

**信息系统设计实训报告**

**第五组——学分制学费管理系统**

**2018-2019-3学期**

**2019.7**

小组成员组成及成绩评定

|  |  |  |  |
| --- | --- | --- | --- |
| **小组总评成绩** | |  | |
| 姓名 | 学号 | 负责内容概况 | 成绩 |
| 陶新月 | 1608020131 | 1. 负责实训报告的后半部分内容撰写 2. 共同讨论系统架构，编写后台代码 3. 设计与开发登录界面和左侧导航栏；公告信息添加和查看界面；缴费信息管理栏，包括管理员对缴费信息的增删改；学生身份进入时的缴费信息查询及支付界面。 |  |
| 程婷 | 1608020225 | 1. 负责实训报告前半部分内容撰写 2. 共同讨论系统架构，编写后台代码 3. 设计与开发管理员身份进入时的用户信息管理栏内容，包括管理员账号和学生账号的增删改功能；学生身份进入时的选课信息栏，可查询已选课程及学分信息。 |  |
| 刘烨 | 1608020114 | 1. 完善实训报告代码部分 2. 共同讨论系统架构，编写后台代码 3. 开发与设计管理员身份进入时的收费信息管理栏，包括收费信息的增删改功能，导出收费Excel表格；学生身份进入时的个人信息管理栏，包括密码修改等。 |  |

撰写报告说明：

1. 信息系统设计实训是以项目为基础，以小组合作完成，因此，报告为小组报告，每小组只需要上交一份，但在每小组各个成员需全程参与，分工完成，报告中需要明确每位小组成员负责内容，作为成绩评定的依据。
2. 报告需按照指定格式完成，不得随意删减内容。报告正文使用宋体小四号字，1.5倍行距，一律用A4纸单面打印，页边距上、下、左、右均为2.5厘米。
3. 正文分章节撰写，第一级标题用“第1章”、“第2章”、“第3章”等连续编号，每章应另起一页，标题末尾不加标点(问号、叹号、省略号除外)，标题居中排列，下空一行接写第二级标题。从第二级标题开始，用阿拉伯数字连续编号，在不同层次的数字之间加一个下圆点相隔，最末数字后不加标点。如第二级标题为“1.1”、“2.1”、“3.1”等，第三级标题为“1.1.1”、“2.1.1”、“3.1.1”等，第四级标题为“1.1.1.1”、“2.1.1.1”、“3.1.1.1”等。正文中的标题一般不超过四级，标题层次要清晰，第二至第四级标题均单独占一行，且靠左端书写，第二级标题序数前不留空格，第三、四级标题序数前要空两个汉字位置。各级标题序数后均空一格接写标题。
4. 每幅图都应有图题，图题由图号和图名组成。图号按章编排，如“图2-4”表示第二章第4张插图，图号与图名之间空一格排写，图题居中置于图下，图中若有分图时，分图号用(a)、(b)等置于分图之下。每个表格应有自已的表题和表序，表题应写在表格上方正中，表序写在表题左方不加标点，空一格接写表题，表题末尾不加标点。表格应逐章编序，如“表2-2”表示第二章的第2张表。表序必须连续。表格允许下页接写，接写时表题省略，表头应重复书写，并在右上方写“续表××”。数字空缺的格内加“－－”字线（占2个数字），不允许为空；表中有附注时，写在表的下方，句末加标点。
5. 首页、成员组成页、撰写说明页均为一页，其余内容根据实际情况确定页数。

目录

[第一章 项目概况及要求 5](#_Toc12865311)

[1.1 项目背景 5](#_Toc12865312)

[第二章 系统需求分析 6](#_Toc12865313)

[2.1 用户功能要求 6](#_Toc12865314)

[2.2 系统性能要求 6](#_Toc12865315)

[2.3输入输出要求 6](#_Toc12865316)

[2.4 用户其他要求 7](#_Toc12865317)

[2.5 可行性分析 7](#_Toc12865318)

[第三章 系统分析 9](#_Toc12865319)

[3.1 组织结构及业务流程分析 9](#_Toc12865320)

[3.1.1组织结构分析 9](#_Toc12865321)

[3.1.2业务流程分析 9](#_Toc12865322)

[3.2用例图分析 10](#_Toc12865323)

[3.3实体类图分析 10](#_Toc12865324)

[第四章 系统设计 11](#_Toc12865325)

[4.1 总体设计( 功能树设计） 11](#_Toc12865326)

[4.2 顺序图设计 11](#_Toc12865327)

[4.2.1用户登录 11](#_Toc12865328)

[4.2.2学生信息管理 12](#_Toc12865329)

[4.3 类图设计 12](#_Toc12865330)

[4.3.1 类图 12](#_Toc12865331)

[4.3.2 类关系图 13](#_Toc12865332)

