

湖南科技大学计算机科学与工程学院

Web 编程课程设计报告

题目：学生信息管理系统

学号：1005020201

姓名：叶剑飞

班级：10 网络二班

指导老师：周新莲

完成时间：2013 年 9 月 13 日

目录

1	系统需求分析	3
1.1	系统功能分析	3
1.2	系统功能模块设计	3
1.3	与其它系统的关系	3
1.4	数据流程图	3
2	数据库设计	4
2.1	数据库需求分析	4
2.2	数据库概念结构设计	5
2.3	数据库逻辑结构设计	6
2.4	数据库的建立	7
2.4.1	数据库的建立	7
2.4.2	初始数据的输入	10
3	各功能模块的设计与实现	11
3.1	功能说明	11
3.1.1	用户登录	11
3.1.2	登录用户管理	11
3.1.3	学生个人信息输入	11
3.1.4	学籍变更情况的输入	11
3.1.5	奖励情况的输入	11
3.1.6	处罚情况的输入	12
3.2	用户界面设计	12
3.3	各功能模块的实现	17
3.3.1	用户登录界面	17
3.3.2	登录用户管理界面	17
3.3.3	学生个人信息输入界面	17
3.3.4	学籍变更情况输入界面	17
3.3.5	奖励情况输入界面	17
3.3.6	处罚情况输入界面	18
4	系统实现	18
4.1	浏览器 JavaScript 脚本样例	18
4.2	服务器端 PHP 脚本样例	22
5	系统发行	26

1 系统需求分析

学校每年都有新生入学，老生毕业，还有其他各种人事变动。如何有效地管理这些学生的信息，帮助学校和名师掌握学生的情况，就是学生信息管理系统需要完成的功能。

1.1 系统功能分析

本系统主要的功能是收集学生的个人信息，以便向教师提供每个学生在校的情况。系统的主要功能有：

- 学生个人信息的输入，包括：姓名、性别、院系、生日、籍贯、生源所在地等。
- 学生流动情况的输入，包括：转系、休学、复学、退学、毕业。
- 奖惩情况的输入。
- 学生个人情况查询和修改，包括流动情况和奖罚情况。

1.2 系统功能模块设计

更具系统功能要求可以讲系统分解成几个模块来分别设计应用程序界面，如图 1 所示。

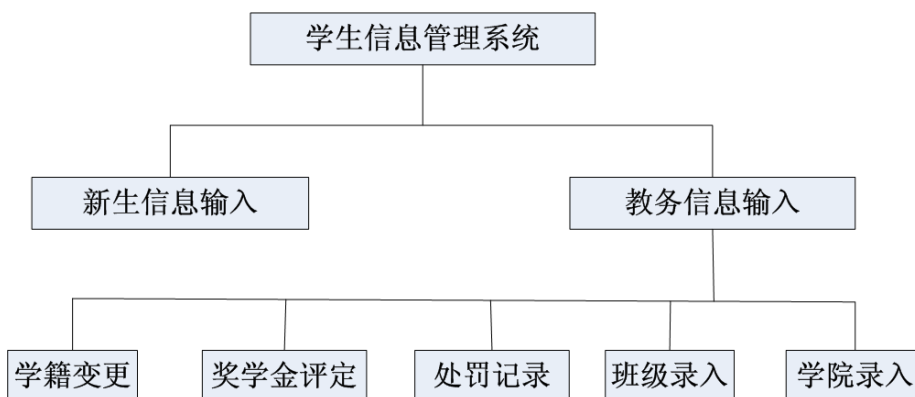


图 1: 学生信息管理系统功能模块图

1.3 与其它系统的关系

学生信息管理系统是校园信息管理系统的一个组成部分。它为其它系统，如班级信息管理系统、教学管理系统、成绩单系统、宿舍分配系统等，提供学生的基本情况。同时需要其它系统提供的班级设置和院系设置等情况。这些系统可以使用同一个数据库，直接相互调用所需的数据。

