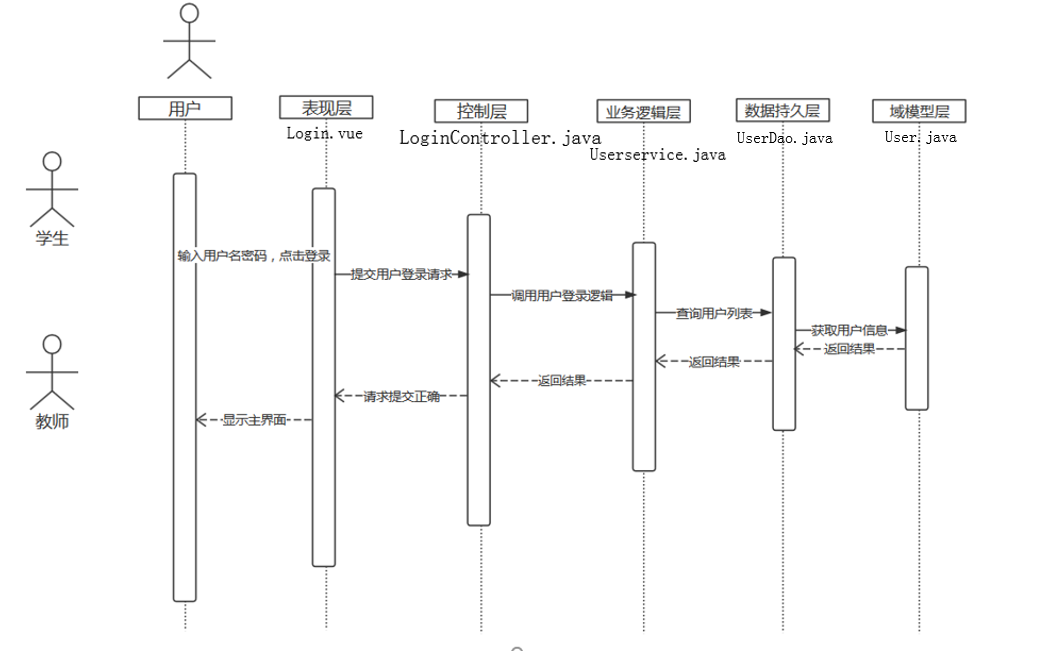
**public** Integer executeHql(String hql); /\*执行HQL语句\*/

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

}

## 4.2、学生、教师登录模块

在学生、教师登录模块时，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vue页面列表见表?-?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.vue | 用户（教师、学生）登入功能，当登入出错时给出提示。 |

login.vue的流程图如图4-3所示。

进入游首页

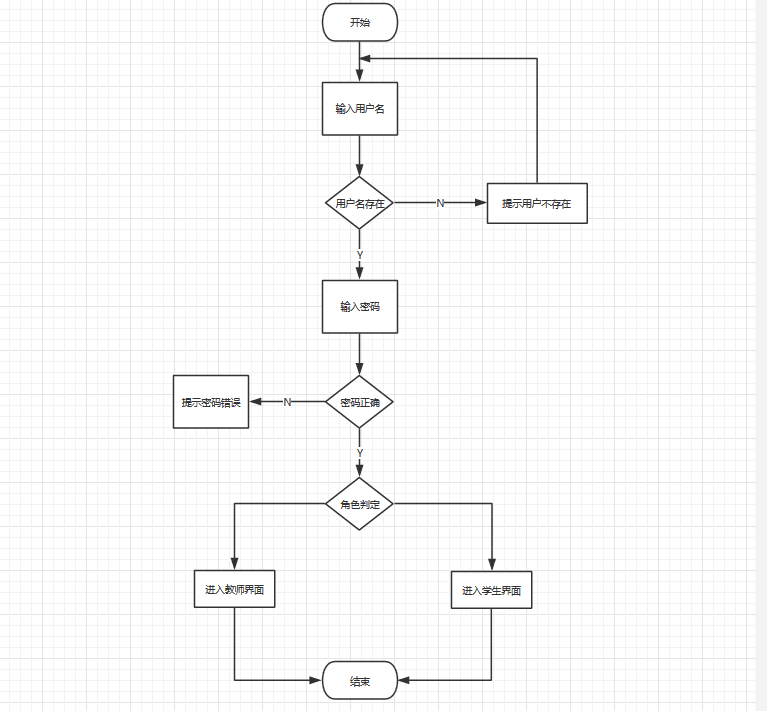


图 4-3 登录模块详细设计流程图

### 4.2.2 控制层

学生、教师登录模块的控制层负责接受来自login.vue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 | LoginController.java | SUCCESS | StudentHome.vue或TeacherHome.vue—登入成功，显示提示窗口，并跳转到相应的身份界面。 |
| ERROR | Login.vue---登入失败，显示提示窗口。 |

在控制层中LoginController.java的描述如下所示：

package com.example.workpigai.controller;

/\*\*

\* *@description* 实现登录页面Action

\*/

**public** **class** LoginController

{

UserService userService;

/\*实现登入模块的控制层方法\*/

**public Result login(@RequestBody User requestUser) {}**

}

### 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用户名是否存在、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 | UserService.service | 调用UserDao | 返回给Controller |

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

UserService.java主要方法：

package com.example.workpigai.service;

**public class UserService {**

**UserDao userDao;**

/\*登入检测\*/

**public boolean isExist(String account) {}**

/\*查找用户\*/

**public User getByAccount(String account) {}**

/\*查找密码\*/

**public User get(String account, String password){}**

/\*增加用户\*/

**public void add(User user) { }**

/删除用户\*/

**public void deleteById(int id) { }**

}

UserService.java调用数据持久层的UserDao.Java，利用实例化对象来实现对数据持久层接口的调用。

### 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 | UserDao.java | 调用 user.Java | 对用户信息进行增删改查操作 |

UserDao.java定义了用户信息进行增、删、改、查的接口：

package com.example.workpigai.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** UserDao JpaRepository<User,Integer>{

