**“银行业务管理系统”**

系统设计与实现报告

**姓名：徐昊天**

**学号：PB19071535**

**计算机科学与技术学院**

**中国科学技术大学**

2022年6月

**目 录**

1 概述 1

1.1 系统目标 1

1.2 需求说明 1

1.3 本报告的主要贡献 2

2 总体设计 2

2.1 系统模块结构 2

2.2 系统工作流程 3

2.3 数据库设计 3

3 详细设计 7

3.1 客户管理模块 7

3.2 账户管理模块 9

3.3 贷款管理模块 10

3.3 业务统计模块 11

4 实现与测试 12

4.1 实现结果 12

4.2 测试结果 14

4.3 实现中的难点问题及解决 29

5 总结与讨论 29

# 

# 1 概述

## 1.1 系统目标

本系统主要目标为开发一个银行管理系统。系统中数据库以mysql为基础，后端开发语言为python，并通过python中的pyQt工具进行图形界面开发。该系统需要实现客户管理、账户管理、贷款管理、业务统计四个方面的功能需求。

## 1.2 需求说明

**银行业务管理系统的数据需求如下：**

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。

**要求实现的主要功能需求如下：**

**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；

**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；

**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；

**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

1. 提供该系统的模块结构，展现系统的工作流程和数据库设计。

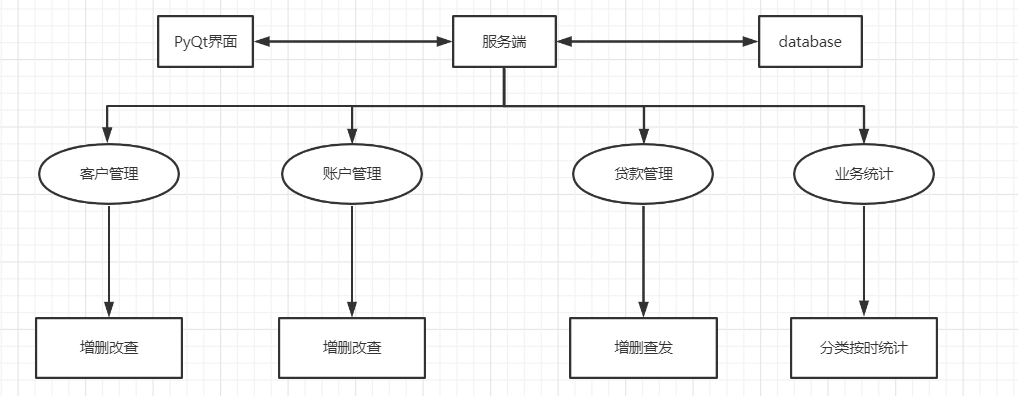
2. 对系统的每一个模块的详细设计进行叙述。

3. 展示系统的功能实现，确保系统的正确性和可用性。

4. 对此次系统开发进行总结和改进。

# 2 总体设计

## 2.1 系统模块结构



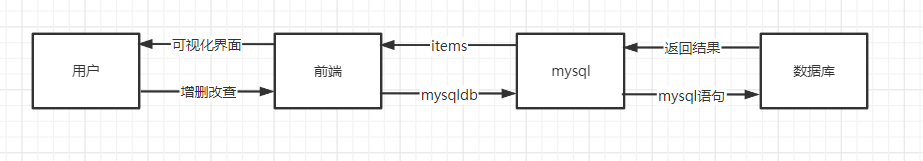
1. 本系统利用PyQt开发可视化界面。利用qt designer设计界面并通过qt-tools工具将生成的.ui文件转换成.py文件。

2. 本系统存在四个模块：客户管理、账户管理、贷款管理、业务统计。在模块中通过python语言实现对数据库的操作。

3. 四个模块中对于每一种功能均有对应的函数接口，函数用于通过设计mysql语句，对数据库执行对应的操作。

4. 通过在后端python中设计print函数，将数据库信息展示在Qt界面上。

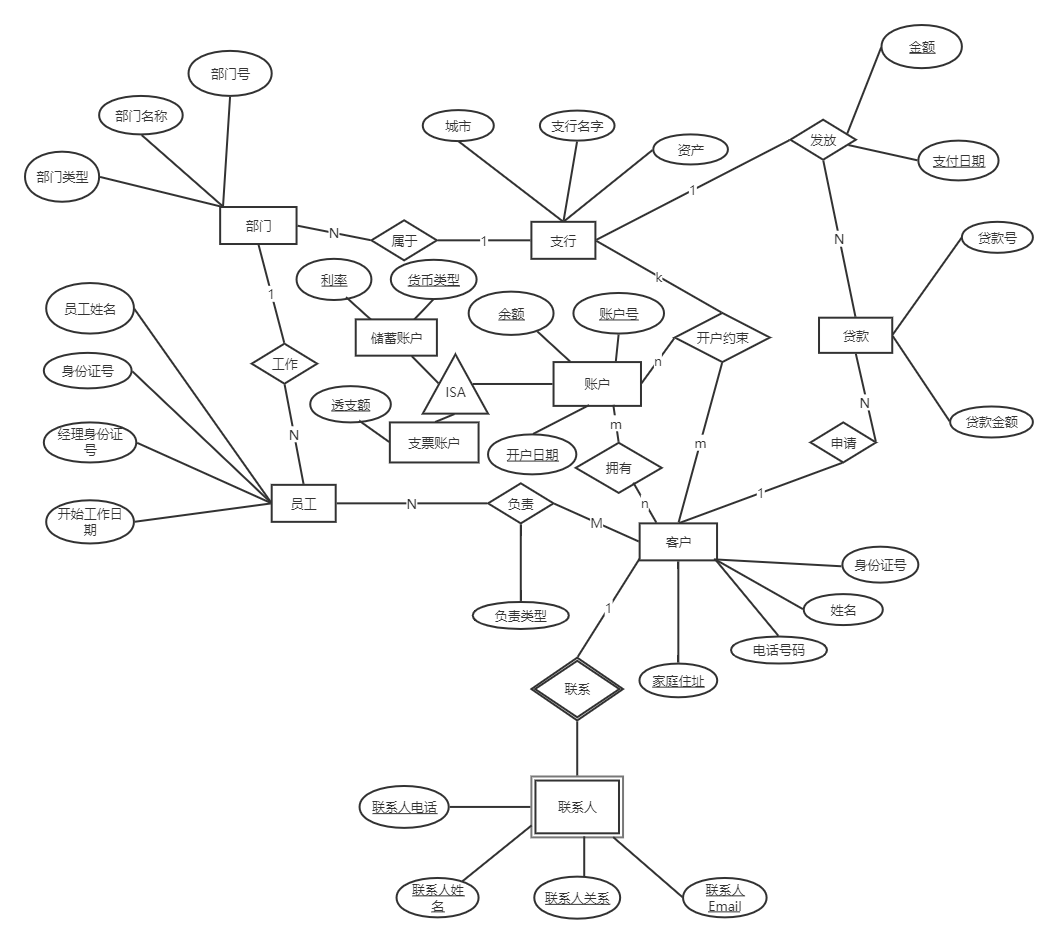
## 2.2 系统工作流程



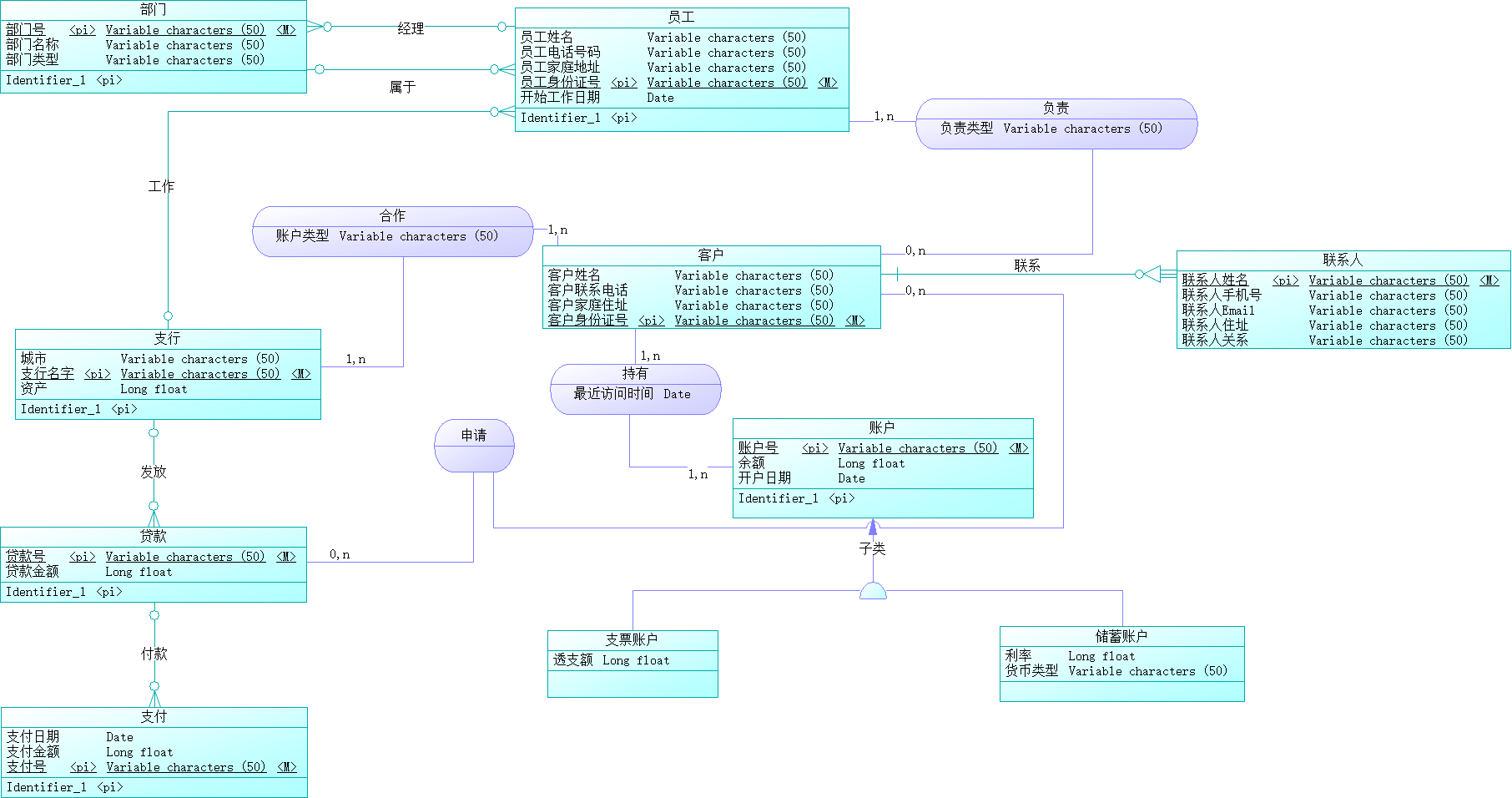
用户通过在QT前端界面上选择增删改查操作，后端python代码通过执行mysql语句对数据库进行对应操作。