1.4 数据流程图

学生信息管理系统的数据流程图如图 2 所示。

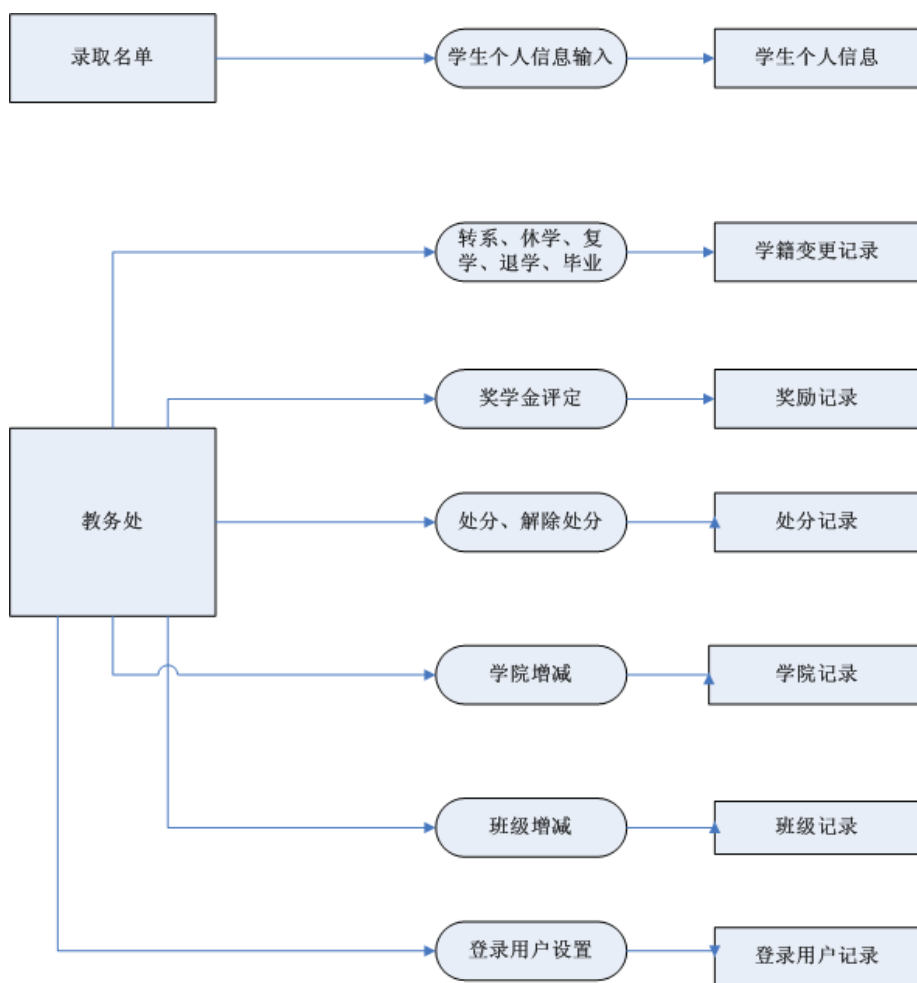


图 2: 学生信息管理系统数据流程图

2 数据库设计

2.1 数据库需求分析

根据上一届的数据流程图，可以列出以下记录学生信息所需的数据项和数据结构：

- 学生：学号、姓名、性别、班级、所属院系、生日、籍贯。
- 学籍变更记录：记录号、学号、变更情况、记录时间、详细描述
- 奖励记录：记录号、学号、奖励级别、记录时间、详细描述
- 处罚记录：记录号、学号、处罚级别、记录时间、详细描述

