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

**作业批改系统详细设计**

**V2.0**

**广西民族大学**

评审日期： 2019年6月27日

目录

[1．导言 4](#_Toc12546195)

[1.1 目的 4](#_Toc12546196)

[1.2 范围 4](#_Toc12546197)

[1.3 引用标准 4](#_Toc12546198)

[1.4 参考资料 4](#_Toc12546199)

[1.5 版本更新信息 5](#_Toc12546200)

[2．详细设计简述 5](#_Toc12546201)

[2.1 设计简介 5](#_Toc12546202)

[2.2 模块简介 6](#_Toc12546203)

[3、界面详细设计 6](#_Toc12546204)

[3.1、页面设计说明 6](#_Toc12546205)

[3.2、静态页面详细设计 6](#_Toc12546206)

[4、学生端模块详细设计 6](#_Toc12546207)

[4.1、提交作业模块详细设计 6](#_Toc12546208)

[4.1.1 表现层 7](#_Toc12546209)

[4.1.2 控制层 9](#_Toc12546210)

[4.1.3 业务逻辑层 10](#_Toc12546211)

[4.1.4 数据持久层 12](#_Toc12546212)

[4.1.5 域模型层 14](#_Toc12546213)

[4.2作业查询模块详细设计 15](#_Toc12546214)

[4.2.1 表现层 16](#_Toc12546215)

[4.2.2 控制层 17](#_Toc12546216)

[4.2.3 业务逻辑层 18](#_Toc12546217)

[4.2.4 数据持久层 20](#_Toc12546218)

[4.2.5 域模型层 21](#_Toc12546219)

[5、 教师端模块详细设计 22](#_Toc12546220)

[5.1、作业发布模块详细设计 22](#_Toc12546221)

[5.1.1 表现层 23](#_Toc12546222)

[5.2.2 控制层 24](#_Toc12546223)

[5.2.3 业务逻辑层 26](#_Toc12546224)

[5.2.4 数据持久层 27](#_Toc12546225)

[5.2.5 域模型层 29](#_Toc12546226)

[5.2、作业批改模块详细设计 31](#_Toc12546227)

[5.2.1 表现层 32](#_Toc12546228)

[5.2.2 控制层 33](#_Toc12546229)

[5.2.3 业务逻辑层 35](#_Toc12546230)

[5.2.4 数据持久层 37](#_Toc12546231)

[5.2.5 域模型层 38](#_Toc12546232)

[5.3、成绩查询模块详细设计 39](#_Toc12546233)

[5.3.1 表现层 40](#_Toc12546234)

[5.3.2 控制层 41](#_Toc12546235)

[5.3.3 业务逻辑层 42](#_Toc12546236)

[5.3.4 数据持久层 43](#_Toc12546237)

[5.3.5 域模型层 44](#_Toc12546238)

[6、 公共部分模块详细设计 45](#_Toc12546239)

[6.1、登录模块 45](#_Toc12546240)

[6.1.1 表现层 46](#_Toc12546241)

[6.1.2 控制层 47](#_Toc12546242)

[6.2.3 业务逻辑层 49](#_Toc12546243)

[6.2.4 数据持久层 50](#_Toc12546244)

[6.2.5 域模型层 52](#_Toc12546245)

# 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

因为该系统的开发使用增量模型，故该文档内容涵盖作业批改系统的核心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 | 2019.6.27 | 0.1 | 全部 | 初始发布版本 |
| 002 | 2019.6.27 | 2.0 |  | 第二发布版本 |

表1-1 文档更新记录

# 2．详细设计简述

## 2.1 设计简介

作业批改系统分层结构图如图2-1所示。

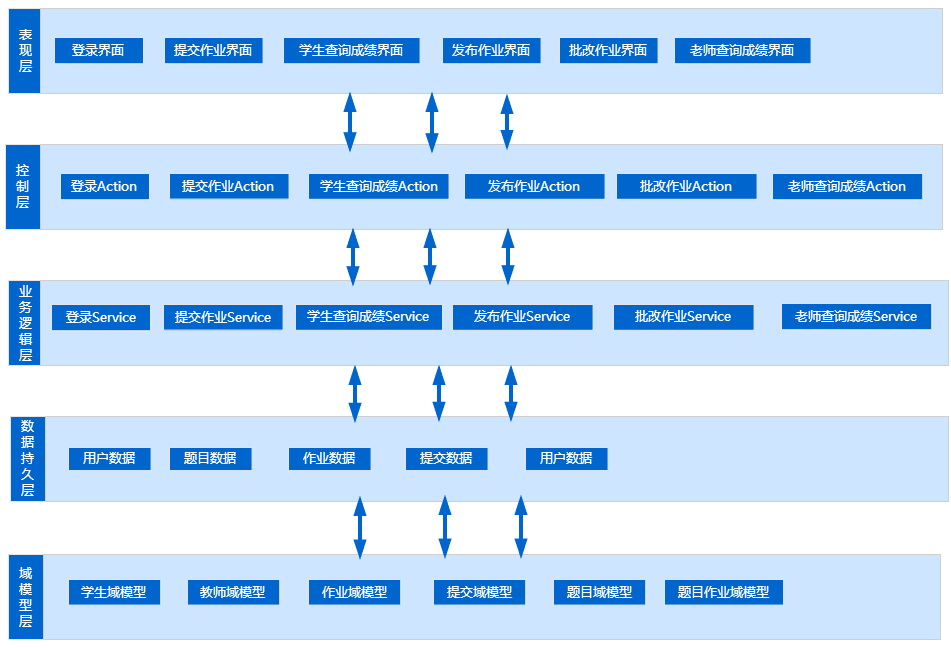


图 2-1 作业批改系统分层结构示意图

## 2.2 模块简介

系统功能模块图如图2-2所示。

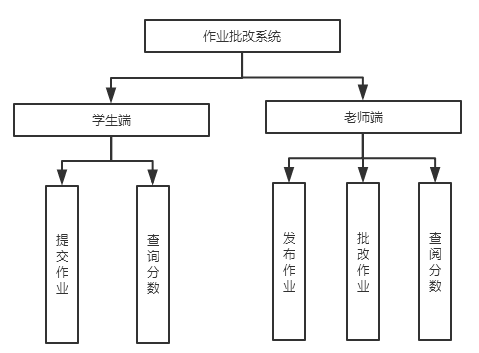


图2—2作业批改系统功能模块图

# 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3-1所示

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 学生首页 | StudentFace |  |
| 提交作业 | StuWrite | 学生可进行答题，答题后提交作业 |
| 学生成绩查询 | Stuquery | 学生查询成绩接口 |
| 发布作业 | TeaWorkPub | 老师发布作业接口，可对学生发布作业 |
| 批改作业 | TeaWorkCorret | 老师批改作业接口，对学生答题情况进行打分 |
| 教师成绩查询 | TeaQurey | 教师成绩查询接口 |
| 题库管理 | TeaWorkPub | 老师题库管理接口，可查看题目 |

