

“银行业务管理系统”

系统设计与实现报告

姓名：徐语林

学号：JL19110004

计算机科学与技术学院

中国科学技术大学

2021 年 6 月

目 录

1	概述
1.1	系统目标
1.2	需求说明
1.3	本报告的主要贡献
2	总体设计
2.1	系统模块结构
2.2	系统工作流程
2.3	数据库设计
3	详细设计
3.1	客户管理模块
3.2	账户管理模块
3.3	贷款管理模块
3.4	业务统计模块
4	实现与测试
4.1	实现结果
4.2	测试结果
4.3	实现中的难点问题及解决
5	总结与讨论

1 概述

1.1 系统目标

某银行准备开发一个银行业务管理系统，该银行的数据需求如下：

银行有多个支行。各个支行位于某个城市，每个支行有唯一的名字。银行要监控每个支行的资产。银行的客户通过其身份证号来标识。银行存储每个客户的姓名、联系电话以及家庭住址。为了安全起见，银行还要求客户提供一位联系人的信息，包括联系人姓名、手机号、Email 以及与客户的关系。客户可以有帐户，并且可以贷款。客户可能和某个银行员工发生联系，该员工是此客户的贷款负责人或银行帐户负责人。银行员工也通过身份证号来标识。员工分为部门经理和普通员工，每个部门经理都负责领导其所在部门的员工，并且每个员工只允许在一个部门内工作。每个支行的管理机构存储每个员工的姓名、电话号码、家庭地址、所在的部门号、部门名称、部门类型及部门经理的身份证号。银行还需知道每个员工开始工作的日期，由此日期可以推知员工的雇佣期。银行提供两类帐户——储蓄帐户和支票帐户。帐户可以由多个客户所共有，一个客户也可开设多个账户，但在一个支行内最多只能开设一个储蓄账户和一个支票账户。每个帐户被赋以唯一的帐户号。银行记录每个帐户的余额、开户日期、开户的支行名以及每个帐户所有者访问该帐户的最近日期。另外，每个储蓄帐户有利率和货币类型，且每个支票帐户有透支额。每笔贷款由某个分支机构发放，能被一个或多个客户所共有。每笔贷款用唯一的贷款号标识。银行需要知道每笔贷款所贷金额以及逐次支付的情况（银行将贷款分几次付给客户）。虽然贷款号不能唯一标识银行所有为贷款所付的款项，但可以唯一标识为某贷款所付的款项。对每次的付款需要记录日期和金额。

1.2 需求说明

1. 客户管理：提供客户所有信息的增、删、改、查功能；如果客户存在着关联账户或者贷款记录，则不允许删除；
2. 账户管理：提供账户开户、销户、修改、查询功能，包括储蓄账户和支票账户；账户号不允许修改；
3. 贷款管理：提供贷款信息的增、删、查功能，提供贷款发放功能；贷款信息一旦添加成功后不允许修改；要求能查询每笔贷款的当前状态（未开始发放、发放中、已全部发放）；处于发放中状态的贷款记录不允许删除；
4. 业务统计：按业务分类（储蓄、贷款）和时间（月、季、年）统计各个支行的业务总金额和用户数，统计的结果以表格形式展示。

1.3 本报告的主要贡献

本报告主要是阐述整个银行系统的完成过程，自己对于本实验的理解，以及对于实验过程中所碰到的坑的解决。

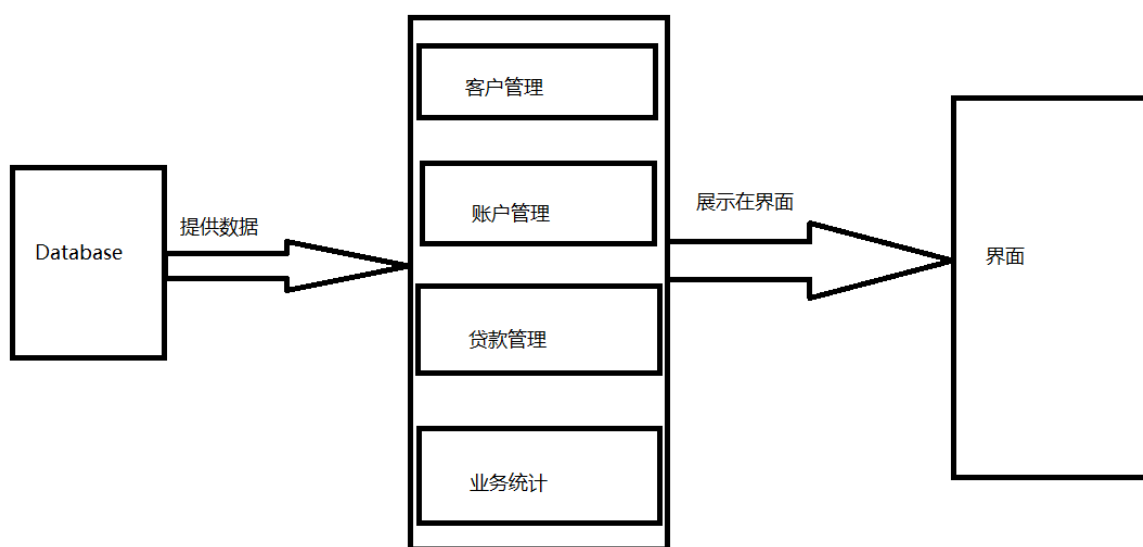
同时，由于本实验接触的实验工具除去 mysql 后全是全新的东西，本报告也有一定的对于 pyqt 以及前端与后端交互的介绍性作用，具有一定的入门意义。