/\*查找用户信息\*/

**User findByAccount(String account);**

/\*匹配用户信息\*/

**User getByAccountAndPassword(String account,String password);** **public** boolean addUser(User user);

}

### 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.example.workpigai.model;**

/\*\*

\* *@description* User表的信息记录

\*/

**public** **class** User {

**private** **String** id; /\*id识别\*/

**String account;** /\*用户名\*/

**String password;** /\*密码\*/

**int type;** /\*身份\*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

## 4.3、学生作业列表模块

学生登录成功后，系统内部的相应响应操作示意图如图4-2所示。

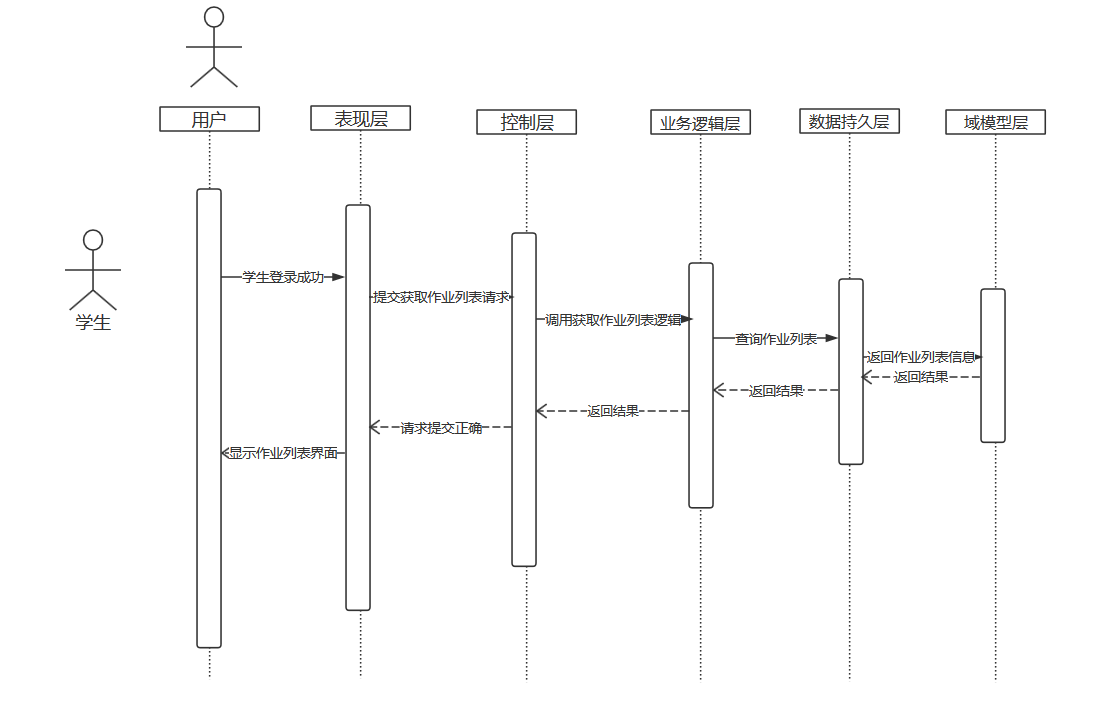


图 4-？ XXXXXX图

### 4.3.1 表现层

学生作业列表模块的表现层主要完成学生对作业的各种功能，学生作业列表页面时要求能完成学生所需要完成的任务（查看作业列表，作业详情，提交作业），提交作业确认后页面给出响应消息，提示提交成功或失败的提示。表现层对应的vue页面列表见表?-?所示。

表 4-3 登入模块表现层JSP列表

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 学生主界面/作业列表界面 | StudentIndex.vue | 学生登陆成功后，进入的学生主界面/查看作业列表作业 |
| 学生查看详情界面 | WorkDetail.vue | 学生查看作业详情界面，可进行作业提交 |

StudentIndex.vue和WorkDetail.vue的流程图如图4-3所示。

进入游首页

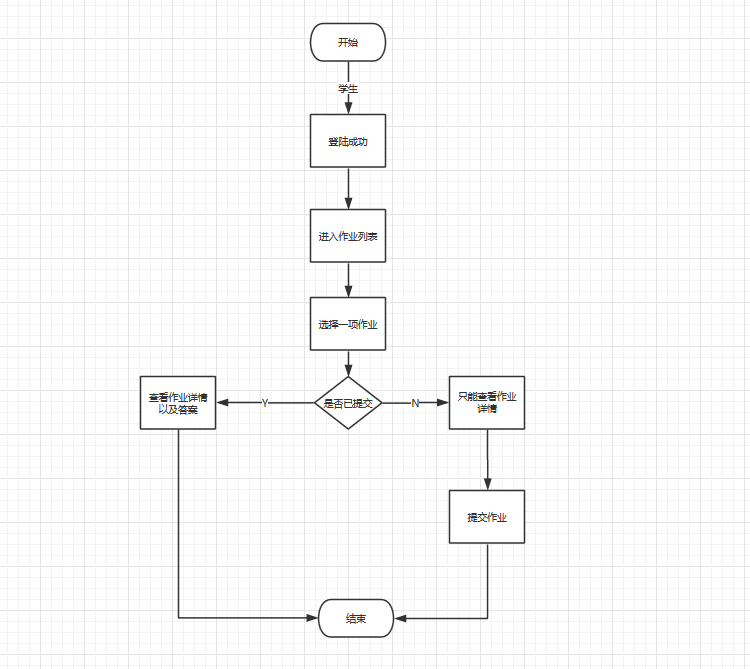


图 4-3 登录模块详细设计流程图

### 4.3.2 控制层

学生作业列表模块的控制层负责接受来自StudentIndex.vue和WorkDetail.vue的操作输入，同时调用学生作业列表模块的业务逻辑接口，将操作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 |
| 查看作业详情 | StudentIndex.vue | SUCCESS | WorkDetail.vue—显示相应的作业详情界面 |
| 作业提交 | WorkDetail.vue | SUCCESS | StudentIndex.vue—返回并显示提交成功的结果 |
| ERROR | WorkDetail.vue—显示提交成功的结果 |
| 查看答案 | StudentIndex.vue | SUCCESS | StudentIndex.vue—显示作业答案的窗口 |

在控制层中LoginController.java的描述如下所示：

package com.example.workpigai.controller;

/\*\*

\* *@description* 实现登录页面Action

\*/

**public** **class** LoginController

{

UserService userService;

/\*实现登入模块的控制层方法\*/

**public Result login(@RequestBody User requestUser) {}**

}

### 4.3.3 业务逻辑层

学生作业列表模块的业务逻辑层主要完成对学生业务操作的判定，同时调用学生作业列表模块的业务逻辑接口。比如：提交是否成功。学生作业列表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 |
| 查看作业详情 | UserService.service | 调用WorkDao和WorkDetailDao | 返回给Controller |
| 提交答案 | UserService.service | 调用WorkDao和WorkDetailDao | 返回给Controller |
| 提交作业 | UserService.service | 调用WorkDao | 返回给Controller |

在学生作业列表模块的业务逻辑层是调用了公用的UserService接口，同时在实现该接口。

UserService.java主要方法：

package com.example.workpigai.service;

**public class UserService {**

**UserDao userDao;**

/\*登入检测\*/

**public boolean isExist(String account) {}**

/\*查找用户\*/

**public User getByAccount(String account) {}**

/\*查找密码\*/

**public User get(String account, String password){}**

/\*增加用户\*/

**public void add(User user) { }**

/删除用户\*/

**public void deleteById(int id) { }**

}

UserService.java调用数据持久层的UserDao.Java，利用实例化对象来实现对数据持久层接口的调用。

### 4.3.4 数据持久层

学生作业列表模块的是数据持久层能对作业数据数据进行操作。学生作业列表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 |
| 查看作业详情 | WorkDao.java和WorkDeteilDao.Java | 调用 Work.Java和WorkDetailDao.java | 对作业详情进行查看 |
| 提交答案 | WorkDao.java | 调用 Work.Java | 对作业信息进行提交 |
| 查看作业答案 | WorkDao.java和WorkDeteilDao.Java | 调用 Work.Java和WorkDetailDao.java | 查看作业答案信息 |

WorkDao.java定义了作业信息进行增、删、改、查的接口：

WorkDeteilDao.Java定义了作业详情信息息进行增、删、改、查的接口：

package com.example.workpigai.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** UserDao JpaRepository<User,Integer>{

/\*查找用户信息\*/

**User findByAccount(String account);**

/\*匹配用户信息\*/

**User getByAccountAndPassword(String account,String password);** **public** boolean addUser(User user);

}

### 4.n.5 域模型层

学生作业列表模块用到域模型层中Work.java和WorkDetail.java，Work.java和WorkDetail.java是两个个公用域模型，在涉及到作业信息查询等操作时，就会调用到该模型，学生作业列表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 |
| Work.java | 对作业信息的增、删、改、查操作 |
| WorkDetail.java | 对作业详情信息的增、删、改、查操作 |

Work.java主要属性与方法：

**package com.example.workpigai.model;**

/\*\*

\* *@description* User表的信息记录

\*/

**public** **class** User {

**private** **String** id; /\*id识别\*/

**private Teacher teacher;** /\*教师信息\*/

**private WorkDetail workDetail;** /\*作业详情信息\*/

**private Student student;** /\*学生信息\*/

**String submitContent;** /\*提交内容 \*/

**String state;** /\*提交情况\*/

**String startTime;** /\*作业发布时间\*/

**String endTime;** /\*作业截止时间\*/

**int score;** /\*编写对上述各个属性的*getter*与*setter*方法\*/

}

WorkDetail.java主要属性与方法：

package com.example.workpigai.model;