所需的外部数据支持

- 班级：班级编号、编辑名称、所属院系
- 院系：代码、名称

2.2 数据库概念结构设计

图 3 是本系统所需数据的数据库模型图。

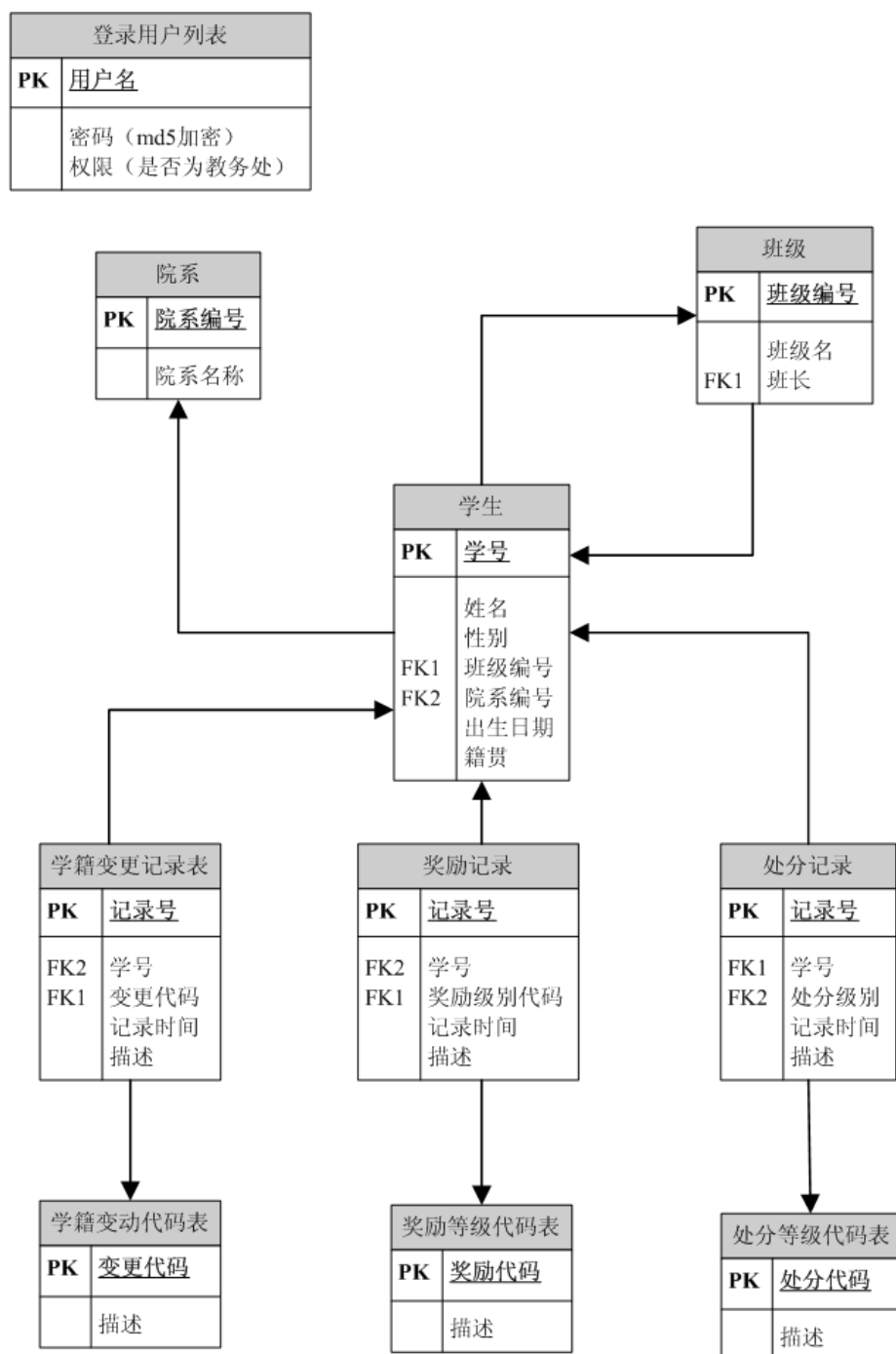


图 3: 学生管理系统的数据库图

2.3 数据库逻辑结构设计

根据数据库模型图，需要设计 4 个数据表来存放学生信息。为了系统的完整，系统中包括了应用程序设计中所需的 2 个外部数据表。这 9 个数据表的结构如表 1 到表 9 所示。

表 1: student 学生个人信息表

字段名	数据类型	是否可空	说明
studentid	varchar(255)	否	学号（主键）
name	varchar(255)	是	姓名
sex	varchar(10)	是	性别
class	int	是	所属班级编号（外键）
department	int	是	所属院系编号（外键）
birthday	date	是	出生日期
native_place	varchar(255)	是	籍贯

表 2: changes 学籍变更信息表

字段名	数据类型	是否可空	说明
id	int auto_increment	否	记录号（主键）
studentid	int	否	学号（外键）
changes	varchar(255)	是	变更代码（外键）
rec_time	date	是	记录时间
description	text	是	描述

表 3: reward 奖励记录信息表

字段名	数据类型	是否可空	说明
id	int auto_increment	否	记录号（主键）
studentid	int	否	学号（外键）
levels	varchar(255)	否	奖励级别代码（外键）
rec_time	date	是	记录时间
description	text	是	描述

表 4: punishment 处罚记录信息表

字段名	数据类型	是否可空	说明
id	int auto_increment	否	记录号（主键）
studentid	int	否	学号（外键）
levels	varchar(255)	否	处罚级别代码（外键）
rec_time	date	是	记录时间
description	text	是	描述

表 5: department 院系信息表

字段名	数据类型	是否可空	说明
id	int auto_increment	否	学院编号（主键）
name	varchar(255)	否	学院名称

表 6: class 班级信息表

字段名	数据类型	是否可空	说明
id	int auto_increment	否	班级编号（主键）
name	varchar(255)	否	班级名称
monitor	varchar(255)	是	班长学号（外键）

表 7: change_code 学籍变动代码表

字段名	数据类型	是否可空	说明
code	varchar(255)	否	学籍变动代码（主键）
description	varchar(255)	否	描述

表 8: reward_code 奖励等级代码表

字段名	数据类型	是否可空	说明
code	varchar(255)	否	奖励等级代码（主键）
description	varchar(255)	否	描述

表 9: punish_code 处罚等级代码表

字段名	数据类型	是否可空	说明
code	varchar(255)	否	处罚等级代码（主键）
description	varchar(255)	否	描述

2.4 数据库的建立

2.4.1 数据库的建立

这次数据库课程设计，我用的是 MySQL 数据库，建库过程如下：

```
CREATE DATABASE students_info;
```

```
USE students_info;
```

```
Table structure for table login_users
```

```
DROP TABLE IF EXISTS login_users;
```

```

CREATE TABLE login_users (
    username varchar(255) NOT NULL,
    password varchar(255) NOT NULL,
    admin bool NOT NULL DEFAULT false ,
    PRIMARY KEY (username)
) ;

—
— Table structure for table department
—

DROP TABLE IF EXISTS department;

CREATE TABLE department (
    id integer NOT NULL AUTO_INCREMENT,
    name varchar(255) NOT NULL,
    PRIMARY KEY (id),
    UNIQUE KEY name (name)
) ;

—
— Table structure for table class
—

DROP TABLE IF EXISTS class;

CREATE TABLE class (
    id integer NOT NULL AUTO_INCREMENT,
    name varchar(255) NOT NULL,
    monitor varchar(255) DEFAULT NULL,
    PRIMARY KEY (id)
) ;

—
— Table structure for table student
—

DROP TABLE IF EXISTS student;

CREATE TABLE student (
    id varchar(255) NOT NULL DEFAULT '',
    name varchar(255) NOT NULL,
    sex varchar(10) NOT NULL,
    class integer DEFAULT NULL,
    department integer DEFAULT NULL,
    birthday date DEFAULT NULL,
    native_place varchar(255) DEFAULT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY class REFERENCES class(id),
    FOREIGN KEY department REFERENCES department(id)

