**“银行业务管理系统”**

系统设计与实现报告

**姓名：袁一玮**

**学号：PB18000221**

**计算机科学与技术学院**

**中国科学技术大学**

2021年5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 4

2.3 数据库设计 5

3 详细设计 8

3.1 实验环境 8

3.2 客户管理模块 9

3.3 账户管理模块 9

3.4 贷款管理模块 10

3.5 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 18

4.3 实现中的难点问题及解决 22

5 总结与讨论 23

# 

# 1 概述

## 1.1 系统目标

设计一个银行业务管理系统，有客户管理、账户管理、贷款管理、业务统计等功能，满足银行业务管理的实际需求，保证数据的一致性，具有较强的鲁棒性和可用性。

## 1.2 需求说明

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。客户可以有账户，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账户负责人。 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。 银行提供两类账户——储蓄账户和支票账户。账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。每个账户被赋以唯一的账户号。银行记录每个账户的余额、开户日期、开户的支行名以及每个账户所有者访问该账户的最近日期。另外，每个储蓄账户有利率和货币类型，且每个支票账户有透支额。 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。

该程序要完成以下功能：

**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。

不需要进行对支行、部门和员工的信息维护。

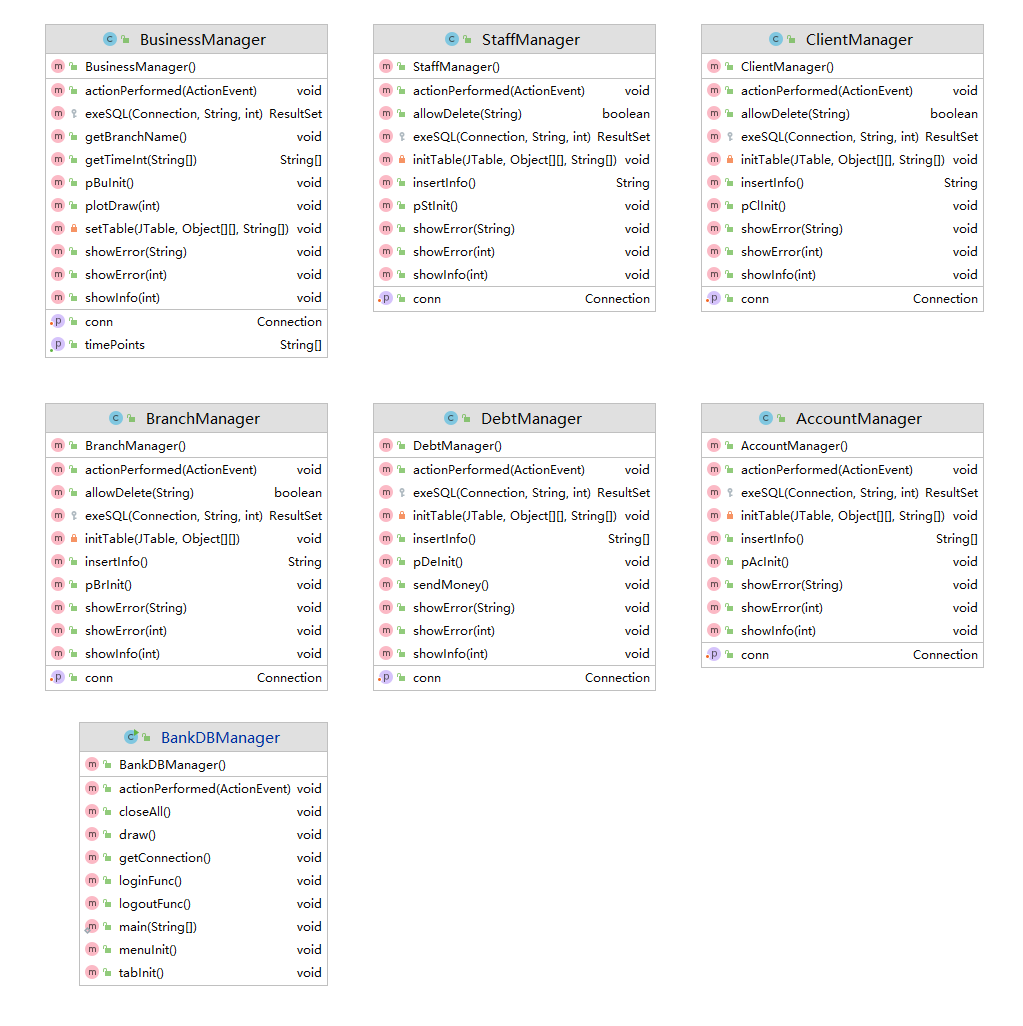
## 1.3 本报告的主要贡献

1. 提供了银行业务管理系统各项功能需求的方案
2. 提供了满足要求的数据库设计
3. 给出了各个功能的结构设计和模块关系
4. 详细说明了各个功能模块的具体设计
5. 给出了系统的测试结果
6. 为系统的使用提供参考

