|  |  |
| --- | --- |
| 项目编号 | 20191101 |
| 文档编号 | 03 |
| 密 级 | 内部 |

学生管理系统详细设计

**V1.0**

学号：18061615

姓名：李国宇

学院：卓越学院

学校：杭州电子科技大学

目录

[**1 导言** 5](#_Toc23962197)

[**1.1 目的** 5](#_Toc23962198)

[**1.2 范围** 5](#_Toc23962199)

[**1.3 缩写说明** 5](#_Toc23962200)

[**1.4 术语定义** 6](#_Toc23962201)

[**1.5 引用标准** 6](#_Toc23962202)

[**1.6 参考资料** 6](#_Toc23962203)

[**1.7 版本更新信息** 6](#_Toc23962204)

[**2 系统设计概述** 6](#_Toc23962205)

[**3 详细设计概述** 7](#_Toc23962206)

[**4 用户登录模块的详细设计** 7](#_Toc23962207)

[**4.1 前端架构** 8](#_Toc23962208)

[**4.1.1前台展示伪代码** 8](#_Toc23962209)

[**4.1.2前台处理伪代码** 9](#_Toc23962210)

[**4.2 后台接口设计** 9](#_Toc23962211)

[**5 系统管理主框架设计** 10](#_Toc23962212)

[**5.1 主框架前台设计** 10](#_Toc23962213)

[**5.1.1 主框架前台代码** 11](#_Toc23962214)

[**5.1.2 主框架js代码** 11](#_Toc23962215)

[**5.2 主框架后台设计** 11](#_Toc23962216)

[**6 学生信息管理系统详细设计** 12](#_Toc23962217)

[**6.1 主界面表格设计** 13](#_Toc23962218)

[**6.1.1 主界面表格前端展示设计** 13](#_Toc23962219)

[**6.1.2 主界面表格前端js设计** 14](#_Toc23962220)

[**6.1.3 主界面表格后端接口设计** 14](#_Toc23962221)

[**6.2 添加学生界面设计** 14](#_Toc23962222)

[**6.2.1 添加学生前端dialog设计** 14](#_Toc23962223)

[**6.2.2 添加学生前端js设计** 14](#_Toc23962224)

[**6.2.3 添加学生后端接口设计** 15](#_Toc23962225)

[**6.3 删除学生界面设计** 16](#_Toc23962226)

[**6.3.1 删除学生前端dialog设计** 16](#_Toc23962227)

[**6.3.2 删除学生前端js设计** 16](#_Toc23962228)

[**6.3.3 删除学生后端接口设计** 17](#_Toc23962229)

[**6.4 统计学生界面设计** 17](#_Toc23962230)

[**6.4.1 统计学生前端dialog设计** 17](#_Toc23962231)

[**6.4.2 统计学生前端js设计** 18](#_Toc23962232)

[**6.4.3 统计学生后端接口设计** 18](#_Toc23962233)

[**6.5 编辑学生界面设计** 18](#_Toc23962234)

[**6.5.1 编辑学生前端dialog设计** 18](#_Toc23962235)

[**6.5.2 编辑学生前端js设计** 19](#_Toc23962236)

[**6.5.3 编辑学生后端接口设计** 19](#_Toc23962237)

[**6.6 搜索学生功能设计** 19](#_Toc23962238)

[**6.6.1 搜索学生前端输入框设计** 19](#_Toc23962239)

[**6.6.2 搜索学生后端js设计** 19](#_Toc23962240)

[**6.6.3 搜索学生后端接口设计** 20](#_Toc23962241)

[**7 课程管理功能设计** 20](#_Toc23962242)

