**“银行业务管理系统”**

系统设计报告

**学号:PB15051065**

**姓名：罗永平**

**计算机科学与技术学院**

**中国科学技术大学**

2018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 1

2 总体设计 1

2.1 系统模块结构 1

2.2 系统工作流程 1

2.3 数据库设计 1

3 详细设计 2

3.1 支行管理模块 2

3.2 员工管理模块 2

3.3 客户管理模块 2

3.4 账户管理模块 2

3.5 贷款管理模块 2

3.6 统计模块 2

4 实现与测试 2

4.1 实现结果 2

4.2 测试结果 2

5 总结与讨论 2

# 

# 1 概述

## 系统目标

# 本实验实现了一个银行数据库管理系统，该系统主要有6个功能，分别为支行管理，员工管理，客户管理，账户管理，贷款管理，统计查询等功能。该系统为每个功能实现了基本的条件查询、数据库插入，数据库删除和数据库更改等基本的数据库访问需求，针对每个功能还实现了简单美观的界面，很大程度地方便了用户进行数据管理。

## 1.2 需求说明

**1. 数据库需求**：

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。

**2. 功能需求**

支行管理：提供支行所有信息的增、删、改、查功能；如果支行存在着关联信息，如员工、账户等，则不允许删除；

员工管理：提供支行员工所有信息的增、删、改、查功能；如果员工存在着关联数据，则不允许删除；

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。

## 本报告的主要贡献

* 明确银行数据库管理系统的功能需求和基本的数据库需求
* 总体概括系统的设计结构
* 详细介绍系统的设计方法和程序流程
* 对系统设计进行正确性测试和分析
* 系统设计的总结与报告

# 2 总体设计

## 系统模块结构

* 银行系统主要分为六个功能模块，不同功能模块之间功能完全分离，互不影响。
* 每个功能模块均有对应的UI部分和事件逻辑部分。事件逻辑调用UI界面的组件进行输入获取，输出写入和事件触发。
* 每个功能模块实现了增、删、改、查等基本数据库功能，分别对应insert\_xx，update\_xx, delete\_xx和select\_xx等接口，以上接口负责相应功能模块下增、删、改、查事件的管理逻辑，事件触发依赖于PyQt的信号槽机制，设计过程中绑定对应的信号和接受槽函数实现事件触发。
* 上述每个接口均会调用同一的接口get\_query()实现对数据库执行sql语句并提交到数据库。
* 数据库的查询结果通过feed\_table\_xx接口展填充到每个功能界面对应的表格中。

## 系统工作流程

**查询流程：**

用户输入查询条件 >> select\_xx获得对应的query语句 >> 调用对应的get\_query()函数访问数据库 >> 调用feed\_table\_xx将查询结果写入表格

**增加流程：**

用户输入查询条件 >> insert\_xx获得对应的query语句 >> 弹窗询问是否确认写入 >>确认写入则调用对应的get\_query()函数访问数据库，否则推退出

**修改流程：**

用户双击表格中对应项目 >> 双击项目写入界面 >> 修改界面上项目的值 >> 用户点击修改按钮 >> update\_xx获得对应的query语句 >> 弹窗询问是否确认写入 >> 确认写入则调用对应的get\_query()函数访问数据库，否则推退出

**删除流程：**

用户输入删除项目条件信息 >> 调用select\_xx选中待删除项目并显示在表格中 >> 弹窗询问确定删除选中项目 >>

**清空：**

清空UI界面上的全部输入 >> 设置为初始值

## 2.3 数据库设计

**数据库：**

支行:

**create** **table** **支**行

**(**

城市 nvarchar2**(**20**)** **not** **null,**

名字 nvarchar2**(**20**)** **not** **null,**

资产 **NUMBER(**8**,**2**)** **not** **null,**

**constraint** PK\_**支**行 **primary** **key** **(**名字**)**

**);**

员工:

