

# “银行业务管理系统” 系统设计与实现报告

---

范翔宇 PB18000006

---

## 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\(时间原因未选\)](#)
- [4.实现与测试](#)
- [4.1实现结果](#)
- [4.2测试结果](#)
- [5.总结与讨论](#)

## 1.概述

### 1.1系统目标

本次lab3采用B/S架构，目的是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基本的增删改查，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。

### 1.2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，统计的结果以表格形式展示。

### 1.3本报告的主要贡献

- 介绍数据库的总体设计以及工作流程
- 展示实现与测试成果

## 2.总体设计

### 2.1系统模板结构

本系统是一个前后端分离的B/S架构，主要的逻辑模块分为前端部分和后端部分。

- 前端：主要有一下模块界面：初始界面、客户管理界面、账户管理界面、贷款管理界面、业务统计界面以及对应功能具体实现增删改查等界面，并提供了一系列的按钮和输入来实现与用户交互。
- 后端：分为三个模块，db模块，check模块，main模块。其中db模块定义了链接数据库函数、封装sql语句函数、以及使用一些固定的sql语句(用来业务统计等)的函数。check模块定义了检查一些输入数据的函数。main模块主要实现路由并且调用db模块和check模块中的函数来实现输入检查并且与数据库和网页交互。

### 2.2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确认或者提交按钮后，前端向后端服务器 post 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渲染在页面上，从而完成一次与服务器的交互。

### 2.3数据库设计

基本照搬本人的lab2，但是进行了如下修改：

- 为了便于标识，对不少表和变量进行重命名
- 联系人合并到客户内
- 经理合并到部门内
- 增加了一些视图
- 增加了一些触发器来判断非法删除

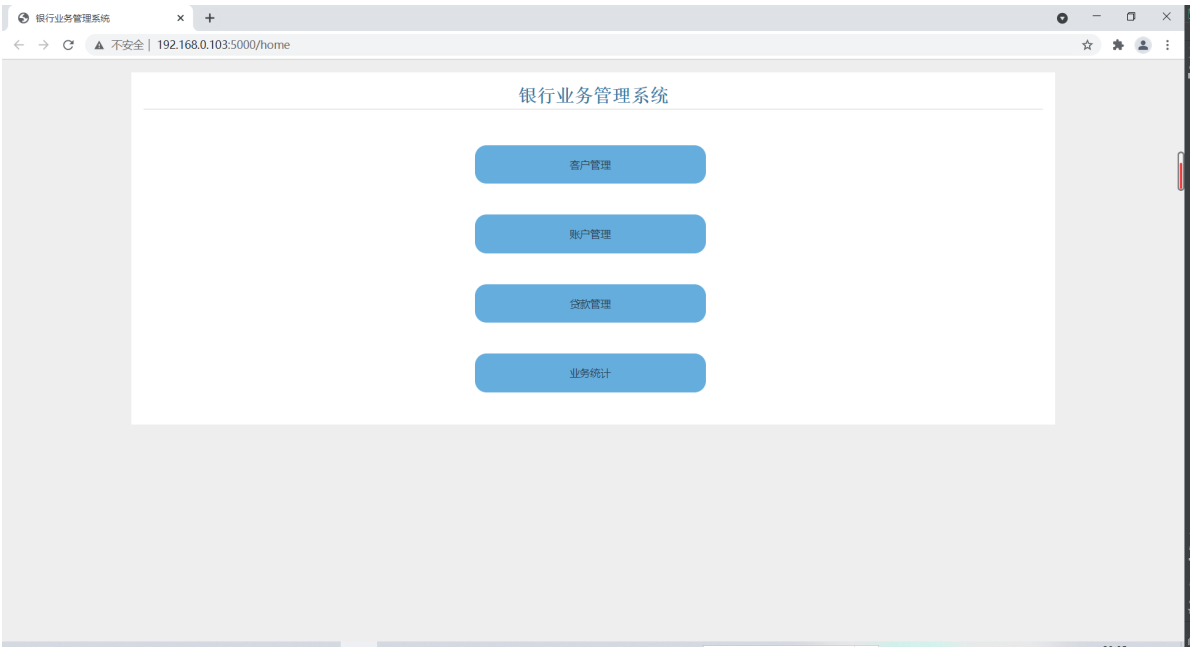
## 3.详细设计(选做)