**1 导言**

**1.1 目的**

该文档的目的是描述《学生信息管理系统》项目的详细设计，其主要内容包括：

* 系统功能简介
* 系统详细设计简述
* 各个模块的三层划分
* 最小模块组件的伪代码

本文档的预期读者是：

* 开发人员
* 项目管理人员
* 测试人员

**1.2 范围**

该文档定义了系统的各个模块和模块接口，但未确定单元的具体实现，这部分内容将在实现中确定。

**1.3 缩写说明**

* **ADMIN** *Administrator(管理员)*
* **USER** *用户*
* **UML** *Unified Modeling Language（统一建模语言）*
* **SPA** *Singo Page Applications（单页面应用）*
* **MVVM** *Model-View-ViewModel*

**1.4 术语定义**

* **HTML** *HyperText Markup Language(超文本标记语言)*
* **PHP** *PHP: Hypertext Preprocessor(超文本预处理器)*
* **AJAX** *Asynchronous Javascript And XML(异步 JavaScript 和 XML)*
* **JS** *JavaScript(直译式脚本语言)*
* **SQL** *Structured Query Language(结构化查询语言)*

**1.5 引用标准**

* [1]缺省

**1.6 参考资料**

* [1]缺省

**1.7 版本更新信息**

本文档的更新记录如表C-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** |
| --- | --- | --- | --- | --- |
| 000 | 2019.11.6 | 0.1 | 全部 | 初始发布版本 |

**2 系统设计概述**

根据《学生信息管理系统》的概要设计，本系统按照功能角度分解，可以分为客户端子系统和管理端子系统。根据页面流的设计，二者系统分为用户登录、学生管理、课程管理、学生信息查询、学生信息统计等模块，他们的关系如图C-1(B-12)，以下将分小节对各个部分分别进行详细设计。  
图C-1：模块设计

**3 详细设计概述**

由于本系统采用了前后端分离的结构设计，基本采用一个功能对应一个后台接口的方法，后台采用当下流行的PHP语言，在整个开发过程中，尽可能采用复用的原则，例如采用统一的数据库连接类，请求返回类，统一的结果显示等。  
本文档的详细设计主要是按照前后端两个模块的伪代码，为下一步的编码提供基础。伪代码(Pseudocode)是一种算法描述语言。使用伪代码的目的是为了使被描述的算法可以容易地以任何一种编程语言实现。因此， 伪代码必须结构清晰，代码简单，可读性好，并且类似自然语言。

**4 用户登录模块的详细设计**

用户登录模块主要实现用户权限验证和session创建，它的前后端如表C-2所示  
表C-2：用户登录模块的前后端设计

| **前端页面** | **前端div** | **后端接口** |
| --- | --- | --- |
| login.php | login-form | modules/loginAuth.php |
| adminFrame.php | logout | modules/logout.php |

**4.1 前端架构**

前端主要实现界面的展示和数据处理，以及与后台接口做对接。

**4.1.1前台展示伪代码**

<?php

session\_start();

if (isset($\_SESSION['isLogin']) && $\_SESSION['isLogin'] === 1) {

}else{

header('Location: ./adminFrame.php');

exit;

}

?>

<!DOCTYPE html>

<html>

<head>

...

</head>

<body class="mdui-theme-primary-indigo mdui-theme-accent-pink mdui-appbar-with-toolbar"

style="background-color: whitesmoke;">

//外层框架 任务栏等

<div class="mdui-container mdui-valign mdui-m-t-5" style="height: 100%">

<div class="logincard mdui-shadow-8 mdui-center" style="max-width: 480px;" id="login-form">

<h1 class="mdui-typo-display-1-opacity">用户登录</h1>

<div class="mdui-textfield mdui-textfield-floating-label">

<label class="mdui-textfield-label">用户名</label>

<input class="mdui-textfield-input" type="text" id="username" required />

</div>

<div class="mdui-textfield mdui-textfield-floating-label">

<label class="mdui-textfield-label">密码</label>

<input class="mdui-textfield-input" type="password" id="password" required />

</div>

<button class="mdui-btn mdui-color-theme-accent mdui-ripple mdui-btn-block login" id="loginbtn">登录</button>

</div>

</div>

<div class="mdui-dialog" id="dialog">

<div class="mdui-dialog-title">登录</div>

<div class="mdui-dialog-content" id="loginmsg"></div>

<div class="mdui-dialog-actions">

<button class="mdui-btn mdui-ripple" id="closedia">确认</button>

</div>

</div>

</body>

...一些js文件...