2 总体设计

2.1 系统模块结构

1.本实验采用的 pyqt5+mysql+pymysql 来实现的，分别有客户管理，账户管理，贷款管理，业务统计四个功能。四个部分相互独立。属于 CS 架构。

2.设计图如下：

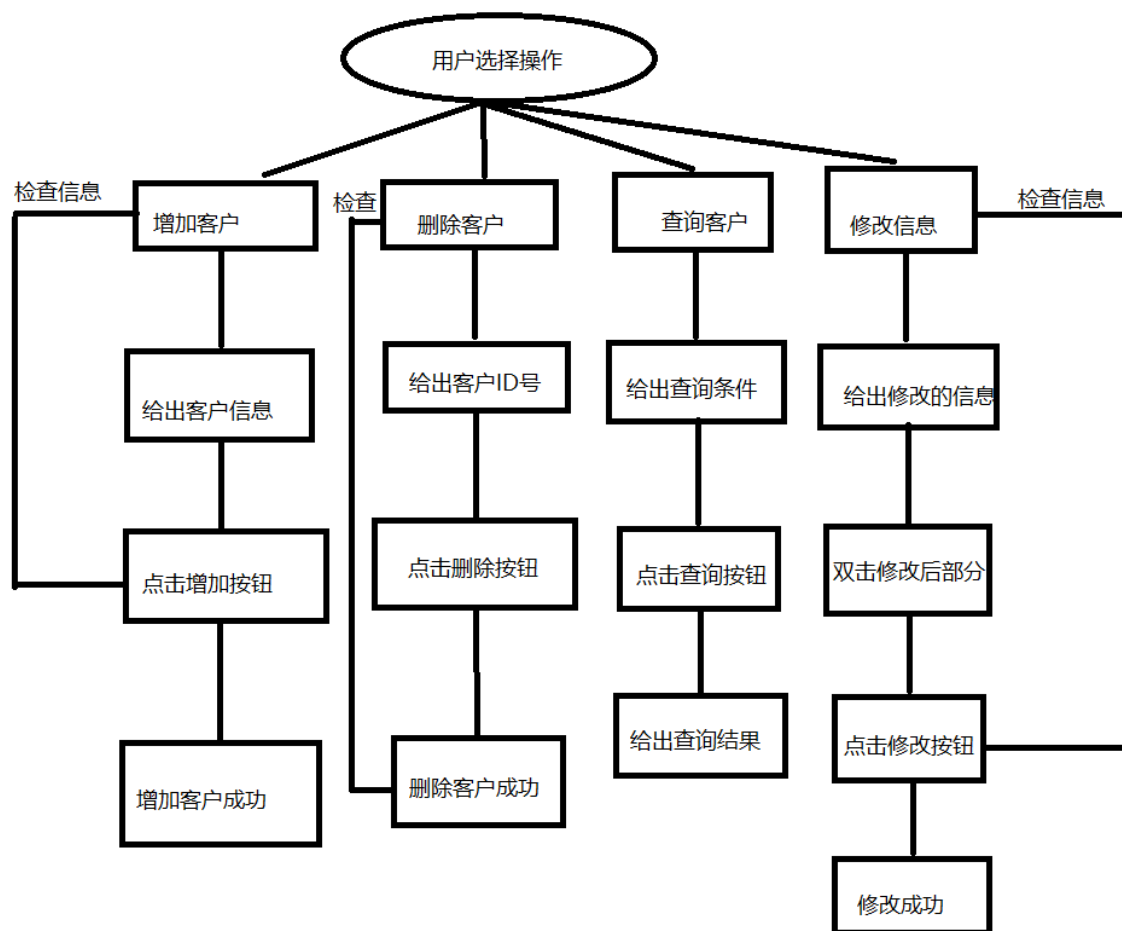


3.接口部分的设计是用一个 `get_database` 这样一个函数来进行数据库操作语句的接收与反馈。

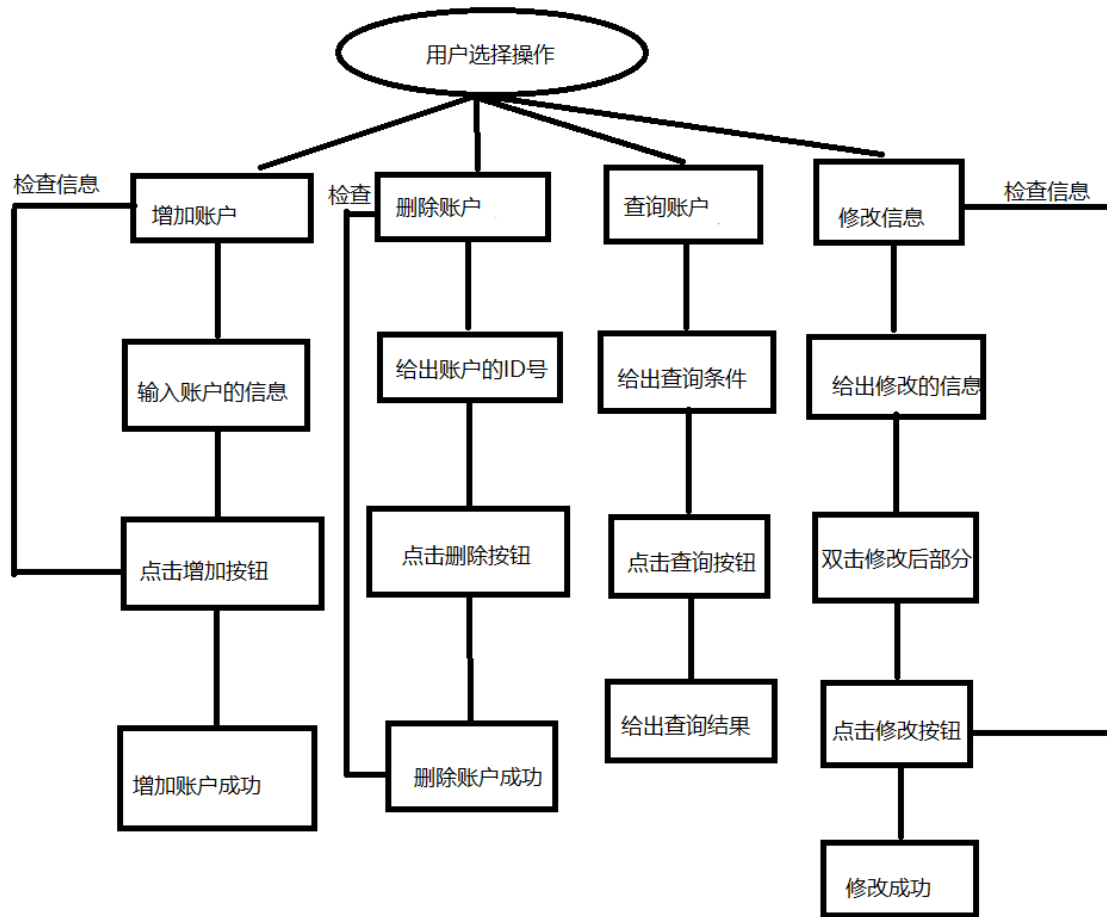
4.界面的设计是用的 `qt designer` 工具，再将 `ui` 文件转换为 `py` 文件即可。界面与数据库的交互部分主要是依据用户的需求，响应用户的操作。例如用户选择客户管理中的查询操作，便给查询操作在界面设置一个按钮，一旦触发此事件，便进行数据库的操作，最后通过一个自己封装的 `show_treeWidget` 函数反馈到前端当中。