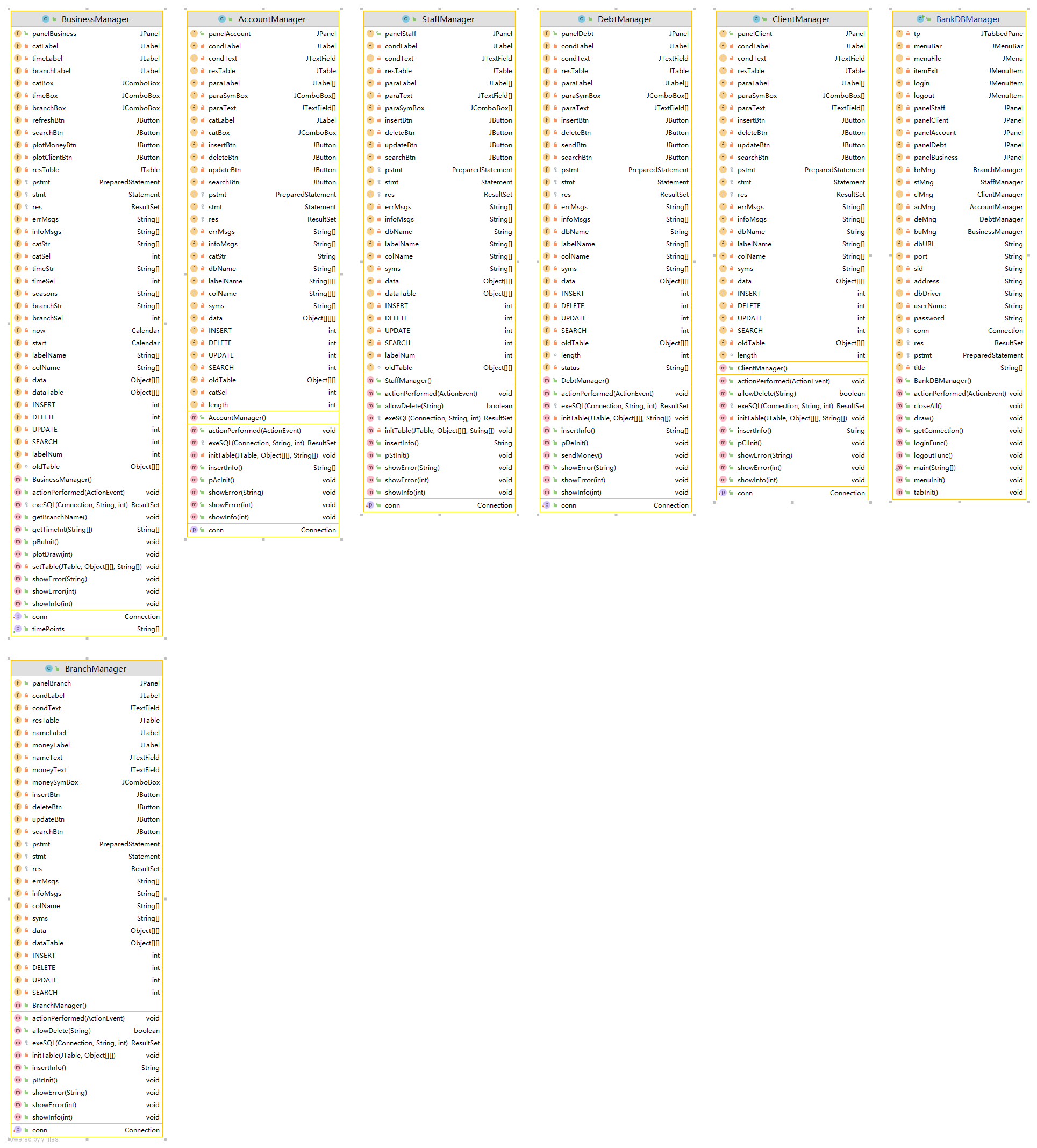
# 2 总体设计

## 2.1 系统模块结构

下面是IDEA给出的模块方法、参数UML用例图：

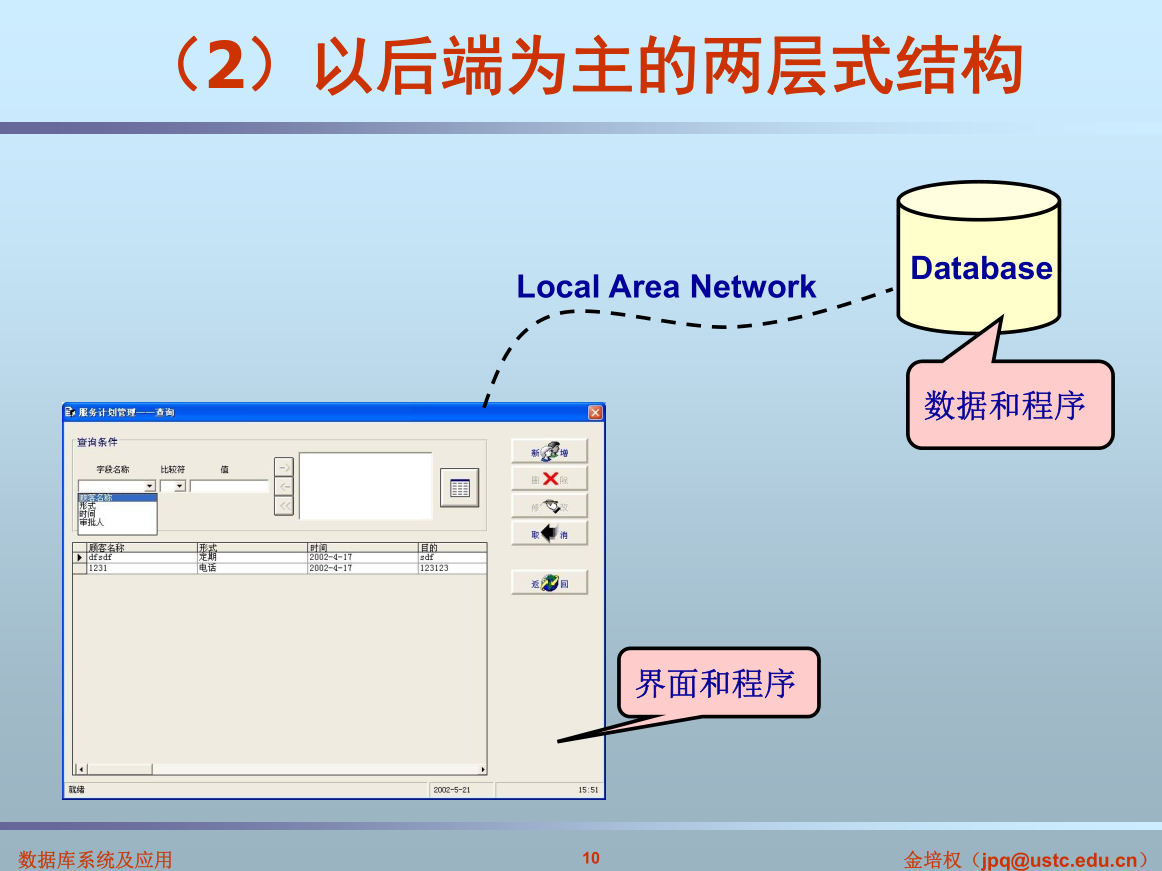


下面是IDEA给出的模块filed调用UML，可以看到BankDBManager实例化了客户管理ClientManager、账户管理AccountManager、贷款管理DebtManager和业务统计BusinessManager。



