

# 北京邮电大学教务系统客户端

## 程序设计文档



SCHEDULE\_BUPT

## 目录

程序简介.....	3
程序使用概览.....	3
设计任务描述.....	8
功能需求说明.....	9
总体方案设计.....	10
各模块实现及数据结构说明.....	13
评价及改进意见.....	18

## 一、 程序简介

- 程序名称

BuptSchedule (北京邮电大学教务系统客户端)

- 开发环境

Qt 5.9 集成开发环境

- 开发语言

C++面向对象

- 版本号

V 1.3

## 二、 程序使用概览

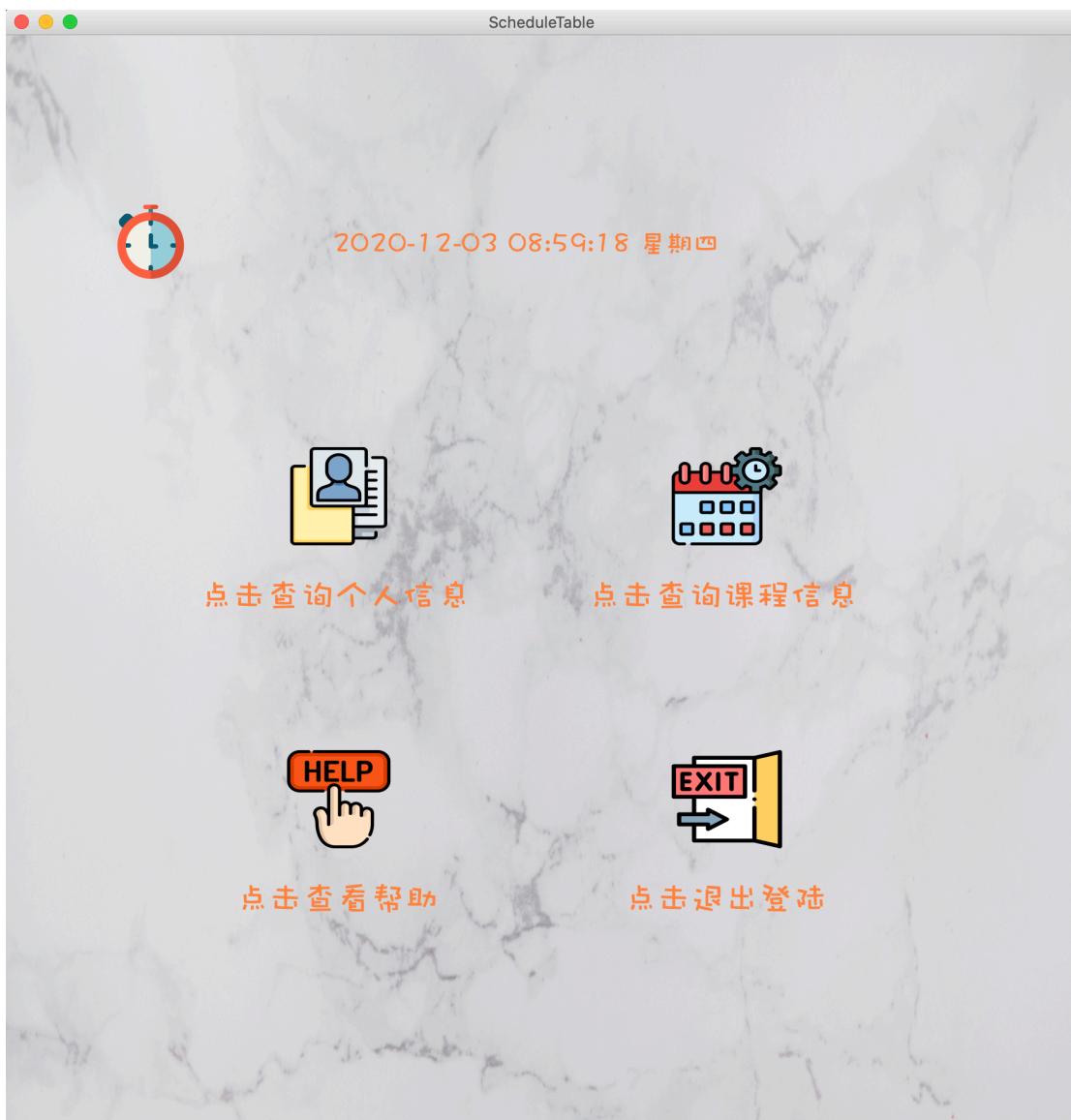
- 主界面

用户在打开程序会显示如下界面，学生可以输入相应的个人信息登入到教务系统中。



## ● 教务系统界面

如果学生信息验证成功，则程序会显示如下的教务系统界面，学生可以根据个人需求去查询不同信息。



如图，用户可以选择查询个人信息（显示学号，姓名，学院，专业，已修学分，课程绩点等信息），也可选择查询课程信息（显示课程表信息，并可以查询课程详细信息）。

程序还提供了帮助功能，不熟悉程序相关操作的用户可以先查看程序的帮助（程序使用教程）。

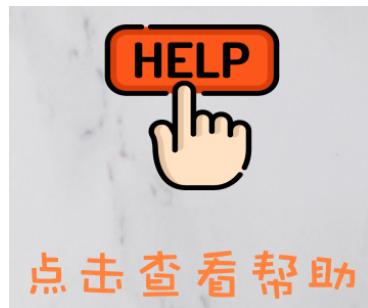
查看个人信息：



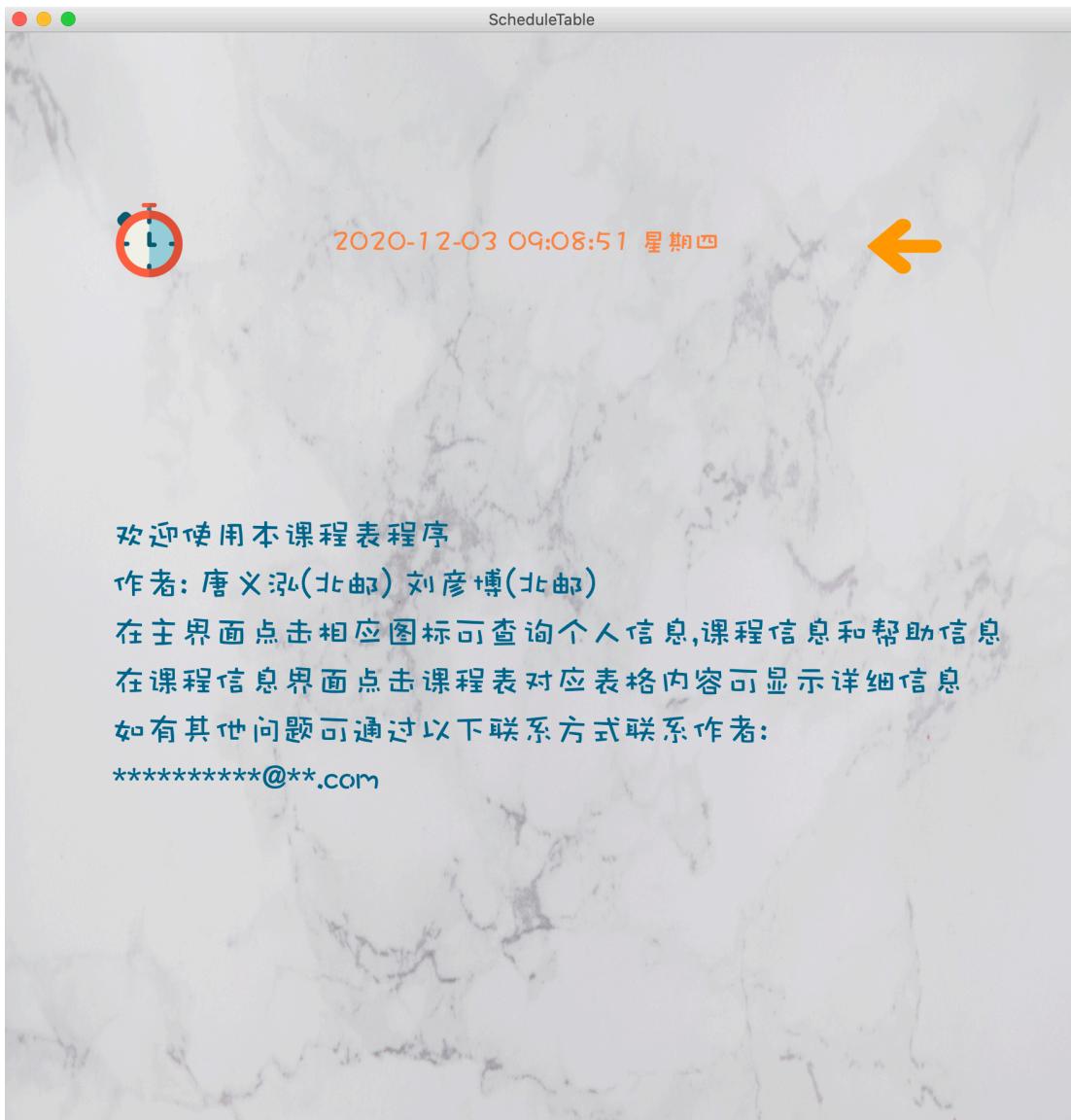
点击如上按钮以后会显示如下界面：



查看帮助信息：



