**华 南 农 业 大 学 数 学 与 信 息（软 件） 学 院**

《.net 数据库开发课程实验》成绩单

开设时间：2019学年第一学期

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 小组成员、组内分工、工作量比例、各成员个人成绩 | | | | | | | | | | | |
| **学号** | 201725110105 | **姓名** | 邓泽骏 | **分工** | 邓:界面交互设计 | | | **工作量比例** | 邓：50% | **成绩** |  |
| 实 验 题 目 | 简单数据库交互实验 | | | | | | | | | | |
| 自 我 评 价 | 本次课程设计中，为方便数据库的可移植性，我使用了Sqlite 3 来做本地数据库，相对路径的文件保证了随项目迁移而依旧能够使用。虽然本次的实验比较简陋，但是 对于数据库的基本的 增，改，删，查操作俱全。加深了我对于C#连接数据库的操作理解。  相对来说，我重新学习了数据库的相关知识，更加加强了对于数据快速操作的一些方法。 | | | | | | | | | | |
| 教 师 评 语 | 评价指标：   * 题目内容和要求完成情况 优 □ 良 □ 中 □ 差 □ * 对算法原理的理解程度 优 □ 良 □ 中 □ 差 □ * 程序设计水平 优 □ 良 □ 中 □ 差 □ * 程序运行效果及正确性 优 □ 良 □ 中 □ 差 □ * 课程设计报告结构清晰 优 □ 良 □ 中 □ 差 □ * 报告中总结和分析详尽 优 □ 良 □ 中 □ 差 □ | | | | | | | | | | |
| **成绩** |  | | | | | **教师签名** |  | | | | |

**目录**

[1 需求分析 3](#_Toc27173505)

[2 概要设计 3](#_Toc27173506)

[2.1 架构选用 3](#_Toc27173507)

[2.2 模块设计 3](#_Toc27173508)

[2.3 数据结构设计 4](#_Toc27173509)

[2.4 方法设计 4](#_Toc27173510)

[2.5 图形用户界面设计 4](#_Toc27173511)

[3 详细设计（本人主要完成文件操作相关功能） 4](#_Toc27173512)

[3.1 界面设计 4](#_Toc27173513)

[3.2 代码交互设计 6](#_Toc27173514)

[3.3 图形用户界面设计 8](#_Toc27173515)

[4 调试分析 10](#_Toc27173516)

[4.1 调试方法 10](#_Toc27173517)

[5 总结与展望 10](#_Toc27173518)

[6参考资料 11](#_Toc27173519)

# 1 需**求分**析

（1）学生信息管理系统

（2）教师信息管理系统

# 概要设计

## 2.1 架构选用

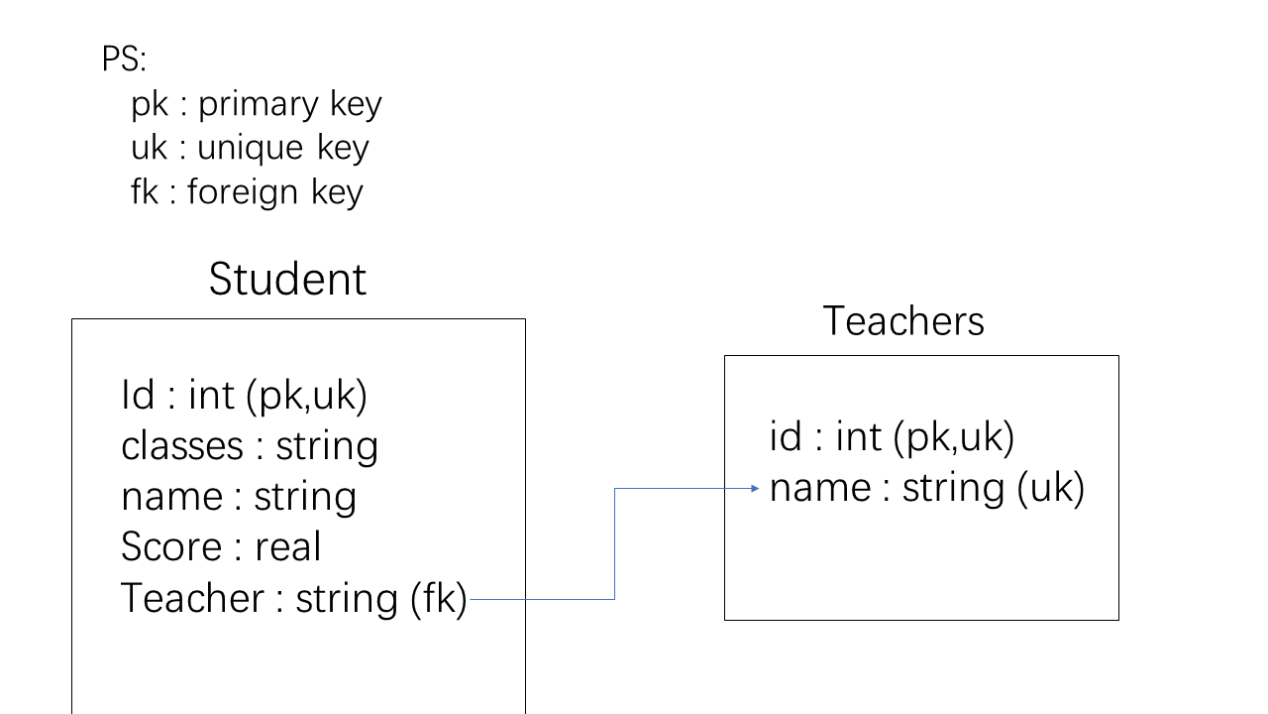
本程序使用C#语言编写，其既可以进行类似于C语言的过程化程序设计，还可以以委托等特性完成的面对对象程序设计，对于Windows系统的支持很好。