[4.4 编码设计（编码规则） 13](#_Toc12865333)

[4.5输入输出设计 14](#_Toc12865334)

[4.5.1输出设计 14](#_Toc12865335)

[4.5.2输入设计 14](#_Toc12865336)

[4.6 模块算法设计 15](#_Toc12865337)

[第五章 系统测试与部署 16](#_Toc12865338)

[5.1 系统架构选择（或应用程序结构设计） 16](#_Toc12865339)

[5.2系统部分代码示例 16](#_Toc12865340)

[5.3系统界面实现（贴界面实际图） 17](#_Toc12865341)

[5.4系统测试 17](#_Toc12865342)

[5.5 系统实施应用 17](#_Toc12865343)

[第六章 系统测试与部署 18](#_Toc12865344)

[第七章 其他说明 19](#_Toc12865345)

[第八章 反思日志 20](#_Toc12865346)

第一章 项目概况及要求

1.1 项目背景

1.1.1 系统名称

学分制学费管理系统

[1.1.2 系统来源](#_Toc292631819)

学分制学费管理系统是个典型的管理信息系统（MIS）。其设计开发主要包括了后台数据库设计建立和维护以及前台应用程序的设计开发两方面，详细分析了学校缴费流程的各个环节，针对出现的一些问题，设计出来了这样的一个方便、简洁、有效的学生缴费管理系统。本系统开发语言选用java，数据库选用MySQL，使用JDBC数据库连接技术，使用Eclipse作为系统应用程序的开发工具，Web服务器选用Tomcat9.0版本。

[1.1.3 系统背景](#_Toc292631820)

21世纪，网络技术已经越来越广泛的应用于科学管理领域，越来越多的领域已经实现了管理信息化，管理者也更较重视服务绩效。而国内的管理系统，由于计算机网络技术发展起步晚，速度相对缓慢，加之传统的人事管理观念，数据管理的质量与服务绩效等问题尚未引起人们的重，导致国内相当多的单位并未采用科学的管理方法，出现管理不规范、管理混乱、效率低下等问题。

学生收费管理系统的出现能将高校缴费的难度大大地减轻，很符合现在电子信息化的时代理念。学校方面可以将本系统安装到PC机上面，再发布到学校的专门的网页，学生在缴费的同时接入互联网，用学号登入到系统，根据自己的缴费情况进行操作，系统将把学生的信息进行收集分类，然后修改数据库的相关内容。学校只需以特定的用户帐号进行登入，然后进行学生缴费信息的统计处理。操作很方便，安全性能也很有保证。

第二章 系统需求分析

2.1 用户功能要求

在学分制下，学生的学费与所修课程学分紧密相连，通过学分制学费管理系统可以达到以下目标：

☑ 学生可通过前台登陆进行相关操作。

☑ 学生可以查询所选课程及对应课程的学费情况。

☑ 学生可以实现线上自主缴费，并生成订单。

☑ 管理员可通过后台登陆进行相关操作，包括收费信息的增删改等。

☑ 管理员可以导出统计数据。

[2.2 系统性能要求](http://learn.upc.edu.cn/meol/common/ueditor/content.html?name=520#_Toc292631829)

2.2.1 安全性

完善的权限管理，通过系统对学生选课的数据进行全面的统计和管理，计算学分学费，避免人为处理数据产生的各种问题，实现收费的安全性。

2.2.2 高效性

网络化管理，使系统更加规范，实现通过网络对信息进行发布、传递、接收等一系列功能，大大提高了学校相关部门的办公效率。

2.2.3 易安装性、易维护性、易操作性

2.3输入输出要求

学生输入学号为十位数字，账号密码默认为身份证后六位（包括最后一位为X），若已修改，则与数字石大密码相同。进入登录页面密码出错次数超过三次无法进入服务界面，且系统会给出提醒：密码错误！请一个小时后再次进入，并一小时内禁止该用户登录；

输出页面显示本学年已修学分及应缴学费数额（包括学分学费、专业注册学费、住宿费，单位：元）。对于输入输出的数据进行的规定一定要严格按照用户的实际情况而进行，以便能更好的进行数据管理。

2.4 用户其他要求

1）用户输入错误登录信息时，界面友好提醒错误；

2）系统严格按照设定的安全权限机制运行，防止非法用户进入；

3）保证界面操作简便，信息显示清晰；

4）对于运行阶段出现的程序问题，按照需求规定记录各种日志，并及时做出纠正，确保系统稳定可靠；

5）采取数据备份和数据还原的功能模块，在系统意外崩溃的时候能及时还原原来的数据资料，以减少损失

2.5 可行性分析

本系统设计基于B/S 模式下，运用Java、JSP技术，采用的是MySQL数据库和Eclipse实现，总体的可行性共分为以下三个方面­：

2.5.1 技术可行性

采用的是当下流行的JAVA语言，JSP技术，它具有如下优点——强大的可伸缩性。从只有一个小的Jar文件就可以运行Servlet/JSP，到由多台服务器进行集群和负载均衡，到多台Application进行事务处理，消息处理，一台服务器到无数台服务器，Java显示了一个巨大的生命力。JSP可以使用成熟的JAVA BEANS 组件来实现复杂功能。