点击如上按钮以后会显示如下界面：



查看帮助信息：



点击如上按钮以后会显示如下界面：

ScheduleTable

2020-12-03 09:12:56 星期四

	MON	TUE	WED	THU	FRI
1	ID: 10004 量子计算导论				
2	ID: 10005 Python实践	ID: 10008 高等数学A上	ID: 10010 英语情景视听说		
3				ID: 10012 网络存储技术	
4		ID: 10009 大学英语B上		ID: 10013 计算机网络	
5	ID: 10006 数据库原理				
6			ID: 10011 无线传感器网络		ID: 10003 DS数据结构
7				ID: 10014 形式与政策	
8	ID: 10007 离散数学			ID: 10015 创新创业实践	

点击按住课表相应表格内容以后会显示课程详细信息：



### 三、设计任务描述

程序为学生提供教务系统客户端，学生可以通过自己的账号和密码登陆教务系统并查询自己的个人信息，学业信息，课程信息。

为保证程序高度模块化以及日后方便扩展功能，将程序划分为三个模块通过类实现，模块之间通过相应类方法进行信息交互。模块之间的内部构造彼此相互屏蔽。



程序设计任务可分为数据库模块，查询模块以及图形化模块分别实现存储和查询学生信息，以及通过图形化界面将信息显示给学生的功能。各个模块彼此分离，扩展性强。

## 四、组内分工说明

我组由唐义泓(学号 2018211328), 刘彦博(学号 2018211328)组成, 二人分工一起完成本次的程序设计实践。

