



大连理工大学
DALIAN UNIVERSITY OF TECHNOLOGY

大连理工大学经济管理学院
《信息系统开发工具》课程实验报告

学生选课及成绩查询管理信息系统

学 院（系）： 经济管理学院
专 业： 物流管理
姓 名： 严梓锴
学 号： 201903020
指 导 教 师： 曲刚
完 成 日 期： 2022 年 4 月 26 日

大连理工大学

Dalian University of Technology

目录

1	背景介绍	3
2	技术选型	3
3	需求分析	3
3.1	系统实现的功能流程	4
3.2	管理员角色拥有的功能	4
3.3	学生角色拥有的功能	5
4	数据库设计	6
4.1	数据库关系图	6
4.2	完成系统中数据表的结构设计	6
5	数据库创建脚本	8
6	程序结构	10
6.1	程序结构图	10
6.2	程序文件名及实际含义对应	11
7	界面展示	12
7.1	管理员角色	13
7.2	学生界面	18
7.3	关于界面	21
8	关键技术介绍	22
8.1	左侧点击选项，右侧展开界面	22
8.2	在其他的窗口得到在登录窗口输入的信息	25
8.3	新开设课程时教室使用冲突处理	26
8.4	学生选课时间冲突处理	28
9	优化方案	30
9.1	提高访问数据库的速度	30
9.2	引入教师角色及相应功能界面	31
10	完成情况总结	31
11	使用说明	32
11.1	数据库数据导入	32
11.1.1	方式一：运行数据库脚本后手动添加数据	32
11.1.2	方式二（推荐）：导入 mdf 及 ldf 文件	32
11.2	数据库连接路径修改	33
11.3	提供的供登陆查看的用户名和密码	33

图目录

图 1 系统实现功能流程图.....	4
图 2 数据库关系图.....	6
图 3 程序结构图.....	10
图 4 登录界面.....	12
图 5 管理员主界面.....	13
图 6 管理员——添加管理员界面.....	13
图 7 管理员——修改管理员界面.....	14
图 8 管理员——开设课程界面.....	14
图 9 管理员——查询课程界面.....	15
图 10 管理员——录入成绩界面.....	15
图 11 管理员——统计成绩页面.....	16
图 12 管理员——修改成绩界面.....	16
图 13 管理员——添加学生信息界面.....	17
图 14 管理员——修改学生信息界面.....	17
图 15 管理员——修改密码界面.....	18
图 16 学生登录主界面.....	18
图 17 学生——选择课程界面.....	19
图 18 学生——查询课程界面.....	19
图 19 学生——显示课表界面.....	20
图 20 学生——我的成绩单.....	20
图 21 学生——密码修改界面.....	21

表目录

表 1 学生信息表.....	6
表 2 管理员信息表.....	7
表 3 课程表.....	7
表 4 课程细节表.....	7
表 5 选课表.....	7
表 6 程序文件名及实际含义对应表.....	11

1 背景介绍

学生选课及成绩查询系统是一个学校不可缺少的部分，传统的人工管理档案的方式存在着很多的缺点，如：效率低、保密性差等，所以开发一套综合教务系统管理软件很有必要，它应该具有传统的手工管理所无法比拟的优点，如：检索迅速、查找方便、可靠性高、存储量大、保密性好、寿命长、成本低等。这些优点能够极大的提高学校的管理效率，有助于推进学校的信息化建设。

2 技术选型

- (1) C#的窗体程序开发
- (2) Sql server 数据库

3 需求分析

本系统包括两类用户：学生、管理员。管理员可以通过系统来添加管理员信息、修改管理员信息、添加学生信息、修改学生信息；开设课程、查询课程、录入成绩、统计成绩、修改成绩、修改个人密码等，而学生则可以通过系统来选择课程、查询课程、显示自己的课表、查询自己的成绩单、修改个人密码等等。

3.1 系统实现的功能流程

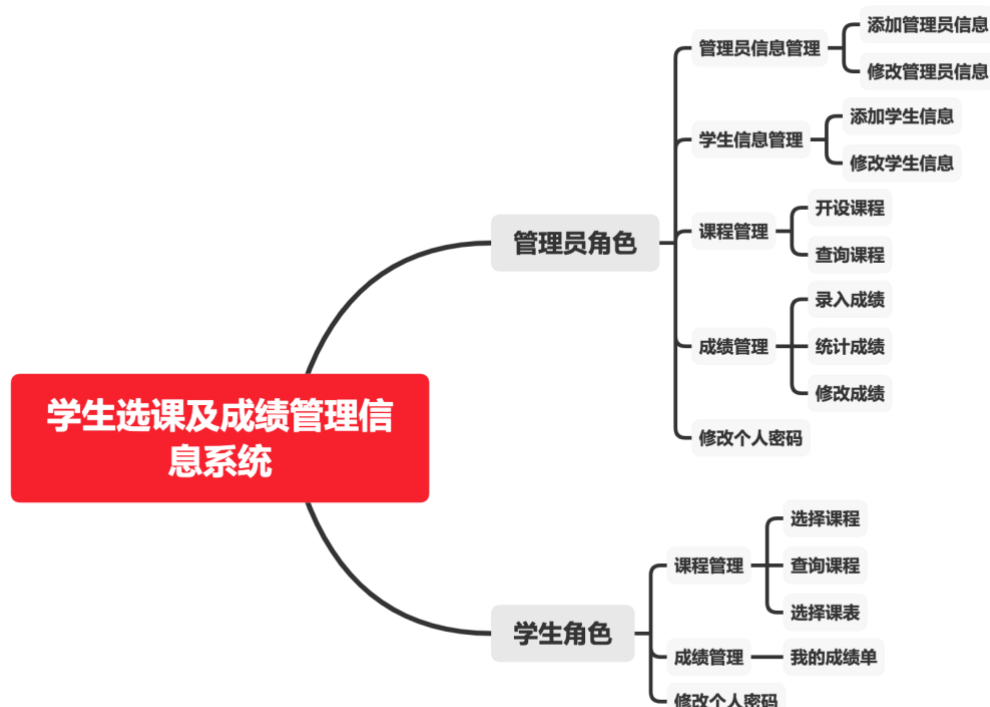


图 1 系统实现功能流程图

3.2 管理员角色拥有的功能

- (1) 管理员信息管理
 - i. 添加管理员信息：管理员可以添加其他的管理员信息。
 - ii. 修改管理员信息：管理员可以删除其他的管理员信息。
- (2) 学生信息管理
 - i. 添加学生信息：管理员可以添加学生信息，这些信息包括学生的学号，密码，年级，专业，出生日期等等。
 - ii. 修改学生信息：管理员可以修改学生的信息；
- (3) 课程管理
 - i. 开设课程：管理员可以录入课程信息，学生就可以浏览到这些课程信息，从而选课。
 - ii. 查询课程：可以通过学期查询课程，或通过学期和课程结合的方式查询课程。
- (4) 成绩管理

- i. **录入成绩：**管理员可以录入学生的成绩。
 - ii. **统计成绩：**管理员可以通过学期和课程信息来统计某门课的学生成绩，平均分、最高分、最低分以及可以通过成绩区间来查询某个学期某门课的学生的成绩信息。并且可以实现排序功能。
 - iii. **修改成绩：**管理员可以对成绩出错的学生的成绩进行修改。
- (5) **修改个人密码**
- i. 管理员可以修改自己的密码。

3.3 学生角色拥有的功能

- (1) **课程管理**
- i. **选择课程：**学生可以根据本学期管理员发布的课程来选课。
 - ii. **查询课程：**可以通过学期查询课程，或通过学期和课程结合的方式查询课程。
 - iii. **显示课表：**学生可以根据自己的选课情况来查询自己的课表。
- (2) **成绩管理**
- i. **我的成绩单：**学生可以根据学期来查询自己某个学期所有选修课程的成绩信息。
- (3) **修改个人密码**
- i. 学生可以修改自己的密码。

