

# 银行业务管理系统 系统设计与实验报告

- 姓名：谭泽霖
- 学号：PB18010454

## 概述

### 系统目标

本实验采用C/S的形式，目的是为了构建一个银行管理系统，完成对顾客信息、账户信息和贷款信息的增删改查，同时后台数据库还需维护和保持这三类数据之间的一些约束。

### 需求说明

- **数据库：**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。
- **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，要求对统计结果同时提供表格和曲线图两种可视化展示方式。

### 本报告的主要贡献

本报告主要有以下贡献：

- 对发布的需求进行了进一步的分析与与扩充，对所需要实现的功能进行拆分和合并。
- 介绍满足给出需求的一个前后端网页应用的设计思路和实现细节
- 对上述实现的系统进行了较为完备的测试

# 总体设计

## 系统模块结构

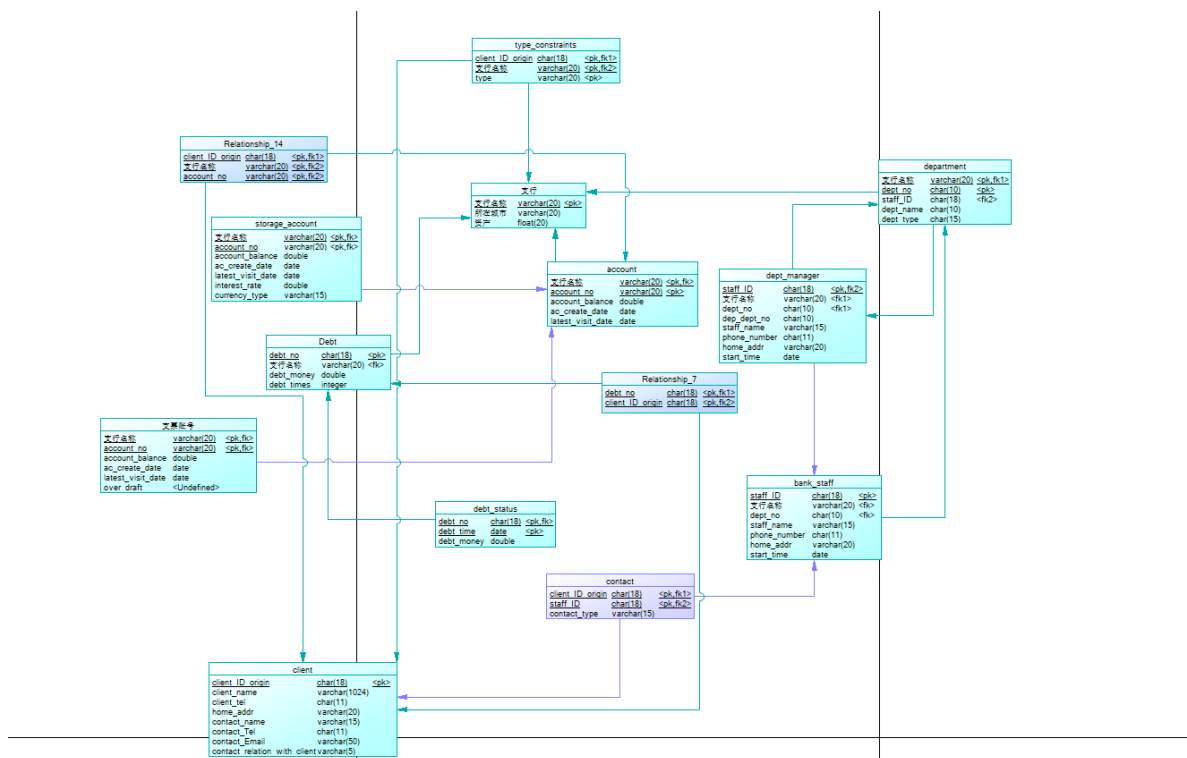
- **前端**：主要分为7个模块，分别是客户信息搜索删除、客户信息添加更新、账户搜索删除、账户添加更新、贷款搜索、贷款添加和业务统计界面，每个界面分别对应需求中一大类功能，并提供了一些按钮和输入框来与用户交互。
- **后端**：主要分为两个模块，数据库模块和接口（API）模块，数据库模块中定义了Mysql数据库中各个table中的字段的类型以及table之间的外键关系；接口模块负责接收、处理并回复来自前端的信息。