```



```

);

ALTER TABLE class
ADD FOREIGN KEY (monitor)
REFERENCES student(id);

--
-- Table structure for table change_code
--

DROP TABLE IF EXISTS change_code;

CREATE TABLE change_code (
  code varchar(255) NOT NULL,
  description varchar(255) DEFAULT NULL,
  PRIMARY KEY (code)
) ;

--
-- Table structure for table changes
--

DROP TABLE IF EXISTS changes;

CREATE TABLE changes (
  id integer NOT NULL AUTO_INCREMENT,
  studentid integer DEFAULT NULL,
  changes varchar(255) DEFAULT NULL,
  rec_time date DEFAULT NULL,
  description text,
  PRIMARY KEY (id),
  FOREIGN KEY studentid REFERENCES student(id),
  FOREIGN KEY changes REFERENCES change_code(code)
) ;

--
-- Table structure for table punish_levels
--

DROP TABLE IF EXISTS punish_levels;

CREATE TABLE punish_levels (
  code varchar(255) DEFAULT NULL,
  description text
) ;

--
-- Table structure for table punishment
--

```

```

DROP TABLE IF EXISTS punishment;

CREATE TABLE punishment (
    id integer NOT NULL AUTO_INCREMENT,
    studentid integer DEFAULT NULL,
    levels varchar(255) DEFAULT NULL,
    rec_time date DEFAULT NULL,
    description text,
    PRIMARY KEY (id),
    FOREIGN KEY studentid REFERENCES studentid(id),
    FOREIGN KEY levels REFERENCES punish_levels(code)
);

--
-- Table structure for table reward_levels
--

DROP TABLE IF EXISTS reward_levels;

CREATE TABLE reward_levels (
    code varchar(255) NOT NULL,
    description text,
    PRIMARY KEY (code)
);

--
-- Table structure for table reward
--

DROP TABLE IF EXISTS reward;

CREATE TABLE reward (
    id integer NOT NULL AUTO_INCREMENT,
    studentid integer DEFAULT NULL,
    levels varchar(255) DEFAULT NULL,
    rec_time date DEFAULT NULL,
    description text,
    PRIMARY KEY (id),
    FOREIGN KEY studentid REFERENCES studentid(id),
    FOREIGN KEY levels REFERENCES reward_levels(code)
);

```

2.4.2 初始数据的输入

首先，为了保证用户可以正常登录，我先导入 login_user 的数据。其次，为了保证外键能用，我先 change_code、reward_levels 和 punish_levels 的数据。再导入 class 和 department 的数据。然后其他表格才能用。数据导入如下：

```

INSERT INTO login_users VALUES ( 'admin',
'cc6879ff2b0d1fe6dec8269c9d77aa86', true );

```

```
INSERT INTO change_code VALUES ('0','转系'),('1','休学'),  
('2','复学'),('3','退学'),('4','毕业');
```

```
INSERT INTO reward_levels VALUES ('0','校特等奖学金'),  
('1','校一等奖学金'),('2','校二等奖学金'),  
('3','校三等奖学金'),('4','系一等奖学金'),  
('5','系二等奖学金'),('6','系三等奖学金');
```

```
INSERT INTO punish_levels VALUES ('0','警告'),  
('1','严重警告'),('2','记过'),('3','记大过'),  
('4','开除');
```

其余的东西都可以在网页上插入，无需在数据库终端插入。

3 各功能模块的设计与实现

3.1 功能说明

3.1.1 用户登录

这个设置为站点的主页，只需键入账号和密码，单击“登录”即可。若账号密码错误，则无法登录；若账号密码正确，将根据不同权限，重定向到不同页面。若该用户已经登录，则将被 HTTP 302 重定向到对应页面。

3.1.2 登录用户管理

只有教务处有权限访问，游客和其余用户访问这个页面将被禁止（浏览器将收到 HTTP/1.1 403 Forbidden 的响应头）。密码经过 md5 加密的，以保证安全。

3.1.3 学生个人信息输入

班主任登录后，跳转此页面。只有录入名单的权限，不能修改或删除学生个人信息。输入学生信息后，单击“录入数据”即可。只有班主任和教务处才有权限访问这个页面。游客访问这个页面将被禁止（浏览器将收到 HTTP/1.1 403 Forbidden 的响应头）。

3.1.4 学籍变更情况的输入

可以增删改学生的个人信息。只有教务处才有权限访问此页面，游客和其它用户登录此界面将访问这个页面将被禁止（浏览器将收到 HTTP/1.1 403 Forbidden 的响应头）。

3.1.5 奖励情况的输入

输入奖学金的评定结果，单击“新增”。只有教务处才有权限访问此页面，游客和其它用户登录此界面将访问这个页面将被禁止（浏览器将收到 HTTP/1.1 403 Forbidden 的响应头）。

3.1.6 处罚情况的输入

这个是处分录入表。只有教务处才有权限访问此页面，游客和其它用户登录此界面将访问这个页面将被禁止（浏览器将收到 HTTP/1.1 403 Forbidden 的响应头）。

3.2 用户界面设计

完成数据库创建和功能说明后，我们可以经一部进行下一步工作，即设计用户界面。这次数据库课程设计，我采用的是 B/S 模式，用 PHP 写的界面，以网页形式呈现。主要有以下几个界面