程序使用wpf(Windows Presentation Foundation)框架进行GUI(Graphic User Interface)设计。界面使用.xaml文件，使用类似于html文件的标记语言进行渲染，功能实现使用C#，是一个很好的MFC框架。

## 2.2 模块设计

基于需求分析的结果，程序设计有如下设计：

数据库设计：



## 2.3 数据结构设计

实验中主要是对于功能的实现，基于数据结构的实现几乎没有。主要以功能拼接实现。

## 2.4 方法设计

本程序底层实现的接口如下所述。

* + SqliteDataAccess类(以Student表为例)
    - LoadPeoples 装载表格全部内容
    - LoadConnectionString 得到数据库位置
    - SearchStudent 数据库查找语句
    - SaveStudent 数据库保存语句
    - AddStudent 数据库添加语句
    - DeleteStudent 数据库删除语句

## 2.5 图形用户界面设计

程序包含两个主要模块，主要为图像设计的.xaml和后台的.cs文件

主界面的简单跳转

两个分界面的功能实现

# 详细设计（本人主要完成文件操作相关功能）

## 3.1 界面设计

程序使用VS(visual studio)内置的 designer来完成主要的界面设计工作，通过对.xaml文件编辑调整界面细节设计

<Page x:Class="DatabaseExperiment.ManageStudent"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:DatabaseExperiment"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="ManageStudent" Background="#FFF8F8F8">

<Grid>

<DataGrid Name="defaultData" HorizontalAlignment="Left" Height="298" Margin="0,142,0,0" VerticalAlignment="Top" Width="800" ColumnWidth="\*" AutoGenerateColumns="True" IsReadOnly="True">

<DataGrid.ItemContainerStyle>

<Style TargetType="DataGridRow">

<EventSetter Event="MouseDoubleClick" Handler="RowDoubleClick"/>

</Style>

</DataGrid.ItemContainerStyle>

</DataGrid>

<TextBox x:Name="idText" HorizontalAlignment="Left" Height="32" Margin="82,35,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="122"/>

<TextBox x:Name="nameText" HorizontalAlignment="Left" Height="32" Margin="281,35,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="122"/>

<TextBox x:Name="classText" HorizontalAlignment="Left" Height="32" Margin="482,35,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="122"/>

<TextBox x:Name="scorceText" HorizontalAlignment="Left" Height="32" Margin="663,35,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="121"/>

<TextBox x:Name="teacherText" HorizontalAlignment="Left" Height="32" Margin="82,90,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="122"/>

<Label Content="StudentID" HorizontalAlignment="Left" Margin="6,35,0,0" VerticalAlignment="Top" Height="36" Width="69"/>

<Label Content="Name" HorizontalAlignment="Left" Margin="223,35,0,0" VerticalAlignment="Top" Height="36" Width="45"/>

<Label Content="Class" HorizontalAlignment="Left" Margin="425,35,0,0" VerticalAlignment="Top" Height="36" Width="39"/>

<Label Content="Scorce" HorizontalAlignment="Left" Margin="610,35,0,0" VerticalAlignment="Top" Height="36" Width="48"/>

<Label Content="Teacher" HorizontalAlignment="Left" Margin="10,91,0,0" VerticalAlignment="Top" Height="36" Width="56"/>

<Grid HorizontalAlignment="Left" Height="65" Margin="209,72,0,0" VerticalAlignment="Top" Width="575"/>

