|  |  |
| --- | --- |
| **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**

编写人员：黄旭辉、姚光磊、黄龙

编写日期：2019年6月25日

评审日期： 2019年6月25日

目录

[1．导言 4](#_Toc902)

[1.1 目的 4](#_Toc16098)

[1.2 范围 4](#_Toc4508)

[1.3 引用标准 4](#_Toc1779)

[1.4 参考资料 5](#_Toc10894)

[1.5 版本更新信息 5](#_Toc4171)

[2．详细设计简述 5](#_Toc6934)

[2.1 设计简介 5](#_Toc32162)

[2.2 模块简介 6](#_Toc26726)

[3、界面详细设计 6](#_Toc24400)

[3.1、页面设计说明 6](#_Toc18848)

[3.2、静态页面详细设计 7](#_Toc32626)

[4、客户端模块详细设计 7](#_Toc6635)

[4.1、学生模块 7](#_Toc2087)

[4.1.1 查看作业模块 7](#_Toc28717)

[4.1.2 填写、提交作业模块 12](#_Toc7387)

[4.1.3搜索作业模块 18](#_Toc12736)

[4.2、教师模块 25](#_Toc1718)

[4.2.1教师搜索作业 25](#_Toc24469)

[4.2.2教师布置作业 30](#_Toc21925)

[4.2.3教师批改作业 35](#_Toc6560)

[4.2.4教师查看作业 40](#_Toc8861)

[5、 管理端模块详细设计 45](#_Toc29322)

[5.1管理员添加账号 46](#_Toc17671)

[5.1.1 表现层 46](#_Toc15858)

[5.1.2 控制层 46](#_Toc7286)

[5.1.3 业务逻辑层 46](#_Toc5581)

[5.1.4 数据持久层 46](#_Toc22801)

[5.1.5 域模型层 46](#_Toc30622)

[5.2管理员修改密码 46](#_Toc16475)

[5.2.1 表现层 46](#_Toc8353)

[5.2.2 控制层 46](#_Toc6471)

[5.2.3 业务逻辑层 46](#_Toc18464)

[5.2.4 数据持久层 46](#_Toc425)

[5.2.5 域模型层 46](#_Toc4481)

[5.3管理员搜索账号 46](#_Toc16527)

[5.3.1 表现层 46](#_Toc7131)

[5.3.2 控制层 46](#_Toc1028)

[5.3.3 业务逻辑层 46](#_Toc782)

[5.3.4 数据持久层 46](#_Toc11853)

[5.3.5 域模型层 47](#_Toc1711)

[5.4管理员查看信息 47](#_Toc13724)

[5.4.1 表现层 47](#_Toc27979)

[5.4.2 控制层 47](#_Toc29388)

[5.4.3 业务逻辑层 47](#_Toc11981)

[5.4.4 数据持久层 47](#_Toc24238)

[5.4.5 域模型层 47](#_Toc1804)

[6、公共端模块详细设计 47](#_Toc31456)

[6.1、登录模块 47](#_Toc22986)

[6.1.1 表现层 48](#_Toc19021)

[6.1.2 控制层 49](#_Toc23468)

[6.1.3 业务逻辑层 50](#_Toc7881)

[6.1.4 数据持久层 51](#_Toc20467)

[6.1.5 域模型层 52](#_Toc27746)

# 1．导言

## 1.1 目的

该文档是对学生作业提交系统的概要设计进行的详细说明，是为了指导和规范学生作业提交系统的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的所有功能模块的详细设计描述，学生系统下的，作业查找，作业书写，作业提交。教师系统下的，作业搜索，作业发布，作业批改功能。

## 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1-1所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 修改编号 | 修改日期 | 修改后版本 | 修改位置 | 修改内容概述 |
| 001 | 2019.6.25 | 0.1 | 全部 | 初始发布版本 |
| 002 | 2019.6.26 | 1.0 | 4章节 | 增加 |
| 003 | 2019.6.27 | 2.0 | 4章节 | 修改 |
| 004 | 2019.6.28 | 3.0 | 4、6章节 | 修改 |
| 005 | 2019.6.29 | 4.0 | 6、7章节 | 修改、增加 |

表1-1 文档更新记录

# 2．详细设计简述

## 2.1 设计简介

学生作业提交系统分层结构图如图2-1所示。

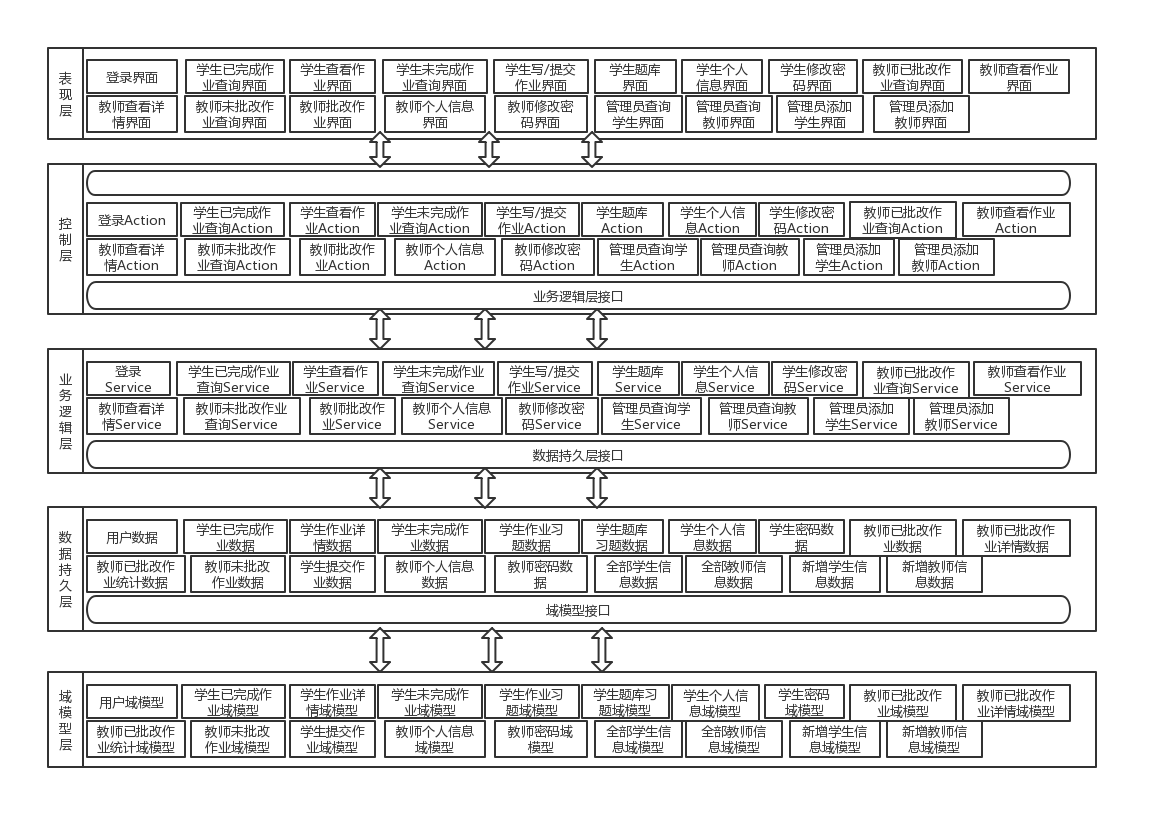


图 2-1 系统分层结构图

## 2.2 模块简介

系统功能模块示意图如图2-2所示。

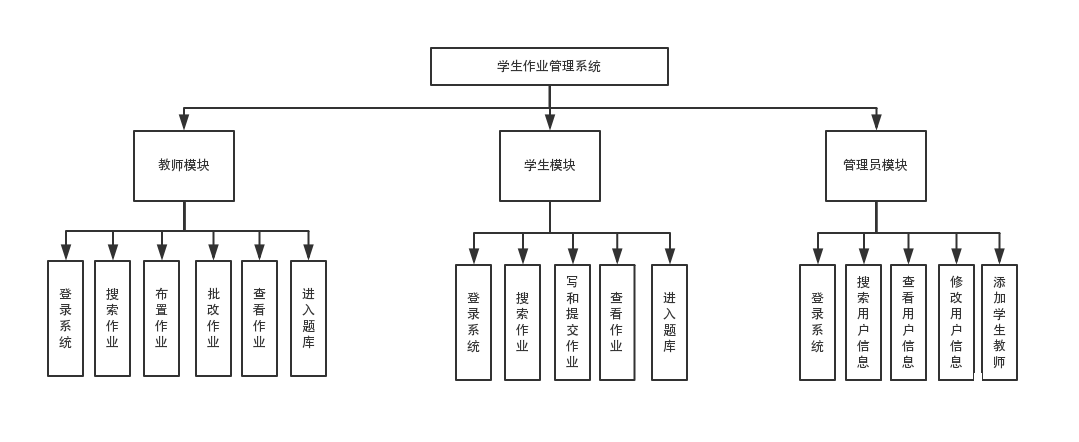


图 2-2 系统功能模块示意图

# 3、界面详细设计

## 3.1、页面设计说明

1.统一采用相同的色调风格

2.界面模块大小适中

3.字体大小适中

4.事件有相应的响应反馈

5.页面统一使用html以及jsp进行编写

## 3.2、静态页面详细设计

登录模块系统静态页面如下表3-1所示

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 登录界面 | Index.jsp | 实现学生、教师、管理员登录 |