图 4: 用户登录

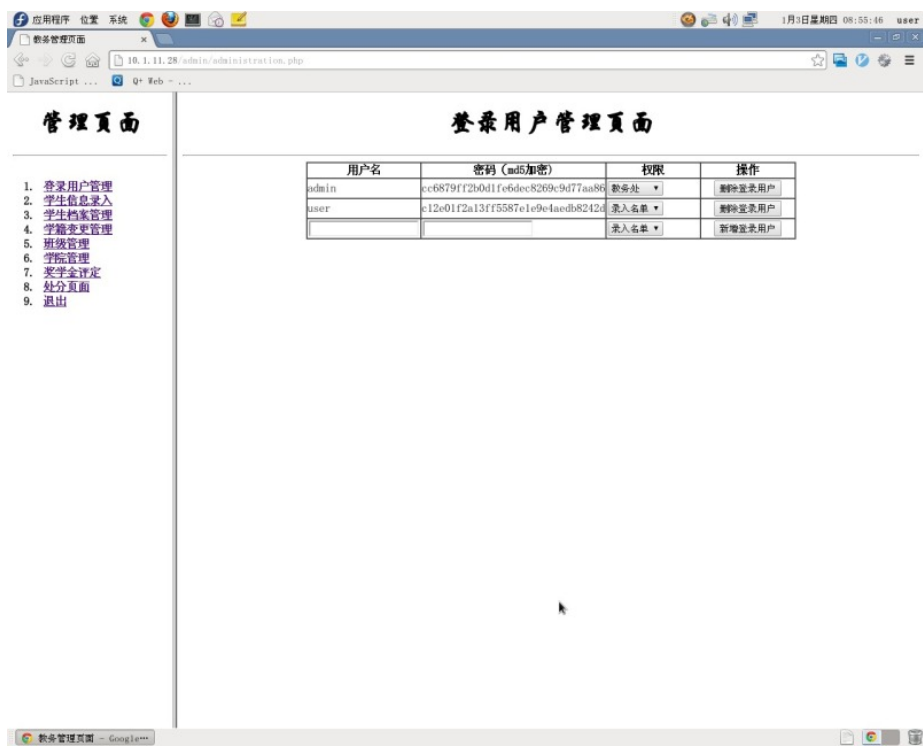


图 5: 登录用户管理页面



图 6: 学生个人信息输入

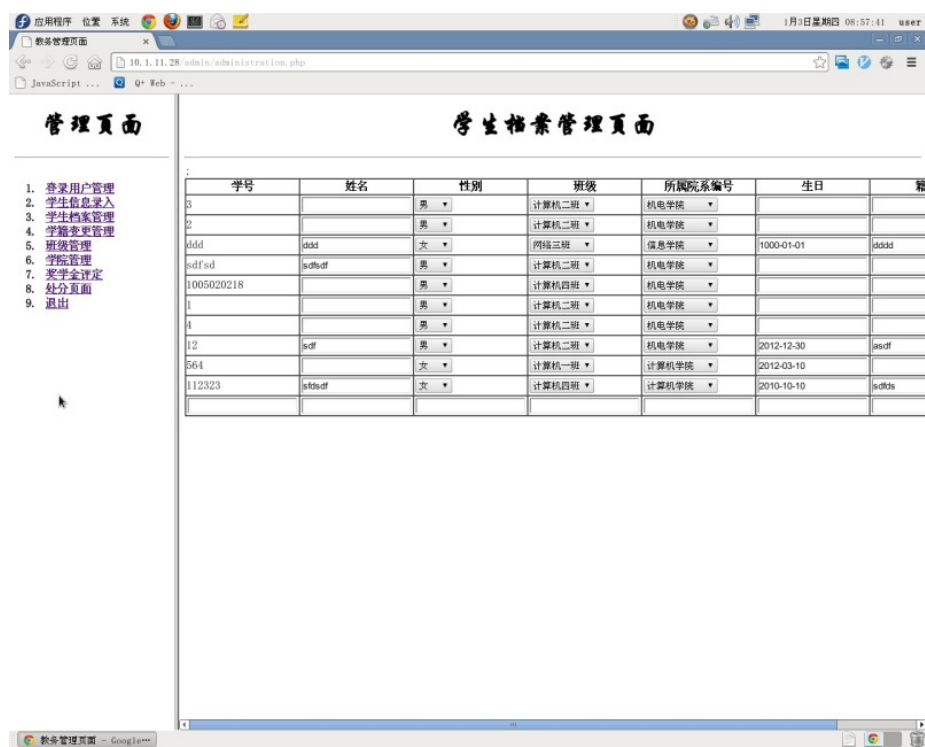


图 7: 学籍变更情况的输入

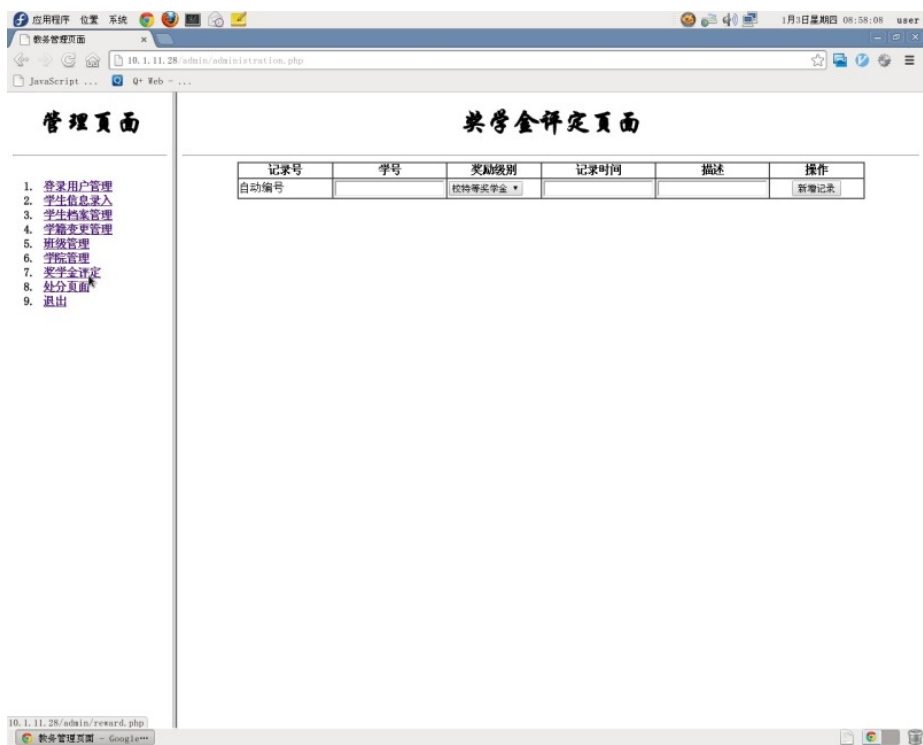


图 8: 奖励情况的输入



图 9: 处罚情况的输入

3.3 各功能模块的实现

3.3.1 用户登录界面

把网页上填写的账号、密码，用 AJAX 技术以表单形式发送 HTTP 请求报文到服务器，服务器用 PHP 脚本相应 AJAX 的 HTTP 请求，把提交上来的密码 md5 加密后，把账号和 md5 加密过的密码，用于查询 MySQL 数据库。查到匹配，则查其权限，根据其权限，设置 SESSION 值，并指明重定向的页面，结果以 XML 文件的形式回发。AJAX 读取到 HTTP 响应报文，并用 JavaScript 的 XML 解析函数解析 XML 文件，然后运行重定向语句。

3.3.2 登录用户管理界面

如果 SESSION 值不是教务处，则发出的 HTTP 响应头以“HTTP/1.1 403 Forbidden”为状态行，HTTP 响应报文无正文，浏览器会显示如下所示的错误信息。

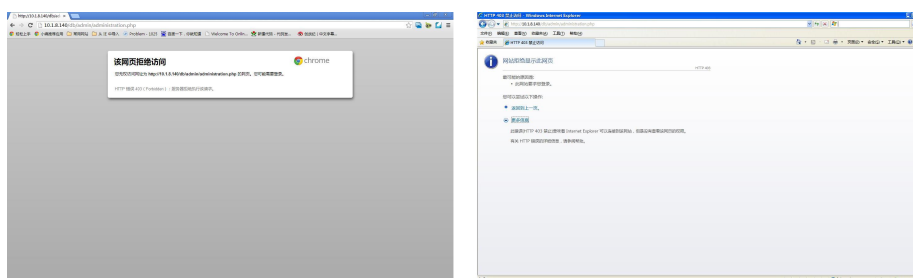


图 10: HTTP/1.1 403 Forbidden

