## 系统整体架构的描述

前端

UI 界面：

***##在这插入一张主窗口的图片***

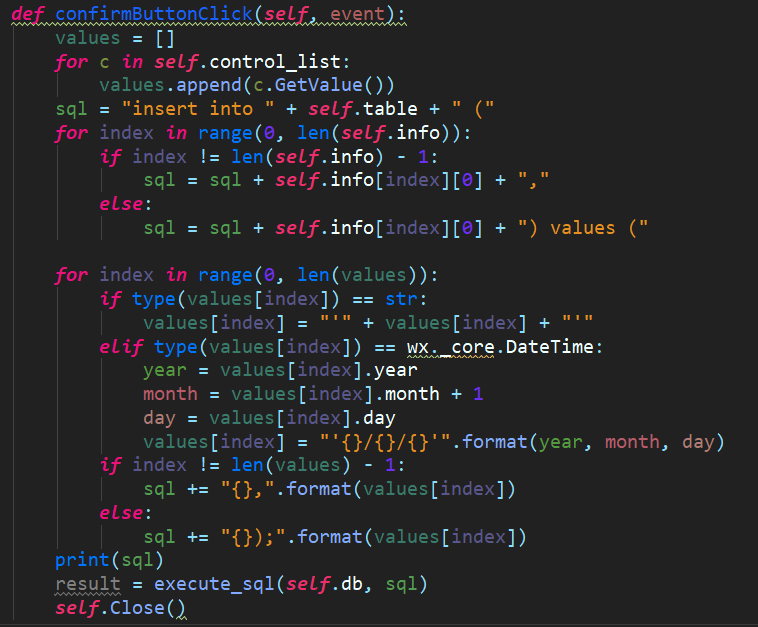
1. 程序的UI界面主要使用wxpython进行搭建。使用wxpython可以实现一次编写多平台运行的效果，使得程序更具可移植性。
2. mainFormTemplate文件中的mainFrame为主程序窗口类的父类，其定义了各个控件的位置以及可操作控件的事件绑定。mainForm文件中的mainForm类继承于mainFrame，在mainFrame原有的布局上对其绑定方法进行重写，从而实现各个控件的具体功能。
3. insert文件中insertForm类为所有插入，删除，更新table的相关窗口的父类。其描述了这一类窗口的一个范式，使得该窗口可以随着table的结构的不同而做出相应的变化，这样可以不需要为每一个table的修改而另外制作一个窗口，具有较强的可适应性，代码重用率高。在生成一个insert窗口前，首先执行sql命令desc tablename，然后根据返回的table的结构来在窗口中放置不同的label及其它对应的数据输入控件（例如，输入的是还书日期，那么对应的位置就放置一个datepicker，如果是输入书名则放置一个textctrl），从而构建窗口。

操作转换：

用户使用图书管理系统在根本上无非就是对数据库进行增删改查4种操作，如何将用户的键鼠操作转换成对应的sql语句是这个图书管理系统的核心。

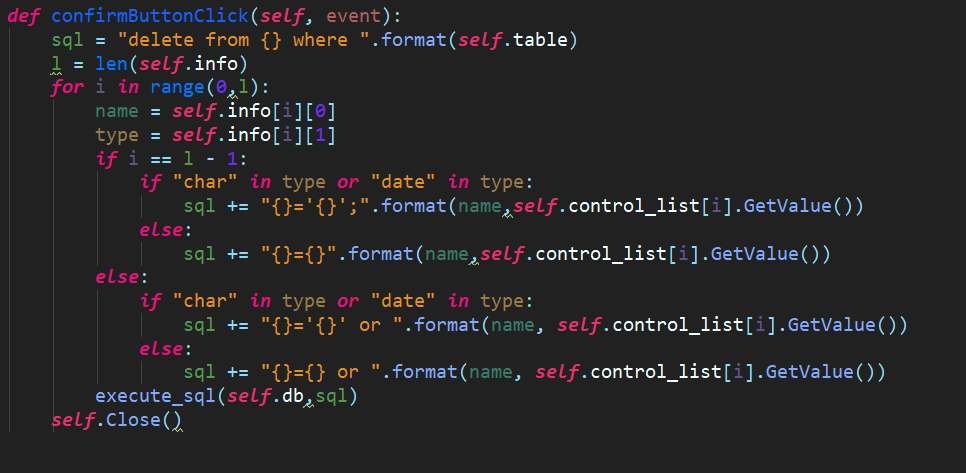
1. 插入操作。该操作主要通过弹出insert窗口，让用户通过填写窗口中的输入框来获取需要插入的values。通过字符串的拼接”insert into” + 表名 + “values (“ + 从窗口获取的值 + “);”最终将用户的操作转换成sql指令。涉及的模块包括图书入库，图书证的添加，管理员的注册，借书。

