# 1.编译环境：

Qt6

[Qt是一个跨平台的C++框架，主要用来开发图形用户界面（GUI）程序，也可以开发不带界面的命令行（CUI）程序。Qt开发框架包含一整套高度直观、模块化的C++库类，并加载可简化应用程序开发的API。Qt可生成高可读、易维护和可重用的代码，具有较高的运行时性能，且内存占用小。Qt是面向对象的框架，使用特殊的代码生成扩展（称为元对象编译器 (Meta Object Compiler, moc)）以及一些宏，Qt很容易扩展，并且允许真正的组件编程。](https://www.bing.com/ck/a?!&&p=6f827caf02cf2312JmltdHM9MTcxOTEwMDgwMCZpZ3VpZD0wMWNmN2QxYy00OWMzLTY5ZjYtMzIxNi02OTk3NDgxMTY4YTUmaW5zaWQ9NTgwMw&ptn=3&ver=2&hsh=3&fclid=01cf7d1c-49c3-69f6-3216-6997481168a5&psq=Qt%e4%bb%8b%e7%bb%8d&u=a1aHR0cHM6Ly96aHVhbmxhbi56aGlodS5jb20vcC82MjE0MDg5NDk&ntb=1)

Sqlite3

SQLite是一个进程内的库，实现了自给自足的、无服务器的、零配置的、事务性的 SQL 数据库引擎。它是一个零配置的数据库，这意味着与其他数据库不一样，您不需要在系统中配置。就像其他数据库，SQLite 引擎不是一个独立的进程，可以按应用程序需求进行静态或动态连接。SQLite 直接访问其存储文件。

Navicat Premium 16

Navicat Premium 是一套数据库开发工具，让你从单一应用程序中同时连接 MySQL、MariaDB、MongoDB、SQL Server、Oracle、PostgreSQL 和 SQLite 数据库。它与 Amazon RDS、Amazon Aurora、Amazon Redshift、Microsoft Azure、Oracle Cloud、MongoDB Atlas、阿里云、腾讯云和华为云等云数据库兼容。你可以快速轻松地创建、管理和维护数据库。

Navicat for MySQL 是管理和开发 MySQL 或 MariaDB 的理想解决方案。它是一套单一的应用程序，能同时连接 MySQL 和 MariaDB 数据库，并与 Amazon RDS、Amazon Aurora、Oracle Cloud、Microsoft Azure、阿里云、腾讯云和华为云等云数据库兼容。这套全面的前端工具为数据库管理、开发和维护提供了一款直观而强大的图形界面。

# 2.思路

## 1,搭建数据库。

1.利用Navicat Premium 16搭建一个Sqlite3的表格。输入以下命令。

CREATE TABLE "student" (

"id" TEXT,

"name" TEXT,

"class" TEXT,

"Cpp" integer,

"Multimedia" integer,

"English" integer,

"Math" integer,

"PE" integer,

"Marxism" integer,

"Electronics" integer

"sum" integer

);

2.利用Navicat Premium 16随机生成1000个模拟数据。

## 2,项目架构

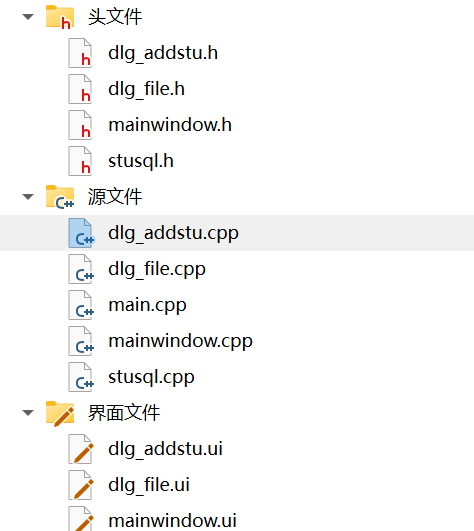
整个项目分为四个模块：stusql,mainwindow,dlg\_addstu和dlg\_file。