TnT，时间来不及了，细讲不了了。

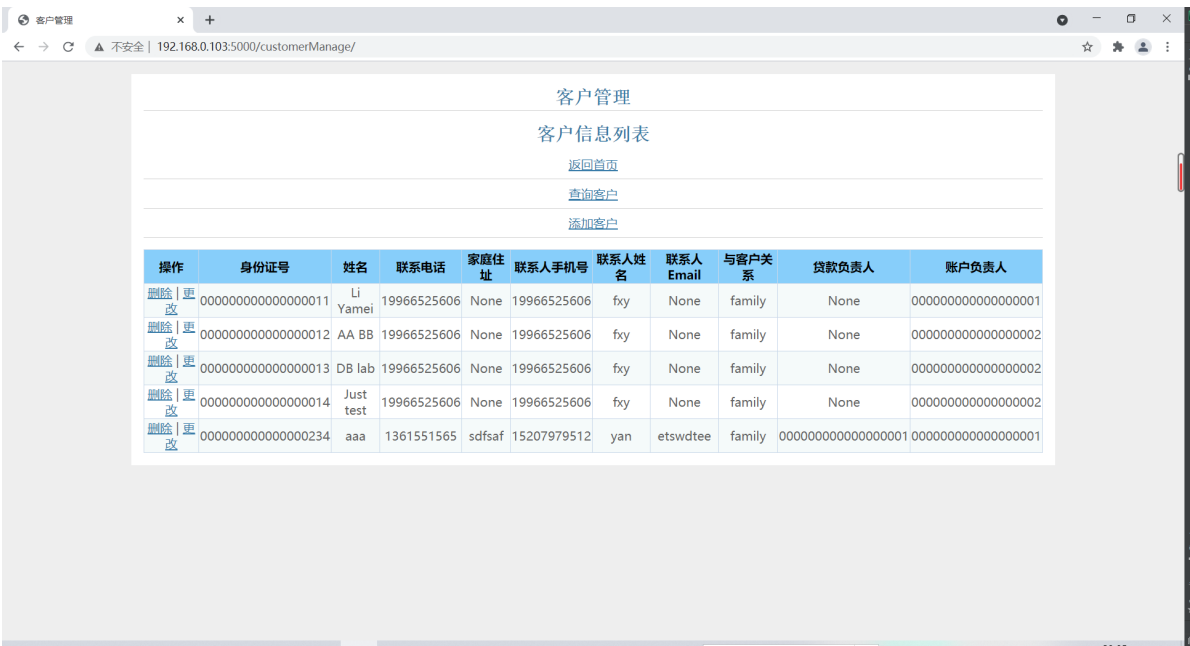
## 4.实现与测试(选做)

