|  |  |
| --- | --- |
| **项目编号** | **ZYPI001** |
| **文档编号** | **004** |
| **密级** | **秘密** |

**作业批改系统详细设计**

**V3.0**

**广西民族大学**

评审日期： 2019年6月28日

目录

[1．导言 4](#_Toc13181292)

[1.1 目的 4](#_Toc13181293)

[1.2 范围 4](#_Toc13181294)

[1.3 引用标准 4](#_Toc13181295)

[1.4 参考资料 5](#_Toc13181296)

[1.5 版本更新信息 5](#_Toc13181297)

[2．详细设计简述 5](#_Toc13181298)

[2.1 设计简介 5](#_Toc13181299)

[2.2 模块简介 6](#_Toc13181300)

[3、界面详细设计 7](#_Toc13181301)

[3.1、页面设计说明 7](#_Toc13181302)

[3.2、静态页面详细设计 7](#_Toc13181303)

[4、公共部分接口设计 8](#_Toc13181304)

[5、学生端模块详细设计 8](#_Toc13181305)

[5.1、提交作业模块详细设计 8](#_Toc13181306)

[5.1.1 表现层 9](#_Toc13181307)

[5.1.2 控制层 11](#_Toc13181308)

[5.1.3 业务逻辑层 12](#_Toc13181309)

[5.1.4 数据持久层 12](#_Toc13181310)

[5.1.5 域模型层 14](#_Toc13181311)

[5.2成绩查询模块详细设计 16](#_Toc13181312)

[5.2.1 表现层 16](#_Toc13181313)

[5.2.2 控制层 17](#_Toc13181314)

[5.2.3 业务逻辑层 18](#_Toc13181315)

[5.2.4 数据持久层 19](#_Toc13181316)

[5.2.5 域模型层 20](#_Toc13181317)

[6、 教师端模块详细设计 21](#_Toc13181318)

[6.1、作业发布模块详细设计 21](#_Toc13181319)

[6.1.1 表现层 21](#_Toc13181320)

[6.1.2 控制层 22](#_Toc13181321)

[6.1.3 业务逻辑层 24](#_Toc13181322)

[6.1.4 数据持久层 25](#_Toc13181323)

[6.1.5 域模型层 28](#_Toc13181324)

[6.2、作业批改模块详细设计 30](#_Toc13181325)

[6.2.1 表现层 30](#_Toc13181326)

[6.2.2 控制层 31](#_Toc13181327)

[6.2.3 业务逻辑层 33](#_Toc13181328)

[6.2.4 数据持久层 34](#_Toc13181329)

[6.2.5 域模型层 35](#_Toc13181330)

[6.3、成绩查询模块详细设计 36](#_Toc13181331)

[6.3.1 表现层 37](#_Toc13181332)

[6.3.2 控制层 38](#_Toc13181333)

[6.3.3 业务逻辑层 39](#_Toc13181334)

[6.3.4 数据持久层 40](#_Toc13181335)

[6.3.5 域模型层 41](#_Toc13181336)

[7、 公共部分模块详细设计 43](#_Toc13181337)

[7.1、登录模块 43](#_Toc13181338)

[7.1.1 表现层 43](#_Toc13181339)

[7.1.2 控制层 44](#_Toc13181340)

[7.1.3 业务逻辑层 45](#_Toc13181341)

[7.1.4 数据持久层 46](#_Toc13181342)

[7.1.5 域模型层 49](#_Toc13181343)

# 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 | 修改 | 第二发布版本 |
| 003 | 2019.6.28 | 3.0 | 增添4、公共部分接口设计 | 第三发布版本 |