表3-1 登录模块表现层设计

学生模块静态页面如下表3-2所示

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 学生主界面 | Student.jsp | 学生主要界面，显示完成作业 |
| 学生主界面 | Student1.jsp | 学生主要界面，显示未完成作业 |
| 作业详情界面 | studentWatchHomework.jsp | 作业完成情况，批改情况 |
| 作业书写界面 | WriteHomework.jsp | 作业填写答案，保存答案 |
| 学生主界面 | Student.jsp | 学生主要界面，显示完成作业 |

表3-2 学生模块表现层设计

教师模块静态页面如下表3-3所示

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 学生主界面 | Teacher.jsp | 教师主界面主要界面，各功能模块入口，显示已批改作业。 |
| 学生主界面 | Teacher1.jsp | 教师主界面主要界面，各功能模块入口，显示未批改作业。 |
| 作业详情界面 | TeacherWatchHomework.jsp | 作业题目详细情况 |
| 作业提交情况界面 | homeworkDetail.jsp | 学生提交情况，学生成绩，平均分 |
| 布置作业界面 | AssignHomeworkStepOne.jsp | 作业所需题目详情。 |

表3-3 教师模块表现层设计

# 4、客户端模块详细设计

客户端模块主要包括有俩个模块，分别为：学生模块以及教师模块。

## 4.1、学生模块

学生模块分为：查看作业模块，搜索作业和填写、提交作业。

### 4.1.1 学生查看作业

在学生作业查看模块时，系统内部的相应响应操作时序图如图4-1所示。

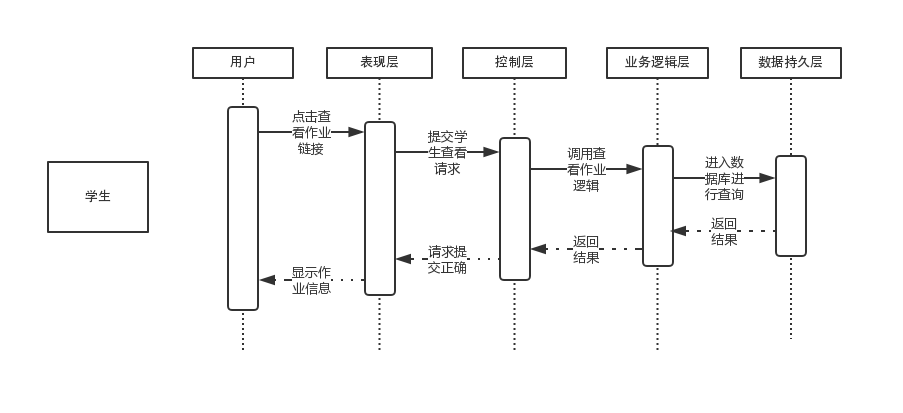


图 4-1学生查看作业系统响应示意图

#### 4.1.1.1 表现层

查看作业模块的表现层主要完成作业详情的显示。表现层对应的JSP页面列表见表4-1所示。

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 查看作业页面 | StudentWatchHomework.jsp | 用户（学生）查看已经的作业的详情。 |

表 4-1 查看作业模块表现层JSP列表

#### 4.1.1.2 控制层

查看模块的控制层负责接收来自StudentIndex.jsp的用户输入，同时调用登入模块的业务逻辑接口，将相应信息传到表现层，并决定显示页面。查看作业模块控制层列表见表4-2所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 查看详情 | StudentAssignmentController.java | SUCCESS | StudentWatchHomework.jsp，显示对应的页面。 |

表 4-2查看作业模块控制层列表

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

**package com.gxun.jms.controller;**

/\*\*

\* *@description* 实现登录页面的控制

\*/

@RestController（控制器）

public class StudentAssignmentController{

/\*对用户输入数据的获取\*/

/\* 对数据进行简单的处理\*/

}

#### 4.1.1.3 业务逻辑层

查看作业模块的业务逻辑层主要完成对用户所需作业信息的获取，同时调用查看模块的业务逻辑接口。查看作业模块业务逻辑层列表见表4-3所示

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 查看详情 | StudentAssignmentService.java | 调用StudentAssignmentMapper.java | 返回给Action |

表 4-3查看作业模块业务逻辑层列表

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

StudentAssignmentService.java接口：

package com.gxun.jms.service;

@Service

public class StudentAssignmentService{

@Autowired  
private StudentAssignmentMapper studentAssignmentMapper;

}

StudentAssignmentService.java调用数据持久层的StudentAssignmentMapper.java

StudentAssignmentMapper.java接口：

package com.gxun.jms.dao;

@Mapper

public interface StudentAssignmentMapper{

}

#### 4.1.1.4 数据持久层

查看作业模块的数据持久层可以在选择查看作业后系统获取相应的作业数据，查看模块数据持久层列表见表4-4所示

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 查看 | StudentAssignmentMapper.java | （ps：没找到返回谁） | 利用接口对用户信息进行操作 |

表 4-4查看作业模块数据持久层列表

StudentAssignmentMapper.java定义了对用户信息操作的接口：

package com.gxun.jms.dao;

/\*

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

\*/

**@Mapper**

**public interface** StudentAssignmentMapper**{**

public ArrayList<HomeworkView> getHomework(@Param("studentID") int studentID, @Param("homeworkID") int homeworkID);

**}**

#### 4.1.1.5 域模型层

查看作业模块用到域模型层中HomeworkView.java，并且它也是是公用域模型，在涉及到作业信息调用操作时，就会调用到该模型，学生查看作业模块域模型层列表如表4-5所示。

|  |  |
| --- | --- |
| 域模型 | 描述 |
| HomeworkView.java | 对数据库中的信息进行查询和显示 |

表 4-5查看作业模块域模型层列表

HomeworkView.java主要属性与方法：

**package com.gxun.jms.pojo;**

**public class HomeworkView {**

**private int studentID;**

**private int homeworkID;**

**private int questionID;**

**private String studentAnswer;**

**private float questionScore;**

**private int questionType;**

**private int courseID;**

**private String questionContent;**

**private String questionOptionA;**

**private String questionOptionB;**

**private String questionOptionC;**

**private String questionOptionD;**

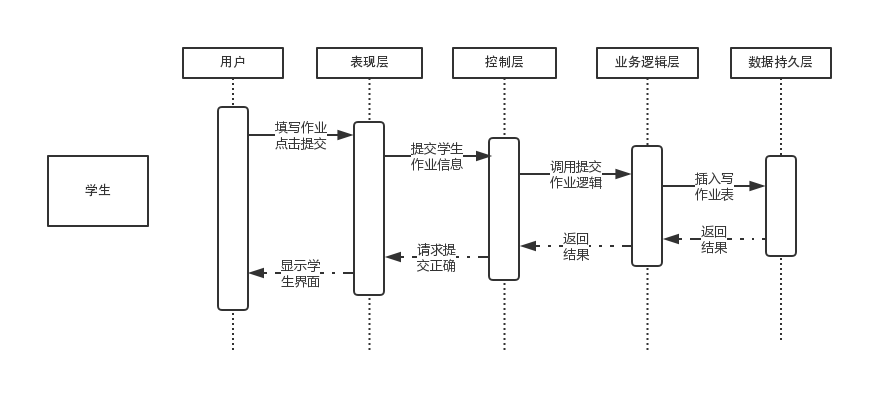
**private String questionRightAnswer;**

**private float Expr1;** /\*编写对上述各个属性的*get*与*set*方法\*/

}

### 4.1.2 学生填写、提交作业

在学生填写并提交作业时，系统内部的相应响应操作时序图如图4-2所示。

图 4-2填写、提交作业系统响应示意图

#### 4.1.2.1 表现层

学生模块的填写、提交作业的表现层主要完成学生完成作业的功能，填写作业时要求学生根据题目来将答案填写到相对应的区域中，确认提交后页面就会自动跳转回学生主界面。表现层对应的JSP页面列表见表4-6所示。

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 学生填写、提交作业页面 | WriteHomework.jsp | 学生进入页面以后填写完答案，点击提交。 |

表 4-6 填写、提交作业模块表现层JSP列表

#### 4.1.2.2 控制层

填写提交作业的控制层负责接收来自WriteHomework.jsp的学生输入，同时调用填写提交作业模块的业务逻辑接口，将学生输入的题目答案等关键信息保存。等到数据库操作完成，就将相应信息传到表现层，并决定显示页面。填写、提交作业模块控制层列表见表4-7所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 填写、提交作业 | StudentAssignmentController.java | SUCCESS | StudentIndex.jsp、—提交作业成功，显示回到学生主界面。 |

表 4-7填写、提交作业模块控制层列表

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

**package com.gxun.jms.controller;**

/\*\*

\* *@description* 实现填写、提交作业页面的控制

\*/

@GetMapping("/writeHomework") //写作业的控制层

public ModelAndView writeHomework(HttpServletRequest request, HttpServletResponse response){

}

@PostMapping("/submitHomework") //提交作业的控制层

public ModelAndView submitHomework(HttpServletRequest request, HttpServletResponse response) {

}

#### 4.1.2.3 业务逻辑层

学生填写、提交作业模块的业务逻辑层主要完成学生所填写的作业答案的数据获取，同时调用学生填写、提交作业模块的业务逻辑接口。比如说：在学生提交作业后，对其填写的作业内容进行获取，并将作业状态以及信息显示在表现层中。学生填写、提交作业业务逻辑层列表如表4-8所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 填写、提交作业 | StudentAssignmentService.java | 调用StudentAssignmentMapper.java | 返回给Action |