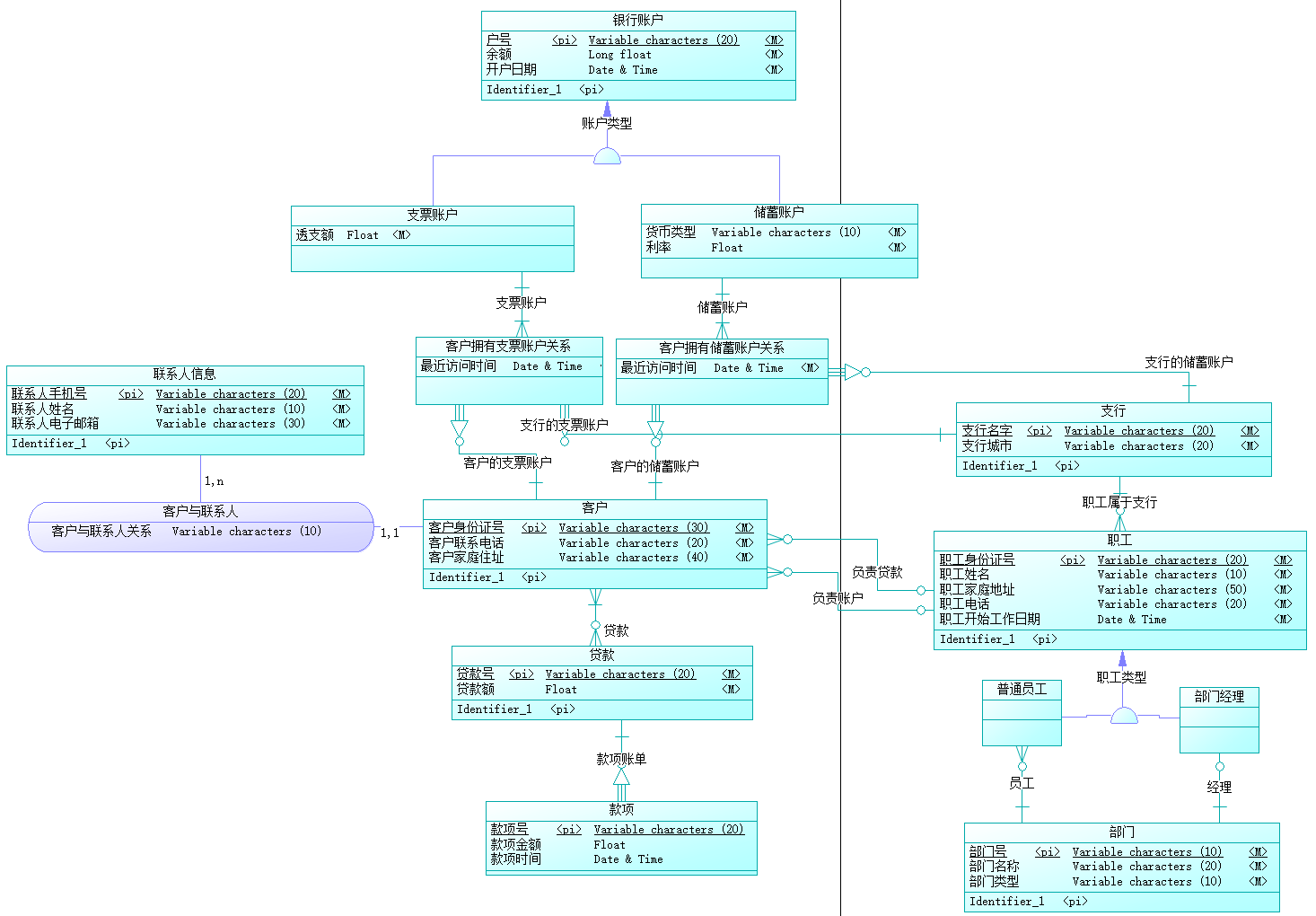
## 2.2 系统工作流程

参照PPT中两层式结构，Client程序去访问数据库，把用户的要求合成SQL语句去执行。

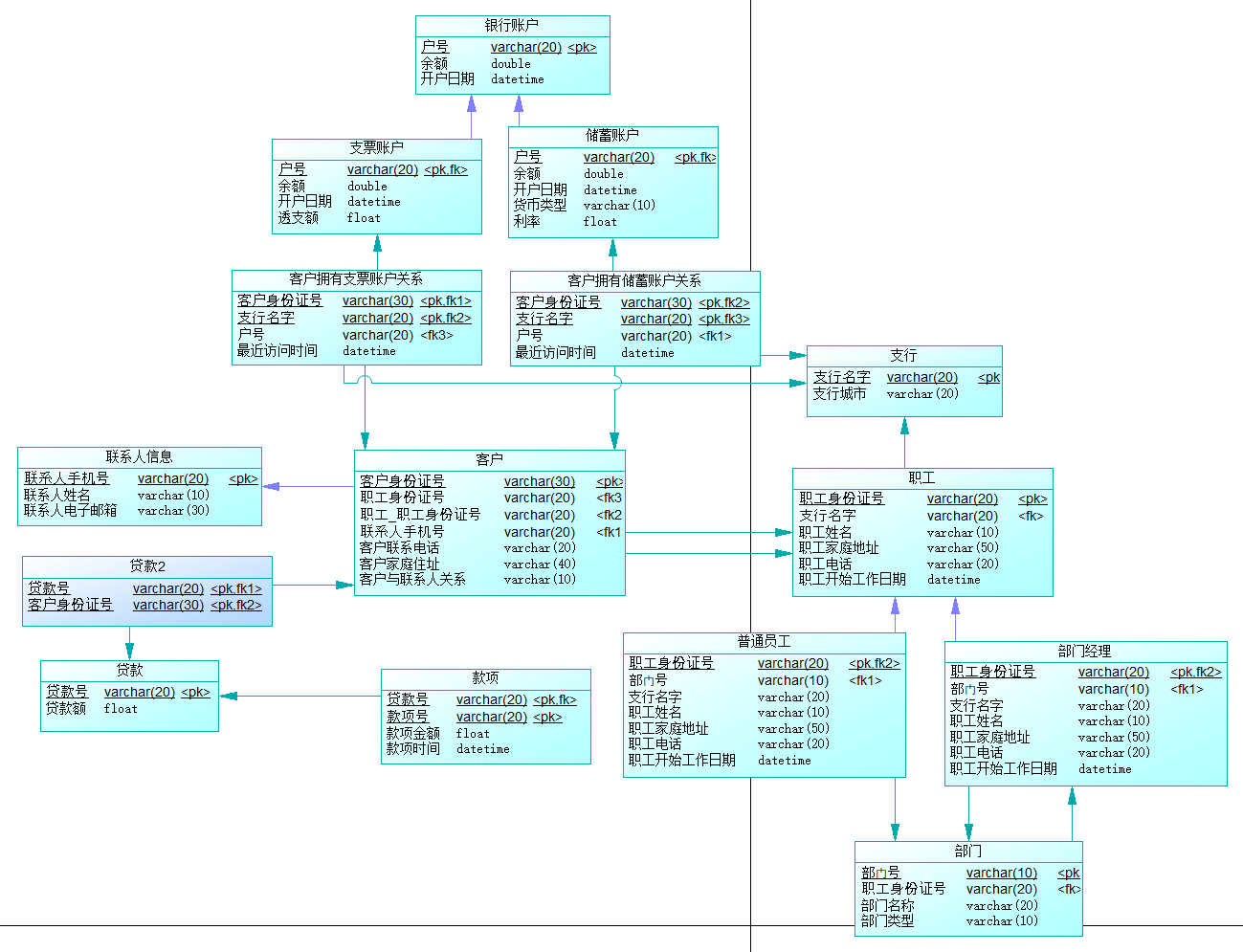


## 2.3 数据库设计

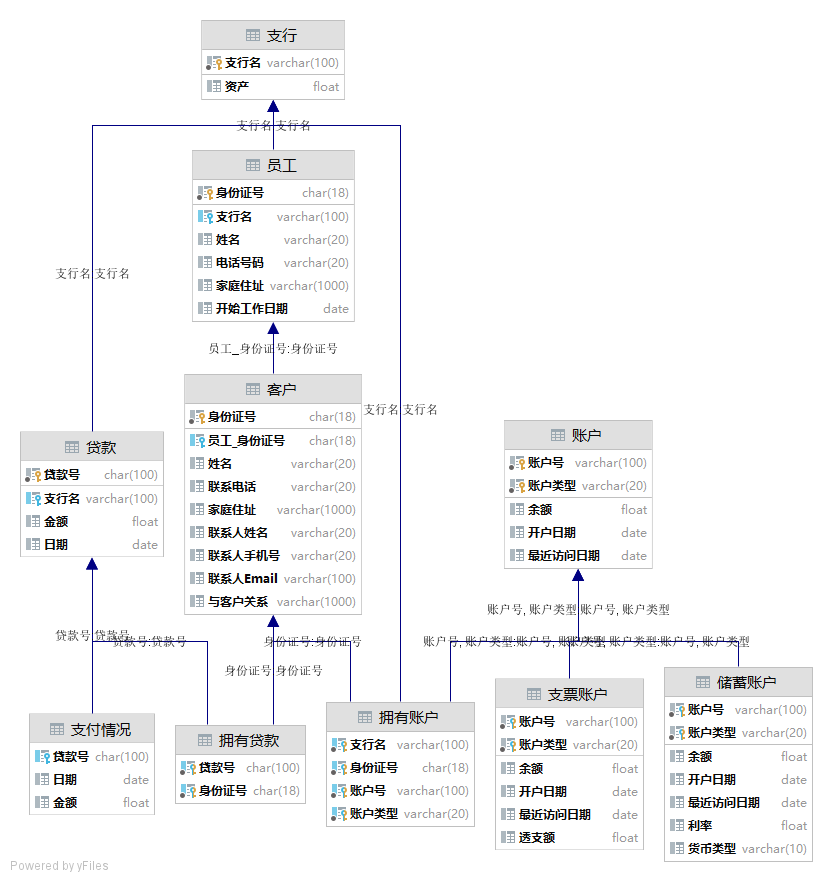
数据库的ER（CDM）图如下



逻辑结构（PDM）图如下



最后物理数据库结构如下（由IDEA导出）：



