

银行业务管理系统

系统设计与实现报告

姓名:王家伟

学号:PB17000023

少年班学院计算机科学与技术系

中国科学技术大学

2020年6月

1 概述

1.1 系统目标

1.2 需求说明

1.2.1 客户管理

1.2.2 账户管理

1.2.3 贷款管理

1.2.4 业务统计

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

2.3.1 ER图设计

2.3.2 逻辑数据库结构

2.3.3 物理数据库结构

3 详细设计

3.1 客户管理模块

3.1.1 添加客户

3.1.2 删除客户

3.1.3 修改客户信息

3.1.4 查询客户信息

3.2 账户管理模块

3.2.1 新增账户

3.2.2 删除账户

3.2.3 修改账户信息

3.2.4 查询账户信息

3.3 贷款管理模块

3.3.1 新增贷款

3.3.2 删除贷款

3.3.3 发放贷款

3.3.4 查询贷款信息

3.4 业务统计模块

3.4.1 储蓄业务统计

3.4.2 贷款业务统计

3.5 其他

4 实现与测试

4.1 实现结果

4.1.1 客户管理

4.1.2 账户管理

4.1.3 贷款管理

4.1.4 业务统计

4.2 测试结果

4.2.1 客户管理

4.2.2 账户管理

4.2.3 贷款管理

4.2.4 业务统计

5 总结与讨论

1 概述

1.1 系统目标

开发一个银行业务管理系统，使得用户可以在此系统上查询、修改以及增删银行的客户信息、账户信息、贷款信息，并能够展示各个支行的业务统计信息，包括储蓄业务和贷款业务；此外本系统还需要在后台维护相应的数据，并做好错误处理。

1.2 需求说明

1.2.1 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。

1.2.2 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。

1.2.3 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。

1.2.4 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。

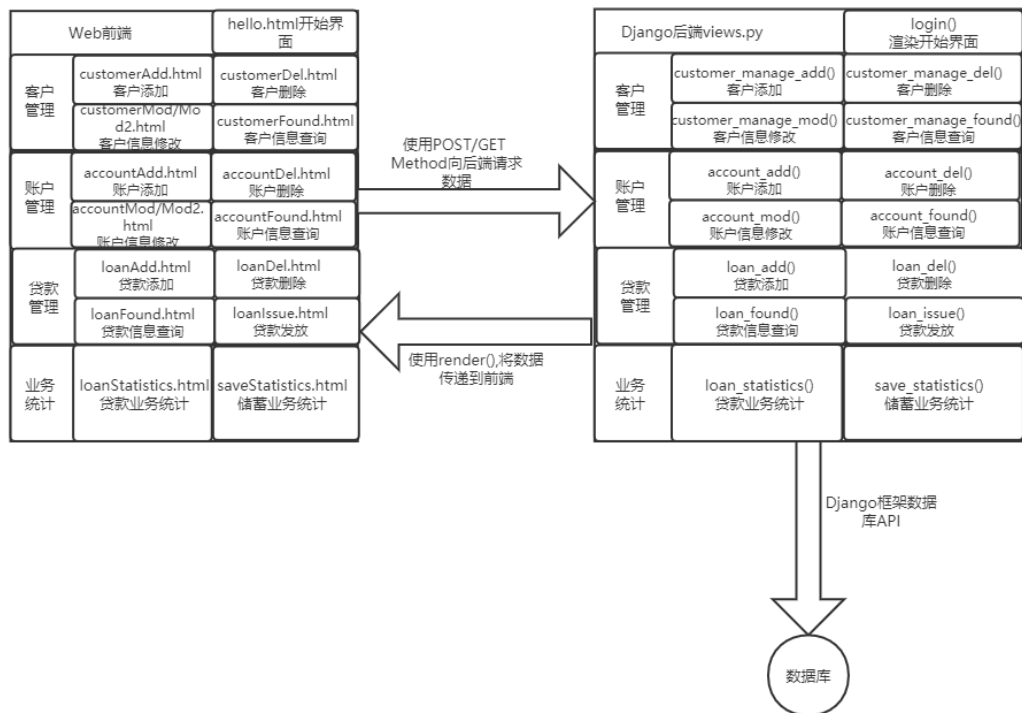
1.3 本报告的主要贡献

本报告首先简要说明系统模块结构和工作流程，并给出数据库的设计方案，然后详细说明各个模块的功能设计和执行流程，并且使用部分测试数据进行各个功能需求的测试，演示整个系统的使用方式，最后对系统开发过程进行总结。

2 总体设计

2.1 系统模块结构

本系统使用python+Django+bootstrap方案，构建一个B/S架构的银行业务管理系统。



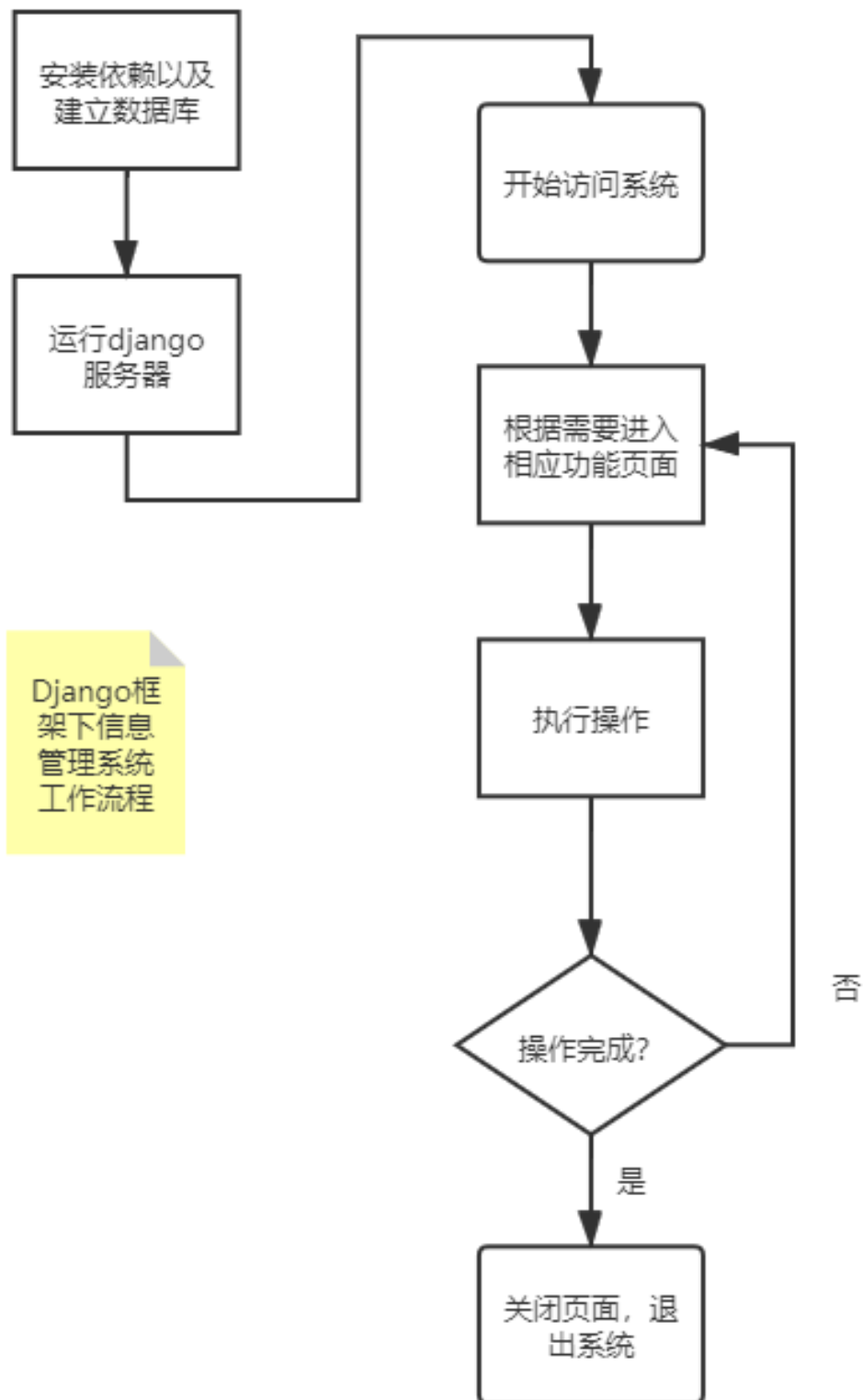
2.2 系统工作流程

安装必要依赖：python3、Django3.0、Mysql，其余已打包上传

在Mysql中运行BankManage.sql，建立相关表项和视图

运行Django：python manage.py runserver localhost:8000

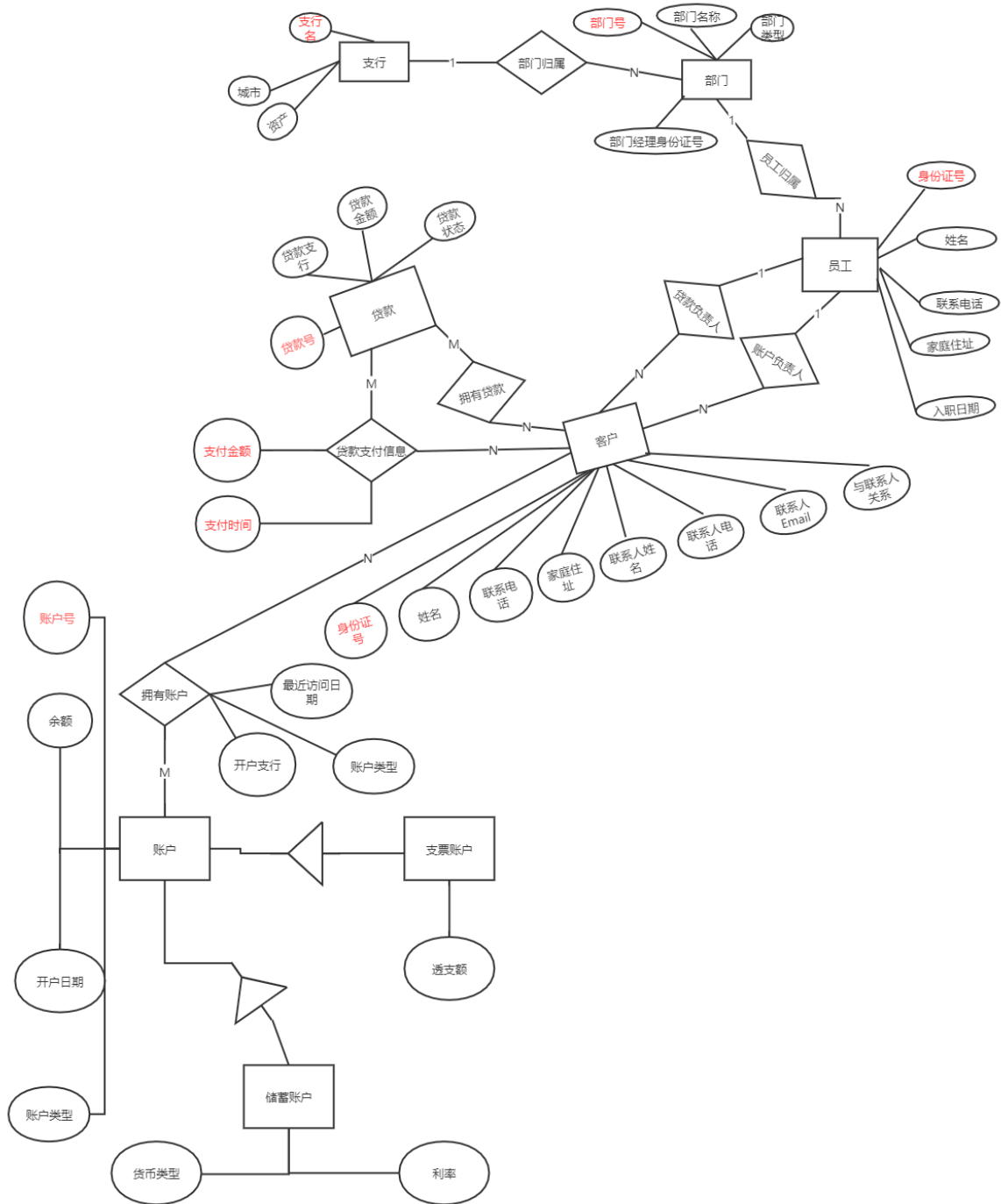
至此可以在浏览器输入localhost:8000访问系统



2.3 数据库设计

2.3.1 ER图设计

[注：由于下划线显示不清晰，所以使用红色标注主键]