表 4-8填写、提交作业模块业务逻辑层列表

在学生填写、提交作业模块的业务逻辑层是调用了StudentAssignmentMapper.java接口。

StudentAssignmentService.java接口：

**package com.gxun.jms.service;**

@Service

public class StudentAssignmentService {

@Autowired

private StudentAssignmentMapper studentAssignmentMapper;

}

StudentAssignmentService.java调用数据持久层的StudentAssignmentMapper.java

StudentAssignmentMapper.java接口：

package com.gxun.jms.dao;

@Mapper

public interface StudentAssignmentMapper {

}

#### 4.1.2.4 数据持久层

填写、提交作业模块的数据持久层可以在用户输入填写完作业答案，点击保存以后获取到作业的信息。填写、提交作业模块数据持久层列表见表4-9所示

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 填写、提交作业 | StudentAssignmentMapper.java | 调用 WriteQuestion.java  StudentAssignment.java | 利用接口对用户信息进行操作 |

表 4-9填写、提交作业模块数据持久层列表

StudentAssignmentMapper.java定义了对用户信息操作的接口：

package com.gxun.jms.dao;

/\*

\* @description 数据持久层

\*/

**@Mapper**

**public interface StudentAssignmentMapper {**

**public void submitHomework(@Param("studentID") int studentID, @Param("homeworkID") int homeworkID, @Param("questionID") int question, @Param("answer") String answer);// 提交作业模块**

**public void confirmHomework(@Param("studentID") int studentID, @Param("homeworkID") int homeworkID, @Param("time") String time);//保存作业模块**

**}**

#### 4.1.2.5 域模型层

填写、提交作业模块用到域模型层中student.java和teacher.java，student.java和teacher.java是公用域模型，在涉及到用户信息操作时，就会调用到该模型，填写、提交作业模块域模型层列表如表4-10所示。

|  |  |
| --- | --- |
| 域模型 | 描述 |
| WriteQuestion.java  StudentAssignment.java | 对数据库中的信息进行增、删、改操作 |

表 4-10填写、提交作业模块域模型层列表

WriteQuestion.java主要属性和方法：

**package com.gxun.jms.pojo;**

**public class WriteQuestion {**

**private int studentID;**

**private int homeworkID;**

**private int questionID;**

**private String studentAnswer;**

**private float questionScore;**

/\*编写对上述各个属性的*get*与*set*方法\*/

}

StudentAssignment.java主要属性和方法：

**package com.gxun.jms.pojo;**

**public class StudentAssignment {**

**private int teacherID;**

**private String teacherName;**

**private int studentID;**

**private int courseID;**

**private String courseName;**

**private int homeworkID;**

**private String homeworkDescribe;**

**private String assignSubmissionTime;**

**private String assignDeadTime;**

**private float assignScore;**

**private int assignSubmissionFlag;**

**private int assignCheckFlag;**

**private String className;**

**private String studentName;**

/\*编写对上述各个属性的*get*与*set*方法\*/

}

### 4.1.3学生搜索作业

在搜索作业模块时，系统内部的相应响应操作时序图如图4-3所示。

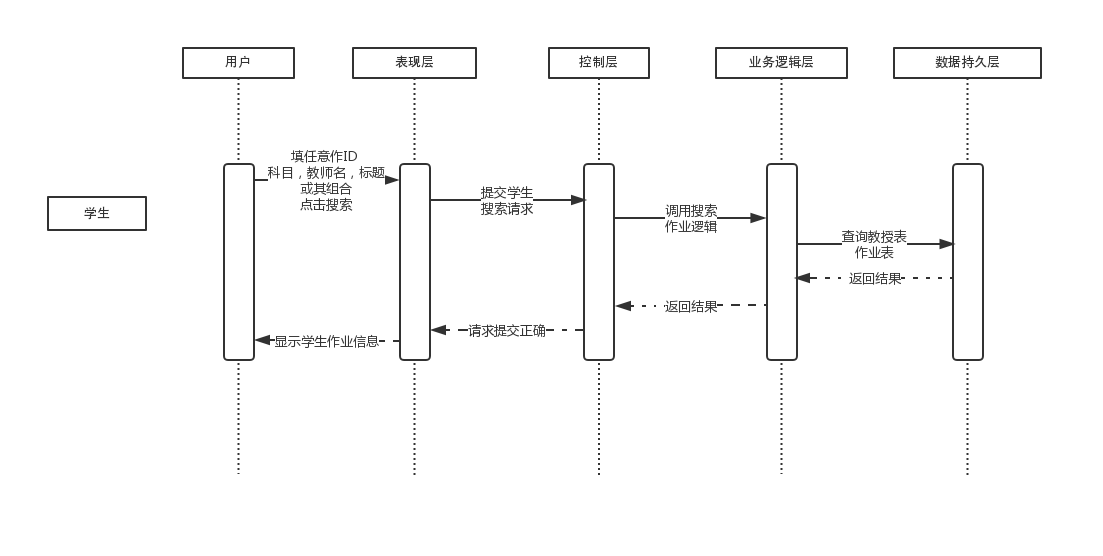


图 4-3搜索作业系统响应示意图

#### 4.1.2.1 表现层

搜索作业模块的表现层主要实现学生搜索作业的功能，学生在搜索框输入搜索的内容，点击搜索按钮后页面展示搜索结果。表现层对应的JSP页面列表见表4-11所示。

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 学生页面 | StudentIndex.jsp | 学生用户界面，显示该学生的作业情况。 |

表 4-11 搜索作业模块表现层JSP列表

#### 4.1.2.2 控制层

搜索作业模块的控制层负责接收来自StudentIndex.jsp的学生用户输入，同时调用搜索作业模块的业务逻辑接口，查询与学生用户输入的搜索信息匹配的内容。等到查询完成之后，将相应信息传到表现层，并在页面显示。搜索作业模块控制层列表见表4-12所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 搜索作业 | StudentAssignmentController.java | SUCCESS | StudentIndex.jsp—搜索成功，页面显示出搜索结果。 |

表 4-12搜索作业模块控制层列表

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

@PostMapping("/searchDoneStudentAssignment")

public ModelAndView searchDoneStudentAssignment(HttpServletRequest request, HttpServletResponse response)

{

IsNumberUtil isNumberUtil = new IsNumberUtil();

StudentAssignment studentAssignment = new StudentAssignment();

studentAssignment.setStudentID((int)request.getSession().getAttribute("studentID"));

Boolean flag = isNumberUtil.isInteger(request.getParameter("homeworkID"));

if(!request.getParameter("homeworkID").trim().equals("") && flag)

{

studentAssignment.setHomeworkID(Integer.parseInt(request.getParameter("homeworkID")));

}

if(!request.getParameter("homeworkDescribe").trim().equals(""))

{

studentAssignment.setHomeworkDescribe(request.getParameter("homeworkDescribe"));

}

if(!request.getParameter("courseName").trim().equals(""))

{

studentAssignment.setCourseName(request.getParameter("courseName"));

}

if(!request.getParameter("teacherName").trim().equals(""))

{

studentAssignment.setTeacherName(request.getParameter("teacherName"));

}

ArrayList<StudentAssignment> searchedDoneStudentAssignments = studentAssignmentService.searchDoneAssignment(studentAssignment);

request.getSession().setAttribute("doneStudentAssignments", searchedDoneStudentAssignments);

return new ModelAndView("StudentIndex");

}

#### 4.1.2.3 业务逻辑层

搜索作业模块的业务逻辑层主要完成对用户输入的搜索信息的获取，同时调用搜索作业模块的业务逻辑接口。比如说：在用户点击搜索后，对其输入的搜索信息进行获取，并将搜索结果显示在表现层中。搜索作业模块业务逻辑层列表如图4-13所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 搜索作业 | StudentAssignmentService.java | 调用StudentAssignmentMapper.java | 返回给Action |

表 4-13搜索作业模块业务逻辑层列表

在搜索作业模块的业务逻辑层是调用了公用 StudentAssignmentMapper.java接口，同时在实现该接口。

StudentAssignmentService.java接口：

**package com.gxun.jms.service;**

@Service

public class StudentAssignmentService{

@Autowired

private StudentAssignmentMapper studentAssignmentMapper;

}

StudentAssignmentService.java调用数据持久层的StudentAssignmentMapper.java

StudentAssignmentMapper.java接口：

package com.gxun.jms.dao;

@Mapper

public interface StudentAssignmentMapper{

}

#### 4.1.2.4 数据持久层

搜索作业模块的数据持久层可以在用户输入搜索条件点击搜索按钮以后获取到与搜索条件匹配的信息。登入模块数据持久层列表见表4-14所示

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 搜索作业 | StudentAssignmentMapper.java | 调用 HomeworkView.java | 利用接口对作业信息进行查询 |

表 4-14搜索作业模块数据持久层列表

StudentAssignmentMapper.java定义了对用户信息操作的接口：

package com.gxun.jms.dao;