表3-1 页面表现层设计

# 4、学生端模块详细设计

学生端模块主要包括提交作业模块、作业查询。

## 4.1、提交作业模块详细设计

在提交作业模块时，提交作业系统内部的相应响应操作示意图如图4-1所示。

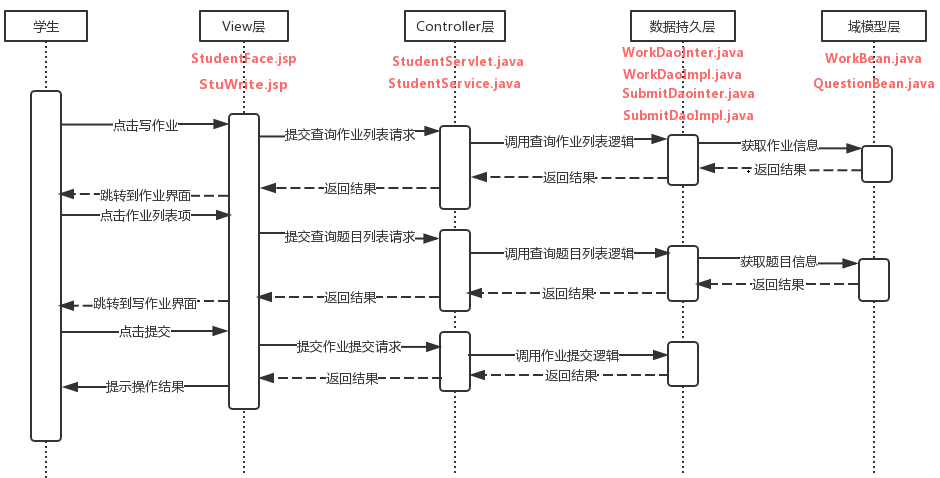


图 4-1 提交作业响应示意图

### 4.1.1 表现层

提交作业模块的表现层主要完成学生提交作业功能，进入学生端界面时要求学生点击相应作业名称，跳转到相应做题界面，完成作业后点击提交。表现层对应的JSP页面列表见表4-2所示。

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 学生界面 | StudentFace.jsp | 学生端界面 |
| 写作业页面 | StuWrite.jsp | 学生写作业功能，当页面出错时给出提示。 |

表 4-2 提交作业模块表现层JSP列表

StudentFace.jsp、StuWrite.jsp的流程图如图4-3所示。

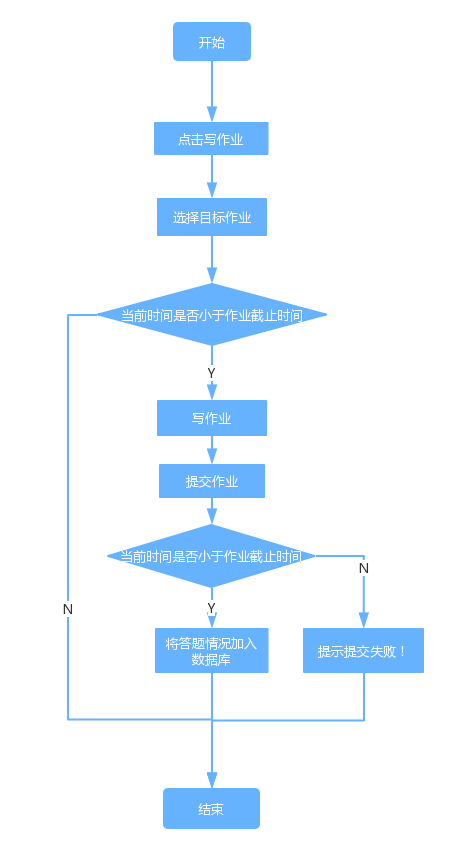


图 4-3 学生提交作业模块详细设计流程图

### 4.1.2 控制层

学生体积提交作业模块的控制层负责接受来自StudentFace.jsp和StuWrite.jsp的学生操作，同时调用学生提交作业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。学生提交作业模块控制层列表见表4-4所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 逻辑 | Java | | 转移说明 | 出口 |
| 学生控制层 | | StudentServlet.java | SUCCESS | StuWrite.jsp—作业提交成功，显示提示窗口。 |
| ERROR | Error.jsp---作业提交失败，显示提示窗口。 |

表 4-4 学生提交作业模块控制层列表

在控制层中StudentServlet.java的描述如下所示：

**package** com.gxun.servlet;

/\*\*\* *@description* 实现学生控制\*/

**public** **class** StudentServlet **extends** HttpServlet

{ *@Resource*

**private StudentService studentService = new StudentService();**

**protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {}**

**protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {}**

**private void getWokeList(HttpServletRequest request, HttpServletResponse response){} //获取作业列表**

**private void getQuestionList(HttpServletRequest request, HttpServletResponse response){} //获取该作业的所有题目列表**

**private void submit(HttpServletRequest request, HttpServletResponse response){} //提交作业**

**private void query(HttpServletRequest request, HttpServletResponse response){} //查询成绩**

}

### 4.1.3 业务逻辑层

提交作业模块的业务逻辑层主要完成对提交作业逻辑的判定，同时调用提交作业模块的业务逻辑接口。提交作业模块业务逻辑层列表如图4-5所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 业务逻辑 | Java | 调用说明 | 出口 |
| 学生  业务逻辑 | StudentService.java | 调用WorkDaoImpl;  调用SubmitDaoIpml; | 返回给StuWirte.jsp |

表 4-5提交作业模块业务逻辑层列表

在学生提交作业模块的业务逻辑层是调用了WorkDaoImpl、SubmitDaoIpml接口，同时在实现该接口。

StudentService.java接口主要方法：

**package** com.gxun.service;

/\*\*

\* *@description* 实现学生业务逻辑

\*/

**public class** StudentService **{**

WorkDaoInter workDao = new WorkDaoImpl ();

SubmitDaoInter submitDao = new SubmitDaoIpml ();

/\*\*获取作业列表

\* @return 作业列表

\*/

**private String getWokeList(){}**

/\*\*获取该作业的所有题目

\* @param 作业号

\* @return 作业列表

\*/

**private String** **getQuestionList** (String Wid){}

/\*\* 获取该作业的所有题目

\* @param 作业答题内容

\*/

**private void** **submit** (SubmitBean sub){}

/\*\* 成绩查询

\* @ return 成绩列表

\*/

**private String** **query** (String Sid){}

}

### 4.1.4 数据持久层

学生提交作业模块的数据持久层能对作业数据进行查询，对提交数据进行增加。学生提交作业模块数据持久层列表见表4-6所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 提交作业 | WorkDaoInter.java  WorkDaoImpl.java | 调用 workBean.Java | 对作业数据进行查询 |
| SubmitDaoInter.java  SubmitDaoIpml.java | 调用submitBean.java | 对提交数据进行增加 |