2.3.2 逻辑数据库结构

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
departID	CHAR(4)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
departname	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
departtype	VARCHAR(15)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
manager	CHAR(18)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
bank	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

5. 账户

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
accountID	CHAR(6)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
money	DOUBLE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
settime	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
accounttype	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

6. 储蓄账户

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
accountID	CHAR(6)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
interestrate	FLOAT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
savetype	CHAR(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

7. 支票账户

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
accountID	CHAR(6)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
overdraft	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

8. 贷款

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
loanID	CHAR(4)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
money	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
bank	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
state	CHAR(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'

9. 贷款支付信息

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
loanID	CHAR(4)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
cusID	CHAR(18)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
money	DOUBLE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
paytime	DATETIME	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

10. 客户拥有账户

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
accountID	CHAR(6)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
bank	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
cusID	CHAR(18)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
visit	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
accounttype	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

11. 客户拥有贷款

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
loanID	CHAR(4)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
cusID	CHAR(18)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

说明：客户拥有账户这个关系里添加了一个冗余属性accounttype,因为在需求分析中我们知道一个客户在一个支行对每种账户类型只能开通一次，所以在此关系里可以加上unique的限制使得(cusID,bank,accounttype)唯一。除此之外也添加了一些视图和触发器方便操作数据库。

```

1  create view checkaccounts
2  as select accounts.accountID,bank,accounttype,money,settime,overdraft
3  from accounts,checkacc,(select distinct accounts.accountID,bank from
4  accounts,cusforacc where accounts.accountID=cusforacc.accountID) tmp
5  where accounts.accountID=checkacc.accountID and
6  accounts.accountID=tmp.accountID;
7
8  create view saveaccounts
9  (accountID,bank,accounttype,money,settime,interestrate,savetype)
10 as select
11 accounts.accountID,bank,accounttype,money,settime,interestrate,savetype
12 from accounts,saveacc,(select distinct accounts.accountID,bank from
13 accounts,cusforacc where accounts.accountID=cusforacc.accountID) tmp
14 where accounts.accountID=saveacc.accountID and
15 accounts.accountID=tmp.accountID;
16
17 create view checkstat (bank, totalmoney,totalcustomer)
18 as select t1.bank,totalmoney,totalcustomer
19 from (select bank,sum(money) as totalmoney from checkaccounts group by bank)
20 t1,
21 (select bank,count(distinct cusID) as totalcustomer from cusforacc where
22 accounttype='CheckingAccount' group by bank) t2
23 where t1.bank=t2.bank;
24
25 create view savestat (bank, totalmoney,totalcustomer)
26 as select t1.bank,totalmoney,totalcustomer
27 from (select bank,sum(money) as totalmoney from saveaccounts group by bank)
28 t1,
29 (select bank,count(distinct cusID) as totalcustomer from cusforacc where
30 accounttype='SavingAccount' group by bank) t2
31 where t1.bank=t2.bank;
32
33 create view loanstat (bank, totalmoney,totalcustomer)
34 as select t1.bank,totalmoney,totalcustomer
35 from (select bank,sum(money) as totalmoney from loan group by bank) t1,
36 (select bank, count(distinct cusID) as totalcustomer
37 from loan,cusforloan where loan.loanID=cusforloan.loanID group by bank) t2
38 where t1.bank=t2.bank;
39
40 DELIMITER $$
41 create trigger loanDelete
42 Before delete on loan
43 for each row
44 begin
45     declare a int;
46     select state into a from loan where old.loanID=loan.loanID;
47     if a = 1 then
48         signal sqlstate 'HY000' set message_text = 'In the loan origination';
49     end if;
50 end $$
51 DELIMITER ;
52
53 DELIMITER $$
54 create trigger loanState
55 After insert on payinfo
56 for each row
57 begin
58     declare pay int;
59     declare total int;
60     select sum(money) into pay from payinfo where payinfo.loanID=new.loanID;
61     select money into total from loan where loan.loanID=new.loanID;
62     if pay > 0 and pay < total then
63         update loan set state='1' where loan.loanID=new.loanID;
64     elseif pay=total then

```



```

27         update loan set state='2' where loan.loanID=new.loanID;
28     elseif pay>total then
29         delete from payinfo where payinfo.loanID=new.loanID and
payinfo.cusID=new.cusID and payinfo.paytime=new.paytime and
payinfo.money=new.money;
30         signal sqlstate 'HY001' set message_text = 'Exceed the loan amount';
31     end if;
32 end $$
33 DELIMITER ;

```

3 详细设计

说明：本系统很多错误检测以及处理放在了前端，比如限制客户身份证号为18位、金额必须为正数、必须选择支行等等要求，已经通过对前端的设计进行实现，这样做大大简化了后端的检查，专注于功能的实现。如果有发现Bug，请联系我! wangjiawei@mail.ustc.edu.cn

3.1 客户管理模块

3.1.1 添加客户

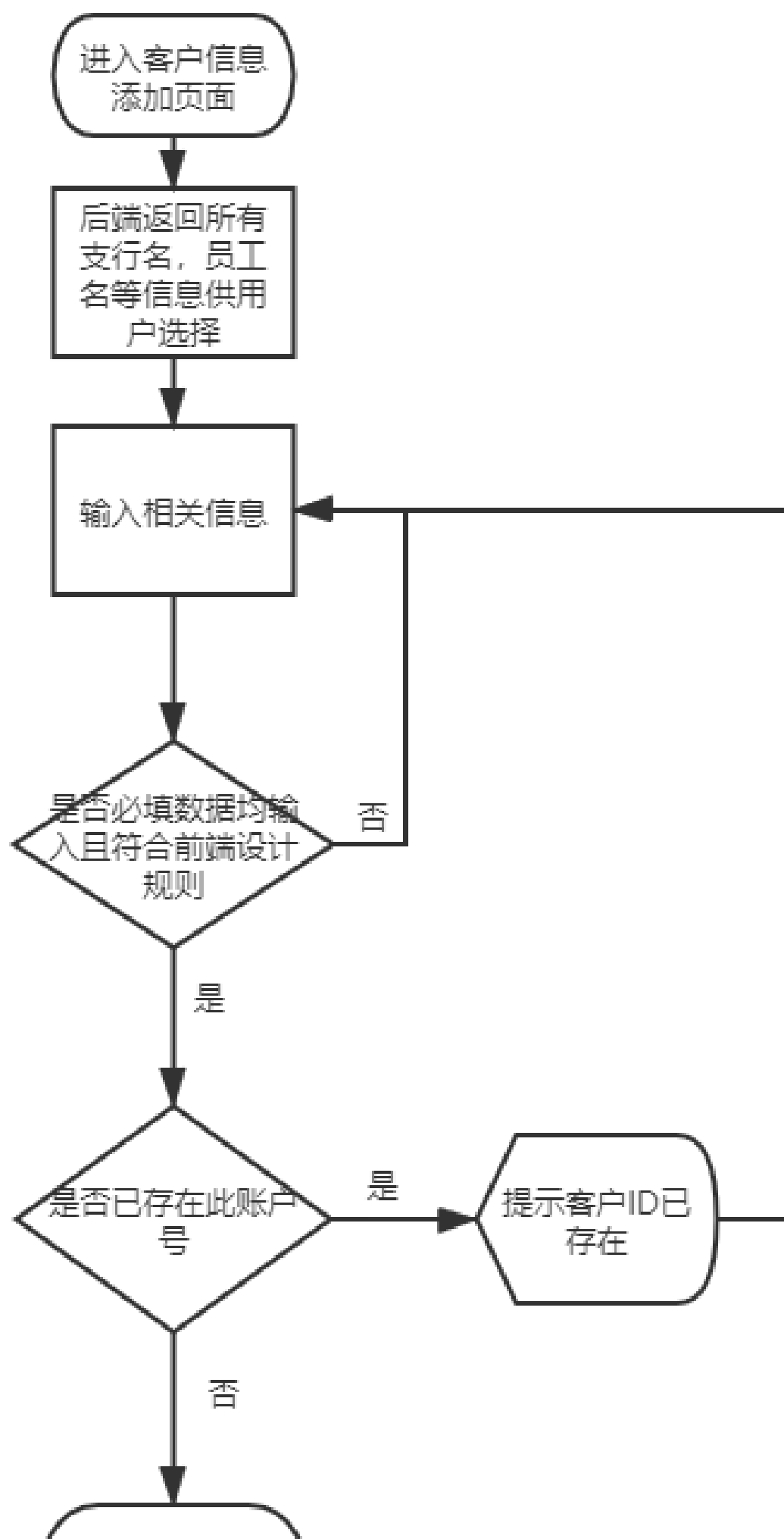
输入：

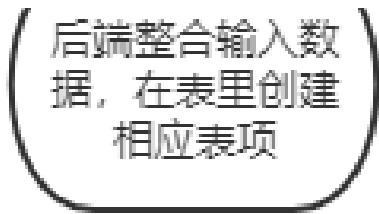
- 客户ID（必填）
- 客户姓名（必填）
- 客户电话（必填）
- 客户地址
- 联系人姓名（必填）
- 联系人电话（必填）
- 联系人Email
- 与联系人关系（必填）
- 贷款负责人（必填）
- 账户负责人（必填）