4.1实现结果

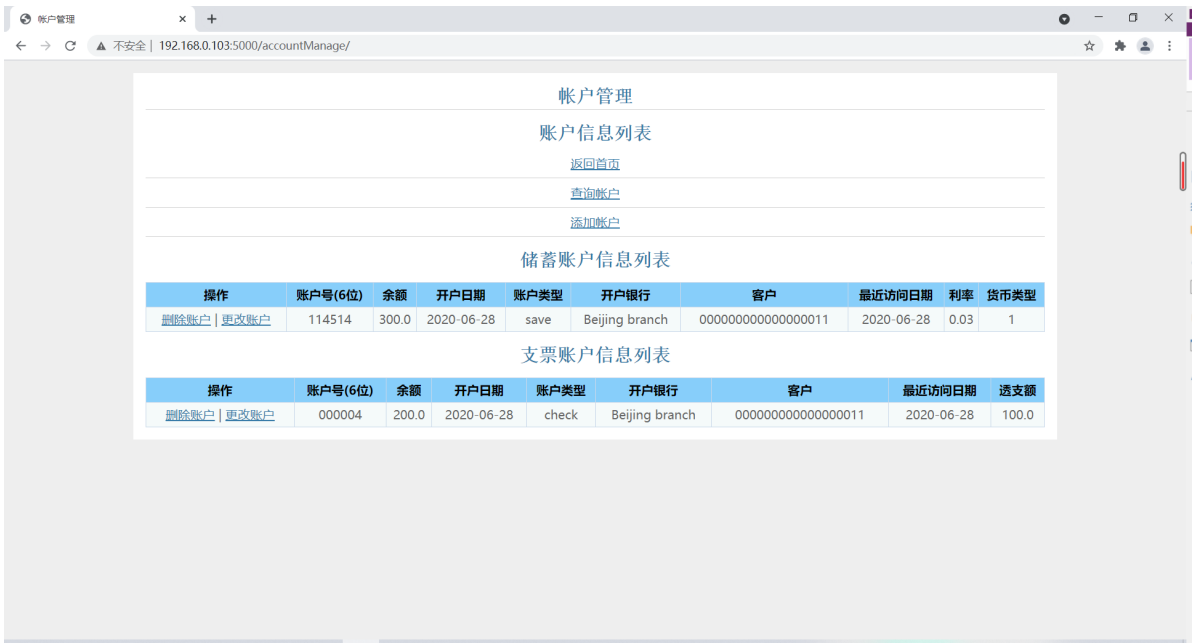
初始化界面：



客户管理界面：



账户管理界面：



贷款管理界面：