## 2.3 数据库设计

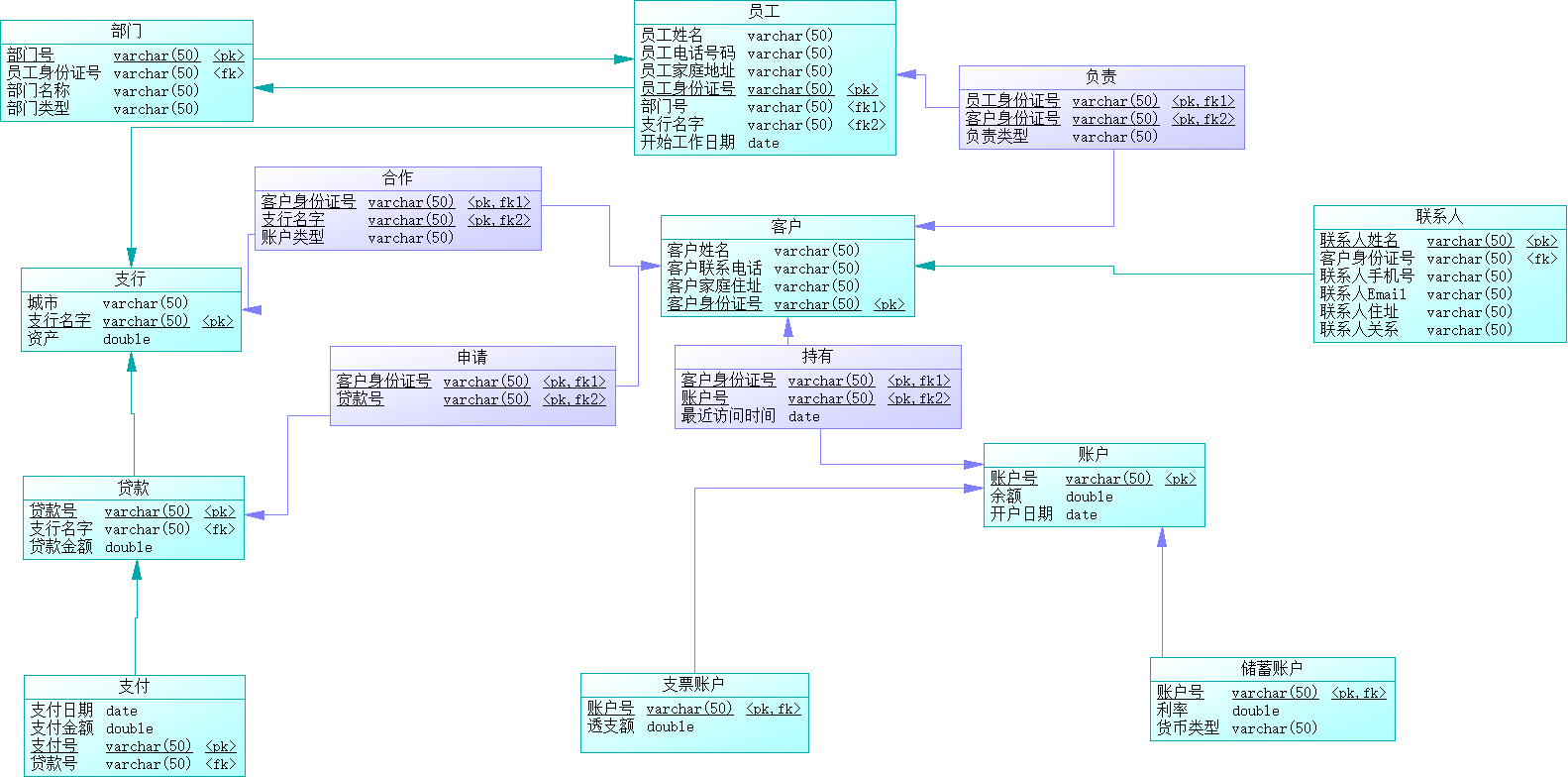
ER图如下：



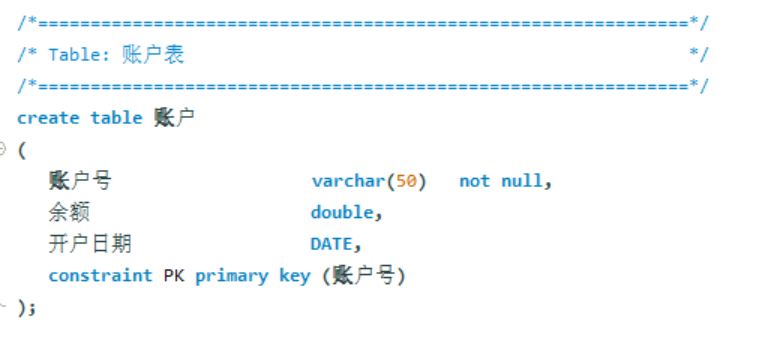
CDM如下：



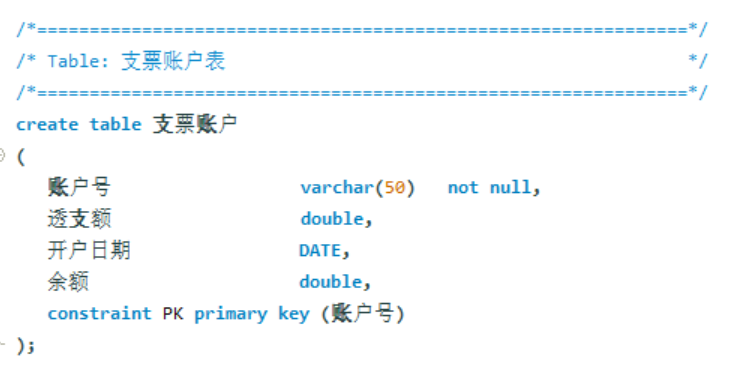
PDM如下：

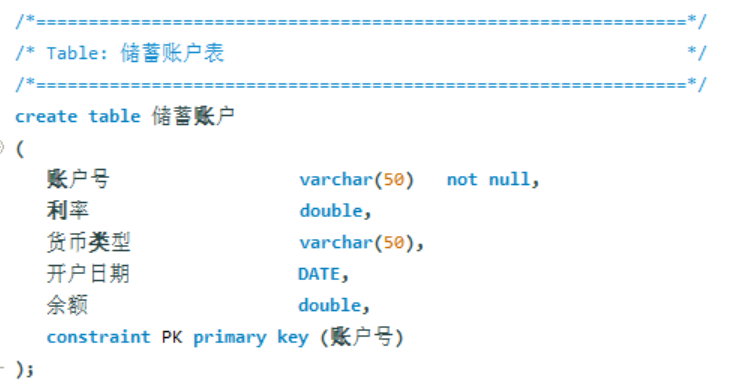


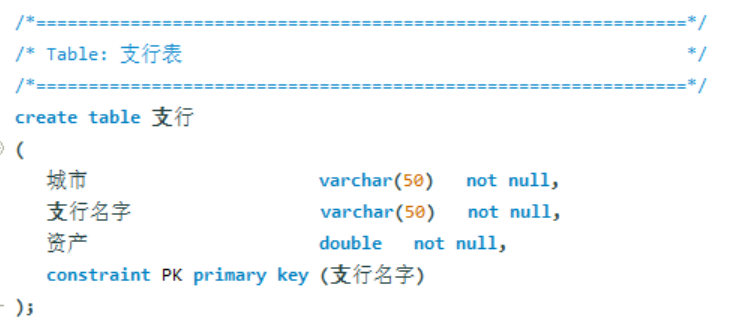
最终建立的mysql表如下：

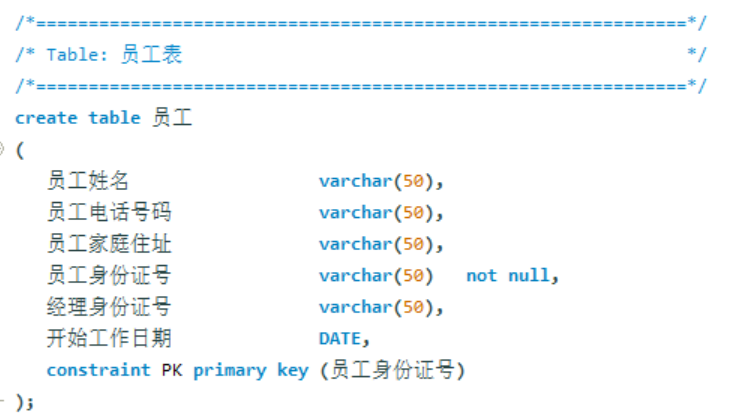


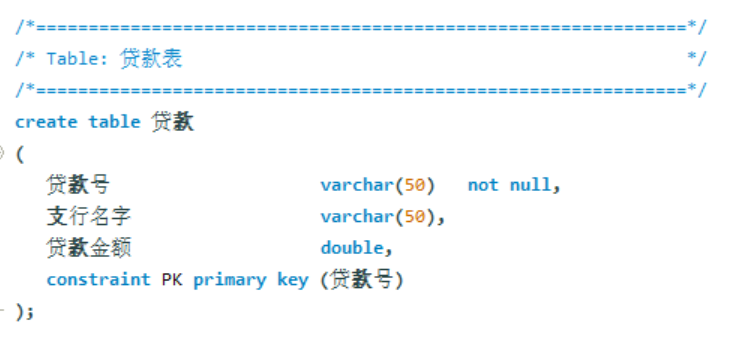


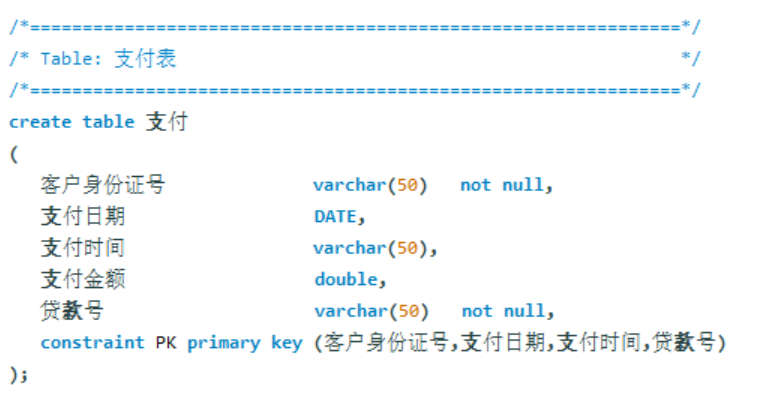


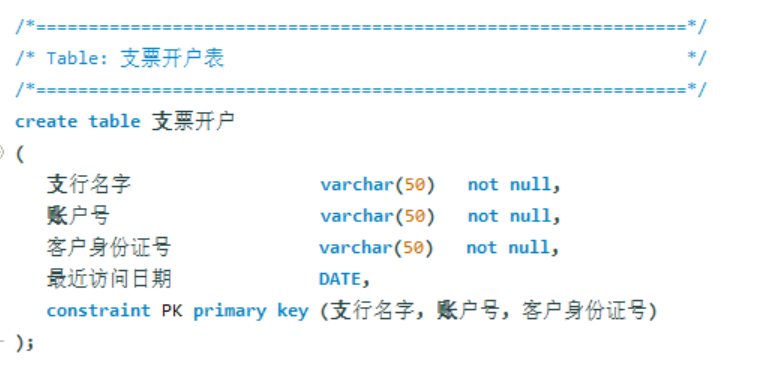


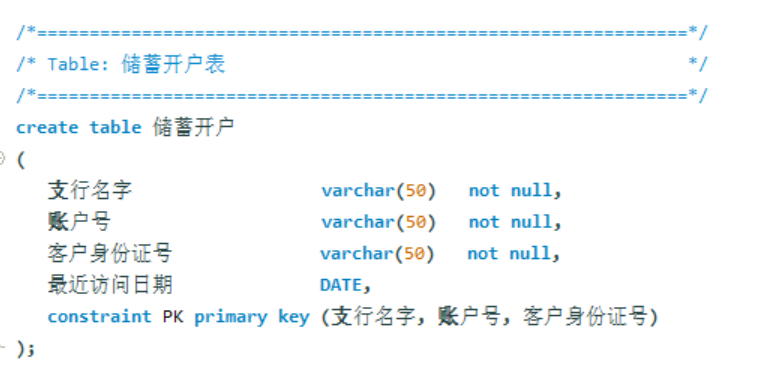