/\*\*

\* *@description* User表的信息记录

\*/

**public** **class** User {

**private** **String** id; /\*id识别\*/

**String workTitle;** /\*作业标题\*/

**String publishContent;** /\*作业内容\*/

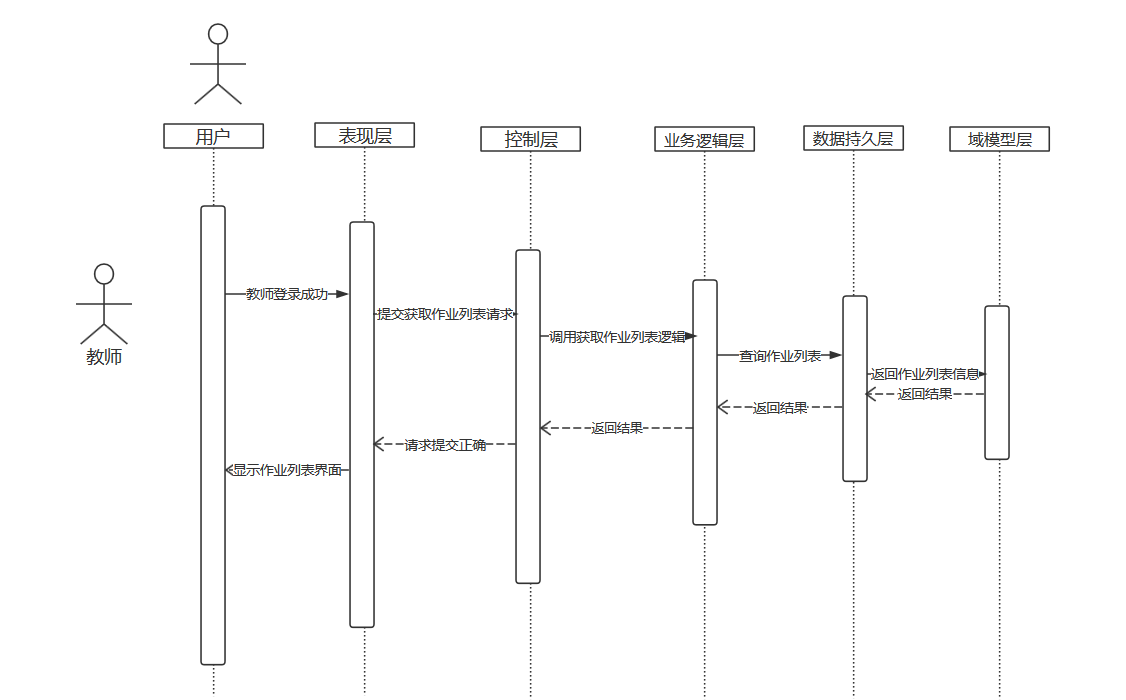
**String answer;**  /\*作业答案\*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

## 4.4、教师作业管理模块

教师登录成功后，系统内部的相应响应操作示意图如图4-2所示。

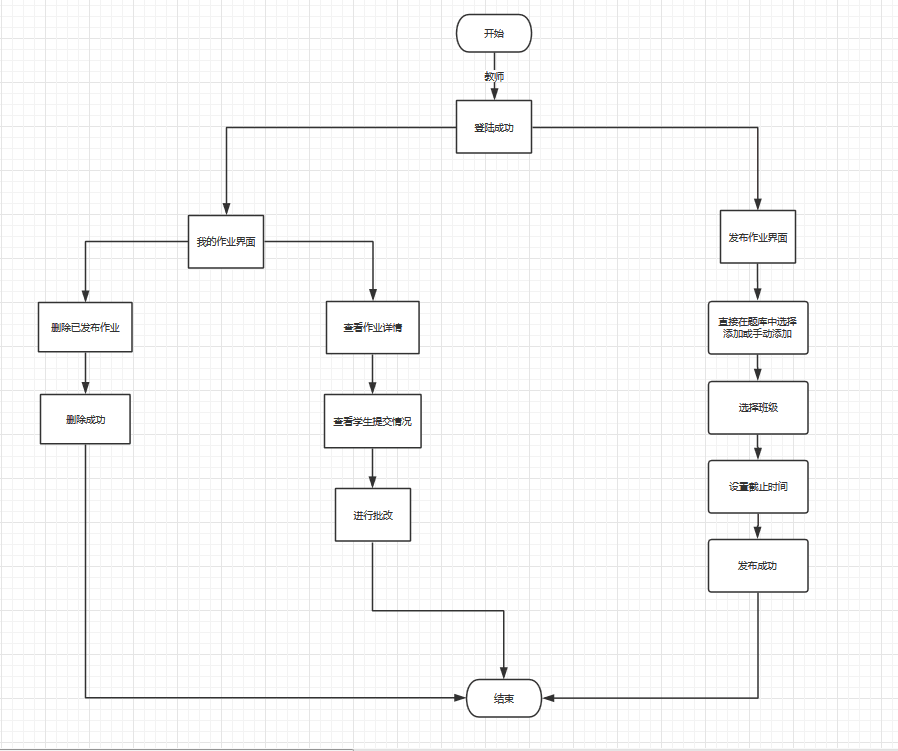


### 4.4.1 表现层

教师作业管理模块的表现层主要完成教师对作业的各种功能，教师作业管理页面时要求能完成学士所需要完成的任务（查看作业列表，作业详情，批改作业，删除作业，发布作业），发布作业、批改作业确认后页面给出响应消息，提示提交、批改成功或失败的提示。表现层对应的vue页面列表见表?-?所示。

表 4-3 登入模块表现层JSP列表

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 教师主界面/作业列表界面 | TeacherWork.vue | 教师登陆成功后，进入的教师主界面/查看作业管理列表 |
| 教师查看详情界面 | WorkDetailList.vue | 教师查看作业详情界面，可进行作业的批改 |
| 教师发布作业 | WorkPublish.vue | 教师发布作业界面，进行作业的发布 |

TeacherWork.vue、WorkDetailList.vue和WorkPublish.vue的流程图如图4-3所示。

### 4.n.2 控制层

教师作业管理模块的控制层负责接受来自TeacherWork.vue、WorkDetailList.vue和WorkPublish.vue的操作输入，同时调用教师作业管理模块的业务逻辑接口，将操作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。教师作业管理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 |
| 查看作业详情 | TecherWork.vue | SUCCESS | WorkDetailList.vue—显示相应的作业详情界面 |
| 发布作业 | WorkPublish.vue | SUCCESS | WorkPublish.vue—显示发布成功的结果 |
| ERROR | WorkPublish.vue—显示发布失败的结果 |
| 批改作业 | WorkDetailList.vue | SUCCESS | WorkDetailList.vue—显示批改成功的结果 |
| 删除作业 | TecherWork.vue | SUCCESS | TecherWork.vue—显示删除成功的结果 |
| ERROR | TecherWork.vue—显示删除失败的结果 |

在控制层中LoginController.java的描述如下所示：

package com.example.workpigai.controller;

/\*\*

\* *@description* 实现登录页面Action

\*/

**public** **class** LoginController

{

UserService userService;

/\*实现登入模块的控制层方法\*/

**public Result login(@RequestBody User requestUser) {}**

}

### 4.n.3 业务逻辑层

教师作业管理模块的业务逻辑层主要完成对教师业务操作的判定，同时调用教师管理作业模块的业务逻辑接口。比如：发布是否成功，删除是否成功，批改是否成功。教师作业管理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 |
| 查看作业详情 | UserService.service | 调用WorkDao和WorkDetailDao | 返回给Controller |
| 发布作业答案 | UserService.service | 调用WorkDao和WorkDetailDao | 返回给Controller |
| 删除作业 | UserService.service | 调用WorkDao | 返回给Controller |
| 批改作业 |  | 调用WorkDao | 返回给Controller |

在教师作业管理模块的业务逻辑层是调用了公用的UserService接口，同时在实现该接口。

UserService.java主要方法：

package com.example.workpigai.service;