</html>

**4.1.2前台处理伪代码**

进行处理的js文件运用了JQuery技术和后端进行对接，以下是jquery的伪代码部分

$(document).ready(function () {

var dialog = new mdui.Dialog('#dialog');

$("#loginbtn").click(function () {

....点击事件响应代码....

});

$("#closedia").click(()=>dialog.close());

});

**4.2 后台接口设计**

<?php

$opt=array();

session\_start();

//验证用户是否已经登陆？已经就不用请求了

}

require "db.php";

$db=new Db;

if(isset($\_POST['UserName'])&&isset($\_POST['Password'])){

//查询用户是否存在？

if($data){

if($data[0][4]!=1){

//用户被禁止使用

}else{

//正常登录

$\_SESSION=。。。

//设定用户session

}

}else{

//用户名或密码错误

}

}else{

//数据不完整

}

echoend:

//输出反馈json

header('Content-type: application/json');

echo json\_encode($opt,JSON\_UNESCAPED\_UNICODE|JSON\_UNESCAPED\_SLASHES);

?>

**5 系统管理主框架设计**

表C-3：系统管理主框架的前后端设计

| **前端页面** | **前端div** | **后端接口** |
| --- | --- | --- |
| adminFrame.php | body appbar drawer | modules/logout.php |

**5.1 主框架前台设计**

主框架用来展示系统所有功能和用户当前登录状态，管理界面的所有功能都会在主框架内通过iframe显示。

**5.1.1 主框架前台代码**

<html>

<head>

//http头文件 标题

</head>

<body>

//菜单和应用栏

//根据登陆状态展示功能

//用户主操作界面

<iframe id="frame" src="adminStudent.php" frameborder="0" style="height:100%;width:100%"></iframe>

//js

</body>

</html>

**5.1.2 主框架js代码**

//定义菜单

function changeFrame(target){

//改变iframe内容

}

$("#logout").click(function () {

//监听logout按钮点击并登出

});

**5.2 主框架后台设计**

主要用于未登录直接跳转登陆界面

<?php

session\_start();

if (isset($\_SESSION['isLogin']) && $\_SESSION['isLogin'] === 1) {

}else{

header('Location: ./login.php');

exit;

}

?>

**6 学生信息管理系统详细设计**

学生管理系统是本系统的核心，作为单独的一个功能，其内部有很多完成功能的对话框，各个部分的概要设计如下：  
表C-4：学生信息管理系统的前后端设计

| **前端页面** | **前端div** | **后端接口** |
| --- | --- | --- |
| adminFrame.php | body appbar drawer | ./modules/logout.php |
| adminStudent.php | add-dialog | ./modules/addStudent.php |
| adminStudent.php | del-dialog | ./modules/deleteStudent.php |
| adminStudent.php | score-dialog | ./modules/getScore.php |
| adminStudent.php | mod-dialog | ./modules/updateStudent.php |
| adminStudent.php | search | ./modules/searchStudent.php |
| adminStudent.php | getStudent | ./modules/getStudentByNo.php |

**6.1 主界面表格设计**

主界面表格负责展示搜索到的学生信息或全部信息，每个表项内含有按钮，可以对该学生进行相应的操作。

**6.1.1 主界面表格前端展示设计**

<head>

//html头

</head>

<body>

<div>

<div class="clearfix">

<div>

//判断有无权限增加或删除学生，展示按钮。

<?php

if($\_SESSION['Auth']==0){//不是管理员

echo <<<btngroup

<div class="btngroup">

<button id="add" class="mdui-btn mdui-color-green mdui-ripple"><span

style="color:white">添加学生</span></button>

<button id="delete" class="mdui-btn mdui-color-red mdui-ripple"><span

style="color:white">删除学生</span></button>

</div>

btngroup;

