

清华大学篮球联赛 管理系统

数据库项目实验总结

符亦铭 未央-水木 01 2020012851

周镐源 未央-水木 01 2020012843

一、问题与解决方案

1. 数据库的连接

本次实验中，我们使用 pymysql 库来实现数据库连接。而考虑到每次连接时都要重新输入用户名/密码等信息过于繁琐，因此我们定义了一个连接函数，能够直接返回跟数据库的连接，在代码编写时更为方便简洁。

```
def get_conn():
    user = '' # Here input your own user account
    password = '' # Here input your own user password
    return pymysql.connect(host='127.0.0.1',
                           user=user,
                           password=password,
                           db='final') # Please ensure that your database is named 'final'!
```

2. 比赛结果记录/更改出现的 bug 及修复

在记录比赛结果时，如若之前这场比赛已经完赛并且有相关的比赛结果记录，那么这次操作便会被视作比赛结果更改。但我们在操作时发现会有球队进行完两场比赛后出现例如“2胜1负”的“虚空”战绩，经检查后发现是因为在更改比赛信息的时候，并未将之前的记录从球队战绩中抹除。因此我们对代码进行修改，当判断是比赛结果更改时，会先扣除之前记录对战绩的影响，再更改赛果。

3. 用户输入异常信息处理

由于很多时候我们都需要根据用户输入的信息对数据库进行操作，而用户一旦输入有误，数据库便会报错从而终止程序。为尽可能避免这种情况的发生，我们在编程中进行输入检查，从而保证用户输入的合法性。例如添加比赛场次时，我们会对于用户输入的信息进行检查，如若错误则会弹出警告窗口。

```
if mno in mno_list:
    print(QMessageBox.warning(self, '错误', '比赛已存在', QMessageBox.Yes, QMessageBox.Yes))
elif court not in court_list:
    print(QMessageBox.warning(self, '错误', '场地不存在', QMessageBox.Yes, QMessageBox.Yes))
elif len(time) != 17:
    print(QMessageBox.warning(self, '错误', '时间输入有误', QMessageBox.Yes, QMessageBox.Yes))
elif h_tno not in tno_list:
    print(QMessageBox.warning(self, '错误', '主队不存在', QMessageBox.Yes, QMessageBox.Yes))
elif g_tno not in tno_list:
    print(QMessageBox.warning(self, '错误', '客队不存在', QMessageBox.Yes, QMessageBox.Yes))
elif h_tno == g_tno:
    print(QMessageBox.warning(self, '错误', '主客队相同', QMessageBox.Yes, QMessageBox.Yes))
else:
    sql = 'SELECT CNo FROM final.court WHERE `CName`=\'' + str(court) + '\'';
    cur.execute(sql)
```

由于篇幅所限，此处仅展示若干警告弹窗实例，其余的输入界面我们也基本上都完成了异常处理问题，以保证程序稳定运行，更多请参见具体程序。



4. UI 文件编辑问题

起初我们通过 QtDesigner 设计 ui 文件后（例如 test.ui），使用 PYUIC 生成对应的 py 文件 test.py。如若我们直接对其进行修改，则再对原 ui 文件通过 QtDesigner 进行修改后，之前的 py 文件也会被覆盖，导致我们的代码被清除。对此，我们根据助教的建议，选择新建一个 test_al.py 文件，再导入 test.py 的内容，从而进行内置函数的编写与定义，同时也能实时更新 ui 文件的更改，保证了 py 文件的独立性，如下图在 utils.py 中导入 ui_design.py 所示。

```

utils.py ×  main.py ×  ui_design.py ×
1  import time
2  import pymysql
3  from PyQt5.QtWidgets import *
4  from PyQt5.QtGui import *
5  from PyQt5.QtCore import *
6  from DataBase_Course.py_ui.ui_design import * # Here input your own route(relative route is OK)
7  _translate = QtCore.QCoreApplication.translate

```

5. 不同界面间的连接问题

在主程序 main.py 中，我们通过引入控制器 Controller 类，来实现不同窗口间的串联，其主要逻辑为：每个窗口类中都有若干个 pyqtSignal，负责实现对应功能的连接。而当触发特定条件后，这些信号便会被激活，从而根据 Controller 类中的逻辑设置，打开/关闭对应的窗口，进而实现界面连接。