<Button Content="Search" HorizontalAlignment="Left" Margin="223,90,0,0" VerticalAlignment="Top" Width="76" Height="28" Click="SearchFromSQL"/>

<Button Content="Edit" HorizontalAlignment="Left" Margin="317,90,0,0" VerticalAlignment="Top" Width="76" Height="28" Click="EditClick"/>

<Button Content="Add" HorizontalAlignment="Left" Margin="411,90,0,0" VerticalAlignment="Top" Width="76" Height="28" Click="AddStudent"/>

<Button Content="Delete" HorizontalAlignment="Left" Margin="506,90,0,0" VerticalAlignment="Top" Width="76" Height="28" Click="DeleteStudent"/>

<Button Content="Back" HorizontalAlignment="Left" Margin="701,91,0,0" VerticalAlignment="Top" Width="83" Height="28" Click="ReturnMenu"/>

<Button Content="Reset" HorizontalAlignment="Left" Margin="600,90,0,0" VerticalAlignment="Top" Width="76" Height="28" Click="ResetClick"/>

</Grid>

</Page>

## 3.2 代码交互设计

本程序大部分的代码用于功能的拼接和与界面的交互。

private void SearchFromSQL(object sender, RoutedEventArgs e)

{

Student select = new Student();

string search = "";

bool flag\_id = false;

if (this.idText.Text.ToString() != "")

{

flag\_id = true;

select.id = int.Parse(this.idText.Text.ToString());

//search = "id = " + this.idText.Text.ToString();

search = "id = @id";

}

if (!flag\_id)

{

if (this.nameText.Text.ToString() != "")

{

select.name = this.nameText.Text.ToString();

//search += "name = "+this.nameText.Text.ToString();

search += "name = @name";

}

if (this.classText.Text.ToString() != "")

{

select.classes = this.classText.Text.ToString();

//search += "classes = " + this.classText.Text.ToString();

search += "classes = @classes";

}

if (this.scorceText.Text.ToString() != "")

{

select.score = Double.Parse(this.scorceText.Text.ToString());

//search += "score = " + this.scorceText.Text.ToString();

search += "score = @score";

}

if (this.teacherText.Text.ToString() != "")

{

select.teacher = this.teacherText.Text.ToString();

// search += "teacher = " + this.teacherText.Text.ToString();

search += "teacher = @teacher";

}

}

if (search.Length > 0)

{

MessageBoxResult result = MessageBox.Show("Will search from no empty box,and the id's priority is the highest", "Please CHECK!", MessageBoxButton.OKCancel);

switch (result)

{

case MessageBoxResult.OK:

{

LoadSearchList(select, search);

break;

}

default: return;

}

}else

{

LoadList();

}

}程序段3.2.1 查询语句

伪程序段3.2.2展示了程序保存文件时操作

// Edit item after clicking 'Edit' Button

private void EditClick(object sender, RoutedEventArgs e)

{

if (defaultData.SelectedIndex < 0)

{

return;

}

Student select = studentsList[defaultData.SelectedIndex];

// id is empty and repeat

if (select.id.ToString() != this.idText.Text.ToString())

{

MessageBox.Show("ID haven't been changed!");

return;

}

select.name = this.nameText.Text.ToString() == "" ? null : this.nameText.Text.ToString();

select.classes = this.classText.Text.ToString() == "" ? null : this.classText.Text.ToString();

select.score = this.scorceText.Text.ToString() == "" ? -1 : (Double.Parse(this.scorceText.Text.ToString()));

// teacher is exist

Teacher tempTeacher = new Teacher();

tempTeacher.name = this.teacherText.Text.ToString();

List<Teacher> tempList = SqliteDataAccess.SearchTeacher(tempTeacher, "name = @name");

if (tempList.Count > 0)

{

select.teacher = this.teacherText.Text.ToString();

}else

{

MessageBox.Show("This teacher isn't existed in Teacher Table. Please Add it first.");

LoadList();

return;

}

SqliteDataAccess.SaveStudent(select);

LoadList();

