## 系统整体架构的描述

前端 UI 界面：

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