}

?>

</div>

<div>

<div>

//搜索框

<input id="search" class="mdui-textfield-input" type="text" placeholder="查找学生姓名" />

</div>

</div>

</div>

<div class="mdui-row">

<div class="mdui-table-fluid" style="margin-top:20px;">

<table class="mdui-table">

//学生信息表

</table>

</div>

</div>

</div>

**6.1.2 主界面表格前端js设计**

主表格仅作展示，其内容由后面的“搜索”功能的表格更新函数更新数据。

**6.1.3 主界面表格后端接口设计**

主界面表格后端不提供接口，仅作数据展示，内容操作由前端js实现

**6.2 添加学生界面设计**

添加学生按钮经权限判断后才会显示，其表现形式为弹出添加学生对话框。

**6.2.1 添加学生前端dialog设计**

//添加学生的dialog

<div class="mdui-dialog" id="add-dialog">

<div class="mdui-dialog-title">添加学生</div>

<form class='mdui-m-a-5'>

//添加学生的表格

</form>

</div>

**6.2.2 添加学生前端js设计**

js用于点击后发送学生数据到后台api，伪代码如下：

$("#addbtn").click(function () {

$.post("./modules/addStudent.php",

{

//学生信息post

},

function (data, status) {

//操作状态提示

}

)

});

**6.2.3 添加学生后端接口设计**

具体调用方法详见概要设计

<?php

session\_start();

$code=400;$msg="";$dout=array();

$login=false;

$admin=false;

if(isset($\_SESSION['isLogin'])){

if($\_SESSION['isLogin']!=1){

//判断权限

}else{

//权限不足退出

//exit

}

}

if($login&&$admin){

if(isset($\_POST['Sno'])&&isset($\_POST['Ssex'])&&isset($\_POST['Clno'])&&isset($\_POST['Sname'])){

//数据库操作

require "db.php";

$db=new Db;

$res=$db->myinsert("

INSERT INTO Student (Sno,Ssex,Clno,Sname,CreateBy)

VALUES ('".$\_POST['Sno']."',

'".$\_POST['Ssex']."',

'".$\_POST['Clno']."',

'".$\_POST['Sname']."',

'".$\_SESSION["Uid"]."')");

if($res){

//成功返回

}else{

//执行错误处理

}

} else{

//数据错误处理

}

}

//结果json返回

echoend:

$opt=array(

"code"=>$code,

"msg"=>$msg,

"data"=>$dout

);

echo json\_encode($opt,JSON\_UNESCAPED\_UNICODE|JSON\_UNESCAPED\_SLASHES);

?>

**6.3 删除学生界面设计**

删除学生按钮在判断权限符合后出现，在权限满足后弹出对话框进行删除操作。

**6.3.1 删除学生前端dialog设计**

<div class="mdui-dialog" id="delete-dialog">

<div class="mdui-dialog-title">请输入要删除的学生学号...</div>

//查询学号并显示在下面的显示区

<div id="del-inf">

//学生信息显示区

</div>

<button id="del-query-conf" style="display: inline-block;"

class="mdui-m-t-3 mdui-btn mdui-color-red mdui-ripple"><span style="color:white">删除</span></button>

</div>

</div>

</div>

**6.3.2 删除学生前端js设计**

负责post给后端删除的学号，后端验证完成后返回操作结果。

$("#list-delete-conf").click(function () {

$.post("./modules/deleteStudent.php",

{

//删除学生的学号

},

function (data, status) {

//删除状态提示

}

)

});

**6.3.3 删除学生后端接口设计**

大体与6.2.3相同，仅数据库处理有所差别，关键代码如下:

$res=$db->delete("DELETE FROM score WHERE Sno='".$\_POST['Sno']."'");

$res=$db->delete("DELETE FROM student WHERE Sno='".$\_POST['Sno']."'");

**6.4 统计学生界面设计**

统计学生功能前后端都可以使用，点击表格内部的眼睛按钮，会弹出学生的统计信息对话框。显示学生的平均分，最高分，最低分，详细信息等。

**6.4.1 统计学生前端dialog设计**

<div class="mdui-dialog" id="score-dialog">

<div class="mdui-dialog-title">成绩统计信息</div>

<div class="mdui-m-x-5 mdui-m-b-5">

//学号姓名显示

//平均分、最高最低分显示

<div class="mdui-typo-subheading mdui-m-y-3">详细成绩信息</div>

//详细信息显示

<div class="score-list mdui-m-l-2" id="score\_context">

</div>

</div>

</div>

**6.4.2 统计学生前端js设计**

后端js再点击按钮后得到学生学号，再将其发送给后端查询接口，接收返回值并更新显示。

function score(id) {

$.post("./modules/getScore.php",

{

//传入学号信息

},

function (data, status) {

//获得返回信息，解析并更新到页面上

//打开dialog

}

}

)

}

**6.4.3 统计学生后端接口设计**

大体与6.2.3相同，仅数据库处理有所差别，关键代码如下:

$res=$db->myquery("SELECT s.Sname,c.Cname,sc.Score FROM student s,course c,score sc WHERE s.Sno='".$\_POST['Sno']."' AND s.Sno=sc.Sno AND c.Cno=sc.Cno ORDER BY sc.Score DESC");

**6.5 编辑学生界面设计**

编辑学生界面入口由表格后面笔状按钮触发，只有管理员有此按钮，判断权限后打开编辑学生对话框。

**6.5.1 编辑学生前端dialog设计**

<div class="mdui-dialog" id="modify-dialog">

<div class="mdui-dialog-title">编辑</div>

<form action="" method="get" class='mdui-m-a-5'>

//编辑学生对话框

</form>

</div>

**6.5.2 编辑学生前端js设计**

$("#modify-conf").click(function () {

$.post("./modules/updateStudent.php",

{

//发送被编辑的学生信息

},

function (data, status) {

//接受返回数据，刷新主表格

}

)

});

**6.5.3 编辑学生后端接口设计**

大体与6.2.3相同，仅数据库处理有所差别，关键代码如下:

$res=$db->myupdate("

UPDATE Student SET Ssex='".$\_POST['Ssex']."',Clno='".$\_POST['Clno']."',Sname='".$\_POST['Sname']."' WHERE Sno='".$\_POST['Sno']."'");

**6.6 搜索学生功能设计**

搜索学生管理员和用户都可以使用，在表格右上方输入框输入关键字，自动实时刷新主表格反馈结果。

**6.6.1 搜索学生前端输入框设计**

<input id="search" class="mdui-textfield-input" type="text" placeholder="查找学生姓名" />

**6.6.2 搜索学生后端js设计**

后端js实时监测输入框变化，实时刷新主表格内容

//实时绑定变化

$("#search").bind("input propertychange", function (event) {

$.post("./modules/searchStudent.php",

{

//关键字查询

},

function (data, status) {

//解析返回值，刷新表格

}

}

)

});

**6.6.3 搜索学生后端接口设计**

大体与6.2.3相同，仅数据库处理有所差别，关键代码如下:

if($\_POST['keywords']!=""){

$res=$db->myquery("SELECT \* FROM student WHERE Sno like '%".$\_POST['keywords']."%' OR Sname like '%".$\_POST['keywords']."%' limit 50");

}else{

$res=$db->myquery("SELECT \* FROM student limit 50");

}

**7 课程管理功能设计**

功能与6 学生管理系统功能设计大部分相同，仅数据库有所区别，详见代码文件。