2.5.2 经济可行性

学分制学费管理系统采用的插件都是开源免费的，开发工具Eclipse，以及数据库是MySQL，服务器采用Tomcat，也当下流行的开发所用的服务器，所以在开发前期，开发时用于项目的经费将会大大降低，不会让开发该软件在项目启动期受到经费的影响，所以经济上还是可行的。

2.5.3 操作可行性

本系统实现功能的操作很简单，普通电脑的常见配置就可以运行本软件，并且只要粗通电脑使用的基本常识就可以流畅的使用本软件。电脑具备连接互联网的能力，就可以正常访问系统，并不需要操作者有什么高超的能力，只需了解业务流程进行正确操作即可，所以学生收费管理系统具备操作可行性。

第三章 系统分析

3.1 组织结构及业务流程分析

3.1.1组织结构分析

组织结构是一个组织的组成以及这些组成部分之间的关系，通常可用组织结构图来表示。我们按照管理的特点将整个学校的财务室划分为以下几个部分。组织机构图如下所示：



图3-1 组织结构图

3.1.2业务流程分析

业务流程图是一种物理模型，它描述系统内部各单位与人员之间的业务关系和管理信息的流向：



图3-2 业务流程图

3.2用例图分析

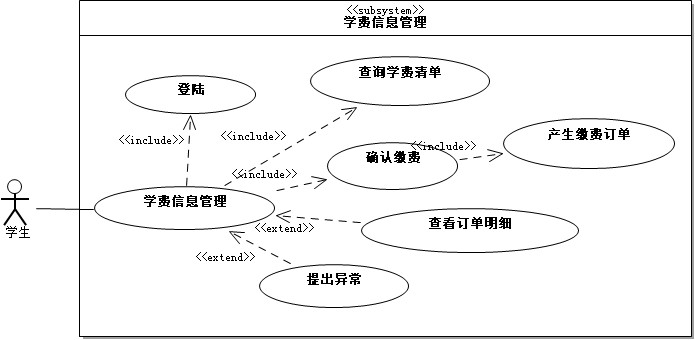


图3-3 学费信息管理

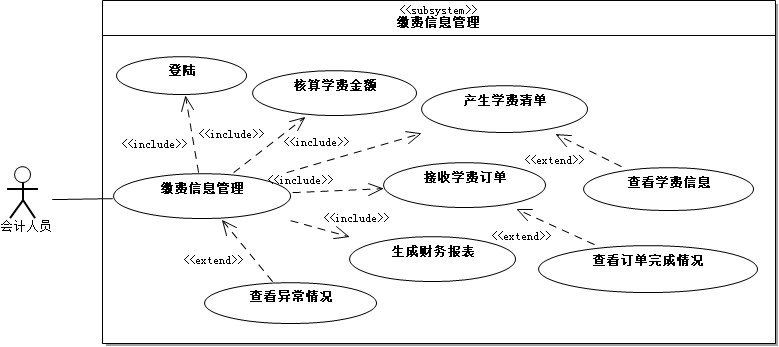