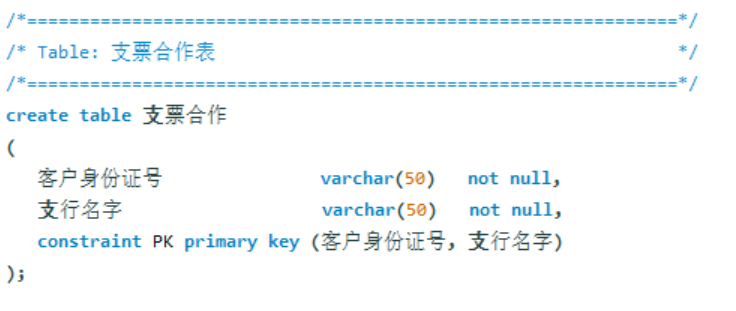


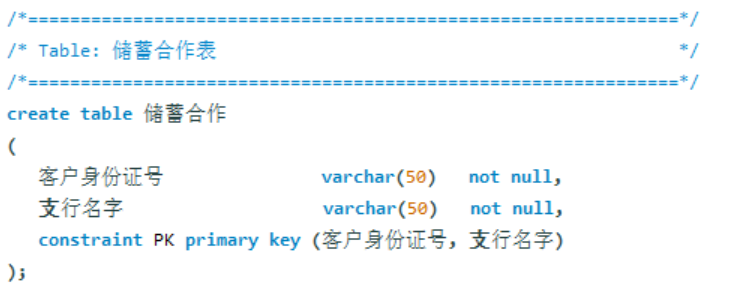












# 3 详细设计

## 3.1 客户管理模块

**输入：**

客户姓名

客户身份证号

客户家庭住址

客户电话

联系人姓名

联系人Email

联系人关系

联系人电话

负责人身份证号

负责人类型

执行的操作

**输出：**

**查询：**

客户姓名

客户身份证号

客户家庭住址

客户电话

联系人姓名

联系人Email

联系人关系

联系人电话

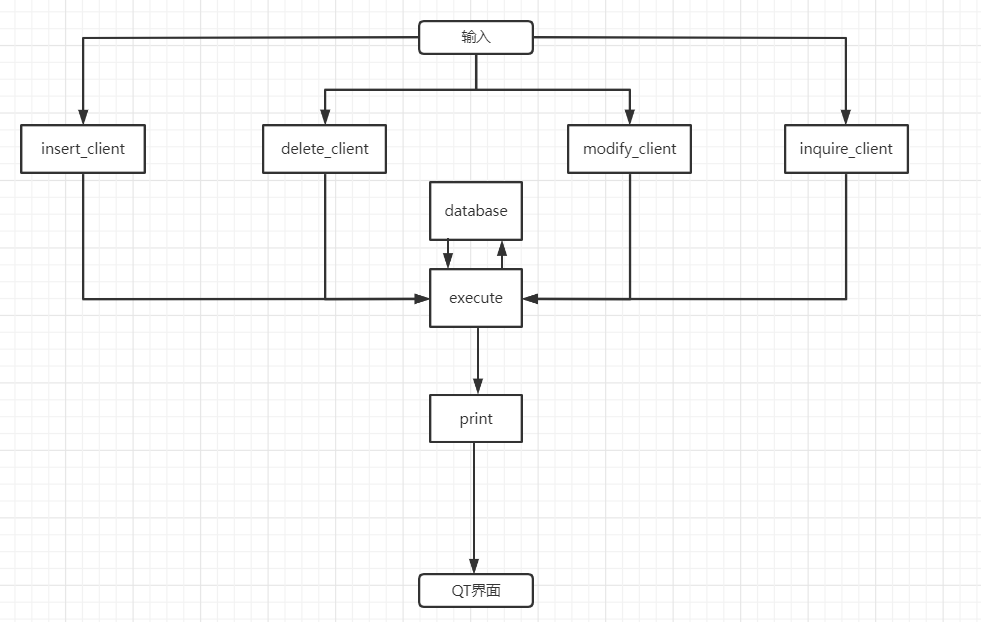
负责人身份证号

负责人类型

**增加、删除、修改、清空：**

无输出

**程序流程图：**



## 3.2 账户管理模块

**输入：**

账户号

余额

开户日期

开户支行

客户身份证号

账户类型