# 3 详细设计

## 3.1 实验环境

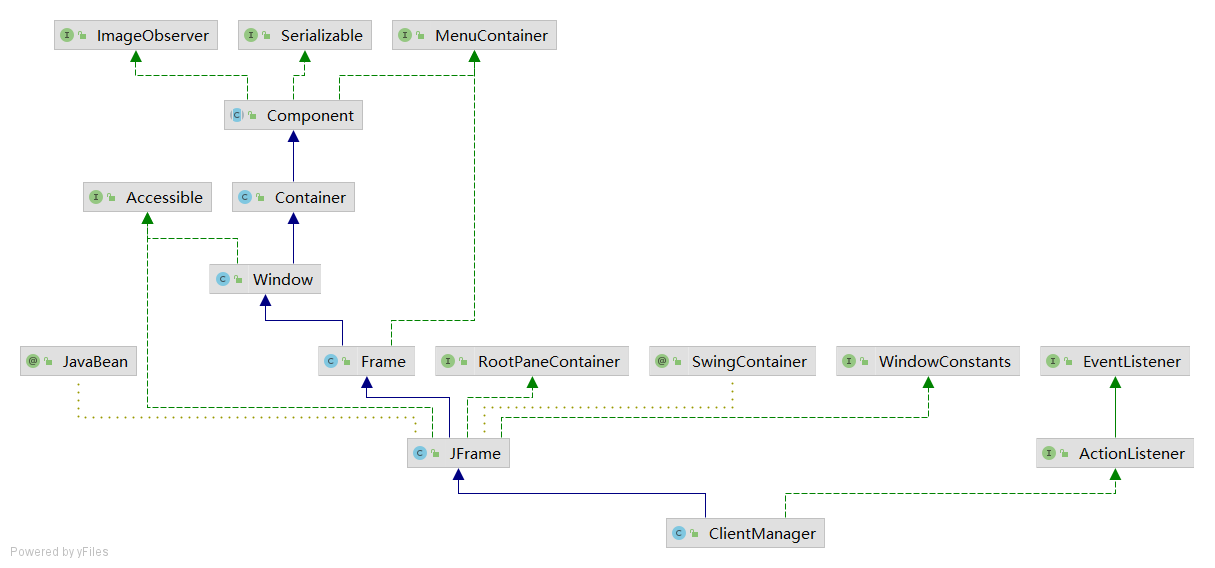
操作系统：Windows10

GUI框架：Java Swing

数据库：MySQL 8.0.25（使用官方Docker镜像）

## 3.2 客户管理模块

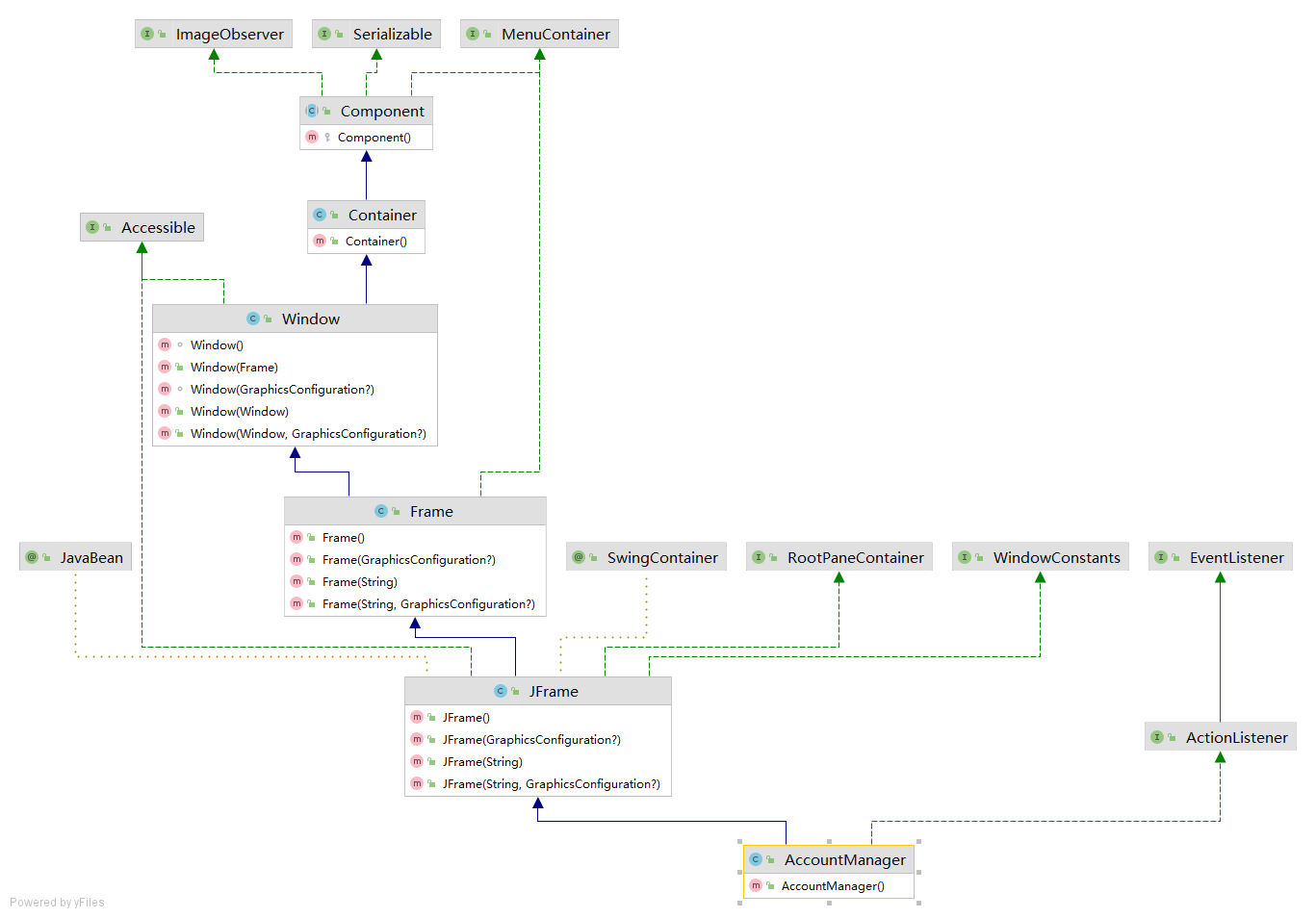
这个模块构造了JFrame去绘制表格来显示内容、绘制按钮来实现功能，其中ActionListener绑定在不同的按钮上。按钮完成的动作会被发射到监听的函数上，这和QT的监听信号槽有些类似。