4 数据库设计

4.1 数据库关系图

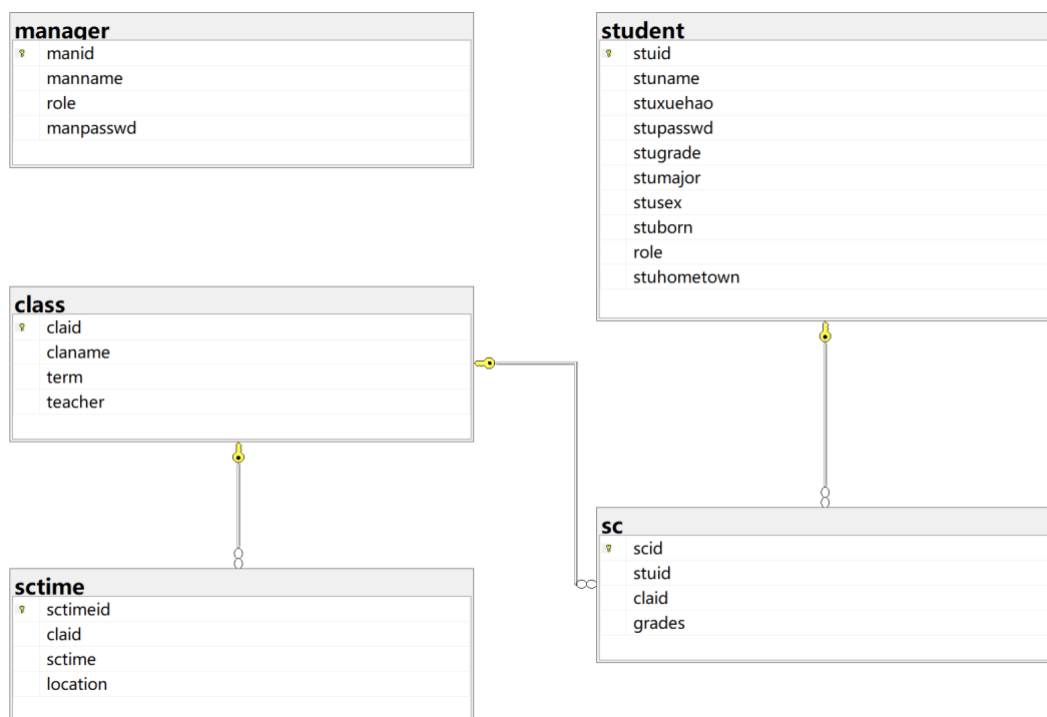


图 2 数据库关系图

4.2 完成系统中数据表的结构设计

表 1 学生信息表

用户表：student				
字段名称	数据类型	可空	默认值	说明
stuid	int	不可		主键，学生 id，无实际意思，每次自增 1
stuname	Nvarchar(30)	不可		学生姓名
stuxuehao	Nvarchar(30)	不可		学生学号，不能重复，登陆时的用户名默认是学号。
stupasswd	Nvarchar(30)	不可		学生密码
stugrade	Nvarchar(30)			学生年级
stumajor	Nvarchar(30)			学生专业
stusex	varchar(2)			学生性别
stuborn	Nvarchar(30)			学生出生日期

role	Nvarchar(30)	学生	角色，学生
stuhometown	Nvarchar(30)		学生籍贯

表 2 管理员信息表

管理员信息表：manager				
字段名称	数据类型	可空	默认值	说明
manid	int	不可		主键，管理员 id，无实际意思，每次自增 1
manname	Nvarchar(30)	不可		管理员用户名
role	Nvarchar(30)	不可	管理员	角色，管理员
manpasswd	Nvarchar(30)	不可		管理员密码

表 3 课程表

课程表：class				
字段名称	数据类型	可空	默认值	说明
claid	int	不可		主键，课程 id，无实际意思，每次自增 1
claname	Nvarchar(30)	不可		课程名
term	Nvarchar(30)	不可		开课的学期
teacher	Nvarchar(30)	不可		开课的教师

表 4 课程细节表

课程细节表：sctime				
字段名称	数据类型	可空	默认值	说明
sctimeid	int	不可		主键，课程细节 id，无实际意思，每次自增 1
claid	int	不可		外键，课程号
sctime	Nvarchar(30)	不可		课程上课的时间
location	Nvarchar(30)	不可		课程上课的地点

表 5 选课表

选课表：sc				
字段名称	数据类型	可空	默认值	说明
scid	int	不可		主键，选课 id，无实际意思，每次自增 1
stuid	int	不可		外键，学生 id
claid	int	不可		外键，课程 id
grades	int	不可		学生成绩

5 数据库创建脚本

相关 sql 脚本代码如下：

```
1. -- 数据库默认设置，老师名字不重复。一个老师开的一门课只能在一个地方上。一个
   班级。
2. -- 学生信息表（学生 id， 年级， 专业， 用户名， 密码， 性别， 年龄）
3. create table student(
4. stuid bigint primary key identity(1,1),-- 用户 id
5. stuname varchar(30),-- 学生的真实姓名
6. stuxuehao varchar(30) not null unique,-- 学生的学号（默认为学生登录的用
   户名）
7. stupasswd varchar(30) not null,-- 密码
8. stugrade varchar(30),-- 年级
9. stumajor varchar(30),-- 专业
10. stusex varchar(2) default '男',-- 性别
11. stuborn varchar(30),-- 出生日期
12. role varchar(30) default '学生',
13. stuhometown varchar(30),-- 籍贯
14.)
15.
16. -- 管理员信息表（管理员 id， 用户名， 密码）
17. create table manager(
18. manid bigint primary key identity(1,1),-- 用户 id
19. manname varchar(30) not null unique,-- 用户名
20. role varchar(30) default '管理员',
21. manpasswd varchar(30) not null,-- 密码
22.)
23.
24. -- 课程表（课程 id， 课程名， 学期， 学时， 老师 id）
25. create table class(
26. claid bigint primary key identity(1,1),-- 课程 id
27. claname varchar(30),-- 课程名
28. term varchar(30),-- 学期
29. teacher varchar(30),-- 老师姓名
30.)
31.
32. -- 上课时间的表
33. create table sctime(
34. sctimeid bigint primary key identity(1,1),-- 上课时间 id
35. claid bigint constraint csctime_id references class(claid),-- 课程的
   id
36. sctime varchar(30),-- 上课时间
37. location varchar(30),-- 上课地点
```

```
38.)  
39.  
40.--选课表(选课id, 课程id, 学生id, 成绩)  
41.create table sc(  
42.scid bigint primary key identity(1,1),--选课id  
43.stuid bigint constraint ssc_id references student(stuid),--课程的id  
44.claid bigint constraint scc_id references class(claid),--课程id  
45.grades bigint --成绩  
46.)
```

