



**C++面向对象程序设计**

**课程报告**

|  |
| --- |
| **基于C++的学生信息管理系统设计** |
|  |

**一 系统功能需求**

**// (系统默认登陆用户名密码为{莫世冰、123456}{201934410316 、123456})**

**1.系统基本要求与个人优化.**

**（1）实现账号登陆界面**。（例如：使用用户名和密码来进入登陆界面。如果输入的用户名和密码有一项为空或者是输入的用户名和密码是错误的，则不能进入系统）

**专选优化**：

1. 密码未输入时，提示“密码不能为空”；
2. 用户名未输入时，提示“用户名不能为空”；
3. 密码错误，提示“用户名或者密码错误”；
4. 用户名可以是学生姓名，也可以是学生个人学号，且实现与密码对应。

**（2）实现任意添加学生的信息**。（例如：添加学生的个人学号，姓名，性别，出生日期，专业，年级。添加记录时，要求学号要唯一。输入的时候不允许为空，并且有相应的提示功能；如果输入的时候有一项为空则提示输入相关得到消息。如果添加了重复的学号，则提示数据添加重复。）

**专项优化**：

1. 不仅仅可以在学生消息管理系统的主页面添加学生对应信息，也可以实现直接从文件库(Database.txt)批量添加。

**（3）实现删除学生的信息**。（例如：实现对已添加的学生信息进行删除。如果当前系统中没有相应的记录，则提示“记录为空！”并返回操作。）

**专项优化**：

1. 可以实现批量删除；
2. 实现多功能化，不仅仅可以在学生信息管理系统的主页面“勾选+删除键”进行删除；也可以直接按“删除键”，进入查询窗口，进行学生信息（姓名或者个人学号）查询进行精准删除。并且删除方式系统实现自动选择，即同一个按键——两个功能。

**（4）实现查询学生的信息**。（例如：可根据学号或姓名等信息对已添加的学生信息进行查询，如果未找到，给出相应的提示信息，如果找到，则显示相应的记录信息）。

**（5）实现编辑学生的信息**。可根据查询结果对相应的记录进行修改。（例如：修改时注意学号的唯一性，包括编辑学生的个人学号，姓名，性别，专业，年级，出生日期。输入不允许为空，并且有提示功能，如果输入的时候有一项为空则提示输入相关的信息。）

**专项优化**：

1. 可以实现勾选多个单项框，以及结合“全选键”与“反选键”进行操作的组合，实现对多个学生信息的编辑、修改。
2. 增加系统用户交互功能，实现“双击学生信息管理系统页面的学生信息即可以进行修改”。

**（6）实现学生的信息保存功能。**（例如：可将学生的信息存入文件中，存入方式任意。保存学生信息包括个人学号，姓名，性别，专业，放回登陆界面）。

**专项优化**：

1. 建立文件筛选格式，可以实现将学习的信息存储为（\*.txt、\*.dat、\*.\*）的文件格式。（一般存储为”.dat”格式的文件，因为该格式文件默认打不开，可以防止打开文件进行误操作而导致学生信息丢失等情况！但是这里建立“.txt”格式的文件，方便开发，同时也方便研究、检查）。