唐义泓同学主要负责程序整体的架构设计(模块设计, 模块交互逻辑设计, 程序整体逻辑设计), 图形化模块的设计, 程序界面的布局以及显示逻辑的设计及实现。

刘彦博同学主要负责数据库模块与线程查询模块, 程序并发性功能的设计与实现, 同时设计了数据库, 并且制作了一些相关的数据用于测试开发。

二人各自模块的设计与提供给对方模块的接口由二人商讨决定, 在各自模块实现之后将模块进行接合形成了最终的程序。

文档方面, 二人各自完成各自的模块的文档及截图, 最后将二人所写文档拼合而成。

总体来说, 在本次程序设计实践中, 二人的工作量较为平均, 对于目前所进行的程序设计实践的程序规模来说, 平均的分工其实并不是容易的事情, 所以此次程序设计的分工以及最后的结果是合理且对本次程序设计的效率有提高的。

## 五、模块功能说明

程序的三个模块，数据库模块，线程查询模块，图形化显示模块

### ● 数据库模块

数据库模块可通过传入参数设置数据库的相关信息。

数据库模块可连接数据库以访问学生个人信息，学生课程信息以及学生学业信息，并设置相应的接口与其他模块实现数据交互操作。

模块独立性强，部分函数设置为虚函数以便日后扩展。

### ● 线程查询模块

线程查询模块是图形化模块和数据库模块交互的中间模块。

线程查询模块实现了并发查询，每次查询通过增加一个子线程的方式实现并发功能。以保证在大量用户同时进行访问教务系统的同时不会导致程序假死，以提高用户体验的同时保证程序的鲁棒性。