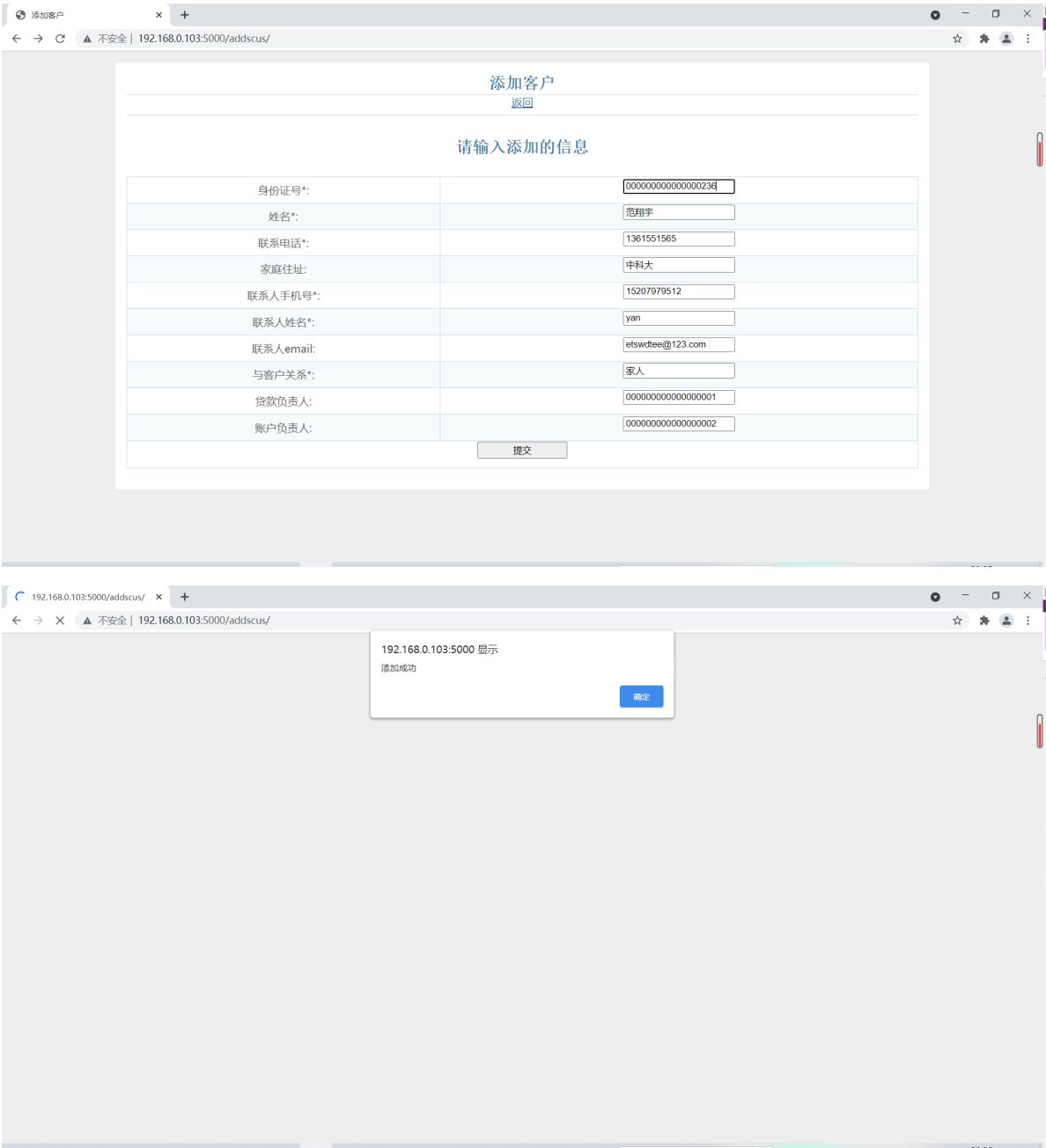
业务统计界面：

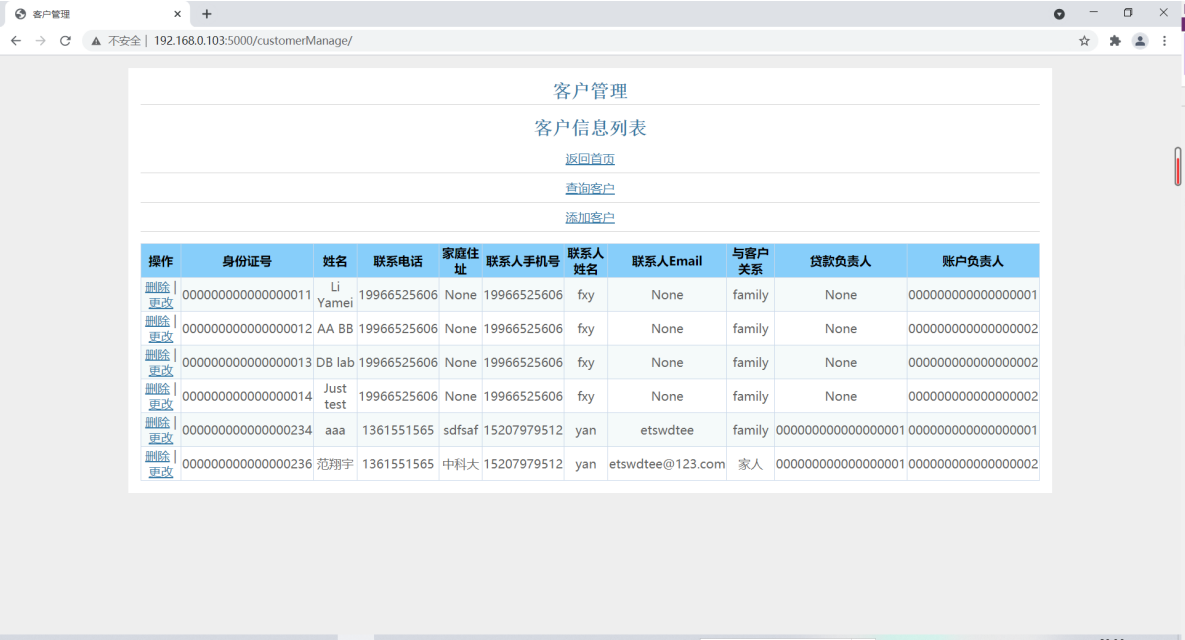


4.2测试结果