表 4-6 学生提交作业模块数据持久层列表

WorkDaoInter.java：

**package** com.gxun.dao;

/\*\*

\* @description

\*/

**public interface** WorkDaoInter {

**public** List<WorkBean> queryWorkList(); /\*对作业数据进行查询\*/

}

WorkDaoImpl.java是WorkDaoInter.java接口的实现：

**package** com.gxun.dao;

/\*\*

\* @description

\*/

@Repository

**public** **class** WorkDaoImpl implements WorkDaoInter {

**public** List<WorkBean> queryWorkList(){}; /\*对作业数据进行查询\*/

}

SubmitDaoInter.java：

**package** com.gxun.dao;

/\*\*

\* @description

\*/

**public interface** SubmitDaoInter {

**public** void addSubmit(SubmitBean sub); /\*对提交数据进行增加\*/

}

SubmitDaoImpl.java是SubmitDaoInter.java接口的实现：

**package** com.gxun.dao;

/\*\*

\* @description

\*/

@Repository

**public** **class** SubmitDaoImpl implements SubmitDaoInter {

**public** void addSubmit(SubmitBean sub){}; /\*对提交数据进行增加\*/

}

### 4.1.5 域模型层

学生提交作业模块用到域模型层中WorkBean.java和SubmintBean.java，学生提交作业模块域模型层列表如表4-7所示。

|  |  |
| --- | --- |
| 域模型 | 描述 |
| WorkBean.java | 作业实体 |
| SubmintBean.java | 学生答案 |

表 4-7学生提交作业模块域模型层列表

WorkBean.java主要属性与方法：

**package** com. gxun.bean;

/\*\*

\* *@description* Work表的信息记录

\*/

**public** **class** WorkBean **implements** Serializable{

**private** **String** wid; /\*作业号\*/

**private** **String** tid; /\*工号\*/

**private** **String** wname; /\*作业名称 \*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

SubmintBean.java主要属性与方法：

**package** com. gxun.bean;

/\*\*

\* *@description* Submit表的信息记录

\*/

**public** **class** SubmintBean **implements** Serializable{

**private** **String** num; /\*编号\*/

**private** **String** sid; /\*学号\*/

**private** **String** tid; /\*工号 \*/

**private** **String** wid; /\*作业号 \*/

**private** **String** answer; /\*学生答案 \*/

**private** **String** grade; /\*分数 \*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

## 4.2作业查询模块详细设计

在学生作业查询模块中，作业查询模块系统内部的响应操作示意图如图4-8所示。

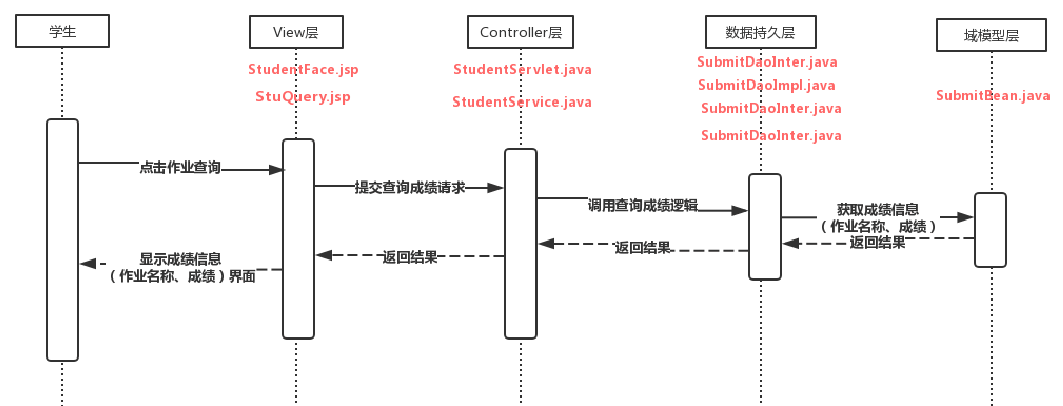


图 4-8 学生作业查询模块系统响应示意图

### 4.2.1 表现层

作业查询模块的表现层主要完成学生查询作业成绩功能，进入学生端界面时，学生点击分数查询时，跳转到相应成绩显示界面，若学生此前已答题，且教师已批改，则显示学生作业成绩，否则则显示“暂无成绩”。表现层对应的JSP页面列表见表4-9所示。

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 学生界面 | StudentFace.jsp | 学生端界面 |
| 作业查询页面 | StuQuery.jsp | 学生查询作业（成绩）功能，当页面出错时给出提示。 |

表 4-9作业查询模块表现层JSP列表

StudentFace.jsp、StuQuery.jsp的流程图如图4-10所示。

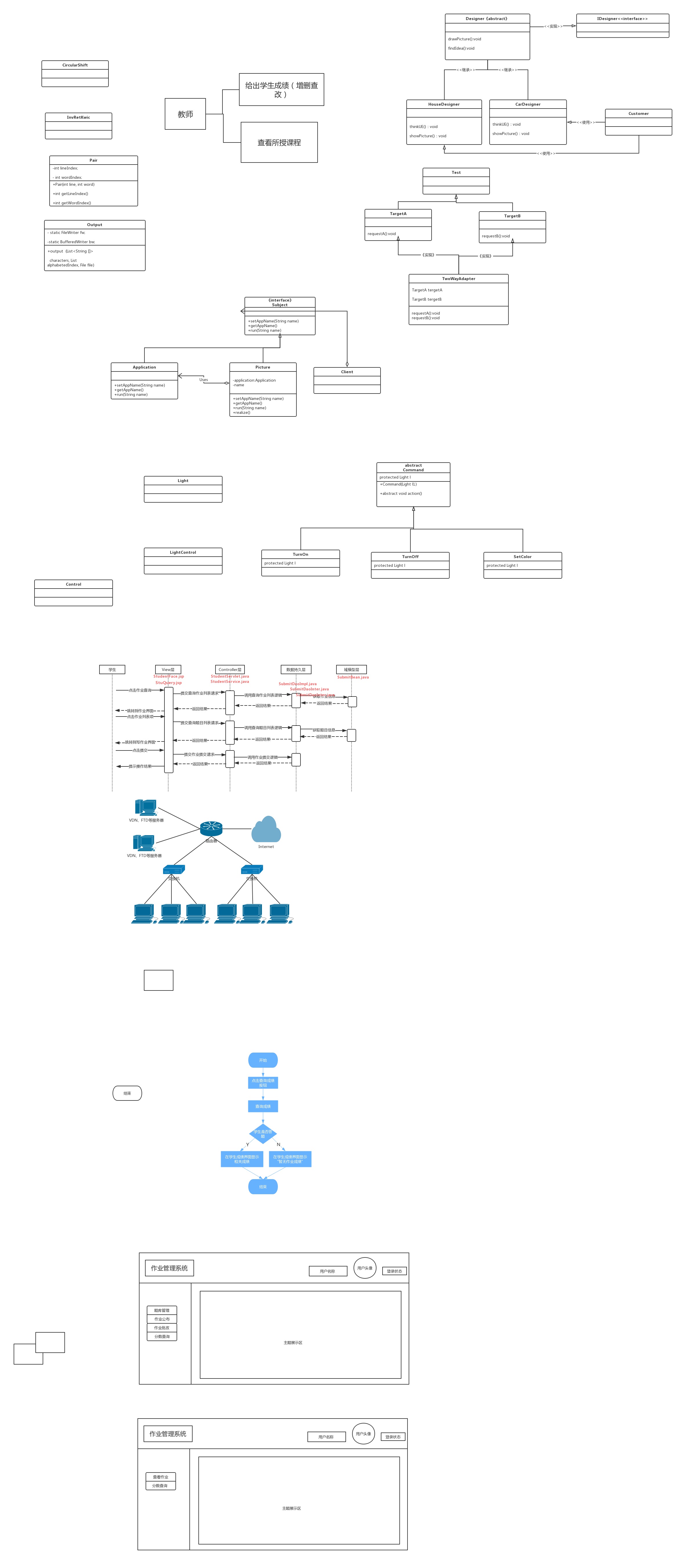


图 4-10 作业查询模块系统流程图

### 4.2.2 控制层

学生作业查询模块的控制层负责接受来自StudentFace.jsp和StuQuery.jsp的学生操作，同时调用学生作业查询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。学生作业查询模块控制层列表见表4-11所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 逻辑 | Java | | 转移说明 | 出口 |
| 学生控制层 | | StudentServlet.java | SUCCESS | StuQuery.jsp—查询成绩窗口。 |
| ERROR | Error.jsp---查询成绩失败，显示提示窗口。 |