**（7）实现注释功能**。（例如：注释当前用户后，返回登陆界面）。

**2. 系统优化功能的增加**

**（1）实现全选功能。**（例如：通过按“全选键”实现一次令学生管理系统中全部学生信息的勾选）。

**（2）实现反选功能。**（例如；勾选几个特定的不删除的学生信息，通过“反选键”，实现其它全部学生信息的勾选，再进行进一步的操作）。

**（3）实现用户交互。**（例如：直接双击学生信息管理系统，弹出编辑窗口，实现选择学生信息的编辑修改）。

**（以上的系统优化功能大大地方便了用户的操作，大大增强了系统的可操作性与实用性）**

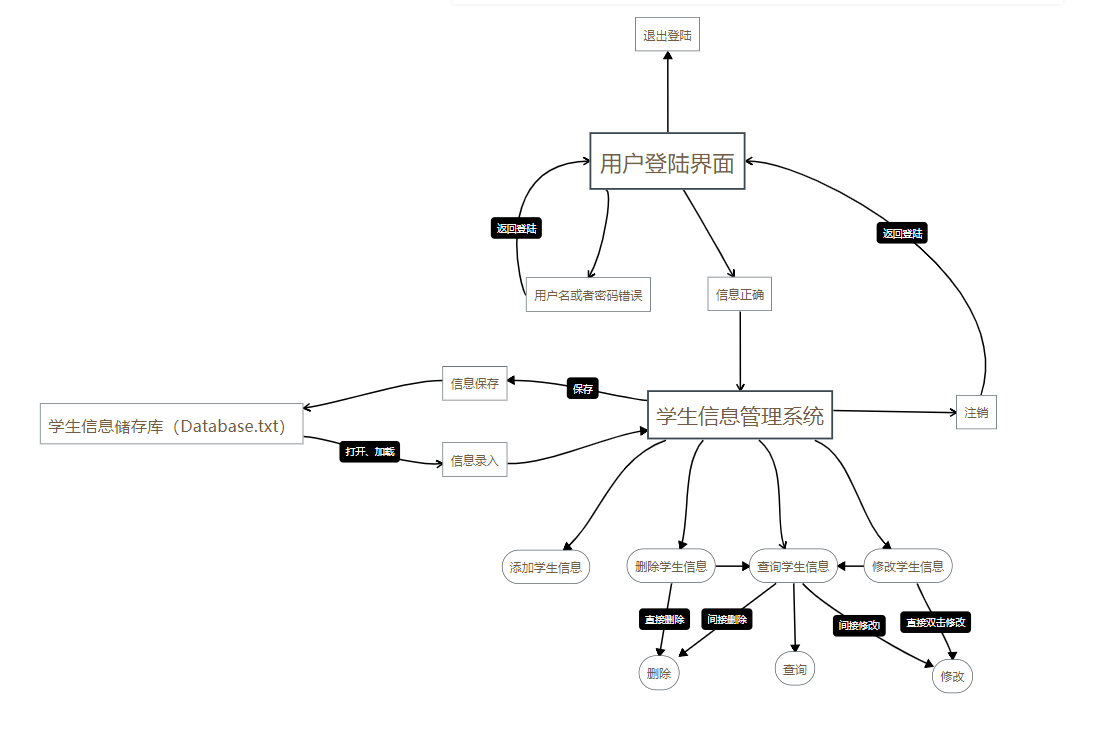
**二 系统功能分析与设计**

（1）本学生成绩管理系统（以下简称“信息系统”）在Microsoft Visual Studio 2017的开发环境中通过可视化编程的方式去实现。

（2）根据本系统的功能设计需求，大体可分为登陆板块，信息系统板块，文件的输入输出等三大部分，其中信息系统板块内还含有多个子窗口，窗口之间形成交互，不仅仅在实现设计功能需求的前提，还可以提高系统的操作性。

（3）需要掌握如何运用面向对象技术对具体的应用系统进行分析和设计。了解了MFC类的层次结构及主要类的用法，掌握使用MFC和向导来编写Windows应用程序。明白了控件是Windows图形用户界面的主要组成部分之一，掌握静态控件、按钮、流动条、编辑框、列表框、组合框和对话框的创建、消息发送和控件控制的基本用法。

**三 系统功能模块详细设计和流程图**



**四 系统功能模块设计主要代码**

1. **登陆界面**

MFCStudentManagerControlDlg.cpp中的主要关键代码：

1. **if** (isSuccess)
2. {
3. ShowWindow(SW\_HIDE); // 隐藏登陆窗口，但是需要在关闭系统主窗口的时候同时把子窗口也关闭，进入MangerDlg进行编译，增加WM-CLOSE信息和OnInitDialogc（）函数
4. // 创造一个对话框，以至于能够弹出系统主页面（注意要引入主页面“MangerDlg”的头文件）
5. MangerDlg Dlg;
6. Dlg.DoModal(); // 弹出主系统界面,利用DoModal（）函数
7. }
8. **else**
9. {
10. MessageBox(\_T("用户名或者密码错误"), \_T("登陆失败")); // 返回提示信息
11. }