## 系统工作流程

当用户进入到银行系统的页面后，可以在四种主要的功能中进行选择和切换。当输入了满足要求的数据并点击相应按钮后，前端向后端发送用户填入的所有数据，服务器在解析用户的信息后，按照对应的操作检查参数是否合法，若合法则执行对应的操作并返回执行后的状态信息到前端服务器，前端服务器将接收到的消息显示在窗口上，从而完成一次与服务器的交互。

## 数据库设计

本实验数据库的设计完全照搬本人的lab2，在此不再赘述，仅放两张截图：





[illegible]

```

49         else:
50             res.append(tab)
51     else:
52         if(not live_addr.isspace() and len(live_addr) !=
0):
53             if(live_addr.lower() == str(tab[3]).lower()):
54                 res.append(tab)
55             else:
56                 res.append(tab)
57     else:
58         if(not tele_num.isspace() and len(tele_num) != 0):
59             if(tele_num.lower() == str(tab[2]).lower()):
60                 if(not live_addr.isspace() and len(live_addr) !=
0):
61                     if(live_addr.lower() == str(tab[3]).lower()):
62                         res.append(tab)
63                     else:
64                         res.append(tab)
65                 else:
66                     if(not live_addr.isspace() and len(live_addr) != 0):
67                         if(live_addr.lower() == str(tab[3]).lower()):
68                             res.append(tab)
69                     else:
70                         res.append(tab)
71
72
73     cursor.close()
74     return res

```

## main\_2.py

在这个文件中我实现了所有的接口定义，通过它将前端和后端数据库连接在一起

```

1  from MySQLdb import Connect
2  from PyQt5.Qtwidgets import QApplication, QMessageBox
3  from PyQt5.Qtwidgets import QMainWindow, QMessageBox
4  from PyQt5.Qtwidgets import QTableWidgetItem, QHeaderView
5  from PyQt5.Qtwidgets import QAbstractItemView
6  from PyQt5.QtGui import *
7  from PyQt5 import QtCore
8  from UI.Ui_fristwindow import Ui_Firstwindow
9  from db import *

```

```

1  class MainWindow(QMainWindow):
2      def __init__(self):
3          super().__init__()
4
5          self.ui = Ui_Firstwindow()
6          self.ui.setupUi(self)
7          #菜单栏动作绑定
8          self.ui.statusbar.showMessage("status:ok")
9          self.ui.actionSearch.triggered.connect(self.clientSearch_page)
10         self.ui.actionAdd_Update.triggered.connect(self.clientAdd_page)
11         self.ui.actionSearch_2.triggered.connect(self.accountSearch_page)
12         self.ui.actionAdd_Update_2.triggered.connect(self.accountAdd_page)