**create** **table** 员工

**(**

身份证号Y nvarchar2**(**20**)** **not** **null,**

员工\_身份证号Y nvarchar2**(**20**),**

姓名Y nvarchar2**(**50**),**

电话号码Y nvarchar2**(**50**),**

家庭住址Y nvarchar2**(**50**),**

开始工作日期 **DATE,**

**constraint** PK\_员工 **primary** **key** **(**身份证号Y**)**

**);**

客户：

**create** **table** 客户

**(**

身份证号 nvarchar2**(**20**)** **not** **null,**

身份证号Y nvarchar2**(**20**),**

姓名 nvarchar2**(**50**),**

联系电话 nvarchar2**(**50**),**

家庭住址 nvarchar2**(**50**),**

联系人姓名 nvarchar2**(**50**)** **not** **null,**

联系人手**机**号 nvarchar2**(**50**),**

联系人Email nvarchar2**(**50**),**

联系人与客户关系 nvarchar2**(**50**),**

负责人**类**型 nvarchar2**(**50**),**

**constraint** PK\_客户 **primary** **key** **(**身份证号**)**

**);**

账户：

**create** **table** **账**户

**(**

**账**户号 **INTEGER** **not** **null,**

开户日期 **DATE,**

余额 **NUMBER(**8**,**2**),**

**);**

储蓄账户：

基于账户增加**（透支额 NUMBER(8,2),）**

贷款账户：

基于账户增加

(

货币**类**型 nvarchar2**(**50**)，**

**利**率 **FLOAT,**

)