图3-4 缴费信息管理

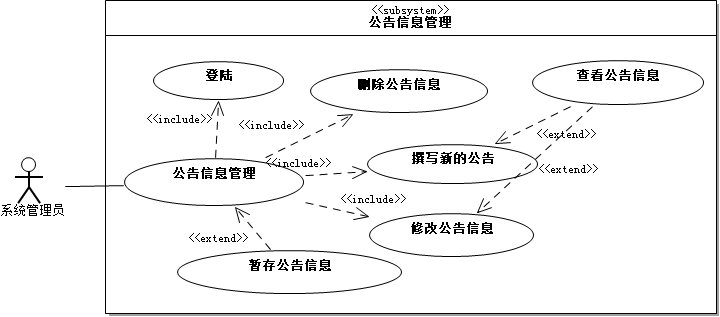


图3-5 公告信息管理

3.3实体类图分析

类图显示了模型的静态结构，特别是模型中存在的类、类的内部结构以及它们与其他类的关系等。本系统的实体类包括学生、管理员、登录信息、选课信息、学费信息及统计信息。

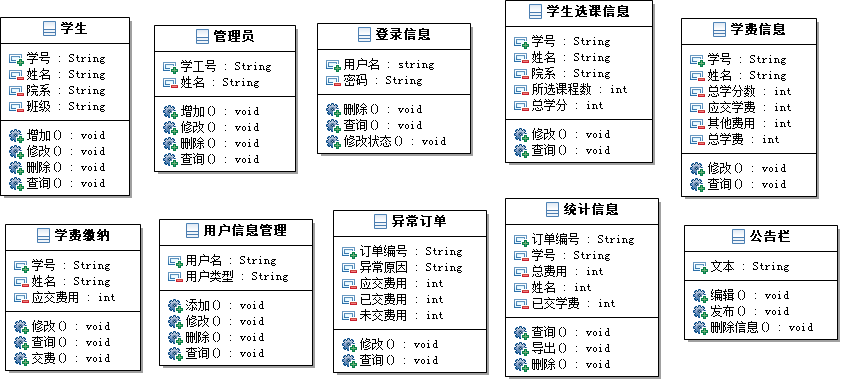


图3-6 实体类图

## 第四章 系统设计

### 4.1 总体设计( 功能树设计）



图4-1 功能树

### 4.2 顺序图设计

### 4.2.1用户登录

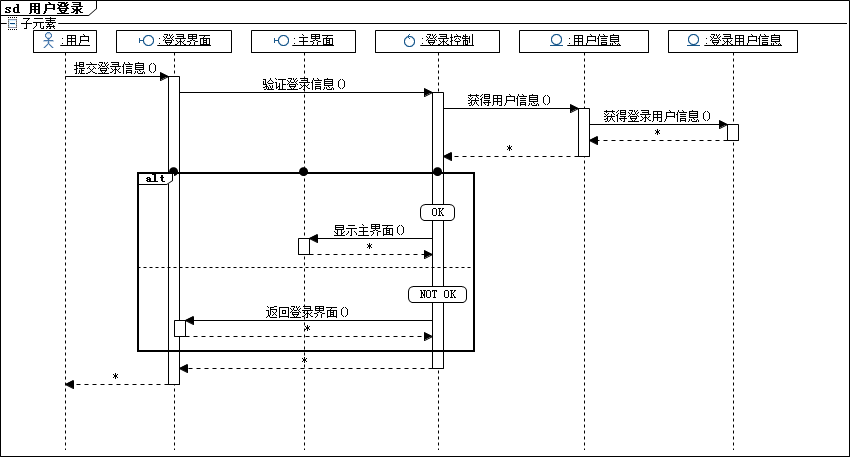


图4-2 用户登录用例图

### 4.2.2学生信息管理

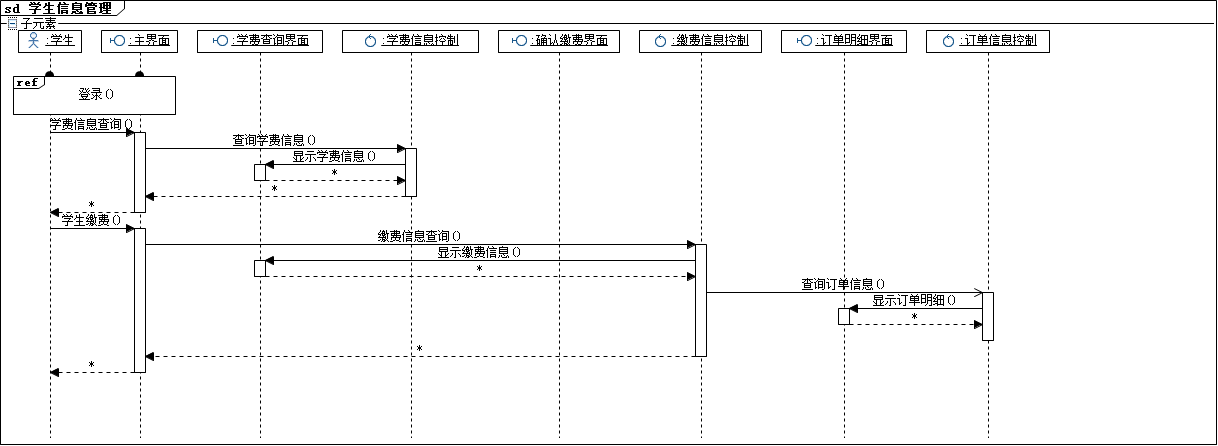


图4-3 学生信息管理用例图

### 4.2.3公告信息管理

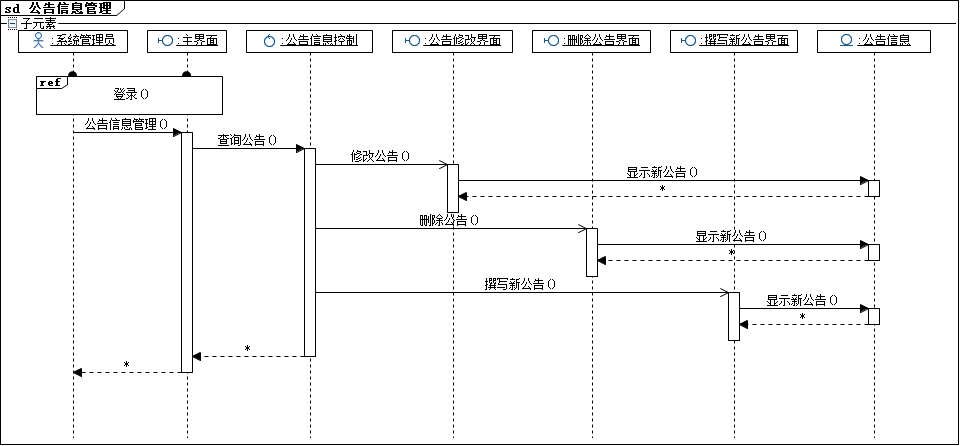
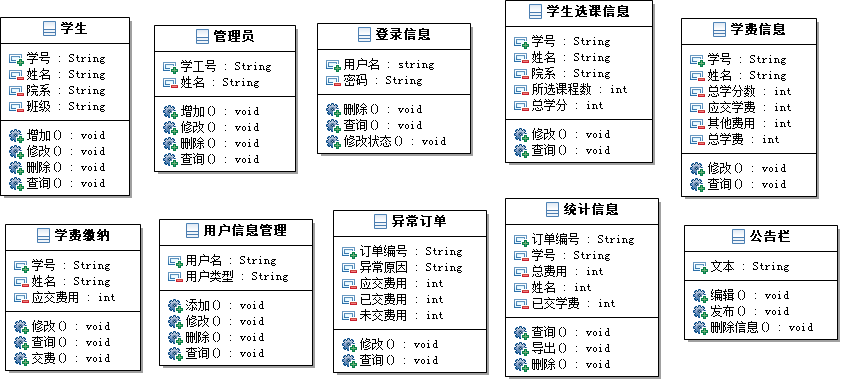


图4-4 公告信息管理用例图

### 4.3 类图设计

### 4.3.1 类图

图4-5 类图建模