每个模块包括一个头文件，一个cpp文件，一个ui文件。

ui文件为Qt自动根据所设计的图形界面生成，一般不直接更改，通过更改图形设计界面进行更改。

头文件主要包括所需要库的声明，类的定义和其中的成员函数的声明。

cpp文件主要是头文件中所声明函数的具体实现。



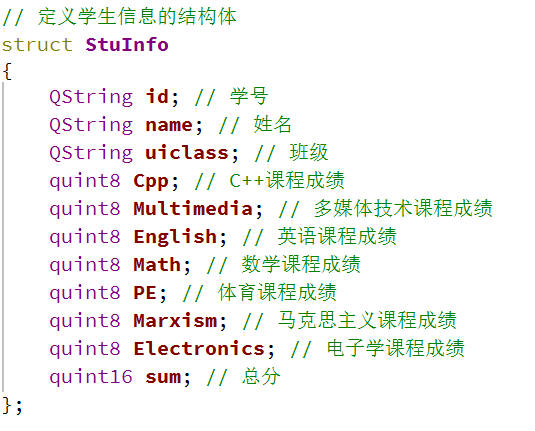
### Stusql

Stusql是主要的数据处理模块，包括一个头文件，一个cpp文件。

该模块的头文件主要包括所需要库的声明，类的定义和其中的成员函数的声明。成员函数的主要功能调用各种功能的SQL语句。

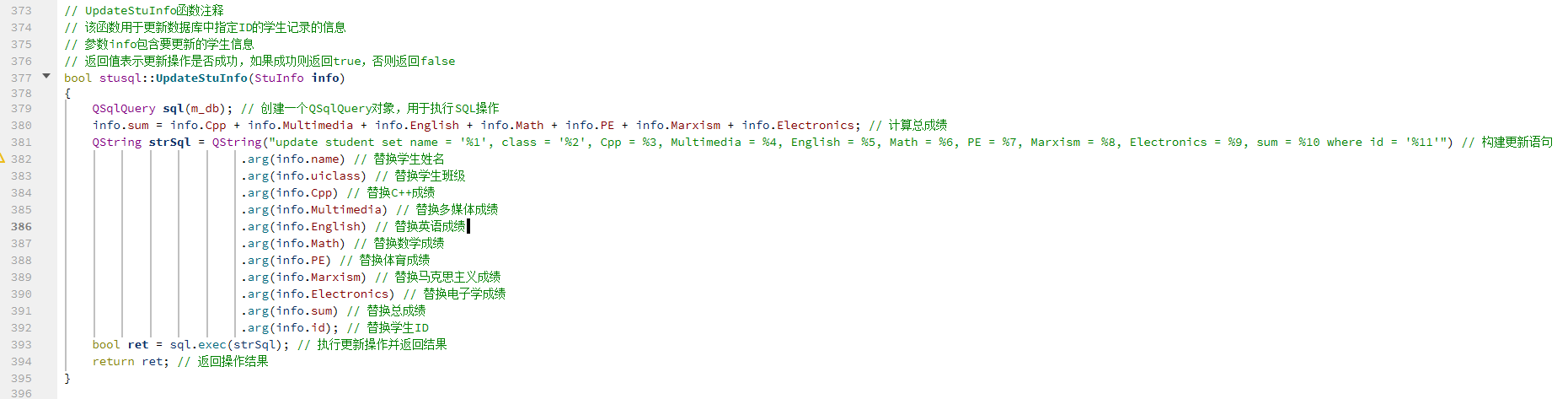
该模块的cpp文件主要是对头文件声明的成员函数的实现。

下面是他的头文件的部分定义与函数声明。





下面是cpp文件中一个函数（更新某个学生的信息的函数，即“修改”功能）的实现，作为例子。



其中这个语句便是这个函数所调用的SQL语句：

“update student set name = '%1', class = '%2', Cpp = %3, Multimedia = %4, English = %5, Math = %6, PE = %7, Marxism = %8, Electronics = %9, sum = %10 where id = '%11'”