***##在这插一张插入窗口的图片（最好每一个窗口都有一张）***



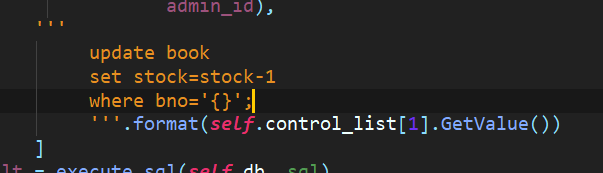
**插入操作相关代码示例**

1. 删除操作。该操作通过弹出删除窗口，用户通过填写相关的条件，进行条件匹配来对符合条件的row进行删除。通过字符串拼接”delete from “ + 表名 + “ where ” + 条件来实现删除操作的sql指令转换。涉及的模块包括图书的删除，图书卡的删除，管理员的删除。



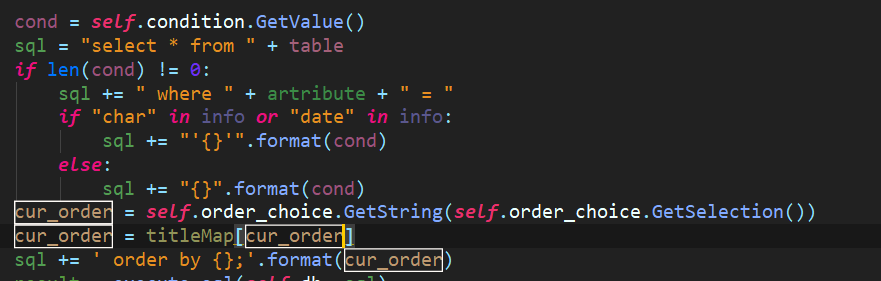
**删除操作代码示例**

1. 更改操作。更改操作主要是在借书与还书操作的时候进行，借书需要对book表中的库存进行更新，还书操作需要对book表中的库存与borrow表中的还书日期进行更新。通过字符串拼接”update “ + 表名 + “ set ” + “属性 = 值”来实现更改操作的sql指令转换。涉及模块借书、还书。



**更新操作代码示例**

1. 查询操作。查询操作是用户经常使用的一个功能，用户通过选择想要查询的表，选择、输入想要搜索的条件，选择结果的排序方式，从而实现信息的查询。通过字符串拼接”select \* form“ + 表名 +”where” + 条件 + “order by” + 属性来实现更改操作的sql指令转换。涉及模块图书查询，借书卡查询，借书记录查询。