在开发系统的过程中，可以将isSuccess赋值为True，实现“不用输入直接”登陆，可以大大方便开发。

应该注意的一点就是：登陆成功后需要弹出消息系统主页面（MangerDlg窗口）之后，需要关闭原来的登陆界面，但不能单单地隐藏，还需在MangerDlg使用指针实现窗口的自我摧毁，以防止下次再次执行程序时产生冲突，具体代码如下：

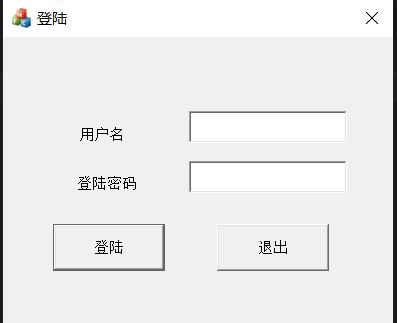
1. // 当子窗口关闭时执行的相关操作（实现当我们关闭系统主页面的时候关闭登陆页面）
2. **void** MangerDlg::OnClose() // 创建一个信息，即当窗口关闭的时候执行的一个函数
3. {
4. // TODO: 在此添加消息处理程序代码和/或调用默认值
5. CDialog\* dplg = (CDialog\*)AfxGetMainWnd(); // 获得子窗口的指针
6. dplg->DestroyWindow(); // 利用指针是操作子窗口（自我销毁）
7. CDialogEx::OnClose();
8. }
9. **学生信息管理系统主页面**

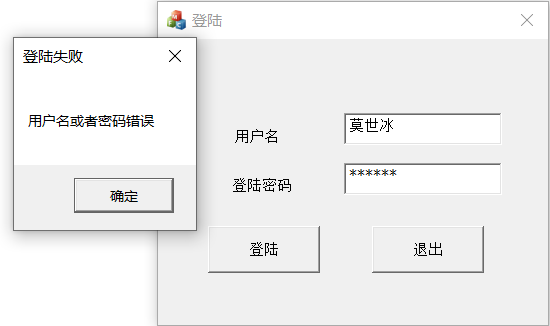
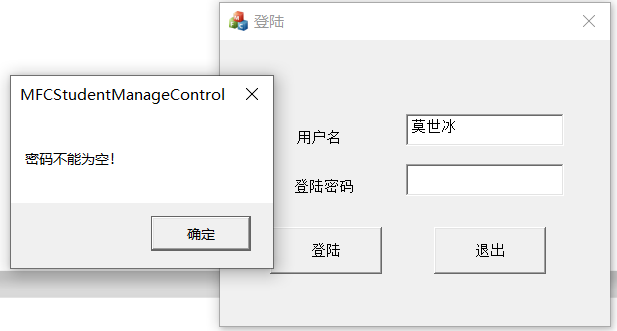
MangerDlg.cpp中的主要关键类与对象：