输出：

- 返回状态信息，提示是否创建成功，或者是否需要修改
- 主要通过前端输入框的限制来限制输入规则以及通过后端检查主键是否冲突来提醒是否创建成功

流程图：





3.1.2 删除客户

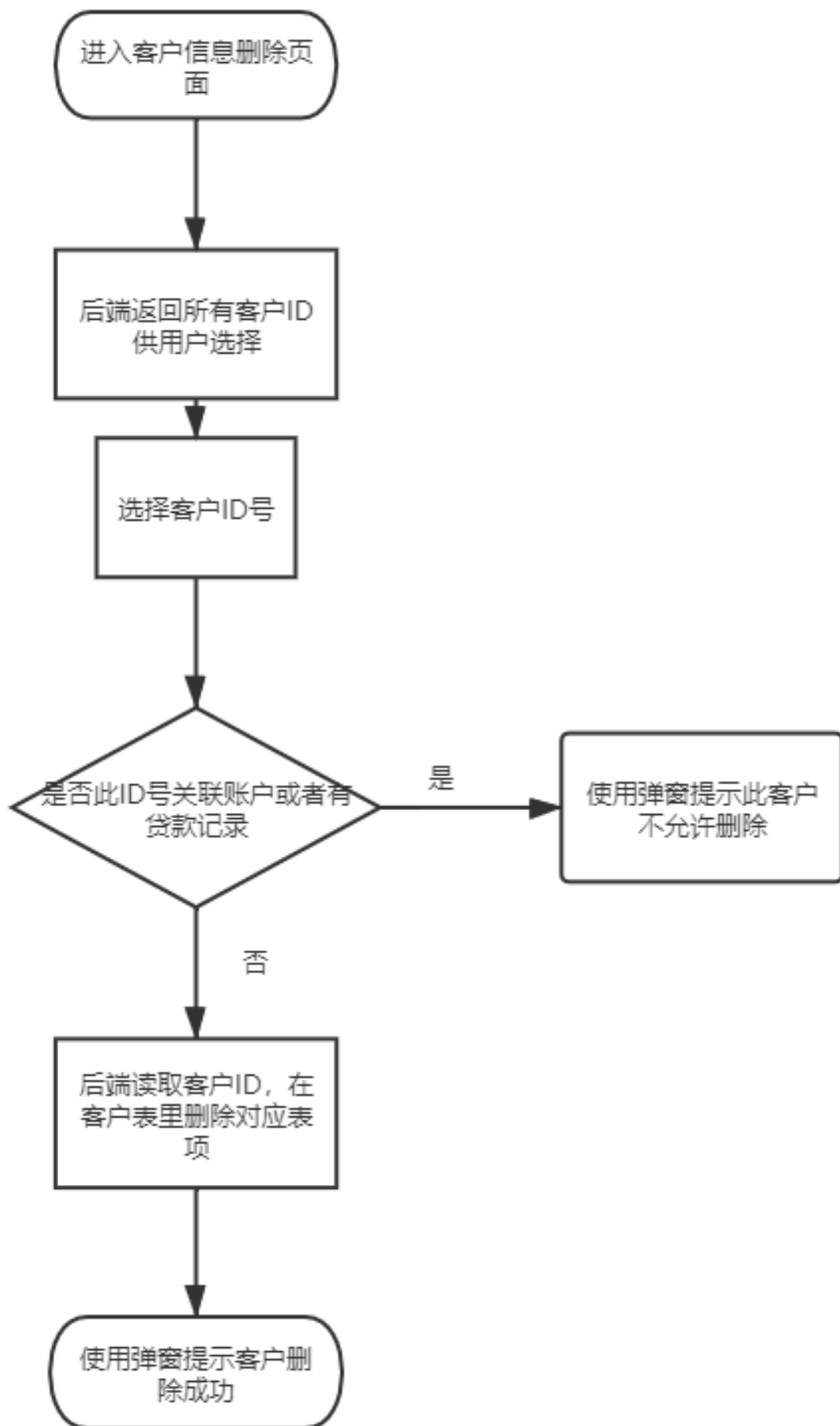
输入：

- 客户ID

输出：

- 返回状态信息，提示是否删除成功，主要通过后端检查此客户是否在贷款或者有关联账户

流程图：



3.1.3 修改客户信息

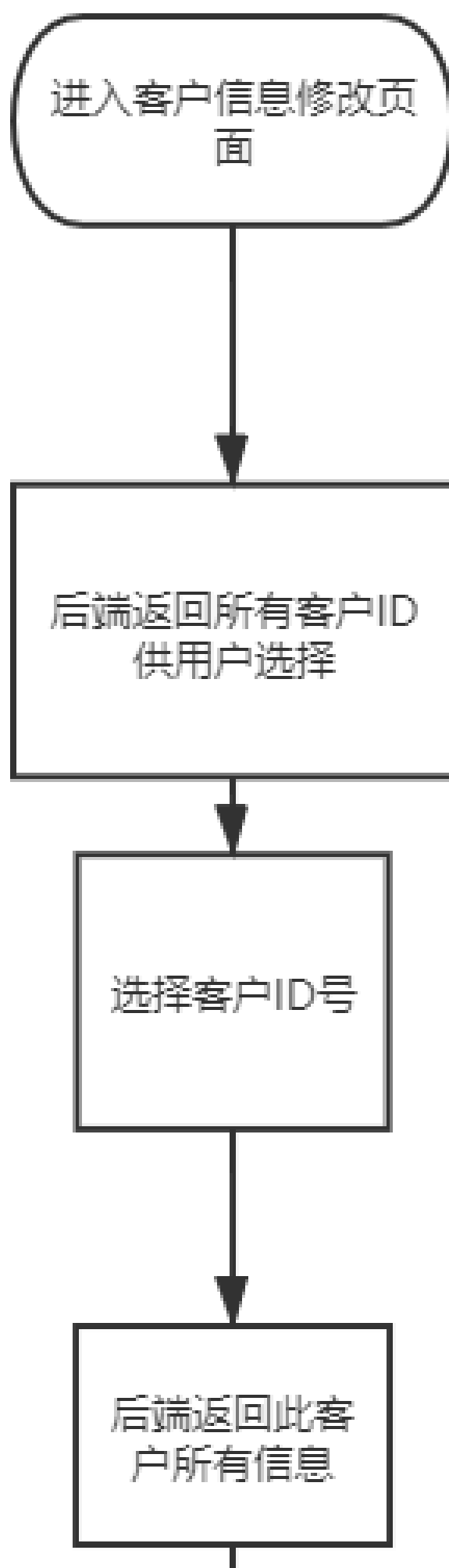
输入：

- 客户ID
- 在返回的客户信息基础上修改可修改的信息

输出：

- 第一次输入客户ID时返回客户所有信息
- 用户修改之后返回是否修改成功

流程图：





3.1.4 查询客户信息

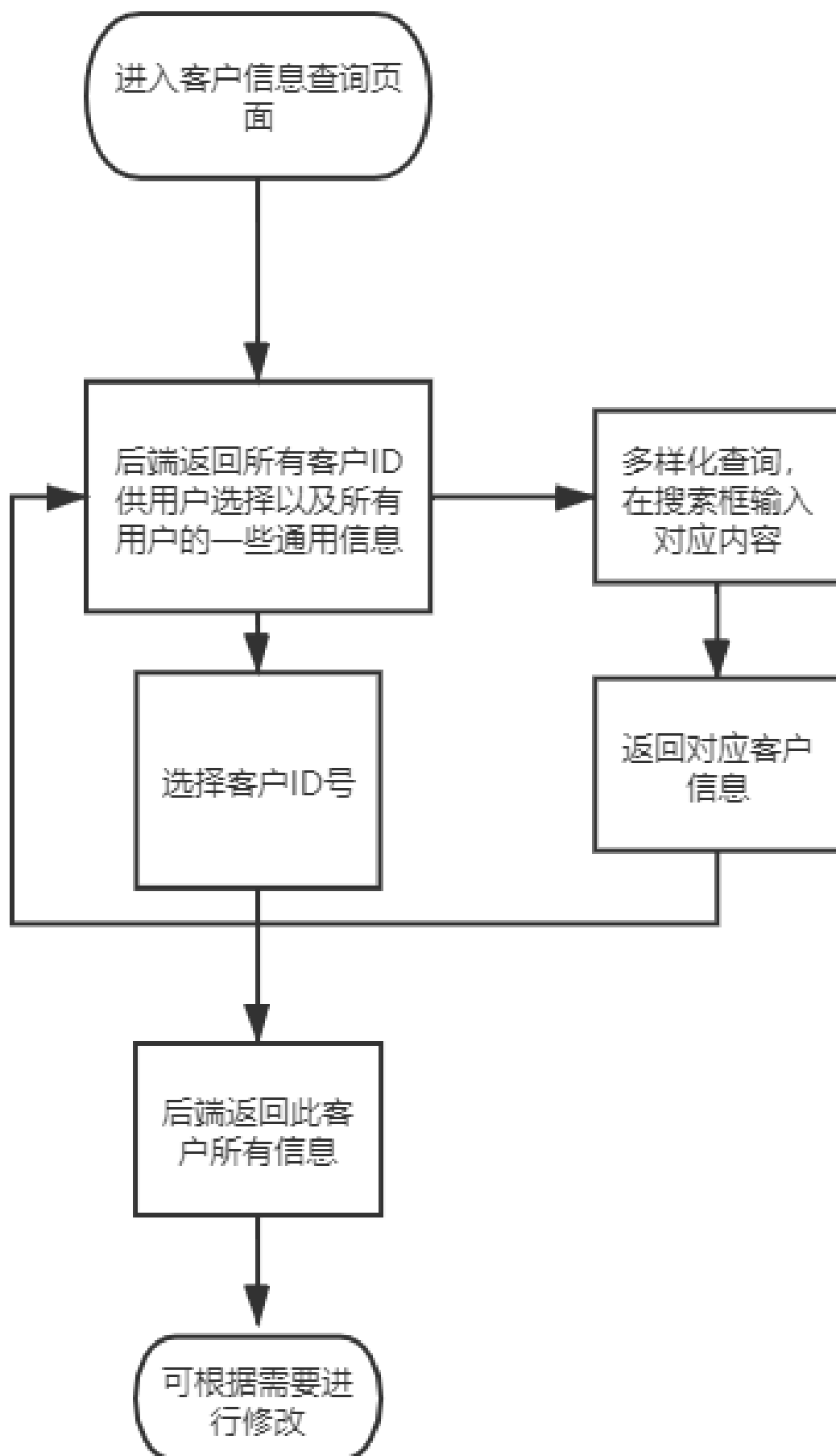
输入：

- 详细信息查询：客户ID
- 多样化查询：在搜索框里输入想要搜索的内容，比如员工A负责的所有客户，直接输入员工名即可

输出：

