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

**XXX详细设计**

**V1.0**

**广西民族大学**

评审日期： 2016年2月28日

目录

[1．导言 4](#_Toc444248412)

[1.1 目的 4](#_Toc444248413)

[1.2 范围 4](#_Toc444248414)

[1.3 引用标准 5](#_Toc444248415)

[1.4 参考资料 5](#_Toc444248416)

[1.5 版本更新信息 5](#_Toc444248417)

[2．详细设计简述 6](#_Toc444248418)

[2.1 设计简介 6](#_Toc444248419)

[2.2 模块简介 6](#_Toc444248420)

[3、界面详细设计 7](#_Toc444248421)

[3.1、页面设计说明 7](#_Toc444248422)

[3.2、静态页面详细设计 7](#_Toc444248423)

[4、客户端模块详细设计 7](#_Toc444248424)

[4.1、公用模块的详细设计 7](#_Toc444248425)

[4.1.1 表现层 8](#_Toc444248426)

[4.1.2 控制层 8](#_Toc444248427)

[4.1.3 数据持久层 9](#_Toc444248428)

[4.2、登录模块 12](#_Toc444248429)

[4.2.1 表现层 12](#_Toc444248430)

[4.2.2 控制层 14](#_Toc444248431)

[4.2.3 业务逻辑层 15](#_Toc444248432)

[4.2.4 数据持久层 17](#_Toc444248433)

[4.2.5 域模型层 19](#_Toc444248434)

[4.n、XXXX模块 20](#_Toc444248435)

[4.n.1 表现层 20](#_Toc444248436)

[4.n.2 控制层 21](#_Toc444248437)

[4.n.3 业务逻辑层 21](#_Toc444248438)

[4.n.4 数据持久层 21](#_Toc444248439)

[4.n.5 域模型层 21](#_Toc444248440)

[5、管理端模块详细设计 21](#_Toc444248441)

[5.1、XXXXX模块详细设计 21](#_Toc444248442)

[5.1.1 表现层 21](#_Toc444248443)

[5.1.2 控制层 21](#_Toc444248444)

[5.1.3 业务逻辑层 21](#_Toc444248445)

[5.1.4 数据持久层 22](#_Toc444248446)

[5.1.5 域模型层 22](#_Toc444248447)

[5.n、XXXXX模块详细设计 22](#_Toc444248448)

[5.n.1 表现层 22](#_Toc444248449)

[5.n.2 控制层 22](#_Toc444248450)

[5.n.3 业务逻辑层 23](#_Toc444248451)

[5.n.4 数据持久层 23](#_Toc444248452)

[5.n.5 域模型层 23](#_Toc444248453)

[6、公共部分模块详细设计 23](#_Toc444248454)

[6.1 公共页面 23](#_Toc444248455)

[6.2 安全模块的详细设计 23](#_Toc444248456)

[6.n XXXXX 23](#_Toc444248457)

# 1．导言

## 1.1 目的

该文档是对XXXX概要设计进行的详细说明，是为了指导和规范XXXXX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XXXX系统的所有功能模块的详细设计描述，。。。。。。。

## 1.3 引用标准

[1] 《软件工程案例教程 第2版》 韩万江等 机械工业出版社

。。。。。。

## 1.4 参考资料

[1] 《软件项目管理案例教程 第3版》 韩万江等 机械工业出版社

。。。。。。

## 1.5 版本更新信息

本文档的更新记录如下表？-？所示。

表1-1 文档更新记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 修改编号 | 修改日期 | 修改后版本 | 修改位置 | 修改内容概述 |
| 001 | 2015.4.3 | 0.1 | 全部 | 初始发布版本 |
| 002 | 2015.4.12 | 1.0 | 4章节 | 增加 |
| 003 | 2015.4.16 | 2.0 | 4章节 | 修改 |
| 004 | 2015.4.23 | 3.0 | 4、6章节 | 修改 |
| 005 | 2015.4.25 | 4.0 | 6、7章节 | 修改、增加 |
| 006 | 2015.4.28 | 5.0 | 5、7章节 | 修改、增加 |
| 007 | 2015.6.3 | 6.0 | 4、6章节 | 修改 |
| 008 | 2015.6.8 | 7.0 | 5、7章节 | 修改 |
| 009 | 2015.12.17 | 8．0 | 全部 | 修改，增加 |