在ActionListener中，定义了插入INSERT、删除DELETE、更新UPDATE和搜索查找SEARCH这四个类型的函数，下面三个模块类似。

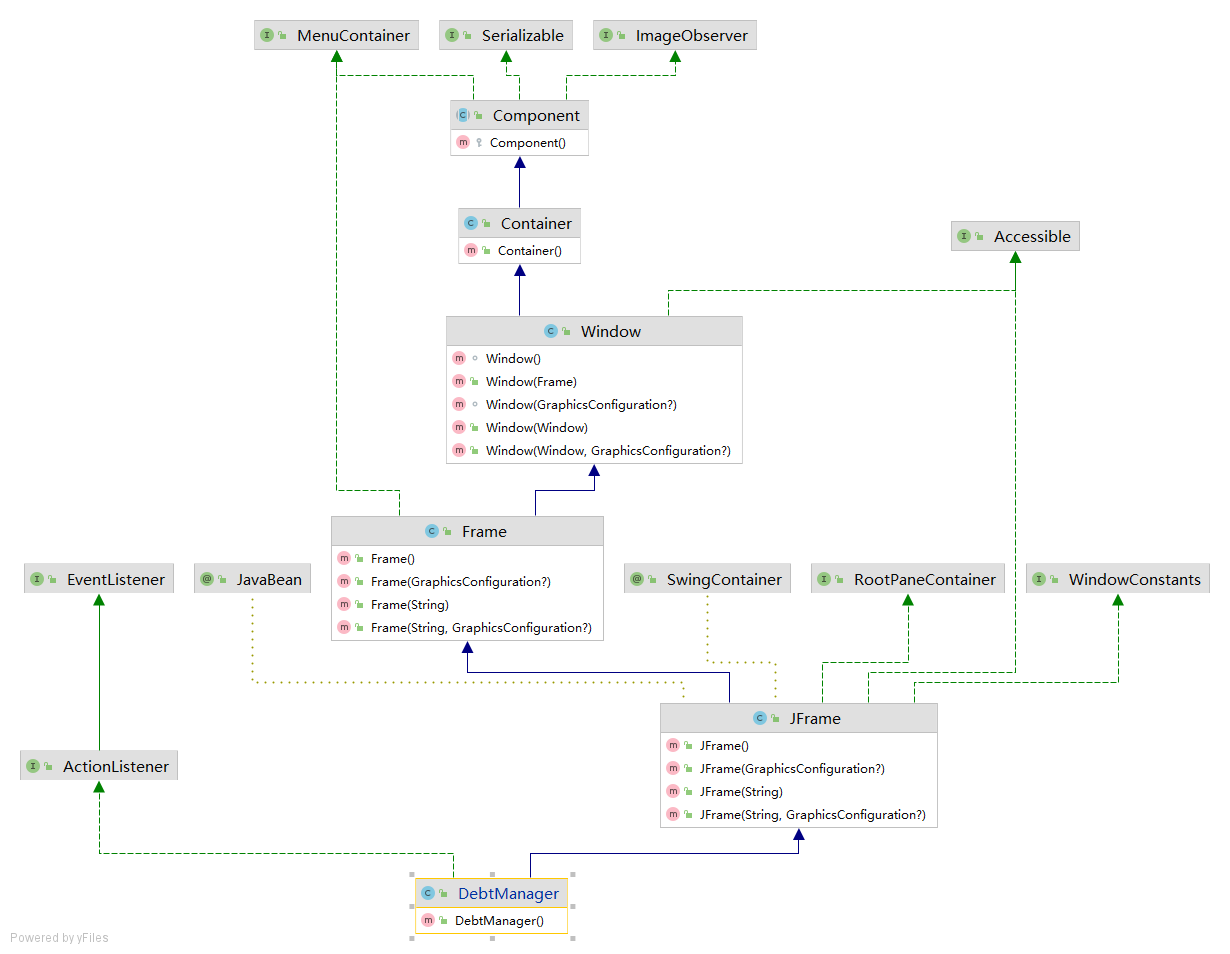
## 3.3 账户管理模块

和客户管理模块类似



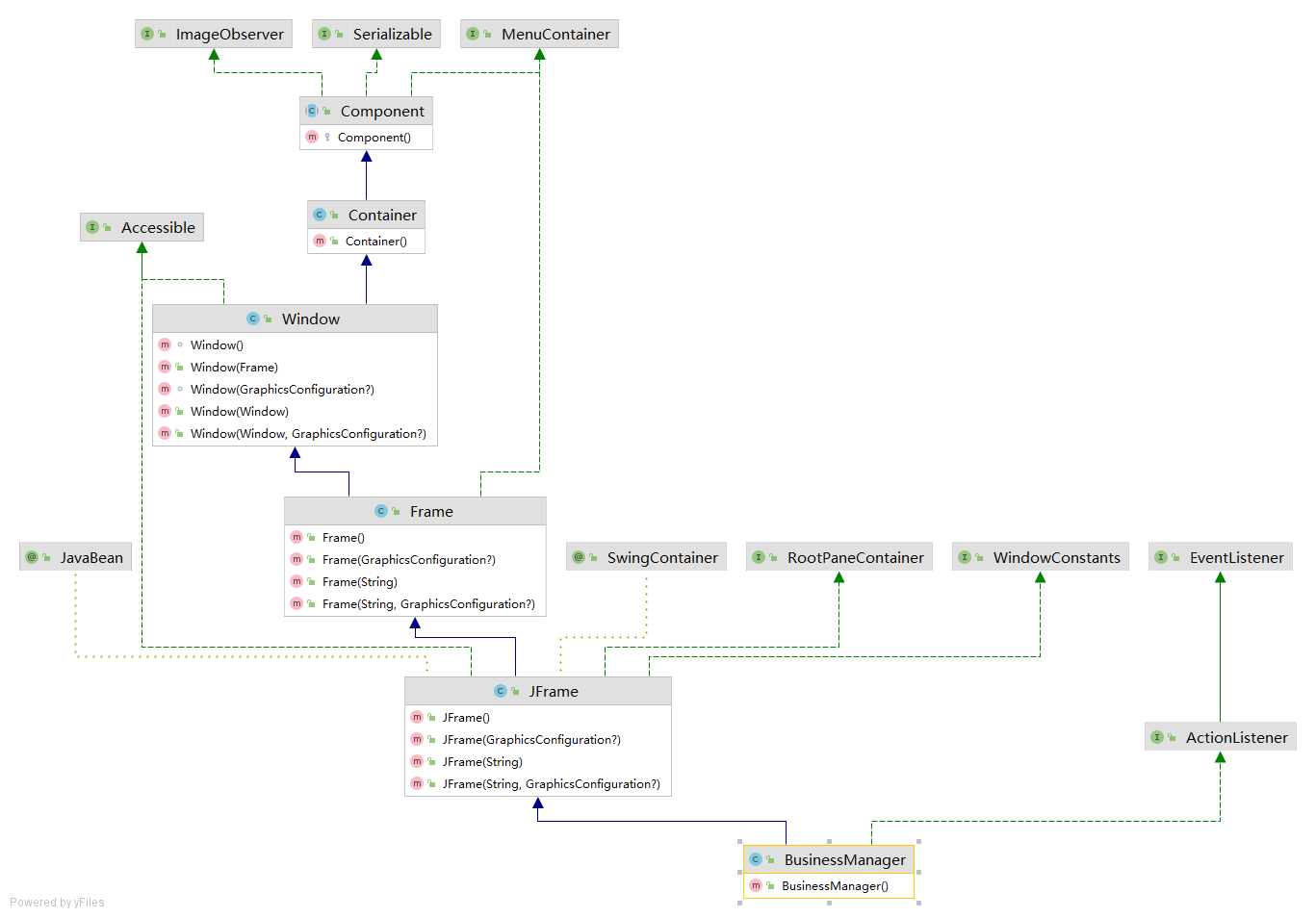
## 3.4 贷款管理模块

类似客户管理模块



## 3.5 业务统计模块

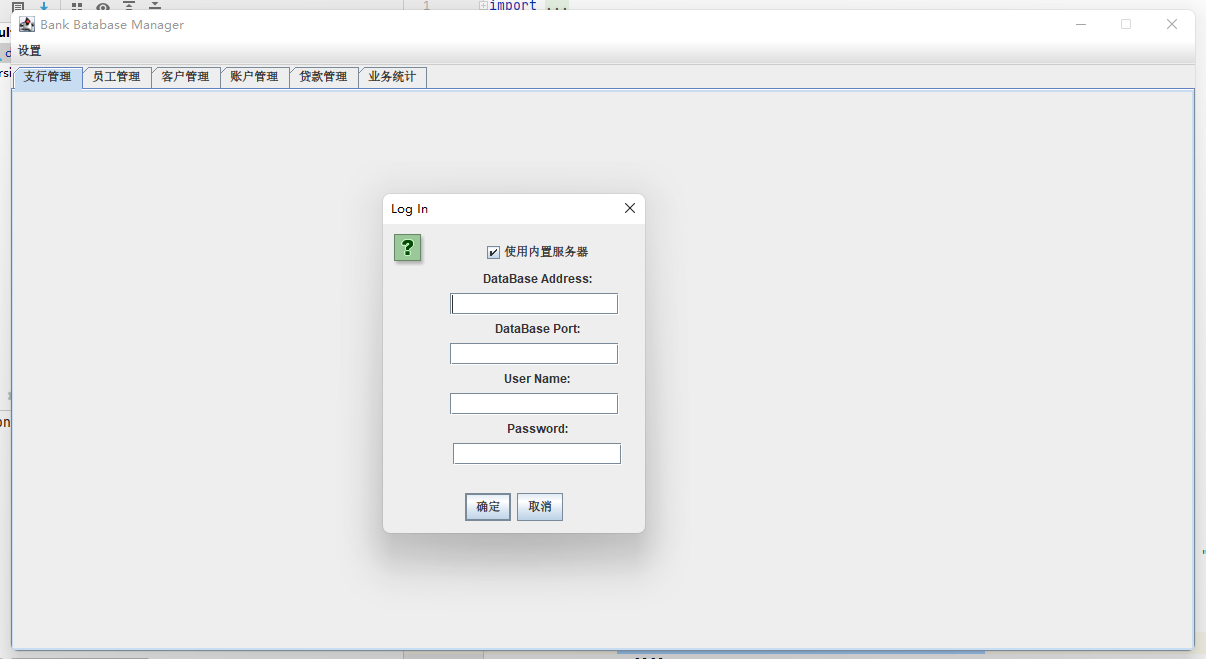
类似客户管理模块



# 4 实现与测试

## 4.1 实现结果

打开设置-登陆页面，进入LogIn界面。为了登录方便，我内置了一个MySQL的服务器，同时也可以取消勾选，使用用户给出的服务器。



具体实现如下：



在支行管理和员工管理中可以进行更新、修改支行和员工，这样比直接在数据库中修改更为方便，但这里不作介绍。

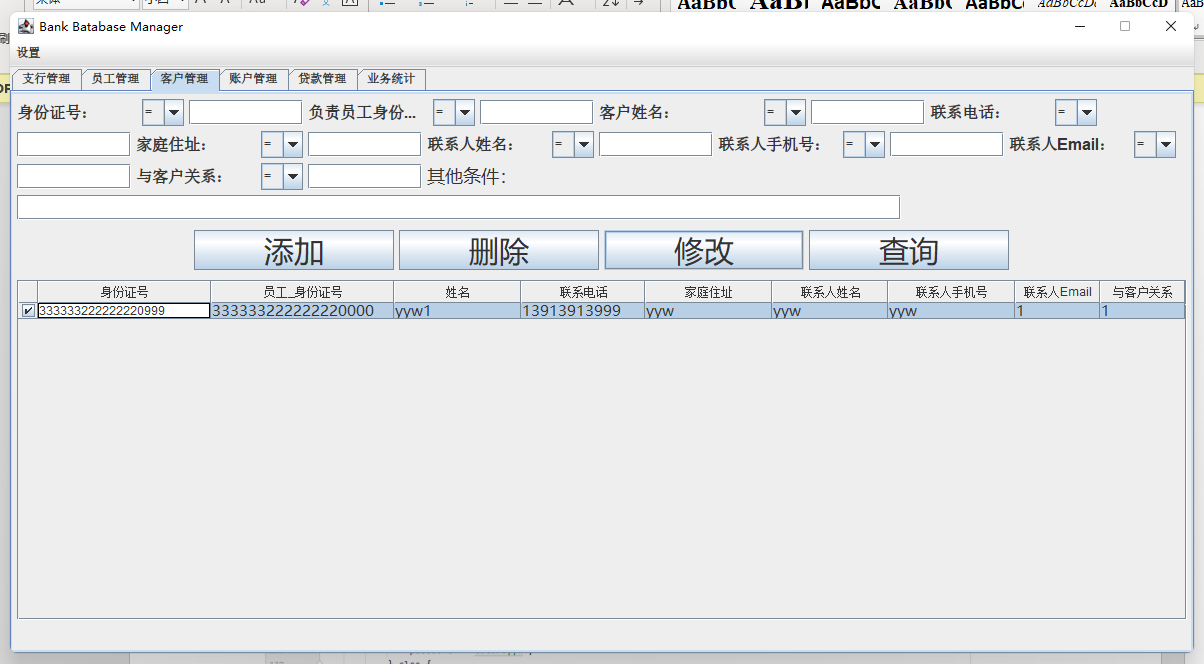
在客户管理中，点查询就可以调用“select \* from 客户”语句，同时JFrame放入插询后的内容。



点添加就可以调用输入框，把用户输入的内容转换成List<String>，同时我还对员工身份等外键信息作了错误处理。在Java的Console中，还可以看到拼接SQL语句的输出。