```

```
13 self.ui.actionAdd_Clients.triggered.connect(self.accountAdd_Client_page)
14     self.ui.actionSearch_3.triggered.connect(self.loanSearch_page)
15     self.ui.actionAdd_Update_3.triggered.connect(self.loanAdd_page)
16
17 self.ui.actionStatistics.triggered.connect(self.statistics_business)
18
19     #客户查询/删除界面设置及按键绑定
20
21 self.ui.tablewidget.setSelectionBehavior(QAbstractItemView.SelectRows)
22     self.ui.tablewidget.setColumnWidth(6, 200)
23     self.ui.tablewidget.setColumnWidth(3, 200)
24     self.ui.tablewidget.setColumnWidth(0, 200)
25     self.ui.pushButton.clicked.connect(self.Searchbtn_client)
26     self.ui.pushButton_3.clicked.connect(self.Deletebtn_client)
27
28     #客户新增/更新界面设置及按键绑定
29     self.ui.radioButton_3.setChecked(True)
30     self.ui.radioButton_5.setChecked(True)
31     self.ui.radioButton_3.toggled.connect(self.Stack_3_Changeto1)
32     self.ui.radioButton_4.toggled.connect(self.Stack_3_Changeto2)
33     self.ui.pushButton_2.clicked.connect(self.Addbtn_client)
34     self.ui.pushButton_7.clicked.connect(self.Updatebtn_client)
35
36     #账户查询/删除界面设置及按键绑定
37
38 self.ui.tablewidget_2.setSelectionBehavior(QAbstractItemView.SelectRows)
39     self.ui.tablewidget_2.setColumnWidth(7, 200)
40     self.ui.tablewidget_2.setColumnWidth(8, 1000)
41     self.ui.pushButton_4.clicked.connect(self.Searchbtn_account)
42     self.ui.pushButton_5.clicked.connect(self.Deletebtn_account)
43
44     #账户添加/更新界面设置及按键绑定
45     self.ui.radioButton.setChecked(True)
46     self.ui.radioButton.toggled.connect(self.Stack_2_Changeto1)
47     self.ui.radioButton_2.toggled.connect(self.Stack_2_Changeto2)
48     self.ui.pushButton_6.clicked.connect(self.Addbtn_account)
49     self.ui.pushButton_8.clicked.connect(self.Updatebtn_account)
50
51     #账户添加/删除关联客户界面设置及按键绑定
52     self.ui.pushButton_9.clicked.connect(self.Addbtn_account_client)
53
54 self.ui.pushButton_10.clicked.connect(self.Deletebtn_account_client)
55
56     #贷款查询/发放/删除界面设置及按键绑定
57
58 self.ui.tablewidget_3.setSelectionBehavior(QAbstractItemView.SelectRows)
59     self.ui.label_42.setFont(QFont("Roman times",10.5))
60     self.ui.tablewidget_3.setColumnWidth(5, 1000)
61     self.ui.tablewidget_6.setColumnWidth(0, 200)
62     self.ui.tablewidget_6.setColumnWidth(1, 120)
63     self.ui.pushButton_11.clicked.connect(self.Searchbtn_loan)
64     self.ui.pushButton_12.clicked.connect(self.Deletebtn_loan)
65
66     self.ui.pushButton_16.clicked.connect(self.Providebtn_loan)
67     self.ui.pushButton_17.clicked.connect(self.Providebtn_loan_2)
68
69     #贷款添加界面设置及按键绑定
```

```

65         self.ui.tablewidget_4.setColumnWidth(0, 400)
66         self.ui.tablewidget_4.itemChanged.connect(self.Cell_Changed)
67         self.ui.pushButton_14.clicked.connect(self.Addbtn_loan)
68         self.ui.pushButton_13.clicked.connect(self.Addbtn_owner_loan)
69
70         #业务统计界面设置及按键绑定
71         self.ui.tablewidget_5.setColumnWidth(0, 195)
72         self.ui.tablewidget_5.setColumnWidth(1, 195)
73         self.ui.tablewidget_5.setColumnWidth(2, 195)
74         self.ui.tablewidget_5.setColumnWidth(3, 195)
75         self.ui.comboBox_3.activated.connect(self.season_change)
76         self.ui.pushButton_15.clicked.connect(self.Business_statistics)

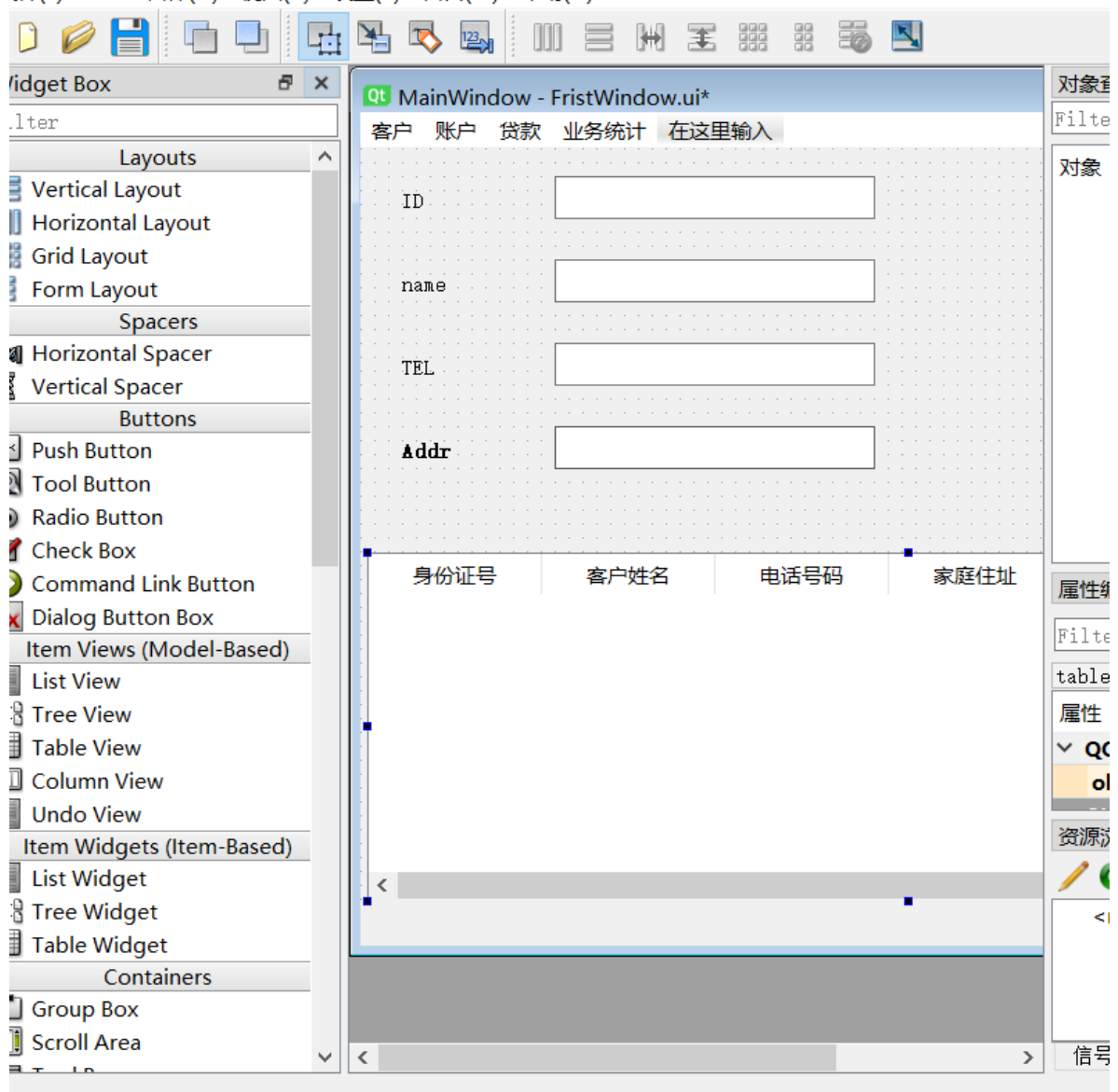
```

