

序号: 08

学号: 13434216



常州大学

课程设计报告

课程名称: 课程设计（三）

设计题目: 教务学籍管理系统

学生姓名: 林洛钊

学院（系）: 信息数理学院 专业班级: 信息 132

校内指导教师: 张燕新 专业技术职务: 讲师

设计时间: 2016 年 9 月 5 日 ~ 2016 年 9 月 16 日

教务学籍管理系统

摘 要：教务学籍管理系统是典型的信息管理系统，其开发主要包括后台数据库的建立及前端应用程序的开发两个方面。对于前者，要求建立起数据一致性和完整性强、数据安全性好的库，而对于后者则要求应用程序具有功能完备，易使用等特点。

本系统采用 B/S 模式，使用三层架构，以 MySQL 作为后台数据库开发工具，使用 VS2010 进行 ASP.Net 项目程序开发。它界面友好使用方便，操作简单，因为采用角色授权机制，保证系统的安全稳定性。可以实现对学生基本数据以及学生成绩的不同层次的管理和操作，使本校的学籍管理工作科学化、规范化、高效化。

本文详尽介绍了开发系统的全过程，包括系统的需求分析、系统功能设计、数据库的设计、系统的实施，重点介绍了系统功能与数据库的设计及系统的实现过程。对系统结构、系统功能设计的流程做了全面而详尽的阐述，然后针对系统的各个功能模块做了具体的分析与研究。通过对系统功能及数据流的分析，得到系统数据库的逻辑结构，并在此基础上完成数据库物理结构的创建，从而实现对各种信息的查询、更新等操作。

关键字：ASP.Net；B/S 模式；三层架构；MySQL 数据库；VS2010；

目 录

| | |
|-------------------|-----|
| 摘 要 | II |
| 目 录 | III |
| (一) 系统背景 | 1 |
| (二) 开发目的及意义 | 1 |
| 二、可行性研究 | 1 |
| (一) 经济可行性 | 1 |
| (二) 操作可行性 | 1 |
| 三、系统设计 | 1 |
| (一) 需求分析 | 1 |
| (二) 数据库设计 | 2 |
| 四、功能实现 | 4 |
| (一) 用户管理模块 | 4 |
| 1. 登录模块 | 4 |
| 2. 密码修改模块 | 5 |
| (二) 管理员模块 | 6 |
| 1. 学生管理 | 6 |
| 2. 教师管理 | 9 |
| 3. 课程管理 | 11 |
| (三) 教师模块 | 14 |
| 1. 教学管理 | 14 |
| 2. 开课管理 | 14 |
| (四) 学生模块 | 16 |
| 1. 选课退课功能 | 16 |
| 2. 毕业检查 | 18 |
| 五、总结 | 20 |
| 参考文献 | 20 |

一、系统概述

（一）系统背景

随着我国高等教育的普及发展和在校学生人数不断增加，学生管理工作越来越复杂。长期以来，学籍管理都是依赖人工进行的，面对日益众多的学籍信息，其工作量可想而知。由于人工管理存在着大量的不可控因素，必然会造成一些不必要的错误发生。因此设计一个跨平台的，网络化的学籍管理系统不但能使教务人员从复杂的手工学籍管理任务中解脱出来，而且对于推动学校管理信息化的发展也将起到非常重要的作用。

（二）开发目的及意义

为推动学校管理信息化的发展，使教务人员从复杂的手工学籍管理任务中解脱出来，故决定使用 ASP.Net 及 MySQL 设计一个跨平台的，网络化的学籍管理系统。这不仅将解决实际问题，也是在信息时代下的新研究，新突破。

二、可行性研究

（一）经济可行性

与其它的系统相比，本系统独立性相当强，只需要投入一台计算机的即可，不会给学校机构构成经济负担。而它的成功运行将给人们节省大量的时间和精力，使教教学籍管理大大的提高效率，该系统是一个全新的系统，它将改变原来落后的人工管理办法，这将会缩小不必要的开支，在系统运行环境中，软硬件都无需做巨大投资，只利用现有计算机设备即可。因此在经济上可行的。

（二）操作可行性

现行教教学籍管理模式采用大量手工操作与少量微机操作相结合，而新开发的系统则全用微机来处理整个过程，另外操作人员要求也不高，只需对 WINDOWS 操作熟练，加之对本系统的操作稍加学习即可使用，而且本系统可视性非常好，所以在操作上是行的。

三、系统设计

（一）需求分析

1. 系统管理：包括用户信息（用户名、密码等）的添加，更改、删除和退出系统等功能。

2. 教师管理：包括教师基本信息的添加，更改、删除功能。
3. 学生管理：包括学生基本信息的添加，更改、删除功能。
4. 课程管理：包括教师对课程信息的添加，更改、删除及学生选课退课功能。
5. 成绩管理：包括教师更新学生成绩，学生查看成绩功能。

(二) 数据库设计

1. 用户信息表

用户信息表储存用户编号，用户密码，用户权限，具体如下表 3.1 所示：

表 3.1 用户信息表

| 字段名 | 字段类型 | 字段说明 | 是否为空 | 其他 |
|----------|---------|------|------|----|
| UserName | varchar | 用户名 | 是 | 主键 |
| PassWord | varchar | 密码 | 是 | |
| UserRole | varchar | 用户权限 | 是 | |