- 所有账户的ID，姓名，贷款负责人，账户负责人，联系人姓名
- 对应客户ID的客户详细信息

流程图：



3.2 账户管理模块

3.2.1 新增账户

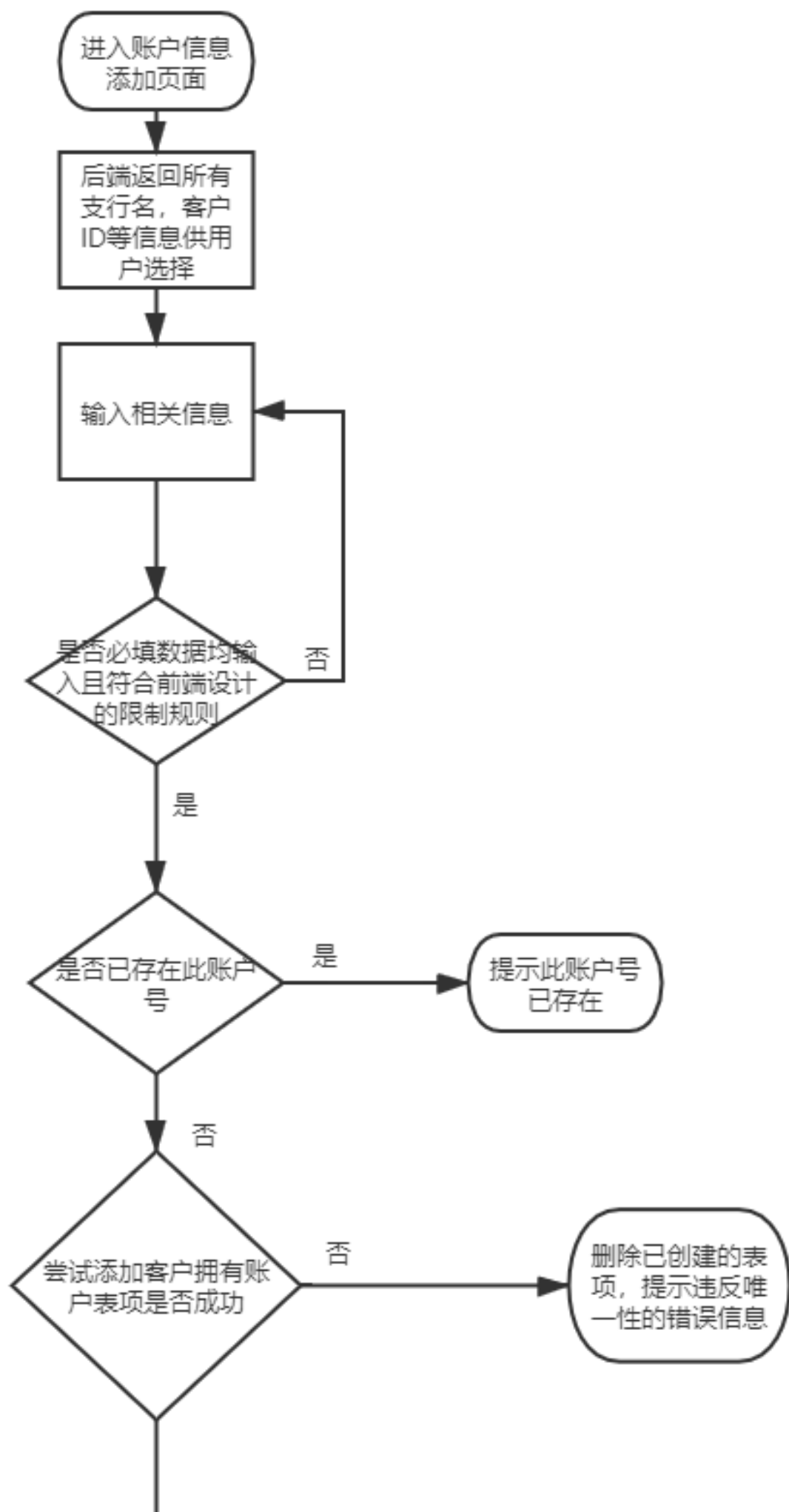
输入：

- 账户ID（必填）
- 客户ID（必填）
- 支行名（必填）
- 开户日期（必填）
- 余额（必填）
- 账户类型（必填）
- 利率/货币类型/透支额

输出：

- 返回状态和信息，提示是否创建账户成功

流程图：





3.2.2 删除账户

输入：

- 账户ID

输出：

- 返回状态信息，提示是否删除成功

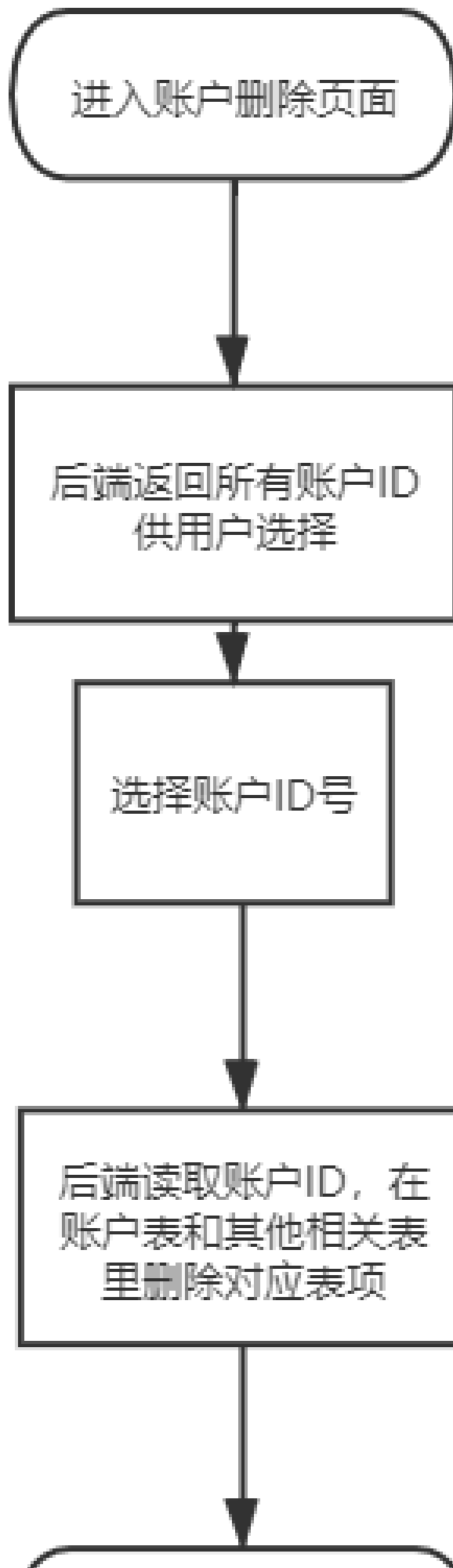
流程图：

进入账户删除页面

后端返回所有账户ID
供用户选择

选择账户ID号

后端读取账户ID，在
账户表和其他相关表
里删除对应表项



使用弹窗提示账户删除成功

3.2.3 修改账户信息

输入：

- 账户ID
- 在返回的账户信息基础上修改可修改的信息

输出：