6 程序结构

6.1 程序结构图

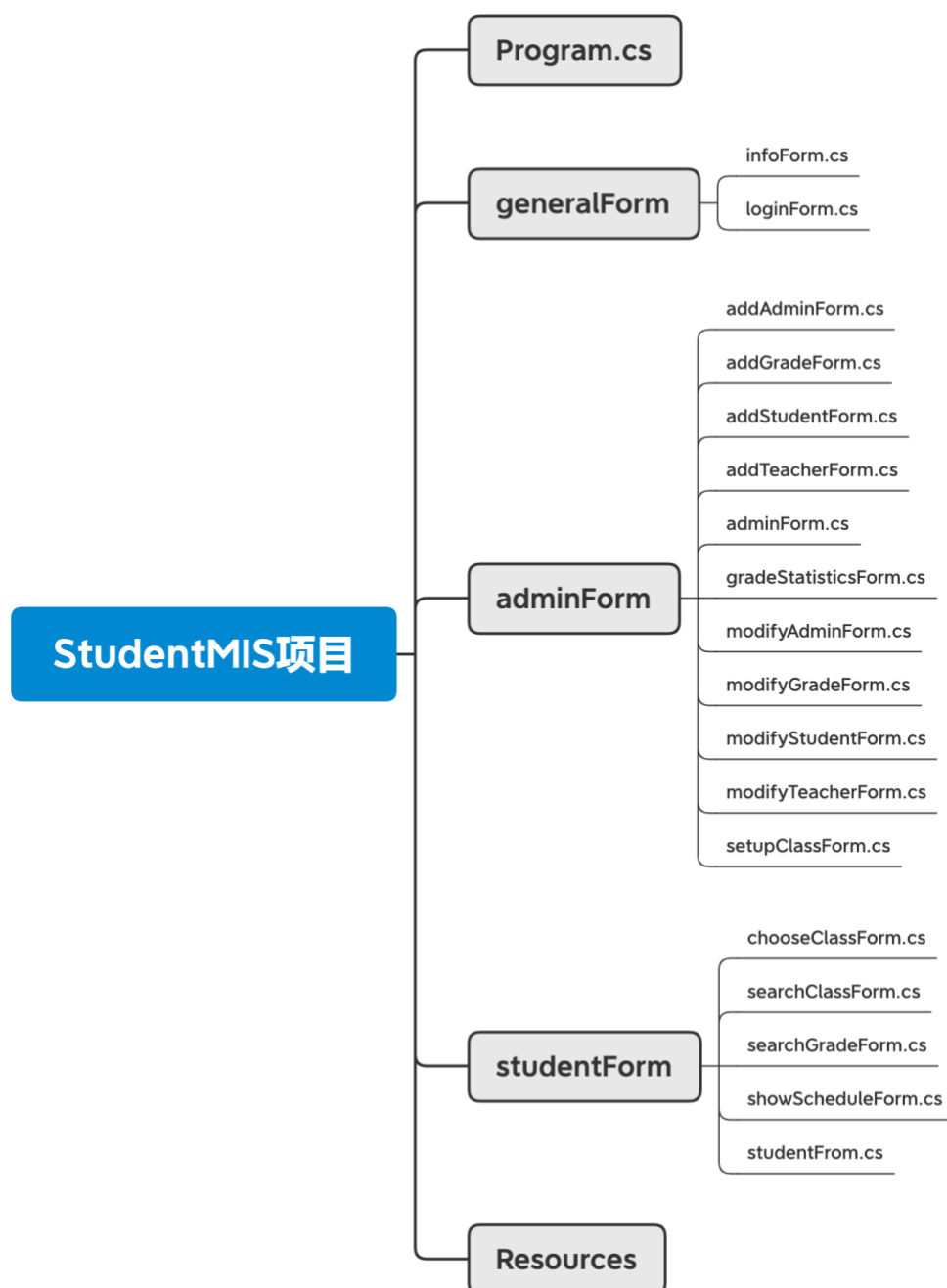


图 3 程序结构图

6.2 程序文件名及实际含义对应

表 6 程序文件名及实际含义对应表

文件名	中文对应	实际含义
generalForm	通用界面	为文件夹,包含所有通用界面的程序设计。
adminForm	管理员界面	为文件夹,包含所有与管理有关的界面的程序设计。
studentForm	学生界面	为文件夹,包含所有与学生有关的界面的程序设计。
infoForm.cs	关于界面	显示本程序的相关基本信息。
loginForm.cs	登陆界面	用于用户的登录。
adminForm.cs	管理员主界面框	管理员主界面框。
addAdminForm.cs	添加管理员界面	管理员进行管理员的新增。
addGradeForm.cs	添加学生成绩界面	管理员对学生的成绩进行添加录入。
addStudentForm.cs	添加学生信息界面	管理员添加新的学生信息。
addTeacherForm.cs	添加教师信息界面	管理员添加新的教师的信息。
gradeStatisticsForm.cs	成绩统计界面	管理员对学生的成绩进行统计输出。
modifyAdminForm.cs	修改管理员信息界面	管理员对已有管理员的信息进行修改。
modifyGradeForm.cs	修改成绩界面	管理员对学生成绩进行修改。
modifyStudentForm.cs	修改学生信息界面	管理员对学生信息进行修改。
modifyTeacherForm.cs	修改教师信息界面	管理员对教师信息进行修改。
setupClassForm.cs	开设课程界面	管理员在系统中输入新开设的课程的信息。
studentFrom.cs	学生主界面框	学生主界面框。
chooseClassForm.cs	选择课程界面	学生进行选课。
searchClassForm.cs	查询课程界面	学生对已开设的课程信息进行查询。

searchGradeForm.cs	查询成绩界面	学生查询自己的成绩信息。
showScheduleForm.cs	显示课表界面	学生可以看到自己当前所有选课的课表。
Resources	资源文件夹	包含程序图标、背景图片等相关的图片资源。