2. 教师信息表

教师信息表储存教师编号，教师姓名，教师年龄，教师性别，教师专业，具体如下表 3.2 所示：

表 3.2 教师信息表

| 字段名 | 字段类型 | 字段说明 | 是否为空 | 其他 |
|--------------|---------|------|------|----|
| TeacherID | char | 教师编号 | 是 | 主键 |
| TeacherName | varchar | 教师姓名 | 是 | |
| TeacherSex | varchar | 教师性别 | 是 | |
| TeacherAge | varchar | 教师年龄 | 是 | |
| TeacherMajor | varchar | 教师专业 | 是 | |

3. 学生信息表

学生信息表储存学生编号，学生姓名，学生年龄，学生性别，学生专业，学生班级，具体如下表 3.3 所示：

表 3.3 学生信息表

| 字段名 | 字段类型 | 字段说明 | 是否非空 | 其他 |
|------------------------|---------|--------|------|----|
| StudentID | char | 学生编号 | 是 | 主键 |
| StudentName | varchar | 学生姓名 | 是 | |
| StudentSex | varchar | 学生性别 | 是 | |
| StudentAge | varchar | 学生年龄 | 是 | |
| StudentMajor | varchar | 学生专业 | 是 | |
| StudentClass | varchar | 学生班级 | 是 | |
| StudentAllScoreRequire | varchar | 毕业学分要求 | 否 | |

4. 课程基本表

课程信息表储存课程编号, 课程名称, 完成课程所得学分, 如下表 3.4 所示:

表 3.4 课程信息表

| 字段名 | 字段类型 | 字段说明 | 是否非空 | 其他 |
|------------|---------|------|------|----|
| CourseID | char | 课程编号 | 是 | 主键 |
| CourseName | varchar | 课程名称 | 是 | |
| Credit | varchar | 学分 | 是 | |

5. 教师开课表

教师开课表储存课程编号, 教师编号, 上课地点等信息, 具体如下表 3.5 所示:

表 3.5 教师开课表

| 字段名 | 字段类型 | 字段说明 | 是否非空 | 其他 |
|-----------|---------|------|------|------|
| CourseID | char | 课程编号 | 是 | 主键 1 |
| TeacherID | char | 教师编号 | 是 | 主键 2 |
| Where | varchar | 上课地点 | 否 | |

6. 选课信息表

选课信息表储存课程编号, 教师编号, 学生编号, 选课状态, 学生成绩等信息, 具体如下表 3.6 所示:

表 3.6 选课信息表

| 字段名 | 字段类型 | 字段说明 | 是否非空 | 其他 |
|-----------|---------|------|------|------|
| CourseID | char | 课程编号 | 是 | 主键 1 |
| TeacherID | char | 教师编号 | 是 | 主键 2 |
| StudentID | char | 学生编号 | 是 | 主键 3 |
| Status | varchar | 选课状态 | 否 | |
| Score | varchar | 学生成绩 | 是 | |

四、功能实现

（一）用户管理模块

1. 登录模块

验证用户身份，防止非法用户使用系统。登录界面如图 4.1 所示：



图 4.1 用户登录界面

用户点击登录按钮后，浏览器将用户输入的用户名及密码发送至服务器端验证，代码如下：

```
protected void OnButtonLogin(object sender, EventArgs e)
{
    string UserName = TextUserName.Text.ToString().Trim();
    string PassWord = TextPassWord.Text.ToString().Trim();
    int status=0;
    status = transfer.Login(UserName, PassWord);
    switch (status)
    {
        case 0://用户名或密码错误
            Response.Write("<script language=javascript>alert(' 您输入的用户名或密码错误请检查后重新输入! )</script>");
            break;
        case 1://学生登陆成功
            Session["UserName"] = UserName;
            Session["UserRole"] = "Student";
    }
}
```

```
Response.Write("<script language=javascript>alert(' 登录成功!')</script>");
Response.Redirect("S_info.aspx");
break;
case 2://教师登陆成功
Session["UserName"] = UserName;
Session["UserRole"] = "Teacher";
Response.Write("<script language=javascript>alert(' 登录成功!')</script>");
Response.Redirect("T_info.aspx");
break;
case 3://管理员登陆成功
Session["UserName"] = UserName;
Session["UserRole"] = "Admin";
Response.Write("<script language=javascript>alert(' 登录成功!')</script>");
Response.Redirect("A_info.aspx");
break;
}
}
```

2. 密码修改模块

用于用户修改自身密码，修改界面如图 4.2 所示：

图 4.2 用户修改密码界面

用户点击修改按钮后，浏览器将用户输入的用户名及密码发送至服务器端验证并执行修改过程，代码如下：


```
protected void OnButtonConfirm(object sender, EventArgs e)
{
    if (transfer.ModifyPassword(Session["UserName"].ToString(),
        OldPassword.Text.ToString(), NewPassword.Text.ToString()))
    {
        Response.Write("<script language=javascript>alert(' 密码修改成功! ')</script>");
    }
    else
    {
        Response.Write("<script language=javascript>alert(' 密码修改失败! ')</script>");
    }
}
```