利率、货币类型（若为储蓄账户）

透支额（若为支票账户）

执行的操作

**输出：**

**查询：**

账户号

余额

开户日期

开户支行

客户身份证号

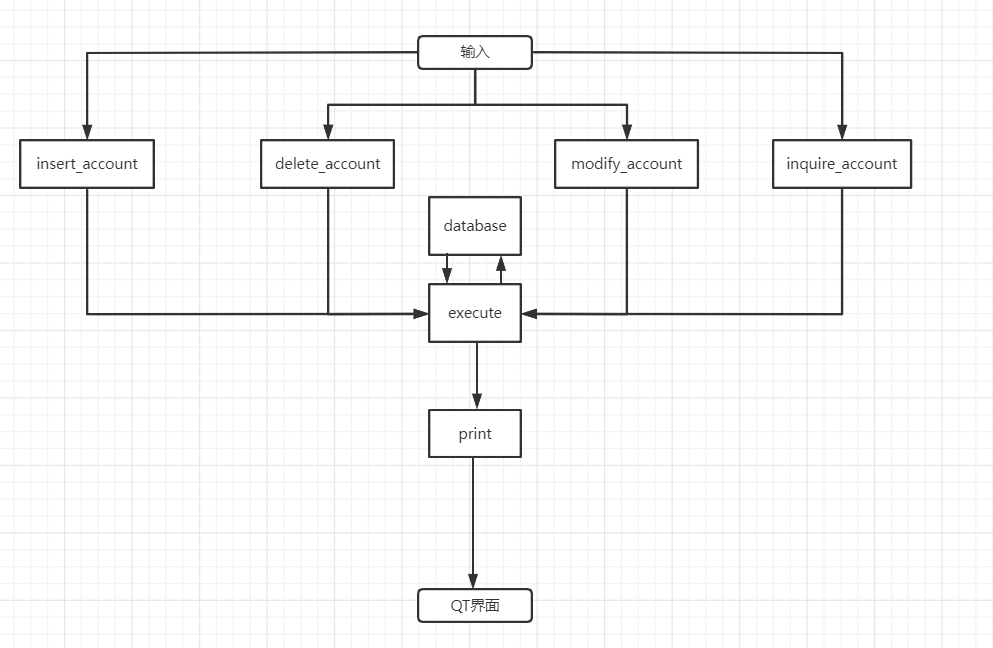
利率、货币类型（若为储蓄账户）

透支额（若为支票账户）

**开户、销户、修改、清空：**

无输出

**程序流程图：**



## 3.3 贷款管理模块

**输入：**

贷款号

总金额

贷款支行

客户身份证号

贷款金额

执行的操作

**输出：**

**查询：**

贷款号

总金额

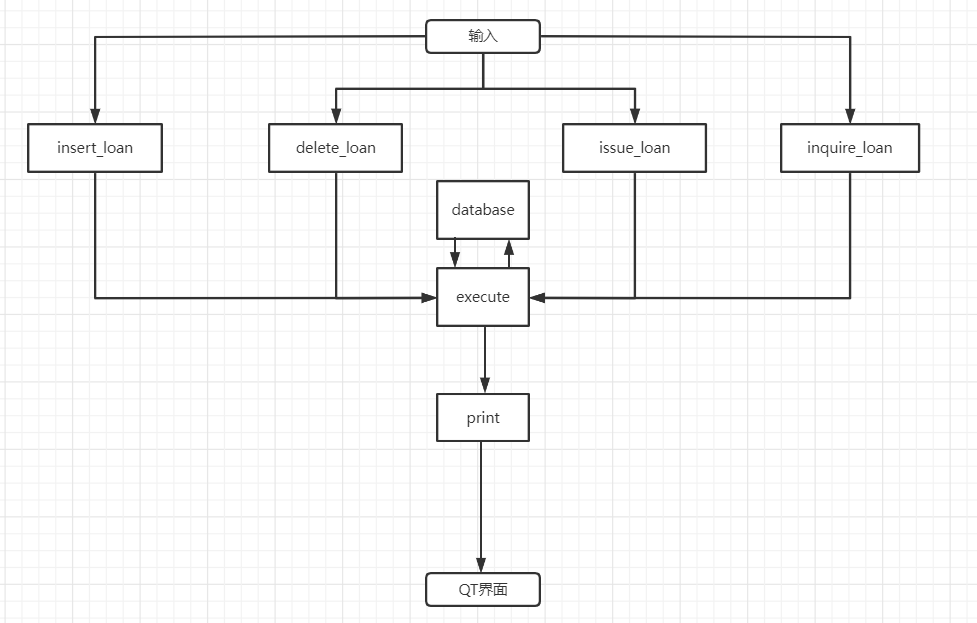
贷款支行

贷款状态

**增加、删除、发放、清空：**

无输出

**程序流程图：**



## 3.4 业务统计模块

**输入：**

统计起始时间

统计结束时间

执行的操作

**输出：**

**储蓄统计、贷款统计：**

业务总金额