如果是教务处身份登录成功，那么访问这个以表格的形式显示注册用户的账号和经 md5 加密过的密码，用 PHP 用 MySQL API 访问 MySQL 数据库的信息并显示即可。教务处可以新增或删除注册用户。浏览器同样使用 AJAX 和服务端交互。服务器端 PHP 脚本仍然动态生成并发送 XML 文件。

3.3.3 学生个人信息输入界面

任何注册用户都可访问这个页面，但游客（服务端无 SESSION 值）不可访问，否则将发送 HTTP 响应头“HTTP/1.1 403 Forbidden”。本页面添加学生信息。与上面相同，B/S 交互同样采用 AJAX+XML 技术，PHP 用 MySQL API 访问 MySQL 数据库。

3.3.4 学籍变更情况输入界面

只有 SESSION 值为教务处的才能访问，与上面相同，B/S 交互同样采用 AJAX+XML 技术，PHP 用 MySQL API 访问 MySQL 数据库。

3.3.5 奖励情况输入界面

只有 SESSION 值为教务处的才能访问，与上面相同，B/S 交互同样采用 AJAX+XML 技术，PHP 用 MySQL API 访问 MySQL 数据库。

3.3.6 处罚情况输入界面

只有 SESSION 值为教务处的才能访问，与上面相同，B/S 交互同样采用 AJAX+XML 技术，PHP 用 MySQL API 访问 MySQL 数据库。

4 系统实现

各个页面的代码基本相同。浏览器 JavaScript 脚本基本都是 AJAX 交互代码和 XML 解析代码；服务端 PHP 脚本基本是读取提交表单，并用 MySQL API 访问 MySQL 数据库。然后回发 XML 数据。这里提供一个简单样例，其余的相仿，限于篇幅，不再一一列出。