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。

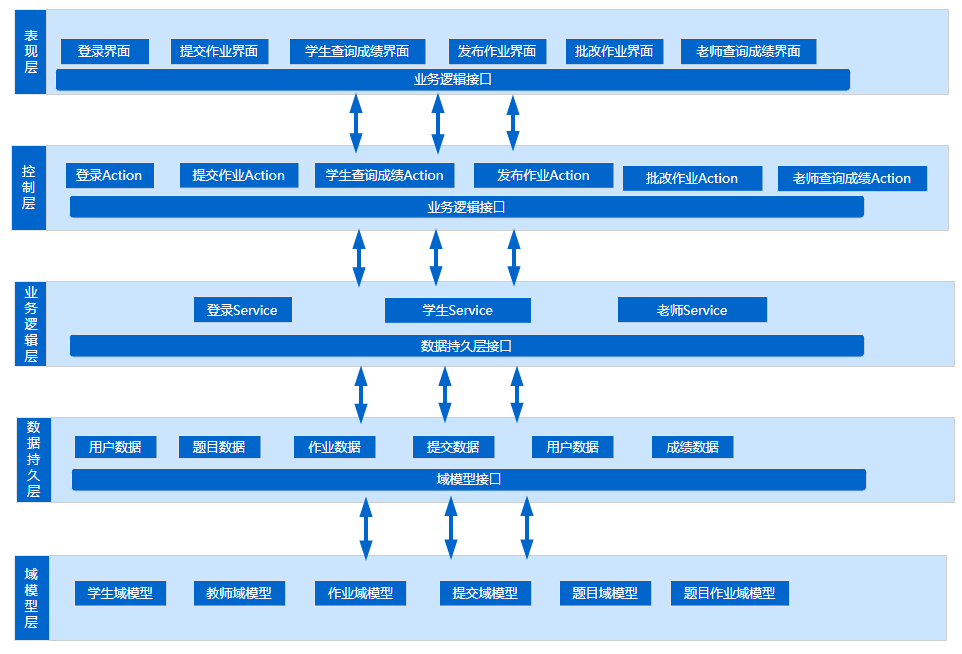


图 2-1 作业批改系统分层结构示意图

## 2.2 模块简介

系统功能模块图如图2-2所示。

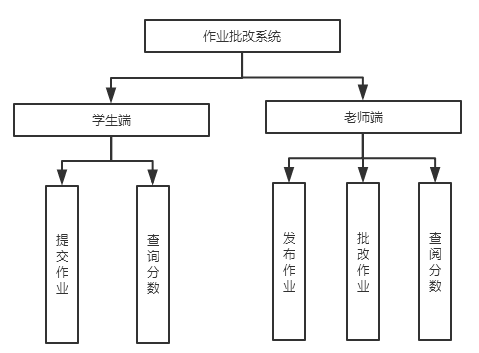


图2—2作业批改系统功能模块图

# 3、界面详细设计

## 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 |
| 教师首页 | TeacherFace |  |
| 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 |

表3-1 页面表现层设计

# 4、公共部分接口设计

在审查过程中在数据持久层发现有很多的代码都是重复用到的，故将它们抽象出来，独立放在一个接口里，降低了代码的冗余度。

实现如下：

**package** com.gxun.servlet;

/\*\*\* *@description* 实现学生控制\*/

**public interface BaseDaoInter {**

**List<Object> getList(Class type, String sql);**

**List<Object> getList(Class type, String sql, Object[] param);**

**Object getObject(Class type, String sql, Object[] param);**

**void insertTransaction(Connection conn, String sql, Object[] param) throws SQLException;**

**void updateTransaction(Connection conn, String sql, Object[] param) throws SQLException;**

# 5、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。

## 5.1、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时序图如图4-1所示。

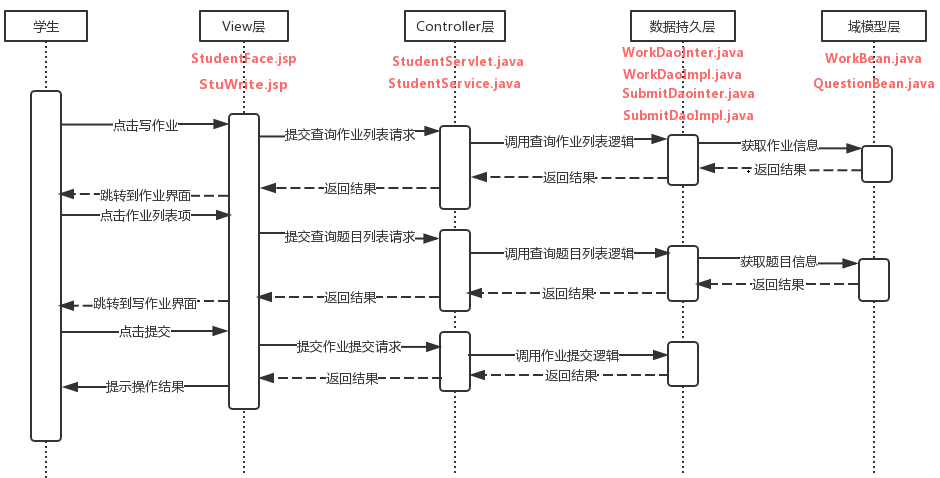


图 4-1 提交作业时序图

### 5.1.1 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。

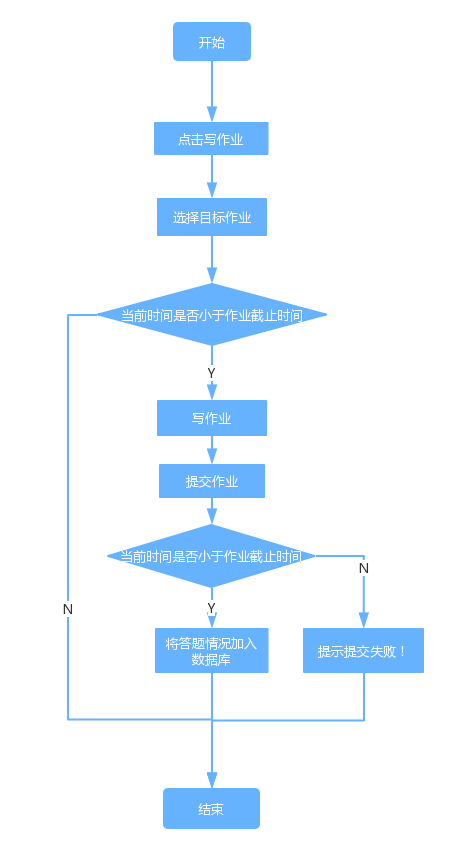


图 4-3 学生提交作业模块详细设计流程图

### 5.1.2 控制层

学生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返回到控制层再传到表现层，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 service = new StudentService(); //创建服务层对象**