**@Mapper**

**public interface StudentAssignmentMapper{**

**public ArrayList<StudentAssignment> searchDoneAssignment**

**(@Param("studentAssignment") StudentAssignment studentAssignment);**

**public ArrayList<StudentAssignment> searchUndoneAssignment**

**(@Param("studentAssignment")StudentAssignment studentAssignment);**

**}**

#### 4.1.2.5 域模型层

搜索作业模块用到域模型层中HomeworkView.java。HomeworkView.java是公用域模型，在涉及到作业信息操作时，就会调用到该模型，搜索作业模块域模型层列表如表4-15所示。

|  |  |
| --- | --- |
| 域模型 | 描述 |
| HomeworkView.java | 对数据库中的信息进行查和显示操作 |

表 4-15搜索作业模块域模型层列表

HomeworkView.java主要属性与方法：

**package com.gxun.jms.pojo;**

**public class Student {**

**private int studentID;**

**private int homeworkID;**

**private int questionID;**

**private String studentAnswer;**

**private float questionScore;**

**private int questionType;**

**private int courseID;**

**private String questionContent;**

**private String questionRightAnswer;**

**private float Expr1;** /\*编写对上述各个属性的*get*与*set*方法\*/

}

## 4.2、教师模块

教师模块分为：查看作业模块，搜索作业模块和填布置作业模块和批改作业模块。

### 4.2.1教师搜索作业

在搜索作业模块时，系统内部的相应响应操作时序图如图4-4所示。

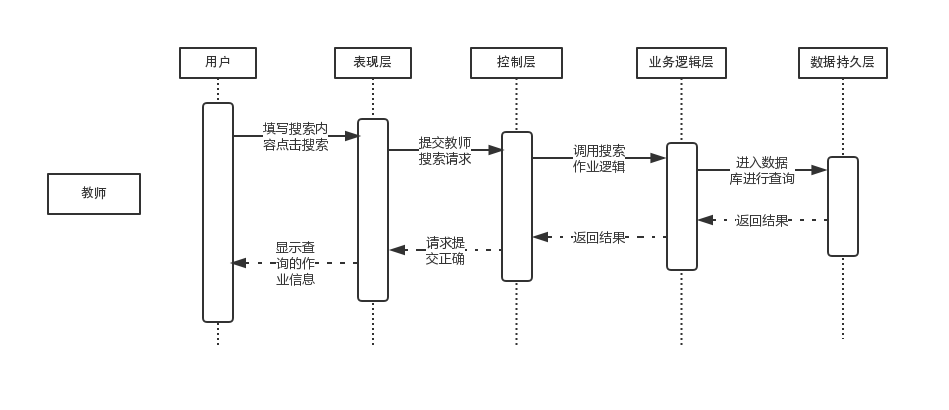


图 4-4搜索作业系统响应示意图

#### 4.2.1.1 表现层

搜索作业模块的表现层主要完成查询信息的显示。表现层对应的JSP页面列表见表4-16所示。

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 作业 | TeacherIndex.jsp和TeacherIndex1.jsp。 | 用户（教师）搜索已经批改的作业，和未批改的作业。 |

表 4-16 搜索作业模块表现层JSP列表

#### 4.2.1.2 控制层

搜索作业模块的控制层负责接收来自TeacherIndex.jsp或者TeacherIndex1.jsp的用户输入，同时调用搜索作业模块的业务逻辑接口，将相应信息传到表现层，并决定显示页面。搜索作业模块控制层列表见表4-17所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 搜索 | TeacherAssignmentController.java | SUCCESS | TeacherIndex.jsp、TeacherIndex1.jsp—搜索成功，显示对应的作业。 |

表 4-17搜索作业模块控制层列表

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

**package com.gxun.jms.controller;**

/\*\*

\* *@description* 实现登录页面的控制

\*/

@RestController（控制器）

public class TeacherAssignmentController{

/\*对用户输入数据的获取\*/

/\* 对数据进行简单的处理\*/

}

#### 4.2.1.3 业务逻辑层

搜索作业模块的业务逻辑层主要完成对用户所需作业信息的获取，同时调用查看模块的业务逻辑接口。搜索作业模块业务逻辑层列表如图4-18所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 搜索 | TeacherAssignmentService.java | 调用TeacherAssignmentMapper.java | 返回给Action |

表 4-18搜索作业模块业务逻辑层列表

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

TeacherAssignmentService.java接口：

package com.gxun.jms.service;

@Service

public class TeacherAssignmentService{

@Autowired  
private TeacherAssignmentMapper teacherAssignmentMapper;

}

TeacherAssignmentService.java调用数据持久层的TeacherAssignmentMapper .java

TeacherAssignmentMapper .java接口：

package com.gxun.jms.dao;

@Mapper

public interface TeacherAssignmentMapper {

}

#### 4.2.1.4 数据持久层

搜索作业模块的数据持久层可以在选择查看作业类型后系统获取相应的作业数据，搜索作业模块数据持久层列表见表4-19所示

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 查看 | TeacherAssignmentMapper .java | HomeworkView.java | 利用接口对用户信息进行操作 |

表 4-19搜索作业模块数据持久层列表

TeacherAssignmentMapper .java定义了对用户信息操作的接口：

package com.gxun.jms.dao;

/\*

\* @description 信息持久层接口定义，包括查询。

\*/

**@Mapper**

**public interface** TeacherAssignmentMapper **{**

**public** ArrayList<StudentAssignment> searchDoneTeacherAssignment(StudentAssignment teacherAssignment)

**public** ArrayList<StudentAssignment> searchUndoneTeacherAssignment(StudentAssignment teacherAssignment)

**}**

#### 4.2.1.5 域模型层

教师搜索作业模块用到域模型层中HomeworkView.java，并且它也是是公用域模型，在涉及到作业信息调用操作时，就会调用到该模型，教师搜索作业模块域模型层列表如表4-20所示。

|  |  |
| --- | --- |
| 域模型 | 描述 |
| HomeworkView.java | 对数据库中的信息进行显示 |

表 4-20教师搜索作业模块域模型层列表

HomeworkView.java主要属性与方法：

**package com.gxun.jms.pojo;**

**public class HomeworkView {**

**private int studentID;**

**private int homeworkID;**

**private int questionID;**

**private String studentAnswer;**

**private float questionScore;**

**private int questionType;**

**private int courseID;**

**private String questionContent;**

**private String questionOptionA;**

**private String questionOptionB;**

**private String questionOptionC;**

**private String questionOptionD;**

**private String questionRightAnswer;**

**private float Expr1;** /\*编写对上述各个属性的*get*与*set*方法\*/

}

### 4.2.2教师布置作业

在布置作业模块时，系统内部的相应响应操作时序图如图4-5所示。

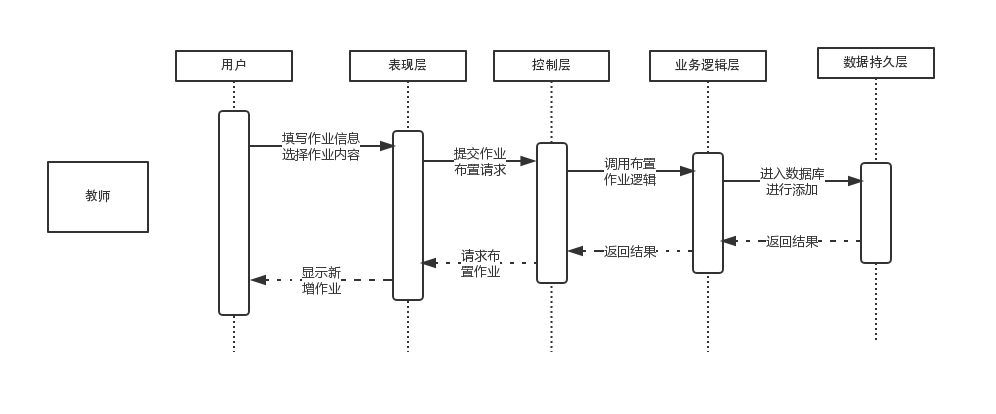


图 4-5教师布置作业系统响应示意图

#### 4.2.2.1 表现层

教师布置作业模块的表现层主要实现教师布置作业的功能，教师用户在布置作业页面内输入要布置的作业的内容以及选择作业的题目，点击布置作业按钮后页面给出响应消息，提示布置成功或失败的提示。表现层对应的JSP页面列表见表4-21所示。

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 布置作业页面 | AssignHomeworkStepOne.jsp以及AssignHomeworkStepTwo.jsp | 布置作业界面，实现教师布置作业的功能。 |

表 4-21 布置作业模块表现层JSP列表

#### 4.2.2.2 控制层

教师布置作业模块的控制层负责接收来自AssignHomeworkStepOne.jsp以及AssignHomeworkStepTwo.jsp的输入，同时调用布置作业模块的业务逻辑接口。布置作业模块控制层列表见表4-22所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 布置作业 | TeacherAssignmentController.java | SUCCESS | AssignHomeworkStepOne.jsp、AssignHomeworkStepTwo.jsp--显示布置成功的提示。 |
| ERROR | AssignHomeworkStepOne.jsp、AssignHomeworkStepTwo.jsp--显示布置失败的提示。 |

表 4-22布置作业模块控制层列表

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

@PostMapping("/AssignHomeworkStepOne")

public ModelAndView assignHomeworkStepOne(HttpServletRequest request, HttpServletResponse response)

{

}

@PostMapping("/addQuestion")

public ModelAndView addQuestion(HttpServletRequest request, HttpServletResponse response)