### ● 图形化显示模块

图形化界面分为两个界面：登陆界面以及教务系统界面。

登陆界面实现学生登陆信息的输入及验证功能，教务系统界面在学生成功登录后显示给学生，学生可通过在教务系统界面进行相关操作访问并查看个人的各种信息。

登陆界面相比教务系统界面更简洁，方便学生通过简单的操作登入自己的账号，在学生操作错误时进行错误处理并返回正确的提示信息。

教务系统界面相比登陆界面复杂，但依旧保持信息显示整洁易于查看，界面对称有美感，对人亲和性强，同登陆界面一样，在学生执行了错误的操作以后进行相应的错误处理并返回正确的提示信息。

## 六、 总体方案设计

程序由三个模块，两个界面组成，界面显示变化频繁，模块间数据交互频率高，数据量大，故将程序总体执行逻辑也进行切分。

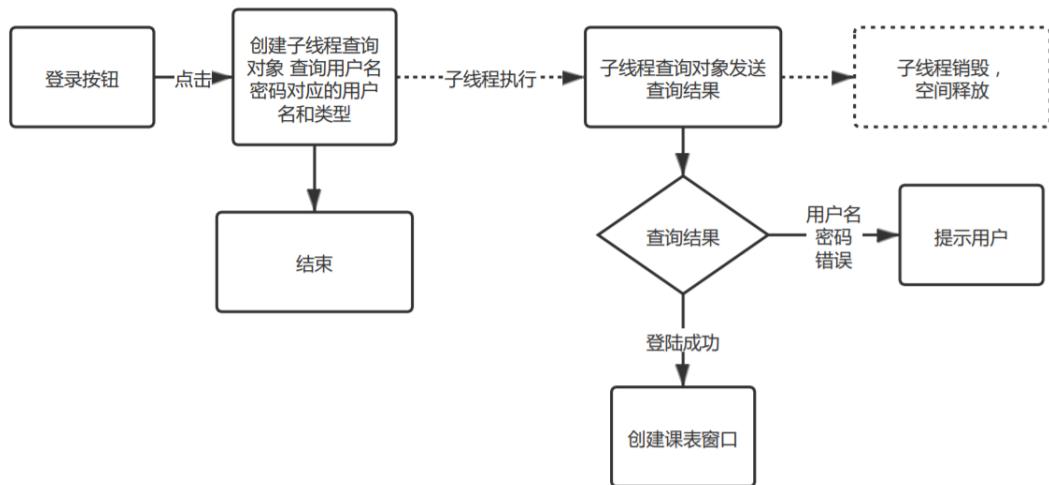
总体执行逻辑方案设计为五个部分：登陆验证逻辑，个人信息查询逻辑，课表查询逻辑，课程详细信息查询逻辑以及安全退出逻辑。

### ● 登陆验证逻辑

学生在图形化的登陆界面个人信息输入完毕，点击登陆时，程序通过利用子线程的并发技术，通过数据库模块的接口访问数据库，查询学生的信息（账号密码）是否均输入正确。

若账号密码均输入正确，则打开教务系统界面，创建子线程，查询学生相关信息并进行后续逻辑操作，否则关闭，将子线程销毁同时释放空间，并显示相应的错误提示信息给学生。

登陆验证逻辑具体如下图所示：



### ● 个人信息查询逻辑

在教务系统界面，学生可以通过点击个人信息按钮打开个人信息窗口以查看个人信息（如学分信息，绩点信息等）。此时程序创建子线程执行相应的查询，并将结果发送图形化模块，图形化将结果显示。