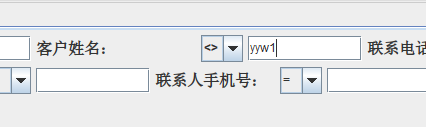


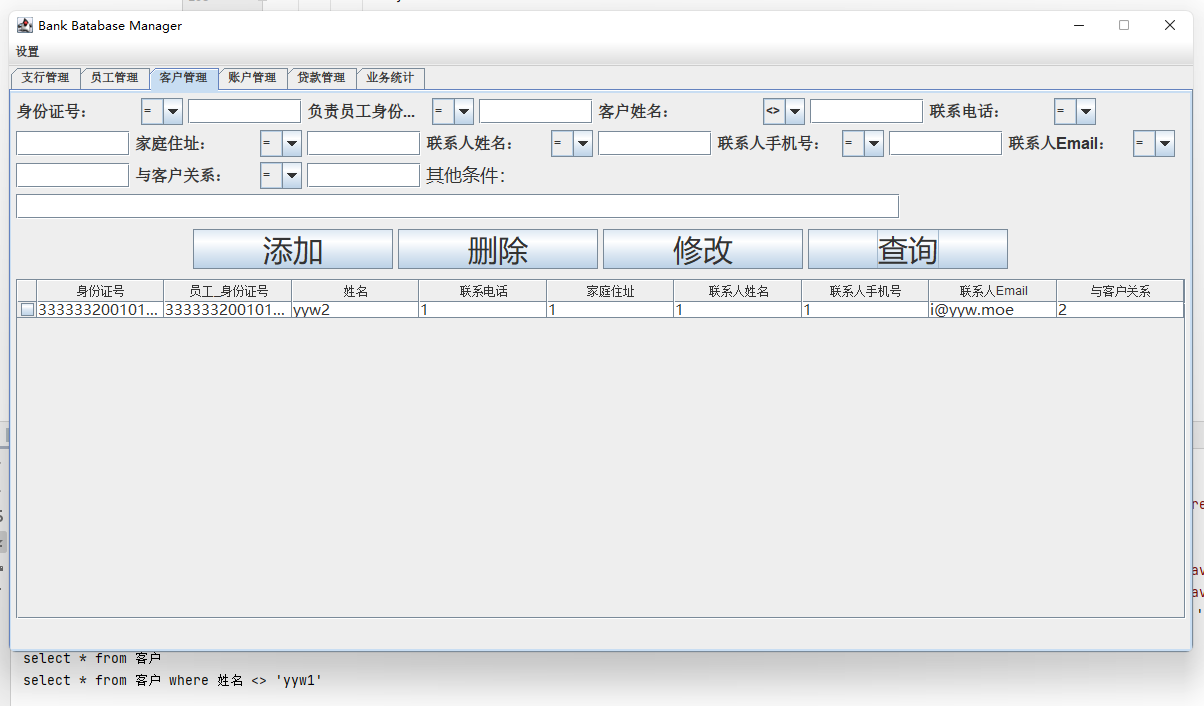
修改内容后勾选指定修改后的行，点修改就可以更新数据库信息。删除操作同理。



满足多样化的查询需求：在程序的上半框中有一系列布尔查询的选项，填入并选择比较方式后就可以进行布尔查询。



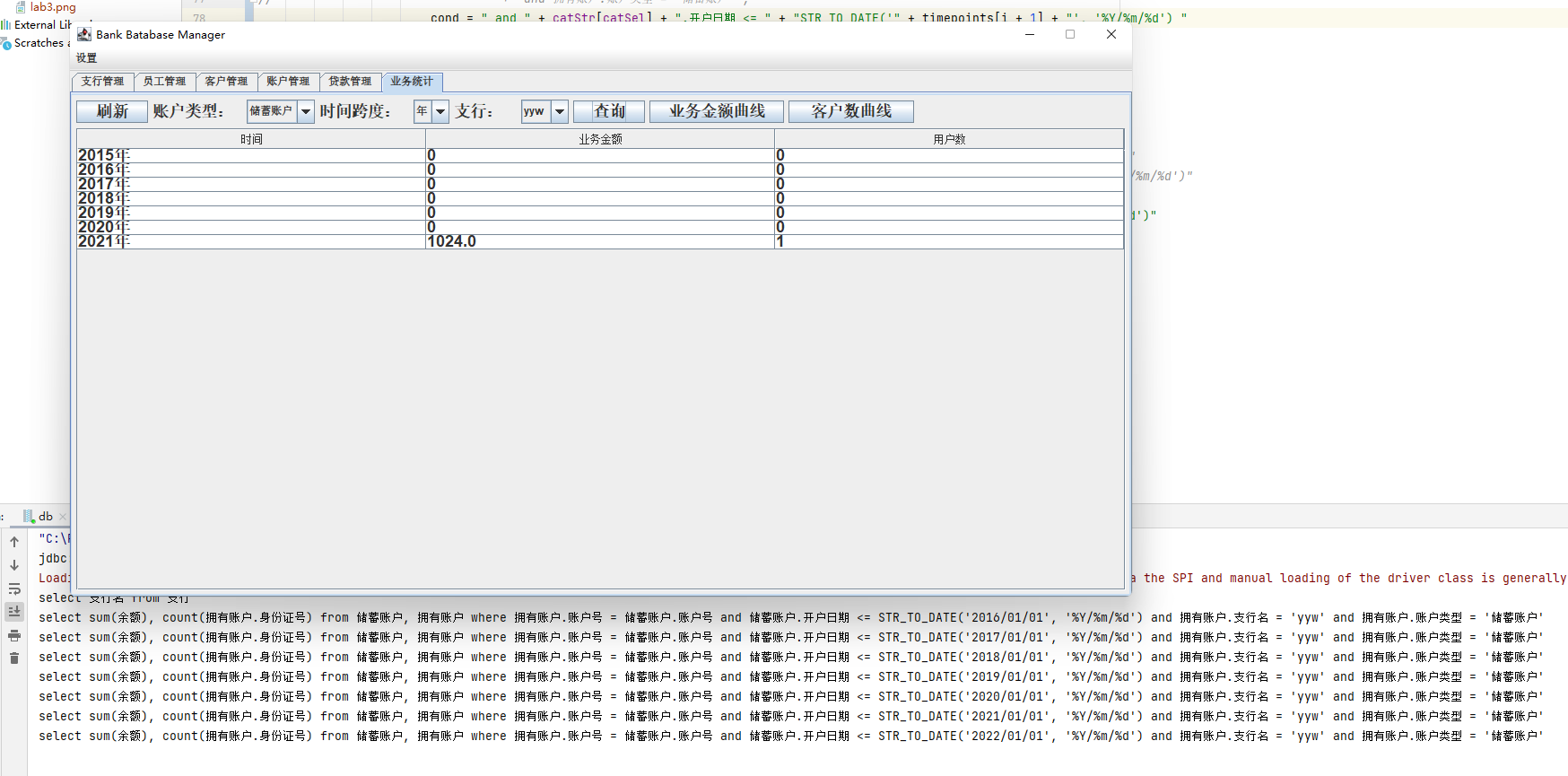




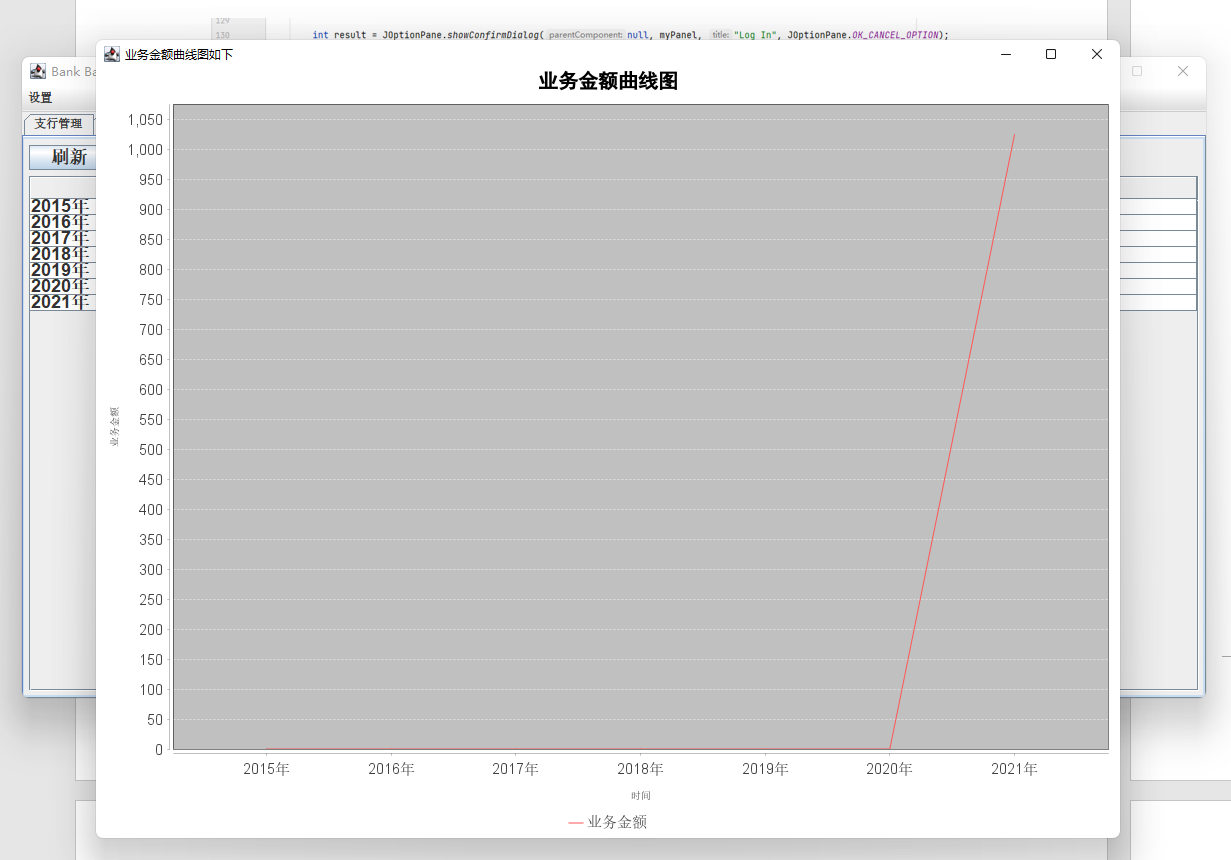
从程序在Console打印的SQL中看出，程序对***客户姓名***这个字段进行了布尔查询，满足多样性查询的要求。