账户拥有：

**create** **table** 拥有5

**(**

贷**款**号 **INTEGER** **not** **null,**

身份证号 nvarchar2**(**20**)** **not** **null,**

**constraint** PK\_拥有5 **primary** **key** **(**贷**款**号**,** 身份证号**)**

**);**

开户：

贷款开户：

**create** **table** 贷**款**开户

**(**

身份证号 nvarchar2**(**20**)** **not** **null,**

名字 nvarchar2**(**20**)** **not** **null,**

**账**户号 **INTEGER** **not** **null,**

最近访问日期\_贷**款** **DATE,**

**constraint** PK\_贷**款**开户 **primary** **key** **(**身份证号**,** 名字**,** **账**户号**)**

**);**

储蓄开户：

**create** **table** 储蓄开户

**(**

身份证号 nvarchar2**(**20**)** **not** **null,**

名字 nvarchar2**(**20**)** **not** **null,**

**账**户号 **INTEGER** **not** **null,**

最近访问日期 **DATE,**

**constraint** PK\_储蓄开户 **primary** **key** **(**身份证号**,** 名字**,** **账**户号**)**

**);**

贷款：

**create** **table** 贷**款**

**(**

金额1 **NUMBER(**8**,**2**),**

贷**款**号 **INTEGER** **not** **null,**

名字 nvarchar2**(**20**),**

**constraint** PK\_贷**款** **primary** **key** **(**贷**款**号**)**

**);**

贷款支付情况：

**create** **table** **支**付情况

**(**

贷**款**号 **INTEGER** **not** **null,**

身份证号 nvarchar2**(**20**)** **not** **null,**

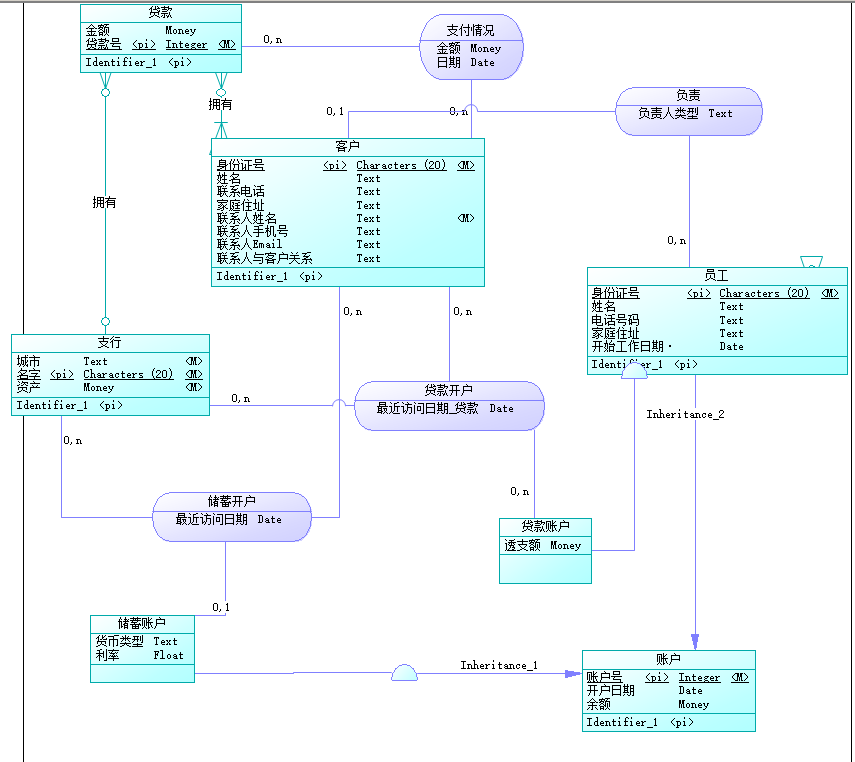
金额1 **NUMBER(**8**,**2**),**