(二) 管理员模块

1. 学生管理

学生管理分为添加学生界面和学生信息维护界面。

添加学生界面如图 4.3 所示：

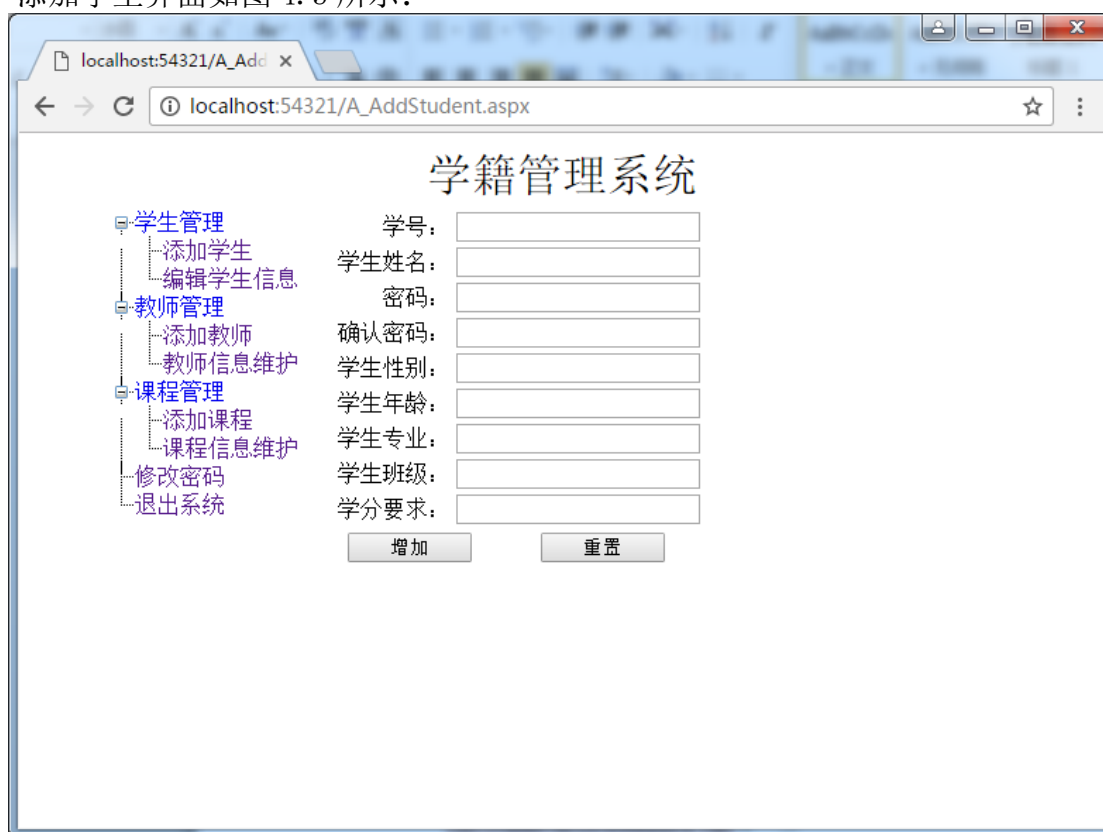


图 4.3 管理员添加学生界面

管理员点击添加按钮后，浏览器将信息发送至服务器端并执行添加信息程序，代码如下：

```
protected void OnButtonAdd(object sender, EventArgs e)
{
```

```
        if (transfer.AddStudent(TextStudentID.Text.ToString(),
TextStudentName.Text.ToString(), TextStudentPassWord.Text.ToString(),
TextStudentSex.Text.ToString(), TextStudentAge.Text.ToString(),
TextStudentMajor.Text.ToString(), TextStudentClass.Text.ToString(),
TextStudentttAllScoreRequire.Text.ToString()))
        {
            Response.Write("<script language=javascript>alert(' 添加学生成功! ')</script>");
            TextStudentID.Text = "";
            TextStudentName.Text = "";
            TextStudentPassWord.Text = "";
            TextConfirmPassWord.Text = "";
            TextStudentSex.Text = "";
            TextStudentAge.Text = "";
            TextStudentMajor.Text = "";
            TextStudentClass.Text = "";
            TextStudentttAllScoreRequire.Text = "";
        }
    else
    {
        Response.Write("<script language=javascript>alert(' 添加学生失败，请检查后重试! ')</script>");
    }
}
```

学生信息维护界面如图 4.4 所示：



图 4.4 管理员学生信息维护界面

管理员对信息编辑后点击更新按钮或点击删除按钮后，浏览器将信息发送至服务器端并执行更新或删除信息程序。

更新代码如下：

```
protected void GridView_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    string ID = GridView.Rows[e.RowIndex].Cells[0].Text.ToString();
    string Name =
        ((TextBox)GridView.Rows[e.RowIndex].Cells[1].Controls[0]).Text.ToString();
    string Sex =
        ((TextBox)GridView.Rows[e.RowIndex].Cells[2].Controls[0]).Text.ToString();
    string Age =
        ((TextBox)GridView.Rows[e.RowIndex].Cells[3].Controls[0]).Text.ToString();
    string Major =
        ((TextBox)GridView.Rows[e.RowIndex].Cells[4].Controls[0]).Text.ToString();
    string Class =
        ((TextBox)GridView.Rows[e.RowIndex].Cells[5].Controls[0]).Text.ToString();
    string AllScoreRequire =
        ((TextBox)GridView.Rows[e.RowIndex].Cells[6].Controls[0]).Text.ToString();
    transfer.UpdateStudent(ID, Name, Sex, Age, Major, Class, AllScoreRequire);
    GridView.EditIndex = -1;
    bind();
}
```

删除代码如下：

```
protected void GridView_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
    string ID = GridView.Rows[e.RowIndex].Cells[0].Text.ToString();
    string Name = GridView.Rows[e.RowIndex].Cells[1].Text.ToString();
    transfer.DeleteStudent(ID);
    GridView.EditIndex = -1;
    bind();
}
```