7 界面展示

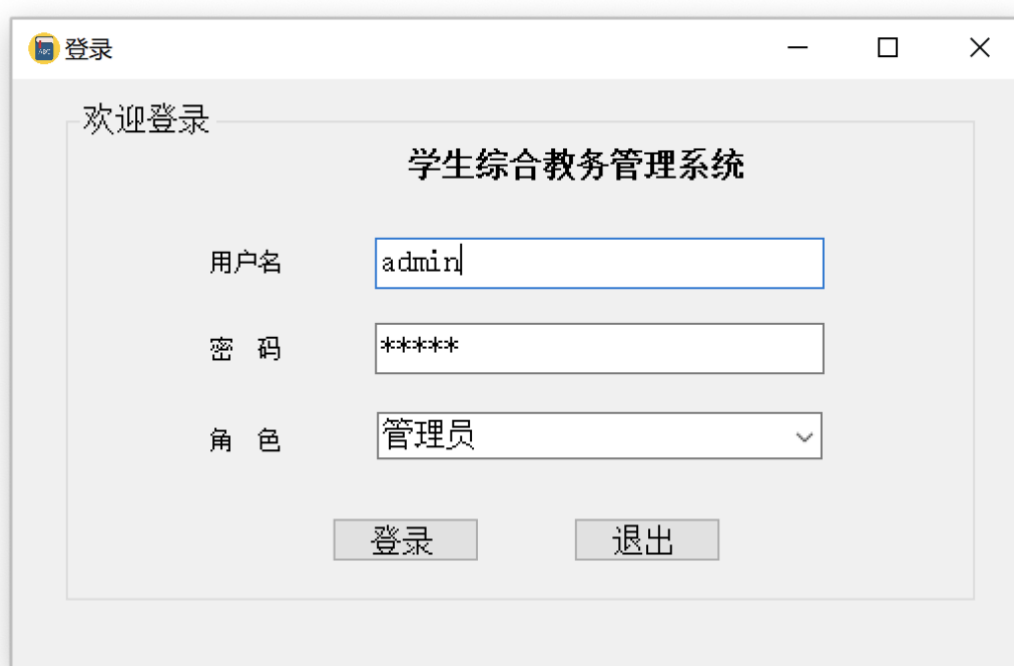


图 4 登录界面

7.1 管理员角色

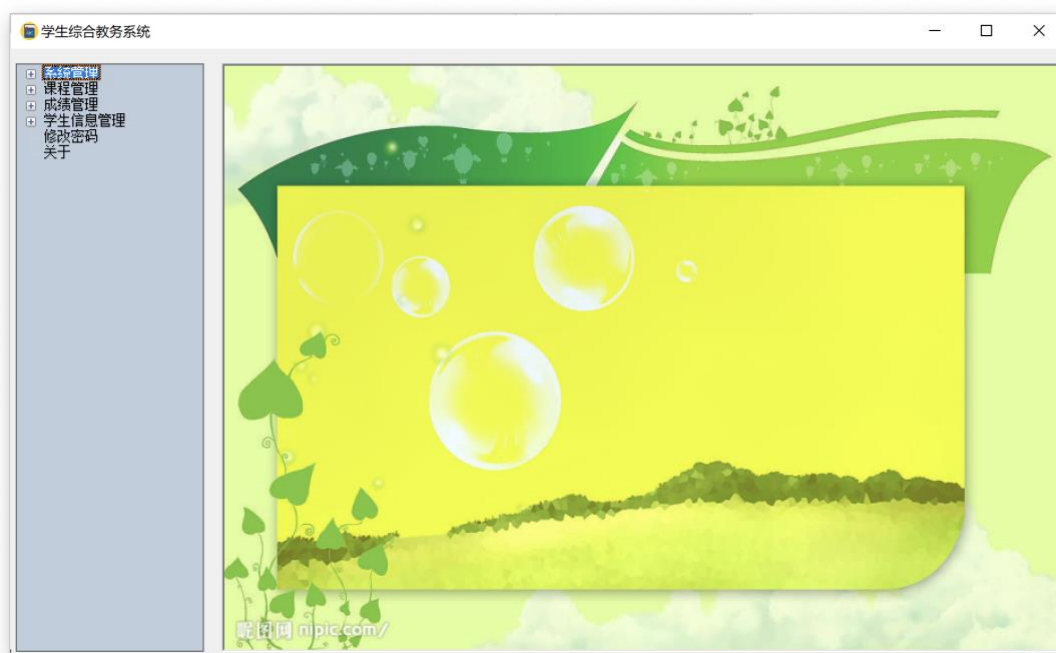


图 5 管理员主界面

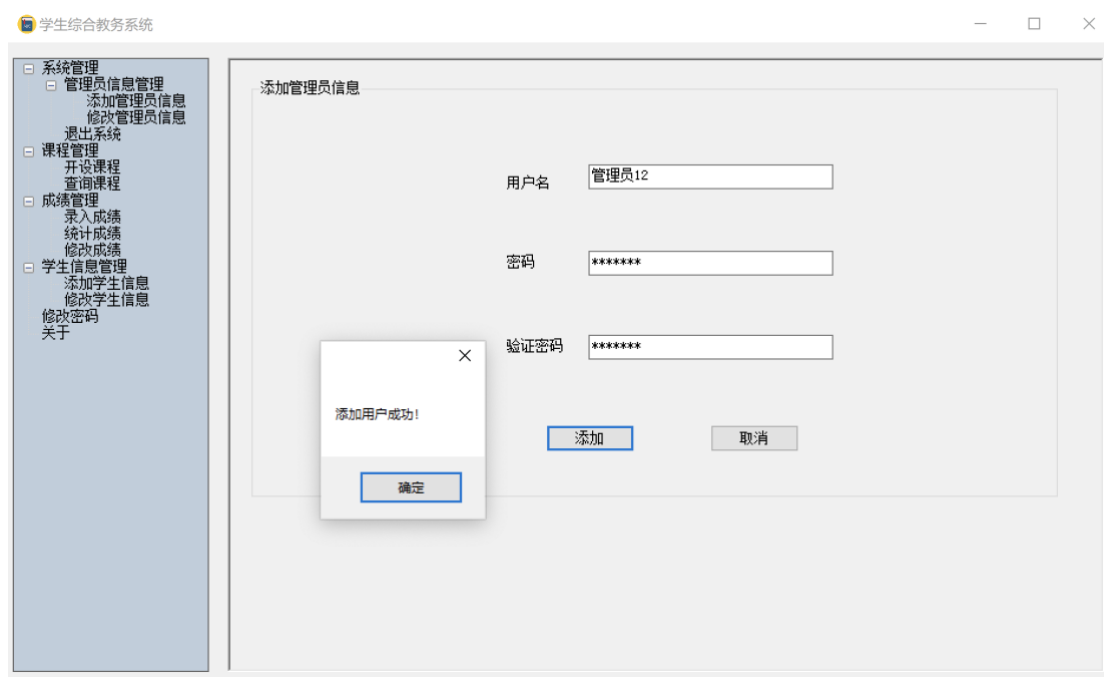


图 6 管理员——添加管理员界面

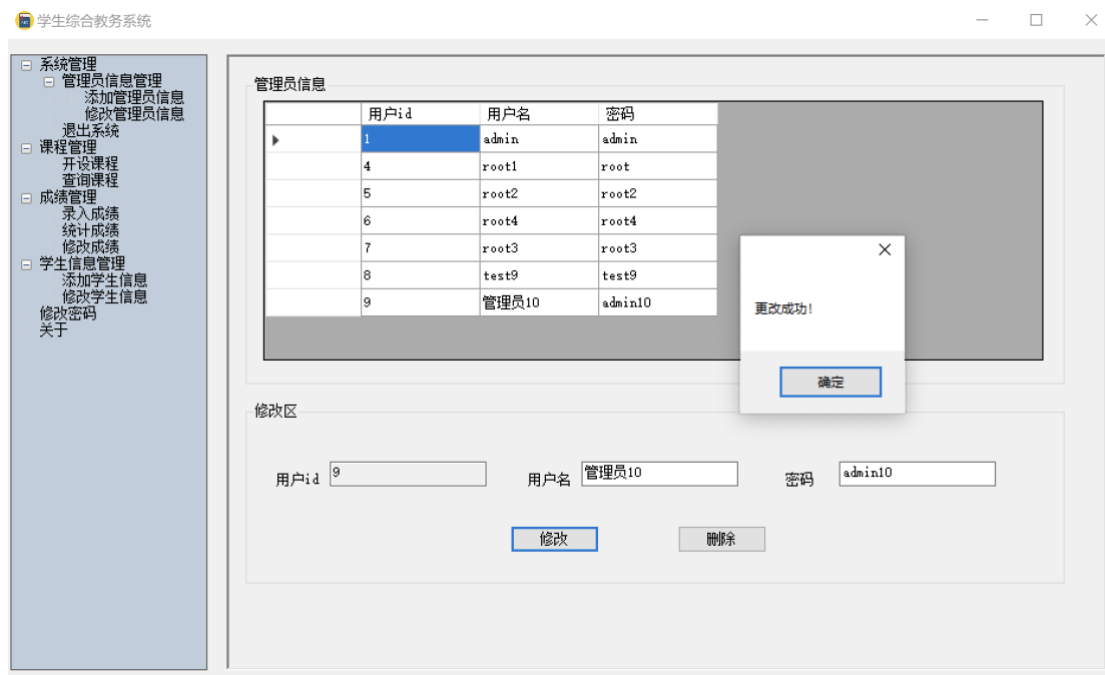


图 7 管理员——修改管理员界面

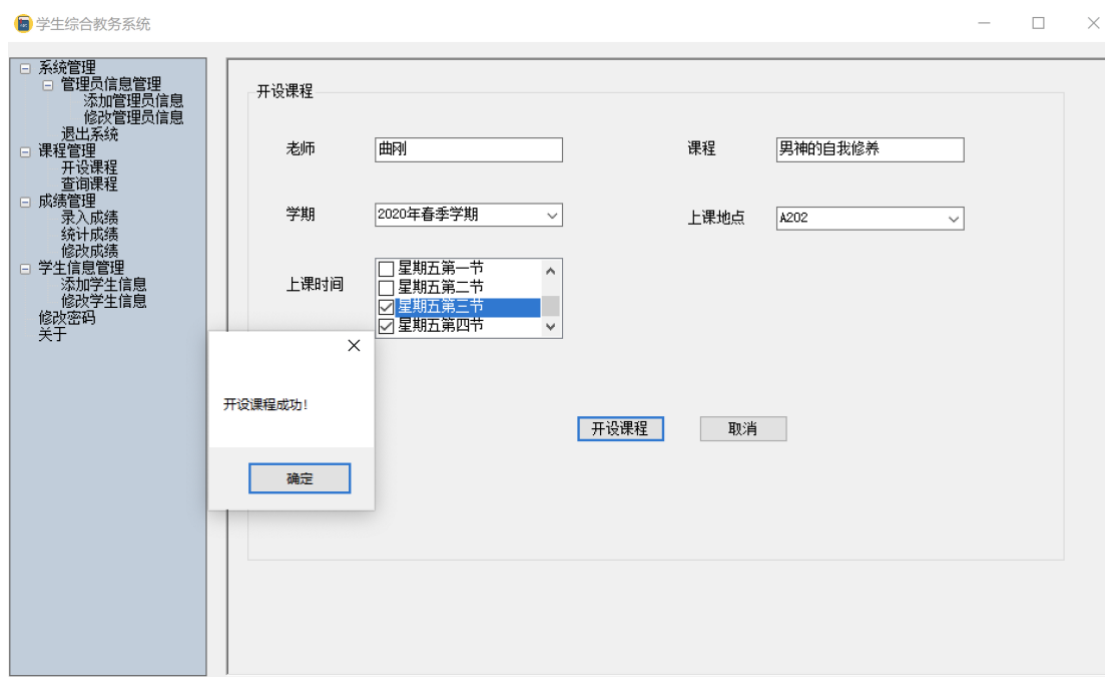


图 8 管理员——开设课程界面

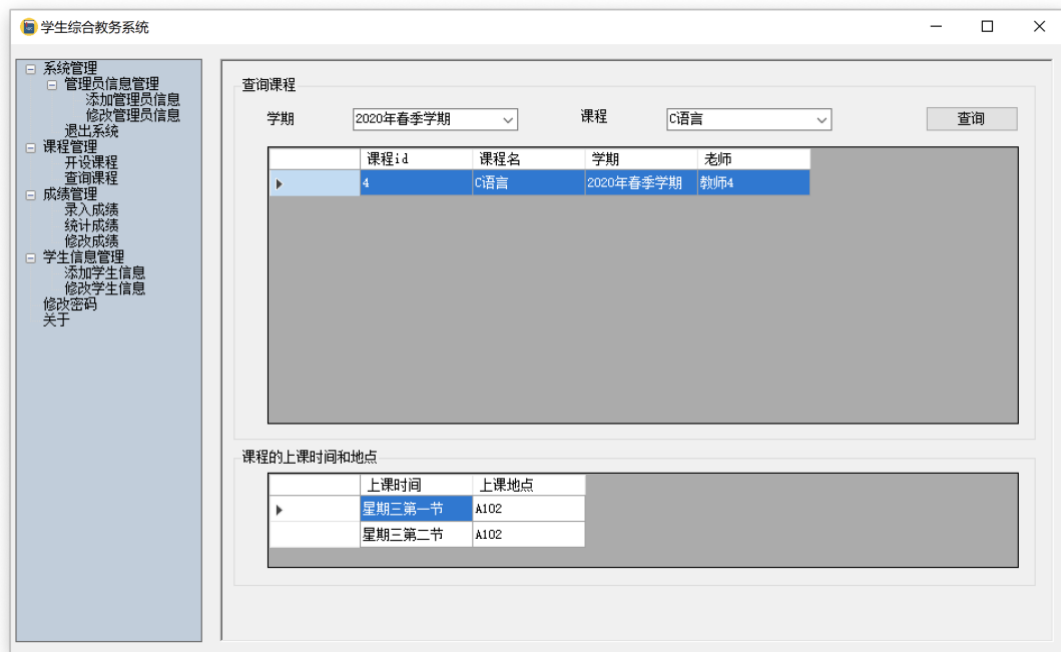


图 9 管理员——查询课程界面

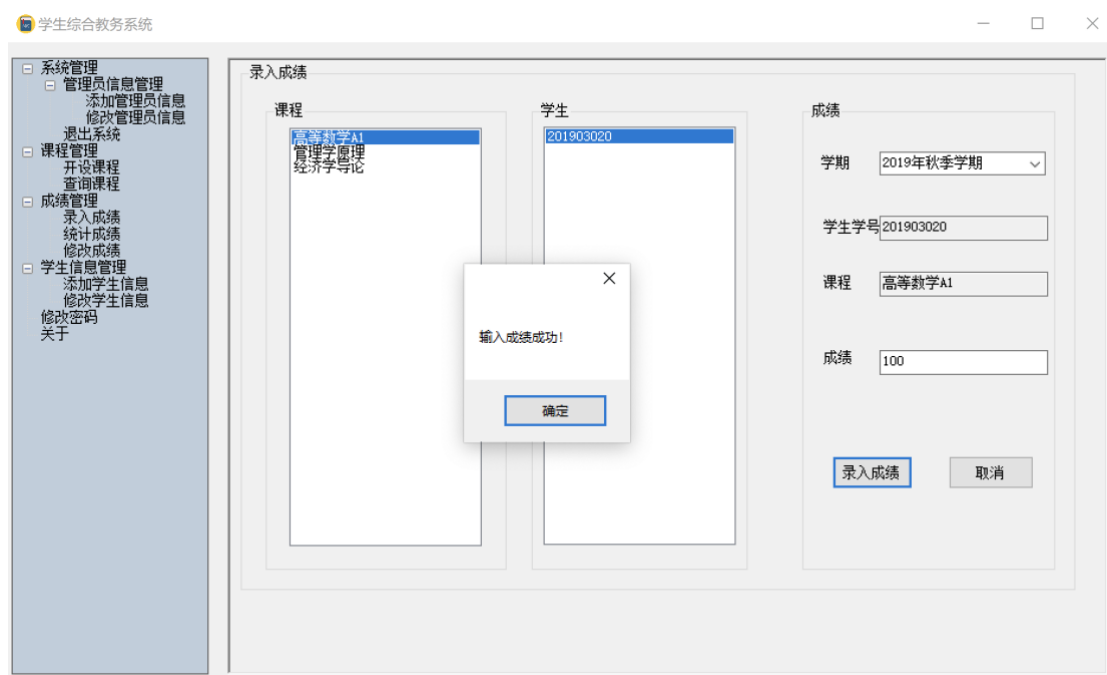


图 10 管理员——录入成绩界面