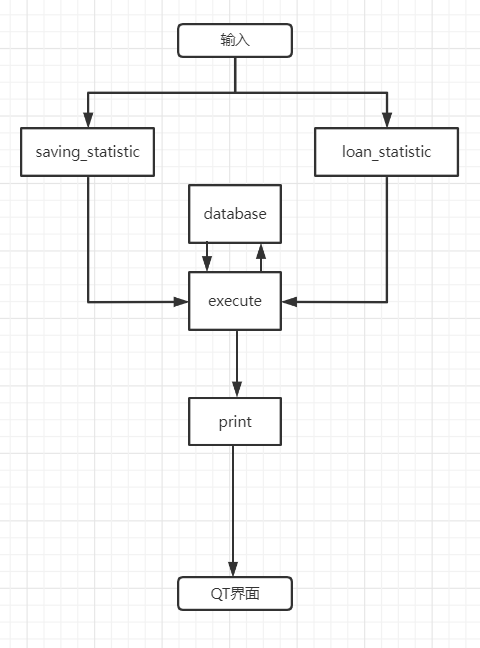
支行名字

用户数

**清空：**

无输出

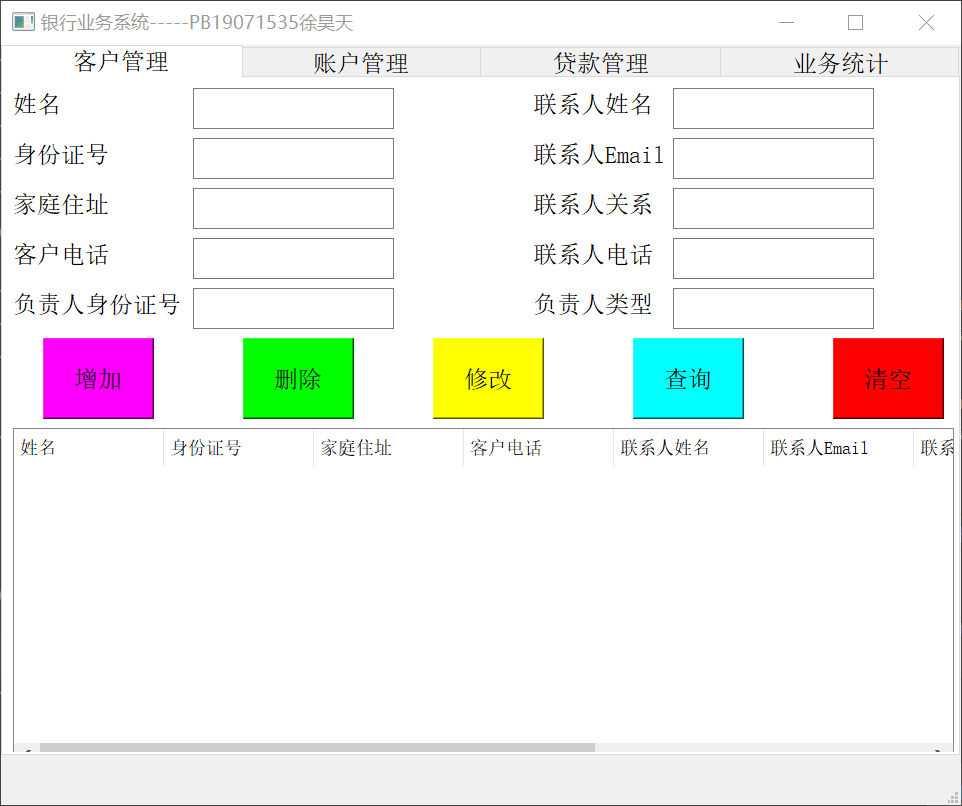
**程序流程图：**



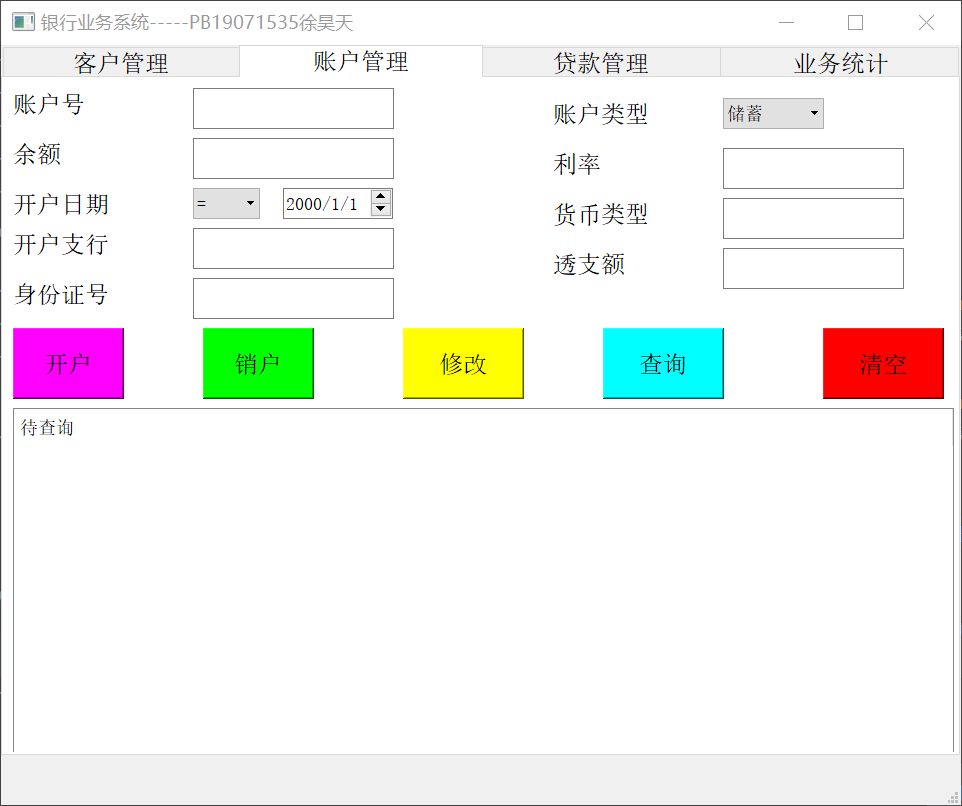
# 4 实现与测试

## 4.1 实现结果

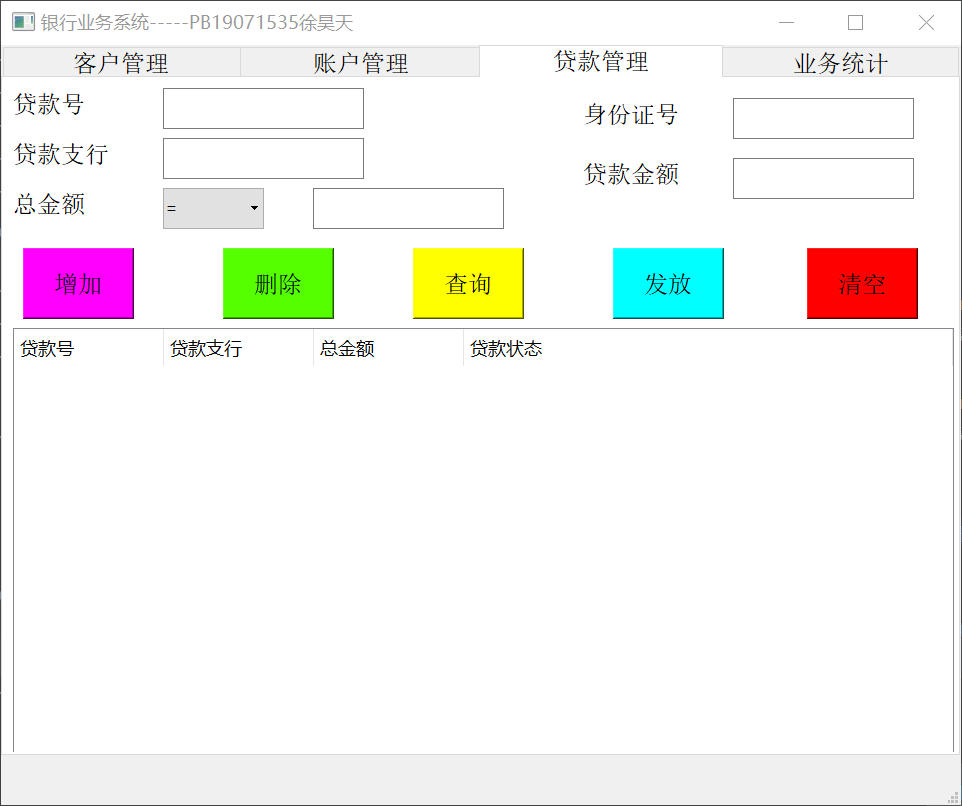
**客户管理模块：**



**账户管理模块：**



**贷款管理模块：**



**业务统计模块：**



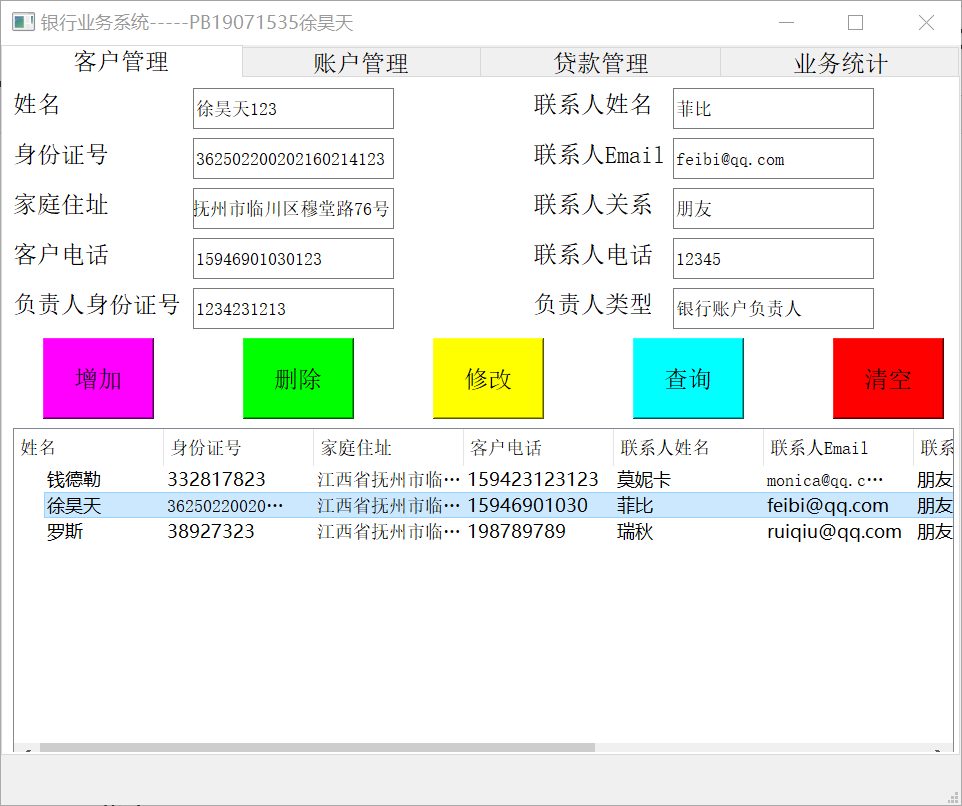
## 4.2 测试结果

**客户管理模块：**

**查询：**



**增加：**

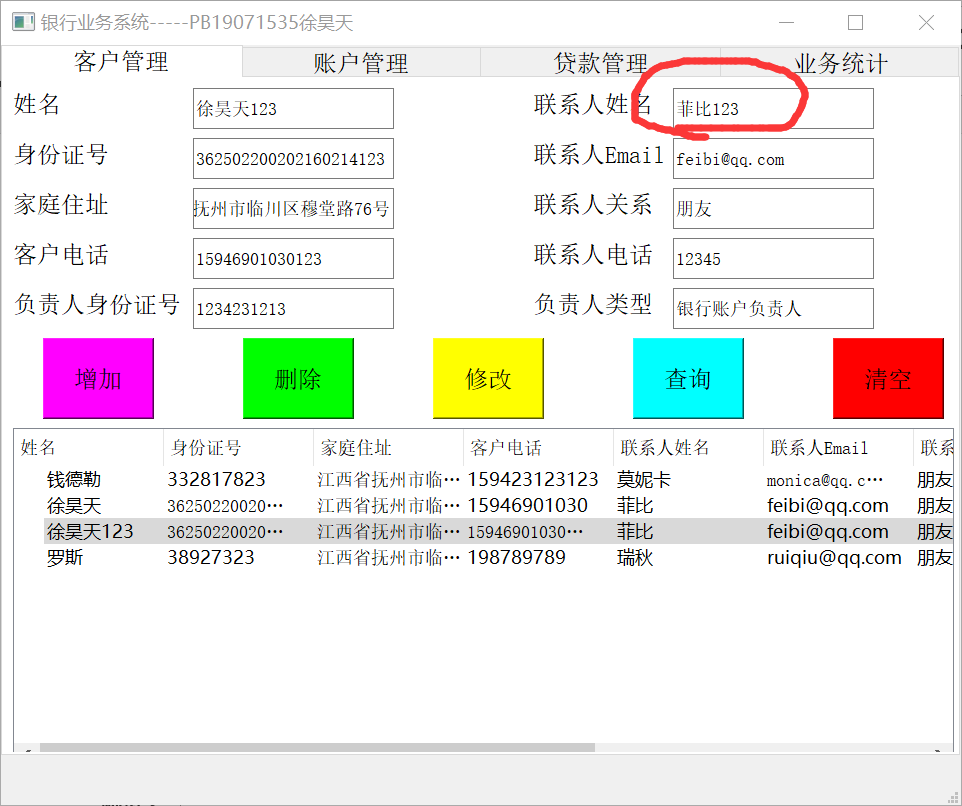


****

**考虑名字中带有引号的情况：**

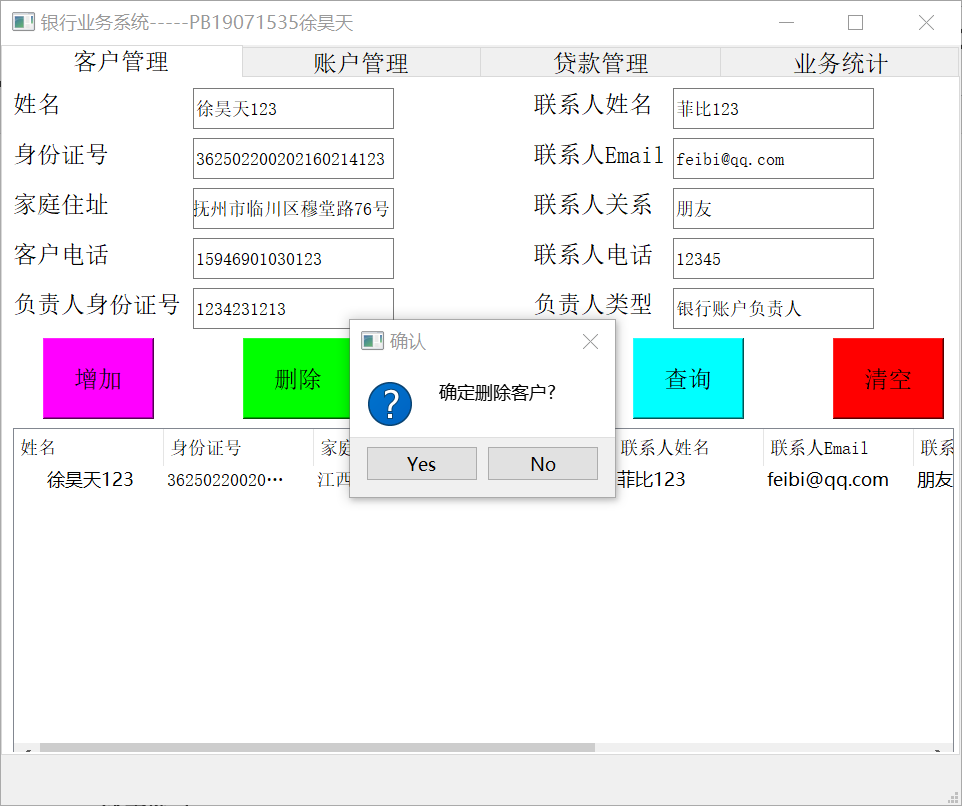