### 4.3.2 类关系图

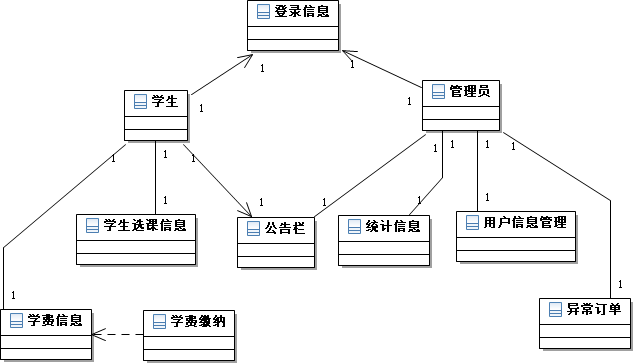


图4-6 类关系图建模

### 4.4 编码设计（编码规则）

代码的作用有标识作用：可用来标识和确定某个具体的对象，以便于计算机的识别；统计和检索作用：当按对象的属性或类别进行编码时，易于优化对象的统计和检索；对象状态的描述作用：可以用来标明事物所处的状态，便于对象的动态管理。

本学生缴费管理系统中主要使用顺序码。顺序码指将顺序的自然数赋予编码对象。具体为用户登录时的学号：顺序码，共12位，其中前四位表示年级，五六位表示院系，七八位表示专业，九十位表示班级，最后两位表示是学生在班级的编号。

### 输入输出设计

### 4.5.1输出设计

输出设计是非常重要的。管理信息系统只有通过输出才能为用户服务。从系统开发的角度看，输出决定输入，即输入信息只有根据输出要求才能确定。

用户是输出信息的主要使用者。在设计输出内容时首先确定用户在使用信息方面的要求。本系统采用信息形式为表格、图形、文字，输出项目为学生缴费信息。

本系统采用缴费凭证作为输出形式。凭证由三部分组成：表头、表体和表尾。对于单个用户一次性使用的表格，因为没有保留价值，设置在显示终端中输出。对于多个用户需要多次使用的表格，则打印输出。