2. 教师管理

教师管理分为添加教师界面和教师信息维护界面。

添加教师界面如图 4.5 所示：

图 4.5 管理员添加教师界面

管理员点击添加按钮后，浏览器将信息发送至服务器端并执行添加信息程序，代码如下：

```
protected void OnButtonAdd(object sender, EventArgs e)
{
    if (transfer.AddTeacher(TextTeacherID.Text.ToString(),
        TextTeacherName.Text.ToString(), TextTeacherPassWord.Text.ToString(),
        TextTeacherSex.Text.ToString(), TextTeacherAge.Text.ToString(),
        TextTeacherMajor.Text.ToString()))
    {

```

```

Response.Write("<script language=javascript>alert(' 添加教师成功! ')</script>");
TextTeacherID.Text = "";
TextTeacherName.Text = "";
TextTeacherPassWord.Text = "";
TextConfirmPassWord.Text = "";
TextTeacherSex.Text = "";
TextTeacherAge.Text = "";
TextTeacherMajor.Text = "";
}
else
{
    Response.Write("<script language=javascript>alert(' 添加教师失败，请检查后重试! ')</script>");
}
}
}

```

教师信息维护界面如图 4.6 所示：

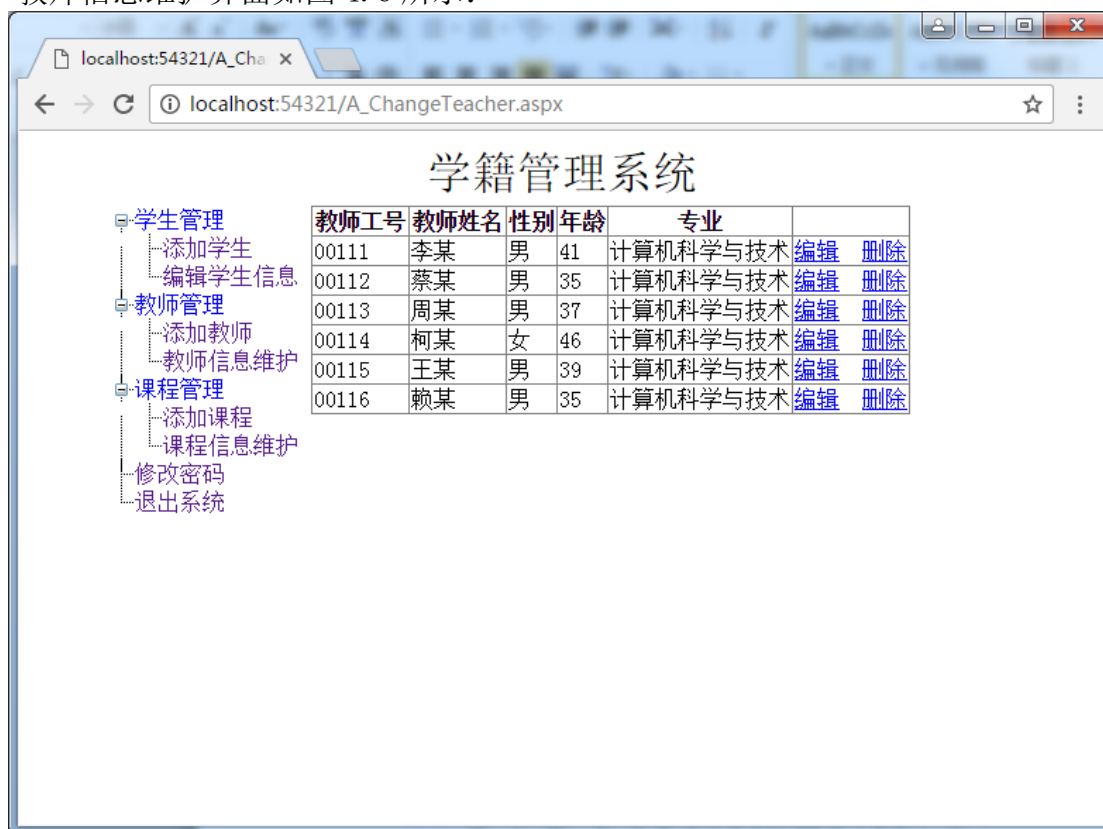


图 4.6 管理员教师信息维护界面

管理员对信息编辑后点击更新按钮或点击删除按钮后，浏览器将信息发送至服务器端并执行更新或删除信息程序。

更新代码如下：

```

protected void GridView_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    string ID = GridView.Rows[e.RowIndex].Cells[0].Text.ToString();
    string Name =

```

```
((TextBox)GridView.Rows[e.RowIndex].Cells[1].Controls[0]).Text.ToString();
    string Sex =
((TextBox)GridView.Rows[e.RowIndex].Cells[2].Controls[0]).Text.ToString();
    string Age =
((TextBox)GridView.Rows[e.RowIndex].Cells[3].Controls[0]).Text.ToString();
    string Major =
((TextBox)GridView.Rows[e.RowIndex].Cells[4].Controls[0]).Text.ToString();
    transfer.UpdateTeacher(ID, Name, Sex, Age, Major);
    GridView.EditIndex = -1;
    bind();
}
```

删除代码如下：

```
protected void GridView_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
    string ID = GridView.Rows[e.RowIndex].Cells[0].Text.ToString();
    string Name = GridView.Rows[e.RowIndex].Cells[1].Text.ToString();
    transfer.DeleteTeacher(ID);
    GridView.EditIndex = -1;
    bind();
}
```