**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 addSubmit (HttpServletRequest request, HttpServletResponse response){} //提交作业**

}

### 5.1.3 业务逻辑层

提交作业模块的业务逻辑层主要是调用提交作业模块的数据持久层接口对作业、题目信息进行查询，并将结果返回给控制层，同时执行增添作业信息的操作。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 ();**

**QuestionDaoInter questionDao = new QuestionDaoImpl();**

**/\*\*获取作业列表 @return 作业列表\*/**

**public String getWokeList(){}**

**/\*\*提交作业 @return 作业列表\*/**

**public void addSubmit(SubmitBean submit){}**

**/\*\* 获取作业题目信息 @return\*/**

**public String getQuestionList(){}**

**}**

### 5.1.4 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

在学生提交作业模块的业务逻辑层是调用了WorkDaoInter 、SubmitDaoInter接口，同时实现该接口

WorkDaoInter.java：

**package** com.gxun.dao;

/\*\* @description \*/

**public interface** WorkDaoInter {

**/\*\*获取作业列表 @return 作业列表\*/**

**List<WorkBean> getWorkList(Class type, String sql, Object[] param){}**

**/\*\* 获取作业题目信息 @return\*/**

**public List<QuestionBean> getQuestionList(String sql,Object[] param){}**

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

**package com.gxun.dao;**

**public class WorkDaoImpl implements WorkDaoInter {**

**/\*\*获取作业列表 @return 作业列表\*/**

**List<WorkBean> getWorkList(Class type, String sql, Object[] param){}**

**/\*\* 获取作业题目信息 @return\*/**

**public List<QuestionBean> getQuestionList(String sql,Object[] param){}**

SubmitDaoInter.java：

**package** com.gxun.dao;

**public interface** SubmitDaoInter {

**public void addSubmit(String sql,Object[] param);** /\*对提交数据进行增加\*/

}

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

**package** com.gxun.dao;

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

**public void addSubmit(String sql,Object[] param) {}** //对提交数据进行增加

}

### 5.1.5 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 \*/

**private String** starttime; /\*开始时间\*/

**private String** endtime; /\*截止时间\*/

/\*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 \*/

**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\*/

}

## 5.2成绩查询模块详细设计

在学生作业查询模块中，作业查询模块系统内部的时序图如图4-8所示。

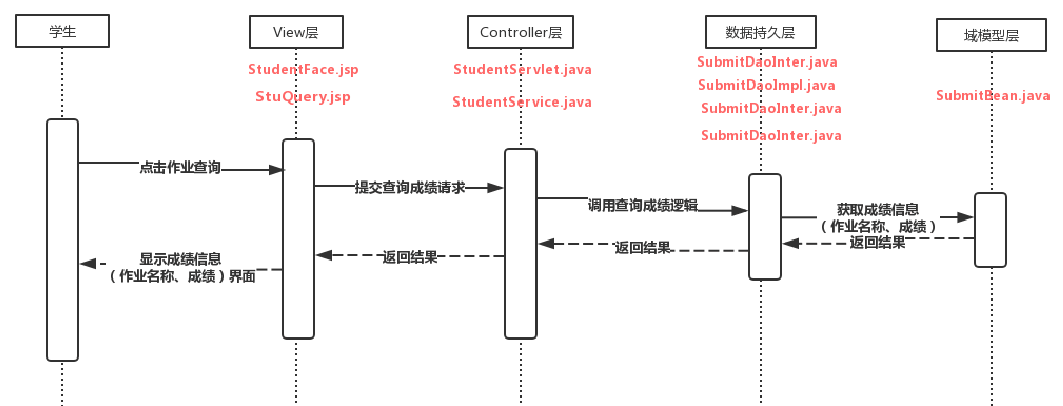


图 4-8 学生作业查询模块时序图

### 5.2.1 表现层

成绩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，学生点击成绩查询时，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。