# 2．详细设计简述

## 2.1 设计简介

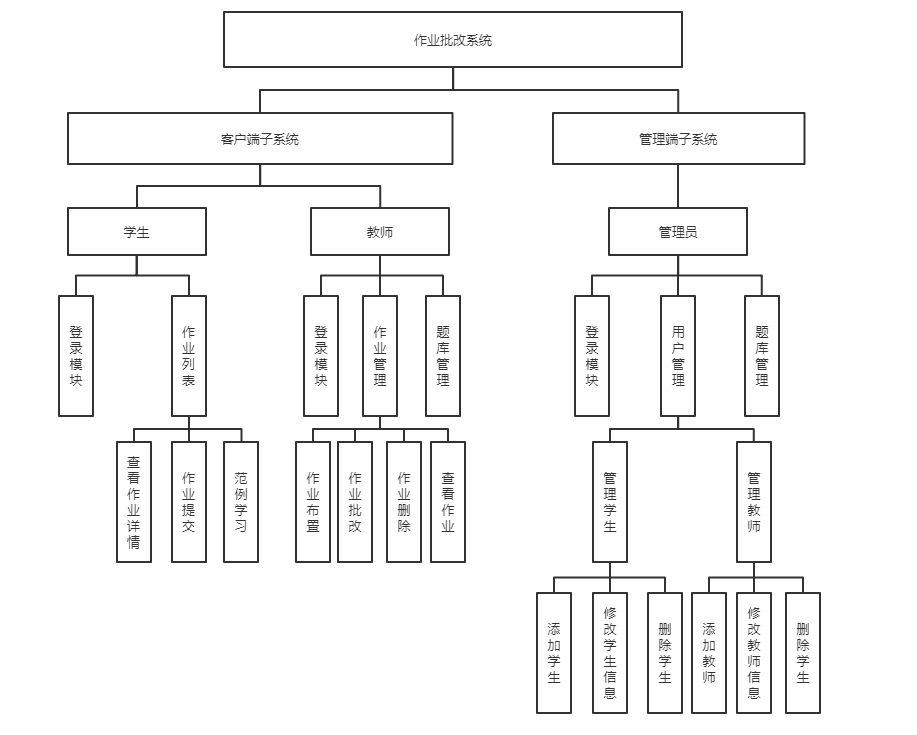
。。。。。。。。。分层结构图如图2-1所示。

图 2-1 XXXX示意图

。。。。。。。。。

## 2.2 模块简介

系统功能模块图如图？-？所示。



[对某个具体模块的详细设计可以采用流程图或者伪代码形式]

# 3、界面详细设计

[界面详细设计 包括界面设计的技术，以及静态页面设计]

## 3.1、页面设计说明

。。。。。。。。

## 3.2、静态页面详细设计

作业批改系统静态页面见表？-？

表3-1 页面表现层设计

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 登录界面 | login.jsp | 提供给用户输入账号、密码并登录 |
| 管理员界面 | admin.jsp | 提供教师管理、学生管理、班级管理、课程安排功能 |
| 学生作业主界面 | studentWork.jsp | 显示学生作业列表，点击查看作业详情、查看作业答案 |
| 学生作业详细界面 | studentWorkDetail.jsp | 提供给学生显示作业详情（完整题目、答案） |
| 教师界面 | teacherWork.jsp | 提供给教师显示发布作业列表，提供作业发布、删除、详情查看功能，题库管理功能 |
| 教师发布作业界面 | workPublish.jsp | 提供给教师从题库选择题目、自定义题目并发布 |
| 教师题库管理界面 | libraryManagement.jsp | 提供给教师查看题库题目、删除题库题目、添加题库题目 |
| 教师作业详情界面 | teacherWorkDetail.jsp | 提供给教师查看已发布作业学生提交情况、查看分数统计、批改学生作业、 |
| 修改密码界面 | passwordModify | 提供给教师、学生修改密码 |
|  |  |  |

。。。。。。。。。。

# 4、客户端模块详细设计

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

## 4.1、公用模块的详细设计

[公用模块是指在系统在运行过程中将会被其他模块调用的基础资源]。

### 4.1.1 表现层

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

表 4-1 公用模块表现层

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 统一异常页面 | exceptionPage.jsp | 在系统运行时出现未知错误时跳转到该页面。 |