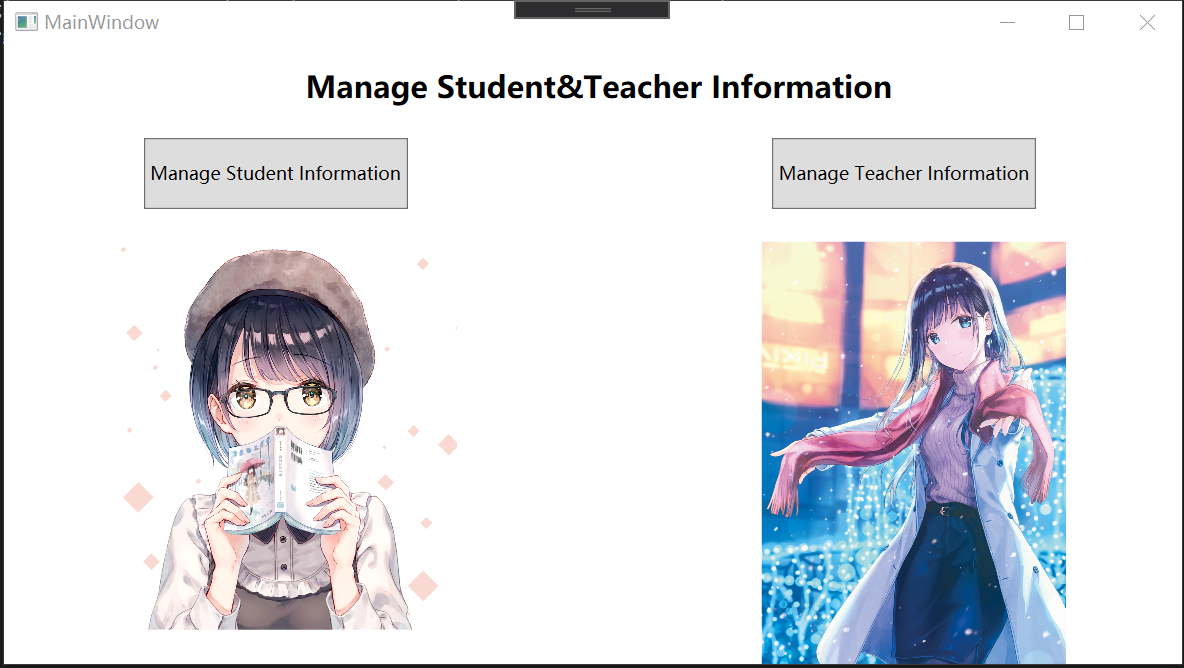
}程序段3.2.2 编辑数据

## 3.3 图形用户界面设计

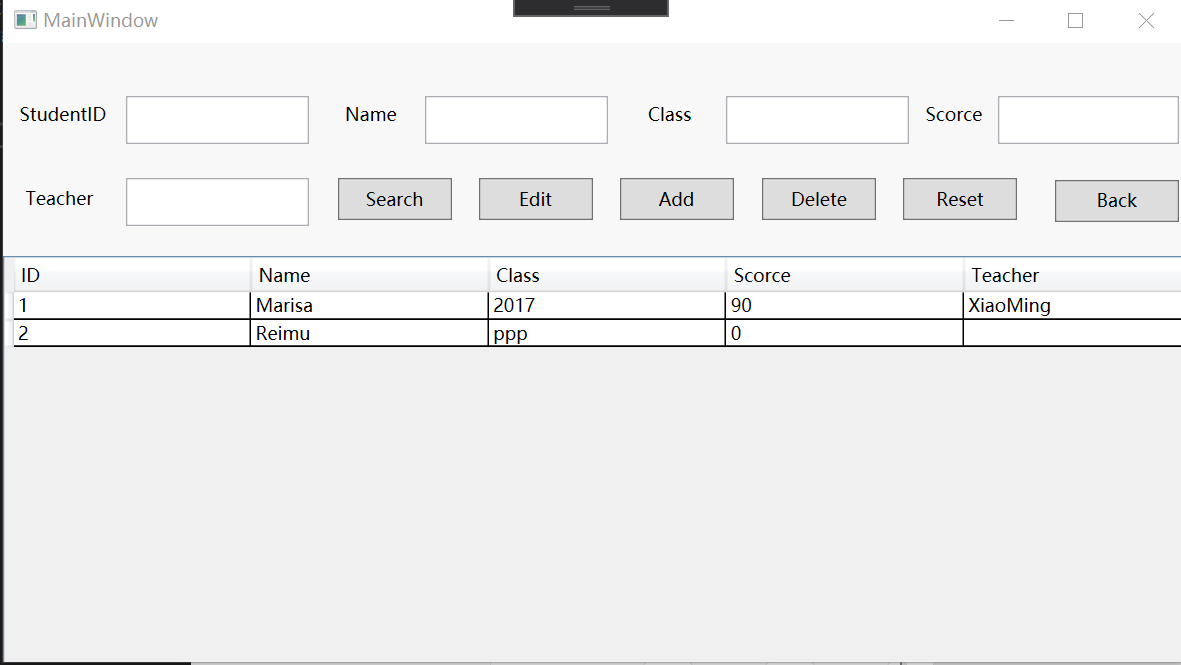
程序的图形界面通过VS的designer设计实现，本次程序界面大部分是对于.xaml文件修改实现完成，而非通过拖动窗口修改，这主要是考虑对于界面修改的便捷性和对于新方法学习的热情。