表 4-11 学生作业查询模块控制层列表

在控制层中StudentServlet.java的描述如下所示

**package** com.gxun.servlet;

/\*\*

\* *@description* 实现学生控制

\*/

**public** **class** StudentServlet **extends** HttpServlet

{ *@Resource*

**private StudentService studentService = new StudentService();**

**protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {}**

**protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {}**

**private void** **submit** (SubmitBean sub){}

/\*\* 成绩查询

\* @ return 成绩列表

\*/

**private String** **query** (){}

}

### 4.2.3 业务逻辑层

学生作业查询模块的业务逻辑层主要完成向数据库内“submit”表进行数据的查询，以最终达到查询学生成绩的目的。学生作业查询模块业务逻辑层列表如图4-12所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 业务逻辑 | Java | 调用说明 | 出口 |
| 学生  业务逻辑 | StudentService.java | 调用SubmitDaoIpml; | 返回给StuQuery.jsp |

表 4-12学生作业查询模块业务逻辑层列表

在学生作业查询模块的业务逻辑层是调用了SubmitDaoIpml接口，同时再实现该接口。

StudentService.java接口主要方法：

**package** com.gxun.service;

/\*\*

\* *@description* 实现学生业务逻辑

\*/

**public class** StudentService **{**

SubmitDaoInter submitDao = new SubmitDaoIpml ();

/\*\*获取提交列表

\* @return 提交列表

\*/

**private String getSubmitList(**Submit sub**){ }**

/\*\*获取提交列表

\* @param 作业号

\* @return 提交列表

\*/

**private void** **submit** (SubmitBean sub){}

/\*\* 成绩查询

\* @ return 成绩列表

\*/

}

### 4.2.4 数据持久层

学生作业查询模块的数据持久层能对作业情况（学生成绩）进行查询，学生作业查询模块数据持久层列表见表4-13所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 提交作业 | SubmitDaoInter.java  SubmitDaoIpml.java | 调用submitBean.java | 对学生成绩进行查询 |

表4-13学生作业查询模块数据持久层列表

SubmitDaoInter.java定义了提交信息进行查的接口

**package** com.gxun.dao;

/\*\*

\* @description

\*/

**public interface** SubmitDaoInter {

**public** List<SubmitBean> querySubmitList(); /\*对提交数据进行查询\*/

}

SubmitDaoImpl.java是SubmitDaoInter.java接口的实现：

**package** com.gxun.dao;

/\*\*

\* @description

\*/

@Repository

**public** **class** SubmitDaoImpl implements SubmitDaoInter {

**public** List<SubmitBean> querySubmitList(){}; /\*对作业数据进行查询\*/

}

### 4.2.5 域模型层

学生作业查询模块用到域模型层中SubmitBean.java，学生作业查询模块域模型层列表如表4-14所示。

|  |  |
| --- | --- |
| 域模型 | 描述 |
| SubmitBean.java | 提交实体 |

表 4-14学生作业查询模块域模型层列表

SubmitBean.java主要属性与方法：

**package** com. gxun.bean;

/\*\*

\* *@description* Submit表的信息记录

\*/

**public** **class** SubmitBean **implements** Serializable{

**private** **String** num; /\*提交编号\*/

**private** **String** wid; /\*作业号\*/

**Private String** tid; /\*工号\*/

**Private String** sid; /\*学号\*/

**Private String** answer /\*学生答案\*/

**Private int grade**; /\*分数\*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/

# 教师端模块详细设计

教师端模块主要包括作业发布、作业批改、成绩查询。

## 5.1、作业发布模块详细设计

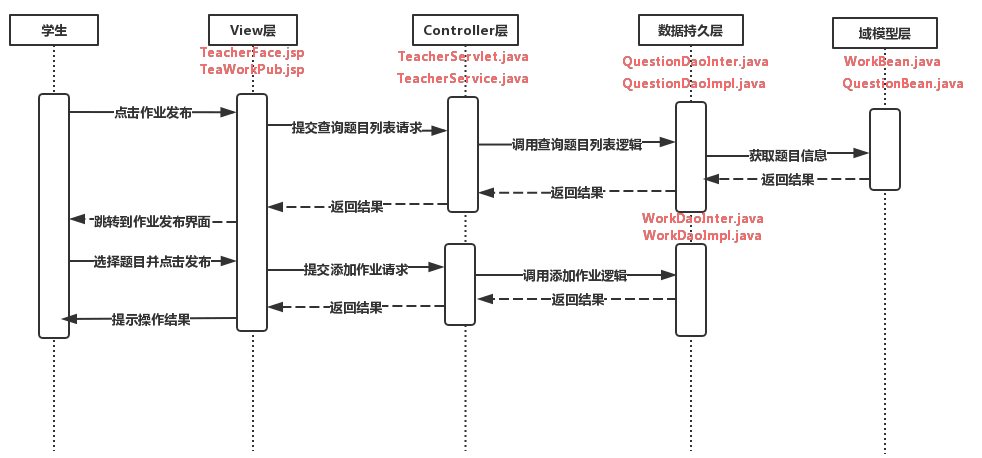


图 5-1 作业发布响应示意图

### 5.1.1 表现层

作业发布模块的表现层主要完成老师发布作业功能，进入教师端界面时点击作业发布，跳转到选择题目界面，选择相应题目后点击发布按钮，若无异常则提示发布成功，否则提示发布失败。作业发布模块表现层对应的JSP页面列表见表5-2所示。

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 教师界面 | TeacherFace.jsp | 教师端界面 |
| 作业发布页面 | TeaWorkPub.jsp | 作业发布界面 |