2.2 系统工作流程

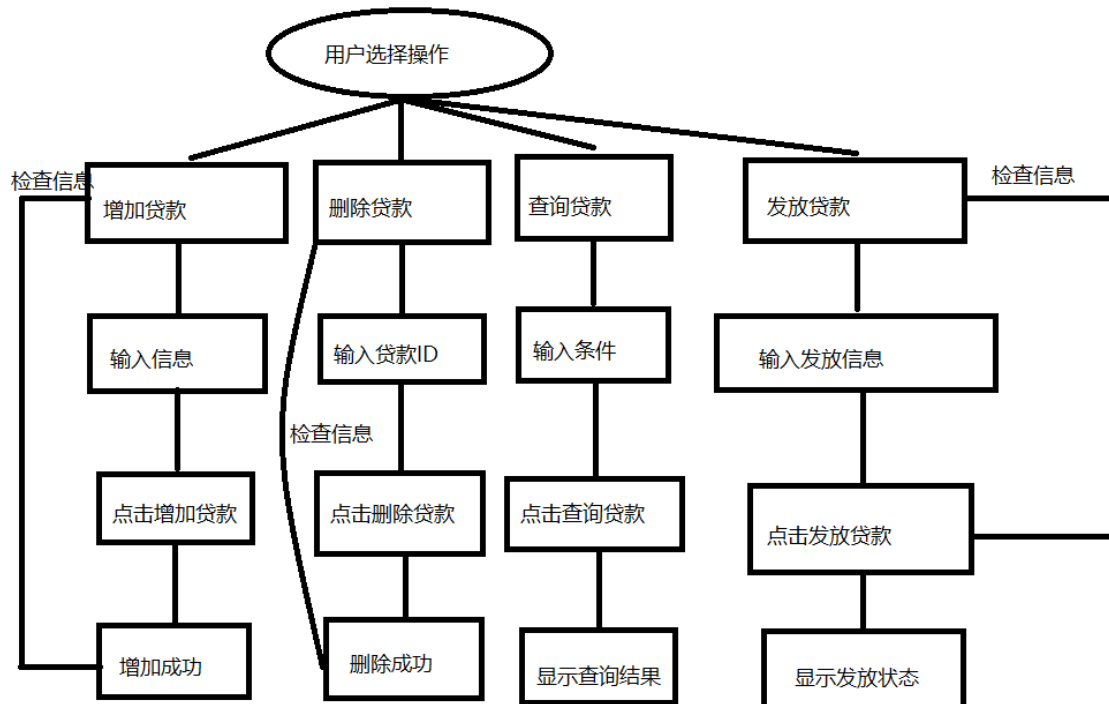
1.客户管理：



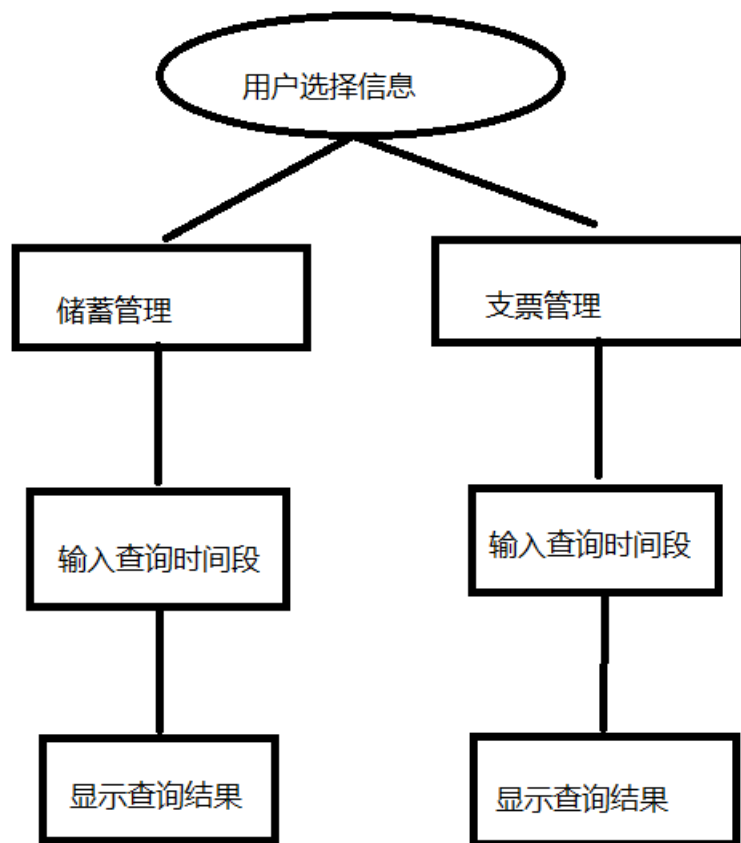
2. 账户管理: (由于储蓄账户和支票账户的工作流程相同, 只是用 combobox 让用户进行了一个选择, 因此此部分展示的一个大类账户)



3. 贷款管理:

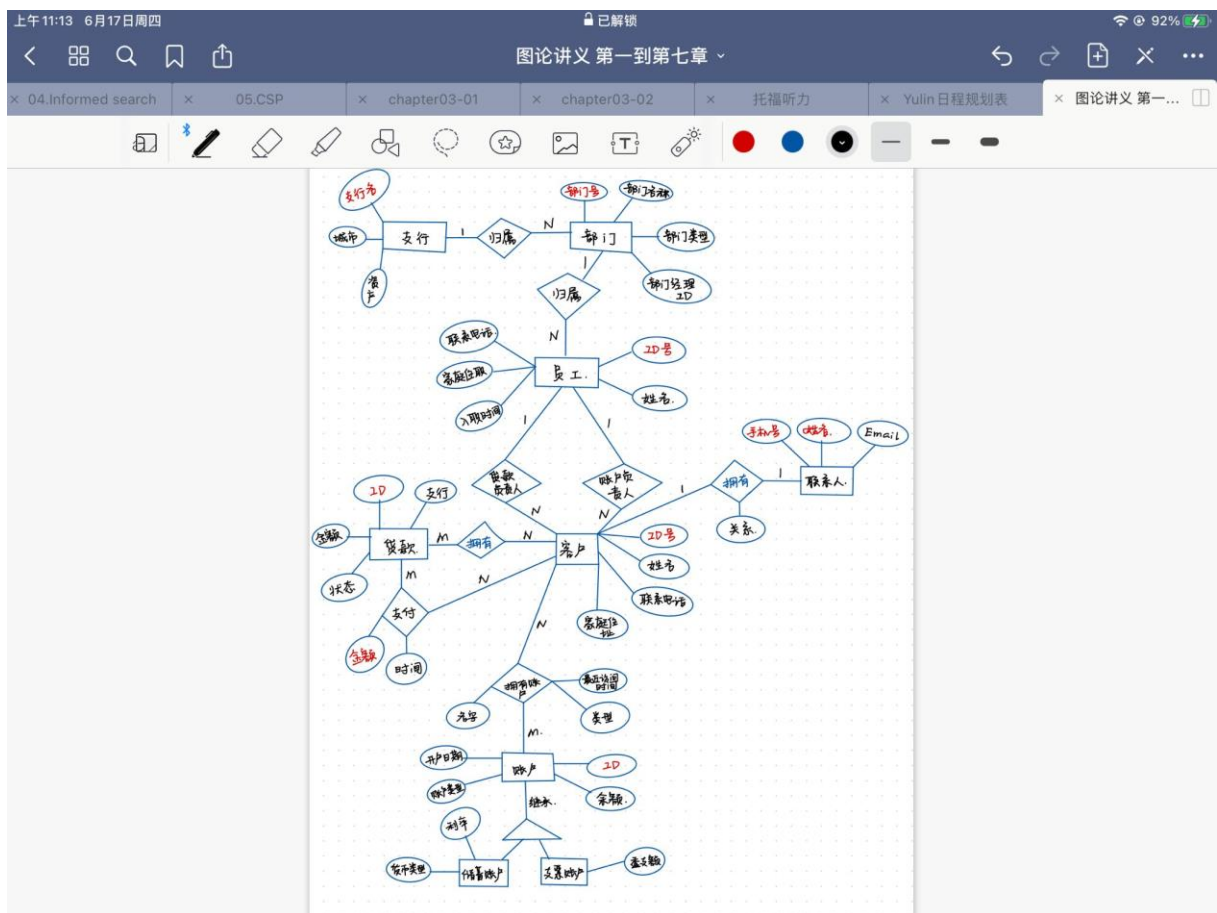


4. 业务统计:

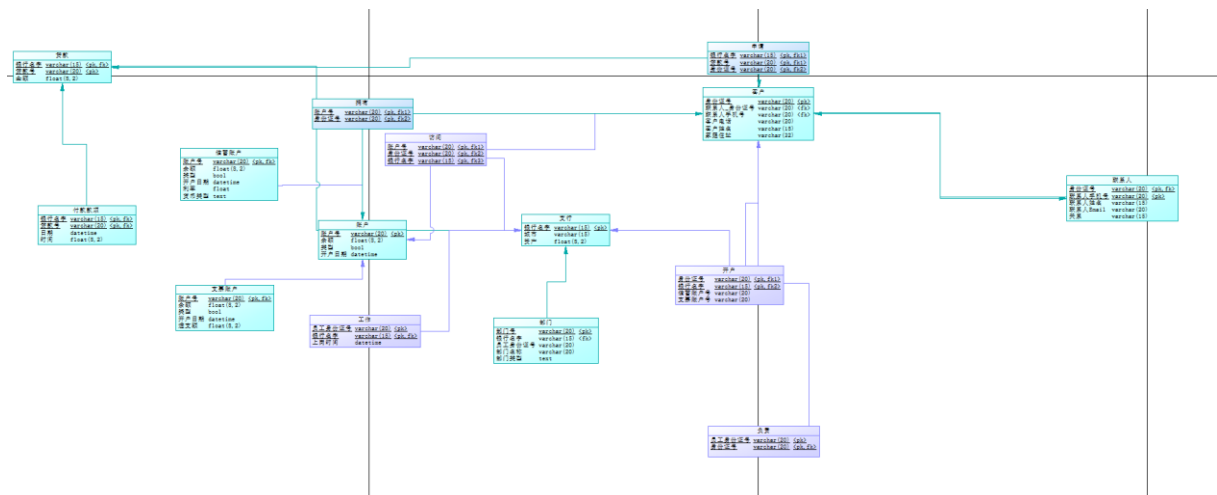


2.3 数据库设计

ER 图:



逻辑数据库结构:



数据库的设计:

注: 由于部门在 lab3 并不是一个主要的部分, 因此本实验将部门的部分做了一个省略
支行:


```
create table 支行
(
    城市          varchar(50)          not null,
    名字          varchar(50)          not null,
    资产          DECIMAL(8,2)         not null,
    constraint PK_支行 primary key (名字)
);
```

员工:

```
create table 员工
(
    员工身份证号    varchar(20)          not null,
    负责人身份证号  varchar(20),
    员工姓名        varchar(50),
    员工电话号码    varchar(50),
    员工家庭住址    varchar(50),
    开始工作日期    DATE,
    constraint PK_员工 primary key (员工身份证号)
);
```

客户:

```
create table 客户
(
    身份证号          varchar(20)          not null,
    员工身份证号      varchar(20),
    客户姓名          varchar(50),
    客户联系电话      varchar(50),
    客户家庭住址      varchar(50),
    联系人姓名        varchar(50)          not null,
    联系人手机号      varchar(50),
    联系人Email       varchar(50),
    联系人与客户关系  varchar(50),
    负责人类型        varchar(50),
    constraint PK_客户 primary key (身份证号)
);
```

储蓄开户:

```
create table 储蓄开户
(
    身份证号          varchar(20)          not null,
    名字              varchar(20)          not null,
    储蓄账户号        INTEGER              not null,
    储蓄账户开户日期    DATE,
    储蓄账户余额        DECIMAL(8,2),
    货币类型          varchar(50),
    利率              DECIMAL(8,2),
    最近访问日期        DATE,
    constraint PK_储蓄开户 primary key (身份证号, 名字, 储蓄账户号)
);
```

支票开户:

```
create table 支票开户
(
    身份证号          varchar(20)          not null,
    名字              varchar(20)          not null,
    支票账户号        INTEGER              not null,
    支票账户开户日期    DATE,
    支票账户余额        DECIMAL(8,2),
    透支额            DECIMAL(8,2),
    贷款最近访问日期    DATE,
    constraint PK_支票开户 primary key (身份证号, 名字, 支票账户号)
);
```

逐次支付:

```
create table 逐次支付
(
    贷款号            INTEGER              not null,
    身份证号          varchar(20)          not null,
    发放金额          DECIMAL(8,2),
    日期付款          DATE,
    constraint PK_逐次支付 primary key (贷款号, 身份证号)
);
```

贷款:

```
create table 贷款
(
    贷款金额          DECIMAL(8,2),
    贷款号            INTEGER          not null,
    身份证号          varchar(20)      not null,
    名字              varchar(20),
    constraint PK_贷款 primary key (贷款号,身份证号)
);
```