****

**修改：**

****



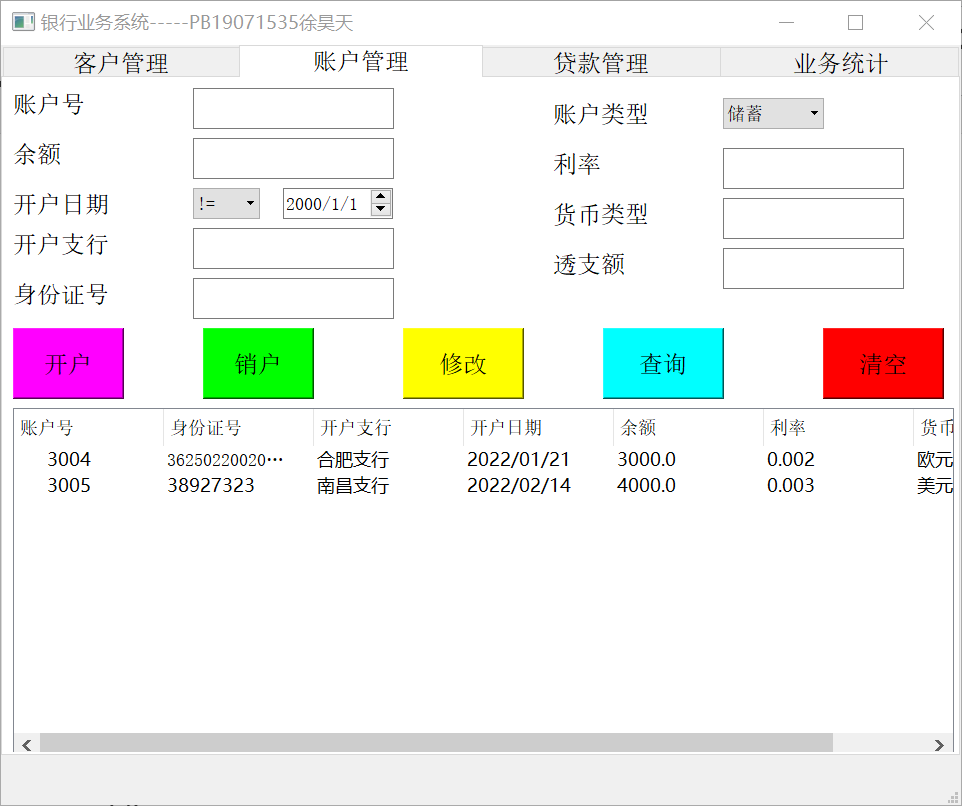
**删除：**

****

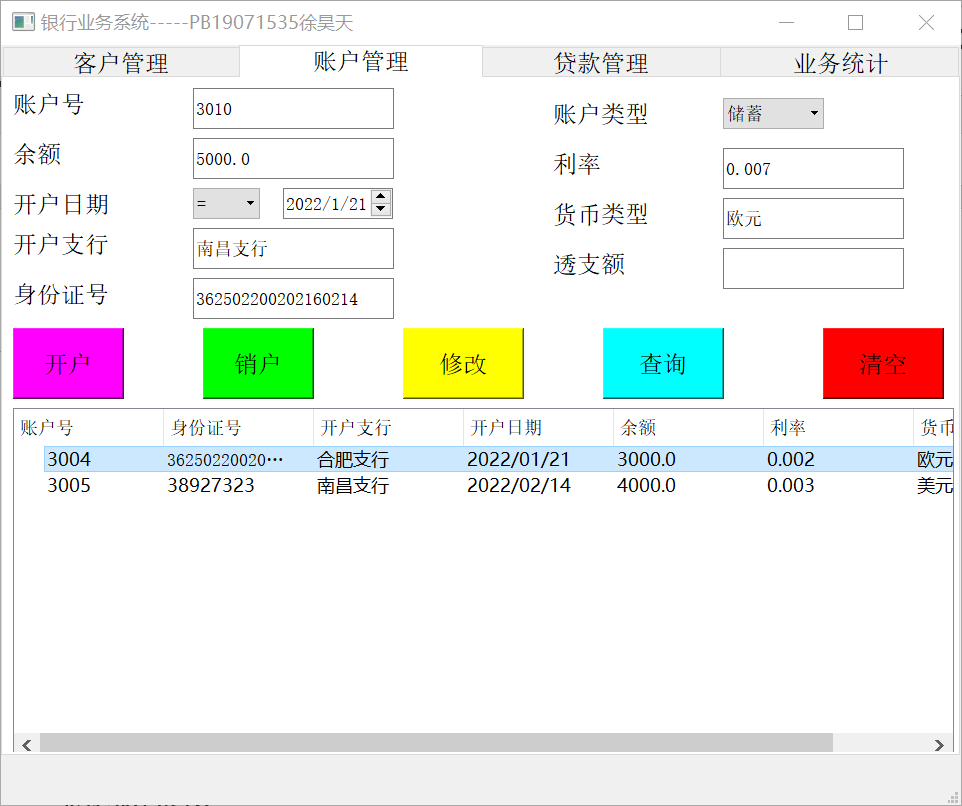
****

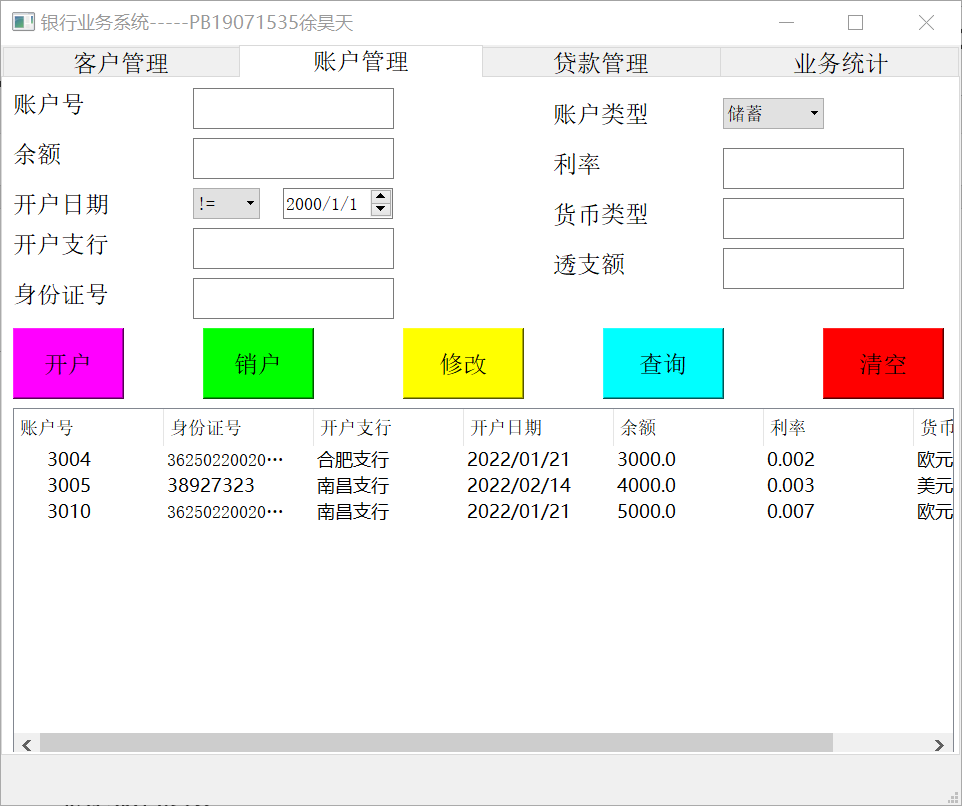
**账户管理模块：**

**查询：**

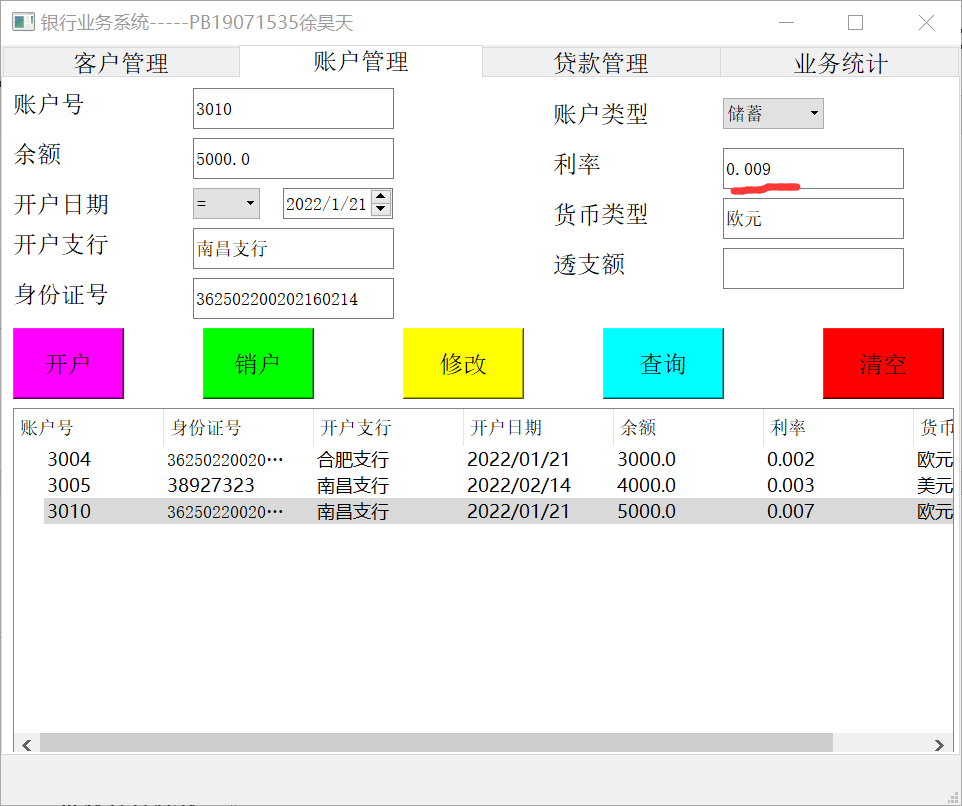
****

**开户：**

****

****

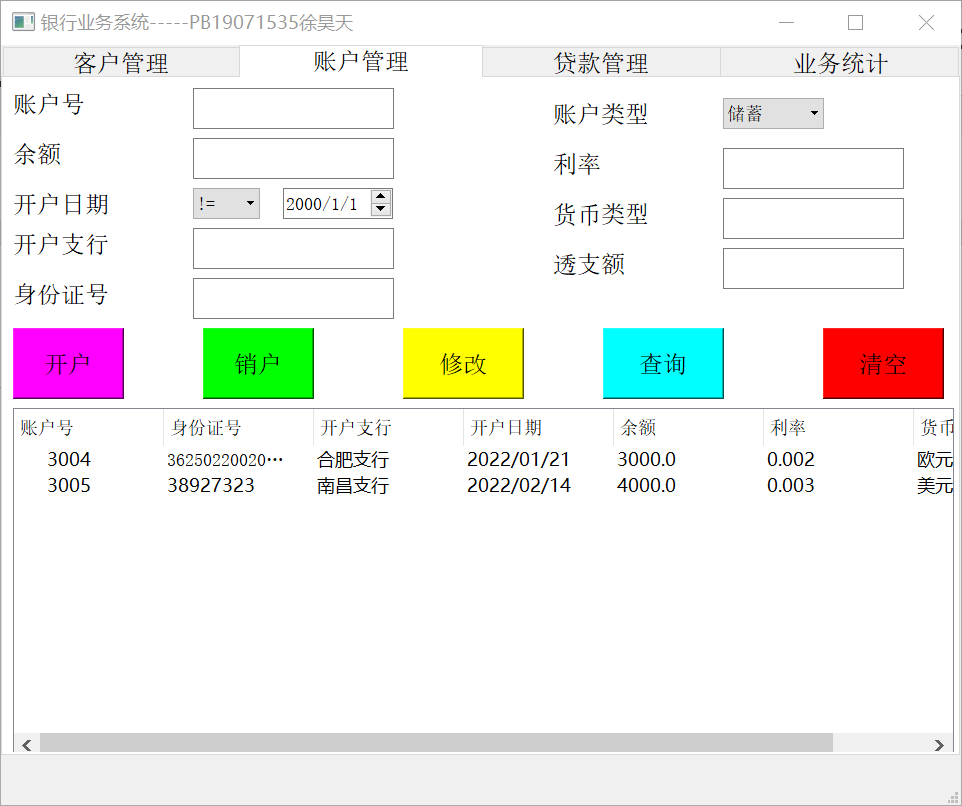
**修改：**

****

****

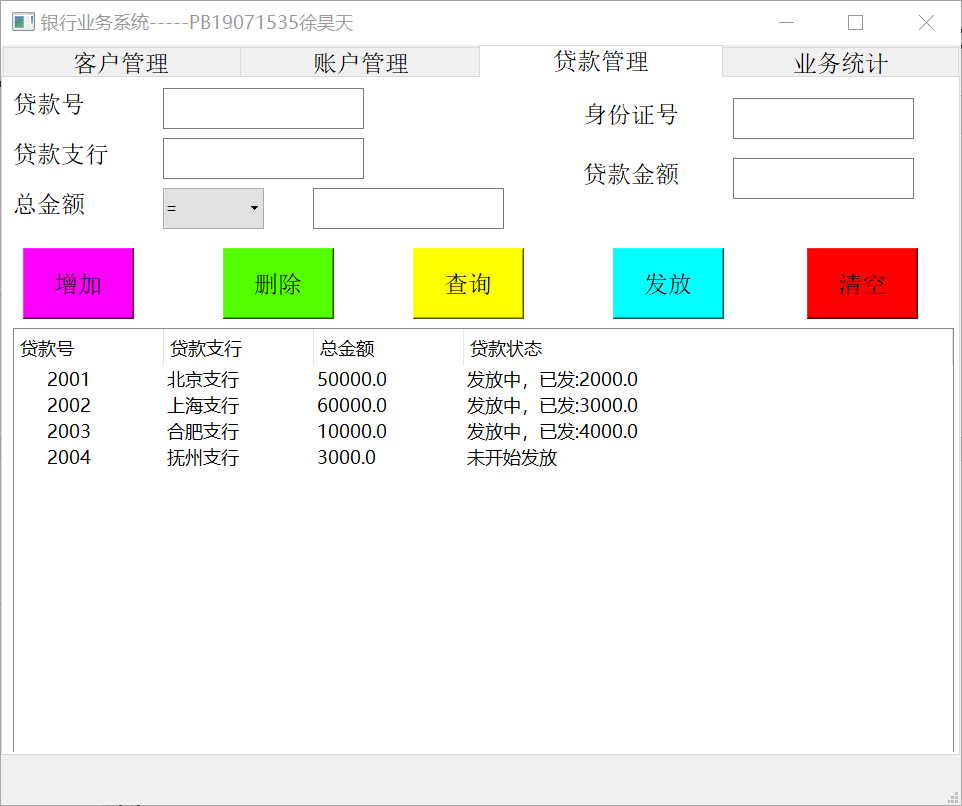
**销户：**

****

****

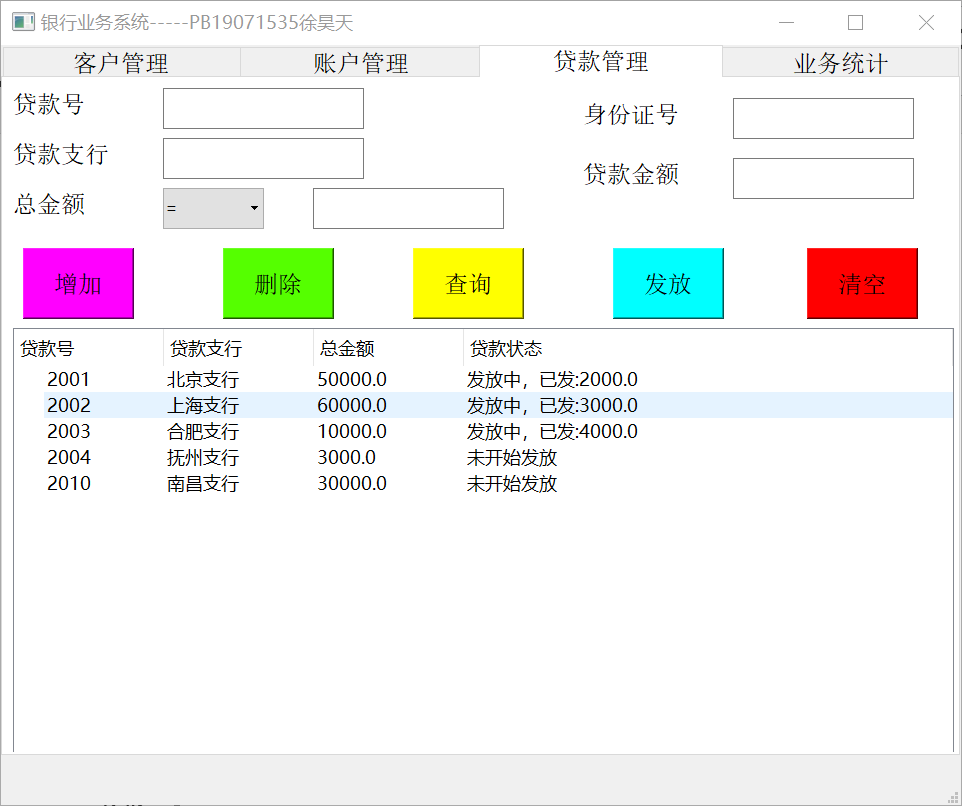
**贷款管理模块：**

**查询：**

****

**增加：**

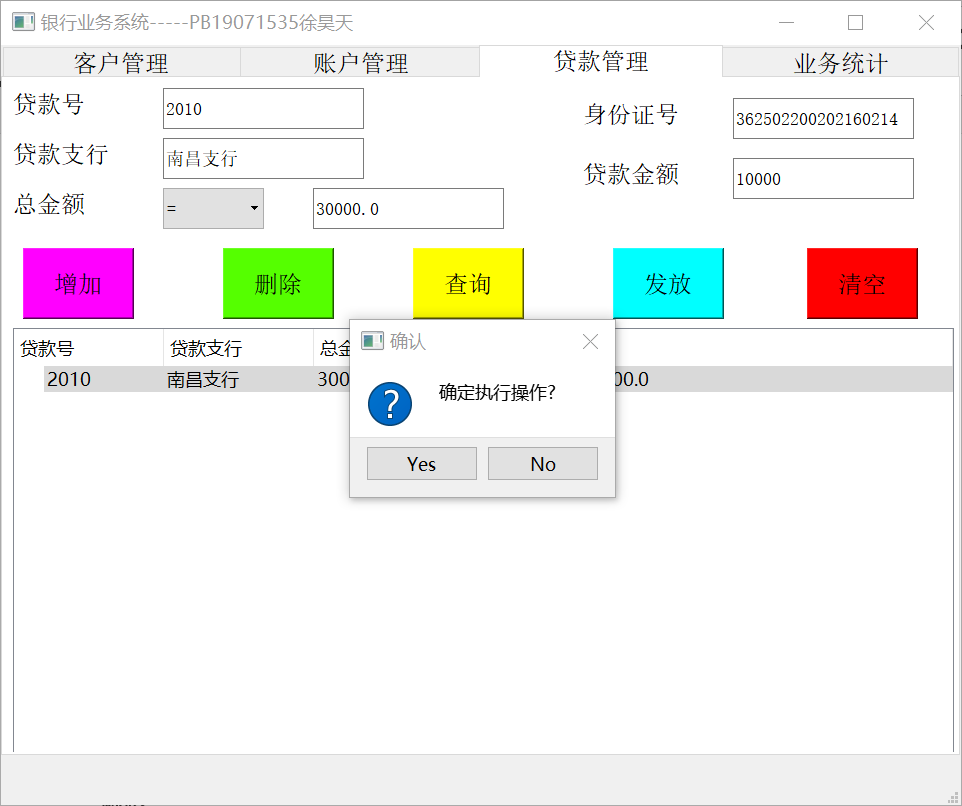