3. 课程管理

课程管理分为添加课程界面和课程信息维护界面。

添加教师界面如图 4.7 所示：

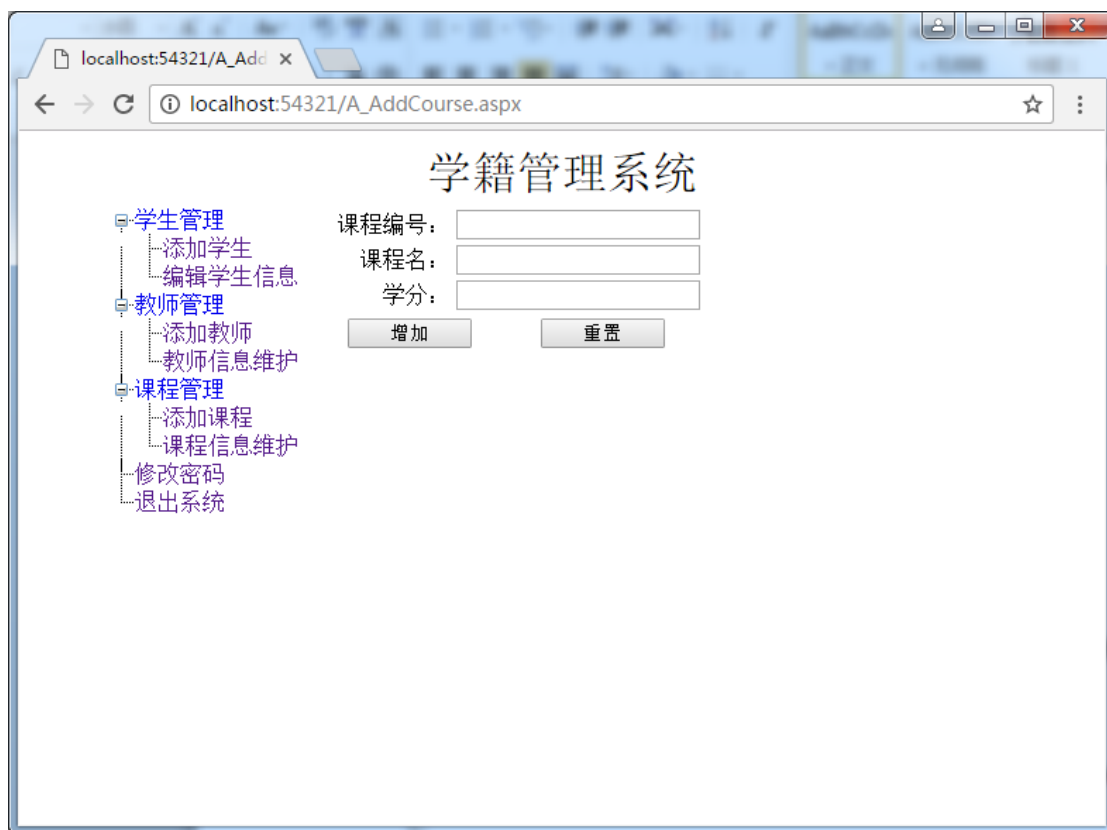


图 4.7 管理员添加课程界面

管理员点击添加按钮后，浏览器将信息发送至服务器端并执行添加信息程序，代码如下：

```
protected void OnButtonAdd(object sender, EventArgs e)
{
    if (transfer.AddCourse(TextCourseID.Text.ToString(), TextCourseName.Text.ToString(),
        TextCredit.Text.ToString()))
    {
        Response.Write("<script language=javascript>alert(' 添加课程成功! ')</script>");
        TextCourseID.Text = "";
        TextCourseName.Text = "";
        TextCredit.Text = "";
    }
    else
    {
        Response.Write("<script language=javascript>alert(' 添加课程失败，请检查后重试! ')</script>");
    }
}
```