完成上述过程后子线程销毁，同时空间被释放

个人信息查询逻辑具体如下图所示：



### ● 课表查询逻辑

学生在教务系统界面可以通过点击课程信息按钮查看自己的课表或详细课程信息。此时程序创建子线程执行相应的查询，并将结果发送图形化模块，图形化将结果显示。

完成上述过程后子线程销毁，同时空间被释放

个人信息查询逻辑具体如下图所示：



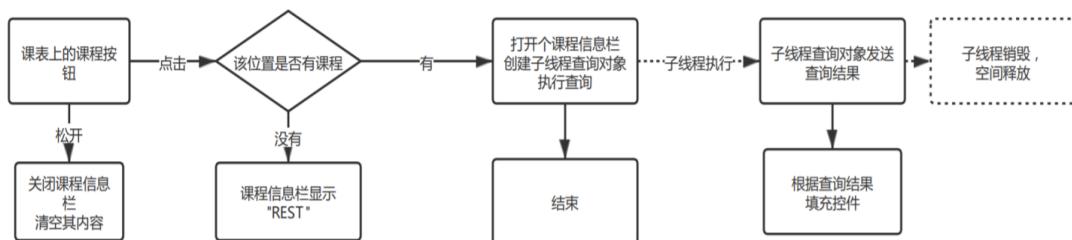
## ● 课程详细信息查询逻辑

学生点击课程信息按钮后可以看到显示的课表，通过点击课表中每节课对应的表格可以显示课程的详细信息。

每次点击时会界面上方显示出课程详细信息，在松开点击后课程详细信息。

内部逻辑是点击事件发生的同时创建子线程进行相应课程详细信息的查询工作，并将结果发送图形化模块，并通过图形化模块的接口将结果显示在界面上，在松开点击的时候结束显示，并销毁子线程，释放空间。

课程详细信息查询逻辑具体如下图所示：



## ● 安全退出逻辑

学生在完成自己所需的查询操作之后，可点击退出按钮退出客户端程序。此时程序需要进行安全退出，确保线程，进程安全。

安全退出逻辑具体如下图所示：



## 七、各模块实现及数据结构说明

- 数据库模块

模块名	头文件	模块简介
MyDatabase	MyDatabase.h	数据库类，用于数据库的连接与操作
<hr/>		
返回值	函数声明	函数功能简介
Public Functions		
	<pre>MyDatabase(     QString Name = "",     QString Addr = "",     QString Username = "",     QString Password = "");</pre>	数据库对象构造函数
	<pre>~MyDatabase();</pre>	数据库对象析构函数。断开连接，删除对象。
	<pre>virtual InnerQuery(     QList&lt;QString&gt; Operation,     &lt;QString&gt; int Col);</pre>	用于对当前数据库对象执行 SQL 语句并返回查询结果
	<pre>void SetDataBase(     QString Name,     QString Addr,     QString Username = "",     QString Password = "");</pre>	用于对当前数据库对象设置基本配置信息
	<pre>virtual int ConnectToServer();</pre>	使数据库对象连接到数据库
	<pre>virtual int DisConnectToServer();</pre>	使数据库对象和数据库断开连接
<hr/>		
Private Variables		
	<pre>QMutex SharedAreaLock;</pre>	维护查询过程线程安全的锁
	<pre>QMutexLocker SharedAreaLocker;</pre>	维护查询过程线程安全的高级锁（可以自动释放）
	<pre>bool Valid;</pre>	标识符，用于指示当前数据库是否可以执行操作
	<pre>QString DatabaseName,     DatabaseAddr,     DatabaseUsername,     DatabasePassword;</pre>	数据库的名称，地址，用户名，密码
	<pre>QSqlDatabase DB;</pre>	数据库 SQL 对象

数据库类 (MyDatabase)：通过传入参数设置数据库的相关信息。也可以通过 SetDatabase() 设置。之后调用连接方法 ConnectToServer() 进行连接。通过 InnerQuery() 方法可以执行 SQL 语句。本类中一些方法设置为虚函数以便于程序扩展中的重写。

## ● 线程查询模块

模块名	头文件	模块简介
SubThreadQuery	SubThreadQuery.h	子线程查询类，用于创建子线程执行 SQL
返回值	函数声明	函数功能简介
Public Functions		
<pre>SubThreadQuery(     MyDatabase* DatabaseUsedArgv,     QString CommandArgv,     int QueryColArgv,     int QueryIdArgv); void start()</pre>		
Protected Functions		
virtual void	run() override;	重写的线程运行方法，用于执行查询操作
Signals		
<pre>SendQueryResult(     QList&lt;QString&gt;, int, int);</pre>		
Private Variables		
<pre>MyDatabase* DatabaseUsed; QString Command; int QueryCol; int QueryId;</pre>		