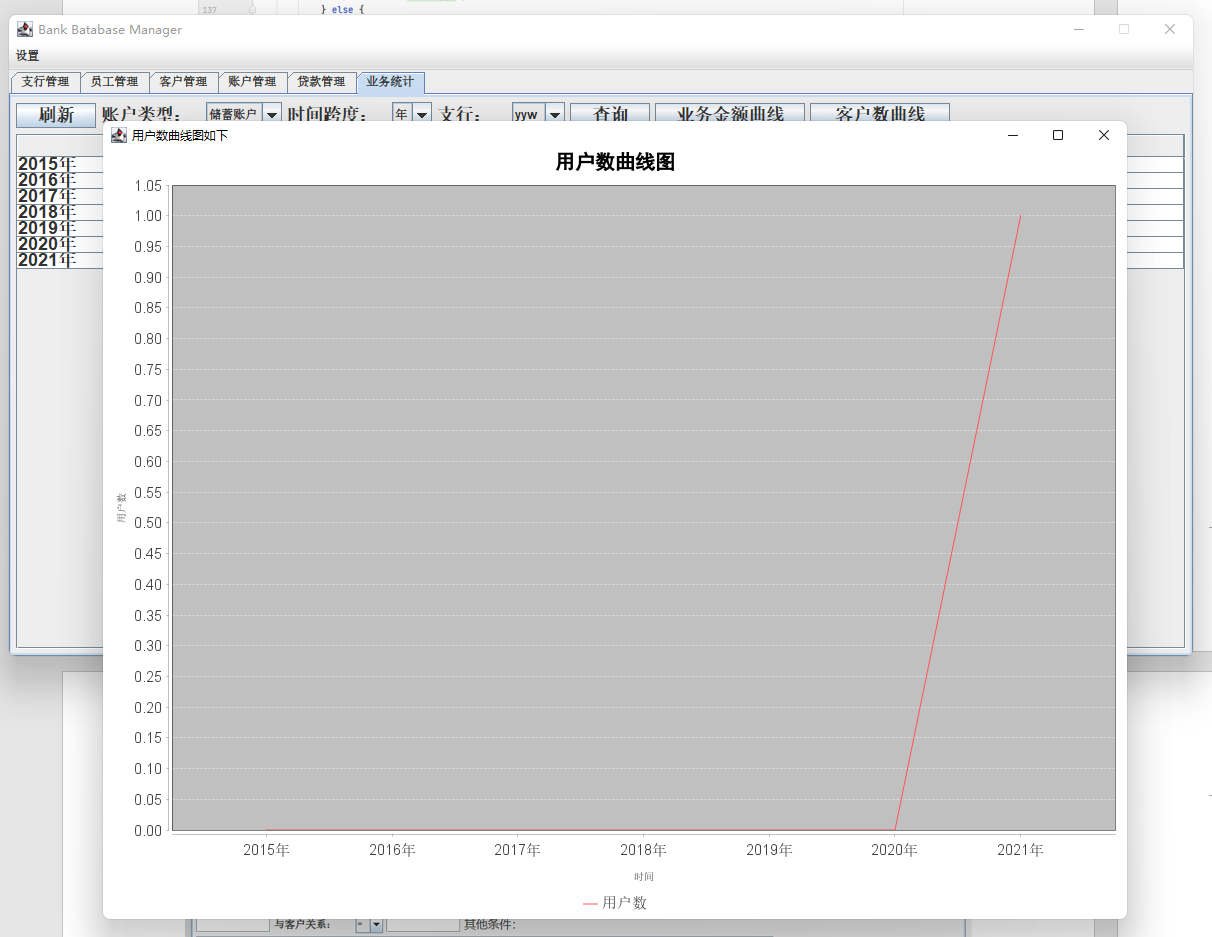
账户和贷款两个模块效果类似，接下来是业务统计：

在进行查询的时候，程序对符合的账户进行统计，同时程序中留下了金额和用户数的List。



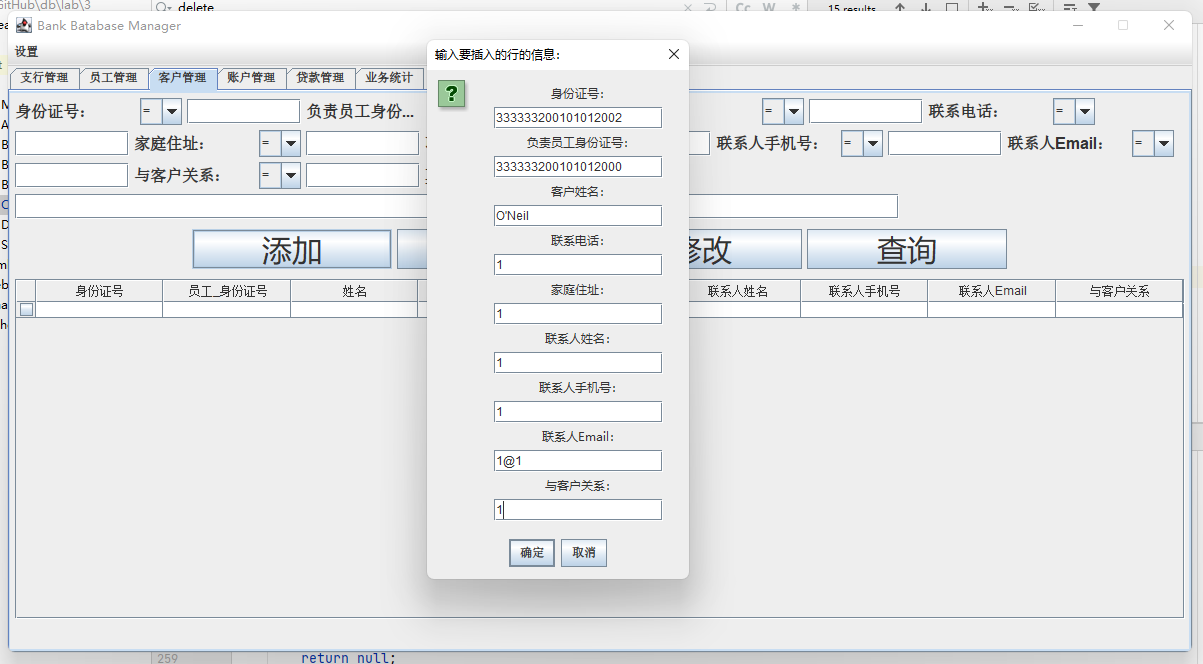
根据查询结果来绘制曲线：



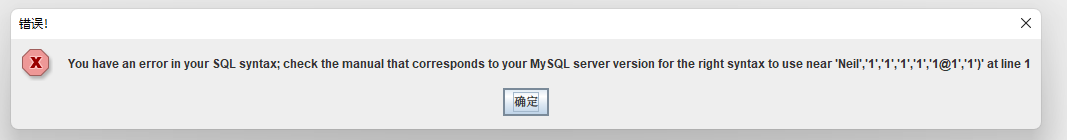


## 4.2 测试结果

**若在名字字段中插入带’的内容，由于这个很容易被ORM框架所转义，这里便不再做过多工作：**



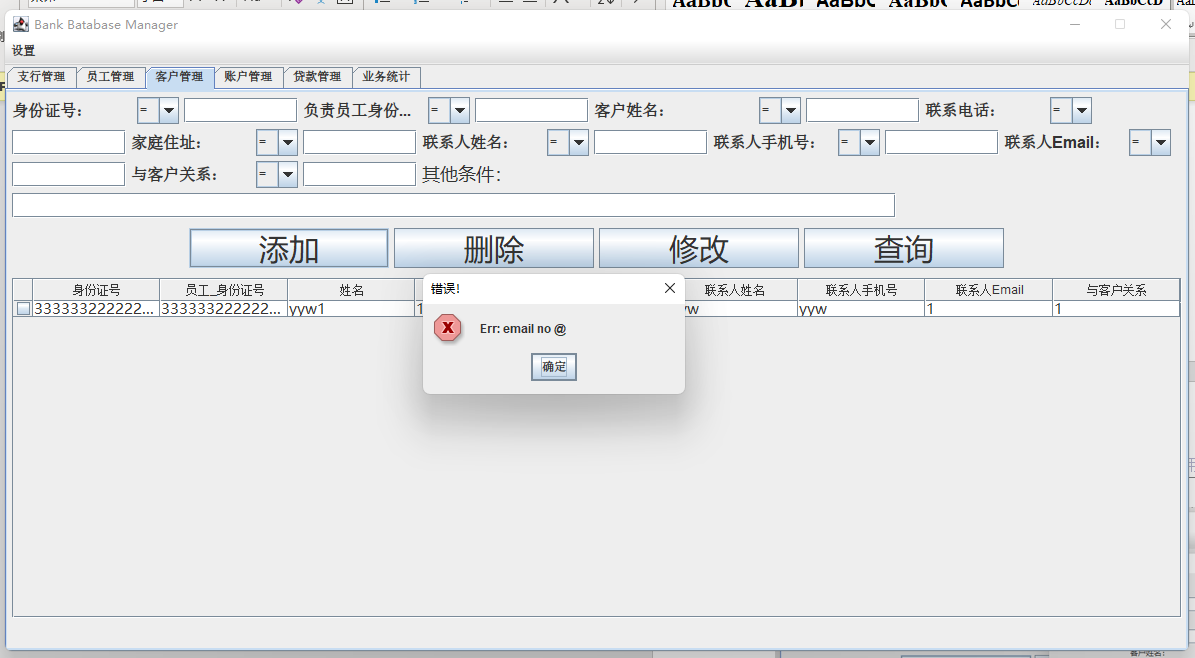
可以看到，程序封装了MySQL的报错：



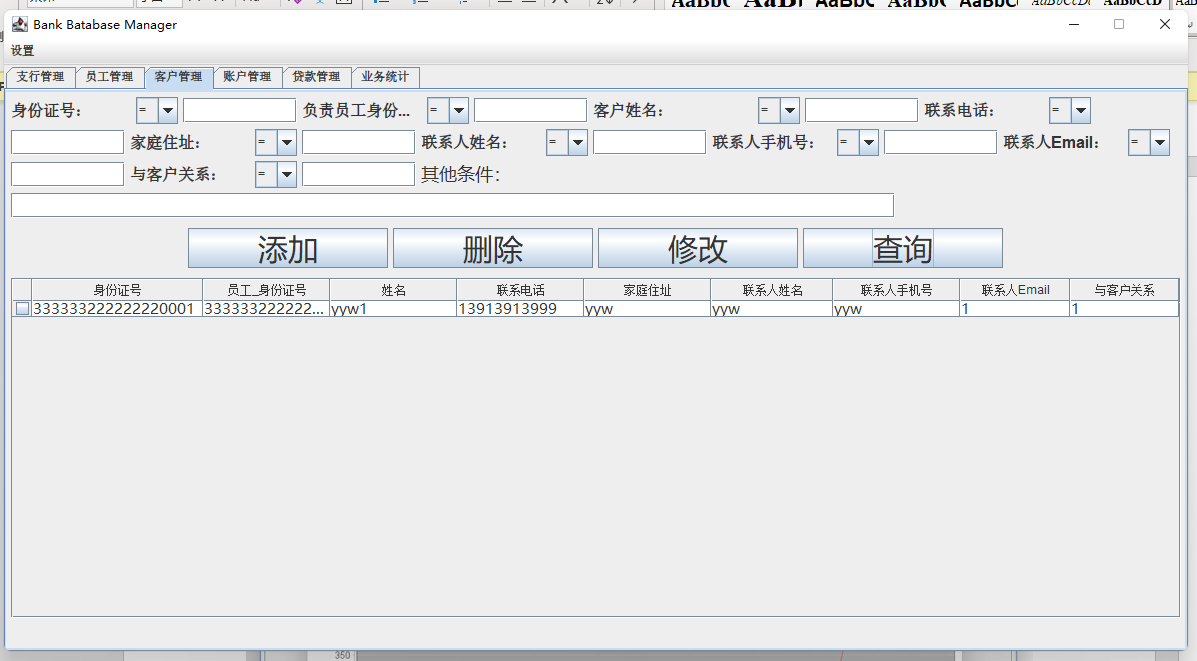
**若对Email字段插入非Email的格式，比如没有@截断符：**

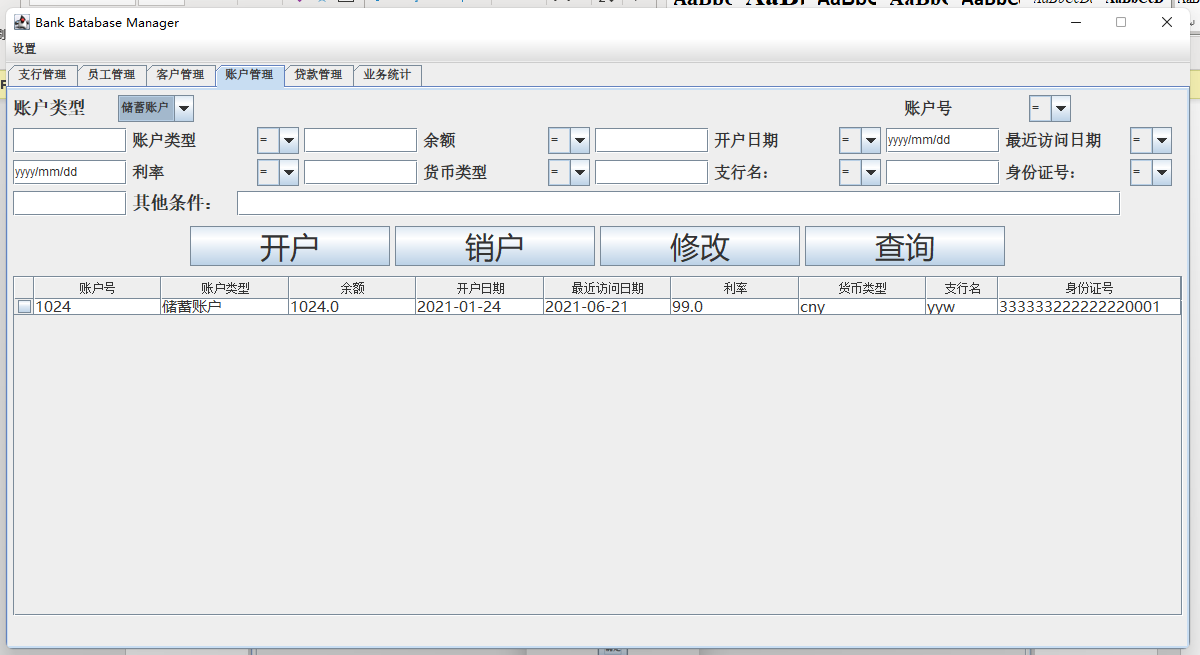


可以看到程序给出错误弹窗：

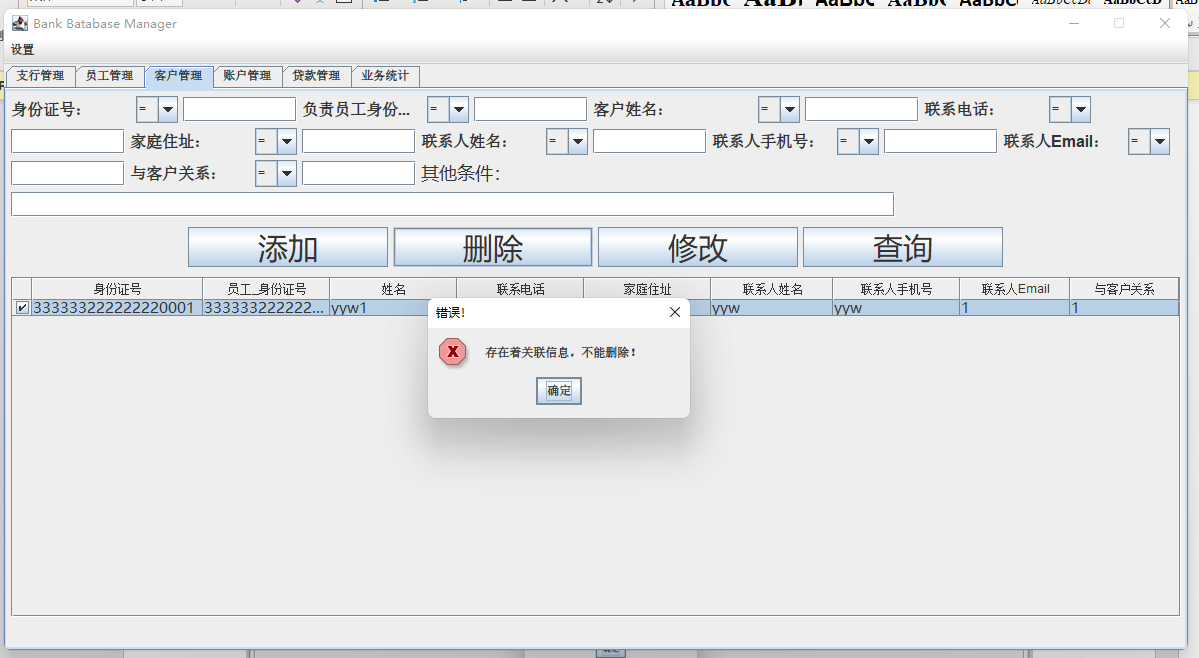


**存在关联账户或贷款记录则不允许删除客户信息：**

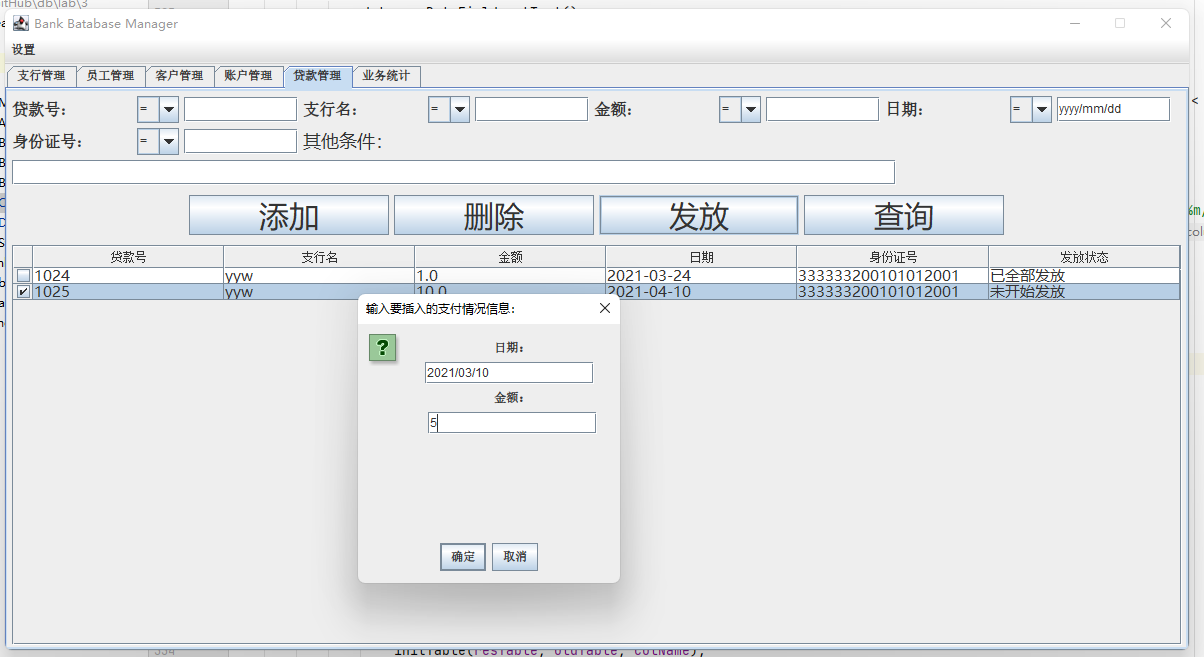