表 5-2 写作业模块表现层JSP列表

StudentFace.jsp、TeaWorkPub.jsp的流程图如图5-3所示。

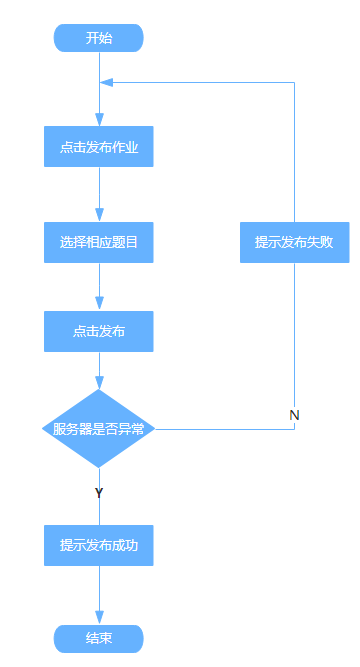


图 5-3 作业发布模块系统流程图

### 5.2.2 控制层

教师发布作业模块的控制层负责接受来自TeacherFace.jsp和TeaWorkPub.jsp的老师操作，同时调用教师提交作业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。教师发布作业模块控制层列表见表5-4所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 逻辑 | Java | | 转移说明 | 出口 |
| 老师控制层 | | TeacherServlet.java | SUCCESS | StuWrite.jsp—作业提交成功，显示提示窗口。 |
| ERROR | Error.jsp---作业提交失败，显示提示窗口。 |

表 5-4 作业发布模块控制层列表

在控制层中TeacherServlet.java的描述如下所示：

**package** com.gxun.servlet;

/\*\*

\* *@description* 实现老师控制

\*/

**public** **class** TeacherServlet **extends** HttpServlet

{ *@Resource*

**private TeacherService teacherService = new TeacherService ();**

**protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {}**

**protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {}**

**private void getQuestionList(HttpServletRequest request, HttpServletResponse response){} //获取所有题目列表**

**private void release(HttpServletRequest request, HttpServletResponse response){} //发布作业**

**private void query(HttpServletRequest request, HttpServletResponse response){} //查询成绩**

}

### 5.2.3 业务逻辑层

作业发布模块的业务逻辑层主要完成对老师的判定，同时调用作业发布模块的数据逻辑接口。作业发布模块业务逻辑层列表如图5-5所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 业务逻辑 | Java | 调用说明 | 出口 |
| 老师  业务逻辑 | TeacherService.java | 调用QuestDaoImpl;  调用WorkDaoIpml; | 返回给TeaWorkPub.jsp |

表 5-5作业发布模块业务逻辑层列表

在老师发布作业模块的业务逻辑层是调用了WorkDaoImpl、QuestDaoImpl接口，同时在实现该接口。

**package** com.gxun.service;

/\*\*

\* *@description* 实现老师业务逻辑

\*/

**public class** TeacherService **{**

WorkDaoInter workDao = new WorkDaoImpl ();

QuestDaoInter questDao = new QuestDaoImpel();

/\*\*获取所有题目

\* @return 题目列表

\*/

**private String** **getQuestionList** (){}

/\*\* 添加作业

\* @param 作业实体

\*/

**private void** **release** (WorkBean work){}

}

### 5.2.4 数据持久层

老师作业发布模块的数据持久层能对题目数据进行查询，对提交数据进行增加。老师作业发布模块数据持久层列表见表5-6所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 发布作业 | WorkDaoInter.java  WorkDaoImpl.java | 调用 workBean.Java | 对提交数据进行增加 |
| QuestDaoInter.java  QuestDaoImpl.java | 调用questBean.java | 对题目数据查询进行 |

表 5-6 作业发布模块数据持久层列表

WorkDaoInter.java定义了用户信息进行增、删、改、查的接口：

**package** com.gxun.dao;

/\*\*

\* @description

\*/

**public interface** WorkDaoInter {

**public** List<WorkBean> queryWorkList(); /\*对作业数据进行查询\*/

**public** void addwork(WorkBean work); /\*对提交数据进行增加\*/

}

**package** com.gxun.dao;

/\*\*

\* @description

\*/

**public interface** QuestDaoInter {

**public** List<questionBean> queryquestionList(); /\*对题目数据进行查询\*/

}

WorkDaoImpl.java是WorkDaoInter.java接口的实现：

**package** com.gxun.dao;

/\*\*

\* @description

\*/

@Repository

**public** **class** WorkDaoImpl implements WorkDaoInter {

**public** List<WorkBean> queryWorkList(){}; /\*对作业数据进行查询\*/

**public** void addwork (WorkBean work){}; /\*对提交数据进行增加\*/

}

**package** com.gxun.dao;

/\*\*

\* @description

\*/

@Repository

**public** **class** QuestDaoImpl implements QuestDaoInter {

**public** List<questionBean> queryquestionList(); /\*对题目数据进行查询

}

### 5.2.5 域模型层

老师作业发布模块用到域模型层中WorkBean.java，QuestionBean.java,老师发布作业模块域模型层列表如表5-7所示。

|  |  |
| --- | --- |
| 域模型 | 描述 |
| WorkBean.java | 作业实体 |
| QuestionBean | 题目实体 |

表 5-7作业发布模块域模型层列表

WorkBean.java主要属性与方法：

**package** com. gxun.bean;

/\*\*

\* *@description* Work表的信息记录

\*/

**public** **class** Work **implements** Serializable{

**private** **String** wid; /\*id识别\*/

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

**private** **String** tid; /\*老师工号 \*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

QuestionBean.java主要属性与方法：

**package** com. gxun.bean;

/\*\*

\* *@description* Question表的信息记录

\*/

**public** **class** Question **implements** Serializable{

**private** **String** Qid; /\*id识别\*/

**private** **String** Content; /\*内容\*/

**private** **String** Type; /\*类型 \*/

private int Snum; /\* \*/分值

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

## 5.2、作业批改模块详细设计

在教师作业批改模块时，教师作业批改模块系统内部的相应响应操作示意图如图所示。

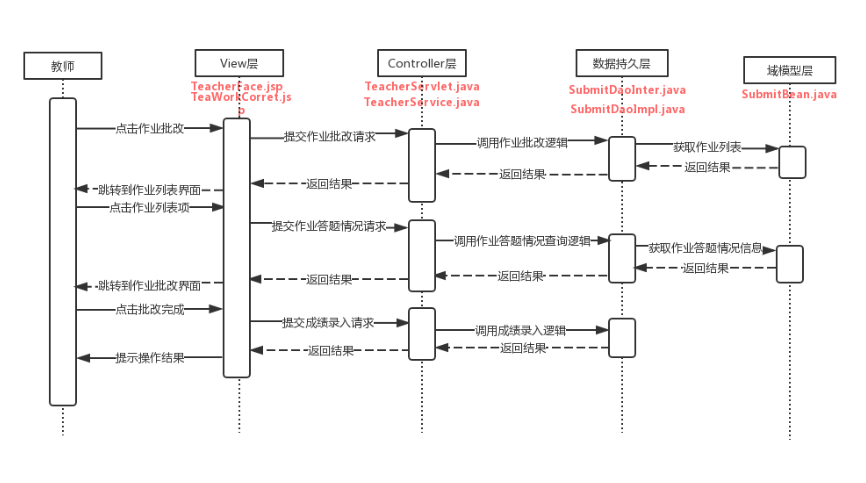


图 5-8 教师作业批改响应示意图

### 5.2.1 表现层

教师作业批改模块的表现层主要完成教师作业批改功能，进入教师端界面时要求教师点击作业批改，跳转到作业列表界面，教师点击对应的作业，调转到对应的作业答题情况信息界面，之后教师进行作业批改，成绩录入。表现层对应的JSP页面列表见表5-9所示。

