**第二次实训工作报告**

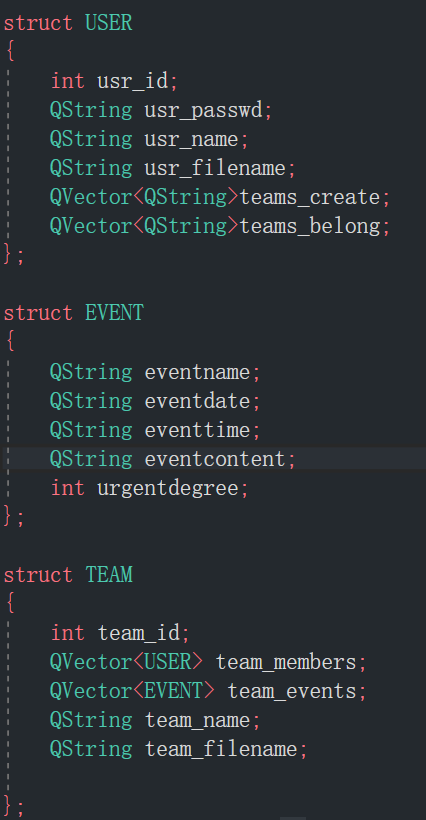
软件2班 杨璐玮

**本周工作小结：**

本周由于黑框框比赛搁置了比较多的时间。进展比较缓慢，但是也有一些激动人心的想法和实现。

上周由于数据库的问题花费了较多的时间，但是经过交流讨论发现我忽略了实训要求不能使用数据库而是要进行文件操作，而我上周的日程信息存储实现刚好是使用txt文件存储。刚好我在黑框框比赛中学习并使用了一些数据通过文件存取的操作。以及用户管理的一些技术方法。于是本周工作如下：

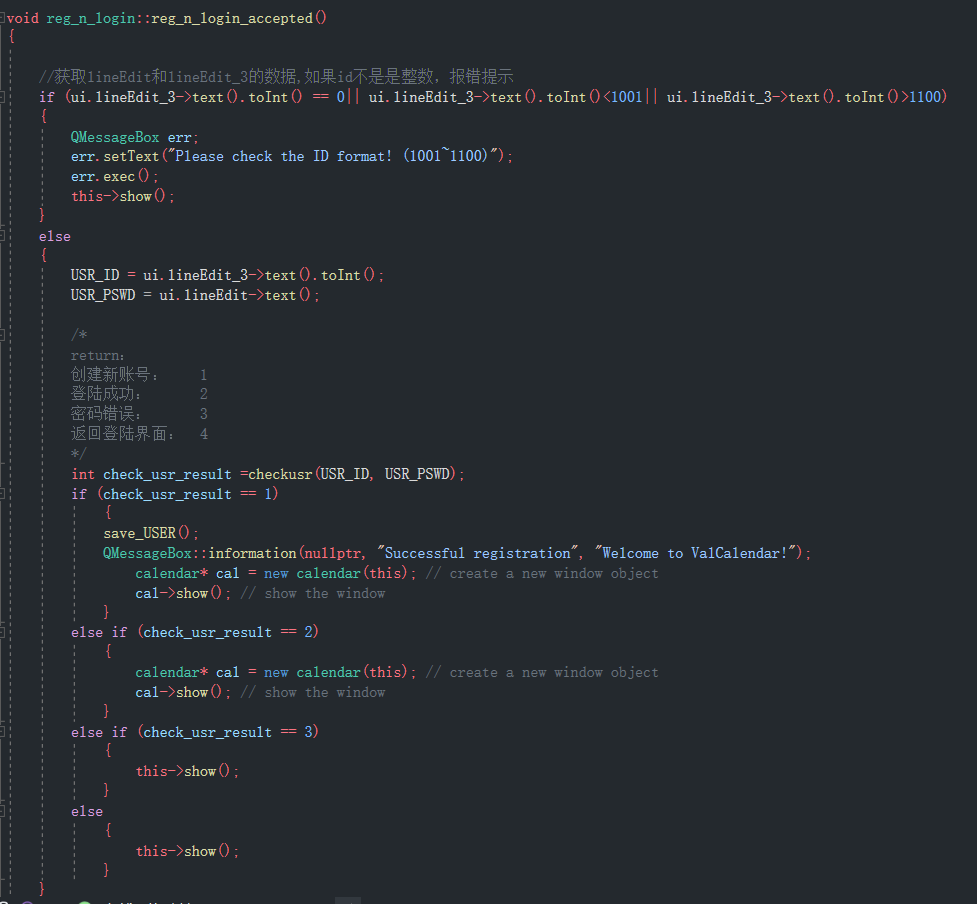
1. 使用结构体数组存放用户信息进行封装，便于后续联网功能的实现。（在socket编程中，使用结构体序列化进行信息传输是常用手段）。
2. 基本实现了用户注册登录的完整逻辑和流程以及相关数据结构。
3. 结合黑框框的项目经验，重构了项目文件的层次结构。
4. **重要数据结构体一览：**