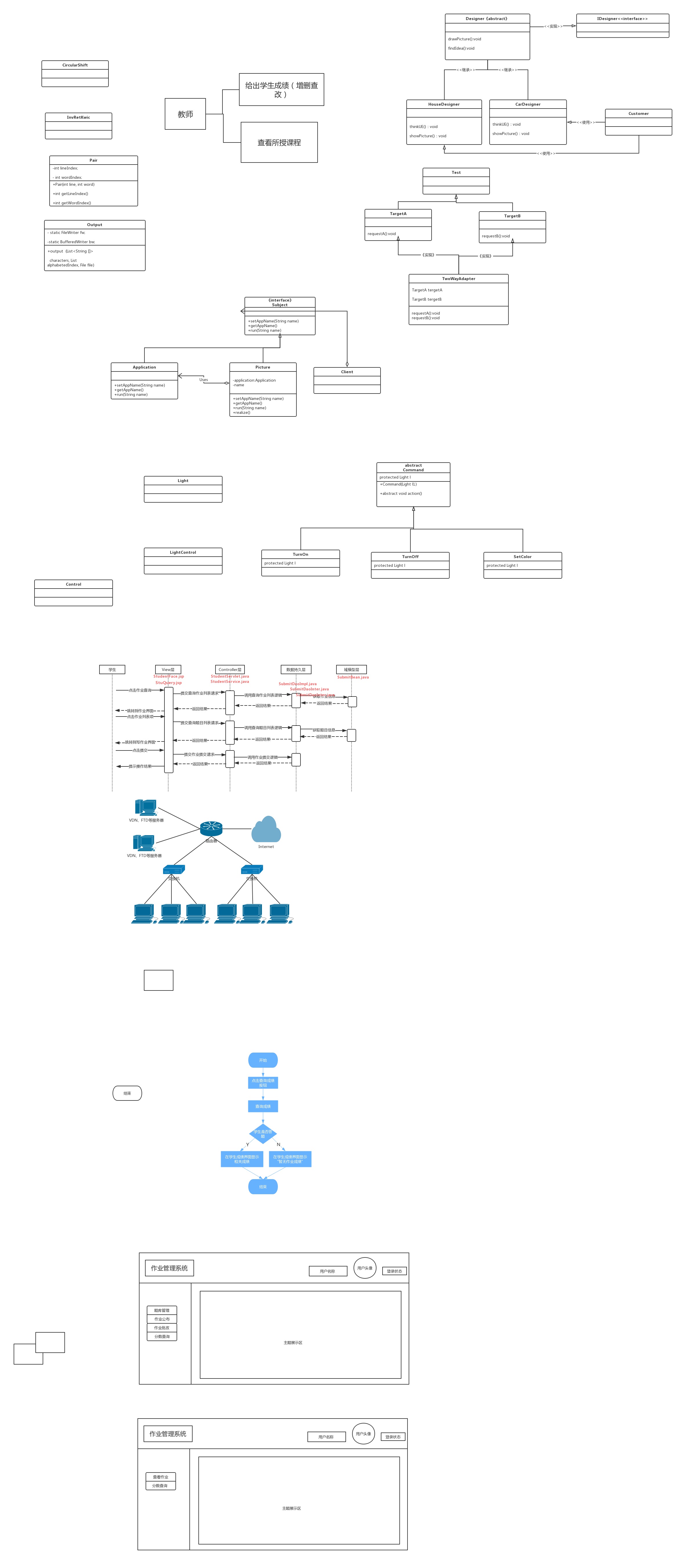


图 4-10 作业查询模块系统流程图

### 5.2.2 控制层

学生成绩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。学生成绩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—成绩查询窗口。 |
| ERROR | Error.jsp---成绩查询失败，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 service = new StudentService();**

**private void getGradeList(HttpServletRequest request,HttpServletResponse response) throws IOException {}**  //获取该学生的全部成绩

}

### 5.2.3 业务逻辑层

学生成绩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 {**

**MarkDaoInter markDao = new MarkDaoImpl();**

**public String getStuScoreList(String sid){} //获取当前学生的全部作业成绩**

**}**

### 5.2.4 数据持久层

学生作业查询模块的数据持久层能对数据库中的mark表进行查询操作。教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 | MarkDaoInter.java  MarkDaoIpml.java | 调用markBean.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

MarkDaoInter.java定义了查询成绩的接口

**package** com.gxun.dao;

**public interface** MarkDaoInter {

**public List<SubmitBean> queryStuScoreList(Class<SubmitBean> class1, String string, Object[] objects);** }

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

**package** com.gxun.dao;

**public** **class** MarkDaoIpml implements MarkDaoInter {

**public List<SubmitBean> queryStuScoreList(Class type, String sql,Object[] param) {** }

}

### 5.2.5 域模型层

学生作业查询模块用到域模型层中MarkBean.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 |
| MarkBean.java | 提交实体 |