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 教师界面 | TeacherFace.jsp | 教师端界面 |
| 作业列表 | TeaWorkCorrect.jsp | 作业批改界面。 |

表 5-9 教师作业批改表现层JSP列表

TeacherFace.jsp、TeaWorkcorrect.jsp的流程图如图4-10所示。

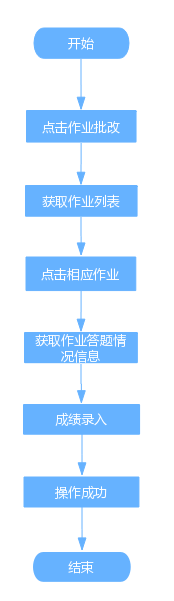


图 5-10 教师作业批改详细设计流程图

### 5.2.2 控制层

教师作业批改模块的控制层负责接受来自TeacherFace.jsp和TeaWorkCorrect.jsp的教师操作，同时调用教师作业批改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。教师作业批改模块控制层列表见表5-11所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 逻辑 | Java | | 转移说明 | 出口 |
| 教师控制层 | | TeacherServlet.java | SUCCESS | TeaWorkCorrect.jsp—作业批改成功，显示提示窗口。 |
| ERROR | Error.jsp---作业批改失败，显示提示窗口。 |

表 5-11 教师作业批改模块控制层列表

在控制层中TeacherServlet.java的描述如下所示：

**package** com.gxun.servlet;

/\*\*

\* *@description* 实现教师控制

\*/

**public** **class** TeacherServlet **extends** HttpServlet

{ *@Resource*

**private TeacherService TeacherService = new TeacherService();**

**protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {}**

**protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {}**

**private void getWokeList(HttpServletRequest request, HttpServletResponse response){} //获取作业列表**

**private void getSubmitList(HttpServletRequest request, HttpServletResponse response){} //获取该作业的所有提交列表**

**private void input(HttpServletRequest request, HttpServletResponse response){} //录入成绩**

}

### 5.2.3 业务逻辑层

教师作业批改模块的业务逻辑层主要完成作业批改逻辑的判定，同时调用作业提交模块的业务逻辑接口。教师作业批改模块业务逻辑层列表如图5-12所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 业务逻辑 | Java | 调用说明 | 出口 |
| 教师  业务逻辑 | TeacherService.java | 调用WorkDaoIpml;  调用SubmitDaoIpml; | 返回给TeaWorlCorret.jsp |

表 5-12教师作业批改模块业务逻辑层列表

在教师作业批改模块的业务逻辑层是调用了WorkDaoIpml、SubmitDaoIpml接口，同时在实现该接口。

TeacherService.java接口主要方法：

**package** com.gxun.service;

/\*\*

\* *@description* 实现教师业务逻辑

\*/

**public class Teacher**Service **{**

WorkDaoInter workDao = new WorkDaoImpl ();

SubmitDaoInter submitDao = new SubmitDaoIpml ();

/\*\*获取作业列表

\* @return 作业列表

\*/

**private String getWokeList(){}**

/\*\*获取该作业的所有题目

\* @param 作业号

\* @return 作业列表

\*/

**private String** **getSubmitList** (String num){}

/\*\* 获取该作业的所有提交信息

\* @param 学生作业答案

\*/

**private void** **correct**(){}

/\*\* 作业批改\*/

**private String** **input**(){}

/\*\* 成绩录入 \*/

}

### 5.2.4 数据持久层

教师作业批改模块的数据持久层能对作业数据进行查询，对提交数据进行批改。教师作业批改模块数据持久层列表见表5-13所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 批改作业 | WorkDaoInter.java  WorkDaoImpl.java | 调用 workBean.Java | 对作业数据进行查询 |
| SubmitDaoInter.java  SubmitDaoIpml.java | 调用submitBean.java | 对提交数据进行查询 |

表 5-13 教师作业批改模块数据持久层列表

WorkDaoInter.java定义了教师批改作业的接口：

**package** com.gxun.dao;

/\*\*

\* @description

\*/

**public interface** WorkDaoInter {

**public** List<WorkBean> queryWorkList(); /\*对作业数据进行查询\*/

**public** List<SubmitBean> querySubmitList(); /\*对提交数据进行查询\*/

**public** void inputSubmit(SubmitBean sub); /\*对作业成绩进行录入\*/

}

WorkDaoImpl.java是WorkDaoInter.java接口的实现：

**package** com.gxun.dao;

/\*\*

\* @description

\*/

@Repository

**public** **class** WorkDaoImpl implements WorkDaoInter {

**public** List<WorkBean> queryWorkList(); /\*对作业数据进行查询\*/

**public** List<SubmitBean> querySubmitList(); /\*对提交数据进行查询\*/

**public** void inputSubmit(SubmitBean sub); /\*对作业成绩进行录入\*/

}

### 5.2.5 域模型层

教师作业批改模块用到域模型层中SubmitBean.java，教师作业批改模块域模型层列表如表5-14所示。

|  |  |
| --- | --- |
| 域模型 | 描述 |
| SubmitBean.java | 作业实体 |

表 5-14教师作业批改模块域模型层列表

SubmitBean.java主要属性与方法：

**package** com. gxun.bean;

/\*\*

\* *@description* Submit表的信息记录

\*/

**public** **class** Submit**implements** Serializable{

**private** **String** num; /\*编号\*/

**private** **String** wid; /\*作业号\*/

**private** **String** tid; /\*工号 \*/

**private** **String** sid; /\*学号\*/

**private** **String** answer; /\*学生答案 \*/

**private** **int** grade; /\*分数\*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

## 5.3、成绩查询模块详细设计

在教师成绩查询模块时，教师成绩查询模块系统内部的相应响应操作示意图如图5-15所示。

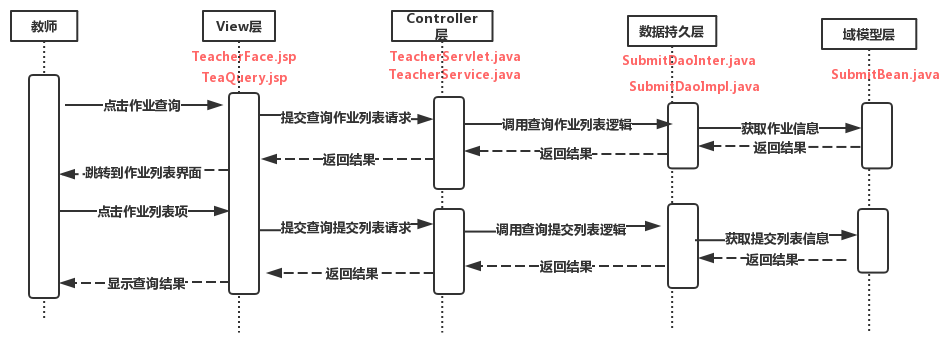


图 5-15 教师成绩查询响应示意图

### 5.3.1 表现层

教师成绩查询模块的表现层主要完成教师查询某次作业的学生成绩、题号、成绩等信息，进入教师端界面时要求教师点击成绩查询，进入成绩作业列表界面，点击相应作业名称，跳转到相应查询结果界面，该界面有相应的题号、学生姓名、成绩等信息。表现层对应的JSP页面列表见表5-16所示。