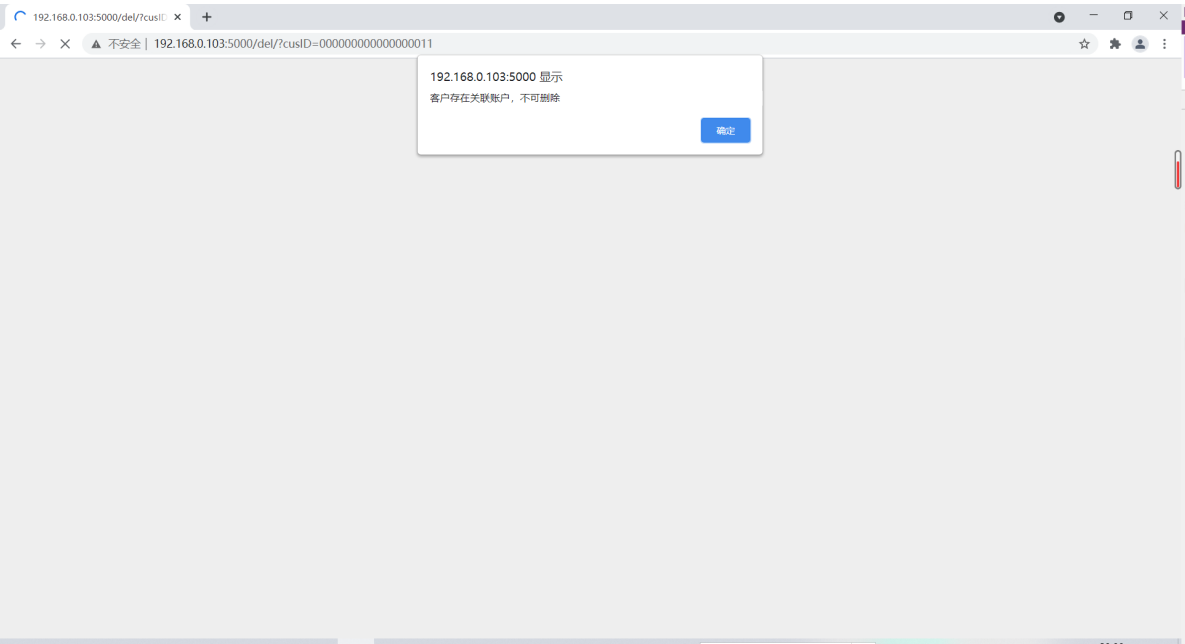
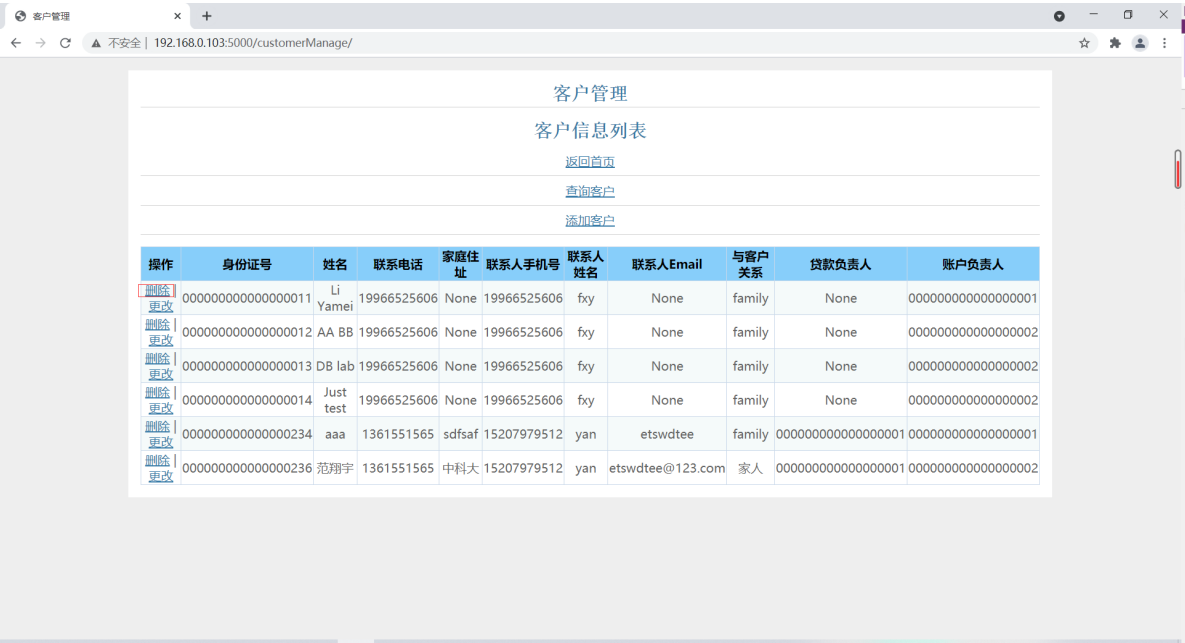
受限于篇幅限制，本文不可能对所有的输入情况进行展示，这里仅展示部分：