表 4-14学生作业查询模块域模型层列表

MarkBean.java主要属性与方法：

**package** com. gxun.bean;

/\*\*

\* *@description* MarkBean的信息记录

\*/

**public class MarkBean {**

**private int num;**

**private String sid;**

**private int grade;**

**/\*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。

## 6.1、作业发布模块详细设计

在作业发布模块时，提交作业系统内部的时序图如图5-1所示。

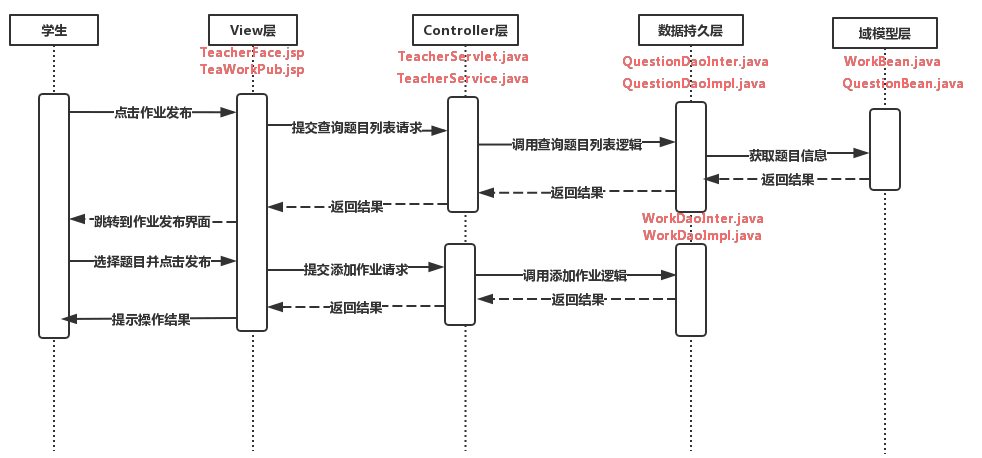


图 5-1 作业发布时序图

### 6.1.1 表现层

教师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。

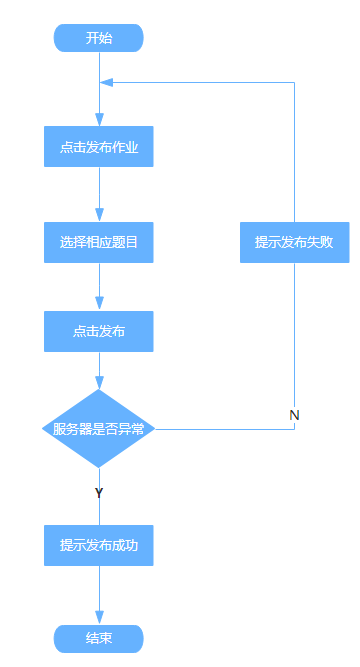


图 5-3 作业发布模块系统流程图

### 6.1.2 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的教师操作，同时调用教师发布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返回到控制层再传到表现层，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 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 getQuestionList(HttpServletRequest request, HttpServletResponse response){} //获取所有题目列表**

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

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

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

}

### 6.1.3 业务逻辑层

教师作业发布模块的业务逻辑层主要是调用教师作业发布模块的数据持久层接口对题目信息进行查询，并将结果返回给控制层，同时执行作业发布和题目添加的操作。教师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 |
| 教师  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

**package com.gxun.service;**

**/\*\***

**\* @description 实现教师业务逻辑**

**\*/**

**public class TeacherService {**

**QuestionDaoInter questionDao = new QuestionDaoImpl();**

**WorkDaoInter workDao = new WorkDaoImpl();**

**WQInter wqDao = new WQImpl();**

**public String getQuestionList() {} //获取题目信息**

**public void deleteQuestion(String[] ids) throws Exception{} //删除题目**

**public void addWork (WorkBean work,String[] ids) {} //作业发布**

**public void addQuestion (QuestionBean question) {} //题目添加**

**}**

### 6.1.4 数据持久层

在教师查询成绩模块的业务逻辑层是调用了QuestDaoInter接口查询出题目列表，同时调用WorkDaoInter和WQInter将相应的作业信息进行增加。教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  WQInter.java  WQImpl.java | 调用 workBean.Java  WQ.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定义了作业信息进行增、删、改、查的接口：