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 教师界面 | TeacherFace.jsp | 教师端界面 |
| 教师查询页面 | TeaQuery.jsp | 教师查询成绩功能，当页面出错时给出提示。 |

表 5-16 教师成绩查询模块表现层JSP列表

TeacherFace.jsp、TeaQuery.jsp的流程图如图5-17所示。

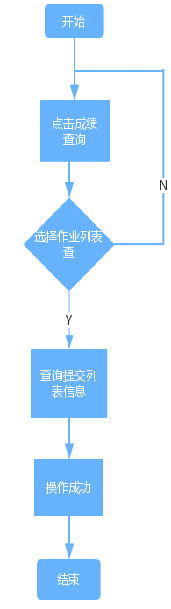


图 5-17 教师成绩查询模块系统流程图

### 5.3.2 控制层

教师成绩查询模块的控制层负责接受来自TeacherFace.jsp和TeaQuery.jsp的教师操作，同时调用教师成绩查询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。教师成绩查询模块列表见表5-18所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 逻辑 | Java | | 转移说明 | 出口 |
| 教师控制层 | | TeacherServlet.java | SUCCESS | TeaQuery.jsp—成绩查询成功，显示提示窗口。 |
| ERROR | Error.jsp---作业提交失败，显示提示窗口。 |

表5-18教师成绩查询模块控制层列表

在控制层中StudentServlet.java的描述如下所示：

**package** com.gxun.servlet;

/\*\* *@description* 实现教师控制

\*/

**public** **class** TeacherServlet **extends** HttpServlet

{ *@Resource*

**private TeacherService service = new TeacherService();**

**protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {}**

**protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {}**

**private void getWokeList(HttpServletRequest request, HttpServletResponse response){} //获取作业列表**

**private void getSubmitList(HttpServletRequest request, HttpServletResponse response){} //获取提交列表信息**

**private void query(HttpServletRequest request, HttpServletResponse response){} //查询成绩等相关信息**}

### 5.3.3 业务逻辑层

成绩查询模块的业务逻辑层主要完成对教师查询逻辑的判定，同时调用成绩查询模块的业务逻辑接口。登录模块业务逻辑层列表如图5-19所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 业务逻辑 | Java | 调用说明 | 出口 |
| 教师  业务逻辑 | TeacherService.java | 调用SubmitDaoIpml; | 返回给TeaQuery.jsp |

表 5-19成绩查询模块业务逻辑层列表

在学生写作业模块的业务逻辑层是调用了SubmitDaoIpml接口，同时实现该接口。

TeacherService.java接口主要方法：

**package com.gxun.service;**

**/\*\*\* @description 实现教师成绩查询业务逻辑\*/**

**public class TeacherService {**

**SubmitDaoInter submitDao = new SubmitDaoIpml ();**

**/\*\*获取作业列表**

**\* @return 作业列表**

**\*/**

**private String getWokeList(){}**

**/\*\*获取所有作业**

**\* @param 作业号**

**\* @return 作业列表**

**\*/**

**private void getSubmitList (SubmitBean sub){}**

**/\*\* 获取提交列表信息**

**\* @ return 提交列表 \*/**

**private String query (){}**

**}**

### 5.3.4 数据持久层

教师成绩查询模块的数据持久层能对作业数据进行查询，获取提交列表信息。教师成绩查询模块数据持久层表见表5-20所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | DAO | 调用数据模型 | 说明 |
| 提交作业 | WorkDaoInter.java  WorkDaoImpl.java | 调用 workBean.Java | 对作业数据进行查询 |
| SubmitDaoInter.java  SubmitDaoIpml.java | 调用submitBean.java | 对提交数据进行查询 |

表 5-20 教师成绩查询模块数据持久层表

WorkDaoInter.java定义了用户信息进行增、删、改、查的接口：

**package** com.gxun.dao;

**public interface** WorkDaoInter {

**public** List<WorkBean> queryWorkList(); /\*对作业数据进行查询\*/

**public** void querySubmit(SubmitBean sub); /\*对提交数据进行查询\*/

}

WorkDaoImpl.java是WorkDaoInter.java接口的实现：

**package** com.gxun.dao;

@Repository

**public** **class** WorkDaoImpl implements WorkDaoInter {

**public** List<WorkBean> queryWorkList(){}; /\*对作业数据进行查询\*/

**public** void querySubmit(SubmitBean sub){}; /\*对提交数据进行查询\*/

}

### 5.3.5 域模型层

教师成绩查询模块用到域模型层中WorkBean.java和SubmitBean.java，教师成绩查询模块域模型层表如表5-21所示。

|  |  |
| --- | --- |
| 域模型 | 描述 |
| WorkBean.java | 作业实体 |
| SubmitBean.java | 提交表实体 |

表 5-21教师成绩查询模块域模型层列表

WorkBean.java包含了作业的基本信息：

**package** com. gxun.bean;

/\*Work表的信息记录\*/

**public** **class** WorkBean **implements** Serializable{

**private** **String** wid; /\*作业号（主） \*/

**private** **String** wname; /\*名称\*/

**private** **String** tid; /\*工号ID\*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/}

}

SubmitBean的实体如下：

**package** com. gxun.bean;

/\*Submit表的信息记录\*/

**public** **class** SubmitBean **implements** Serializable{

**private** **String** num; /\*编号（主） \*/

**private** **String**  wid; /\*作业号\*/

**private** **String** sid; /\*学号（外）\*/

**private** **String** answer; /\*学生答案\*/

**private** **int** grade; /\*分数\*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/}

}

# 公共部分模块详细设计

## 6.1、登录模块

在登录模块时，登录模块系统内部的相应响应操作示意图如图6-1所示。

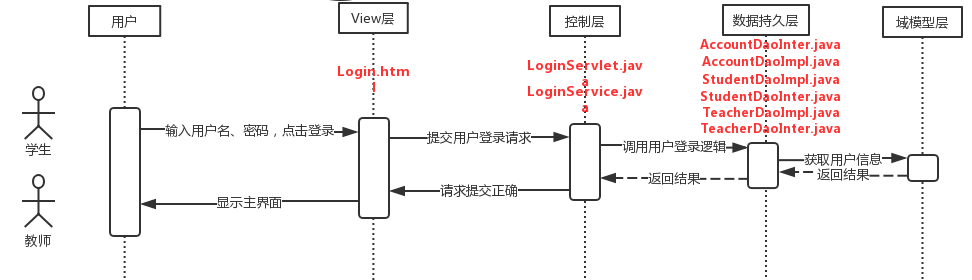


图6-1登录模块系统响应示意图

### 6.1.1 表现层

登录模块的表现层主要是完成用户的登录操作，用户输入账号、密码，选择角色（学生或教师）后，点击登录，根据用户选择的角色去数据库的相应表匹配账号、密码，若存在该用户，跳转到（学生或教师）页面；否则提示登录失败相应错误。表现层对应的HTM页面列表见具体见表6-2。

|  |  |  |
| --- | --- | --- |
| 界面 | HTM | 功能描述 |
| 登录页面 | Login.htm | 登入时根据用户的角色且用户存在跳转相应界面，出错时给出提示 |