以上是代表用户、日程和团队的结构体。后续用户、日程和团队数据的存取会严格按照这个成员数据进行，保证数据的一致性和合法性。

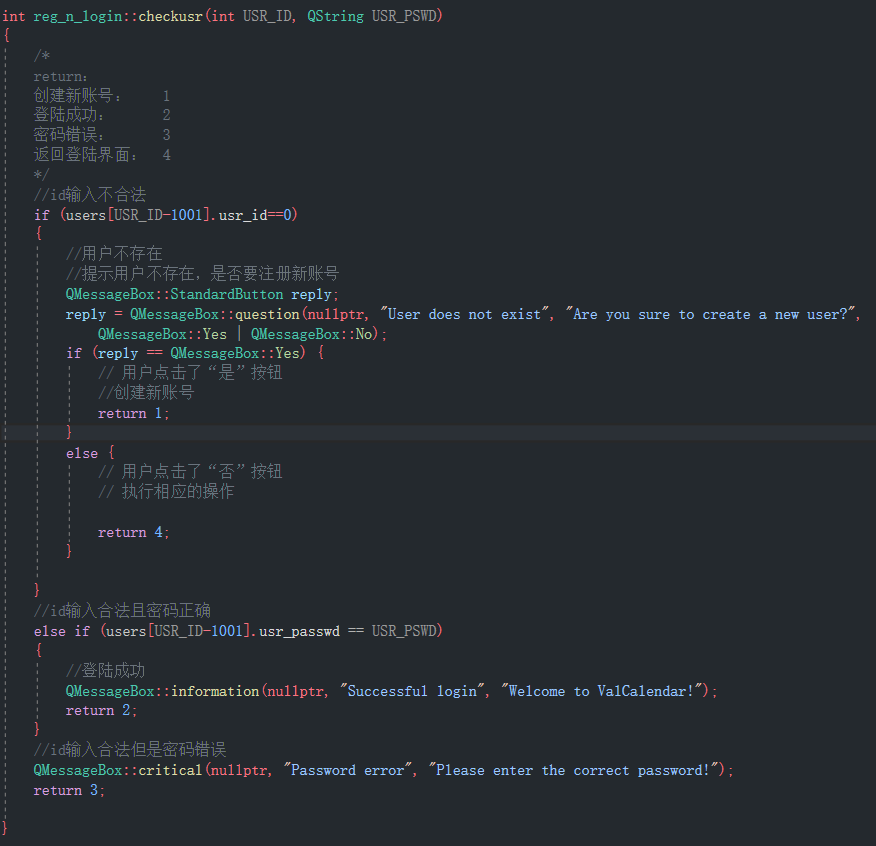
1. **用户登录/注册模块功能的具体实现细节：**

首先是reg\_n\_login窗口会收到用户输入的id和passwd，接下来如果用户点击OK进行登录操作，那么该事件就会connect到reg\_n\_login\_accepted()函数：



reg\_n\_login\_accepted()函数首先初步判断输入的id是否合法（输入的id是不是代表整数以及在不在规定的id范围里）

如果id合法，调用Checkusr（）函数：作用是读取登录/注册窗口用户输入的数据，与当前的用户序列进行比对id、passwd，并返回规定好的返回值，交还给引用checkusr（）函数的reg\_n\_login\_accepted()进行对应的处理。返回值如果是3或者4，进行必要的提示之后会返回到登录/注册界面；如果是2，成功进入日历主界面。如果是1，则调用save\_USER（）函数进行新用户的写入操作。



其中用到的QMessageBox是Qt框架中的一个类，用于创建和管理消息框（Message Box）。消息框是一种用于向用户显示消息、提示、警告或错误信息的常见界面元素。