主界面分为2个功能区，左侧Student的管理和右侧的Teacher的管理

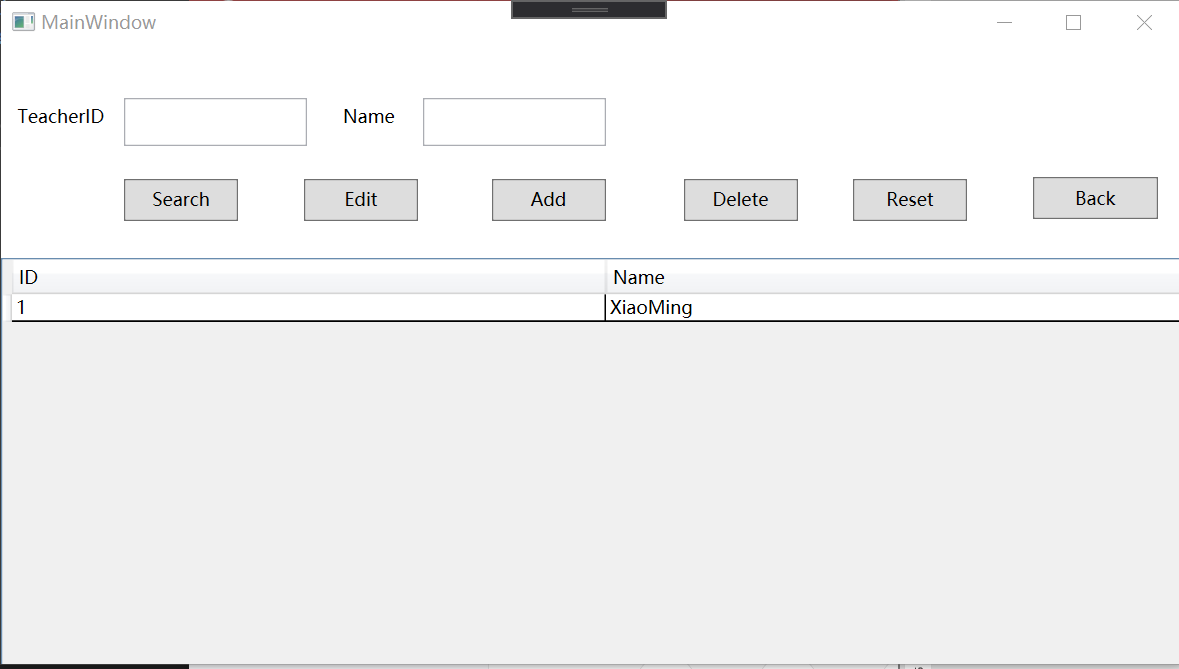
在信息管理界面，每个TextBox就是表格的信息，双击选中行可以在TextBox中看到详细信息，以及进行进一步的增改删查操作。



3.3.1主界面



3.3.2学生信息管理系统



3.3.3教师信息管理系统

# 调试分析

## 4.1 调试方法

程序开发过程中，使用先模块单元测试，再整体调试的方法。函数在控制台观察到输出符合预期 输出后，推送到稳定版本的分支，并通过给定数据流，观察输出来作黑盒测试。函数内部调试主要使用断点调试法、单步调试法和中间结果输出法，即使用Console和窗体显示输出中间结果。

# 5 总结与展望

本次课程设计中，我实现了一个基于wpf和C#的数据库管理系统。有基本的图形界面，用户可以使用工具栏和表格的方式直观的看到数据的分布以及对数据进行一系列的操作。

本程序诚然存在一些待改进之处，如美化界面，更好地显示属性值等

我成员在开发过程中，对C#语言和数据库结构有着一定的程度的进步，对于C#语言的熟悉程度加深。

# 6参考资料

[1] Show properites in DataGrid with DataTable <https://blog.csdn.net/weixin_30617695/article/details/99954197>

[2] Using SQLite in C# <https://www.youtube.com/watch?v=ayp3tHEkRc0>

[3] Using rowDoubleClick event <https://stackoverflow.com/questions/3120616/wpf-datagrid-selected-row-clicked-event>

[4] Switch different page on a window <https://social.msdn.microsoft.com/Forums/vstudio/en-US/d5a8d61c-91e6-4999-add7-5c3474922bae/how-to-open-wpf-page-in-the-wpf-window?forum=wpf>

[5] How to use MessageBox <https://www.wpf-tutorial.com/dialogs/the-messagebox/>