{

}

#### 4.2.2.3 业务逻辑层

布置作业模块的业务逻辑层主要完成对教师要布置的作业的信息的获取，同时调用布置作业模块的业务逻辑接口。比如说：在教师用户点击布置作业后，对其勾选的题目信息进行获取。布置作业模块业务逻辑层列表如图4-23所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 布置作业 | TeacherAssignmentService.java | 调用TeacherAssignmentMapper.java | 返回给Action |

表 4-23布置作业模块业务逻辑层列表

在布置作业模块的业务逻辑层是调用了公用 TeacherAssignmentMapper.java接口，同时在实现该接口。

TeacherAssignmentService.java接口：

**package com.gxun.jms.service;**

@Service

public class TeacherAssignmentService{

@Autowired

private TeacherAssignmentMapper teacherAssignmentMapper;

}

TeacherAssignmentService.java调用数据持久层的TeacherAssignmentMapper.java

TeacherAssignmentService.java接口：

package com.gxun.jms.dao;

@Mapper

public interface TeacherAssignmentMapper{

}

#### 4.2.2.4 数据持久层

布置作业模块的数据持久层可以在教师输入布置作业的信息和勾选题目接着点击布置作业以后获取到与教师要布置的作业的信息。布置作业模块数据持久层列表见表4-24所示

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 布置作业 | TeacherAssignmentMapper.java | 调用 Assign.java | 利用接口对布置作业信息进行插入 |

表 4-24布置作业模块数据持久层列表

TeacherAssignmentMapper.java定义了对用户信息操作的接口：

package com.gxun.jms.dao;

**@Mapper**

**public interface StudentAssignmentMapper{**

**public void insertHomework(@Param("homework") Homework homework);**

**public void insertAssign(@Param("assignment") Assign assign);**

**public Homework getHomework(@Param("courseID") int courseID, @Param("homeworkName") String homeworkName);**

**public ArrayList<Question> getQuestion(@Param("question") Question question);**

**public void insertHomeworkContent(@Param("homeworkID") int homeworkID, @Param("questionNum") int questionNum);**

**}**

#### 4.2.2.5 域模型层

布置作业模块用到域模型层中Assign.java。Assign.java是公用域模型，在涉及到布置作业操作时，就会调用到该模型，布置作业模块域模型层列表如表4-25所示。

|  |  |
| --- | --- |
| 域模型 | 描述 |
| Assign.java | 对数据库中的信息进行增、删、改、查操作 |

表 4-25布置作业模块域模型层列表

Assign.java主要属性与方法：

**package com.gxun.jms.pojo;**

**public class Assign {**

**private int teacherID;**

**private int studentID;**

**private int homeworkID;**

**private float assignScore;**

**private int assignExampleFlag;**

**private int assignSubmissionFlag;**

**private int assignCheckFlag;**

**private String assignSubmissionTime;**

**private String assignDeadTime;**

/\*编写对上述各个属性的*get*与*set*方法\*/

}

### 4.2.3教师批改作业

在批改作业模块时，系统内部的相应响应操作时序图如图4-6所示。

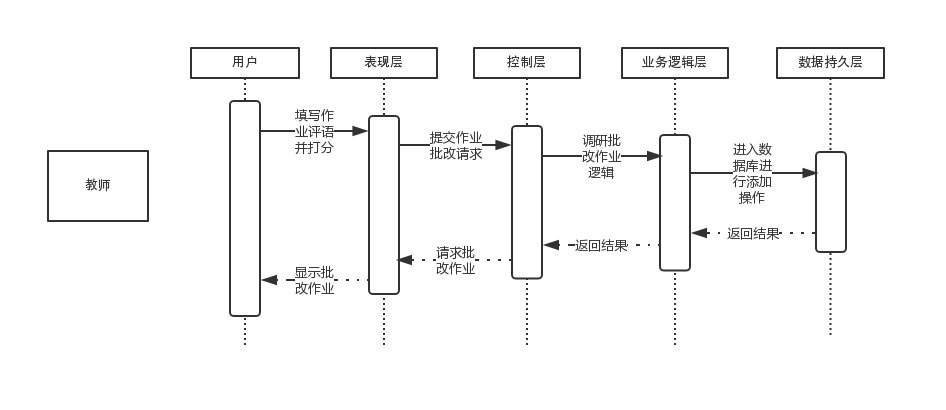


图 4-6批改作业系统响应示意图

#### 4.2.3.1 表现层

批改作业模块的表现层主要完成查询信息的显示。表现层对应的JSP页面列表见表4-26所示。

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 作业 | CheckWork.jsp | 用户（教师）批改作业。 |

表 4-26 批改作业模块表现层JSP列表

#### 4.2.3.2 控制层

批改作业模块的控制层负责接收来自CheckWork.jsp的用户输入，同时调用批改模块的业务逻辑接口，将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4-27所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 确认批改 | TeacherAssignmentController.java | SUCCESS | TeacherIndex.jsp—退出批改界面，返回到教师主界面 |

表 4-27批改作业模块控制层列表

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

**package com.gxun.jms.controller;**

@RestController（控制器）

public class TeacherAssignmentController{

/\*对用户输入数据的获取\*/

/\* 对数据进行简单的处理\*/

}

#### 4.2.3.3 业务逻辑层

批改作业模块的业务逻辑层主要完成对用户所需作业信息的获取和录入，同时调用查看模块的业务逻辑接口。批改作业模块业务逻辑层列表如图4-28所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 确认批改 | TeacherAssignmentService.java | 调用TeacherAssignmentMapper.java | 返回给Action |

表 4-28批改作业模块业务逻辑层列表

在批改模块的业务逻辑层是调用了公用的TeacherAssignmentMapper .java接口，同时在实现该接口。

TeacherAssignmentService.java接口：

package com.gxun.jms.service;

@Service

public class TeacherAssignmentService{

@Autowired  
private TeacherAssignmentMapper teacherAssignmentMapper;

}

TeacherAssignmentService.java调用数据持久层的TeacherAssignmentMapper .java

TeacherAssignmentMapper .java接口：

package com.gxun.jms.dao;

@Mapper

public interface TeacherAssignmentMapper {

}

#### 4.2.3.4 数据持久层

批改作业模块的数据持久层可以在选择查看作业类型后系统获取相应的作业数据，批改模块数据持久层列表见表4-29所示

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 查看 | TeacherAssignmentMapper .java | 待定（ps：） | 利用接口对作业批改信息进行操作 |

表 4-29批改作业模块数据持久层列表

TeacherAssignmentMapper .java定义了对用户信息操作的接口：

package com.gxun.jms.dao;

/\*

\* @description 信息持久层接口定义，包括查询。

\*/

**@Mapper**

**public interface** TeacherAssignmentMapper **{**

public void insertHomeworkContent(@Param("homeworkID") int homeworkID, @Param("questionNum") int questionNum);

**}**

#### 4.2.3.5 域模型层

教师批改作业模块用到域模型层中HomeworkView.java，并且它也是是公用域模型，在涉及到作业信息调用操作时，就会调用到该模型，教师批改作业模块域模型层列表如表4-30所示。

|  |  |
| --- | --- |
| 域模型 | 描述 |
| HomeworkView.java | 对数据库中的信息进行显示 |

表 4-30批改作业模块域模型层列表

HomeworkView.java主要属性与方法：

**package com.gxun.jms.pojo;**

**public class HomeworkView {**

**private int studentID;**

**private int homeworkID;**

**private int questionID;**

**private String studentAnswer;**

**private float questionScore;**

**private int questionType;**

**private int courseID;**

**private String questionContent;**

**private String questionOptionA;**

**private String questionOptionB;**

**private String questionOptionC;**

**private String questionOptionD;**

**private String questionRightAnswer;**

**private float Expr1;** /\*编写对上述各个属性的*get*与*set*方法\*/

}

### 4.2.4教师查看作业

在教师查看作业模块时，系统内部的相应响应操作时序图如图所示。

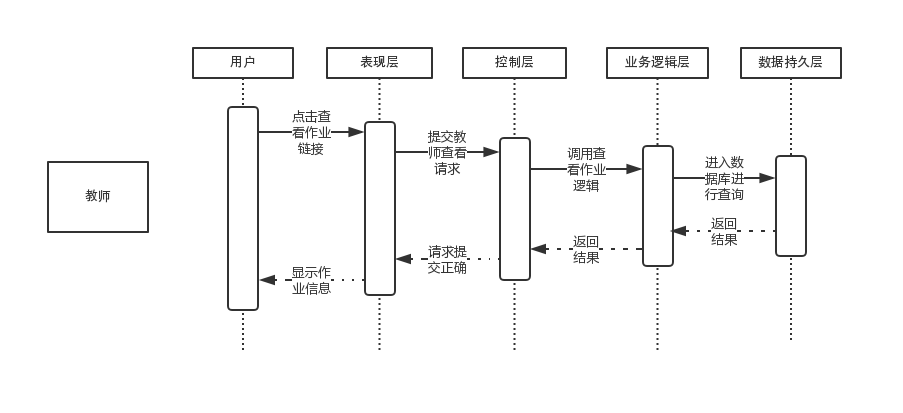


图 4-7查看作业系统响应示意图

#### 4.2.4.1 表现层

教师查看作业模块的表现层主要完成教师查看学生的作业的功能，学生作业页面要显示出学生所填写的作业的详细情况，查看完毕后点击返回会直接跳转回到教师主界面。表现层对应的JSP页面列表如下所示。表现层对应的JSP页面列表见表4-31所示。