### 4.5.2输入设计

输入设计的目标是保证向系统输入正确的数据。在此前提下，应尽量做到输入方法简单、迅速、经济、方便。

输入设计的内容：

本系统输入设计的内容设计如下：输入项的名称为学生缴费信息，数据的内容为学生的选课信息情况和缴费信息情况。

由于数据的发生地点在学校的财务室，故采用联机终端输入。

输入数据的记录格式分为两部分：预先印刷部分和插入数据的空格。需要用户填写的表格，不仅注明了文字，而且在表格下方都有代码说明。

本系统输入数据的校验方法比较多，具体有：数据类型校验，即可以通过检验数据是数字型还是字符型来做合理性校验。格式校验,即用来检验各数据项位数和位置是否合乎事先的定义。界限校验，即学校不同缴费项目的缴费金额是固定的。不会超过某个数字，也不会低于某个数字，设置范围，若检查某项数据时，在此范围之外，则属错误。

本系统采用常规的输入设备，具体如下：键盘、鼠标等。

### 模块算法设计

### 4.6.1登录模块

### 4.6.2学生缴费模块

### 4.6.3公告信息模块

### 4.6.4用户管理模块

## 第五章 系统测试与部署

### 5.1 系统架构选择（或应用程序结构设计）

系统整体采用分层架构，分层就是要把界面部分、业务逻辑部分、数据库访问部分的代码放在各自独立的方法或类中编写，减少代码之间的联系，分层设计思想已经成为开发软件的主流思想。

本系统架构清晰合理，程序运行稳定可靠，遵循了MVC架构模式，采用分层设计，把系统分为表示层、业务层和持久层。表示层是系统与用户直接交互的层面，实现web前端界面和页面之间的跳转，该层使用业务层提供的现成服务来满足用户需求，这一层只需要编写JSP和ACTION的代码，实现页面的跳转功能，利用CSS，JavaScript脚本和html以及JSP标签设计页面的前端界面。业务层由一个个Service构成，每个Service作为一个程序的特定模块完成应用功能，Service层之间相互独立，Service层调用DAO层的方法对后台数据库执行操作，把数据库中的数据传递到JSP页面显示给用户。持久层屏蔽底层数据库的连接，为业务层Service提供一个简洁、统一、面向对象的数据访问接口，各层之间传递参数值时以对象的方式进行传递，同时操作数据库也以对象的形式存取数据库。

其架构图如下图所示。

图5-1 系统架构图

### 5.2系统部分代码示例

### 5.2.1登录模块

<body>

<dl class="admin\_login">

<dt>

<form name="form1" method="post" action="jspmxssfglxtshfwB4?ac=adminlogin&a=a">

<strong><font color=white>学分制学费管理系统</font></strong>

<em>Management System</em>

</dt>

<dd class="user\_icon">

<input type="text" placeholder="账号" class="login\_txtbx" name="username" id="username"/>

</dd>

<dd class="pwd\_icon">

<input type="password" placeholder="密码" class="login\_txtbx" name="pwd" id="pwd"/>

</dd>

<dd>

<div style="padding-top:10px;">

<font color=white>权限：</font><select name="cx" id="cx">

<option value="管理员">管理员</option>

<option value="院系学生">学生</option>

</select>

</div>

</dd>

<dd class="val\_icon">

<div class="checkcode">

<input type="text" placeholder="验证码" maxlength="4" class="login\_txtbx" id="pagerandom" name="pagerandom">

</div>

&nbsp; <a href="javascript:loadimage();"><img alt="看不清请点我！" name="randImage" id="randImage" src="image.jsp" width="60" height="20" border="1" align="absmiddle" style="padding-top:10px;"> </a>

</dd>

<dd>

<input type="submit" value="立即登陆" class="submit\_btn"/> <input name="login" type="hidden" id="login" value="1">

</dd>

</form></dl></body>

### 5.2.2学生缴费模块

### 5.2.3公告信息模块

<body >

<p>公告列表：</p>

<table width="100%" border="1" align="center" cellpadding="3" cellspacing="1" bordercolor="00FFFF" style="border-collapse:collapse">

<tr>

<td width="40" align="center" bgcolor="CCFFFF">序号</td>

<td bgcolor='#CCFFFF'>公告内容</td>

<td width="158" align="center"bgcolor='#CCFFFF'>发布者</td>

<td width="138" align="center" bgcolor="CCFFFF">添加时间</td>

<td width="60" align="center" bgcolor="CCFFFF">操作</td>

</tr>

<%

new CommDAO().delete(request," gonggao ");

String url = "gonggao\_list2.jsp?1=1";

String sql = "select \* from gonggao";

if(request.getParameter("fasongzhe")=="" ||request.getParameter("fasongzhe")==null ){}else{sql=sql+" and fasongzhe like '%"+request.getParameter("fasongzhe")+"%'";}

sql+=" order by id desc";

ArrayList<HashMap> list = PageManager.getPages(url,15,sql, request);

int i=0;

for(HashMap map:list){

i++;