**public class UserService {**

**UserDao userDao;**

/\*登入检测\*/

**public boolean isExist(String account) {}**

/\*查找用户\*/

**public User getByAccount(String account) {}**

/\*查找密码\*/

**public User get(String account, String password){}**

/\*增加用户\*/

**public void add(User user) { }**

/删除用户\*/

**public void deleteById(int id) { }**

}

UserService.java调用数据持久层的UserDao.Java，利用实例化对象来实现对数据持久层接口的调用。

### 4.n.4 数据持久层

教师作业管理模块的是数据持久层能对作业数据数据进行操作。教师作业管理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 |
| 查看作业详情 | WorkDao.java和WorkDeteilDao.Java | 调用 Work.Java和WorkDetail.java | 对作业详情进行查看 |
| 发布作业 | WorkDao.java和WorkDeteilDao.Java | 调用 Work.Java和WorkDetail.java | 对作业信息进行发布 |
| 删除作业 | WorkDao.java和WorkDeteilDao.Java | 调用 Work.Java和WorkDetail.java | 删除作业信息 |
| 批改作业 | WorkDao.java | 调用 Work.Java | 对作业进行批改 |

WorkDao.java定义了作业信息进行增、删、改、查的接口：

WorkDeteilDao.Java定义了作业详情信息息进行增、删、改、查的接口：

package com.example.workpigai.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** UserDao JpaRepository<User,Integer>{

/\*查找用户信息\*/

**User findByAccount(String account);**

/\*匹配用户信息\*/

**User getByAccountAndPassword(String account,String password);** **public** boolean addUser(User user);

}

### 4.n.5 域模型层

教师作业管理模块用到域模型层中Work.java和WorkDetail.java，Work.java和WorkDetail.java是两个个公用域模型，在涉及到作业信息查询等操作时，就会调用到该模型，教师作业管理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 |
| Work.java | 对作业信息的增、删、改、查操作 |
| WorkDetail.java | 对作业详情信息的增、删、改、查操作 |

Work.java主要属性与方法：

**package com.example.workpigai.model;**

/\*\*

\* *@description* User表的信息记录

\*/

**public** **class** User {

**private** **String** id; /\*id识别\*/

**private Teacher teacher;** /\*教师信息\*/

**private WorkDetail workDetail;** /\*作业详情信息\*/

**private Student student;** /\*学生信息\*/

**String submitContent;** /\*提交内容 \*/

**String state;** /\*提交情况\*/

**String startTime;** /\*作业发布时间\*/

**String endTime;** /\*作业截止时间\*/

**int score;** /\*编写对上述各个属性的*getter*与*setter*方法\*/

}

WorkDetail.java主要属性与方法：

package com.example.workpigai.model;

/\*\*

\* *@description* User表的信息记录

\*/

**public** **class** User {

**private** **String** id; /\*id识别\*/

**String workTitle;** /\*作业标题\*/

**String publishContent;** /\*作业内容\*/

**String answer;**  /\*作业答案\*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

## 4.5、教师题库管理模块

教师选择题库管理后，系统内部的相应响应操作示意图如图4-2所示。

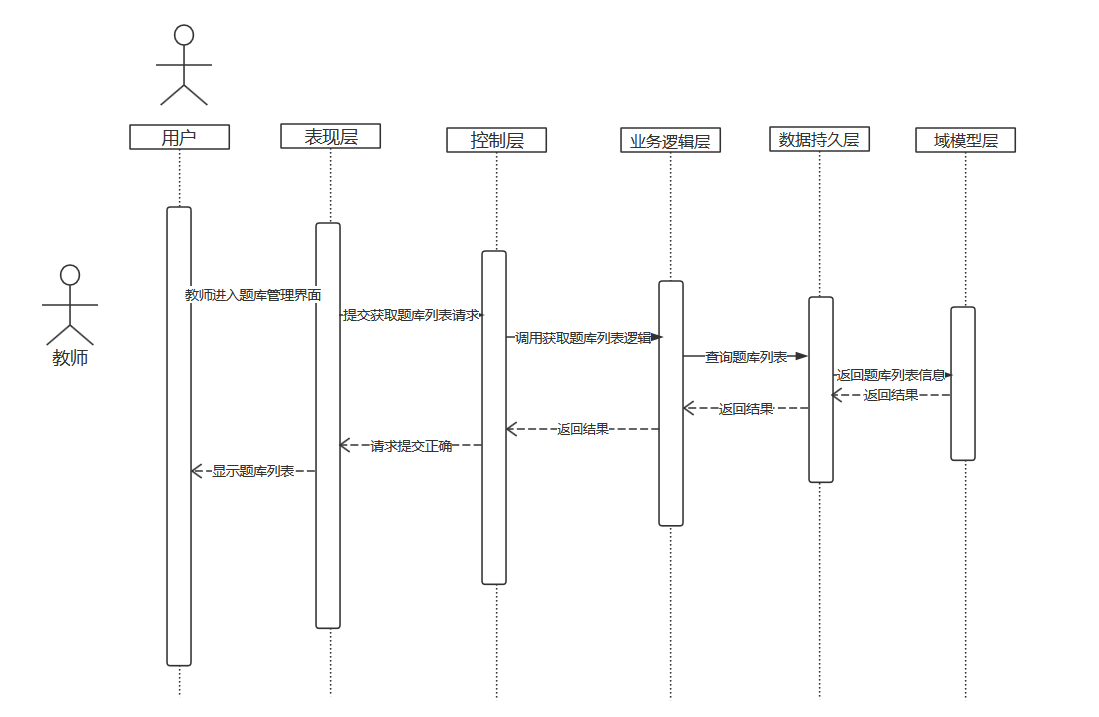


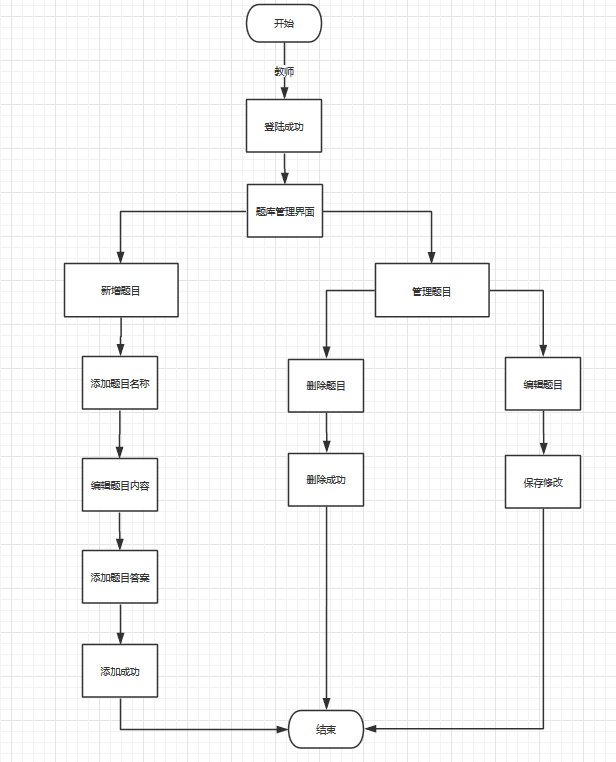
图 4-？ XXXXXX图

### 4.n.1 表现层

教师题库管理模块的表现层主要完成教师对题库的各种功能，教师题库管理页面时要求能完成教师所需要完成的任务（查看题库列表，修改题目信息，删除题目信息，新增题目信息），新增题目、删除题目、修改题目确认后页面给出响应消息，新增题目、删除题目、修改题目成功或失败的提示。表现层对应的vue页面列表见表?-?所示。

表 4-3 教师题库管理模块表现层JSP列表

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 题库管理界面 | BankManagement.vue | 教师选择题库管理后，进入题库管理列表界面 |

BankManagement.vue的流程图如图4-3所示。

### 4.n.2 控制层

教师题库管理模块的控制层负责接受来自BankManagement.vue的操作输入，同时调用教师题库管理模块的业务逻辑接口，将操作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。教师作业管理模块控制层列表见表4-4所示。