- 第一次输入账户ID时返回账户所有信息
- 用户修改之后返回是否修改成功的状态信息

流程图：

进入账户信息
修改页面



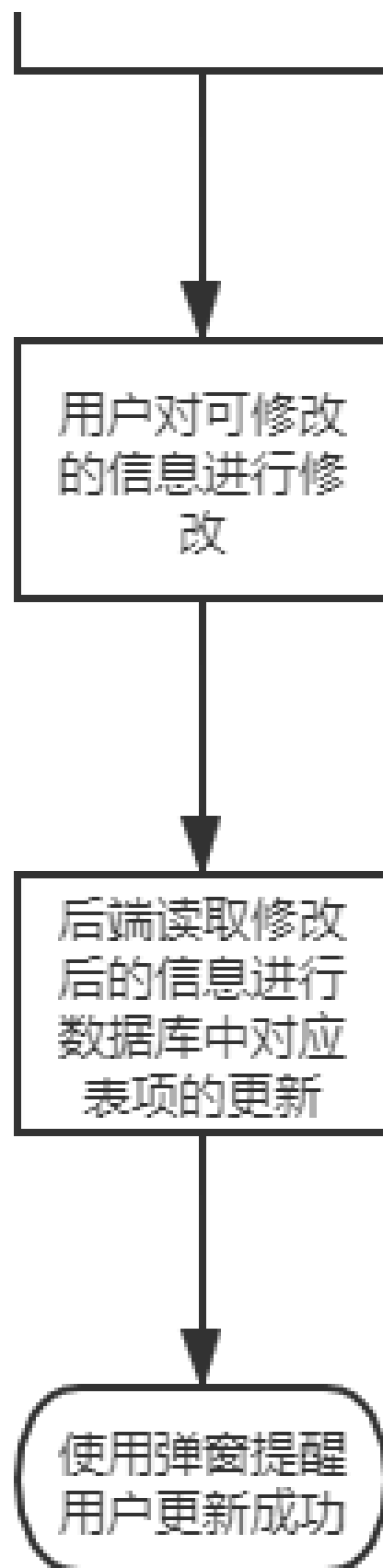
后端返回所有
账户号供用户
选择



选择账户ID



后端返回此账
户所有信息



3.2.4 查询账户信息

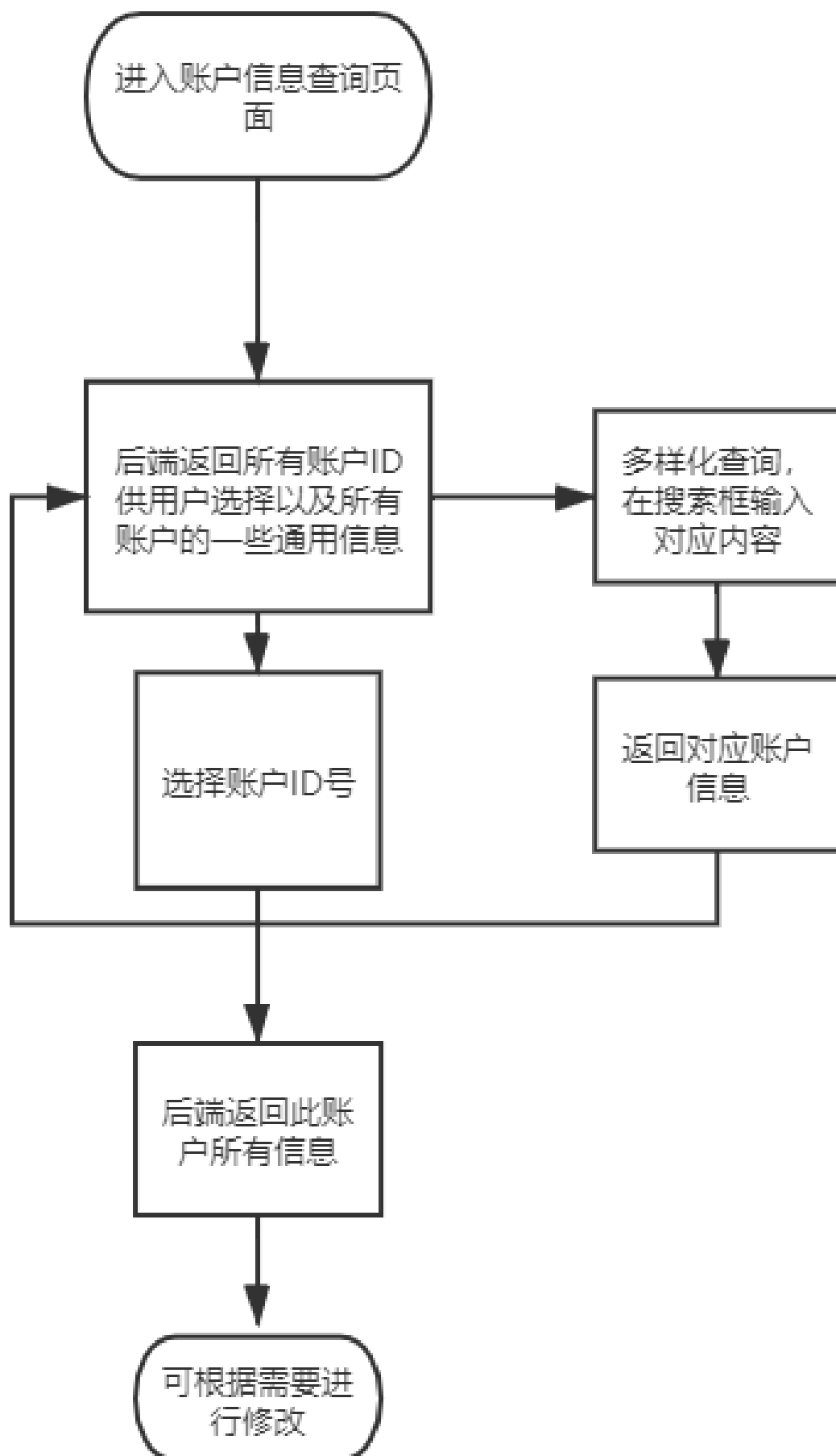
输入：

- 详细信息查询：账户ID
- 多样化查询：在搜索框里输入想要搜索的内容，比如支行A的所有账户，直接输入支行名A即可

输出：

- 所有账户的ID，支行，开户时间，余额，账户类型
- 对应账户ID的账户详细信息

流程图：



3.3 贷款管理模块

3.3.1 新增贷款

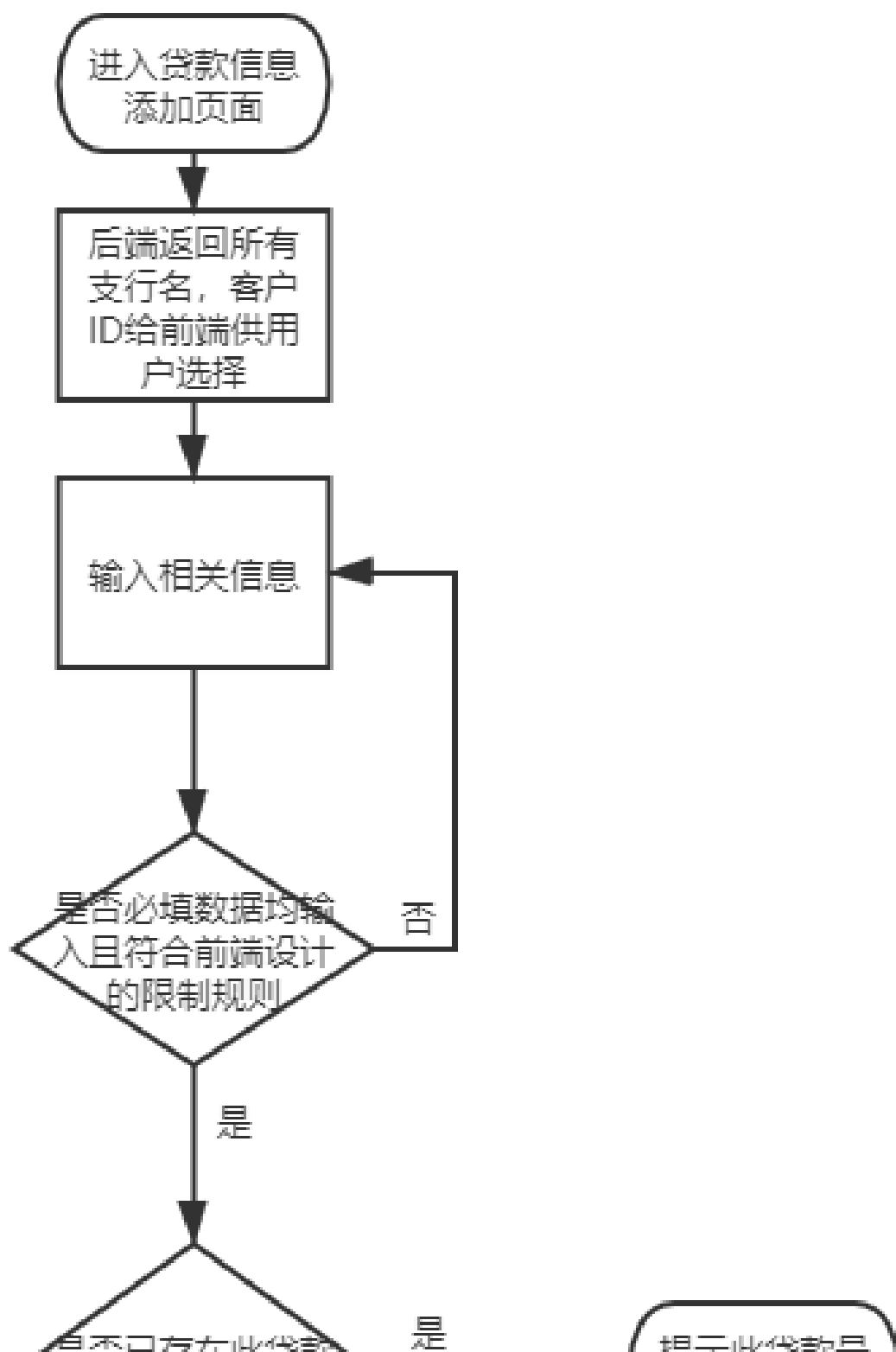
输入：

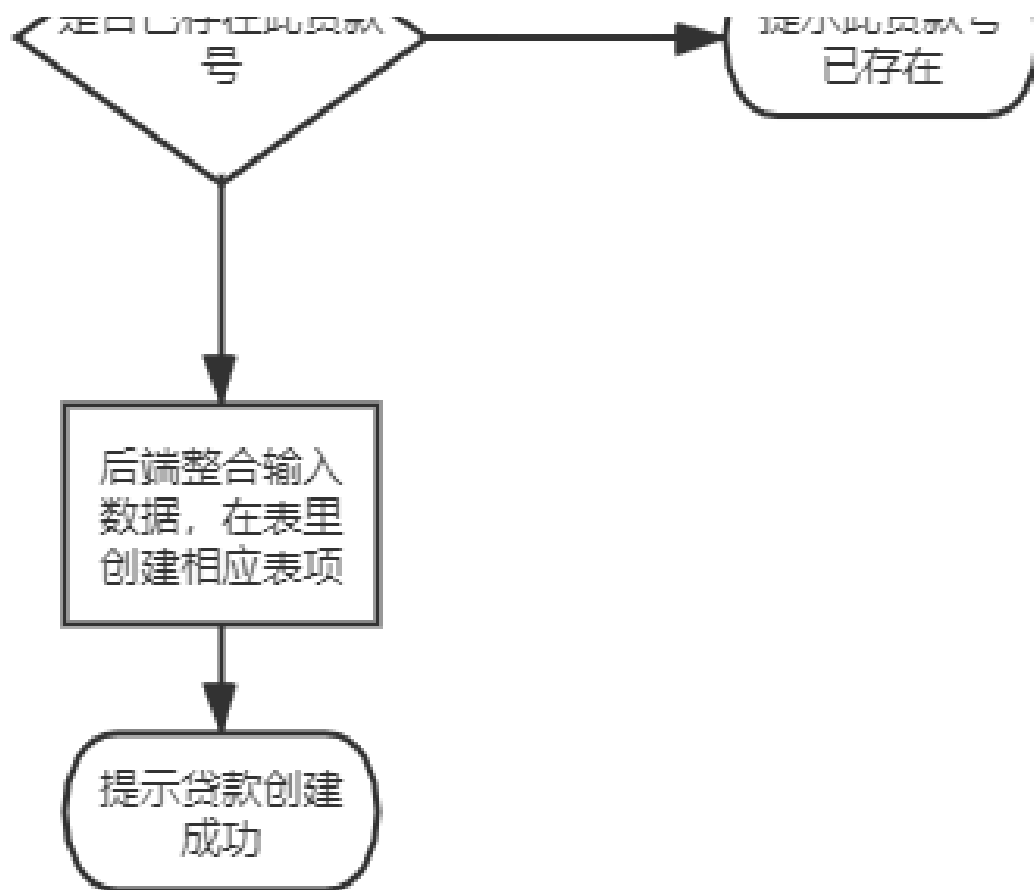
- 贷款ID（必填）
- 客户ID（必填）
- 支行（必填）
- 贷款金额（必填）