%>

<tr>

<td width="30" align="center"><%=i %></td>

<td><%=map.get("xiaoxineirong") %></td>

<td><%=map.get("fasongzhe") %></td>

<td width="138" align="center"><%=map.get("addtime") %></td>

<td width="60" align="center">

<a href=" gonggao \_updt.jsp?id=<%=map.get("id")%>">修改</a>

<a href=" gonggao \_list2.jsp?scid=<%=map.get("id") %>" onclick="return confirm('真的要删除？')">删除</a> <!--qiatnalijne--> </td>

</tr>

### 5.2.4用户管理模块

<%

new CommDAO().delete(request,"yonghuzhuce");

String url = "yonghuzhuce\_list.jsp?1=1";

String sql = "select \* from yonghuzhuce where 1=1 ";

if(request.getParameter("yonghuming")=="" ||

request.getParameter("yonghuming")==null ){}else{sql=sql+" and

yonghuming like '%"+request.getParameter("yonghuming").trim()+"%'";}

if(request.getParameter("xingming")=="" ||

request.getParameter("xingming")==null ){}else{sql=sql+" and

xingming like '%"+request.getParameter("xingming").trim()+"%'";}

if(request.getParameter("youxiang")=="" ||

request.getParameter("youxiang")==null ){}else{sql=sql+" and

youxiang like '%"+request.getParameter("youxiang").trim()+"%'";}

if(request.getParameter("dianhua")=="" ||

request.getParameter("dianhua")==null ){}else{sql=sql+" and dianhua

like '%"+request.getParameter("dianhua").trim()+"%'";}

if(request.getParameter("shenfenzheng")=="" ||

request.getParameter("shenfenzheng")==null ){}else{sql=sql+" and

shenfenzheng like '%"+request.getParameter("shenfenzheng").trim

()+"%'";}

sql+=" order by id desc";

//out.print(sql);

ArrayList<HashMap> list = PageManager.getPages(url,10,sql,

request );

int i=0;

for(HashMap map:list){

i++;