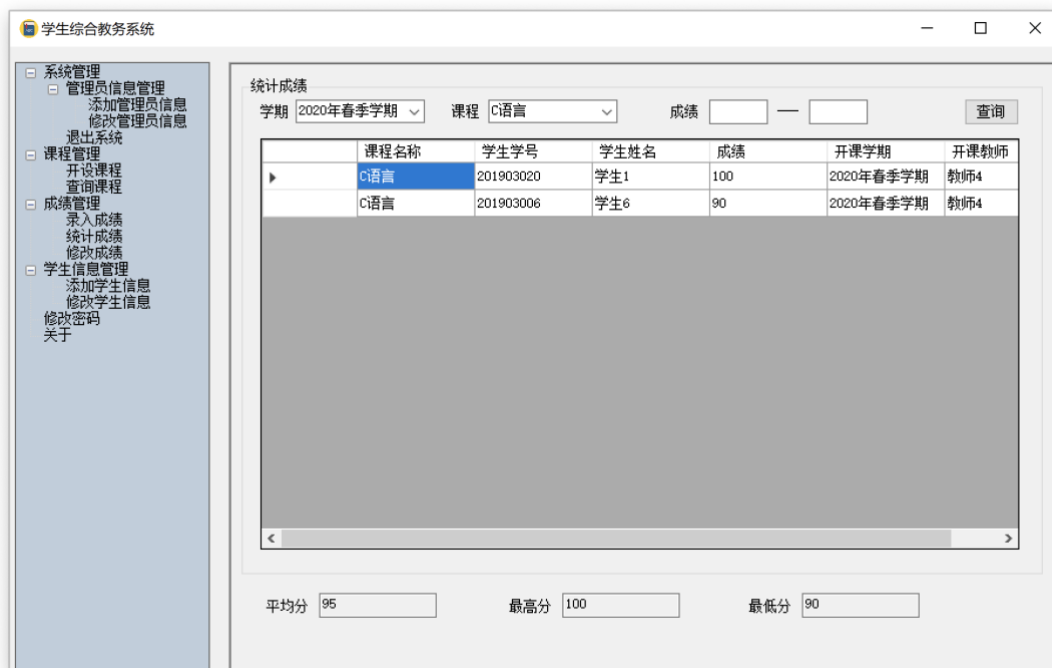


图 11 管理员——统计成绩页面

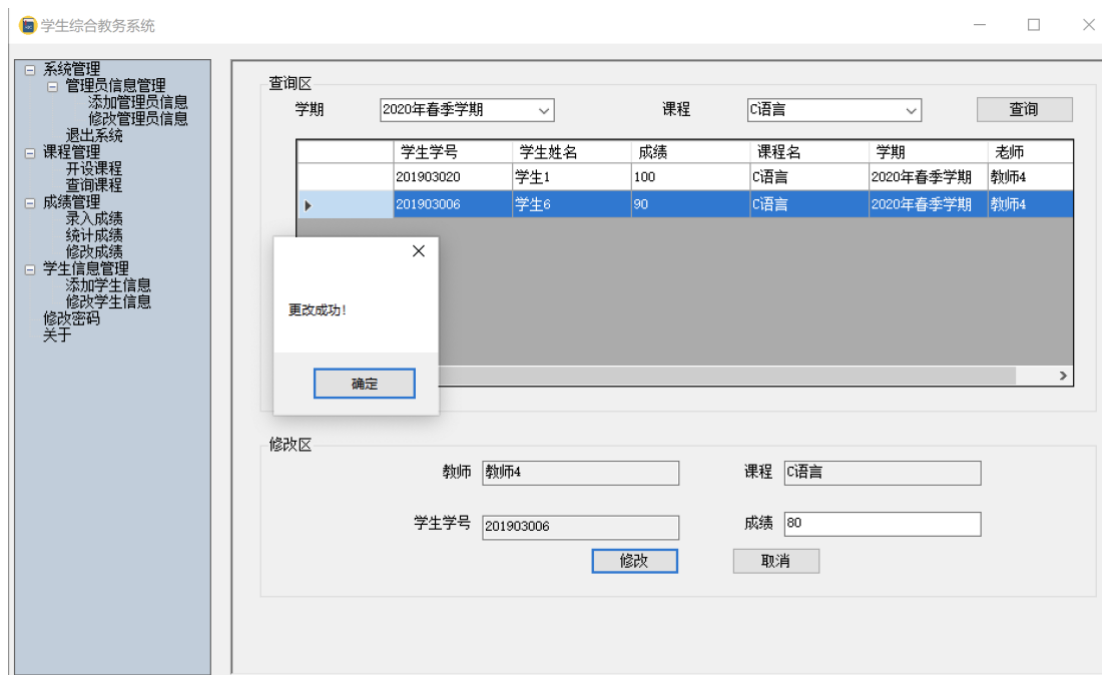


图 12 管理员——修改成绩界面

学生综合教务系统

添加学生信息

学生姓名

学号

密码

年级

专业

性别 ☐ 男 ☐ 女

出生日期

籍贯

添加 取消

图 13 管理员——添加学生信息界面

学生综合教务系统

学生信息

用户id	真实姓名	学号	密码	年级	专业
1	学生1	201903020	201903020	三年级	物流管理
2	学生2	201903002	201903002	三年级	信息管理与信..
3	学生3	201903003	201903003	三年级	大数据管理
5	学生5	201903005	201903005	三年级	金融学
6	学生6	201903006	201903006	三年级	财务管理

更改成功!