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 登录页面 | CheckHomework.jsp | 教师进入页面查看学生提交的作业 |

表 4-31查看作业模块表现层JSP列表

#### 4.2.4.2 控制层

教师查看作业模块的控制层负责接收来自WriteHomework.jsp的学生输入，同时调用教师查看作业模块的业务逻辑接口，将教师在查看作业页面上面的操作进行判定。等到判定完成之后，将相应信息传到表现层，并决定显示页面。教师查看作业模块控制层列表见表4-32所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 确认批改 | TeacherAssignmentController.java | SUCCESS | TeacherIndex.jsp—退出查看界面，返回到教师主界面 |

表 4-32查看作业模块控制层列表

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

**package com.gxun.jms.controller;**

@RestController（控制器）

public class TeacherAssignmentController{

/\*对用户输入数据的获取\*/

/\* 对数据进行简单的处理\*/

}

#### 4.2.4.3 业务逻辑层

查看作业模块的业务逻辑层主要完成对教师操作的获取和保存，同时调用查看模块的业务逻辑接口。查看作业模块业务逻辑层列表如图4-33所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 确认批改 | TeacherAssignmentService.java | 调用TeacherAssignmentMapper.java | 返回给Action |

表 4-33查看作业模块业务逻辑层列表

在查看作业模块的业务逻辑层是调用了公用的TeacherAssignmentMapper .java接口，同时在实现该接口。

TeacherAssignmentService.java接口：

package com.gxun.jms.service;

@Service

public class TeacherAssignmentService{

@Autowired  
private TeacherAssignmentMapper teacherAssignmentMapper;

}

TeacherAssignmentService.java调用数据持久层的TeacherAssignmentMapper .java

TeacherAssignmentMapper .java接口：

package com.gxun.jms.dao;

@Mapper

public interface TeacherAssignmentMapper {

}

#### 4.2.4.4 数据持久层

查看作业模块的数据持久层可以在选择查看作业类型后系统获取相应的作业数据，查看作业模块数据持久层列表见表4-34所示

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 查看 | TeacherAssignmentMapper .java | HomeworkView.java | 利用接口对作业查看进行操作 |

表 4-34登入模块数据持久层列表

TeacherAssignmentMapper .java定义了对用户信息操作的接口：

package com.gxun.jms.dao;

/\*

\* @description 信息持久层接口定义。

\*/

**@Mapper**

**public interface** TeacherAssignmentMapper **{**

public void insertHomeworkContent(@Param("homeworkID") int homeworkID, @Param("questionNum") int questionNum);

**}**

#### 4.2.4.5 域模型层

教师查看作业模块用到域模型层中HomeworkView.java，并且它也是是公用域模型，在涉及到作业信息调用操作时，就会调用到该模型，教师查看作业模块域模型层列表如表4-35所示。

|  |  |
| --- | --- |
| 域模型 | 描述 |
| HomeworkView.java | 对数据库中的信息进行显示 |

表 4-35登入模块域模型层列表

HomeworkView.java主要属性与方法：

**package com.gxun.jms.pojo;**

**public class HomeworkView {**

**private int studentID;**

**private int homeworkID;**

**private int questionID;**

**private String studentAnswer;**

**private float questionScore;**

**private int questionType;**

**private int courseID;**

**private String questionContent;**

**private String questionOptionA;**

**private String questionOptionB;**

**private String questionOptionC;**

**private String questionOptionD;**

**private String questionRightAnswer;**

**private float Expr1;** /\*编写对上述各个属性的*get*与*set*方法\*/

}

# 管理端模块详细设计

管理端模块只包括一个管理员模块。

## 5.1管理员添加账号

在管理员添加账号模块时，系统内部的相应响应操作时序图如图5-1所示。

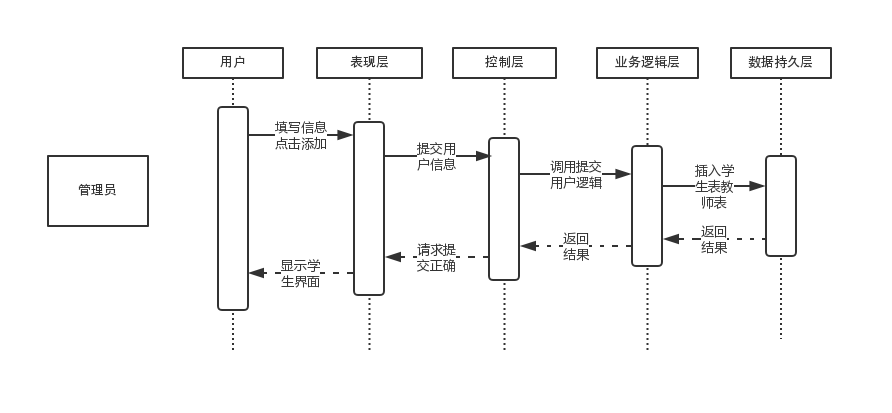


图 5-1添加账号系统响应示意图

### 5.1.1 表现层

管理员添加账号模块的表现层主要完成管理员向系统内部添加学生和教师的功能，添加角色时要求管理员输入学生或者老师的基本信息，确认后页面给出响应消息，提示添加成功。表现层对应的JSP页面列表见表5-1所示。

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 添加学生和教师页面 | AddStudent.jsp  AddTeacher.jsp | 管理员分别在这两个页面上面添加账号信息。 |

表 5-1 添加账号模块表现层JSP列表

### 5.1.2 控制层

管理员添加账号模块的控制层负责接收来自AddStudent.jsp和AddTeacher.jsp的管理员输入，同时调用管理员添加账号模块的业务逻辑接口，将管理员输入的账号关键信息进行保存。然后将相应信息传到表现层显示出来。管理员添加账号模块控制层列表见表5-2所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 管理员添加账号 | AdminController.java | SUCCESS | AddStudent.jsp  AddTeacher.jsp—添加成功，在对应的页面显示出来。 |

表 5-2管理员添加账号模块控制层列表

在控制层中AdminController.java的描述如下所示：

**package com.gxun.jms.controller;**

/\*\*

\* *@description* 实现对学生和教师的添加

\*/

@PostMapping("/addStudent")

public ModelAndView AddStudent(HttpServletRequest request, HttpServletResponse response){

}

@PostMapping("/addTeacher")

public ModelAndView addTeacher(HttpServletRequest request, HttpServletResponse response){

}

### 5.1.3 业务逻辑层

管理员添加账号模块的业务逻辑层主要完成对管理员添加的账号信息的获取，同时调用管理员添加账号模块的业务逻辑接口。比如说：在管理员添加账号后，对该账号的关键信息进行获取，并将账号等信息显示在表现层中。管理员添加账号模块业务逻辑层列表如图5-3所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 管理员添加账号 | AdminService.java | 调用AdminMapper.java | 返回给Action |

表 5-3管理员添加账号模块业务逻辑层列表

在管理员添加账号模块的业务逻辑层是调用了AdminMapper.java接口，同时实现该接口。

AdminService.java接口：

**package com.gxun.jms.service;**

@Service

public class AdminService {

@Autowired

private AdminMapper adminMapper;

}

AdminService.java调用数据持久层的AdminMapper.java

AdminMapper.java接口：

package com.gxun.jms.dao;

@Mapper

public interface AdminMapper {

}

### 5.1.4 数据持久层

管理员添加账号模块的数据持久层可以在管理员填写了新添加的账号信息的数据，点击添加以后获取到新增的账号信息。管理员添加账号模块数据持久层列表见表5-4所示

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 登录 | AdminMapper.java | 调用 Teacher.java  和Student.java | 利用接口对用户信息增加操作 |

表 5-4登录模块数据持久层列表

AdminMapper.java定义了对用户信息操作的接口：

package com.gxun.jms.dao;

/\*

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

\*/

**@Mapper**

**public interface AdminMapper {**

**public void insertStudent(@Param("student") Student student);**

**public void insertStudentTwo(@Param("student") Student student);**

**public void insertTeacher(@Param("teacher")Teacher teacher);**

**public void insertTeacherTwo(@Param("teacher")Teacher teacher);**

**}**

### 5.1.5 域模型层

管理员添加账号模块用到域模型层中student.java和teacher.java，student.java和teacher.java两个是公用域模型，在涉及到用户信息操作时，就会调用到该模型，管理员添加账号模块域模型层列表如表5-5所示。

|  |  |
| --- | --- |
| 域模型 | 描述 |
| student.java和teacher.java | 对数据库中的信息进行增、删、改、查操作 |