4.1 浏览器 JavaScript 脚本样例

```
// ajax.js
function urlencode( str )
{
    str = encodeURIComponent( str );
    str = str.replace(/%20/g, "+");
    str = str.replace(/(%0D)?(%0A)/g, "%0D%0A");
    return str;
}

function CreateHttpRequest()
{
    if (window.XMLHttpRequest)
        return new XMLHttpRequest();
    else if (window.ActiveXObject)
    {
        try
        {
            return new ActiveXObject("Msxml2.XMLHTTP.6.0");
        }
        catch(e)
        {
            try
            {
                return new ActiveXObject("Msxml2.XMLHTTP");
            }
            catch(e)
            {
                try
                {
                    return new ActiveXObject("Microsoft.XMLHTTP");
                }
                catch(e)
                {
                    return null;
                }
            }
        }
    }
}
```

```

        }
    }
}
else
{
    return null;
}
}

// login.js
var httpRequest = null;

function ShowResult()
{
    var response = null;
    if ( httpRequest.readyState == 4 )
    {
        if ( httpRequest.status == 200 )
        {
            response = httpRequest.responseXML.documentElement;
            if ( response.getElementsByTagName("success")[0].firstChild.nodeValue == "success" )
                window.location = response.getElementsByTagName("location")[0].firstChild.nodeValue;
            else
                window.alert(response.getElementsByTagName("text")[0].firstChild.nodeValue);
            response = null;
        }
        else
        {
            window.alert("服务器发生异常");
        }
        httpRequest = null;
    }
}

function check()
{
    try
    {
        var submitstr = null;
        if ( document.getElementById("username").value == "" )
        {
            window.alert("账号不能为空");
            return false;
        }
        else
        {
            submitstr = "username=" + encodeURIComponent(document.getElementById("username").value);
            if ( document.getElementById("password").value == "" )
            {
                window.alert("密码不能为空");
            }
        }
    }
    catch (e)
    {
        window.alert("发生异常");
    }
}

```

```

        return false;
    }
    else
    {
        submitstr += "&";
        submitstr += "password=" + urlencode( document.getElementById( "password" ).value );
        httpRequest = CreateHttpRequest();
        if ( !httpRequest )
        {
            window.alert( "您的浏览器不支持 AJAX，请升级您的浏览器" );
            return false;
        }
        httpRequest.onreadystatechange = ShowResult;
        httpRequest.open( "POST", "login.php", true );
        httpRequest.setRequestHeader( "Content-Type", "application/x-www-form-urlencoded" );
        httpRequest.send( submitstr );

        return true;
    }
}
}
catch(e)
{
    return false;
}
return false;
}

```

```

// check.js
function is_valid_date ( year, month, day )
{
    if ( month == "00" || day == "00" )
        return false;
    if ( month > 12 )
        return false;
    if ( month == "01" || month == "03" || month == "05" || month == "07" || month == "08" || month == "10" || month == "12" )
    {
        if ( day > 31 )
            return false;
    }
    else if ( month == "04" || month == "06" || month == "09" || month == "11" )
    {
        if ( day > 30 )
            return false;
    }
    else // month == 2
    {
        if ( is_leap_year( year ) )
        {

```

```

        if ( day > 29 )
            return false;
    }
    else
    {
        if ( day > 28 )
            return false;
    }
}
return true;
}

function check_date( dateinput )
{
    var regex = /^[1-2][0-9][0-9][0-9])-([0-1][0-9])-([0-3][0-9])$/g;
    if ( regex.test( dateinput ) )
    {
        var year = dateinput.replace( regex, "$1" );
        var month = dateinput.replace( regex, "$2" );
        var day = dateinput.replace( regex, "$3" );

        if ( is_valid_date( year, month, day ) )
        {
            return true;
        }
        else
        {
            window.alert( "您输入的出生日期不合法。不存在这样的时间。
");
            return false;
        }
    }
    else
    {
        window.alert( "您输入的出生日期格式不合法。应以_yyyy-mm-dd_形
式输入。");
        return false;
    }
}

function check()
{
    if ( document.getElementById("studentid").value == "" )
    {
        window.alert( "学号不能为空！");
        return false;
    }
    else if ( document.getElementById("birthday").value != "" )

```

```

    {
        return check_date( document.getElementById("birthday").value );
    }
    else
        return true;
}

```

4.2 服务器端 PHP 脚本样例

```

<?php
// login.php
ob_start();
session_start();
header("Pragma: no-cache");
if ( !isset($_POST["username"]) || !isset($_POST["password"]) )
{
    header("HTTP/1.0 400 Bad Request");
    header("Content-Type: text/plain; charset=UTF-8");
    exit(0);
}
require_once("conf/sqlexec.php");
header("Content-Type: text/xml; charset=UTF-8");
unset($_SESSION["admin"]);
unset($_SESSION["user"]);
unset($_SESSION["username"]);
session_destroy();
session_start();
$success = "false";
$text = "账号或密码错误";
$result = sqlexec( "select * from login_users where username='".mysql_escape_string($_POST["username"])."' and password='".mysql_escape_string($_POST["password"])."'");
while ( $row = mysql_fetch_array( $result ) )
{
    if ( $row["password"] == md5($_POST["password"]) )
    {
        $success = "true";
        $text = "登录成功";
        $_SESSION["username"] = $row["username"];
        if ( $row["admin"] )
        {
            $_SESSION["admin"] = true;
            $location = "admin/administration.php";
        }
        else
        {
            $_SESSION["user"] = true;
            $location = "user.php";
        }
    }
}
?>