表 4-4 教师题库管理模块控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 查看题目详情 | BankManagement.vue | SUCCESS | BankManagement.vue—显示相应的作业详情界面 |
| 新增题目 | BankManagement.vue | SUCCESS | BankManagement.vue—显示新增成功的结果 |
| ERROR | BankManagement.vue—显示新增失败的结果 |
| 修改题目 | BankManagement.vue | SUCCESS | BankManagement.vue—显示修改成功的结果 |
| ERROR | BankManagement.vue—显示修改失败的结果 |
| 删除题目 | BankManagement.vue | SUCCESS | BankManagement.vue—显示删除成功的结果 |
| ERROR | BankManagement.vue—显示删除失败的结果 |

在控制层中LoginController.java的描述如下所示：

package com.example.workpigai.controller;

/\*\*

\* *@description* 实现登录页面Action

\*/

**public** **class** LoginController

{

UserService userService;

/\*实现登入模块的控制层方法\*/

**public Result login(@RequestBody User requestUser) {}**

}

### 4.n.3 业务逻辑层

教师题目管理模块的业务逻辑层主要完成对教师业务操作的判定，同时调用教师管理作业模块的业务逻辑接口。比如：新增是否成功，修改是否成功，删除是否成功。教师题目管理模块业务逻辑层列表如图4-5所示。

表 4-5教师题目管理模块业务逻辑层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 查看题目详情 | UserService.service | 调用QuestionBankDaoDao | 返回给Controller |
| 新增题目 | UserService.service | 调用QuestionBankDao | 返回给Controller |
| 删除题目 | UserService.service | 调用QuestionBankDao | 返回给Controller |
| 批改修改 |  | 调用QuestionBankDao | 返回给Controller |

在教师作业管理模块的业务逻辑层是调用了公用的UserService接口，同时在实现该接口。

UserService.java主要方法：

package com.example.workpigai.service;

**public class UserService {**

**UserDao userDao;**

/\*登入检测\*/

**public boolean isExist(String account) {}**

/\*查找用户\*/

**public User getByAccount(String account) {}**

/\*查找密码\*/

**public User get(String account, String password){}**

/\*增加用户\*/

**public void add(User user) { }**

/删除用户\*/

**public void deleteById(int id) { }**

}

UserService.java调用数据持久层的UserDao.Java，利用实例化对象来实现对数据持久层接口的调用。

### 4.n.4 数据持久层

教师题目管理模块的是数据持久层能对题目数据数据进行操作。教师题目管理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 |
| 查看题目详情 | QuestionBankDao.java | 调用 QuestionBank.java | 对题目详情进行查看 |
| 新增题目 | QuestionBankDao.java | 调用 QuestionBank.java | 新增题目信息 |
| 删除题目木 | QuestionBankDao.java | 调用 QuestionBank.java | 删除题目信息 |
| 修改题目 | QuestionBankDao.java | 调用 QuestionBank.java | 对题目信息进行修改 |

QuestionBank.java定义了题目信息进行增、删、改、查的接口：

package com.example.workpigai.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** UserDao JpaRepository<User,Integer>{

/\*查找用户信息\*/

**User findByAccount(String account);**

/\*匹配用户信息\*/

**User getByAccountAndPassword(String account,String password);** **public** boolean addUser(User user);

}

### 4.n.5 域模型层

教师题目管理模块用到域模型层中QuestionBank.java是一个公用域模型，在涉及到题目信息查询等操作时，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 |
| QuestionBank.java | 对题目信息的增、删、改、查操作 |

QuestionBank.java主要属性与方法：

**package com.example.workpigai.model;**

/\*\*

\* *@description* User表的信息记录

\*/

**public** **class** QuestionBank {

**private** **String** id; /\*id识别\*/

**int questionId;** /\*题号/

**String title;** /\*题目标题\*/

**String content;** /\*题目内容\*/

**String answer;** /\*题目答案\*/

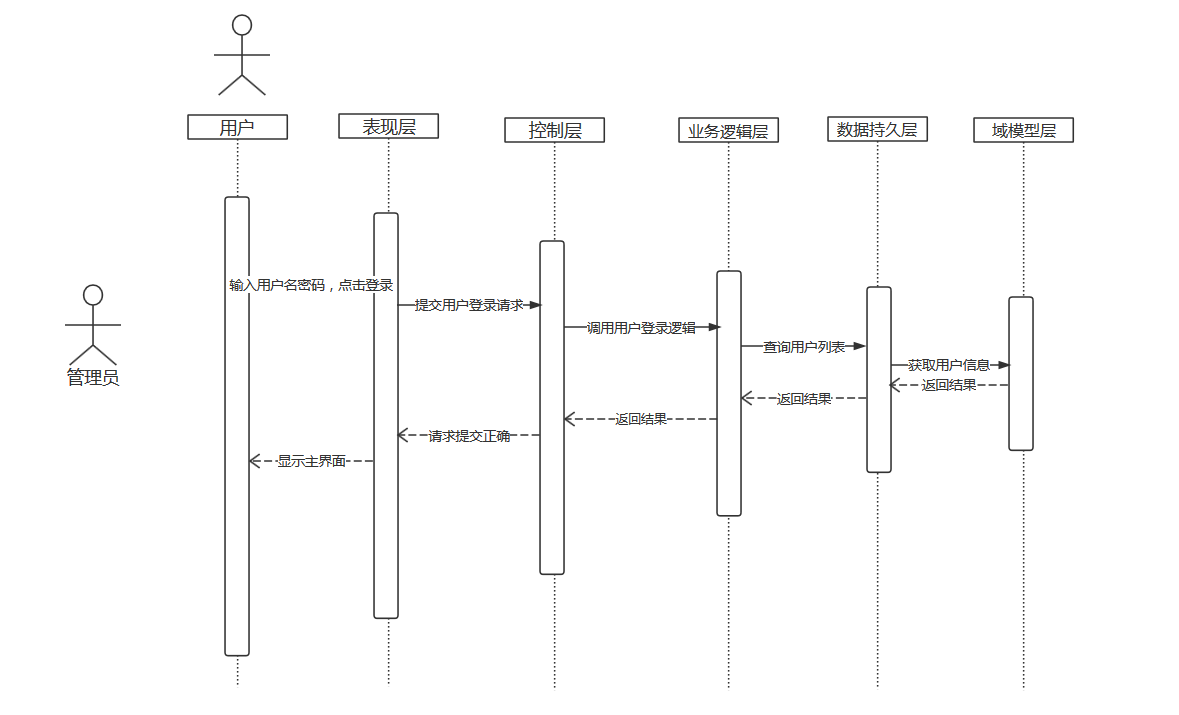
/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

# 5、管理端模块详细设计

## 5.1、管理员登录模块详细设计

在教师登录模块时，系统内部的相应响应操作示意图如图4-2所示。



### 5.1.1 表现层

管理员登录模块的表现层主要完成管理员的登录功能，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vue页面列表见表?-?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 | AdminLogin.vue | 管理员登入功能，当登入出错时给出提示。 |