### 4.1.2 控制层

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

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

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

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

**private** transient TextProvider textProvider;

**private** Container container;

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

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

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

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

}

### 4.1.3 数据持久层

。。。。。。。IBaseDAO.java代码结构：

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

**import** java.io.Serializable;

**import** java.util.List;

/\*\*

\* 基础数据库操作类

\* @author hysd

\*/

**public interface** IBaseDAO<T> {

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

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

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

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

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

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

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

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

\*/

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

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

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

}

BaseDAOImpl.java对IBaseDAO.java实现：

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

/\*\*

\* 基础数据库操作类

\* @author hysd

\*/

*@Repository("baseDAO")*

*@SuppressWarnings("all")*

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

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

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

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

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

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

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

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

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

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

\*/

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

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

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

}

## 4.2、登录模块

在登录模块时，系统内部的相应响应操作示意图如图4-2所示。

图 4-2 系统响应示意图

### 4.2.1 表现层

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

表 4-3 登入模块表现层JSP列表

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

login.jsp的流程图如图4-3所示。

进入游首页



图 4-3 登录模块详细设计流程图

### 4.2.2 控制层

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

表 4-4 登入模块控制层列表

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

在控制层中LoginAction.java的描述如下所示：

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

/\*\*

\* *@description* 实现登录页面Action

\*/

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

{

**private** User user;

*@Resource*

**private** IUserService userService;

/\*实现各种属性的setter和getter方法\*/

/\*实现登入模块的控制层方法\*/

**public** String login(){}

}

### 4.2.3 业务逻辑层

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

表 4-5登入模块业务逻辑层列表

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

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

IUserService.java接口主要方法：

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

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

/\*登入检测\*/

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

/\*查找用户\*/

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

/\*删除用户\*/

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

/\*增加用户\*/

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

}

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

UserServiceImpl.java主要实现属性与方法:

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

*@Transactional*

*@Service*

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

*@Resource*

**private** IUserDao iUserDao;

*@Override*

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

*@Override*

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

*@Override*

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

*@Override*

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

}

### 4.2.4 数据持久层

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

表 4-6 登入模块数据持久层列表

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

IUserDao.java定义了用户信息进行增、删、改、查的接口：

package com.buptsse.spm.dao;

/\*\*

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

\*/

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

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

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

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

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

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

}

UserDaoImpl.java是IUserDao.java接口的实现：

package com.buptsse.spm.dao;

/\*\*

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

\*/

@Repository

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

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

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

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

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

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

}

### 4.2.5 域模型层

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

表 4-7登入模块域模型层列表

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

User.java主要属性与方法：

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

/\*\*

\* *@description* User表的信息记录

\*/

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

**private** **String** id; /\*id识别\*/

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

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

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

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

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

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

/\*编写对上述各个属性的*getter*与*setter*方法\*/

}

## 4.n、XXXX模块

XXX模块的。。。。如图4-？所示。

图 4-？ XXXXXX图

### 4.n.1 表现层

。。。。。。

### 4.n.2 控制层

。。。。。。。

### 4.n.3 业务逻辑层

。。。。。。

### 4.n.4 数据持久层

。。。。。。

### 4.n.5 域模型层

。。。。。。

# 5、管理端模块详细设计

## 5.1、XXXXX模块详细设计

### 5.1.1 表现层

。。。。。。。。。。

### 5.1.2 控制层

。。。。。。。。。。

### 5.1.3 业务逻辑层

。。。。。。。。。。

### 5.1.4 数据持久层

。。。。。。。。。。

### 5.1.5 域模型层

。。。。。。。。。。

。。。。。。。。。

## 5.n、XXXXX模块详细设计

。。。。。。。。。。

### 5.n.1 表现层

。。。。。。。。。。

### 5.n.2 控制层

。。。。。。。。。。

### 5.n.3 业务逻辑层

。。。。。。。。。。

### 5.n.4 数据持久层

。。。。。。。。。。

### 5.n.5 域模型层

。。。。。。。。。。

# 6、公共部分模块详细设计

## 6.1 公共页面

。。。。。。。。。。

## 6.2 安全模块的详细设计

。。。。。。。。。。

## 6.n XXXXX

。。。。。。。。。。