****

****

**发放：**

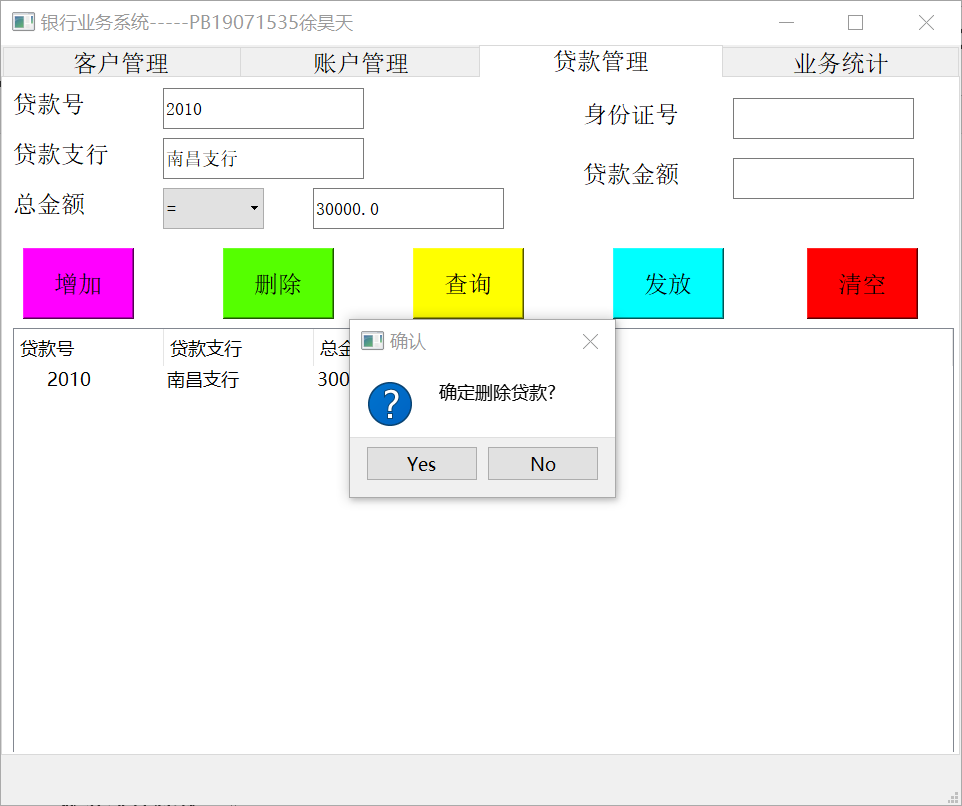
****

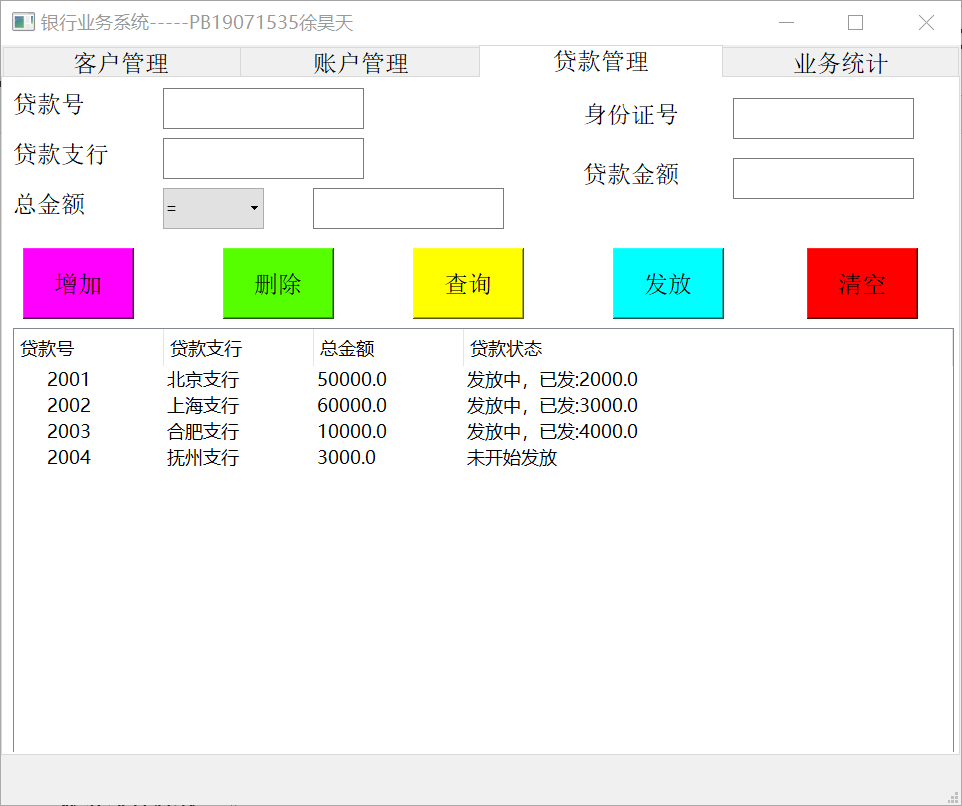
****

****

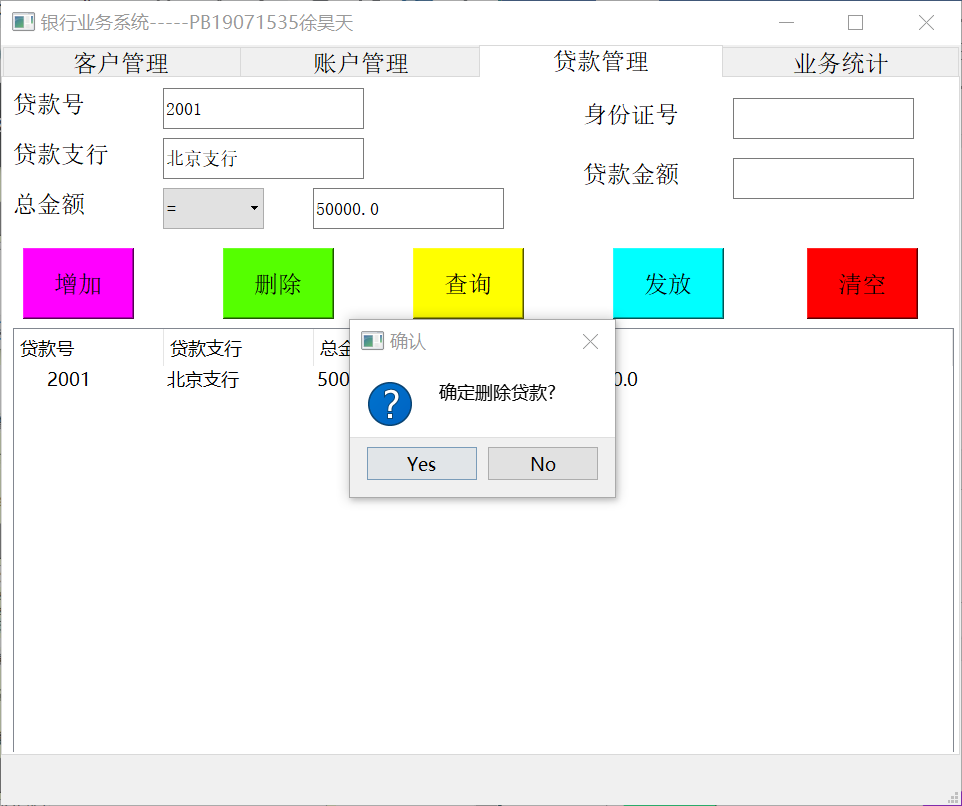
****

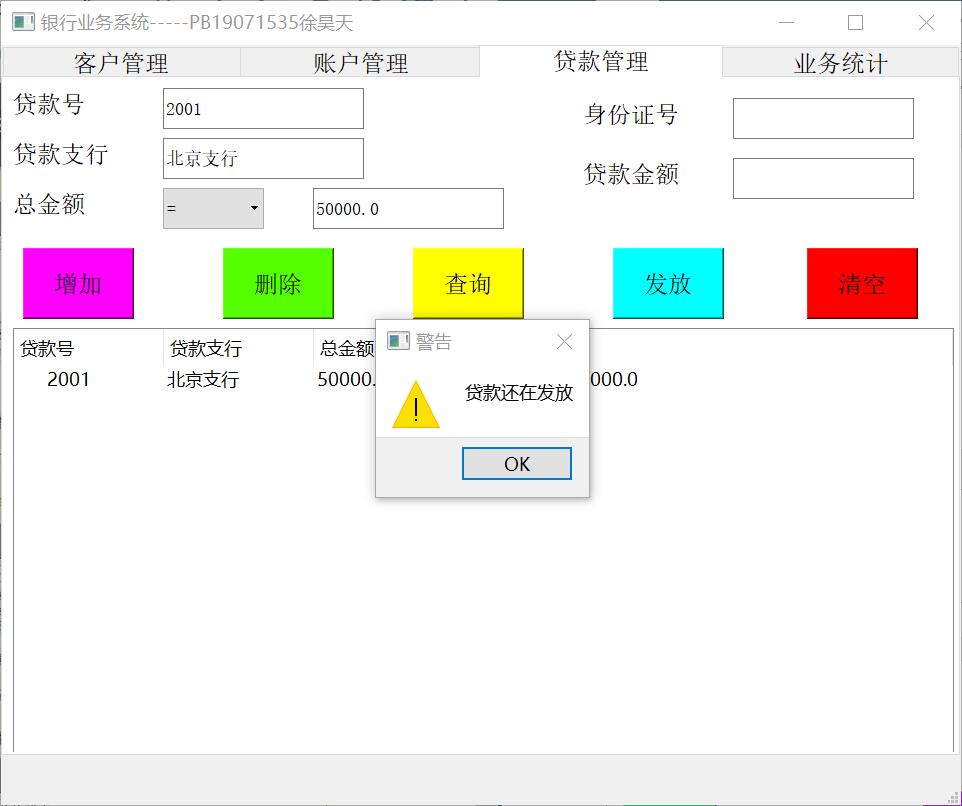
**删除：**

****

****

**考虑正在发放的贷款：**

****

****

**业务统计模块：**

**储蓄统计：**

****

**贷款统计：**



## 4.3 实现中的难点问题及解决

难点问题1. 没有开发界面的经验，没有学过前端语言或者其他界面开发的工具。

解决方案：阅读助教提供的pyqt demo，学习使用qt designer，使用pyqt这一工具进行界面开发。

难点问题2. 在实现某些插入、删除、更改操作时，由于存在外键约束，会导致操作失败。

解决方案：设计mysql时考虑外键的约束 ，并在执行操作时根据可能存在的外键约束合理安排对于不同表的操作顺序，从而尽量避免错误；并在“try…except…”结构中将出现的execption打印出来，便于找出bug。

# 5 总结与讨论

1. 首次在编程中实现界面开发，学习了qt designer的使用，并熟悉了python + pyqt的开发架构。

2. 熟悉了通过将python连接mysql从而对数据库执行操作的方式。

3. 通过对mysql执行一些复杂操作并设计mysql语句，巩固了mysql的语法知识。

4. 需重视错误处理，不仅方便debug，还能提高程序的健壮性。

5. 尽量将每一种功能单独写一个函数，方便对每一种功能单独进行测试；并在写完一种功能后预先测试正确性，防止错误积少成多。

6. 由于数据需求中部分个体在该银行业务管理系统中作用较小，故在设计mysql表时可以略微调整mysql表结构，便于后端对各种操作的执行。