表 5-5登录模块域模型层列表

Student.java主要属性与方法：

**package com.gxun.jms.pojo;**

/\*

\* *@description* student表的信息记录

\*/

**public class Student {**

**private String studentName;**

**private int studentID;**

**private String studentPassword;**

**private int classID;**

**private int collegeID;**

**private int activeFlag;**

**private String studentLoginID;**

/\*编写对上述各个属性的*get*与*set*方法\*/

}

Teacher.java主要属性与方法：

**package com.gxun.jms.pojo;**

/\*

\* *@description* Teacher表的信息记录

\*/

**public class Teacher {**

**private String teacherName;**

**private int teacherID;**

**private String teacherPassword;**

**private int collegeID;**

**private int activeFlag;**

**private String teacherLoginID;**

/\*编写对上述各个属性的*get*与*set*方法\*/

}

## 5.2管理员搜索账号

在管理员搜索模块时，系统内部的相应响应操作时序图如图5-2所示。

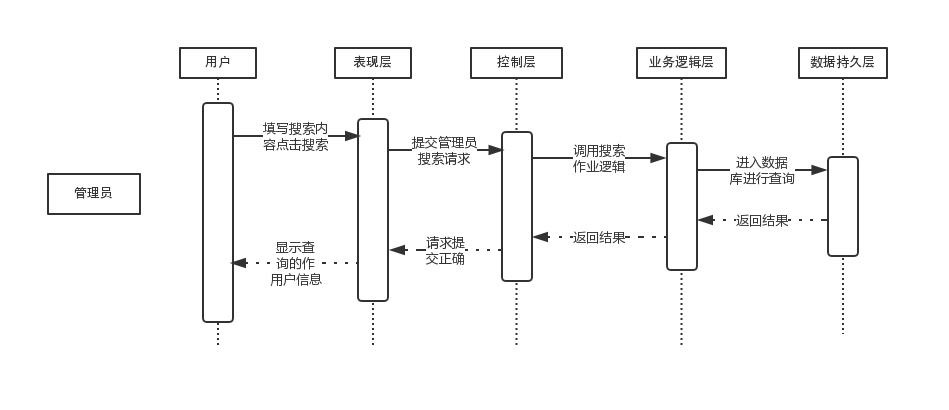


图 5-2管理员搜索时序图

### 5.2.1 表现层

管理员搜索模块的表现层主要实现管理员搜索学生和教师的信息的功能，管理员在搜索框输入搜索的内容，点击搜索按钮后页面展示搜索结果。表现层对应的JSP页面列表见表5-6所示。

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 管理员页面 | AdminIndex.jsp、  AdminIndex1.jsp | 管理员界面，显示学生和教师的信息。 |

表 5-6 管理员搜索模块表现层JSP列表

### 5.2.2 控制层

管理员搜索模块的控制层负责接收来自AdminIndex.jsp和AdminIndex1.jsp的管理员输入，同时调用搜索模块的业务逻辑接口，查询与管理员输入的搜索信息匹配的内容。等到查询完成之后，将相应信息传到表现层，并在页面显示。搜索模块控制层列表见表5-7所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 搜索学生或教师 | AdminController.java | SUCCESS | AdminIndex.jsp、  AdminIndex1.jsp—搜索成功，页面显示出搜索结果。 |
| ERROR | AdminIndex.jsp、  AdminIndex1.jsp--显示搜索失败的提示。 |

表 5-7管理员搜索模块控制层列表

在控制层中AdminController .java的描述如下所示：

@PostMapping("/searchStudent")

public ModelAndView

searchStudent(HttpServletRequest request, HttpServletResponse response)

{

}

@PostMapping("/searchTeacher")

public ModelAndView

searchTeacher(HttpServletRequest request, HttpServletResponse response)

{

}

### 5.2.3 业务逻辑层

管理员搜索模块的业务逻辑层主要完成对管理员输入的搜索信息的获取，同时调用搜索模块的业务逻辑接口。比如说：在管理员点击搜索后，对其输入的搜索信息进行获取，并将搜索结果显示在表现层中。搜索模块业务逻辑层列表如图5-8所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 搜索学生或教师 | AdminService.java | 调用AdminMapper.java | 返回给Action |

表 5-8管理员搜索模块业务逻辑层列表

在搜索模块的业务逻辑层是调用了公用 AdminMapper.java接口，同时在实现该接口。

AdminService.java接口：

**package com.gxun.jms.service;**

@Service

public class AdminService{

@Autowired

private AdminMapper adminMapper;

}

AdminService.java调用数据持久层的AdminMapper.java

AdminMapper.java接口：

package com.gxun.jms.dao;

@Mapper

public interface AdminMapper{

}

### 5.2.4 数据持久层

管理员搜索模块的数据持久层可以在管理员输入搜索条件点击搜索按钮以后获取到与搜索条件匹配的信息。搜索模块数据持久层列表见表5-9所示

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 搜索学生或教师 | AdminMapper.java | 调用 StudentView.java、TeacherView.java | 利用接口对学生和教师信息进行查询 |

表 5-9管理员搜索模块数据持久层列表

AdminMapper.java定义了对用户信息操作的接口：

package com.gxun.jms.dao;

**@Mapper**

**public interface AdminMapper{**

**public ArrayList<StudentView>**

**getStudent(@Param("student") StudentView student);**

**public ArrayList<TeacherView>**

**getTeacher(@Param("teacher") TeacherView teacher);**

**}**

### 5.2.5 域模型层

管理员搜索模块用到域模型层中StudentView.java、TeacherView.java。

StudentView.java、TeacherView.java是公用域模型，在涉及到学生和教师信息操作时，就会调用到该模型，管理员搜索模块域模型层列表如表5-10所示。

|  |  |
| --- | --- |
| 域模型 | 描述 |
| StudentView.java、TeacherView.java | 对数据库中的信息进行增、删、改、查操作 |

表 5-10管理员搜索模块域模型层列表

StudentView.java主要属性与方法：

**package com.gxun.jms.pojo;**

**public class StudentView{**

**private String studentName;**

**private int studentID;**

**private int classID;**

**private int collegeID;**

**private String className;**

**private String collegeName;**

**private int activeFlag;**

/\*编写对上述各个属性的*get*与*set*方法\*/

}

TeacherView.java主要属性与方法：

**package com.gxun.jms.pojo;**

**public class** TeacherView**{**

**private String teacherName;**

**private int teacherID;**

**private int collegeID;**

**private String collegeName;**

**private int activeFlag;**

/\*编写对上述各个属性的*get*与*set*方法\*/

}

## 5.3管理员删除信息

在用户删除模块时，系统内部的相应响应操作时序图如图5-3所示。

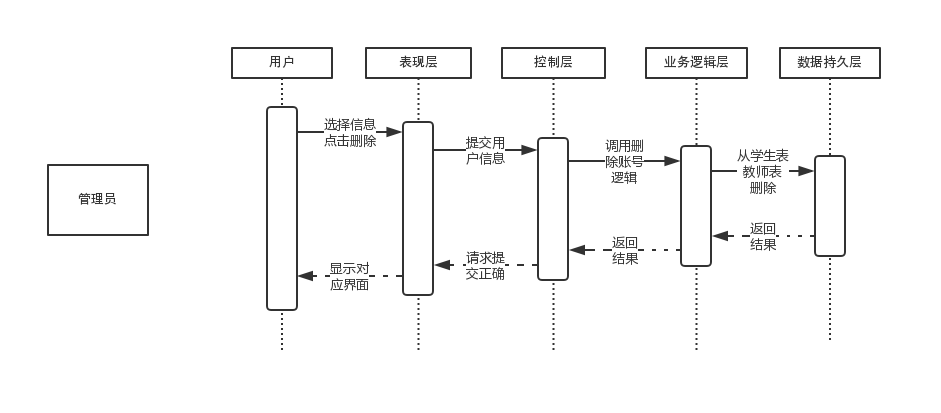


图 5-3系统响应示意图

### 5.3.1 表现层

删除用户模块的表现层主要完成用户的显示。表现层对应的JSP页面列表见表5-11所示。

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 管理员主界面 | AdminIndex.jsp和AdminIndex.jsp | 用户（管理员）删除用户 |

表 5-11 删除模块表现层JSP列表

### 5.3.2 控制层

删除模块的控制层负责接收来自AdminIndex.jsp和AdminIndex1.jsp的用户输入，同时调用管理员功能模块的业务逻辑接口，将相应信息传到表现层，并决定显示页面。用户删除模块控制层列表见表5-12所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 删除 | AdminController.java | SUCCESS | AdminIndex.jsp或AdminIndex.jsp，显示对应的页面。 |

表 5-12登入模块控制层列表

在控制层中AdminController.java的描述如下所示：

**package com.gxun.jms.controller;**

/\*\*

\* *@description* 实现登录页面的控制

\*/

@RestController（控制器）

public class AdminController{

/\*对用户输入数据的获取\*/

/\* 对数据进行简单的处理\*/

}

### 5.3.3 业务逻辑层

删除用户模块的业务逻辑层主要完成对用户账号的删除，同时调用删除模块的业务逻辑接口。删除模块业务逻辑层列表如图5-13所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 删除 | AdminService.java | 调用AdminMapper.java | 返回给Action |

表 5-13删除用户模块业务逻辑层列表

在用户删除模块的业务逻辑层是调用了公用的AdminMapper.java接口，同时在实现该接口。

AdminService.java接口：

package com.gxun.jms.service;

@Service

public class AdminService{

@Autowired  
private AdminMapper adminMapper;

}

AdminService.java调用数据持久层的AdminMapper.java

AdminMapper.java接口：

package com.gxun.jms.dao;

@Mapper

public interface AdminMapper{

}

### 5.3.4 数据持久层

用户删除模块的数据持久层可以在选择删除后删除用户信息，用户删除模块数据持久层列表见表5-14所示

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 删除 | AdminMapper.java | Student.java  Teacher.java | 利用接口对用户信息进行操作 |

表 5-14登入模块数据持久层列表

AdminMapper.java定义了对用户信息操作的接口：

package com.gxun.jms.dao;