%>

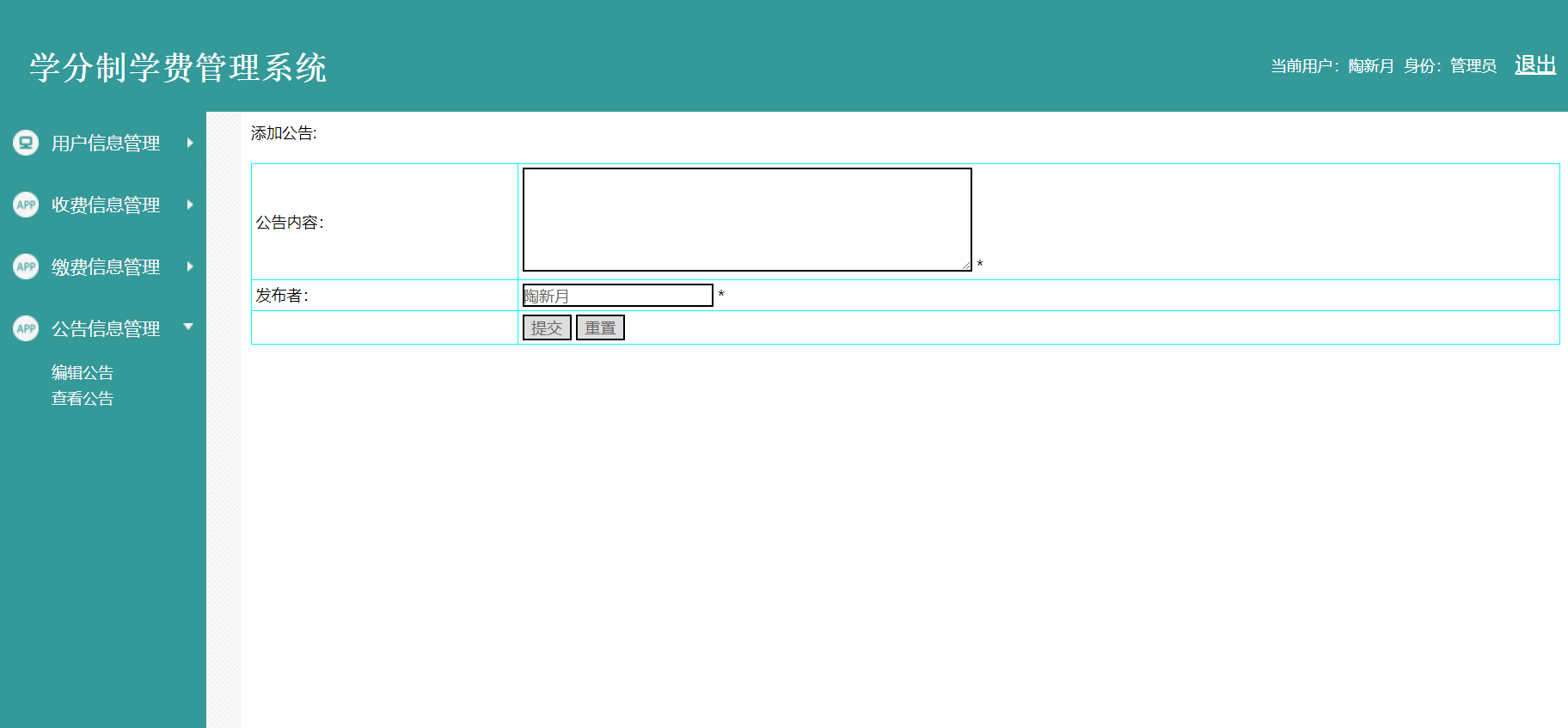
### 5.3系统界面实现（贴界面实际图）

### 5.3.1登录模块



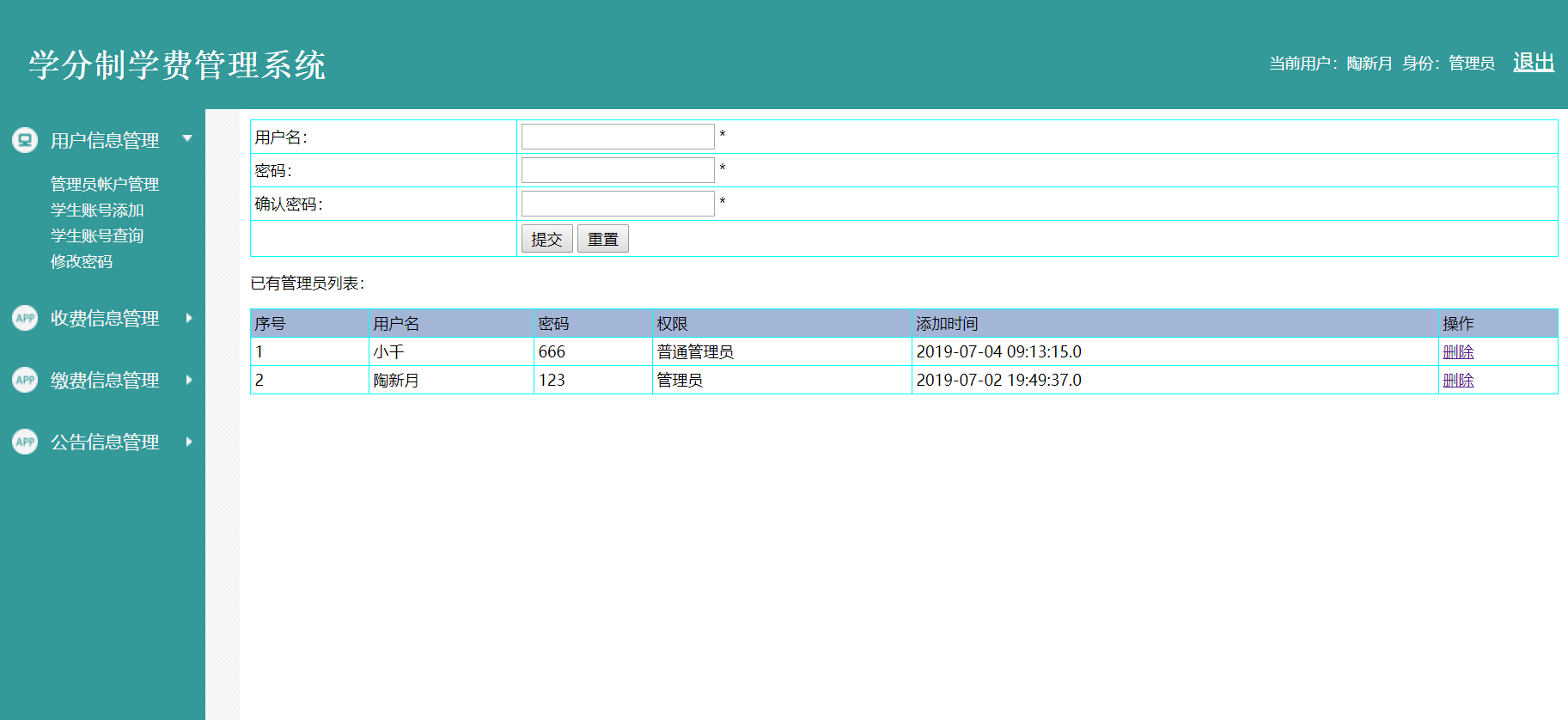
### 5.3.2学生缴费模块

### 5.3.3公告信息模块





### 5.3.4用户管理模块



### 5.4系统测试

### 5.5 系统实施应用

## 第六章 系统测试与部署

## 第七章 其他说明

## 第八章 反思日志