课程信息维护界面如图 4.8 所示：

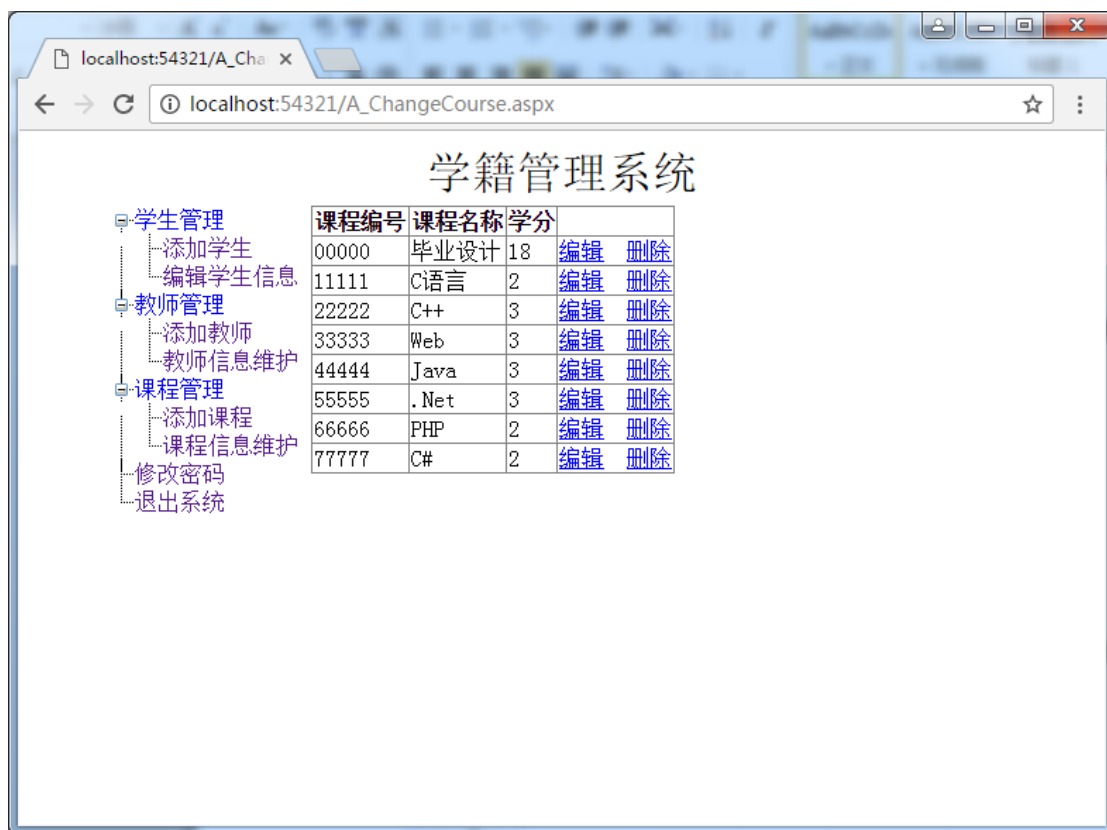


图 4.8 管理员课程信息维护界面

管理员对信息编辑后点击更新按钮或点击删除按钮后,浏览器将信息发送至服务器端并执行更新或删除信息程序。

更新代码如下:

```
protected void GridView_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    string ID = GridView.Rows[e.RowIndex].Cells[0].Text.ToString();
    string Name =
    ((TextBox)GridView.Rows[e.RowIndex].Cells[1].Controls[0]).Text.ToString();
    string Credit =
    ((TextBox)GridView.Rows[e.RowIndex].Cells[2].Controls[0]).Text.ToString();
    transfer.UpdateCourse(ID, Name, Credit);
    GridView.EditIndex = -1;
    bind();
}
```

删除代码如下:

```
protected void GridView_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
    string ID = GridView.Rows[e.RowIndex].Cells[0].Text.ToString();
    transfer.DeleteCourse(ID);
    GridView.EditIndex = -1;
    bind();
}
```


（三）教师模块

1. 教学管理

教师在下拉框在选择自己已开课程后，浏览器加载选修该课程的学生，教师可对每个学生进行成绩录入。界面如图 4.9 所示：

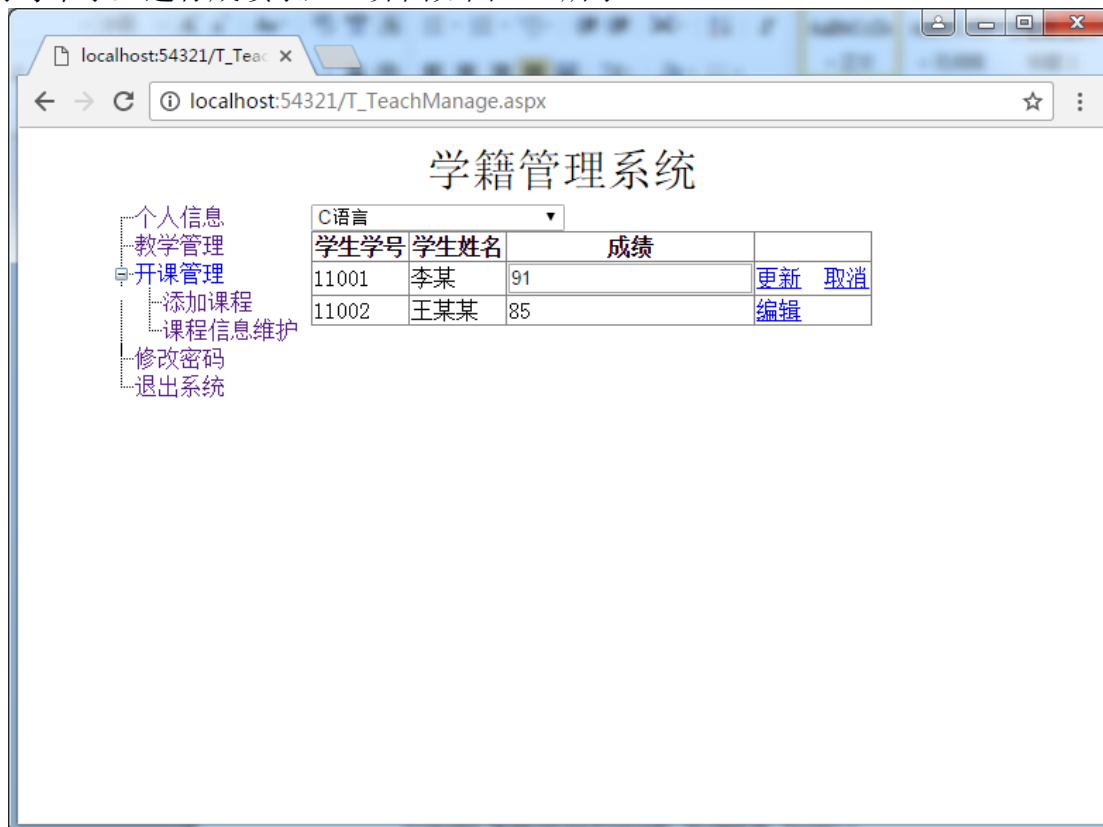


图 4.9 教师教学管理界面

修改成绩代码如下：

```
protected void GridView_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    string CourseID = DropDownList.SelectedValue.ToString();
    string StudentID = GridView.Rows[e.RowIndex].Cells[0].Text.ToString();
    string Score =
((TextBox)GridView.Rows[e.RowIndex].Cells[2].Controls[0]).Text.ToString();
    transfer.UpdateScore(CourseID, StudentID, Score);
    GridView.EditIndex = -1;
    bind();
}
```