```

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
<success><?php echo $success; ?></success>
<text><?php echo $text; ?></text>
<location><?php echo $location; ?></location>
</result>
```

```
<?php
// changes.php
ob_start();
session_start();
header("Pragma: no-cache");
if ( !isset($_SESSION["admin"]) )
{
    header("HTTP/1.0 403 Forbidden");
    header("Content-Type: text/plain; charset=UTF-8");
    exit(0);
}
require_once("conf/sqlexec.php");
ob_end_clean();
ob_get_clean();
if ( isset($_POST["method"]) )
{
    header("Content-Type: text/xml; charset=UTF-8");
    if ( $_POST["method"] == "add_record" )
    {
        $i = 0;
        if ( isset($_POST["studentid"]) )
        {
            $colname[$i] = "studentid";
            $colval[$i] = "" . mysql_escape_string($_POST["studentid"]) . "";
            $i ++;
        }
        if ( isset($_POST["changes"]) )
        {
            $colname[$i] = "changes";
            $colval[$i] = "" . mysql_escape_string($_POST["changes"]) . "";
            $i ++;
        }
        if ( isset($_POST["rec_time"]) )
        {
            $colname[$i] = "rec_time";
            $colval[$i] = "" . mysql_escape_string($_POST["rec_time"]) . "";
            $i ++;
        }
        if ( isset($_POST["description"]) )
        {
            $colname[$i] = "description";
            $colval[$i] = "" . mysql_escape_string($_POST["description"]) . "";
            $i ++;
        }
    }
}
```

```

    }
    $str = "insert into changes (".join(",",$colname)."). values (".join(",",$
    $result = sqlexec( $str );
    if ( $result )
    {
        $success = "true";
        $text = "增加记录成功";
    }
    else
    {
        $success = "false";
        $text = "增加记录失败";
    }
}
else // del_record
{
    $str = "delete from changes where id='".mysql_escape_string($_POST["id"])."
    $result = sqlexec( $str );
    if ( $result )
    {
        $success = "true";
        $text = "删除记录成功";
    }
    else
    {
        $success = "false";
        $text = "删除记录失败";
    }
}
?>
<?xml version="1.0" encoding="UTF-8"?>
<result>
    <success><?php echo $success; ?></success>
    <text><?php echo $text; ?></text>
</result>
<?php
    exit(0);
}
header("Content-Type: text/html; charset=UTF-8");
require_once("conf/sqlexec.php");
$result = sqlexec("select * from changes");
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.c
<html xmlns="http://www.w3.org/1999/xhtml" lang="zh" xml:lang="zh">
<head>
<title> 学生档案管理页面 </title>
<script type="text/javascript" src="js/ajax.js"></script>
<script type="text/javascript" charset="UTF-8" src="js/add_del_record.js"></sc
<!--[if lt IE 9]>
<script type="text/javascript" src="../js/IE9.js"></script>

```



```

<![endif]-->
<link type="text/css" rel="stylesheet" href="../css/user.css" />
</head>

<body>
<h1> 学生档案管理页面 </h1>

<hr />

<table align="center" border="1" cellspacing="0" cellpadding="0">
<?php
$cols = mysql_num_fields($result);
echo "<tr>";
echo "<th>记录号</th>";
echo "<th>学号</th>";
echo "<th>变更代码</th>";
echo "<th>记录时间</th>";
echo "<th>描述</th>";
echo "<th>操作</th>";
echo "</tr>";
while ( $row = mysql_fetch_array($result) )
{
    echo "<tr id=\"".urlencode($row["id"]).\"\">";

    echo "<td>";
    echo $row["id"];
    echo "</td>";
    echo "<td>";
    if ( $row["studentid"] == null )
        echo "暂未填写";
    else
        echo $row["studentid"];
    echo "</td>";
    echo "<td>";
    $change_result = sqlexec( "select * from change_code where code='". $row["chan
    $change_row = mysql_fetch_array($change_result);
    echo $change_row["description"];
    echo "</td>";
    echo "<td>";
    if ( $row["rec_time"] == null )
        echo "暂未填写";
    else
        echo $row["rec_time"];
    echo "</td>";
    echo "<td>";
    if ( $row["description"] == null )
        echo "暂未填写";
    else
        echo $row["description"];
    echo "</td>";

```

```

    echo "<td style=\"text-align:center\">";
    echo "<input type=\"button\" value=删除记录=\"\" onclick=\"del_record('".urlencode($change_row["code"])."\">";
    echo "</td>";
    echo "</tr>";
}
?>
<tr>
<td> 自动编号 </td>
<td><input type="text" id="studentid" /></td>
<td><select id="changes"><?php
$change_result = sqlexec( "select * from change_code" );
while ( $change_row = mysql_fetch_array($change_result) )
    echo "<option value=\"". $change_row["code"]. "\">". $change_row["description"].
?></option></td>
<td><input type="text" id="rec_time" /></td>
<td><input type="text" id="description" /></td>
<td style="text-align:center"><input type="button" value="新
增记录" onclick="add_record()" /></td>
</tr>
</table>
</body>
</html>

```

5 系统发行

如果需要，可以向我索要源代码（studentinfo_1.0.tgz）或者安装包（studentinfo_1.0_all.deb），以及 MySQL 数据库的 SQL 脚本导入文件 data.sql。