输出：

- 返回状态信息，提示是否创建成功

流程图：





3.3.2 删除贷款

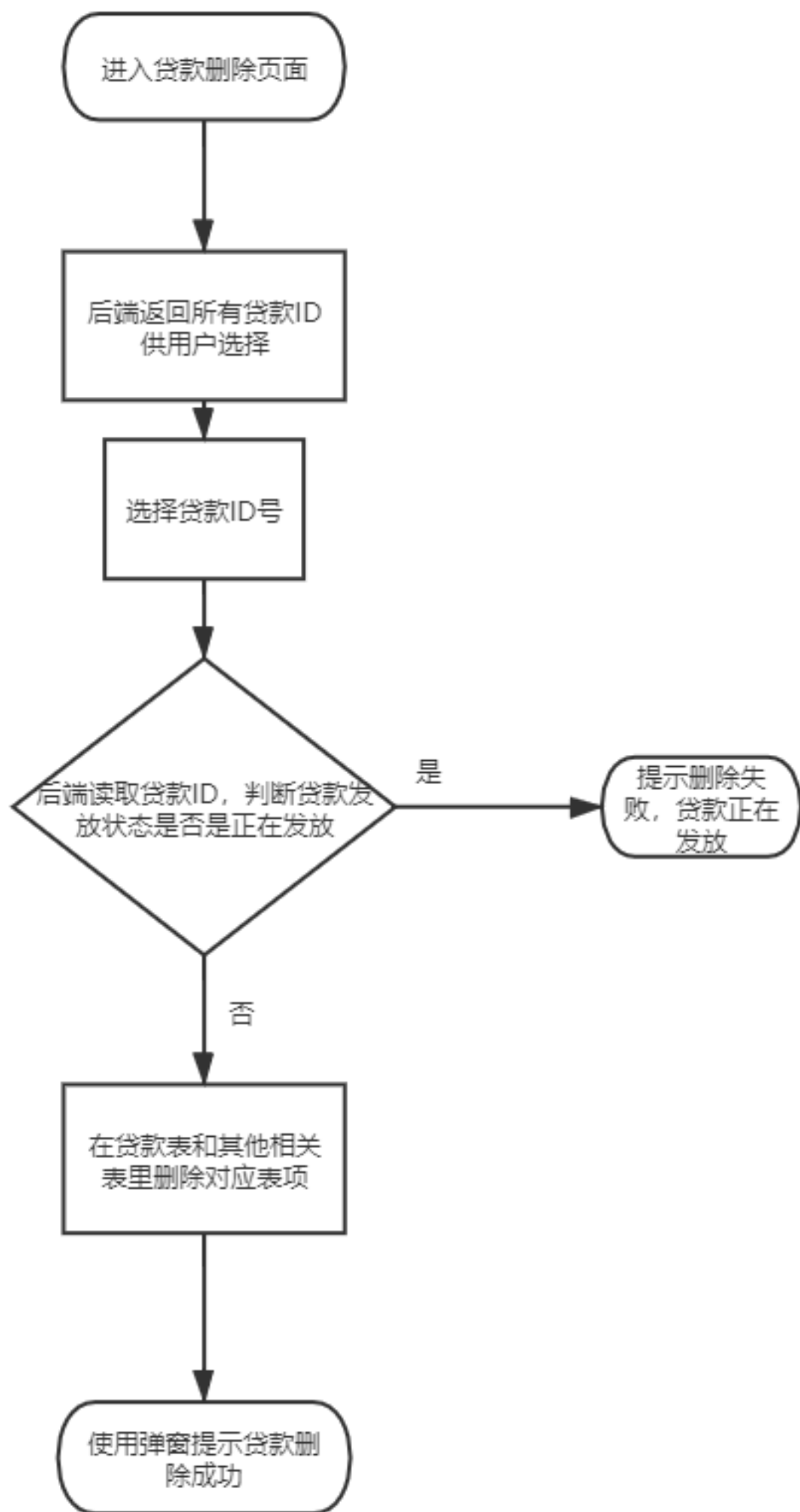
输入：

- 贷款ID

输出：

- 返回状态信息，提示是否删除成功

流程图：



3.3.3 发放贷款

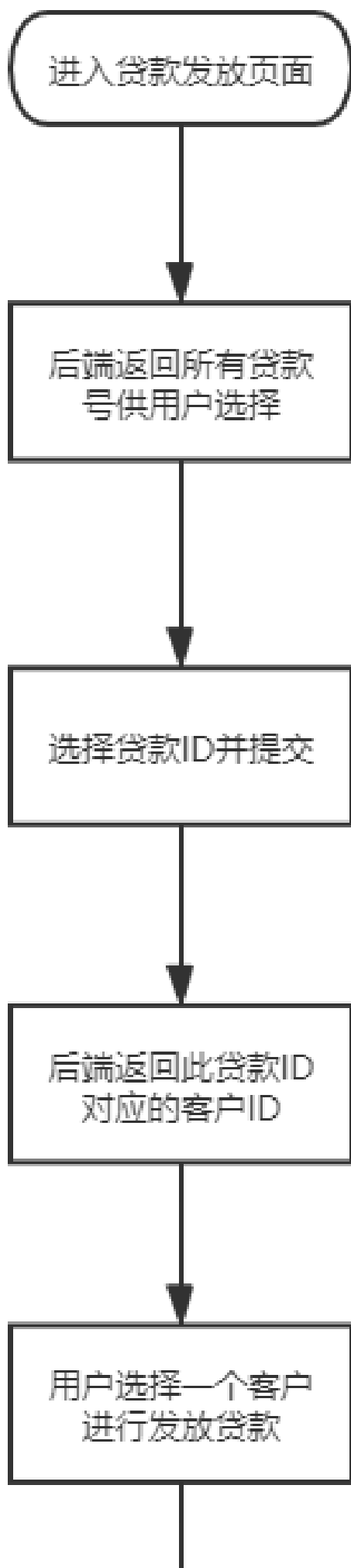
输入：

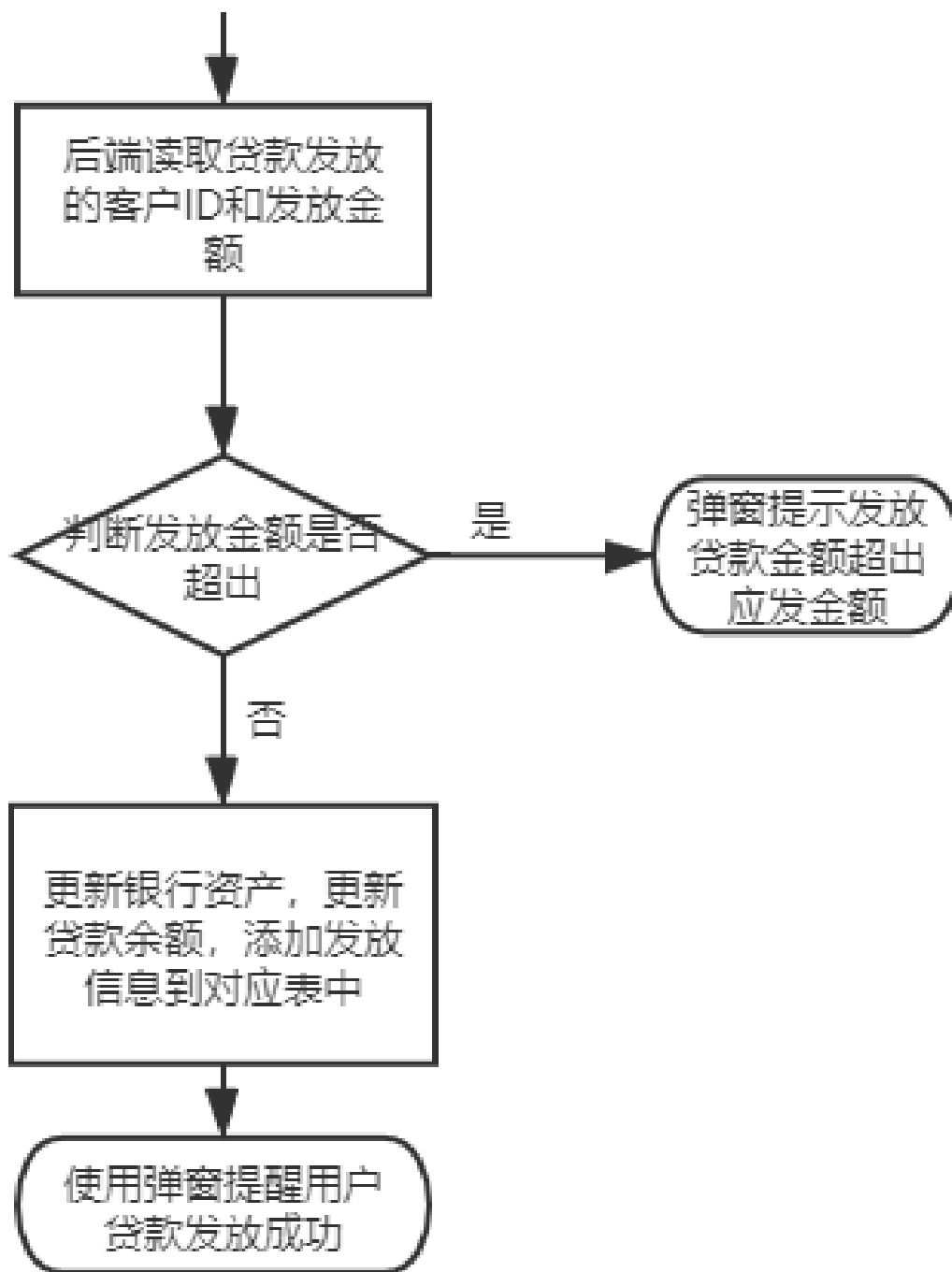
- 输入贷款ID
- 选择为绑定的某个客户发放贷款
- 发放金额

输出：

- 在第一次提交贷款ID后输出对应绑定的所有客户ID
- 返回状态信息，提示是否发放成功

流程图：





3.3.4 查询贷款信息

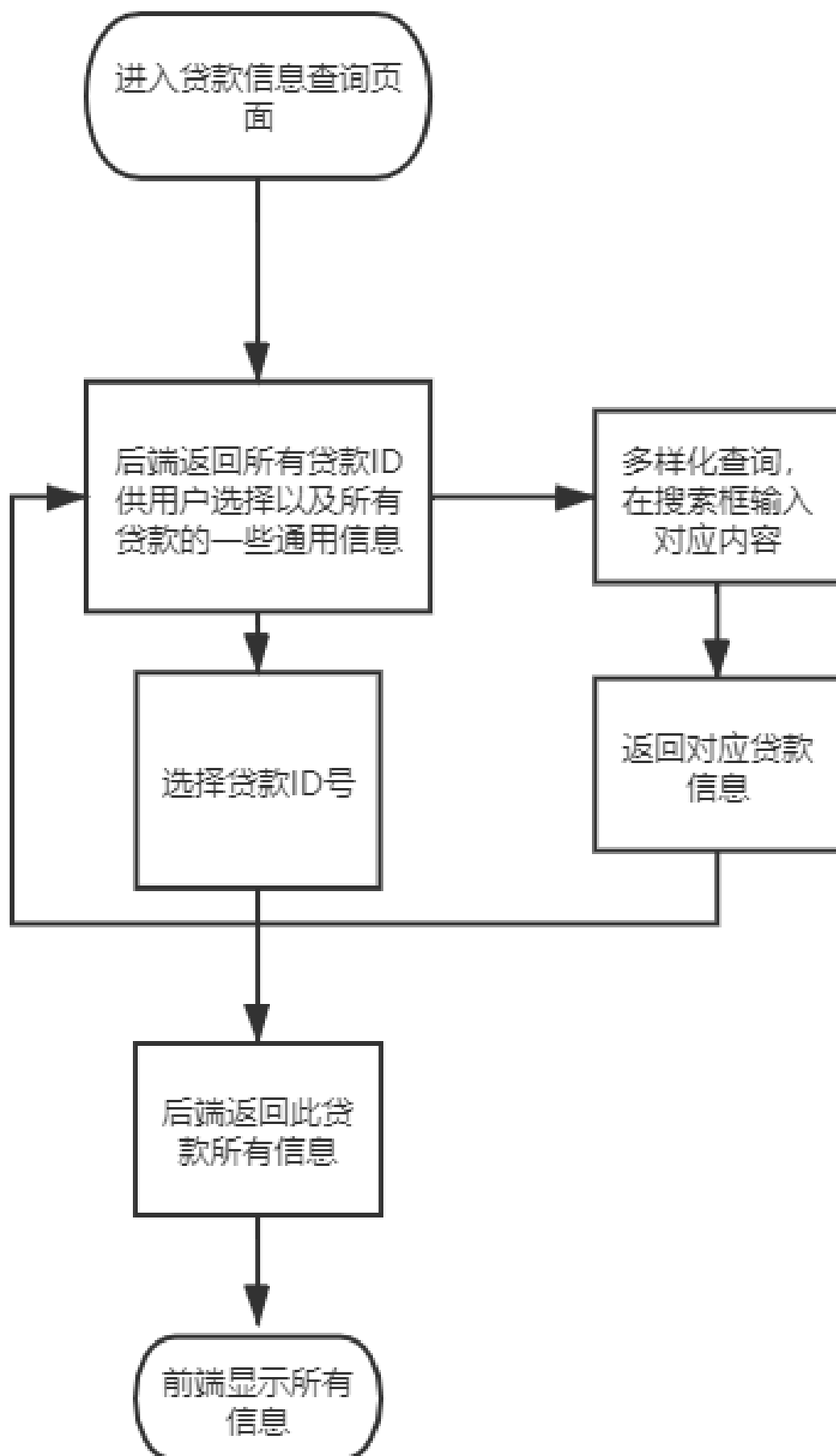
输入：

- 贷款ID
- 多样化查询数据

输出：

- 提交贷款ID之后返回该贷款详细信息：客户，银行，总贷款金额，余额，贷款发放状态
- 多样化查询返回满足条件的贷款ID，支行，总贷款金额，余额，贷款发放状态，例如如果想搜索所有贷款正在发放的贷款ID，可以在搜索框输入“贷款正在发放”