1. BBEGIN\_MESSAGE\_MAP(MangerDlg, CDialogEx)
2. ON\_WM\_CLOSE()
3. ON\_BN\_CLICKED(IDC\_BUTTON\_ALLSele, &MangerDlg::OnBnClickedButtonAllsele)
4. ON\_BN\_CLICKED(IDC\_BUTTON\_REVEALSele, &MangerDlg::OnBnClickedButtonRevealsele)
5. ON\_BN\_CLICKED(IDC\_BUTTON\_ADDM, &MangerDlg::OnBnClickedButtonAddm)
6. ON\_BN\_CLICKED(IDC\_BUTTON\_DELETEM, &MangerDlg::OnBnClickedButtonDeletem)
7. ON\_BN\_CLICKED(IDC\_BUTTON\_SEEKM, &MangerDlg::OnBnClickedButtonSeekm)
8. ON\_BN\_CLICKED(IDC\_BUTTON\_ATERM, &MangerDlg::OnBnClickedButtonAterm)
9. ON\_BN\_CLICKED(IDC\_BUTTON\_OPENM, &MangerDlg::OnBnClickedButtonOpenm)
10. ON\_BN\_CLICKED(IDC\_BUTTON\_SAVEM, &MangerDlg::OnBnClickedButtonSavem)
11. ON\_BN\_CLICKED(IDC\_BUTTON\_END, &MangerDlg::OnBnClickedButtonEnd)
12. ON\_NOTIFY(NM\_DBLCLK, IDC\_LIST1, &MangerDlg::OnNMDblclkList1)
13. END\_MESSAGE\_MAP()
14. 添加功能主要代码：
15. ADD\_SEEKDlg dlg; // 创建对话框
16. **if** (IDCANCEL == dlg.DoModal()) // 弹出对话框
17. {
18. **return**;
19. }
20. **int** nCount = SM\_list.GetItemCount();
22. **if** (dlg.sClass.IsEmpty())
23. {
24. AfxMessageBox("年级不能为空!!!");
25. **return**;
26. }
27. **if** (dlg.sMajor.IsEmpty())
28. {
29. AfxMessageBox("专业班级不能为空");
30. **return**;
31. }
32. **if** (dlg.sName.IsEmpty())
33. {
34. AfxMessageBox("名字不能为空");
35. **return**;
36. }
37. **if** (dlg.sID.IsEmpty())
38. {
39. AfxMessageBox("学号不能为空");
40. **return**;
41. }
42. **if** (dlg.sSex.IsEmpty())
43. {
44. AfxMessageBox("性别不能为空");
45. **return**;
46. }
47. **if** (dlg.sBirth.IsEmpty())
48. {
49. AfxMessageBox("出生日期不能为空");
50. **return**;
51. }
52. **for** (**int** i = 0; i < nCount; i++) // 判断学号是否唯一
53. {
54. **if** (dlg.sID == SM\_list.GetItemText(i, 3))
55. {
56. AfxMessageBox("学号重复，请输入正确的学号");
57. **return**;
58. }
59. }
60. SM\_list.InsertItem(nCount, dlg.sClass); // 编辑信息
61. SM\_list.SetItemText(nCount, 1, dlg.sMajor);