修改区

id

姓名

学号

密码

专业

出生日期

性别 ☐ 男 ☒ 女

籍贯

修改 删除

图 14 管理员——修改学生信息界面

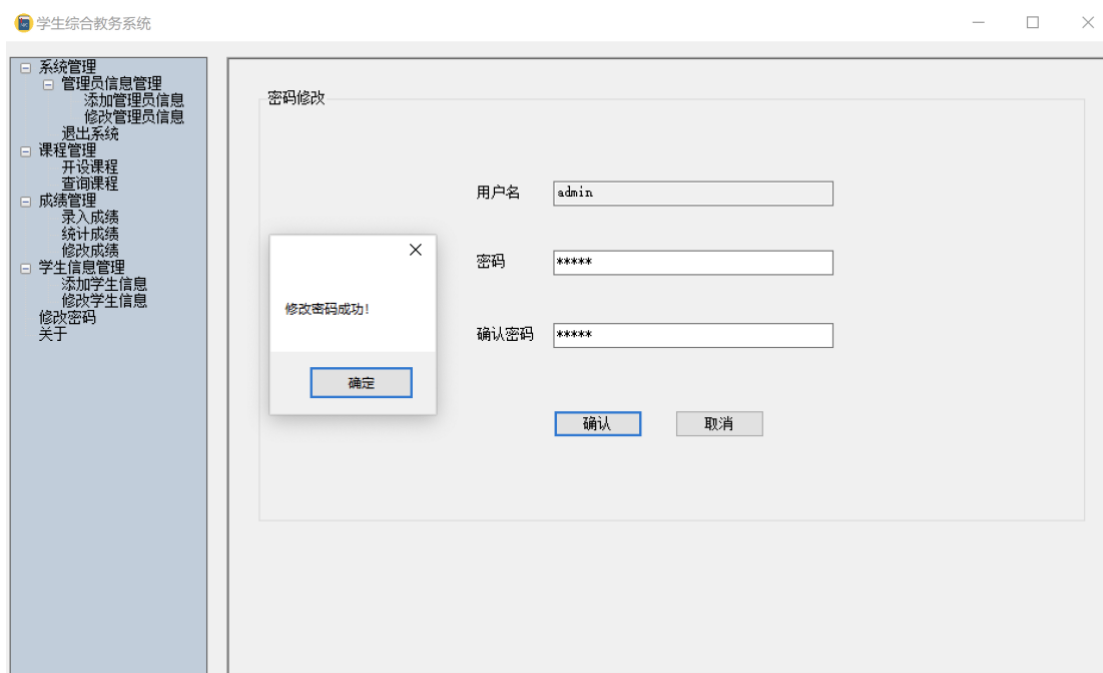


图 15 管理员——修改密码界面

7.2 学生界面

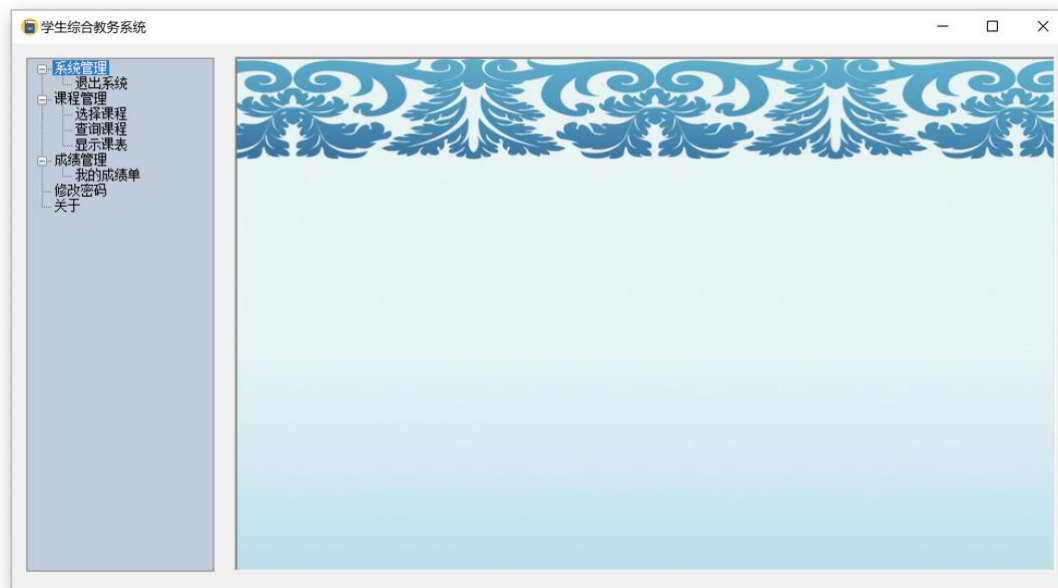


图 16 学生登录主界面



图 17 学生——选择课程界面



图 18 学生——查询课程界面

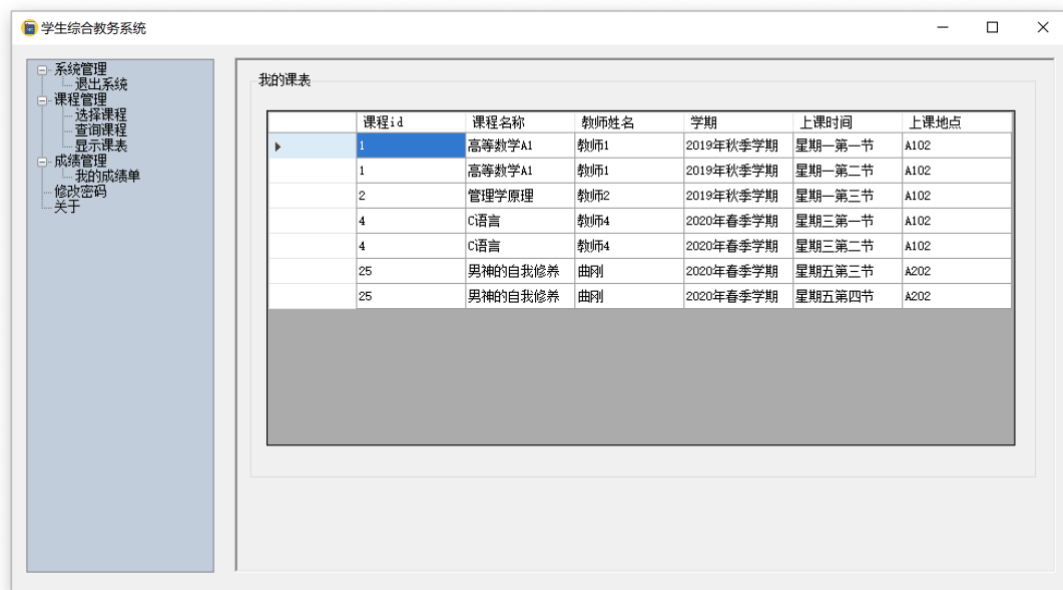


图 19 学生——显示课表界面

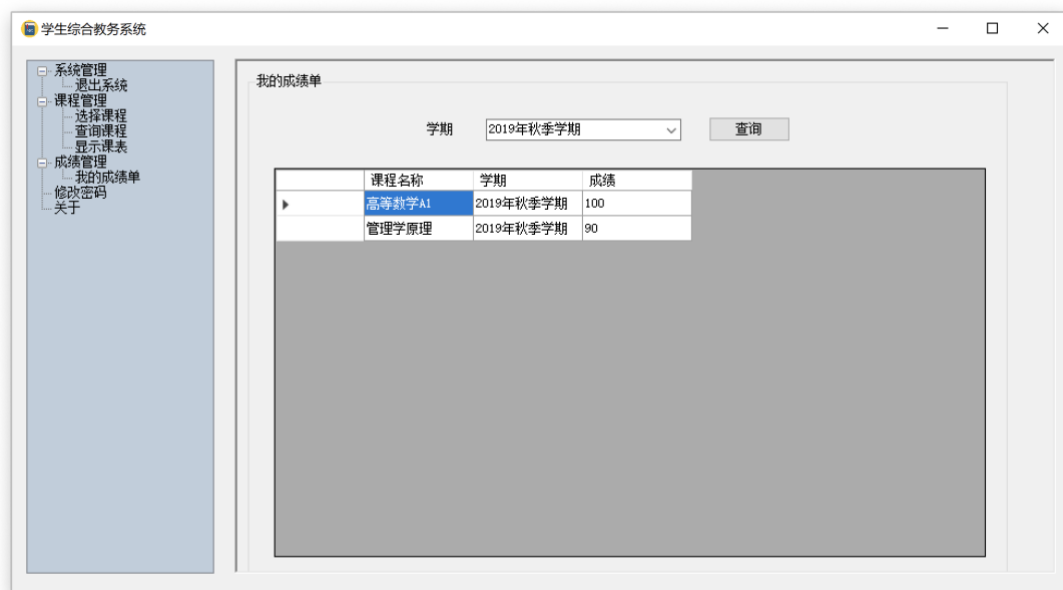
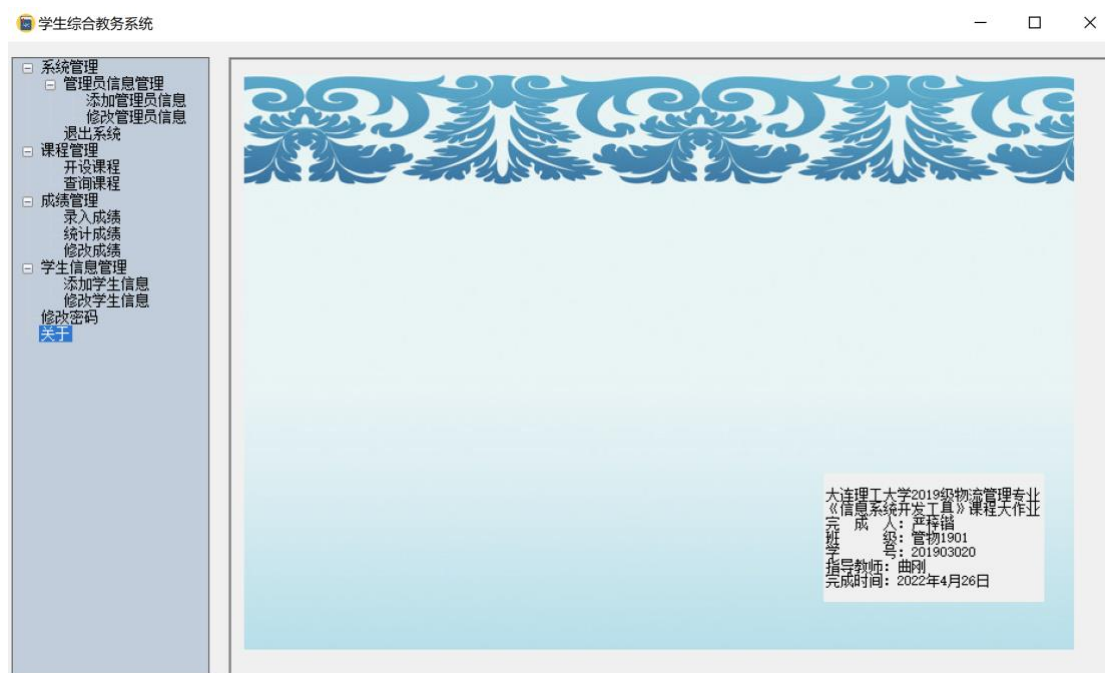