流程图：



3.4 业务统计模块

3.4.1 储蓄业务统计

输入：

- 支行
- 年份

输出：

- 按月统计：返回每个月该支行储蓄业务总金额，构成曲线图和表格两种形式；返回每个月该支行储蓄业务总客户数，构成曲线图和表格两种形式；
- 按季统计：返回每个季度该支行储蓄业务总金额，构成曲线图和表格两种形式；返回每个季度该支行储蓄业务总客户数，构成曲线图和表格两种形式；
- 按年统计：返回该年所有支行储蓄业务总金额，构成曲线图和表格两种形式；返回该年所有支行储蓄业务总客户数，构成曲线图和表格两种形式；(因为每年不同支行之间的比较我觉得更符合现实)

流程图：

进入储蓄业务统计页面

后端返回所有支行名供用户选择

选择支行名，并
填入年份(默认
2020)

后端统计每月、每个季度、该年所有支行对应的业务信息并以列表形式返回前端

前端使用echarts 表格和曲线显示 所有统计信息

3.4.2 贷款业务统计

输入：

- 支行
- 年份

输出：

- 按月统计：返回每个月该支行贷款业务总金额，构成曲线图和表格两种形式；返回每个月该支行贷款业务总客户数，构成曲线图和表格两种形式；
- 按季统计：返回每个季度该支行贷款业务总金额，构成曲线图和表格两种形式；返回每个季度该支行贷款业务总客户数，构成曲线图和表格两种形式；
- 按年统计：返回该年所有支行贷款业务总金额，构成曲线图和表格两种形式；返回该年所有支行贷款业务总客户数，构成曲线图和表格两种形式；(因为每年不同支行之间的比较我觉得更符合现实)

流程图：

进入贷款业务统计页面

后端返回所有支行名供用户选择

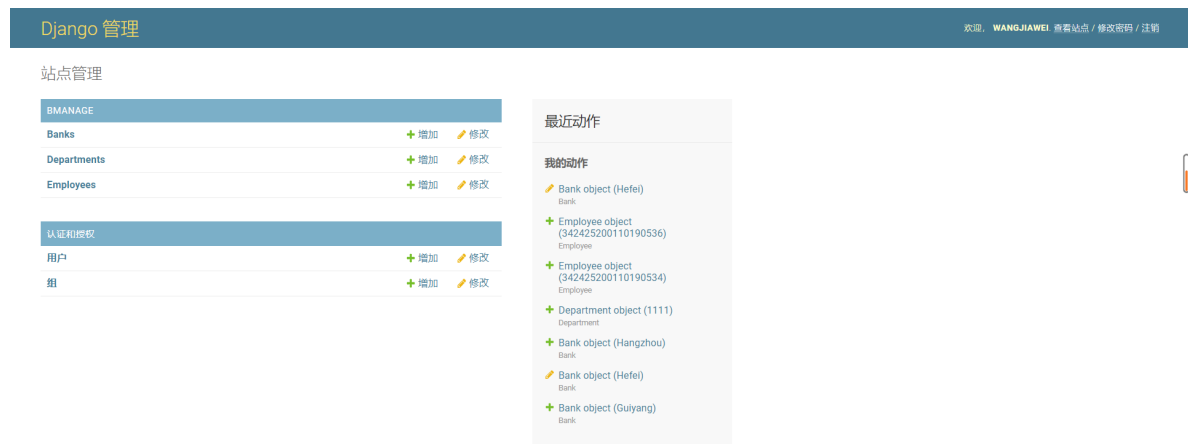
选择支行名，并
填入年份(默认
2020)

后端统计每月、每个季度、该年所有支行对应的业务信息并以列表形式返回前端

前端使用echarts 表格和曲线显示 所有统计信息

3.5 其他

由于支行、部门、员工不需要我们实现前端进行维护，所以我将这三个表链接到了django的管理平台上，可以通过创建超级账户，在django后台方便的进行这三个表内容的添加、删除、修改，方便进行测试和使用。



4 实现与测试

4.1 实现结果

4.1.1 客户管理

客户添加页面：

The screenshot shows the '客户信息添加' form within the '银行信息管理系统'. The form is titled '客户信息添加' and contains several input fields: 'ID' (with a hint 'Enter your 18 - digit ID'), 'Name', 'Phone Number', 'Address', 'Contact Name', 'Phone Number' (for contact), 'Email', 'Contact Relationship', 'Loan responder', and 'Account responder'. Each field has a corresponding input box. At the bottom of the form, there is a 'Submit' button. The footer of the page reads 'Bank System, Copyright © 2020-2020 JarvisJSTC'.

客户删除页面：

银行信息管理系统

信息管理 · 客户管理 · 账户管理 · 贷款管理 · 业务统计 ·

客户信息删除

ID: *

342425200110190534

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

客户修改页面（已提交ID）：

银行信息管理系统

信息管理 · 客户管理 · 账户管理 · 贷款管理 · 业务统计 ·

ID: *

342425200110190534

Name: *

王淑伟

Phone Number: *

17798385512

Address:

Enter Address

Contact Name: *

W

Phone Number: *

17798385512

Email:

Enter Contact Email

Contact Relationship: *

Father

Loan responder: *

342425200110190534

Account responder: *

342425200110190534

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

客户查询页面：

银行信息管理系统

信息管理 · 客户管理 · 账户管理 · 贷款管理 · 业务统计 ·

客户信息查询

ID: *

342425200110190534

Submit

ID	Name	Account Responder	Loan Responder	Contact Name
342425200110190534	王淑伟	Wjw	Wjw	W
342425200110190537	Fjc	Wjw	Wjw	F

查询

Bank System, Copyright © 2020-2020 JarvisUSTC

4.1.2 账户管理

开户页面：

银行信息管理系统

信息管理 · 客户管理 · 账户管理 · 贷款管理 · 业务统计 ·

账户开户

Account ID: *

Enter 6-digit Account ID

Customer ID: *

342425200110190534
342425200110190537

Bank Name:

Gulvans

Datetime:

年/月/日 --:--

Money: *

Enter Money

Save Account: ▾

Interestrate:

Enter Interestrate

Savetype:

Enter savetype

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

销户页面：

银行信息管理系统

信息管理 · 客户管理 · 账户管理 · 贷款管理 · 业务统计 ·

账户销户

Account ID: *

123455

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

账户修改页面（已提交ID）：

银行信息管理系统

信息管理 · 客户管理 · 账户管理 · 贷款管理 · 业务统计

账户信息修改

Account ID: *

123455

ID: *

342425200110190534

Bank Name:

Guiyang

Datetime:

2020年6月23日 13:37

Money: *

1000.0

SavingAccount

InterestRate:

0.0

Savetype:

Enter savetype

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

账户查询页面：

银行信息管理系统

信息管理 · 客户管理 · 账户管理 · 贷款管理 · 业务统计

账户信息查询

Account ID: *

123455

Submit

Account ID	Bank	Type	Money	Setting Time
123455	Guiyang	SavingAccount	1000.0	2020年6月23日 13:37
123456	Hangzhou	SavingAccount	1111111.0	2020年6月29日 19:51

Search

Bank System, Copyright © 2020-2020 JarvisUSTC

4.1.3 贷款管理

添加贷款页面：

银行信息管理系统

信息管理 · 客户管理 · 账户管理 · 贷款管理 · 业务统计

贷款信息添加

Loan ID: *

Enter 4-digit Loan ID

Customer ID: *

342425200110190534
342425200110190537

Bank Name:

Guiyang

Money: *

Enter Money

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

贷款删除页面：

银行信息管理系统

信息管理 · 客户管理 · 账户管理 · 贷款管理 · 业务统计

贷款信息删除

Loan ID: *

1111

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

贷款发放页面：

银行信息管理系统

信息管理 客户管理 账户管理 贷款管理 业务统计

贷款发放

Loan ID: 1111

Submit

Loan ID: None

Customer ID:

Money: Enter Money

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

贷款查询页面：

银行信息管理系统

信息管理 客户管理 账户管理 贷款管理 业务统计

贷款信息查询

Loan ID: 1111

Submit

Loan ID: 1113

Customer ID:

Bank Name: Hangzhou

Total Loan Amount: 200000.0

Balance: 199000.0

State: 贷款正在发放

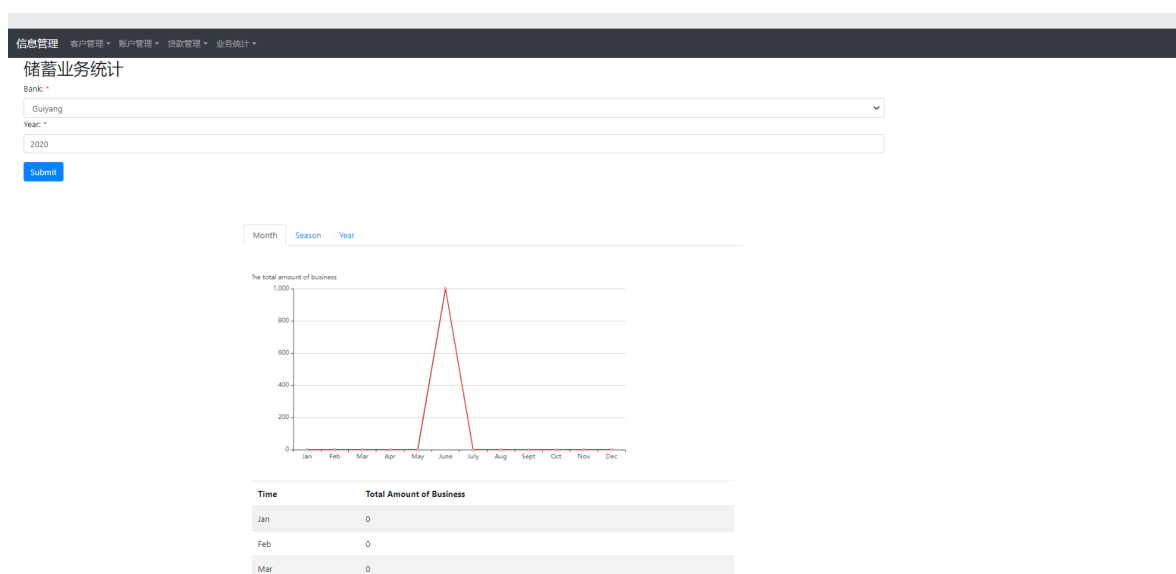
Loan ID	Bank	Total Loan Amount	Balance	State
1111	Guiyang	100000.0	100000.0	贷款未发放
1113	Hangzhou	200000.0	199000.0	贷款正在发放