62. SM\_list.SetItemText(nCount, 2, dlg.sName);
63. SM\_list.SetItemText(nCount, 3, dlg.sID);
64. SM\_list.SetItemText(nCount, 4, dlg.sSex);
65. SM\_list.SetItemText(nCount, 5, dlg.sBirth);
66. 删除功能主要代码：
67. **if** (!flag)
68. {
69. **for** (**int** i = 0; i < SM\_list.GetItemCount(); i++) // 遍历每一项，判断是否被选中
70. {
71. **BOOL** state = SM\_list.GetCheck(i);
72. **if** (state)
73. {
74. SM\_list.DeleteItem(i); // 删除选定的项目
75. i--;  // 因为for循环的原因，会导致程序从1.3.5.7.9这样子去判断每一项的状态，因此让程序返回一项判断，实现每一个项目的一个一个判断
76. }
77. }
78. }
79. 查找功能主要代码：
80. **for** (**int** i = 0; i < nCount; i++) // 通过和文本比较，选定寻找信息
81. {
82. **if** ((SM\_list.GetItemText(i, 3) == dlg.fID || SM\_list.GetItemText(i, 2) == dlg.fName))
83. {
84. SM\_list.SetCheck(i);  // 避免了在用姓名查找时因为名字重复而导致漏选的情况
85. uNum = 0;
86. }
87. **if** (i == nCount - 1)
88. {
89. **if** (uNum)
90. {
91. AfxMessageBox("未找到相应记录！！！");
92. **return**;
93. }
94. }
95. }
96. 编辑功能主要代码：
97. **int** nCount = SM\_list.GetItemCount();
98. **for** (**int** i = 0; i < nCount; i++)
99. {
100. **if** (!SM\_list.GetCheck(i))
101. {
102. **if** (dlg.sID == SM\_list.GetItemText(i, 3))
103. {
104. AfxMessageBox("学号重复，请输入正确的学号");
105. **return**;
106. }
107. }
108. }
109. **添加学生信息窗口ADD\_SEEKDlg（亦学生编辑窗口）主要代码：**
110. GetDlgItemText(IDC\_EDIT\_CLASS, sClass);
111. GetDlgItemText(IDC\_EDIT\_Major, sMajor);
112. GetDlgItemText(IDC\_EDIT\_NAME, sName);
113. GetDlgItemText(IDC\_EDIT\_ID, sID);
114. **if** (((CButton \*)GetDlgItem(IDC\_RADIO\_MAN))->GetCheck()) // 判断哪个给选上，或者俩个都给选，这时候弹出提醒
115. {
116. sSex = "男";
117. }
118. **if** (((CButton \*)GetDlgItem(IDC\_RADIO\_WOMAN))->GetCheck())
119. {
120. sSex = "女";
121. }
122. GetDlgItemText(IDC\_EDIT\_Birth, sBirth);
123. EndDialog(0); // 输出一个0，表示正常输出   }
124. **查找学生信息窗口FindDlg主要代码：**
125. GetDlgItemText(IDC\_EDIT\_fNAME, fName);
126. GetDlgItemText(IDC\_EDIT\_fID, fID);
127. EndDialog(0); // 输出一个0，表示正常输出