表 6-2 登录模块表现层HTML列表

Login.htm的流程图如图6-3所示。

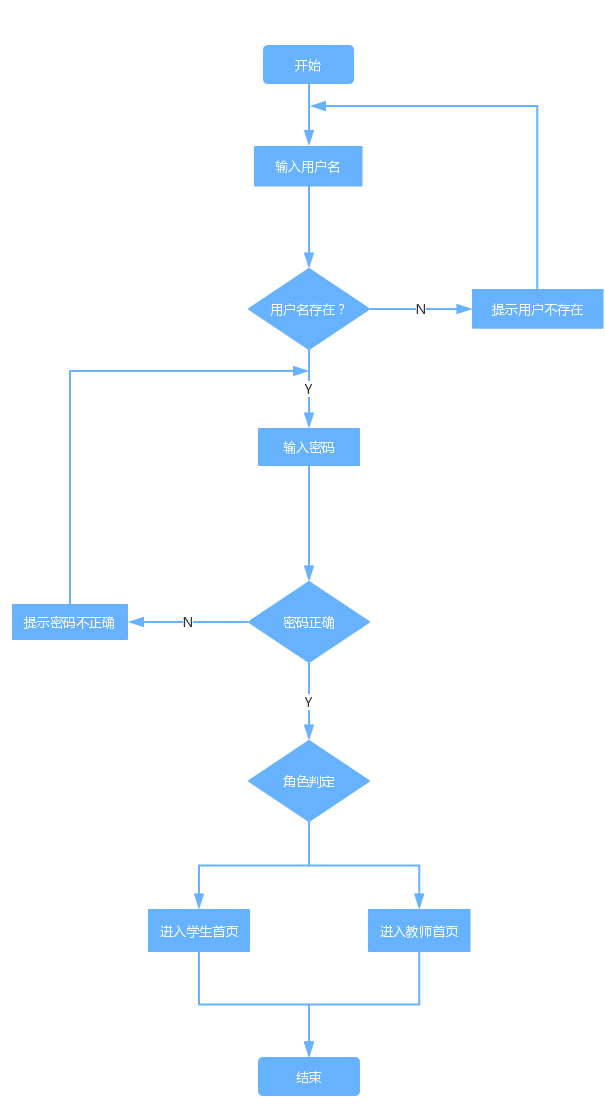


图6-3 登录模块详细设计流程图

### 6.1.2 控制层

登录模块的控制层负责接受来自Login.htm的用户输入，同时调用登录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。登录模块控制层列表见表6-4所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Action | 转移说明 | 出口 |
| 登录 | LoginServlet.java | SUCCESS | StudentFace.jsp或TeacherFace.jsp |
| ERROR | Error.jsp---登录失败，显示提示窗口。 |

表 6-4 登录模块控制层列表

在控制层中**LoginServlet**.java的描述如下所示：

**package** com.gxun.servlet;

**public class LoginServlet extends HttpServlet {**

**private SystemService service = new SystemService();**

**protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {}**

**protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {}**

**/\*\***

**\* 验证用户登录**

**\* @param request**

**\* @param response**

**\* @throws IOException**

**\*/**

**private void login(HttpServletRequest request, HttpServletResponse response) throws IOException {}**

}

### 6.2.3 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6-5所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | Service | 调用说明 | 出口 |
| 登录 | LoginService.java | 调用AccontDaoImpl.java  调用StudentDaoImpl.java  TeacherDaoImpl.java | 返回给Servlet |

表 6-5登录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IUserService接口，同时在实现该接口。

在控制层中LoginService.java的描述如下所示：

**package** com.gxun.service;

**public class** LoginService **extends HttpServlet {**

AccontDaol**nter accountDao = new** AccontDaoImpl **();**

StudentDaol**nter** student**Dao = new** StudentDaoImpl **();**

TeacherDaol**nter** teacher**Dao = new** TeacherDaoImpl **();**

**/\*\***

**\* 验证用户登录**

**\* @param request**

**\* @param response**

**\* @throws IOException**

**\*/**

**Public void login(String id){}**

}

### 6.2.4 数据持久层

AccontDaoInter.java代码结构：

**package** com.gxun.dao;

/\*\*

\* 账号数据操作类

\* @author hysd

\*/

**public interface** AccontDaoInter {

public AccountBean query(String aid);

}

AccontDaoImpl.java对AccontDaoInter.java实现：

**package** com.gxun.dao;

/\*\*

\* 账号数据操作类

\* @author hysd

\*/

*@Repository("baseDAO")*

*@SuppressWarnings("all")*

**public** class AccontDaoImpl implements AccontDaoInter {

public AccountBean query(String aid){};

}

StudentDaoInter.java代码结构：

**package** com.gxun.dao;

/\*\*

\* 账号数据操作类

\* @author hysd

\*/

**public interface** StudentDaoInter {

public StudentBean query(String sid);

}

StudentDaoImpl.java对StudentDaoInter.java实现：

**package** com.gxun.dao;

/\*\*

\* 账号数据操作类

\* @author hysd

\*/

*@Repository("baseDAO")*

*@SuppressWarnings("all")*

**public** class StudentDaoImpl implements StudentDaoInter{

public StudentBean query(String sid){};

}

TeacherDaoInter.java代码结构：

**package** com.gxun.dao;

/\*\*

\* 账号数据操作类

\* @author hysd

\*/

**public interface** TeacherDaoInter {

public TeacherBean query(String tid);

}

TeacherDaoImpl.java对TeacherDaoInter.java实现：

**package** com.gxun.dao;

/\*\*

\* 账号数据操作类

\* @author hysd

\*/

*@Repository("baseDAO")*

*@SuppressWarnings("all")*

**public** class TeacherDaoImpl implements TeacherDaoInter{

public TeacherBean query(String tid){};

}

### 6.2.5 域模型层

登录模块用到域模型层中AccountBean.java、StudentBean.java和TeacherBean.java，登录模块域模型层列表如表6-7所示。

|  |  |
| --- | --- |
| 域模型 | 描述 |
| AccountBean.java | 账号实体 |
| StudentBean.java | 学生实体 |
| TeacherBean.java | 教师实体 |

表 6-7登录模块域模型层列表

AccountBean.java主要属性与方法：

**package** com. gxun.bean;

/\*\*

\* *@description* 账号表的信息记录

\*/

**public** **class** AccountBean **implements** Serializable{

**private** **String** aid; /\*账号\*/

**private** **String** pwd; /\*密码\*/

**private** **String** stid; /\*学工号 \*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

StudentBean.java主要属性与方法：

**package** com. gxun.bean;

/\*\*

\* *@description* 学生表的信息记录

\*/

**public** **class** StudentBean **implements** Serializable{

**private** **String** sid; /\*学号\*/

**private** **String** sname; /\*姓名\*/

**private** **String** simg; /\*头像\*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

TeacherBean.java主要属性与方法：

**package** com. gxun.bean;

/\*\*

\* *@description* 教师表的信息记录

\*/

**public** **class** TeacherBean **implements** Serializable{

**private** **String** tid; /\*工号\*/

**private** **String** tname; /\*姓名\*/

**private** **String** timg; /\*头像 \*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}