## 前端（实现结果）

用QT designer设计前端

Qt 设计师 - Qt Designer

文件(F) Edit 窗体(O) 视图(V) 设置(S) 窗口(W) 帮助(H)



## 实现与测试

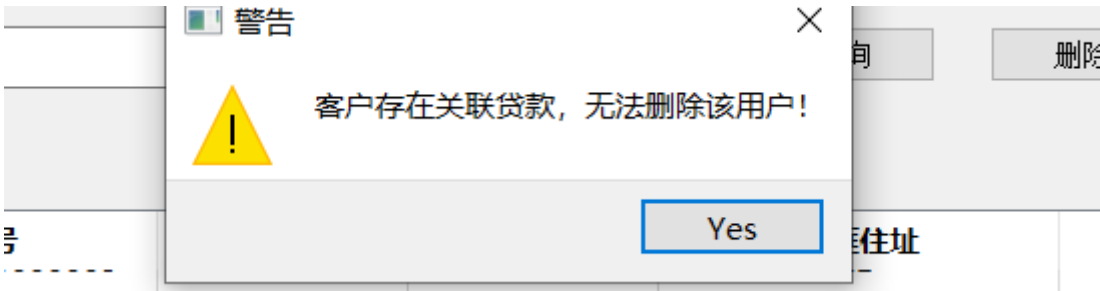
### 测试结果

受限于篇幅限制（软工复习不完了）这里仅展示部分：

添加客户misaki

身份证号	客户姓名	电话号码	家庭住址	联系人姓名
1000000000000000009	misaki	1234567	Chengdu	Funing Ya

尝试删除关联账户或者贷款的客户：



查看一个贷款的状态：

账户号： 000004	
放款日期	放款金额
1 1-23 00:00:00	1000.0
2 2-23 00:00:00	2000.0
<div>&lt; [Progress Bar] &gt;</div>	

删除发放中的贷款



警告

!

贷款正在发放中，无法删除！

Yes

发放

贷款状态  
发放中

5000.0	3	发放中
7000.0	1	未开始发放
12000.0	2	已全部发放

## 总结与讨论

本实验实现了一个前后端分离的银行管理系统，并完成了所有需求。我在进行实验的过程中熟练掌握了mysqldb的使用，收获非常大（没有软工实验打底，属实很为难）。