添加账户：





尝试删除带有账户的客户：



更改客户信息：

修改客户信息

192.168.0.103:5000/changeCustom/?custID=000000000000000236

修改客户信息

更改需要更改的信息

[返回](#)

身份证号:000000000000000236

姓名:范翔宇

联系电话:1361551565

家庭住址:中科大

联系人手机号:15207979512

联系人姓名:yan

联系人Email:etswdtee@123.com

与客户关系:家人

贷款负责人:000000000000000001

账户负责人:000000000000000002

提交

修改客户信息

192.168.0.103:5000/changeCustom/?custID=000000000000000236

修改客户信息

更改需要更改的信息

[返回](#)

身份证号:000000000000000236

姓名:范翔宇

联系电话:1361551565

家庭住址:中科大

联系人手机号:15207979512

联系人姓名:ccm

联系人Email:etswdtee@123.com

与客户关系:家人

贷款负责人:000000000000000001

账户负责人:000000000000000002

提交

192.168.0.103:5000/chas/

192.168.0.103:5000/chas/

192.168.0.103:5000 显示  
更新成功

确定



查询账户：



尝试输入不合法的贷款金额：



添加贷款

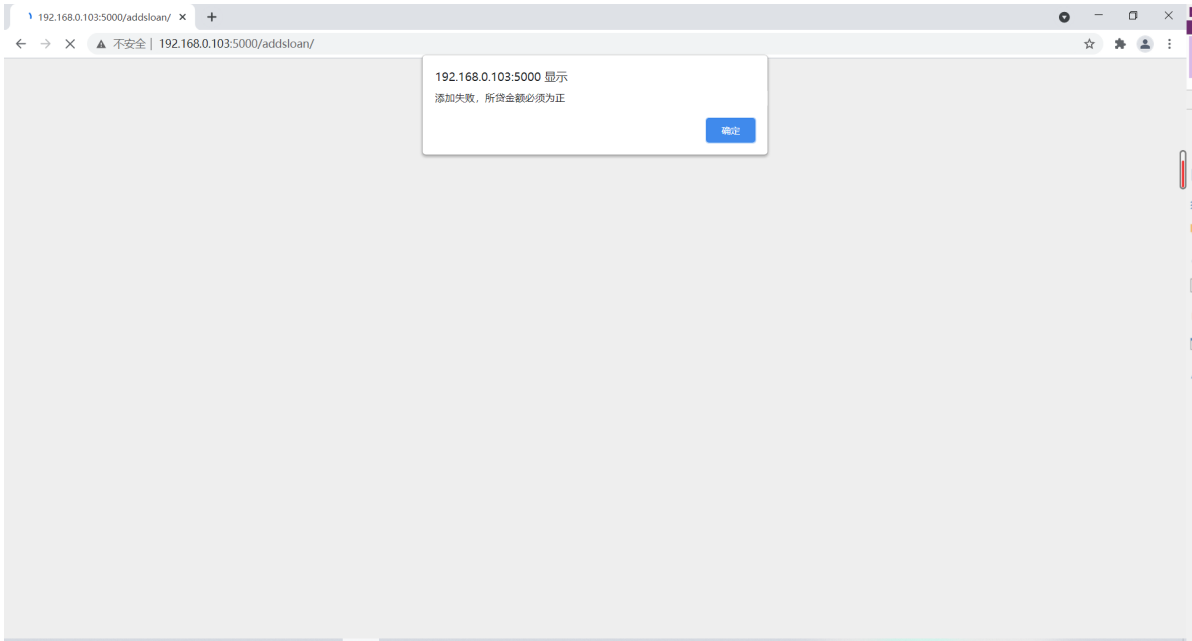
192.168.0.103:5000/addLoan/

添加贷款

[返回](#)