3 详细设计

3.1 客户管理模块

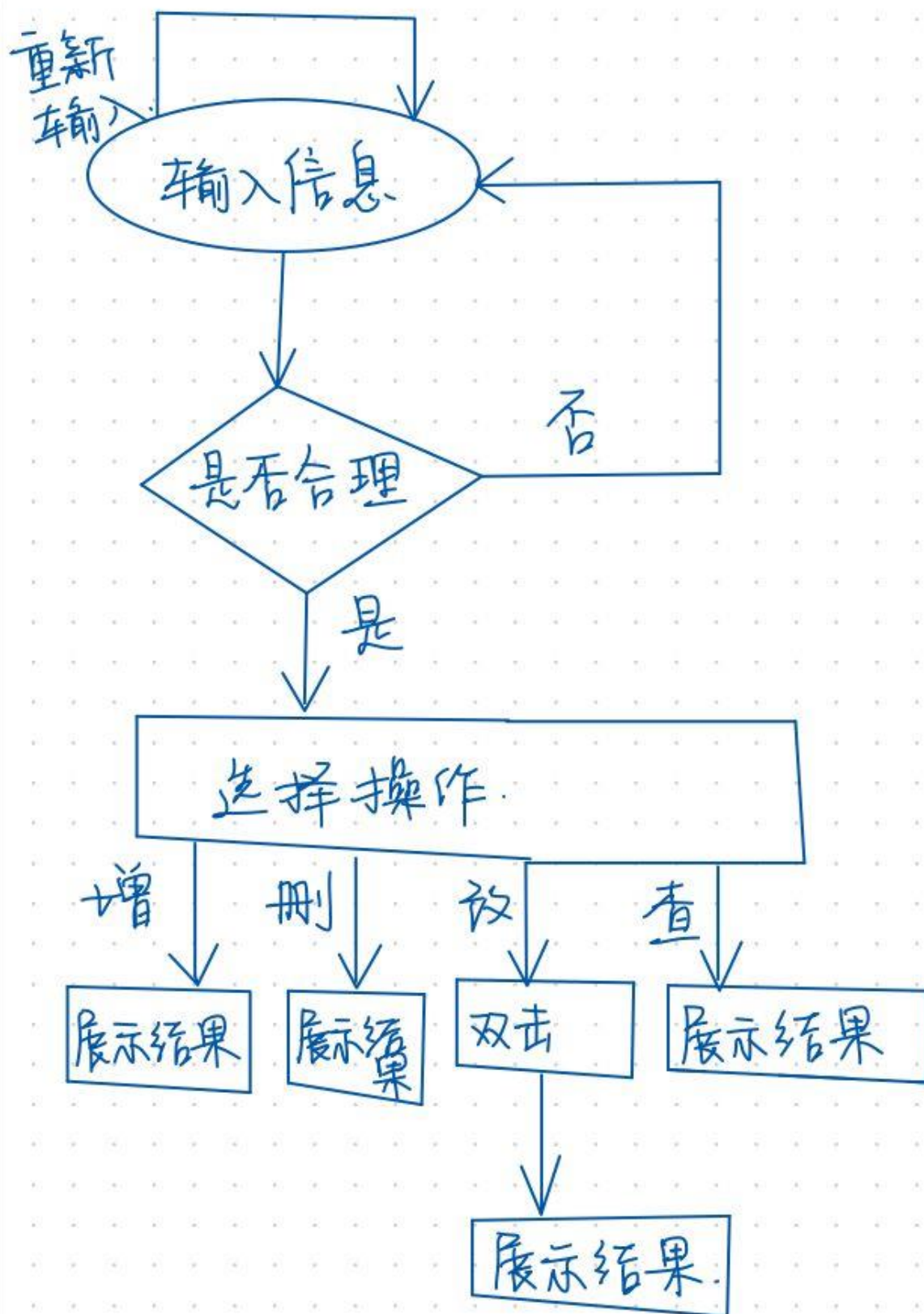
输入:

身份证号(必填)
联系电话
联系人手机号
管理人身份证号(必填)
家庭住址
联系人邮箱
姓名(必填)
联系人姓名
与联系人关系
选择的操作类型
重新输入

输出:

增加客户, 则将插入的信息插入到数据库中
删除客户, 则将选择删除的信息在数据库中删除
查询信息, 则将选择查询的信息显示出来
修改信息, 则将选择修改的信息在数据库中修改
重新输入, 则将之前输入的清空
(增加删除和修改操作均可以通过查询来进行验证正确性)

程序流程图:



3.2 账户管理模块

输入：

账户号(必填)

开户行(必填)

身份证号(必填)