2. 开课管理

开课管理分为新开课程界面和课程信息维护界面。

新开课程界面如图 4.10：



图 4.10 教师新开课程界面

教师点击添加按钮后，浏览器将信息发送至服务器端并执行添加信息程序，代码如下：

```
protected void OnButtonAdd(object sender, EventArgs e)
{
    if (transfer.AddTeacherCourse(DropDownList.SelectedValue.ToString(),
    Session["UserName"].ToString(), TextWhere.Text.ToString()))
    {
        Response.Write("<script language=javascript>alert(' 添加课程成功! ')</script>");
        TextWhere.Text = "";
    }
    else
    {
        Response.Write("<script language=javascript>alert(' 添加课程失败，请检查后重试! ')</script>");
    }
}
```

课程信息维护界面如图 4.11 所示：

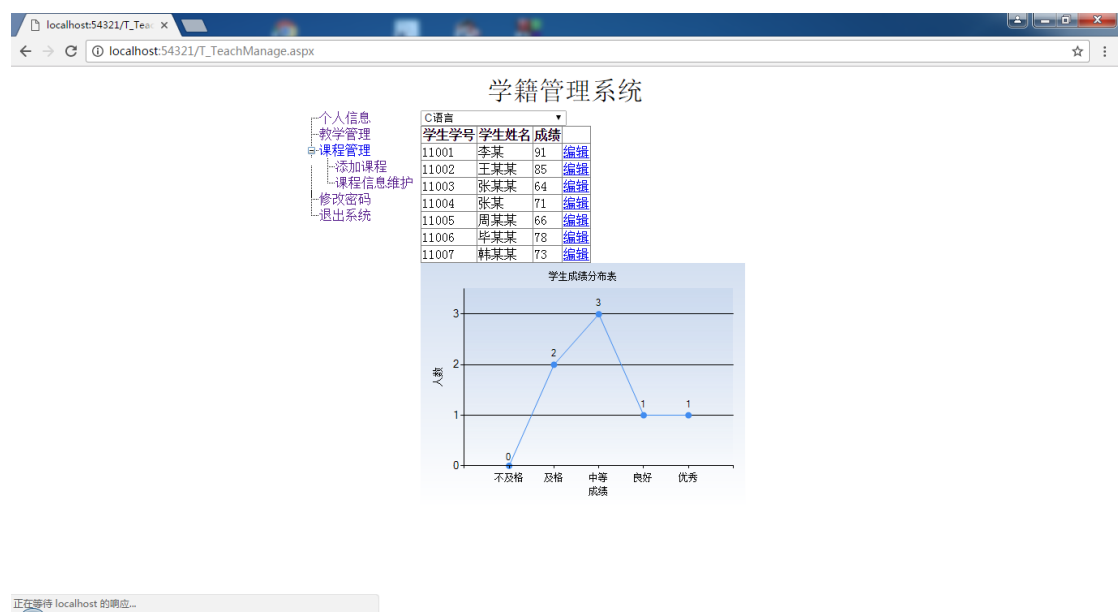


图 4.11 教师课程信息维护界面

教师对信息编辑后点击更新按钮或点击删除按钮后，浏览器将信息发送至服务器端并执行更新或删除信息程序。

更新代码如下：

```
protected void GridView_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    string ID = GridView.Rows[e.RowIndex].Cells[0].Text.ToString();
    string Where =
((TextBox)GridView.Rows[e.RowIndex].Cells[3].Controls[0]).Text.ToString();
    transfer.UpdateTeacherCourse(ID, Session["UserName"].ToString(), Where);
    GridView.EditIndex = -1;
    bind();
}
```

删除代码如下：

```
protected void GridView_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
    string ID = GridView.Rows[e.RowIndex].Cells[0].Text.ToString();
    transfer.DeleteTeacherCourse(ID, Session["UserName"].ToString());
    GridView.EditIndex = -1;
    bind();
}
```

（四）学生模块

1. 选课退课功能

学生在此界面可查看所有可选课程，点击选课或退选按钮后，服务器端将执行对应操作。界面如图 4.12 所示：



图 4.12 学生选课界面

选课退课代码如下：

```
protected void RowCommand(object sender, GridViewCommandEventArgs e)
{
    int index = Convert.ToInt32(e.CommandArgument);
    string CourseID = GridView.Rows[index].Cells[0].Text.ToString();
    string TeacherID = GridView.Rows[index].Cells[2].Text.ToString();
    string StudentID = Session["UserName"].ToString();
    if (e.CommandName == "SelectCourse")
    {
        int ret=transfer.Select(CourseID,TeacherID,StudentID);
        switch (ret)
        {
            case 0:
                Response.Write("<script language=javascript>alert(' 选课操作失败，请稍后重试! ')</script>");
                break;
            case 1:
                Response.Write("<script language=javascript>alert(' 抱歉你已选择该课程! ')</script>");
                break;
            case 2:
                Response.Write("<script language=javascript>alert(' 恭喜，选课成功! ')</script>");
```

```
        break;
    }
    bind();
}
if (e.CommandName == "QuitCourse")
{
    int ret = transfer.Quit(CourseID, StudentID);
    switch (ret)
    {
        case 0:
            Response.Write("<script language=javascript>alert(' 退课操作失败，请稍  
后重试! ')</script>");
            break;
        case 1:
            Response.Write("<script language=javascript>alert(' 抱歉，你未选择该课  
程! ')</script>");
            break;
        case 2:
            Response.Write("<script language=javascript>alert(' 恭喜，退课成功!  
'</script>");
            break;
    }
    bind();
}
}
```