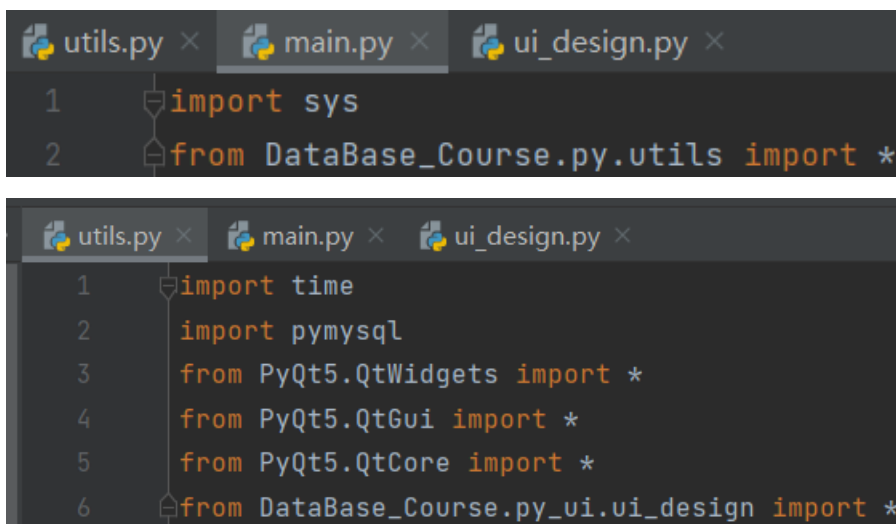
```

# In the Controller, we enable connections between different windows.
# This is realized by launching another target window when the current window's corresponding 'pyqtSignal' is emitted.
# We also need to judge whether to close the current window.
class Controller:
    def __init__(self):
        # Here we store the information of the current user, which will be used later.
        self.account = None
        self.name = None
        self.sex = None
        self.age = 0
        self.state = None

```

6. 文件整合问题

由于之前设计的时候，每个窗口均需要一个 py 文件和 _al.py 文件，而功能又比较多，因此最终共有数十个代码文件，显得较为繁琐。因此在最后，我们将文件进行整合，将所有 ui 文件生成的 py 文件合并进 ui_design.py 中，将所有 _al.py 文件合并进 utils.py 中，并通过导入 ui_design.py 的全部内容（之前也有所提及）来获取之前定义的 ui 界面，同时在 main.py 中导入这两个 py 文件，从而通过主程序来实现总体逻辑控制。



```

1 import sys
2 from DataBase_Course.py.utils import *

utils.py × main.py × ui_design.py ×

1 import time
2 import pymysql
3 from PyQt5.QtWidgets import *
4 from PyQt5.QtGui import *
5 from PyQt5.QtCore import *
6 from DataBase_Course.py_ui.ui_design import *

```

二、收获与心得体会

完成此次的数据库项目对我们来说是具有一定挑战性的考验，也是一次很好的锻炼机会。在课堂上，老师由浅入深地讲解了数据库的基本概念、设计流程、执行机制等等，让我们对数据库的设计有了整体的认知；同时，助教对于 PyQt 和 QtDesigner 的讲解，带领我们初步熟悉了如何设计界面，以及通过 python 语句实现前端界面与后端数据库之间的交互。基于这些基础知识的储备，我们开始推动项目从构思阶段向着落地实现慢慢前进。

在这个过程中，有许多系统的功能是需要我们不断摸索，寻找合适的解决方案来实现的，这不仅锻炼了我们的编程能力，更让我们深刻体会到在一个接近真实的项目中，需要怎样的全局意识、逻辑框架与协作配合，才能使一套系统最终完善地呈现在用户面前，从使用者的角度出发发现问题，再从设计者的角度出发，选择最优的方式解决问题，是一件在项目中贯穿始终的任务。

我们项目的出发点，是为校内的篮球联赛设计出一个交互系统，服务于观众、球队与管理员，也是服务于校园体育事业的发展，弘扬“无体育，不清华”的新时代体育精神。因此，我们试着借助这次大实验的机会，在一个真实的应用背景下设计我们的系统，使其能尽可能满足使用者的大部分真实需求。当项目最终完成，每一项功能都得以实现，顺利地和老师同学面前完成演示时，我们心中的收获感与成就感也是喜悦而实在的。

在此我们要感谢老师与助教的悉心教导与帮助，从这次项目出发，希望在未来，我们能够将这门课的所学所得运用于更广阔的领域！