AdminLogin.vue的流程图如图4-3所示。

进入游首页

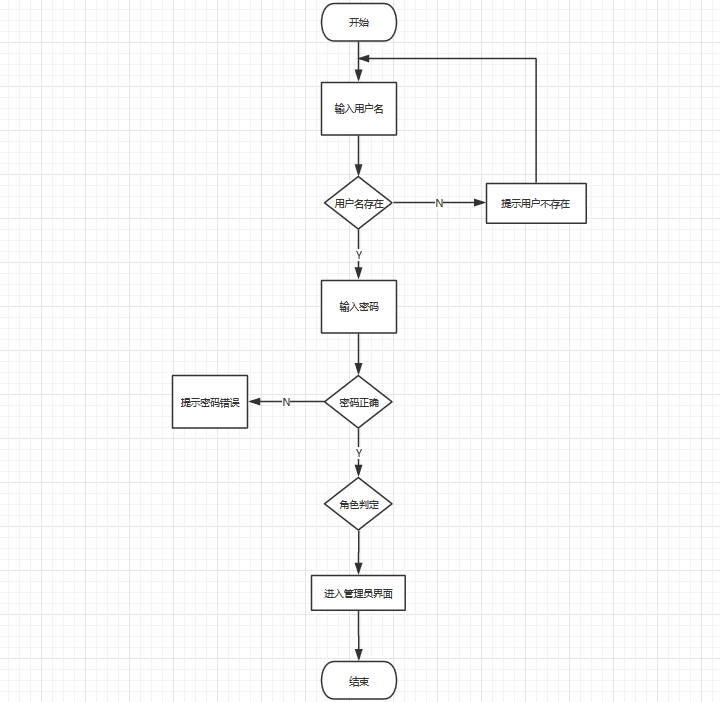


图 4-3 管理员登录模块详细设计流程图

### 5.1.2 控制层

管理员登录模块的控制层负责接受来自AdminLogin.vue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 | AdminLoginController.java | SUCCESS | TeacherInfo.vue—登入成功，显示提示窗口，并管理员主界面。 |
| ERROR | Login.vue---登入失败，显示提示窗口。 |

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

package com.example.workpigai.controller;

/\*\*

\* *@description* 实现登录页面Action

\*/

**public** **class** AdminLoginController

{

UserService userService;

/\*实现登入模块的控制层方法\*/

**public Result login(@RequestBody User requestUser) {}**

}

### 5.1.3 业务逻辑层

管理员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用户名是否存在、密码是否正确。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 | UserService.service | 调用UserDao | 返回给Controller |

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

UserService.java主要方法：

package com.example.workpigai.service;

**public class UserService {**

**UserDao userDao;**

/\*登入检测\*/

**public boolean isExist(String account) {}**

/\*查找用户\*/

**public User getByAccount(String account) {}**

/\*查找密码\*/

**public User get(String account, String password){}**

/\*增加用户\*/

**public void add(User user) { }**

/删除用户\*/

**public void deleteById(int id) { }**

}

UserService.java调用数据持久层的UserDao.Java，利用实例化对象来实现对数据持久层接口的调用。

### 5.1.4 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 | UserDao.java | 调用 user.Java | 对用户信息进行增删改查操作 |

UserDao.java定义了用户信息进行增、删、改、查的接口：

package com.example.workpigai.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** UserDao JpaRepository<User,Integer>{

/\*查找用户信息\*/

**User findByAccount(String account);**

/\*匹配用户信息\*/

**User getByAccountAndPassword(String account,String password);** **public** boolean addUser(User user);

}

### 5.1.5 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.example.workpigai.model;**

/\*\*

\* *@description* User表的信息记录

\*/

**public** **class** User {

**private** **String** id; /\*id识别\*/

**String account;** /\*用户名\*/

**String password;** /\*密码\*/

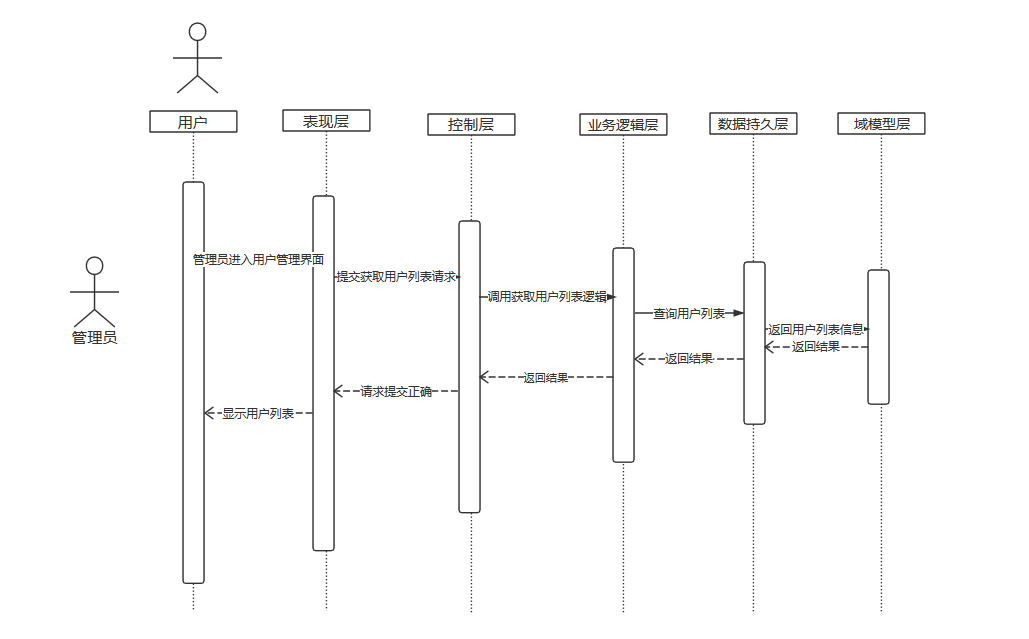
**int type;** /\*身份\*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

## 5.2、用户管理模块详细设计

管理员登录成功后进入用户管理界面，系统内部的相应响应操作示意图如图4-2所示。



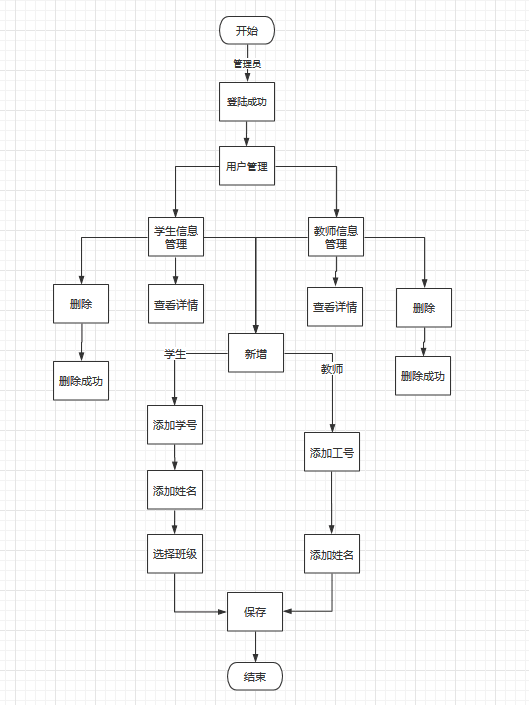
### 5.n.1 表现层

管理员用户管理模块的表现层主要完成管理员对用户信息的各种功能，管理员用户管理页面时要求能完成用户管理所需要完成的任务（查看用户列表，修改用户信息，删除用户信息，新增用户信息），新增用户、删除用户、修改用户确认后页面给出响应消息，新增用户、删除用户、修改用户成功或失败的提示。表现层对应的vue页面列表见表?-?所示。

表 4-3 登入模块表现层JSP列表

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 教师信息管理界面 | TeacherInfo.vue | 管理员登入成功后，进入教师信息管理界面 |
| 学生信息管理界面 | StudentInfo.vue | 管理员选择学生信息管理后，进入学生信息管理界面 |

AdminLogin.vue和StudentInfo.vue的流程图如图4-3所示。



### 5.n.2 控制层

管理员用户管理模块的控制层负责接受来自AdminLogin.vue和StudentInfo.vue的用户输入，同时调用用户管理模块的业务逻辑接口，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。管理员用户管理模块控制层列表见表4-4所示。

表 4-4 用户管理模块控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 新增用户 | TeacherInfo.java和StudentInfo.java | SUCCESS | TeacherInfo.java和StudentInfo.java—显示新增成功信息 |
| ERROR | TeacherInfo.java和StudentInfo.java—显示新增失败信息 |
| 删除用户 | TeacherInfo.java和StudentInfo.java | SUCCESS | TeacherInfo.java和StudentInfo.java—显示删除成功信息 |
| ERROR | TeacherInfo.java和StudentInfo.java—显示删除失败信息 |
| 修改用户 | TeacherInfo.java和StudentInfo.java | SUCCESS | TeacherInfo.java和StudentInfo.java—显示修改成功信息 |
| ERROR | TeacherInfo.java和StudentInfo.java—显示修改失败信息 |