2. 毕业检查

学生在此界面可查看所有已修课程，点击按钮则可检查是否已达到毕业要求，界面如图 4.13 所示：

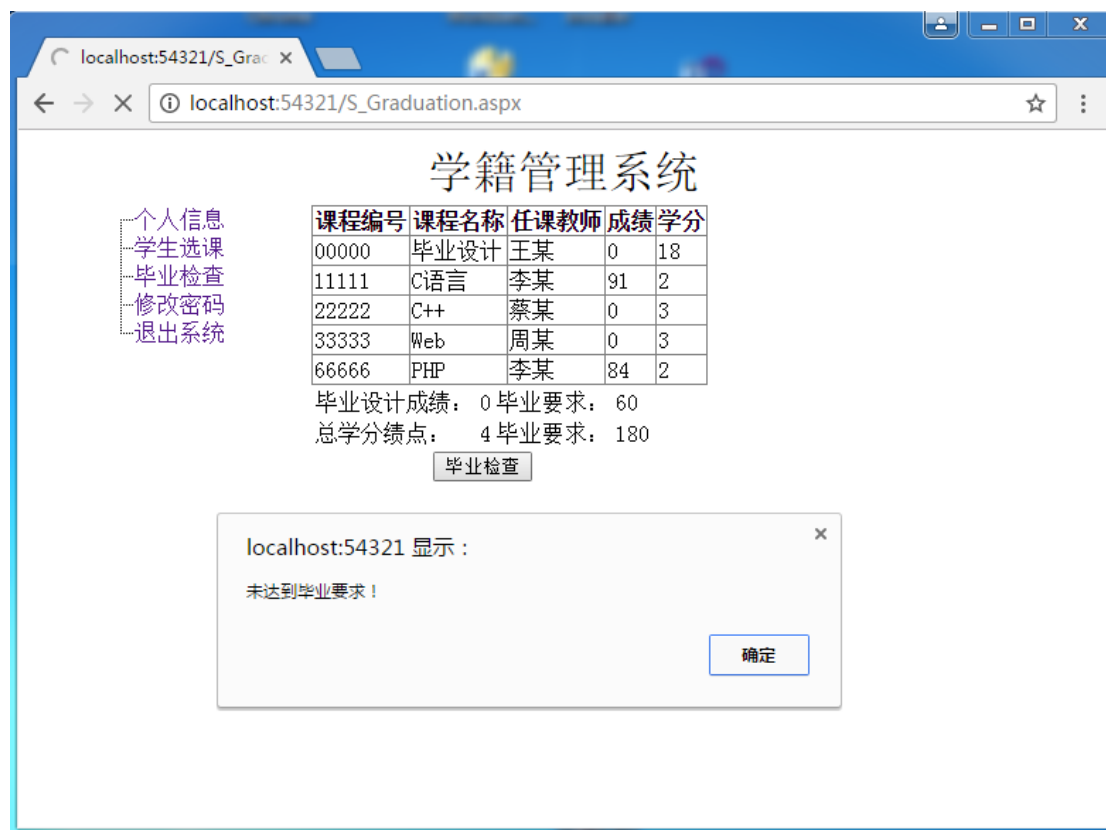


图 4.13 毕业检查界面

检查代码如下:

```
protected void OnButton(object sender, EventArgs e)
{
    bind();
    int GradScore = Int32.Parse(TextGradScore.Text);
    int GradScoreRequire = Int32.Parse(TextGradScoreRequire.Text);
    int AllScore = Int32.Parse(TextAllScore.Text);
    int AllScoreRequire = Int32.Parse(TextAllScoreRequire.Text);

    if ((GradScore >= GradScoreRequire) && (AllScore >= AllScoreRequire))
    {
        Response.Write("<script language=javascript>alert(' 已达到毕业要求!')</script>");
    }
    else
    {
        Response.Write("<script language=javascript>alert(' 未达到毕业要求!')</script>");
    }
}
```

五、总结

通过这次教务学籍管理系统项目的开发,认识到项目开发中需求分析的重要性,对于项目开发的每一步,都有其不容忽视的价值。在项目的实现过程中,掌握了项目开发中的三层架构体系,同时,也对使用 ASP. Net 进行网站项目开发有了进一步的认识。通过三层架构的实现,对软件工程中的低耦合概念有了深入的认识。

参 考 文 献

- [1] 耿超. ASP.NET 4.5 网站开发实例教程[M]. 北京:清华大学出版社, 2015.
- [2] 张荣梅, 赵宝琴, 赵彦霞. ASP.NET 程序设计实用教程:C#版[M]. 北京:北京大学出版社, 2014.
- [3] 国家 863 中部软件孵化器. ASP.NET 从入门到精通[M]. 北京:人民邮电出版社, 2010.
- [4] 龙马工作室. 新编 ASP.NET 从入门到精通[M]. 北京:人民邮电出版社, 2009.