程序设计中构建了一个子线程查询类 (SubThreadQuery)，这个查询类会在

子线程中执行提前设好的 SQL 语句，在执行结束后通过信号量发送查询结果。

程序中所有和数据库模块的交互均是通过以上所描述的类完成的，使用子线程进行并发查询的目的是为了在数据库因有大量交互而响应缓慢的情况下不会导致应用程序假死，可以提高用户体验。如果不这样做，每个查询都是同步查询，那么用户点击按钮后可能 QT 程序一直在执行 SQL 语句，导致用户界面卡住。

查询的具体过程为：随机数生成 ID 添加到对应用于记录查询 ID 的集合中去。构建子线程查询对象并通过构造函数传入包括数据库对象、SQL 语句、查询行、查询 ID 在内的参数。之后通过调用子线程查询的 start() 方法开始查询。

### ● 图形化模块（主窗口）

模块名	头文件	模块简介
MyMainWindow	MyMainWindow.h	主界面类，用于提供用户登录等服务
返回值	函数声明	函数功能简介
Public Functions		
explicit	MyMainWindow( QWidget* parent = 0); ~MyMainWindow();	登录窗口构造函数。初始化控件，连接槽函数。 登录窗口析构函数。
Private Slots		
void	UserConnectToServer();	点击重新连接数据库按钮对应的槽函数
void	UserLogin();	点击登录按钮对应的槽函数
Private Variables		
MyMainWindow* ui;		对象的图形界面指针 ui
QSet<int> LoginApplyQueryID;		登录请求查询的 ID 集合
Public Slots		
void	RecieveQueryResult( QList<QString> QueryResult, int QueryState, int QueryID);	用于接收子查询结果并执行对应操作的槽函数

模块名	头文件	模块简介
ScheduleTable	ScheduleTable.h	课表类，用于显示课表以及控件
返回值    函数声明		
	Public Functions	
<pre>explicit ScheduleTable(     QString User); ~ScheduleTable();</pre>		
<p>ScheduleTable 类构造函数，用于初始化 ui 界面，控件</p> <p>ScheduleTable 类析构函数，用于删除 ui 界面</p>		
Public Slots		
<pre>void RecieveQueryResult(     QList&lt;QString&gt; QueryResult,     int QueryState,     int QueryID);</pre>		
<p>用于接收子线程查询结果并执行对应操作的槽函数</p>		
Private Functions		
<pre>bool eventFilter(     QObject* watched,     QEvent* event);</pre>		
<p>事件过滤器，接收拦截到的事件，用于实现 QPainter 绘图事件</p>		
<pre>void PaintTableStructure(void);</pre>		
<p>使用 QPainter 绘制课程表框架</p>		
<pre>void ApplyStudentClasses(     QString User);</pre>		
<p>申请子线程异步并发查询学生的课表</p>		
<pre>void ApplyStudentInfo(QString User);</pre>		
<pre>void ApplyClassInfo(QString ClassID);</pre>		
<p>申请子线程异步并发查询学生的详细信息</p>		
<pre>void DisplaySchedule(     QList&lt;QString&gt; UserClasses);</pre>		
<p>根据参数显示课程表内容</p>		
<pre>void DisplayStudentInfo(     QList&lt;QString&gt; StudentInfo)</pre>		
<p>根据参数显示学生详细信息</p>		
<pre>void DisplayClassInfo(     QList&lt;QString&gt; ClassInfo);</pre>		
<p>根据参数显示单个课程详细信息</p>		
Private Slots		
<pre>void on_PersonalInfo_clicked();</pre>		
<p>个人信息按钮点击的槽函数</p>		
<pre>void on_ClassInfo_clicked();</pre>		
<p>课程信息按钮点击的槽函数</p>		
<pre>void on_Return_clicked();</pre>		
<p>返回按钮点击的槽函数</p>		
<pre>void on_Help_clicked();</pre>		
<p>帮助按钮点击的槽函数</p>		
<pre>void on_Exit_clicked();</pre>		
<p>退出按钮点击的槽函数。销毁当前对象</p>		
<pre>void timerUpdate(void);</pre>		
<p>更新时间标签的槽函数</p>		
<pre>void on_MonSecTwo_pressed();</pre>		
<p>按钮点击的槽函数</p>		
<pre>void on_MonSecTwo_released();</pre>		
<p>按钮释放的槽函数</p>		