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

package com.example.workpigai.controller;

/\*\*

\* *@description* 实现登录页面Action

\*/

**public** **class** AdminLoginController

{

UserService userService;

/\*实现登入模块的控制层方法\*/

**public Result login(@RequestBody User requestUser) {}**

}

### 5.n.3 业务逻辑层

管理员用户管理模块的业务逻辑层主要完成对用户管理业务操作的判定，同时调用用户管理模块的业务逻辑接口。比如：新增是否成功，修改是否成功，删除是否成功。用户管理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 |
| 新增用户 | UserService.service | 调用UserDao | 返回给Controller |
| 删除用户 |  | 调用UserDao | 返回给Controller |
| 修改用户 |  | 调用UserDao | 返回给Controller |

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

UserService.java主要方法：

package com.example.workpigai.service;

**public class UserService {**

**UserDao userDao;**

/\*登入检测\*/

**public boolean isExist(String account) {}**

/\*查找用户\*/

**public User getByAccount(String account) {}**

/\*查找密码\*/

**public User get(String account, String password){}**

/\*增加用户\*/

**public void add(User user) { }**

/删除用户\*/

**public void deleteById(int id) { }**

}

UserService.java调用数据持久层的UserDao.Java，利用实例化对象来实现对数据持久层接口的调用。

### 5.n.4 数据持久层

用户管理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 |
| 用户管理 | UserDao.java | 调用 user.Java | 对用户信息进行增删改查操作 |

UserDao.java定义了用户信息进行增、删、改、查的接口：

package com.example.workpigai.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** UserDao JpaRepository<User,Integer>{

/\*查找用户信息\*/

**User findByAccount(String account);**

/\*匹配用户信息\*/

**User getByAccountAndPassword(String account,String password);** **public** boolean addUser(User user);

}

### 5.n.5 域模型层

用户管理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.example.workpigai.model;**

/\*\*

\* *@description* User表的信息记录

\*/

**public** **class** User {

**private** **String** id; /\*id识别\*/

**String account;** /\*用户名\*/

**String password;** /\*密码\*/

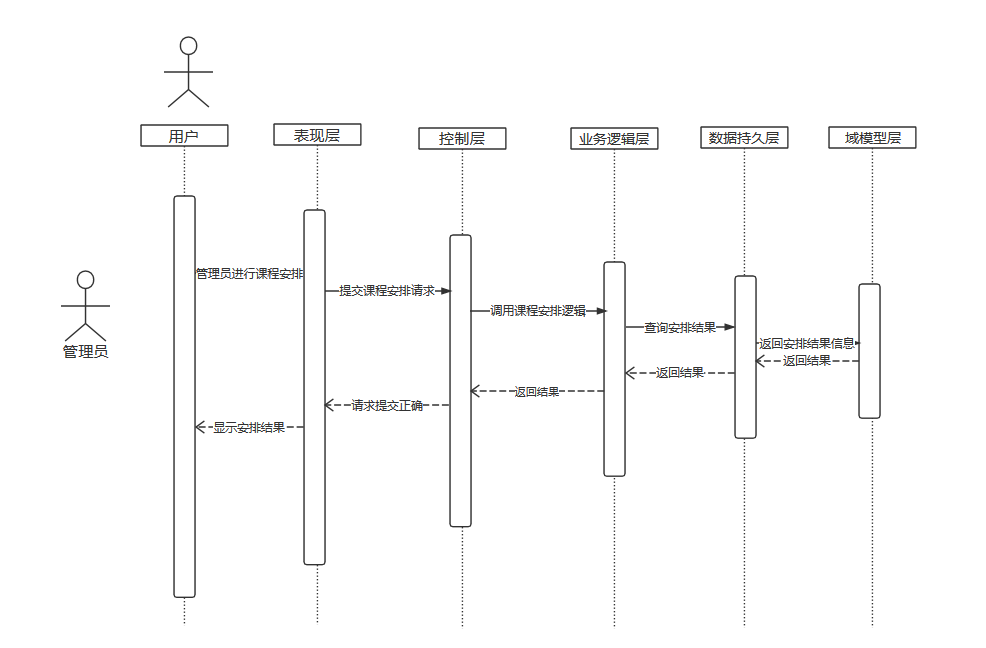
**int type;** /\*身份\*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

## 5.3、课程安排模块详细设计

管理员选择课程安排后，进入课程安排界面，系统内部的相应响应操作示意图如图4-2所示。



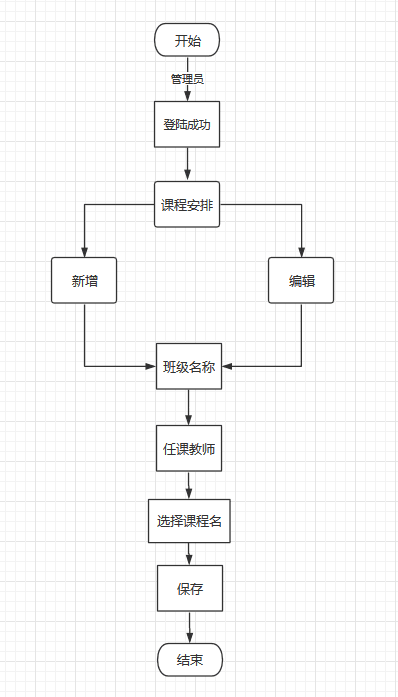
### 5.n.1 表现层

管理员课程安排模块的表现层主要完成管理员对课程信息的各种功能，管理员课程管理页面时要求能完成用户管理所需要完成的任务（查看课程列表，修改课程信息，删除课程信息，新增课程信息），新增课程、删除课程、修改课程确认后页面给出响应消息，新增课程、删除课程、修改课程成功或失败的提示。表现层对应的vue页面列表见表?-?所示。

表 4-3 登入模块表现层JSP列表

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 课程安排界面 | Course.vue | 管理员选择课程管理后，进入课程安排界面 |

Course.vue的流程图如图4-3所示。



### 5.n.2 控制层

管理员课程安排模块的控制层负责接受来自Course.vue的用户输入，同时调用用户管理模块的业务逻辑接口，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。管理员课程安排模块控制层列表见表4-4所示。

表 4-4 用户管理模块控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 新增课程 | Course.vue | SUCCESS | Course.vue—显示新增成功信息 |
| ERROR | Course.vue—显示新增失败信息 |
| 删除课程 | Course.vue | SUCCESS | Course.vue—显示删除成功信息 |
| ERROR | Course.vue—显示删除失败信息 |
| 修改课程 | Course.vue | SUCCESS | Course.vue—显示修改成功信息 |
| ERROR | Course.vue—显示修改失败信息 |

在控制层中AdminController.java的描述如下所示：

package com.example.workpigai.controller;

/\*\*

\* *@description* 实现登录页面Action

\*/

**public** **class** AdminLoginController

{

UserService userService;

/\*实现登入模块的控制层方法\*/

**public Result login(@RequestBody User requestUser) {}**

}

### 5.n.3 业务逻辑层

管理员课程安排模块的业务逻辑层主要完成对用户管理业务操作的判定，同时调用课程安排模块的业务逻辑接口。比如：新增是否成功，修改是否成功，删除是否成功。课程安排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 |
| 新增课程 | UserService.service | 调用ChoseCourseDao | 返回给Controller |
| 删除课程 |  | 调用ChoseCourseDao | 返回给Controller |
| 修改课程 |  | 调用ChoseCourseDao | 返回给Controller |

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

UserService.java主要方法：

package com.example.workpigai.service;