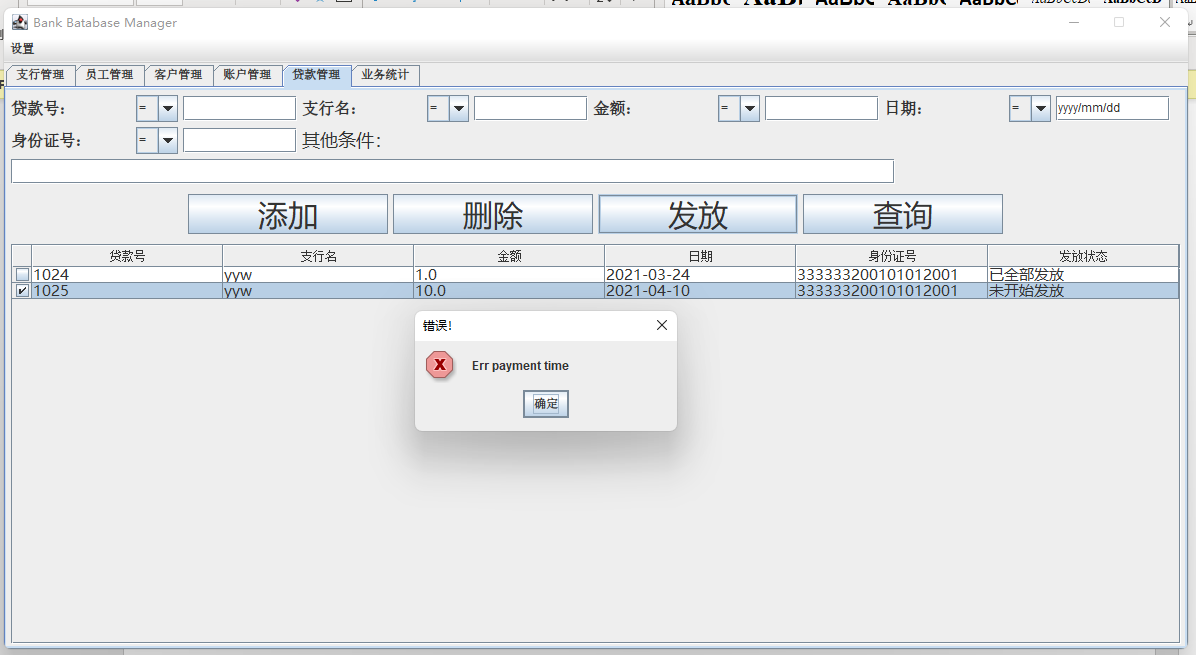
在客户中点击删除，可以看到操作被阻止：



**若贷款发放时间比贷款生效时间早，则应该被阻止：**



则会被程序阻止：



## 4.3 实现中的难点问题及解决

Golang程序只能用官方给出的gORM框架进行查询，并不能直接执行SQL，这和罗永平助教的要求不符，对我开始时造成了极大的困扰，所以放弃了“Web React前端-Gin后端-MySQL数据库系统”这样的结构。

在Java中，错误处理和捕捉是一件很好且很方便的事情，只要Try-Catch语句就可以对错误进行处理或者把错误传递到外层，我使用了一个模块去封装了错误的输出和打印。这比Golang的Err错误传值要方便许多。

关于图形化的实现，Java Swing的GUI比Csharp的WPF之流要丑上不少，布局也不方便，既没法用WPF一样的控件拖动也没法像React一样使用CSS的Fluid Grid来控制布局，只能将就了。

# 5 总结与讨论

在本实验之前，我还用B/S架构和golang实现了后端，使用了gorm对接数据库、用go的高性能HTTP服务器gin承载网站，前端打算用React。但罗永平助教说不能使用ORM框架，而go的MySQL驱动并不支持直接执行语句（要很多错误检查）。于是我只能放弃原有的实现，重新用Java Swing写了本C/S架构的程序。