## Private Variables

ScheduleTable* ui;	本类对应图形界面指针
QString Username;	本课表对应学生的用户名
QSet<int> StudentAllClassesApplyID;	存储“学生课表查询”的子线程查询 ID
QSet<int> StudentInfoApplyID;	存储“学生详细信息查询”的子线程查询 ID
QSet<int> ClassInfoApplyID;	存储“课程详细信息查询”的子线程查询 ID

主窗口类 (MyMainWindow) 提供用户交互的登录界面，包括输入用户名密码，登录按钮。以及连接数据库失败时候连接数据库的按钮和输入框。

主窗口类在创建时会自动执行一次数据库连接操作，如果失败之后则重新设置数据库路径然后点击按钮连接。

点击登录按钮后，会获取输入的用户名密码。进行合法性检验后会创建子线程查询类异步请求登录。收到结果的信号后槽函数会根据查询结果显示是否登陆成功。如果登录成功则会创建课表类并显示。

课程表类 (ScheduleTable) 用于提供学生用户登录后的界面，提供交互界面。一级界面为“个人信息”，“课程信息”，“帮助”，“退出”。其中前两个选项会通过数据库进行交互来查询相关信息，查询方式为异步并发查询。

点开个人信息，会通过构建子线程查询类查询用户的详细信息。查询结束后会根据返回结果的信号设置相应区域的内容。

点开课程信息会通过子线程查询类查询用户的课表并显示在区域上。按住相应课程可以查询该课程的详细信息。松开则会关闭。

## 八、评价及改进意见

### ● 程序评价

程序的功能实用价值很高，现今大多数教务系统的学生成绩显示的并不够直观，学生大多都适用第三方可以爬取教务系统数据的课程表程序来查看成绩，这样其实会浪费不少时间，我们的程序设计很好的解决了这个问题。

程序具备良好的人机交互界面，操作方式符合用户直觉，与主流程序的操作方式一致。图形界面简洁、元素反差对照明显、利用空白分割、显示平衡恰当、元素对齐。是一个视觉效果极好的程序。

程序使用多线程实现了高并发的数据交互功能，支持大量用户在线操作，数据异步更新，可以满足一个教务系统程序的实际需求。

程序的健壮性强，在用户输入错误数据、网络连接中断、鼠标随意点击拖拽等异常情况下不会出现异常退出、挂起、输出内部错误信息等情况，能够提供用户易于理解和接受的反应。

程序扩展性强，程序采用了结构型模式和创建型模式组合的设计模式，高度模块化，程序在实现教务系统课表原型的同时构造了一套由“发送请求”“查询结果”“发送结果信号”构成的异步数据查询框架。在程序未来的开发中可以借助此框架扩展新的功能。

## ● 改进意见

本程序为“教务系统课表”的原型，在程序的设计过程中并不仅考量原型程序的设计，更考虑了程序未来的扩展性和可复用性。从可复用上来讲：如 SubThreadQuery 和 MyDatabase 模块，这两个模块均可以取出用于其他程序的设计，其他程序的设计者可以派生以上两个类并通过一定程度的重写来添加新的功能。

从扩展性上来讲，本程序在实现教务系统课表原型的同时构造了一套由“发送请求”“查询结果”“发送结果信号”构成的异步数据查询框架。在程序未来的开发中可以借助此框架扩展新的功能，如选课，修改个人信息，查看课程教室，查看期末成绩，查看培养方案。同时也可以根据登录用户的不同显示不同的页面，如可以设置老师专用版页面，可以有录入成绩等功能。