图 20 学生——我的成绩单



图 21 学生——密码修改界面

7.3 关于界面



8 关键技术介绍

8.1 左侧点击选项，右侧展开界面

使用 treeview 实现点击不同的选项，在右侧面板中显示不同的界面。具体代码如下：

```
1. private void treeView1_AfterSelect(object sender, TreeViewEventArgs
   e)
2.     {
3.         switch (this.treeView1.SelectedNode.Text)
4.         {
5.             case "录入成绩":
6.                 Form2 f = new Form2();
7.                 f.TopLevel = false;
8.                 f.FormBorderStyle = FormBorderStyle.None;
9.                 f.WindowState = FormWindowState.Maximized;
10.                panel2.Controls.Add(f);
11.                f.Show();
12.                break;
13.            case "我的成绩单":
14.                searchgradeForm f1 = new searchgradeForm();
15.                f1.TopLevel = false;
16.                f1.FormBorderStyle = FormBorderStyle.None;
17.                f1.WindowState = FormWindowState.Maximized;
18.
19.                panel2.Controls.Add(f1);
20.                f1.Show();
21.                break;
22.            case "统计成绩":
23.                countForm f2 = new countForm();
24.                f2.TopLevel = false;
25.                f2.FormBorderStyle = FormBorderStyle.None;
26.                f2.WindowState = FormWindowState.Maximized;
27.
28.                panel2.Controls.Add(f2);
29.                f2.Show();
30.                break;
31.            case "开设课程":
32.                kaisheForm f3 = new kaisheForm();
33.                f3.TopLevel = false;
34.                f3.FormBorderStyle = FormBorderStyle.None;
35.                f3.WindowState = FormWindowState.Maximized;
```

```
36.
37.         panel2.Controls.Add(f3);
38.         f3.Show();
39.         break;
40.     case "选择课程":
41.         chooseForm f4 = new chooseForm();
42.         f4.TopLevel = false;
43.         f4.FormBorderStyle = FormBorderStyle.None;
44.         f4.WindowState = FormWindowState.Maximized;
45.
46.         panel2.Controls.Add(f4);
47.         f4.Show();
48.         break;
49.     case "查询课程":
50.         searchclassForm f5 = new searchclassForm();
51.         f5.TopLevel = false;
52.         f5.FormBorderStyle = FormBorderStyle.None;
53.         f5.WindowState = FormWindowState.Maximized;
54.
55.         panel2.Controls.Add(f5);
56.         f5.Show();
57.         break;
58.     case "添加管理员信息":
59.         addmanForm f6 = new addmanForm();
60.         f6.TopLevel = false;
61.         f6.FormBorderStyle = FormBorderStyle.None;
62.         f6.WindowState = FormWindowState.Maximized;
63.
64.         panel2.Controls.Add(f6);
65.         f6.Show();
66.         break;
67.     case "修改管理员信息":
68.         modifymanForm f7 = new modifymanForm();
69.         f7.TopLevel = false;
70.         f7.FormBorderStyle = FormBorderStyle.None;
71.         f7.WindowState = FormWindowState.Maximized;
72.
73.         panel2.Controls.Add(f7);
74.         f7.Show();
75.         break;
76.     case "添加教师信息":
77.         addteacForm f8 = new addteacForm();
78.         f8.TopLevel = false;
79.         f8.FormBorderStyle = FormBorderStyle.None;
```



```
80.             f8.WindowState = FormWindowState.Maximized;
81.
82.             panel2.Controls.Add(f8);
83.             f8.Show();
84.             break;
85.         case "修改教师信息":
86.             modifyteacForm f9 = new modifyteacForm();
87.             f9.TopLevel = false;
88.             f9.FormBorderStyle = FormBorderStyle.None;
89.             f9.WindowState = FormWindowState.Maximized;
90.
91.             panel2.Controls.Add(f9);
92.             f9.Show();
93.             break;
94.         case "添加学生信息":
95.             addstuForm f10 = new addstuForm();
96.             f10.TopLevel = false;
97.             f10.FormBorderStyle = FormBorderStyle.None;
98.             f10.WindowState = FormWindowState.Maximized;
99.
100.            panel2.Controls.Add(f10);
101.            f10.Show();
102.            break;
103.        case "修改学生信息":
104.            modifystuForm f11 = new modifystuForm();
105.            f11.TopLevel = false;
106.            f11.FormBorderStyle = FormBorderStyle.None;
107.            f11.WindowState = FormWindowState.Maximized;
108.
109.            panel2.Controls.Add(f11);
110.            f11.Show();
111.            break;
112.        case "关于":
113.            aboutForm f12 = new aboutForm();
114.            f12.TopLevel = false;
115.            f12.FormBorderStyle = FormBorderStyle.None;
116.            f12.WindowState = FormWindowState.Maximized;
117.
118.            panel2.Controls.Add(f12);
119.            f12.Show();
120.            break;
121.        case "退出系统":
122.            Application.Exit();
123.            break;
```

```

124.         case "显示课表":
125.             showkebiaoForm f13 = new showkebiaoForm();
126.             f13.TopLevel = false;
127.             f13.FormBorderStyle = FormBorderStyle.None;
128.             f13.WindowState = FormWindowState.Maximized;
129.             panel2.Controls.Add(f13);
130.             f13.Show();
131.             break;
132.
133.         case "修改密码":
134.             modifymimaForm f14 = new modifymimaForm();
135.             f14.TopLevel = false;
136.             f14.FormBorderStyle = FormBorderStyle.None;
137.             f14.WindowState = FormWindowState.Maximized;
138.             panel2.Controls.Add(f14);
139.             f14.Show();
140.             break;
141.         case "修改成绩":
142.             modifygradeFram f15 = new modifygradeFram();
143.             f15.TopLevel = false;
144.             f15.FormBorderStyle = FormBorderStyle.None;
145.             f15.WindowState = FormWindowState.Maximized;
146.             panel2.Controls.Add(f15);
147.             f15.Show();
148.             break;
149.     }
150. }

```

8.2 在其他的窗口得到在登录窗口输入的信息

实现方法：在 loginFram 中设置两个方法，在其他类中只需要引用 loginFram 的这些方法即可。具体实现如下：

```

1. public static String getStudent()
2.     {
3.         String stuxuehao = "";
4.         stuxuehao = loginForm.name;
5.         return stuxuehao;
6.     }
7.
8. public static String getRole()
9.     {
10.        String role1 = "";

```

```

11.         role1 = role;
12.         return role1;
13.     }

```

8.3 新开设课程时教室使用冲突处理

管理员开设课程时，对于同一时间该教室有课的情况怎么做出异常处理。具体实现如下：

```

1. private void button1_Click(object sender, EventArgs e)
2.     {
3.         string teacher = textBoxteacher.Text;
4.         string classes = textBoxclass.Text;
5.         string term = comboBoxterm.SelectedItem.ToString();
6.         string flags = "1";
7.         SqlConnection conn = new SqlConnection(loginForm.connectionString);
8.         conn.Open();
9.         SqlCommand cmd = new SqlCommand();
10.        cmd.Connection = conn;
11.        //将开课信息插入到开课表里
12.
13.        //MessageBox.Show("开设课程成功!");
14.        string sql = "";
15.        //得到上课的地点
16.        string didian = comboBoxdidian.SelectedItem.ToString();
17.        //checkedListBoxtime
18.
19.        for (int i = 0; i < checkedListBoxtime.Items.Count; i++)
20.        {
21.            if (checkedListBoxtime.GetItemChecked(i))
22.            {
23.                String time = checkedListBoxtime.GetItemText(checkedListBoxtime.Items[i]);
24.                sql = "select * from sctime where sctime = '" + time + "' and location = '" + didian + "'";
25.                SqlDataAdapter adp = new SqlDataAdapter(sql, conn);
26.                DataSet ds = new DataSet();
27.                adp.Fill(ds);
28.                if (ds.Tables[0].Rows.Count > 0)
29.                {
30.                    flags = "2";

```

```
31.             MessageBox.Show("该时间该教室已经有课!");
32.             break;
33.         }
34.         else
35.         {
36.             flags = "1";
37.             break;
38.         }
39.     }
40. }
41.
42.     if (flags == "1")
43.     {
44.         sql = "insert into class (claname,term,teacher) values ('" + classes + "','" + term + "','" + teacher + "')";
45.         cmd.CommandText = sql;
46.         cmd.ExecuteNonQuery();
47.
48.
49.
50.         for (int i = 0; i < checkedListBoxtime.Items.Count;
            i++)
51.         {
52.             if (checkedListBoxtime.GetItemChecked(i))
53.             {
54.                 string time = checkedListBoxtime.GetItemText(checkedListBoxtime.Items[i]);
55.                 //将开课表的id 得到
56.                 sql = "select claid from class where claname = '" + classes + "' and term = '" + term + "' and teacher = '" + teacher + "'";
57.                 cmd.CommandText = sql;
58.                 String id1 = cmd.ExecuteScalar().ToString();
59.                 int id = 0;
60.                 int.TryParse(id1, out id);
61.                 sql = "insert into sctime values(" + id + ", '" + time + "','" + didian + "')";
62.                 cmd.CommandText = sql;
63.                 cmd.ExecuteNonQuery();
64.             }
65.         }
66.         MessageBox.Show("开设课程成功!");
67.     }
68.
```

```
69.         conn.Close();
70.     }
```