QMessageBox类提供了一些静态函数，用于方便地创建常见类型的消息框，例如：

information()：显示信息消息框，用于向用户显示一般信息。

warning()：显示警告消息框，用于向用户显示警告信息。

question()：显示询问消息框，用于向用户显示一个问题，并等待用户的回答。

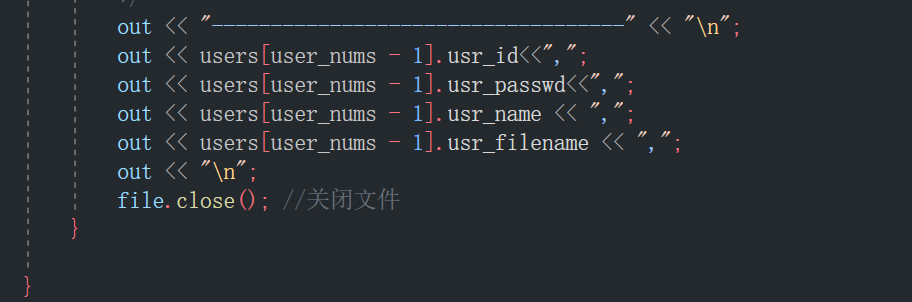
critical()：显示严重错误消息框，用于向用户显示一个严重错误信息。

这些静态函数接受不同的参数，以便设置消息框的标题、文本内容、按钮类型和图标。您可以使用这些函数来快速创建并显示消息框，而无需显式创建 QMessageBox 类的实例。

**Save\_USER():新用户的写入操作：**



User\_nums的值根据用户结构体数组初始化函数从USERS.txt文件中读取的用户数量实时更新，便于新用户的正确写入。



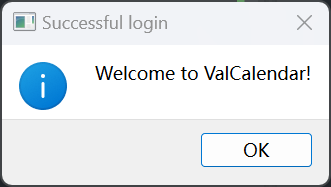
按照用户结构体的顺序，把用户相关的数据字符串以‘,’为分隔符写入USER.txt文件。

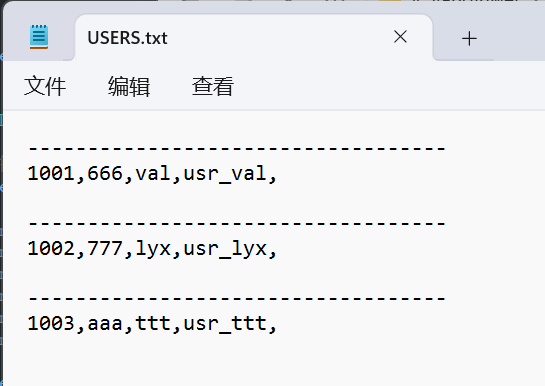
用户结构体数组的初始化函数：init\_users\_structarr()

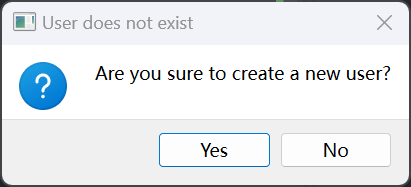


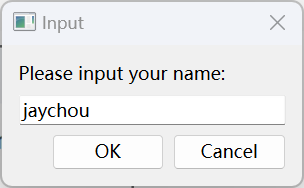
使用split函数以逗号为分隔符切割USERS.txt中的用户数据字段并读入结构体数组中。（后续将根据模块开发进度更新语句）

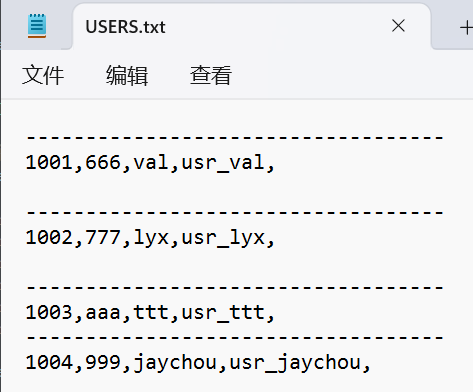
由此，基本实现了用户登录和注册的流程，后续有时间会继续完善注销用户的功能，但是这并不是必要功能，所以先放一边。经过测试，现在可以有效识别已注册用户、有效地注册新用户并输入新用户的昵称。