请输入添加的信息

贷款号*(4位):	1235
所贷金额*:	-200
发放银行*:	Beijing branch
客户*:	000000000000000011
<div>提交</div>	



查看业务统计：

业务统计

192.168.0.103:5000/statistics/

业务统计

[返回首页](#)

[曲线图](#)

请输入业务统计年份

年份:	2020
<div>提交</div>	

储蓄业务年表

分支机构	总金额	用户数
------	-----	-----

储蓄金额季度表

分支机构	季度一	季度二	季度三	季度四
------	-----	-----	-----	-----

储蓄客户季度表

分支机构	季度一	季度二	季度三	季度四
------	-----	-----	-----	-----

储蓄金额月表

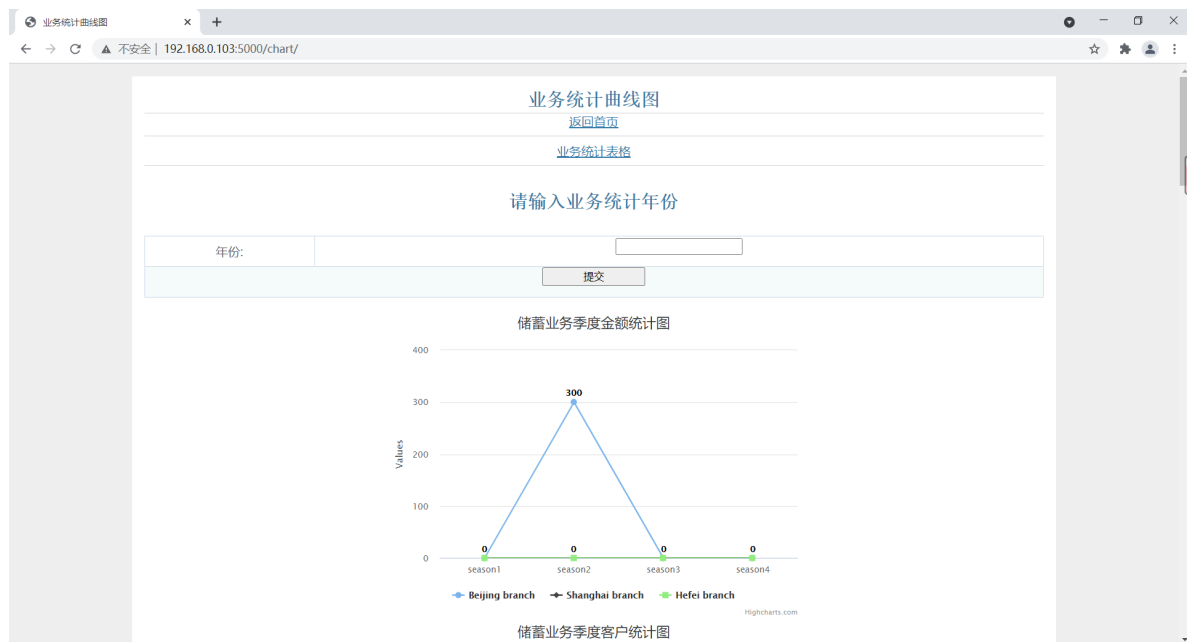
分支机构	一月	二月	三月	四月	五月	六月	七月	八月	九月	十月	十一月	十二月
------	----	----	----	----	----	----	----	----	----	----	-----	-----

储蓄客户月表

分支机构	一月	二月	三月	四月	五月	六月	七月	八月	九月	十月	十一月	十二月
------	----	----	----	----	----	----	----	----	----	----	-----	-----



对应曲线图：



更多的测试可以通过访问项目的网站自行尝试。

## 5.总结与讨论

本次实验主要先基于lab2的cdm建立好数据库进行适当修改，并且在网络上的HMTL模板进行修改来适配本次实验，然后用python-flask架构搭建路由与数据库交互，最后实现一个B/S架构的银行业务管理系统，并且完成了所有需求，同时又具备一定的检错功能。使我对python有了更深一层的了解，并且为我开发网页应用积累了宝贵的经验，收获很大。