通过调用该行SQL语句，根据ID查找该学生，并用用户输入的信息替换该学生的原信息，实现修改功能。

### Mainwindow

Mainwindow模块包括一个头文件，一个cpp文件，一个ui文件。

该模块的ui文件对应的是主界面的设计的实现。

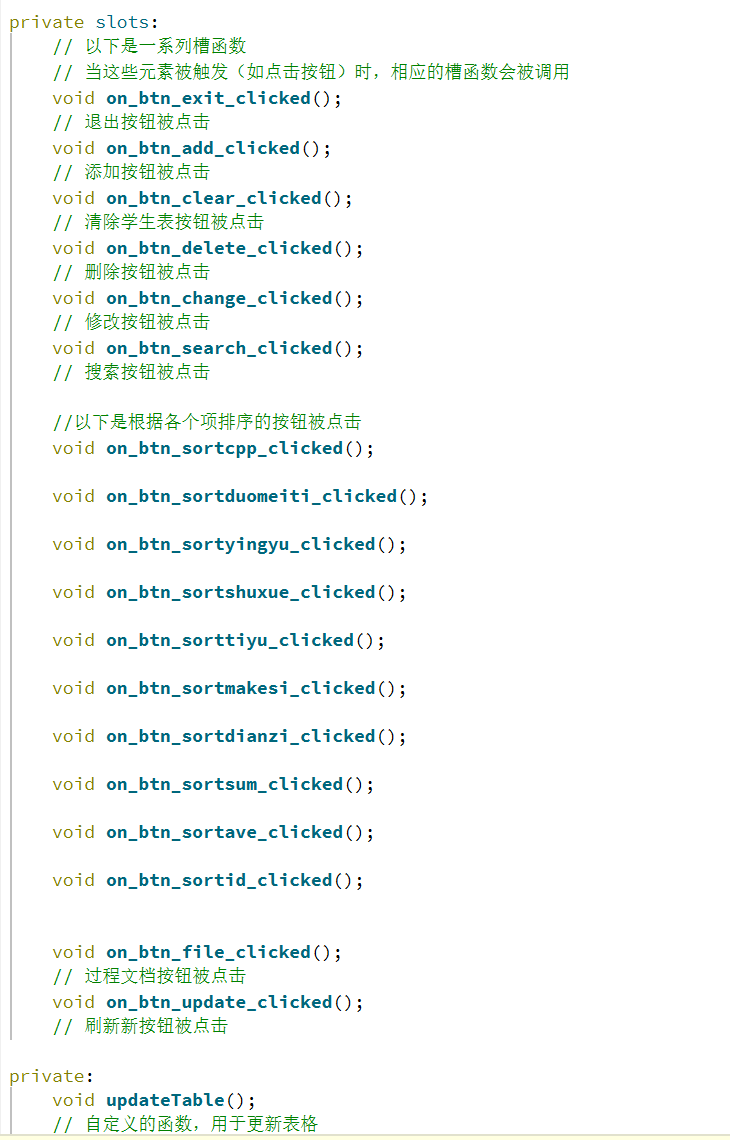
该模块的头文件主要包括所需要库的声明，类的定义和其中的成员函数的声明。成员函数的主要功能是实现每一个主界面按钮对应的的槽函数。

该模块的cpp文件通过调用stusql模块的函数实现每一个槽函数的功能。

下面是它的界面。



下面是它的头文件包含的槽函数的声明。



下面是cpp文件中一个槽函数（按学生的ID进行排序的按钮）的实现，作为例子。



### dlg\_Addstu

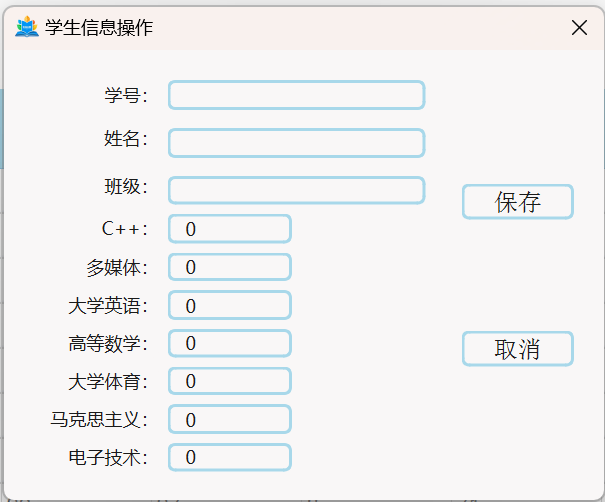
dlg\_Addstu模块包括一个头文件，一个cpp文件，一个ui文件。

该模块的ui文件对应的是点击“添加”或“修改”按钮后出现的子界面。

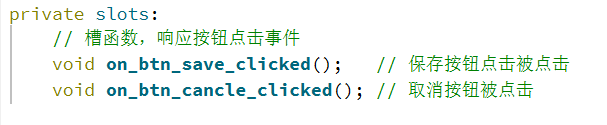
该模块的头文件主要包括所需要库的声明，类的定义和其中的成员函数的声明。成员函数的主要功能是实现每一个该子界面按钮对应的的槽函数。