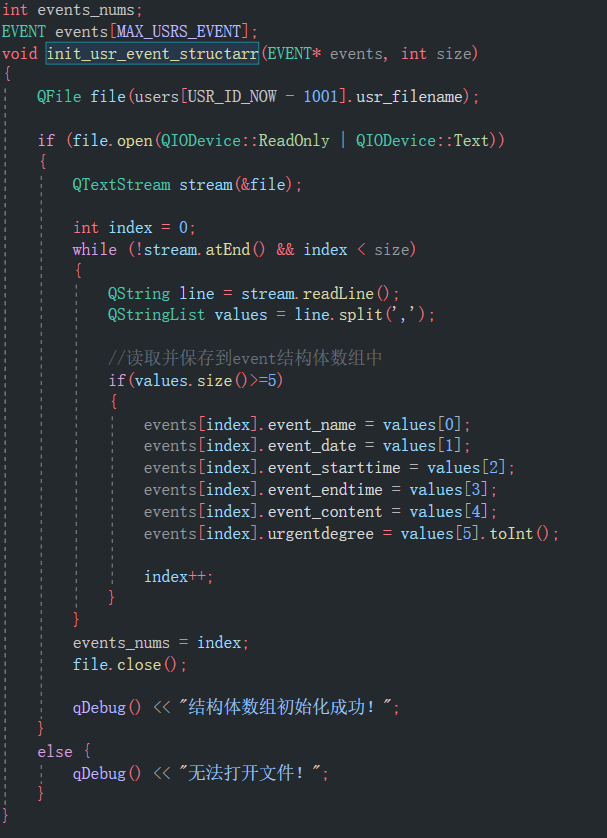


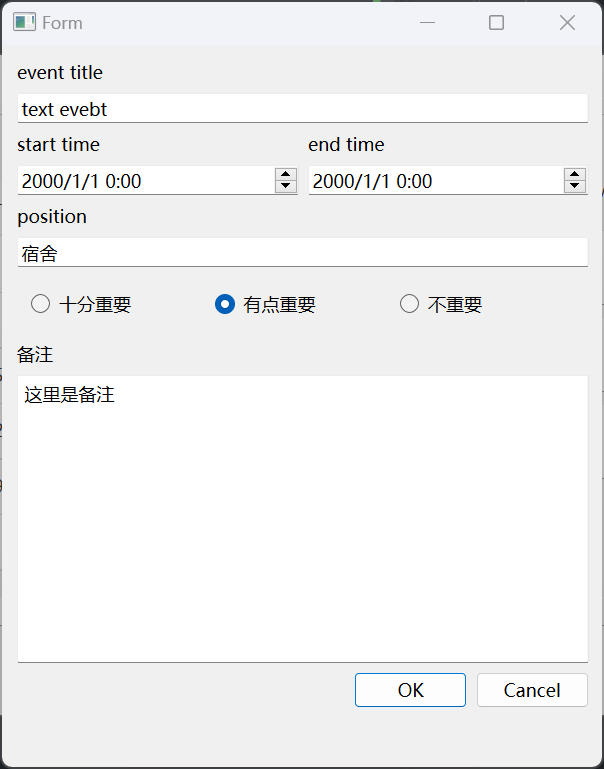


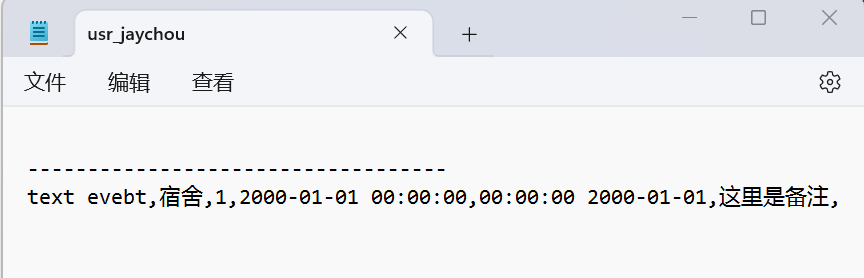


绑定用户账户和对应的日程信息

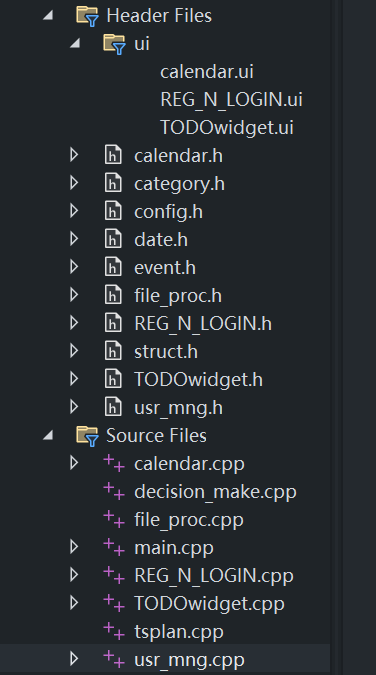
当用户登录之后，会调用void init\_usr\_event\_structarr(EVENT\* events, int size)进行日程信息的初始化：从每个用户专属的.txt文件中读取已经写入的日程信息。







项目文件结构



其中config.h是基础配置文件，放置所有需要调用的库和文件。以及后续需要用到的宏定义。这样我们就拥有了一个可以在项目文件中使用的“万能头“。但是需要注意的是，要关注声明的先后顺序，会影响调用的合法性。

Struct.h文件负责规范地定义各个结构体的成员。如1所示。

Usr\_mng.h则负责定义和存放用户管理所需要的各种数据和函数。如维护用户数量、读取文件初始化用户结构体数组的函数就包含在这里。

需要注意的是，这个初始化函数需要在main函数里调用，以保证登录/注册模块的正常运行。