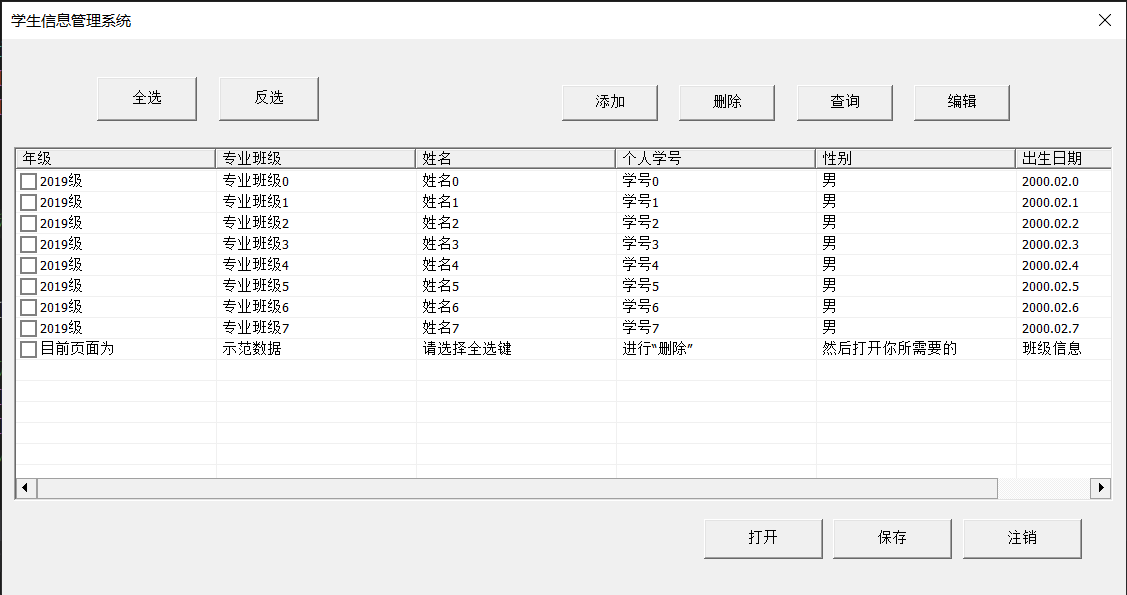
**五 系统功能模块结果图**

1. **登陆界面（以及几种出错的情况提示）**





1. **学生信息管理系统主页面**



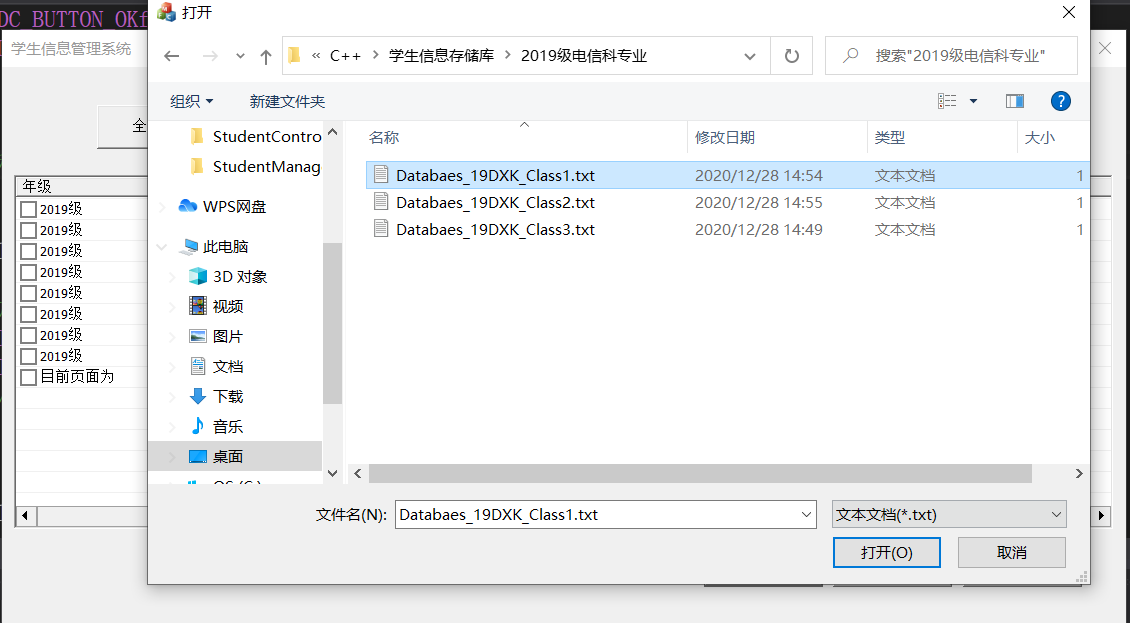
1. **在信息系统里面进行添加、删除、查询、编辑、打开加载文件、保存学生信息等功能的相应界面**