**package** com.gxun.dao;

**public interface** WorkDaoInter {

**public void insertTransaction(Connection conn, String sql, Object[] param) throws SQLExcepublic**

**public Object getObject(Class type, String sql, Object[] param);**

}

WorkDaoInter.java定义了题目信息进行增、删、改、查的接口：

**package** com.gxun.dao;

**public interface** QuestionDaoInter {

**public void insertTransaction(Connection conn, String sql, Object[] param) throws SQLException;**

**public List<QuestionBean> getQuestionList(String sql,Object[] param) ;**

**public void deleteTransaction(Connection conn, String sql, Object[] param)throws SQLException;**}

**package** com.gxun.dao;

**public interface** WorkDaoInter {

**public void insertTransaction(Connection conn, String sql, Object[] param) throws SQLExcepublic**

**public Object getObject(Class type, String sql, Object[] param);**

}

WQInter.java定义了题目信息进行增、删、改、查的接口：

**package** com.gxun.dao;

**public interface** WQInter {

**public void insertTransaction(Connection conn, String sql, Object[] param) throws SQLException;** }

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

**package** com.gxun.dao;

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

**public void insertTransaction(Connection conn, String sql, Object[] param) throws SQLExcepublic{}**

**public Object getObject(Class type, String sql, Object[] param){}**}

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

**package** com.gxun.dao;

**public** **class** QuestionDaoImpl implements QuestDaoInter {

**public void insertTransaction(Connection conn, String sql, Object[] param) throws SQLException{}**

**public List<QuestionBean> getQuestionList(String sql,Object[] param) {}**

**public void deleteTransaction(Connection conn, String sql, Object[] param)throws SQLException{}}**

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

**package** com.gxun.dao;

**public interface** WQimpl {

**public void insertTransaction(Connection conn, String sql, Object[] param) throws SQLException;** }

### 6.1.5 域模型层

老师作业发布模块用到域模型层中WorkBean.java，QuestionBean.java,WQ,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 |
| WQ | 作业和题目关系实体 |

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的信息记录

\*/

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

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

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

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

**private String** starttime; /\*开始时间\*/

**private String** endtime; /\*结束时间\*/

/\*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的信息记录

\*/

**public** **class** QuestionBean **implements** Serializable{

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

**private** **String** qcontent; /\*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\*/

}

**package** com. gxun.bean;

/\*\*

\* *@description* Question的信息记录

\*/

**public** **class** WQ **implements** Serializable{

**private int wqid;**

**private int qid; /\*题号\*/**

**private int wid; /\*作业号\*/**

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

## 6.2、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时序图如图所示。

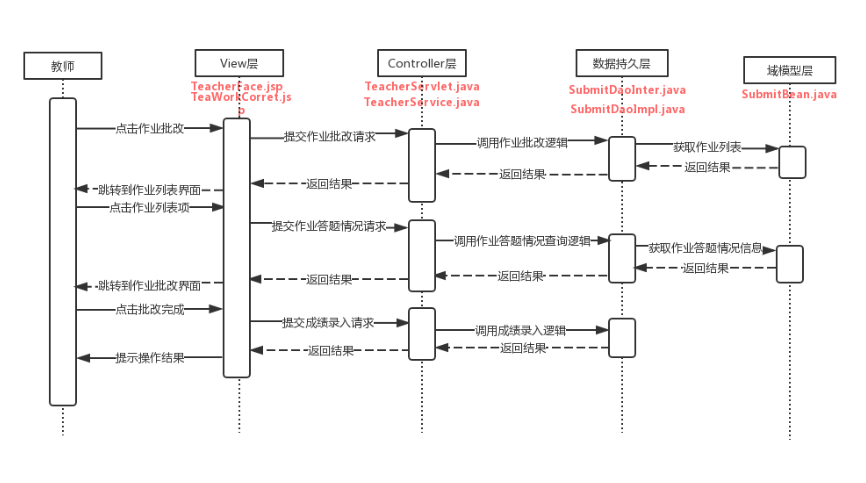


图 5-8 教师作业批改时序图

### 6.2.1 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。

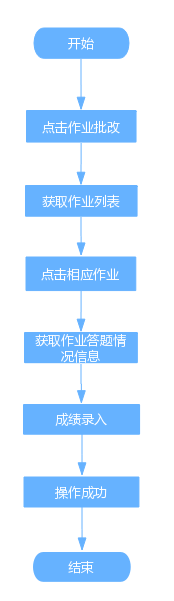


图 5-10 教师作业批改详细设计流程图

### 6.2.2 控制层

教师作业批改模块控制层负责接受来自TeacherFace.jspTeaWorkCorrect.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 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 getSubmitList(HttpServletRequest request, HttpServletResponse response)throws IOException**

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

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

}

### 6.2.3 业务逻辑层

教师作业批改模块的业务逻辑层主要是调用教师作业批改模块的数据持久层接口对作业表、提交表进行查询，并将结果返回给控制层，同时执行批改作业的操作。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WorkDaoInter接口查询出该教师发布的作业列表，同时调用SubmitDaoInter接口查询出学生的提交信息并将每道题的成绩进行录入。

TeacherService.java接口主要方法：

**package** com.gxun.service;

/\*\*

\* *@description* 实现教师业务逻辑

\*/

**public class TeacherService {**

**WorkDaoInter workDao = new WorkDaoImpl();**

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

**public String getWokeList(String tid){**

**public void updateSubmit (int grade,int wid,int qid,String sid) {}**

**public String getSubmitList (String wid){}**

**}**

### 6.2.4 数据持久层

教师成绩查询模块的数据持久层能对数据库中的work表进行查询操作，对submit进行查询和插入操作。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定义了作业的接口：

**package** com.gxun.dao;

**public interface** WorkDaoInter {

**public List<WorkBean> queryWorkList(Class type,String sql,Object[] param) ;**}

SubmitDaoInter.java定义了教师批改作业的接口：

**package** com.gxun.dao;

**public interface** SubmitDaoInter {

**public List<SubmitBean> getSubmitList(Class type, String sql,Object[] param);** **public void updateTransaction(Connection conn, String sql, Object[] param) throws SQLException;**}