开户日期

余额

账户类型

利率

透支额

货币类型

利率

选择的操作类型

重新输入

输出：

增加账户，则在数据库中增加新账户的信息

删除账户，则将删除的账户信息在数据库中删除

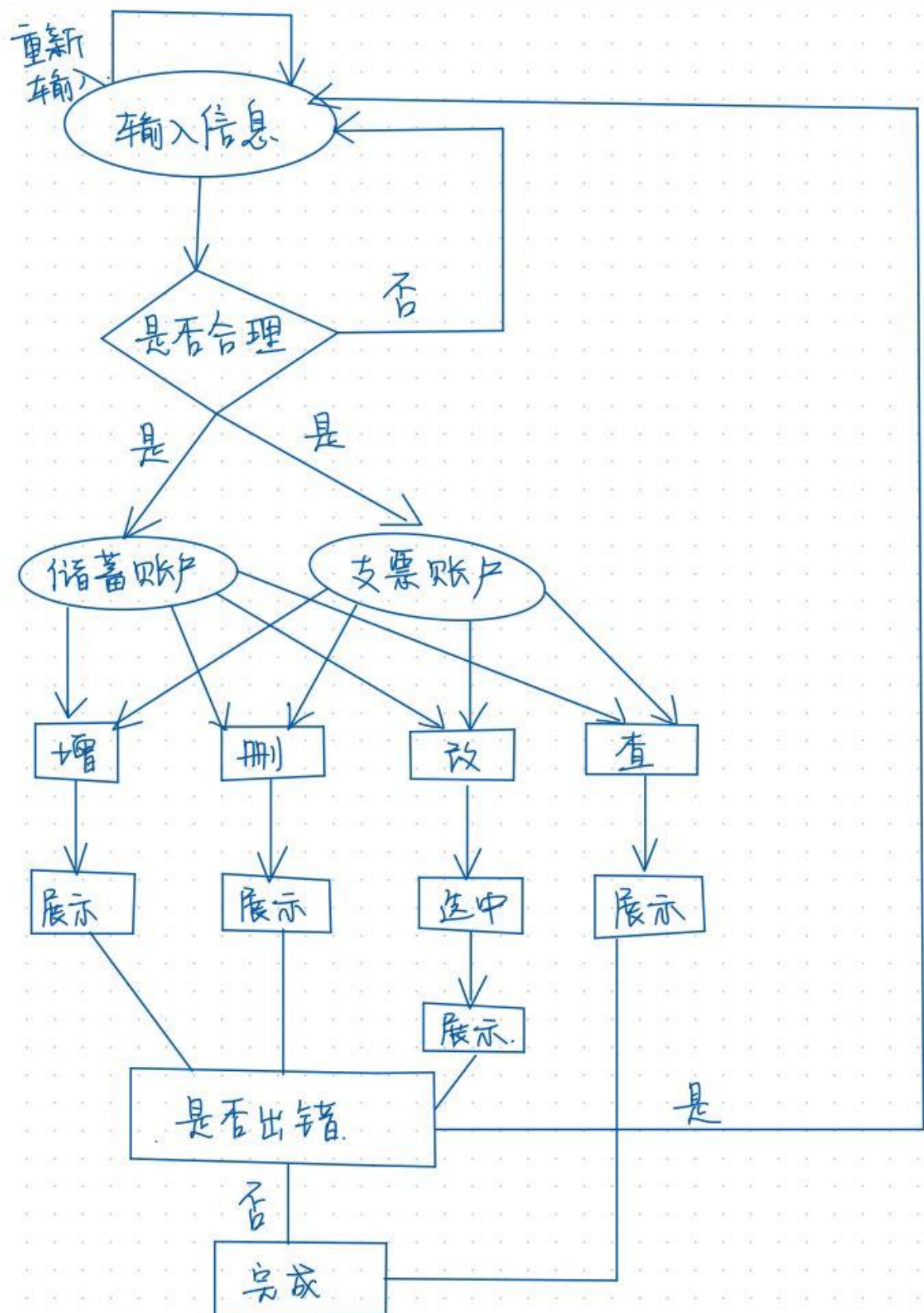
查询账户，查询所满足条件的账户显示出来

修改账户，则在数据库中对此条信息进行修改

重新输入，则将之前输入的清空

(可通过查询信息来进行增加，删除修改的正确性检验)

程序流程图：



3.3 贷款管理模块

输入:

贷款号(增加, 删除, 查询, 发放贷款)

名字(增加, 删除, 查询)

金额(增加, 删除, 查询)

身份证号(增加, 删除, 查询, 发放贷款)

发放金额(发放贷款, 查询)

选择的操作类型

重新输入

输出:

增加贷款, 则在数据库中增加新贷款的信息

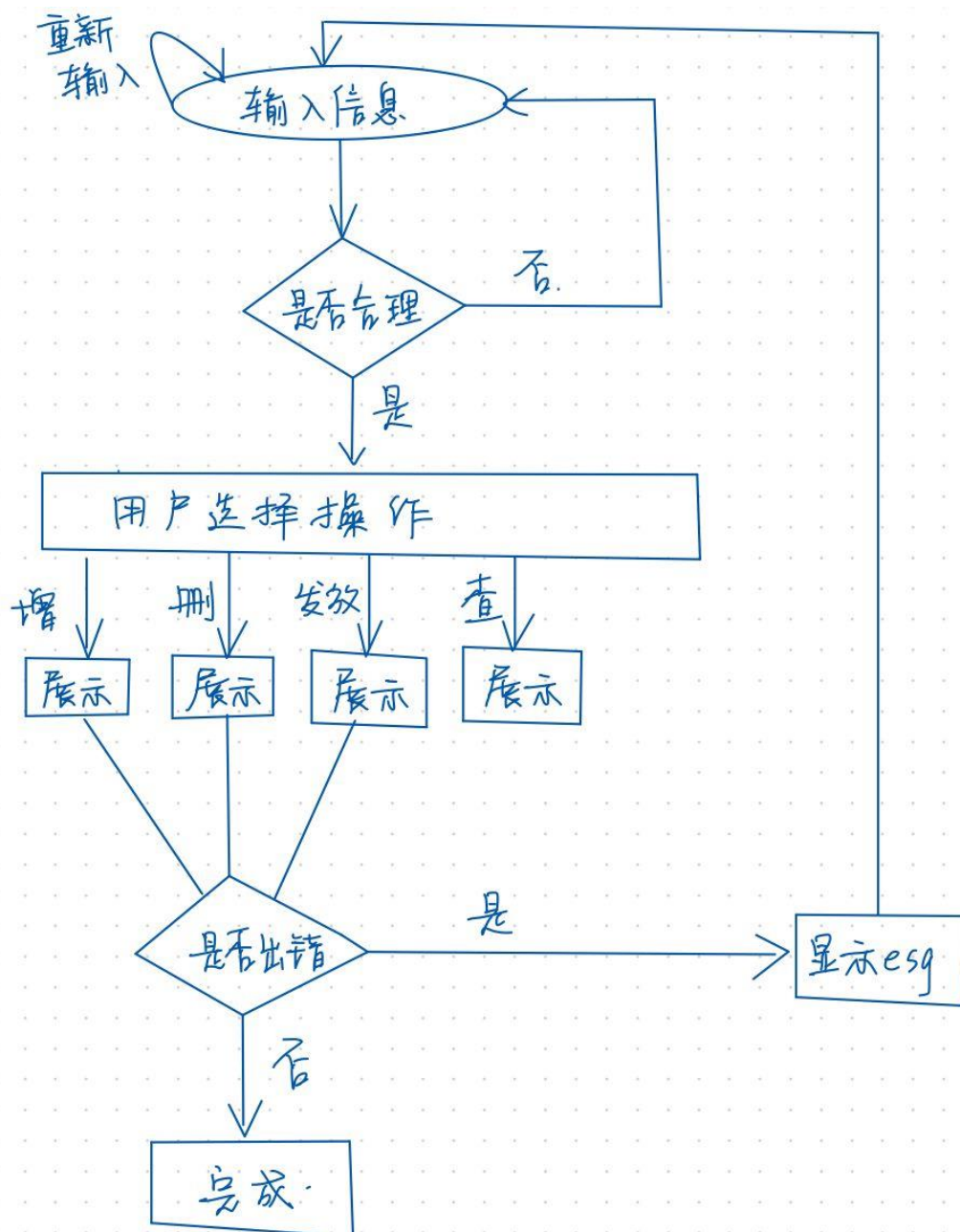
删除贷款, 则在数据库中对进行删除

发放贷款, 改变此贷款的贷款状态, 并在表格中打印出来

查询贷款, 则将满足条件的信息显示出来

重新输入, 则将之前输入的清空

程序流程图:



3.4 业务统计模块

输入:

时间段

选择的查询类型

选择的操作

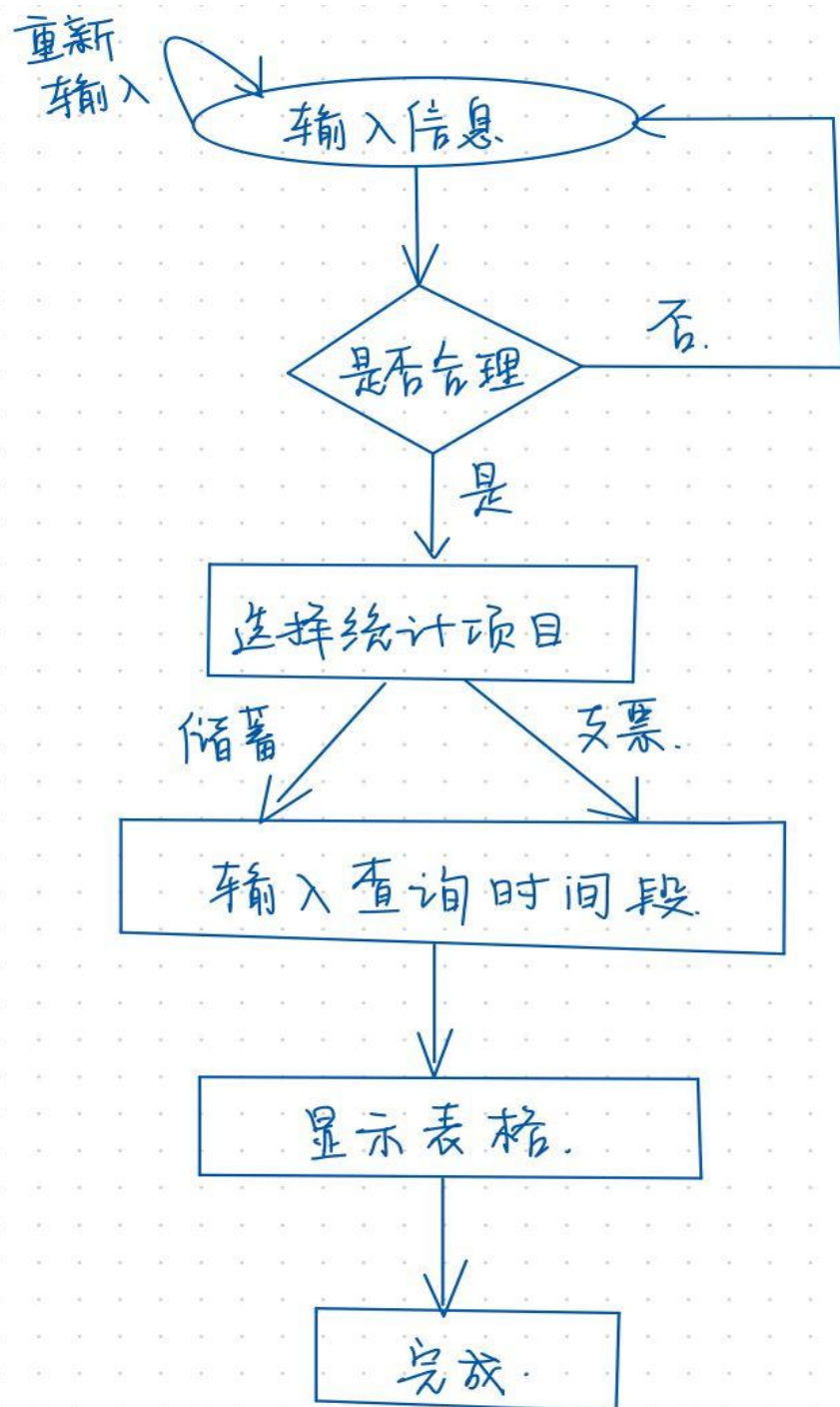
重新输入

输出:

给定时间段的账户统计

重新输入, 则将之前输入的清空

程序流程图：



4 实现与测试

4.1 实现结果


MainWindow

身份证号: 管理人身份证号: 姓名:

联系电话: 家庭住址: 联系人姓名:

联系人手机号: 联系人邮箱: 与联系人关系:

选择所进行的操作:

身份证号	管理人身份证号	姓名	联系电话	家庭住址	联系人姓名	联系人手机号	联系人Email	与联系人关
								



MainWindow

账户号: 开户行: 身份证号:

开户日期: 余额: 账户类型:

利率: 透支额: 货币类型:

选择要进行的操作:

1	
	



MainWindow

贷款号: 名字:

金额: 身份证号:

发放金额:

选择所要进行的操作:

贷款号	金额	名字	发放状态
			

MainWindow

储蓄统计：

选择合法的时间段：

2000/1/1 到 2000/1/1

统计 重新输入

支行名	客户数	总金额
-----	-----	-----

贷款统计：

选择合法的时间段：

2000/1/1 到 2000/1/1

统计 重新输入

支行名	客户数	总金额
-----	-----	-----

4.2 测试结果

客户管理部分：

增：

MainWindow

身份证号: 管理人身份证号: 姓名:

联系电话: 家庭住址: 联系人姓名:

联系人手机号: 联系人邮箱: 与联系人关系:

选择所进行的操作:

身份证号	管理人身份证号	姓名	联系电话	家庭住址	联系人姓名	联系人手机号	联系人Email	与联系人关
117	006	周知方						

删: (删除上图 117)

MainWindow

身份证号: 管理人身份证号: 姓名:

联系电话: 家庭住址: 联系人姓名:

联系人手机号: 联系人邮箱: 与联系人关系:

选择所进行的操作:

身份证号	管理人身份证号	姓名	联系电话	家庭住址	联系人姓名	联系人手机号	联系人Email	与联系人关
111	001	刘诗诗	5056	安徽省合肥市	吴奇隆	189	lss@ustc.edu.cn	夫妻
112	002	王菊	5057	上海市	李磊	190	wj@mail.ustc...	夫妻
113	003	胡莱	5058	北京市	刘浩	191	hx@1ustc.edu...	兄弟
114	004	张梅	5059	浙江省杭州市	肖然	192	zm@yohu.com	兄妹
115	005	李诺	5060	广东省广州市	胡鹏	193	xn@163.com	夫妻
116	006	刘奎	5061	四川省成都市	刘欢	194	ly@qq.com	姐妹

改：(116 的姐妹改为夫妻)

MainWindow

身份证号: 管理人身份证号: 姓名:

联系电话: 家庭住址: 联系人姓名:

联系人手机号: 联系人邮箱: 与联系人关系:

选择所进行的操作:

身份证号	管理人身份证号	姓名	联系电话	家庭住址	联系人姓名	联系人手机号	联系人Email	与联系人关系
111	001	刘诗诗	5056	安徽省合肥市	吴奇隆	189	lss@ustc.edu.cn	夫妻
112	002	王菊	5057	上海市	李磊	190	wj@mail.ustc. ...	夫妻
113	003	胡萧	5058	北京市	刘浩	191	hx@lusc.edu...	兄弟
114	004	张梅	5059	浙江省杭州市	肖然	192	zm@yohu.com	姐妹
115	005	李诺	5060	广东省广州市	胡鹏	193	xn@163.com	夫妻
116	006	刘彦	5061	四川省成都市	刘欢	194	ly@qq.com	夫妻