8.4 学生选课时间冲突处理

学生选择课程时,对选择一门课程与已选择的课程出现上课时间冲突的情况的异常处理。具体实现如下:

```
1. private void button1_Click(object sender, EventArgs e)
2.     {
3.         Boolean flag = true;
4.
5.         //判断必要空格内容是否为空
6.         if (comboBox1.Text == "")
7.         {
8.             MessageBox.Show("请选择学期!");
9.         }
10.        else if (textBoxid.Text == "" || textBoxclass.Text == ""
11.        )
12.        {
13.            MessageBox.Show("请选择课程!");
14.        }
15.        else
16.        {
17.            //得到课程的id
18.            int claid = 0;
19.            int.TryParse(textBoxid.Text, out claid);
20.
21.            //得到课程的学期
22.            string semster = comboBox1.SelectedItem.ToString();
23.
24.            //查询是否已选该课程
25.            SqlConnection conn = new SqlConnection(loginForm.con
26.            nectionString);
27.            string sql = "select * from sc where stuid = " + stu
28.            id + " and claid = " + claid;
29.            conn.Open();
30.            SqlCommand cmd = new SqlCommand(sql, conn);
31.            SqlDataReader myreader = cmd.ExecuteReader();
32.            if (myreader.Read())
33.            {
34.                flag = false;
35.            }
36.        }
37.    }
```

```
32.             MessageBox.Show("已选过该课程，请勿重复选课！");
33.         }
34.         myreader.Close();
35.
36.         // 查询在该时间是否有课
37.         if (flag)
38.         {
39.             sql = "select sctime from sctime where claid ="
+ claid;
40.             SqlDataAdapter adp = new SqlDataAdapter(sql, conn);
41.             DataSet ds = new DataSet();
42.             adp.Fill(ds);
43.             for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
44.             {
45.                 DataRow dr = ds.Tables[0].Rows[i];
46.                 string time = dr[0].ToString(); // 第一列
47.                 sql = "select * from sc,sctime,class where class.claid = sc.claid and class.claid = sctime.claid and sctime = '"
+ time + "' and sc.stuid = " + stuid + " and class.term = '" + semester + "'";
48.                 SqlDataAdapter adp1 = new SqlDataAdapter(sql, conn);
49.                 DataSet ds1 = new DataSet();
50.                 adp1.Fill(ds1);
51.                 if (ds1.Tables[0].Rows.Count > 0)
52.                 {
53.                     flag = false;
54.                     MessageBox.Show("课程上课时间冲突！");
55.                     break;
56.                 }
57.             }
58.         }
59.
60.         // 进行选课操作
61.         if (flag)
62.         {
63.             sql = "insert into sc(claid,stuid) values(" + claid + "," + stuid + ")";
64.             cmd.CommandText = sql;
65.             if (cmd.ExecuteNonQuery() > 0)
66.             {
67.                 MessageBox.Show("选课成功！");
```

```
68.         }
69.     }
70.
71.         // 更新已选课程列表
72.         if (listBox1.Items.Count > 0)
73.         { // 清空所有项
74.             listBox1.Items.Clear();
75.         }
76.         sql = "select class.claname from sc,class where sc.
           claid = class.claid and stuid=" + stuid;
77.         SqlDataAdapter adp2 = new SqlDataAdapter(sql, conn);
78.         DataSet ds2 = new DataSet();
79.         adp2.Fill(ds2);
80.         foreach (DataRow row in ds2.Tables[0].Rows)
81.         {
82.             listBox1.Items.Add(row[0].ToString());
83.         }
84.         conn.Close();
85.     }
86. }
```

9 优化方案

9.1 提高访问数据库的速度

目前,能明显感觉到管理信息系统在处理需要与数据库系统进行交互的任务的时候,会存在一定的响应时间。虽然不足以影响使用,但若能进行优化,就能进一步提升用户的体验。

经过分析,产生上述情况的原因之一是在程序设计的过程中,优先考虑了模块化的思想,对于每一次操作都是独立进行数据库的连接和关闭,而非一次连接一直使用。该思想使得程序吗模块之间的独立性较强,可移植性较好,最重要的是减少了运行时的内存,对于数据库中的数据做到随用随取。但是由于频繁的连接和断开使得每次进行相应操作时都会有一定的延时。

在不改变原本代码逻辑的前提下,可以进行改进方法有:引入视图、触发器、函数、存储过程,优化表结构,建立索引。

9.2 引入教师角色及相应功能界面

本程序项目主要聚焦于“学生”以及“教务管理人员”两类角色，创建了以服务学生为导向的管理新系统。主要满足学生选课、查看成绩、查看课表以及教务管理人员对于学生各类信息的录入和修改等功能。

然而，部分功能，例如学生成绩的输入和修改等，需要由教务人员进行输入。这符合一部分的应用场景（例如大连理工大学的教务管理上就是按照这个流程，由授课教师向教务处提交成绩信息后由教务管理人员对学生的成绩信息进行统一的录入）；但对于另一类的应用场景和需求，例如部分高校允许任课教师直接在系统中上传和修改学生的成绩信息，该管理信息系统就无法满足教师这一部分群体的要求。

受时间等条件的限制，本项目仅针对上述第一类情况进行分析设计和系统的开发，对于第二类情况，本项目也在程序结构中留出了教师相关界面（包括教师查看自己的任课情况等），结课后计划自行摸索进一步开发。

同时，为实现上述目的，需要对数据库表的结构进行修改，包括新增教师表等。

10 完成情况总结

通过一段时间的努力，从前期的需求分析与架构设计到实际的编码阶段，我完成了学生选课及成绩管理系统的开发与设计。已经完成了所有的功能，但还有一些亟待完善的方面，例如对数据的存储采用加密的措施，UI 设计等，这些问题在结课后我会继续进行学习研究。

为了使得程序应用界面的结构性更强，用户能够一眼看懂所有的功能，我引入了 `treeview`，以逻辑树状的结构较好的连接了不同的窗体，做到窗体之间的快速切换。

感受较深的是对程序健壮性的完善。设计好的程序在实际运行过程中不断会出现一些在非常规操作顺序下的报错，而对于这类错误的考虑和捕捉是必要的，因为用户并不会严格按照程序设计者所预想的逻辑去使用程序。并且随着程序设计窗体的不断增加，每个窗体的功能的差异性增大，所需要考虑的各类可能出现的不健壮性问题也逐渐增多。提升程序健壮性的过程很考验耐心和细心，这也许是成为优秀的程序设计开发者所必备的能力之一。

此外，虽然所学是物流管理，但一直以来对计算机技术感兴趣。上学期选修了《移动 APP 开发与设计》，基于 Java 和 Android Studio 平台进行开发。和本

学期所学的 C#以及 Visual Studio 平台进行对比后，能明显感觉到其中面向思想的一致性，但同时也感受到微软框架下为了效率牺牲多样性的妥协。处于反垄断和国家战略的考虑，越来越多的国家的企业在开发平台的选择上从微软转向了 Java 等，个人认为这也一定是大势所趋。因此，感谢曲老师在本门课程中为我们打下的面向对象程序设计思想的基础，在未来我也会进一步学习 Java 等更加与时俱进的语言。

11 使用说明

11.1 数据库数据导入

11.1.1 方式一：运行数据库脚本后手动添加数据

运行5 数据库创建脚本后，手动添加相关数据表的数据。数据库 sql 脚本也已附在项目文件夹下的“数据库”文件夹内。

11.1.2 方式二（推荐）：导入 mdf 及 ldf 文件

打开 SQL Server 后，新建查询，输入以下语句。

```
1. EXEC sp_attach_db @dbname = 'StudentMIS',
2. @filename1 = 'C:\Projects\CSharpProjects\information-management-
system-of-students\数据库\student.mdf',
3. @filename2 = 'C:\Projects\CSharpProjects\information-management-
system-of-students\数据库\student_log.ldf'
```

注意点：

- i. 该命令会新建名为“StudentMIS”的数据库，运行前需要保证原有的数据库和该需要新建的数据库不重名。
- ii. @filename1 和 @filename2 分别为 mdf 文件和 ldf 文件的路径。这两个文件已放在项目文件夹下的“数据库”文件中。
- iii. 若提示“无法打开物理文件 XXX.mdf”。操作系统错误 5:“5(拒绝访问。)”则需要打开 mdf 和 ldf 的安全权限。
 - a) 解决方案一：数据库使用 windows 验证，将被附加的数据库移植到根目录下，即 MSSQL 的 DATA 目录下。
 - b) 解决方案二：windows 验证，然后将要附加的.mdf 文件和.ldf 文件更改权限，选中.mdf 文件---->右击----->安全-----> Authenticated

Users---->编辑---->安全控制允许,按照同样的方法,更改日志文件,然后附加数据库既可。

11.2 数据库连接路径修改

由于不同电脑在做完上述操作,即创建完数据库并导入数据后,访问数据库的用户 id 和密码不同,甚至数据库名字和数据库管理系统的类型也可能不同。因此,需要对程序的 C#代码块进行部分的修改,以适配本机的相关配置信息。

使用 Visual Studio 打开项目文件后,找到 loginForm.cs 文件,点开该文件,进入代码编辑模式,将如下代码块进行修改。

```
public static string connectionString = "uid=sa;pwd=123456;initial cata  
log=StudentMIS;Data Source=localhost;Connect Timeout=900";
```

其中 uid, pwd, initial catalog, Data Source 分别对应本机 SQL Server 的登录名、密码、数据库名、服务器名称。

11.3 提供的供登陆查看的用户名和密码

在未改动数据库数据的情况下,可以用如下的登录名和密码进行登录。

- i. 用户名: admin, 密码: admin, 角色: 管理员
- ii. 用户名: 201903020, 密码: 201903020, 角色: 学生