该模块的cpp文件通过调用stusql模块的函数实现每一个槽函数的功能。

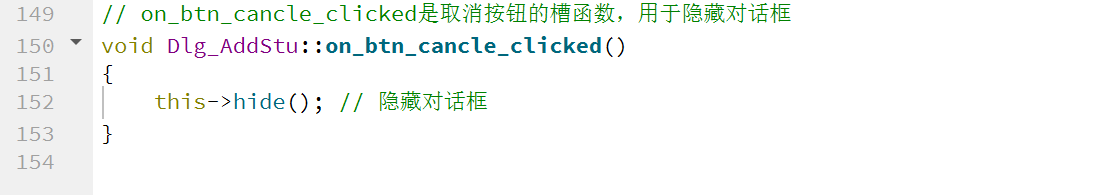
下面是它的界面。



下面是它的头文件包含的槽函数的声明。



下面是cpp文件中一个槽函数（按学生的ID进行排序的按钮）的实现，作为例子。



### dlg\_File

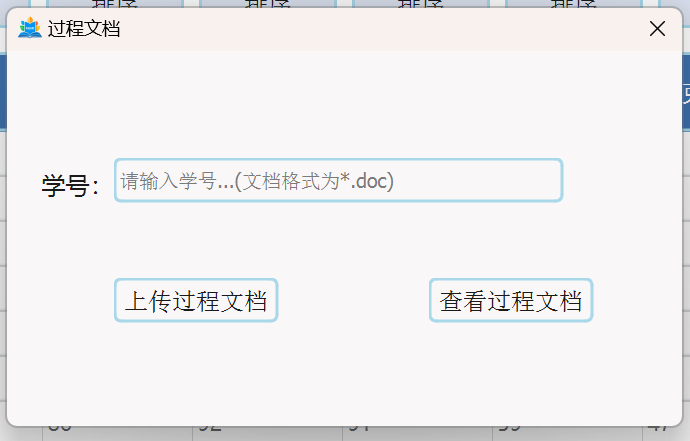
dlg\_File模块包括一个头文件，一个cpp文件，一个ui文件。

该模块的ui文件对应的是点击“过程文档”按钮后出现的子界面。

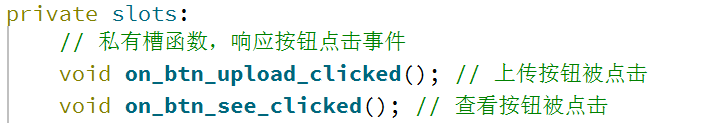
该模块的头文件主要包括所需要库的声明，类的定义和其中的成员函数的声明。成员函数的主要功能是实现每一个该子界面按钮对应的的槽函数，实现“上传”“重命名”“查看”等功能。

该模块的cpp文件通过调用stusql模块的函数实现每一个槽函数的功能。

下面是它的界面。



下面是它的头文件包含的槽函数的声明。



下面是cpp文件中一个槽函数（查看的按钮）的实现，作为例子。



# 困难和解决方法

1.问题：大量的bug，主要是输入的数据与需要的数据不符造成错误。

解决方法：对每一个输入都进行验证，并输出相应的提示信息。

2.问题：过程文档功能的实现。

有两个方案：

方案一、在每个同学的行的后面，都加上“上传”“查看”的按钮。

方案二、单独一个“过程文档”的功能按钮，通过“学号”或者点击选择 来 上传或查看某个同学的过程文档。

解决：考虑到便利性，我选择了方案二。

3.问题：界面不够精致。

解决方法：通过编写Qss文件，美化界面。

Qss文件：Qt Style Sheets（QSS）作为一种强大的样式定制工具，为开发者提供了类似于CSS的样式设置机制，使得界面设计更加灵活和多样化。通过QSS，开发者可以轻松地为Qt应用程序中的控件定义外观和行为，从而创建出既美观又具有良好用户体验的界面。

美化前：



美化后：



程序实现的效果及优劣自评：

效果：实现了项目要求的功能，处理速度快，具有相对美观的界面，易于操作。等等（你可以继续编）；

优点：有界面，美观，易操作，bug几乎没有，在面对各种不规范输入有很全面的信息提示。处理速度快，文件架构清晰，可移植性极强，无广告。

可改进的地方：未能实现多端互联协作功能，不能联网，互相协助只能靠互相交换数据库文件。未能实现权限功能，安全性不高。