查:

MainWindow

身份证号: 管理人身份证号: 姓名:

联系电话: 家庭住址: 联系人姓名:

联系人手机号: 联系人邮箱: 与联系人关系:

选择所进行的操作:

身份证号	管理人身份证号	姓名	联系电话	家庭住址	联系人姓名	联系人手机号	联系人Email	与联系人关
111	001	刘诗诗	5056	安徽省合肥市	吴奇隆	189	lss@ustc.edu.cn	夫妻
112	002	王菊	5057	上海市	李磊	190	wj@mail.ustc...	夫妻
113	003	胡莱	5058	北京市	刘浩	191	hx@1ustc.edu...	兄弟
114	004	张梅	5059	浙江省杭州市	肖然	192	zm@yohu.com	兄妹
115	005	李诺	5060	广东省广州市	胡鹏	193	xn@163.com	夫妻
116	006	刘彦	5061	四川省成都市	刘欢	194	ly@qq.com	夫妻

账户管理: (支票账户为例)

增:

MainWindow

账户号: 开户行: 身份证号:

开户日期: 2000/1/1 余额: 账户类型: 支票

利率: 透支额: 货币类型:

重新输入

选择要进行的操作:

账户号	开户行	身份证号	开户日期	余额	透支额
1001	合肥农行	111	2017/06/20	100000.00	100.00
1002	上海工行	112	2020/04/27	200000.00	10000.00
1003	北京工行	113	2021/01/30	300000.00	30000.00
1004	上海农行	114	2019/01/11	400000.00	10.00
1005	合肥工行	116	2016/02/16	100000.00	12.00

删: (注销 1005)

MainWindow

账户号: 开户行: 身份证号:

开户日期: 2000/1/1 余额: 账户类型: 支票

利率: 透支额: 货币类型:

重新输入

选择要进行的操作:

账户号	开户行	身份证号	开户日期	余额	透支额
1001	合肥农行	111	2017/06/20	100000.00	100.00
1002	上海工行	112	2020/04/27	200000.00	10000.00
1003	北京工行	113	2021/01/30	300000.00	30000.00
1004	上海农行	114	2019/01/11	400000.00	10.00

改: (1004 透支额改为 10)

MainWindow

账户号: 开户行: 身份证号:

开户日期: 2000/1/1 余额: 账户类型: 支票

利率: 透支额: 货币类型:

重新输入

选择要进行的操作:

账户号	开户行	身份证号	开户日期	余额	透支额
1001	合肥农行	111	2017/06/20	100000.00	100.00
1002	上海工行	112	2020/04/27	200000.00	10000.00
1003	北京工行	113	2021/01/30	300000.00	30000.00
1004	上海农行	114	2019/01/11	400000.00	10.00

查：

MainWindow

账户号: 开户行: 身份证号:

开户日期: 2000/1/1 余额: 账户类型: 支票

利率: 透支额: 货币类型:

重新输入

选择要进行的操作:

账户号	开户行	身份证号	开户日期	余额	透支额
1001	合肥农行	111	2017/06/20	100000.00	100.00
1002	上海工行	112	2020/04/27	200000.00	10000.00
1003	北京工行	113	2021/01/30	300000.00	30000.00
1004	上海农行	114	2019/01/11	400000.00	0.00

贷款管理：

增：

MainWindow

账户管理
账户管理
贷款管理
统计查询

贷款号: 名字:

金额: 身份证号:

发放金额:

发放金额 重新输入

选择所要进行的操作:

贷款号	金额	名字	发放状态
1234	10000.00	上海农行	正在发放3000.00
4567	40000.00	南京工行	正在发放5000.00
5678	20000.00	北京工行	正在发放3000.00
8912	50000.00	合肥工行	正在发放6000.00
9123	30000.00	杭州农行	正在发放4000.00

删: (不可删除正在发放的贷款)

MainWindow

账户管理
账户管理
贷款管理
统计查询

贷款号: 名字:

金额: 身份证号:

发放金额:

发放金额 重新输入

选择所要进行的操作:

information

处于发放中, 不能删除!

贷款号	金额	名字	发放状态
9123	30000.00	杭州农行	正在发放4000.00

发放贷款：

MainWindow

贷款号: 名字:

金额: 身份证号:

发放金额:

选择所要进行的操作:

贷款号	金额	名字	发放状态
1234	10000.00	上海农行	正在发放3000.00
4567	40000.00	南京工行	正在发放5000.00
5678	20000.00	北京工行	正在发放3000.00
8912	50000.00	合肥工行	正在发放6000.00
9123	30000.00	杭州农行	已全部发放

查：

MainWindow

账户管理
账户管理
贷款管理
统计查询

贷款号: 名字:

金额: 身份证号:

发放金额:

发放金额 重新输入

选择所要进行的操作:

贷款号	金额	名字	发放状态
1234	10000.00	上海农行	正在发放3000.00
4567	40000.00	南京工行	正在发放5000.00
5678	20000.00	北京工行	正在发放3000.00
8912	50000.00	合肥工行	正在发放6000.00
9123	30000.00	杭州农行	正在发放4000.00

业务统计:

储蓄统计:

选择合法的时间段:

2000/1/1 到 2020/1/1

统计 重新输入

支行名	客户数	总金额
合肥农行	1	0.00
上海农行	1	10000.00
南京工行	1	40000.00

选择合法的时间段:

2000/1/1 到 2020/1/1

统计 重新输入

支行名	客户数	总金额
合肥农行	1	100000.00
上海农行	1	400000.00

4.3 实现中的难点问题及解决

本次实验的难点问题则在于贷款部分的删除问题，删除部分由于数据库中的逐次支付以及贷款两个

表中，都会涉及到贷款号，于是只在贷款一个表中进行删除，便会报错；并且处于发放中的贷款是不能删除的，对于此问题的解决是，首先先在逐次支付表中做一个求和，选出对应贷款号的总共发放金额，与申请的贷款数量进行一个对比，便分为了三种情况，如果可以进行删除的话，要先将逐次支付表中，关于此贷款的消息全部删除，再将贷款表中关于此贷款号的消息删除。如果不能进行删除的话，则直接给用户显示一个错误操作提示。

5 总结与讨论

通过本次实验，让我了解到了数据库是怎样和 python 以及前端进行交互的，体会了 CS 架构，以及整个的交互过程，对数据库有更好的认识。同时也对数据库语句进行了一定的巩固。本次实验，也让我去学习了一下 qt,发现还是很友好的，直接通过拖图便可以完成一些复杂界面的设计，大量的减少了本次实验的工作量。

本次实验的教训主要在于 pyqt 的学习以及一些应用的不熟练，通过了比较多的尝试才摸索到了最后的正解。