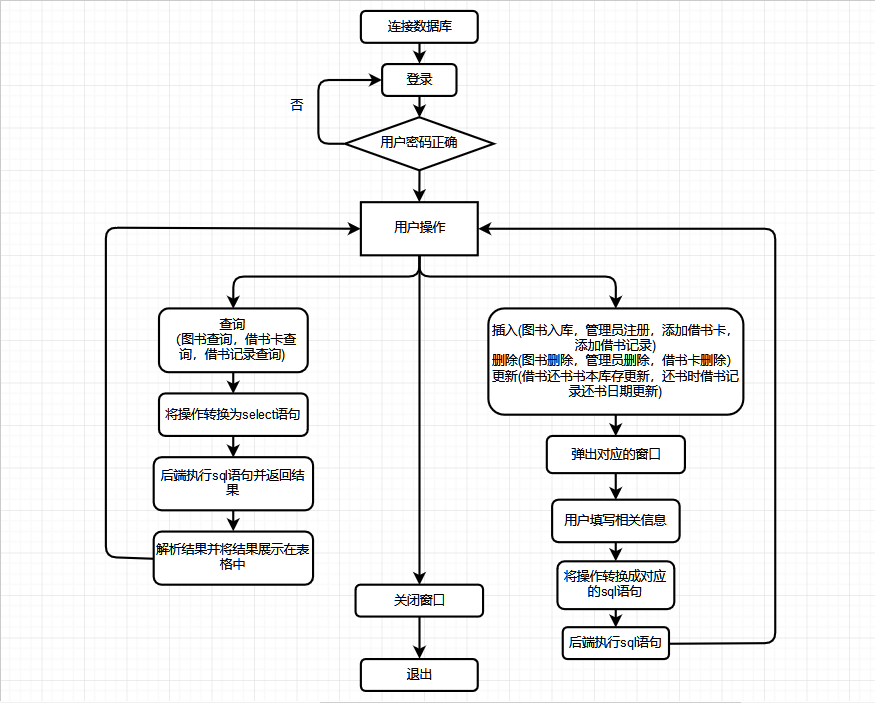


**查询代码示例**

后端：

1. 使用pymysql 进行 Python 与 MySQL 之间的通讯。在execute\_sql.py文件中定义了execute\_single\_sql与execute\_sql两个函数，其中execute\_single\_sql实现了执行一条 sql 语句的任务，而execute\_sql实现了将一系列 sql 语句作为原子进行执行，既做到了一次函数调用执行多条 sql 语句，又实现了当有语句执行出现错误时，同时回滚，取消整个操作。execute\_sql.py文件对外提供execute\_sql函数作为 Python 与 MySQL 之间的接口。
2. 在 function.py文件中，根据不同功能定义了多个函数，每个函数根据自己所执行的功能将所给参数解析成 sql 语句交给execute\_sql执行。（程序最开始使用命令行交互，后期使用wx 模块提供的 UI 界面，故 function.py文件中残存有大量之前使用的函数，这些函数要么经过修改后直接被 UI界面所调用要么通过复制代码段的形式直接插入了按钮所绑定的事件中，主要逻辑基本不变。）

## 模块流程图



## 数据库表结构

setup.py文件是第一次运行程序前必须运行的文件，它提供了初始化程序所需数据库的功能。若连接到的数据库中含有 library 数据库，会先进行 drop 操作，因此每次运行 setup.py都会得到全空的数据库 library。

具体的表格结构在 setup.py中也有定义。主要是使用了四张表格：book，card，administrator和 borrow。book 用于存储书籍信息，拥有主键 bno（书籍编号）其他属性均可设置为 null。card 用于存储借书证信息，拥有主键 cno（借书证编号）其他属性均可设置为 null。administrator用于存储管理员信息，拥有主键 ID（管理员 ID），除了 passwd（管理员密码）之外的其他属性均可设置为 null。borrow用于存储借还书信息，由 bno，cno，borrow\_date 共同组成主键，并且 bno，cno，administrator\_ID 分别是引用 book，card，administrator的外键。只有 return\_date可以设置为 null

## 程序所采用的各项技术

Pymysql 提供 Python 与 MySQL 的交互操作，wx 提供 Python 的 UI 界面

## 各成员的任务描述

郑文浩：execute\_sql.py、setup.py与 function.py主要是 sql 语句的设计与执行，最终报告

钟靖阳：UI 界面、用户操作到sql语句的转换