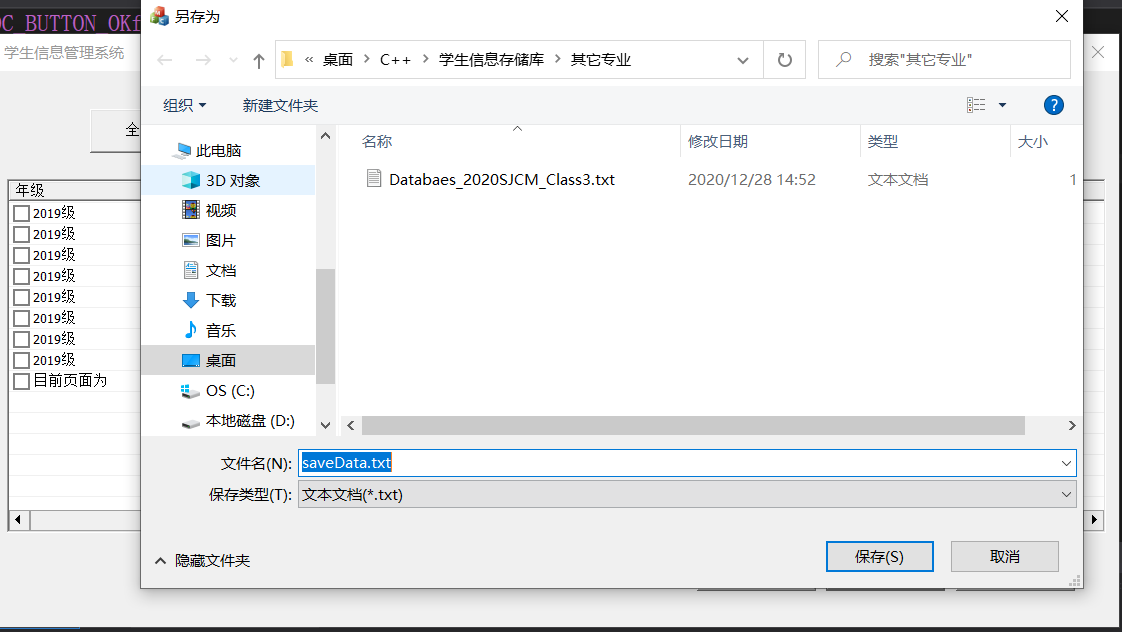
1. 添加功能窗口与编辑功能窗口



1. 删除功能与信息查询窗口

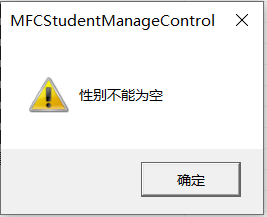


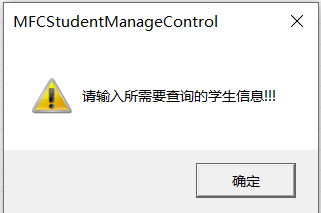
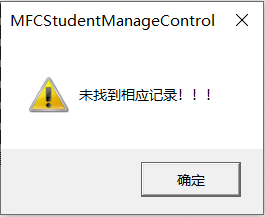
1. 加载文件功能（即加载学生信息）



1. 保存学生信息功能

**以下为几种出错相应的窗口提示信息：**





（**提示：**具体的全选、反选、直接勾选单选框删除、双击信息进行编辑、注销返回登陆界面等等的操作功能因图片无法呈现，具体可以运行程序进行相应操作实现的感受。）

**六 总结**

1. 这个学生成绩管理系统中的安全措施并不完善，整个软件也没有加壳，以至于这个软件可以直接被脱壳并找出管理员的登录账号和密码，我目前的能力水平有限，还不能对软件进行加壳加密操作。而且学生成绩管理系统的用户界面还不是很好看，使用的方面和一般的登录界面还是有点违和感。
2. 刚开始的课设制作的时候，出现了很多变量的设置，变量一多，就容易搞混，后来，我统一使用了微软标准命名法进行控件和窗口ID以及函数，类，变量的命名。这样在多个窗口的界面开发的过程中才不容易搞混。
3. 之前学会了GetDlgItemText()的用法，后来在MFC类库中查询到还有SetDlgItemText()的用法，可以直接在编辑框控件中提取信息，而不用设置变量。
4. 在RadioButton控件的使用中，我不知道如何提取按键和设置按键的信息，后来在网上查到GetCheckedRadioButton(IDC,IDC)和CheckRadioButton(IDC,IDC,IDC)这两个函数才解决问题。
5. 通过对学生成绩管理系统的可视化编程，了解了开发一个实际应用程序的步骤，同时学会使用开发工具实现界面友好的应用程序。通过本实验，掌握如何运用面向对象技术对具体的应用系统进行分析和设计。了解了MFC类的层次结构及主要类的用法，掌握使用MFC和向导来编写Windows应用程序。明白了控件是Windows图形用户界面的主要组成部分之一，掌握静态控件、按钮、流动条、编辑框、列表框、组合框和对话框的创建、消息发送和控件控制的基本用法。

# 参 考 文 献

B站JiUI极视觉. MFC程序设计——捐款管理系统（上）[DB/CP]. 中国: 哔哩哔哩, 2019.05.07. <https://b23.tv/av51677481>

B站JiUI极视觉. MFC程序设计——捐款管理系统（下）[DB/CP]. 中国: 哔哩哔哩, 2019.05.07. <https://b23.tv/av51677481>

**成绩：\_\_\_\_\_\_\_\_**