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(Class type,String sql,Object[] param){}**}

**package** com.gxun.dao;

**public** **class** SubmitDaoImpl implements WorkDaoInter {

**public List<SubmitBean> getSubmitList(Class type, String sql,Object[] param){}**

**public void updateTransaction(Connection conn, String sql, Object[] param) throws SQLException{}**}

### 6.2.5 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** SubmitBean **implements** Serializable{

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

**private** **String** wid; /\*作业号\*/

**private int** qid; /\*题号\*/

**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\*/

}

## 6.3、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时序图如图5-15所示。

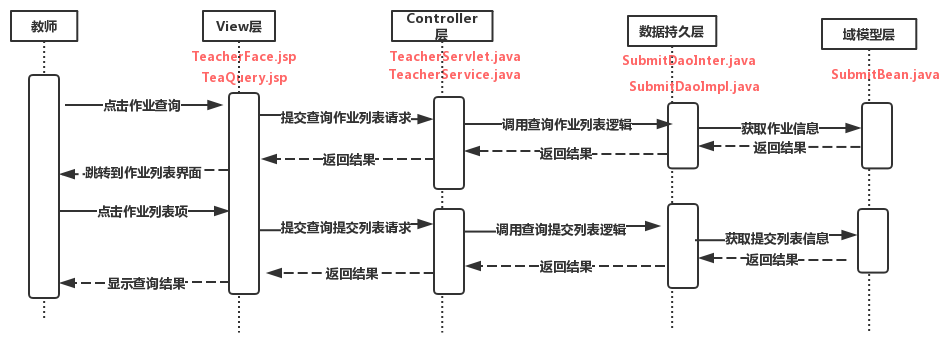


图 5-15 教师成绩查询时序图

### 6.3.1 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。

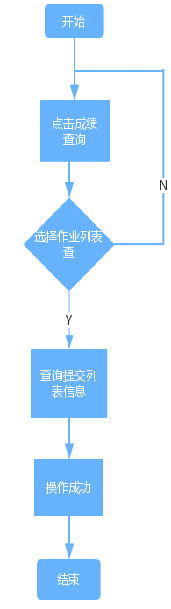


图 5-17 教师成绩查询模块系统流程图

### 6.3.2 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 getGradeList(HttpServletRequest request, HttpServletResponse response)throws IOException {}**

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

}

### 6.3.3 业务逻辑层

成绩查询模块的业务逻辑层主要是调用成绩查询模块的数据持久层接口对当前老师发布的作业进行查询，并将结果返回给控制层，同时执行获取成绩的操作。教师成绩查询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

在教师查询成绩模块的业务逻辑层是调用了WorkDaoInter接口查询出该教师发布的作业列表，同时调用MarkDaoInter查询出所选作业的学生成绩。

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 {**

**WorkDaoInter** workDao = new **WorkDaoImpl**();

**MarkDaoInter** markDao **=** new **MarkDaoImpl();**

**public String getWokeList**(String tid){} //获取当前老师发布的作业

**public String getMarkList** (String wid){} //获取成绩

**}**

### 6.3.4 数据持久层

教师成绩查询模块的数据持久层能对数据库中的mark表和work表进行查询操作，获取相应的信息。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 |
| MarkDaoInter.java  MarkDaoIpml.java | 调用markBean.java | 对成绩数据进行查询 |

表 5-20 教师成绩查询模块数据持久层表

WorkDaoInter.java定义了作业信息进行增、删、改、查的接口：

**package** com.gxun.dao;

**public interface** WorkDaoInter {

**public List<WorkBean> queryWorkList(Class type,String sql,Object[] param) ;**}

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

**package** com.gxun.dao;

@Repository

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

**public List<WorkBean> queryWorkList(Class type,String sql,Object[] param) ;**}

MarkDaoInter.java定义了作业信息进行增、删、改、查的接口：

**package** com.gxun.dao;

@Repository

**public** **interface** MarkDaoInter {

**public List<SubmitBean> queryStuScoreList(Class<SubmitBean> class1, String string, Object[] objects);** }

MarkDaoImpl.java是MarkDaoInter.java接口的实现：

**package** com.gxun.dao;

@Repository

**public** **class** MarkDaoImpl implements MarkDaoInter {

**public List<SubmitBean> queryStuScoreList(Class<SubmitBean> class1, String string, Object[] objects){}**

### 6.3.5 域模型层

教师成绩查询模块用到域模型层中WorkBean.java和MarkBean.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 |
| MarkBean.java | 成绩表实体 |