**public class UserService {**

**UserDao userDao;**

/\*登入检测\*/

**public boolean isExist(String account) {}**

/\*查找用户\*/

**public User getByAccount(String account) {}**

/\*查找密码\*/

**public User get(String account, String password){}**

/\*增加用户\*/

**public void add(User user) { }**

/删除用户\*/

**public void deleteById(int id) { }**

}

UserService.java调用数据持久层的UserDao.Java，利用实例化对象来实现对数据持久层接口的调用。

### 5.n.4 数据持久层

课程安排模块的是数据持久层能对课程数据进行增删改查。课程安排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 |
| 课程安排 | ChoseCourseDao.java | 调用 ChoseCourse.Java | 对课程信息进行增删改查操作 |

ChoseCourseDao.java定义了用户信息进行增、删、改、查的接口：

package com.example.workpigai.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** UserDao JpaRepository<User,Integer>{

/\*查找用户信息\*/

**User findByAccount(String account);**

/\*匹配用户信息\*/

**User getByAccountAndPassword(String account,String password);** **public** boolean addUser(User user);

}

### 5.n.5 域模型层

课程安排模块用到域模型层中 ChoseCourse java，ChoseCourse.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 |
| ChoseCourse.java | 对课程信息的增、删、改、查操作 |

ChoseCourse.java主要属性与方法：

package com.example.workpigai.model;

/\*\*

\* *@description* User表的信息记录

\*/

**public** **class** ChoseCourse {

**private** **String** id; /\*id识别\*/

Teacher teacher; /\*教师信息/

Class mClass; /\*班级信息\*/

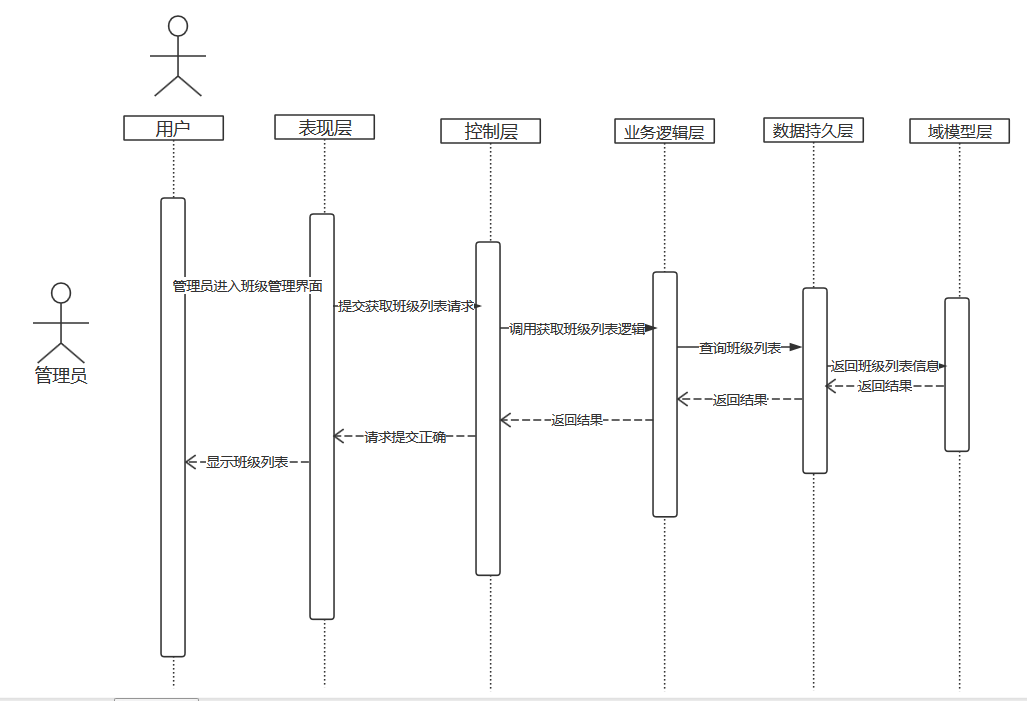
String courseName; /\*课程名称\*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

## 5.4、班级管理模块详细设计

管理员选择班级管理后，进入班级管理界面，系统内部的相应响应操作示意图如图4-2所示。



### 5.n.1 表现层

管理员班级管理模块的表现层主要完成管理员对班级信息的各种功能，管理员班级管理页面时要求能完成班级管理所需要完成的任务（查看班级列表，修改班级信息，删除班级信息，新增班级信息），新增班级、删除班级、修改班级确认后页面给出响应消息，新增班级、删除班级、修改班级成功或失败的提示。表现层对应的vue页面列表见表?-?所示。

表 4-3 登入模块表现层JSP列表

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 班级管理界面 | ClassInfo.vue | 管理员选择班级管理后，进入班级管理界面 |

ClassInfo.vue的流程图如图4-3所示。

### 5.n.2 控制层

管理员班级管理模块的控制层负责接受来自ClassInfo.vue的用户输入，同时调用班级管理模块的业务逻辑接口，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。管理员班级管理模块控制层列表见表4-4所示。

表 4-4 班级管理模块控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 新增班级 | ClassInfo.vue | SUCCESS | ClassInfo.vue—显示新增成功信息 |
| ERROR | ClassInfo.vue—显示新增失败信息 |
| 删除班级 | ClassInfo.vue | SUCCESS | ClassInfo.vue—显示删除成功信息 |
| ERROR | ClassInfo.vue—显示删除失败信息 |
| 修改班级 | ClassInfo.vue | SUCCESS | ClassInfo.vue—显示修改成功信息 |
| ERROR | ClassInfo.vue—显示修改失败信息 |

在控制层中AdminController.java的描述如下所示：

package com.example.workpigai.controller;

/\*\*

\* *@description* 实现登录页面Action

\*/

**public** **class** AdminLoginController

{

UserService userService;

/\*实现登入模块的控制层方法\*/

**public Result login(@RequestBody User requestUser) {}**

}

### 5.n.3 业务逻辑层

管理员班级管理模块的业务逻辑层主要完成对班级管理业务操作的判定，同时调用班级管理模块的业务逻辑接口。比如：新增是否成功，修改是否成功，删除是否成功。班级管理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 |
| 新增课程 | UserService.service | 调用ClassDao | 返回给Controller |
| 删除课程 |  | 调用ClassDao | 返回给Controller |
| 修改课程 |  | 调用ClassDao | 返回给Controller |

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

UserService.java主要方法：

package com.example.workpigai.service;

**public class UserService {**

**UserDao userDao;**

/\*登入检测\*/

**public boolean isExist(String account) {}**

/\*查找用户\*/

**public User getByAccount(String account) {}**

/\*查找密码\*/

**public User get(String account, String password){}**

/\*增加用户\*/

**public void add(User user) { }**

/删除用户\*/

**public void deleteById(int id) { }**

}

UserService.java调用数据持久层的UserDao.Java，利用实例化对象来实现对数据持久层接口的调用。

### 5.n.4 数据持久层

班级管理模块的是数据持久层能对班级数据进行增删改查。班级管理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 |
| 课程安排 | ClassDao.java | 调用 Class.Java | 对班级信息进行增删改查操作 |

ClassDao.java定义了用户信息进行增、删、改、查的接口：

package com.example.workpigai.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** UserDao JpaRepository<User,Integer>{

/\*查找用户信息\*/

**User findByAccount(String account);**

/\*匹配用户信息\*/

**User getByAccountAndPassword(String account,String password);** **public** boolean addUser(User user);

}

### 5.n.5 域模型层

班级管理模块用到域模型层中 Class java，Class.java是一个公用域模型，在涉及到班级信息查询等操作时，就会调用到该模型，班级管理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 |
| Class.java | 对班级信息的增、删、改、查操作 |

Class.java主要属性与方法：

package com.example.workpigai.model;

/\*\*

\* *@description* User表的信息记录

\*/

**public** **class** ChoseCourse {

**private** **String** id; /\*id识别\*/

**String className;**  /\*班级名称\*/

**int classId;**  /\*班级号\*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}