/\*

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

\*/

**@Mapper**

**public interface** AdminMapper**{**

public void deleteStudent(@Param("studentID") int studentID);  
  
public void deleteTeacher(@Param("teacherID") int teacherID);

**}**

### 5.3.5 域模型层

管理员删除账号模块用到域模型层中student.java和teacher.java，student.java和teacher.java两个是公用域模型，在涉及到用户信息操作时，就会调用到该模型，管理员删除账号模块域模型层列表如表5-15所示。

|  |  |
| --- | --- |
| 域模型 | 描述 |
| student.java和teacher.java | 对数据库中的信息进行删除操作 |

表 5-15删除账号模块域模型层列表

Student.java主要属性与方法：

**package com.gxun.jms.pojo;**

/\*

\* *@description* student表的信息记录

\*/

**public class Student {**

**private String studentName;**

**private int studentID;**

**private String studentPassword;**

**private int classID;**

**private int collegeID;**

**private int activeFlag;**

**private String studentLoginID;**

/\*编写对上述各个属性的*get*与*set*方法\*/

}

Teacher.java主要属性与方法：

**package com.gxun.jms.pojo;**

/\*

\* *@description* Teacher表的信息记录

\*/

**public class Teacher {**

**private String teacherName;**

**private int teacherID;**

**private String teacherPassword;**

**private int collegeID;**

**private int activeFlag;**

**private String teacherLoginID;**

/\*编写对上述各个属性的*get*与*set*方法\*/

}

# 6、公共端模块详细设计

系统的角色共有管理员、教师以及学生。而这三个角色都包括有登录系统模块。所以，公共端就有登录这个模块。

## 6.1、登录模块

在登录模块时，系统内部的相应响应操作时序图如图6-1所示。

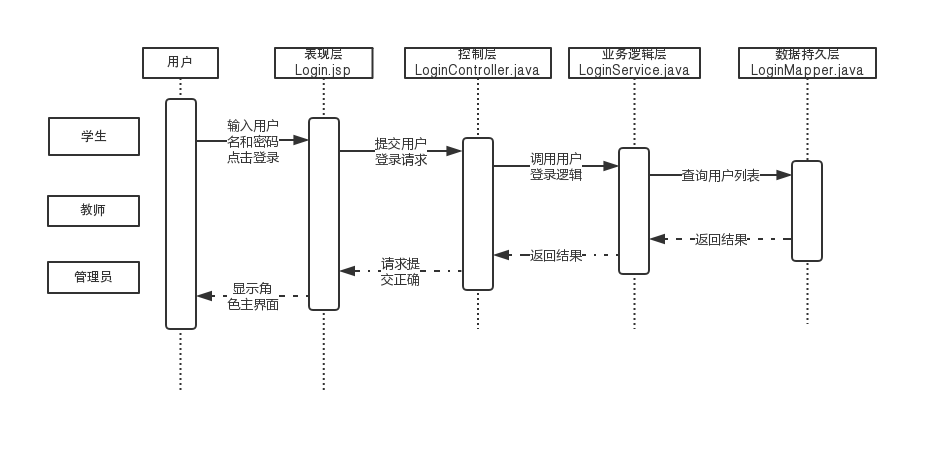


图 6-1登录模块系统响应示意图

### 6.1.1 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，显示登录成功或失败的提示。表现层对应的JSP页面列表见表6-1所示。

|  |  |  |
| --- | --- | --- |
| 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。 |

表 6-1 登录模块表现层JSP列表

Login.jsp的流程图如图6-2所示。

进入游首页

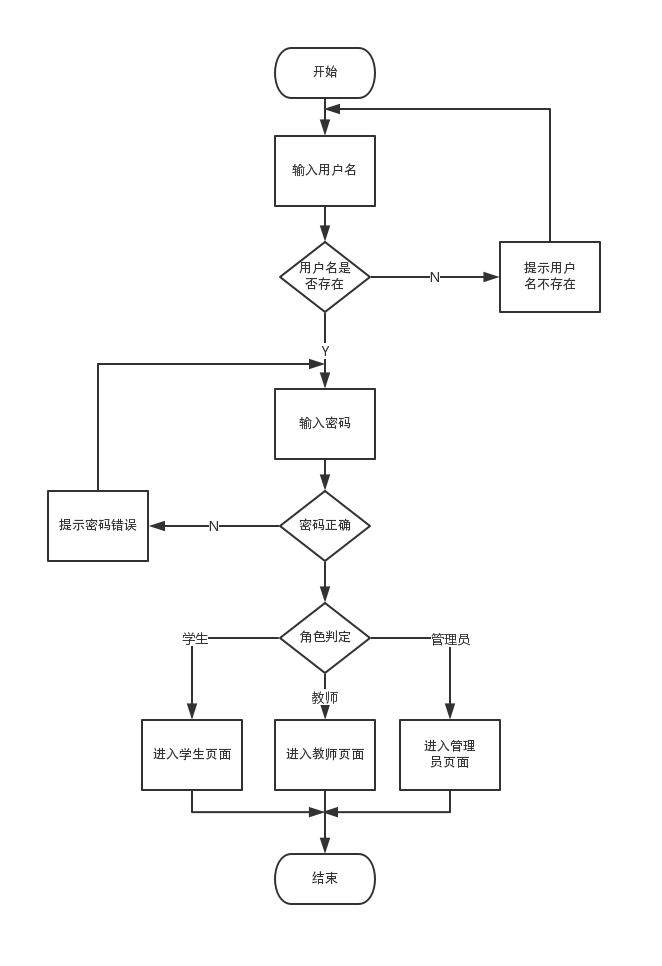


图 6-2 登录模块详细设计流程图

### 6.1.2 控制层

登录模块的控制层负责接收来自Login.jsp的用户输入，同时调用登入模块的业务逻辑接口，将用户输入的账号和密码等关键信息进行判定。等到判定完成之后，将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6-2所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 登录 | LoginController.java | SUCCESS | StudentIndex.jsp、TeacherIndex.jsp、AdminIndex.jsp—登录成功，显示对于的页面。 |
| ERROR | login.jsp--显示登录失败的提示。 |

表 6-2登入模块控制层列表

在控制层中LoginController.java的描述如下所示：

**package com.gxun.jms.controller;**

/\*\*

\* *@description* 实现登录页面的控制

\*/

@RestController（控制器）

public class LoginController {

/\*实现登录角色信息的获取和输出方法\*/

/\*对登录用户进行角色判断\*/

if(userType.equals("student")) {}

else if(userType.equals("teacher")){}

else if(userType.equals("admin")){}

}

### 6.1.3 业务逻辑层

登录模块的业务逻辑层主要完成对登录系统的用户信息的获取，同时调用登录模块的业务逻辑接口。比如说：在用户登录系统后，对其账号和密码等信息进行获取，并将用户名等信息显示在表现层中。登录模块业务逻辑层列表如图6-3所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 登录 | LoginService.java | 调用LoginMapper.java | 返回给Action |

表 6-3登录模块业务逻辑层列表

在登录模块的业务逻辑层是调用了公用的LoginMapper.java接口，同时实现该接口。

LoginService.java接口：

**package com.gxun.jms.service;**

@Service

public class LoginService {

@Autowired

private LoginMapper loginMapper;

}

LoginService.java调用数据持久层的LoginMapper.java

LoginMapper.java接口：

package com.gxun.jms.dao;

@Mapper

public interface LoginMapper {

}

### 6.1.4 数据持久层

登录模块的数据持久层可以在用户输入账号密码以及选择登录类型，登录系统以后获取到登录角色的信息。登入模块数据持久层列表见表6-4所示

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 登录 | LoginMapper.java | 调用 Teacher.java  和Student.java | 利用接口对用户信息进行操作 |

表 6-4登录模块数据持久层列表

LoginMapper.java定义了对用户信息操作的接口：

package com.gxun.jms.dao;

/\*

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

\*/

**@Mapper**

**public interface LoginMapper {**

**public Student studentLogin(@Param("studentLoginID") String studentLoginID, @Param("studentPassword") String studentPassword);**

**public Teacher teacherLogin(@Param("teacherLoginID")String teacherLoginID, @Param("teacherPassword")String teacherPassword);**

**}**

### 6.1.5 域模型层

登录模块用到域模型层中student.java和teacher.java，student.java

teacher.java是公用域模型，在涉及到用户信息操作时，就会调用到该模型，登录模块域模型层列表如表6-5所示。

|  |  |
| --- | --- |
| 域模型 | 描述 |
| student.java和teacher.java | 对数据库中的信息进行增、删、改、查操作 |

表 6-5登录模块域模型层列表

Student.java主要属性与方法：

**package com.gxun.jms.pojo;**

/\*

\* *@description* student表的信息记录

\*/

**public class Student {**

**private String studentName;**

**private int studentID;**

**private String studentPassword;**

**private int classID;**

**private int collegeID;**

**private int activeFlag;**

**private String studentLoginID;**

/\*编写对上述各个属性的*get*与*set*方法\*/

}

Teacher.java主要属性与方法：

**package com.gxun.jms.pojo;**

/\*

\* *@description* Teacher表的信息记录

\*/

**public class Teacher {**

**private String teacherName;**

**private int teacherID;**

**private String teacherPassword;**

**private int collegeID;**

**private int activeFlag;**

**private String teacherLoginID;**

/\*编写对上述各个属性的*get*与*set*方法\*/

}