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; /\*id识别\*/

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

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

**private String** starttime; /\*开始时间\*/

**private String** endtime; /\*结束时间\*/

/\*编写对上述各个属性的*getter*与*setter*方法\*/}

}

MarkBean的实体如下：

**package** com. gxun.bean;

/\*Submit表的信息记录\*/

**public** **class** MarkBean**implements** Serializable{

**private int num;**

**private String sid;**

**private int grade;**

/\*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

## 7.1、登录模块

在登录模块时，登录模块系统内部的时序图如图6-1所示。

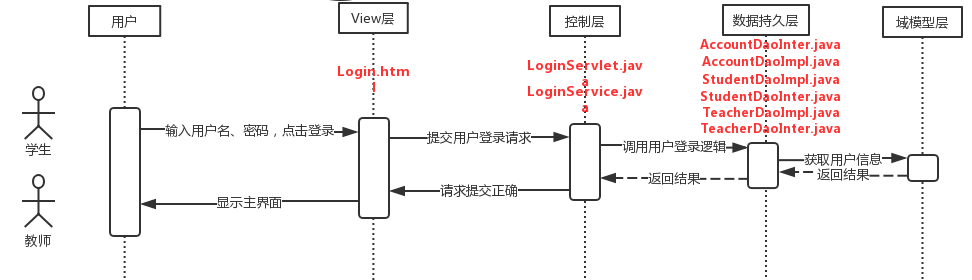


图6-1登录模块系统时序图

### 7.1.1 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。

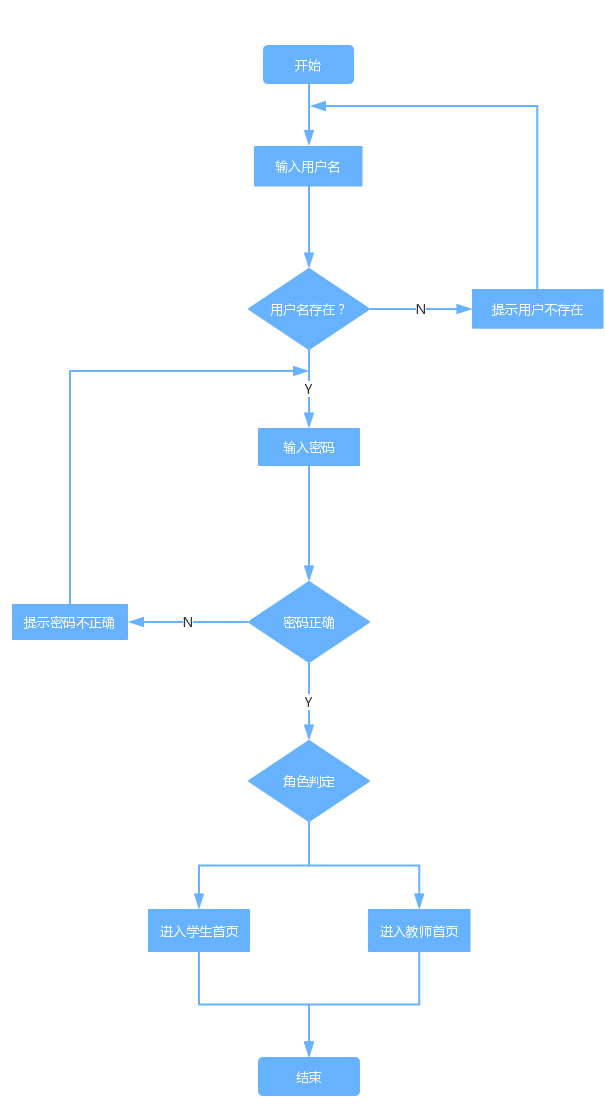


图6-3 登录模块详细设计流程图

### 7.1.2 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 static final long serialVersionUID = 1L;**

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

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

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

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

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

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

}

### 7.1.3 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 {**

AccountDaoInter accountDao = new AccountDaoImpl();

TeacherDaoInter teacherDao = new TeacherDaoImpl();

StudentDaoInter studentDao = new StudentDaoImpl();

**public String login(AccountBean account){}** //验证用户登录

**public TeacherBean Tlogin(String tid){}** //获取登录的教师信息

**public StudentBean Slogin(String sid){}** //获取登录的学生信息

}

### 7.1.4 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 Object getObject(Class type, String sql, Object[] param);**

}

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 Object getObject(Class type, String sql, Object[] param) {}**

}

StudentDaoInter.java代码结构：

**package** com.gxun.dao;

/\*\*

\* 账号数据操作类

\* @author hysd

\*/

**public interface** StudentDaoInter {

**public Object getObject(Class type, String sql, Object[] param);**

}

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 Object getObject(Class type, String sql, Object[] param) { }**

}

TeacherDaoInter.java代码结构：

**package** com.gxun.dao;

/\*\*

\* 账号数据操作类

\* @author hysd

\*/

**public interface** TeacherDaoInter {

**public Object getObject(Class type, String sql, Object[] param);**

}

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 Object getObject(Class type, String sql, Object[] param) { };**

}

### 7.1.5 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\*/

}