日期付**款** **DATE,**

**constraint** PK\_**支**付情况 **primary** **key** **(**贷**款**号**,** 身份证号**)**

**);**

**数据库E-R模型图：**



# 3 详细设计

# 3.1 支行管理模块

输入：

城市名；

银行名字：

资产查询条件

资产数量

按钮点击事件类型

输出：

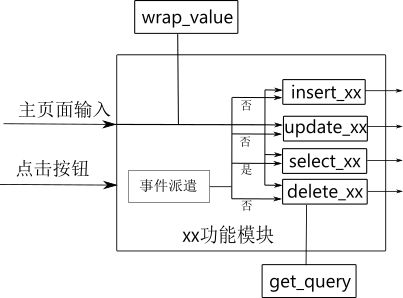
若为增加：无

若为删除：删除的表项显示在表格中

若为修改：无

若为查询: 查询结果显示在表格中

程序流程：



## 3.2 员工管理模块

输入：

身份证号

经理身份证号

姓名

电话号码

家庭住址

开始日期

按钮点击事件类型

输出：

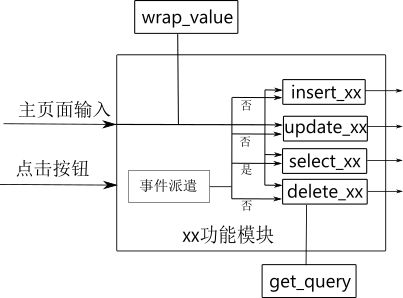
若为增加：无

若为删除：删除的表项显示在表格中

若为修改：无

若为查询: 查询结果显示在表格中

程序流程：



## 3.3 客户管理模块

输入：

身份证号

管理人身份证

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

联系人关系

按钮点击事件类型

输出：

若为增加：无

若为删除：删除的表项显示在表格中

若为修改：无

若为查询: 查询结果显示在表格中

程序流程：

同上

## 3.4 账户管理模块

输入：

账户号

开户日期

余额

利率

货币类型

透支额

按钮点击事件类型

输出：

若为增加：无

若为删除：删除的表项显示在表格中

若为修改：无

若为查询: 查询结果显示在表格中

程序流程：

同上

## 3.5 贷款管理模块

输入：

贷款号

贷款金额

支行名

发放金额

发放身份证号

按钮点击事件类型

输出：

若为增加：无

若为删除：删除的表项显示在表格中

若为修改：无

若为查询: 查询结果显示在表格中

程序流程：

同上

## 3.6 统计查询模块

输入：

统计查询账户类型

查询对应时间区间

输出：

储蓄统计结果下，不同支行在对应时间下的用户开户数和金额总数

贷款统计结果下，不同支行在对应时间下的用户开户数和金额总数

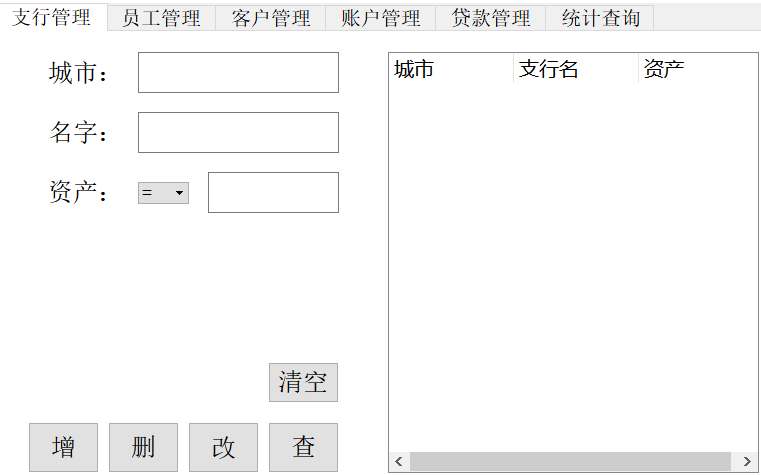
程序流程：

同上

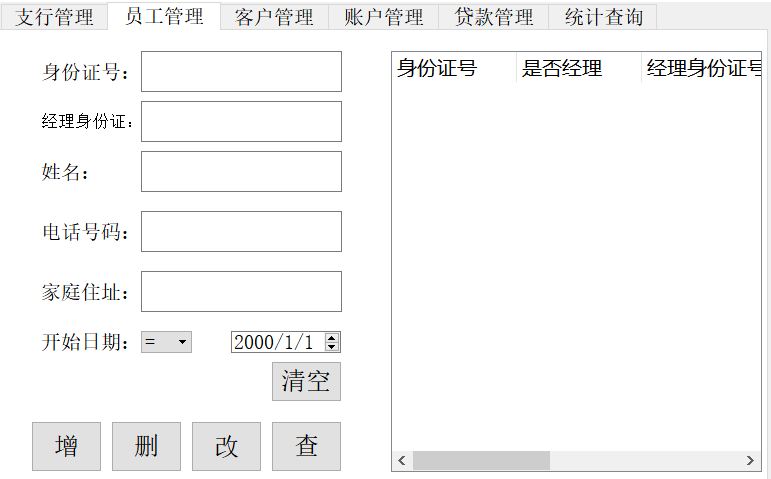
# 4 实现与测试

## 4.1 实现结果

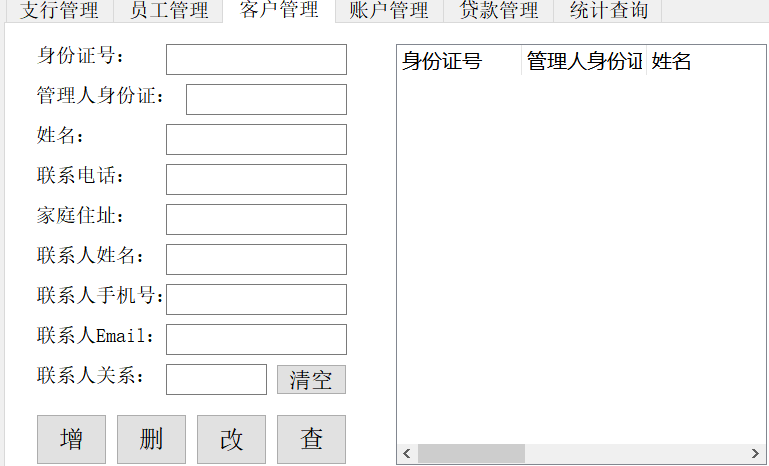
支行管理:



员工管理：



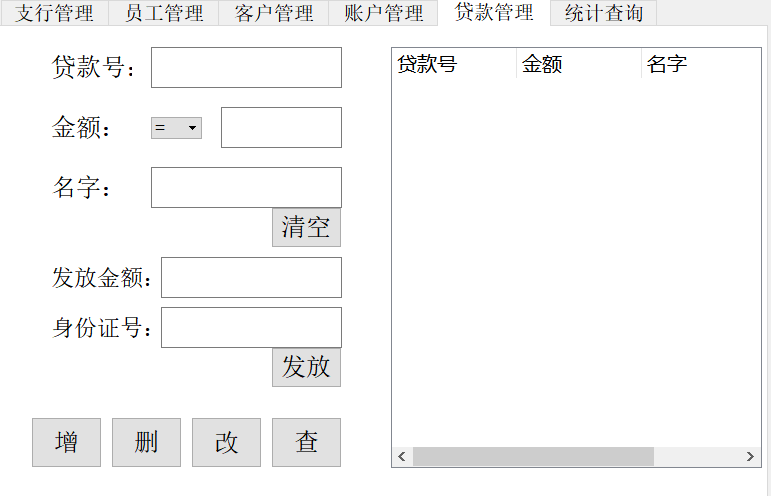
客户管理：



贷款管理：



贷款管理：



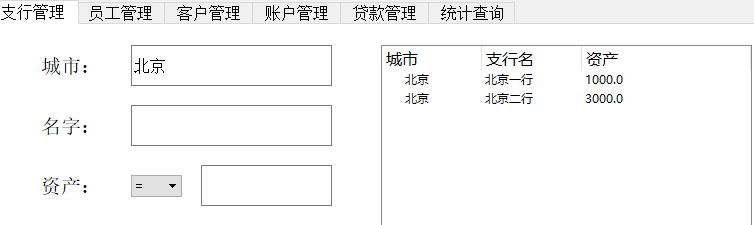
统计查询：



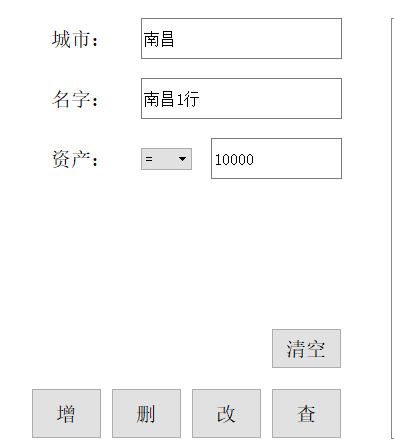
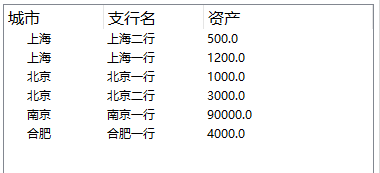
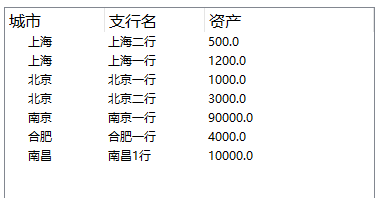
## 4.2 测试结果

1. 支行管理

查询：

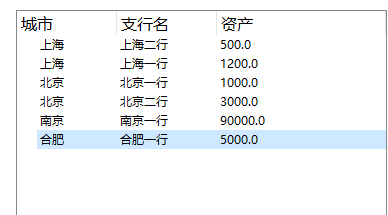
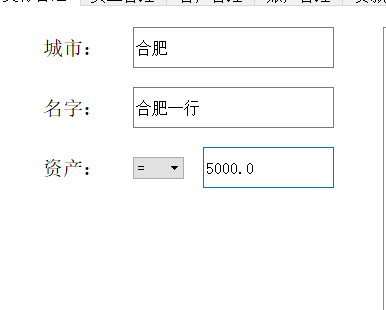


添加：



删除：

修改：



2. 员工管理

查询：



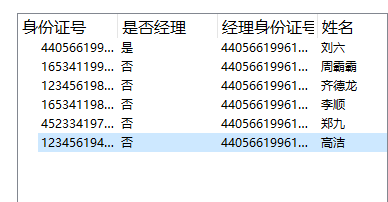
增加（员工高清）：



删除：



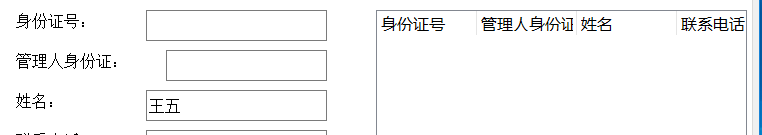
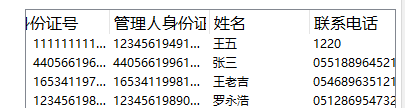
修改：(修改经理身份证号)



3. 客户管理

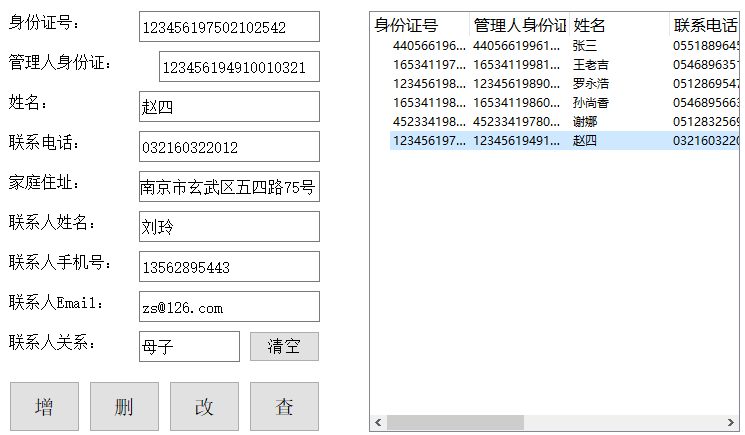
查询(无条件查询)：

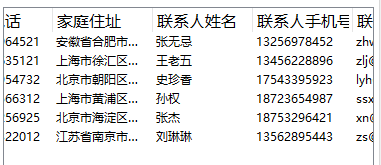
增加：



删除:

修改：（修改联系人姓名：）



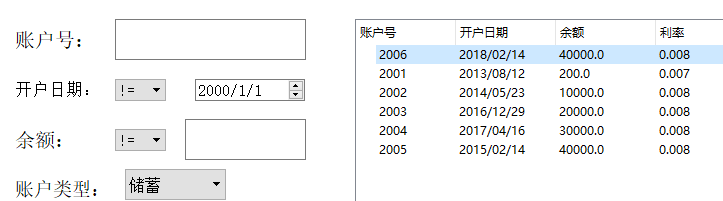


4. 账户管理：

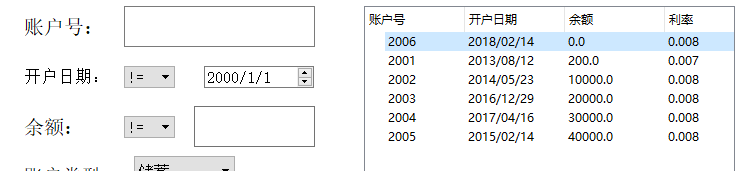


查询：

增加（储蓄账号2006）：



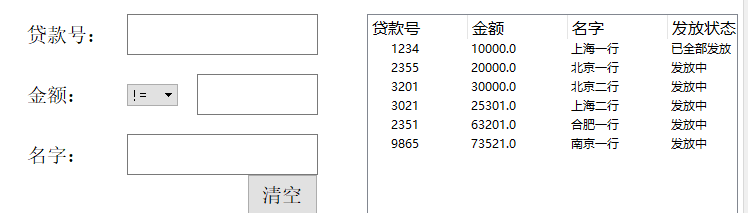
修改（修改储蓄账户2007的余额为0）



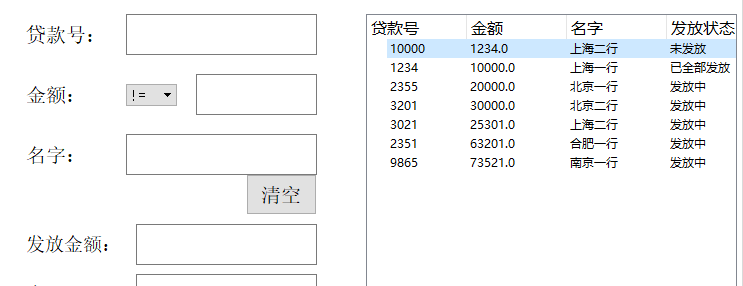
删除 (账户2006)

5. 贷款管理

查询：



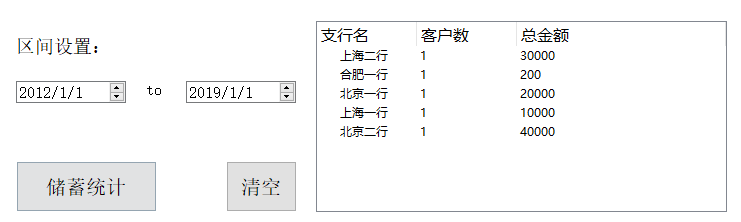
增加（增加一笔新贷款）



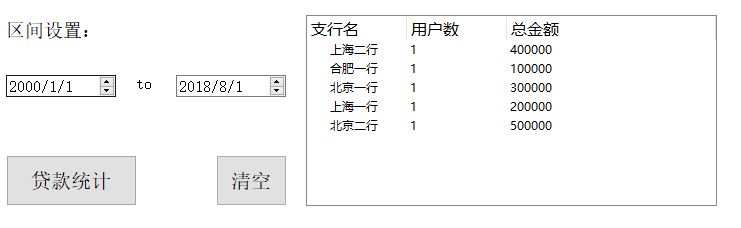
发放贷款：



统计查询：

储蓄查询：

贷款查询：



# 5 总结与讨论

1. 分析好需求，按需建立数据库和插入数据库条目。
2. 保证数据库的完整性，当出现需求分析不足时需修改数据库设计。
3. 界面设计尽量方便用户使用习惯，避免对用户造成不便和不适，例如，金额和日期查询信息默认均为！=号，以便用户快速进行无条件查询。
4. 开发和测试可以同时进行，首先保证单个界面的逻辑通畅，然后迁移到其他的逻辑界面上去。