搜索

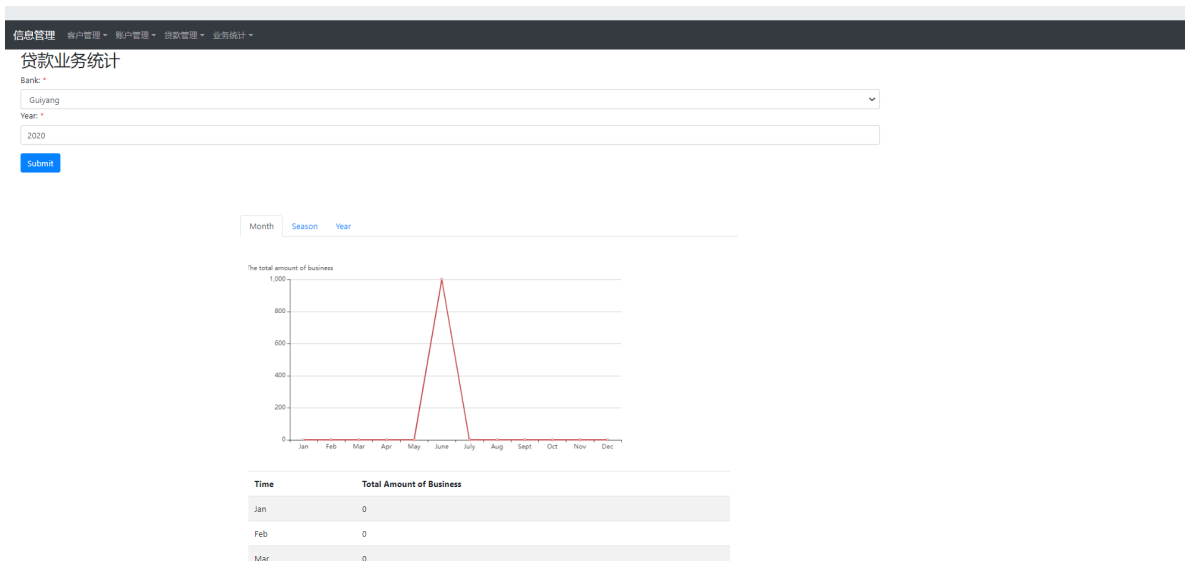
Bank System, Copyright © 2020-2020 JarvisUSTC

4.1.4 业务统计

储蓄业务统计（页面过长）：



贷款业务统计：



4.2 测试结果

4.2.1 客户管理

增加客户：

- 测试用例

客户信息添加

ID: *

Name: *

Phone Number: *

Address:

Contact Name: *

Phone Number: *

Email:

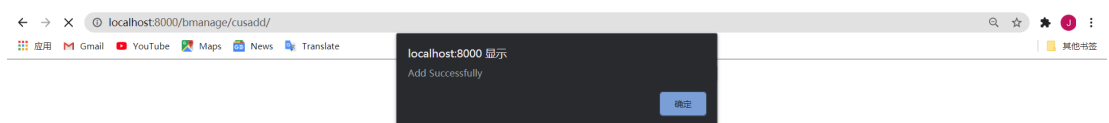
Contact Relationship: *

Loan responder: *

Account responder: *

Bank System, Copyright © 2020-2020 JarvisUSTC

- 测试结果



删除客户：

- 测试用例

客户信息删除

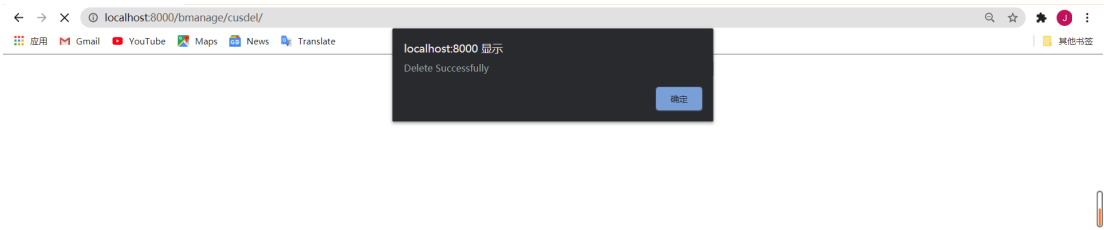
ID: *

342425200110190530

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

- 测试结果



- 测试用例

客户信息删除

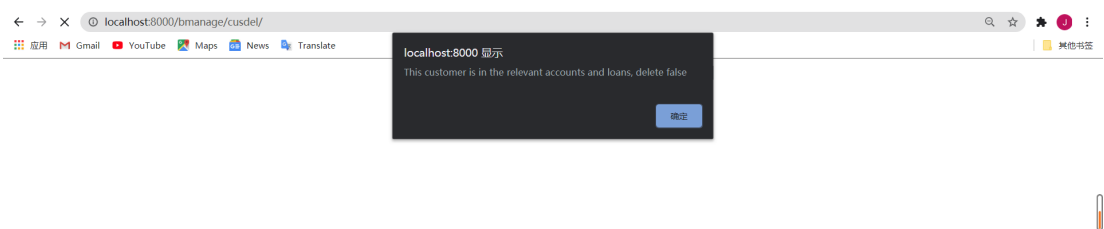
ID: *

342425200110190537

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

- 测试结果



客户信息修改：

- 测试用例

ID: *

342425200110190534

Name: *

王家伟

Phone Number: *

17398385512

Address:

Enter Address

Contact Name: *

张三

Phone Number: *

17398385512

Email:

Enter Contact Email

Contact Relationship: *

朋友

Loan responder: *

342425200110190534

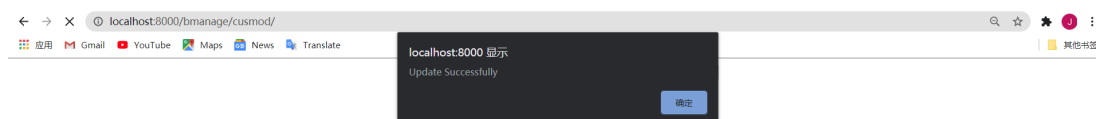
Account responder: *

342425200110190534

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

- 测试结果



客户信息查询：

- 测试用例

客户信息查询

ID: *

342425200110190537

Submit

ID	Name	Account Responder	Loan Responder	Contact Name
342425200110190534	王家伟	Wjw	Wjw	张三
342425200110190537	Fyc	Wjw	Wjw	F

搜索

输入以开始搜索

Bank System, Copyright © 2020-2020 JarvisUSTC

- 测试结果

客户信息查询

ID: *
342425200110190537

Submit

ID	Name	Account Responder	Loan Responder	Contact Name
342425200110190537	Fyc	Wjw	Wjw	F

搜索

F

Bank System, Copyright © 2020-2020 JarvisUSTC

4.2.2 账户管理

开户：

- 测试用例

账户开户

Account ID: *
666666

Customer ID: *
342425200110190534
342425200110190537

Bank Name:
hefei

Datetime:
2020/06/30 17:53

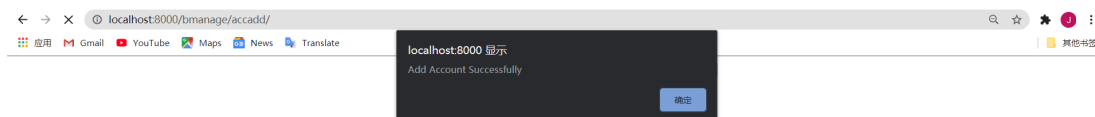
Money: *
1000000

Check Account
Overdraft:
10

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

- 测试结果



销户：

- 测试用例

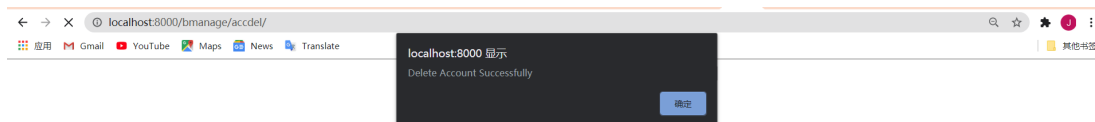
账户销户

Account ID: *
666666

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

- 测试结果



账户信息修改：

- 测试用例

账户信息修改

Account ID: *
123455

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

账户信息修改

Account ID: *

123455

ID: *

342425200110190534

Bank Name:

Guiyang

Datetime:

2020年6月23日 13:37

Money: *

1000.0

SavingAccount

Interstrate:

0.0

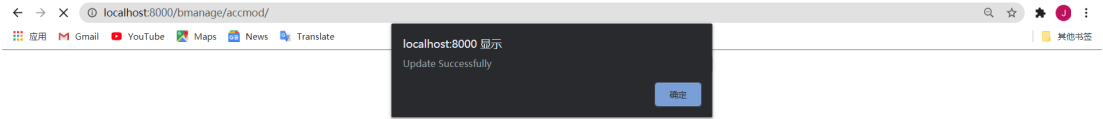
Savetype:

Enter savetype

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

• 测试结果



账户查询:

• 测试用例

账户信息查询

Account ID: *

123455

Submit

Account ID	Bank	Type	Money	Setting Time
123455	Guiyang	SavingAccount	1000.0	2020年6月23日 13:37
123456	Hangzhou	SavingAccount	1111111.0	2020年6月29日 19:51

搜索

输入以开始搜索

Bank System, Copyright © 2020-2020 JarvisUSTC

• 测试结果

账户信息查询

Account ID: *

123455

Submit

Account ID	Bank	Type	Money	Setting Time
123455	Guiyang	SavingAccount	1000.0	2020年6月23日 13:37

搜索

Guiyang

Bank System, Copyright © 2020-2020 JarvisUSTC

账户信息修改

Account ID: *

123455

ID: *

342425200110190534

Bank Name:

Guiyang

Datetime:

2020年6月23日 13:37

Money: *

1000.0

SavingAccount

Interestrates:

0.0

Savetype:

Enter savetype

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

4.2.3 贷款管理

贷款添加：

- 测试用例

贷款信息添加

Loan ID: *

1314

Customer ID: *

342425200110190534
342425200110190537

Bank Name:

Guiyang

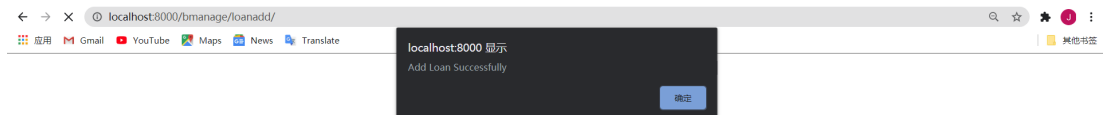
Money: *

100000

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

- 测试结果



贷款删除：

- 测试用例

贷款信息删除

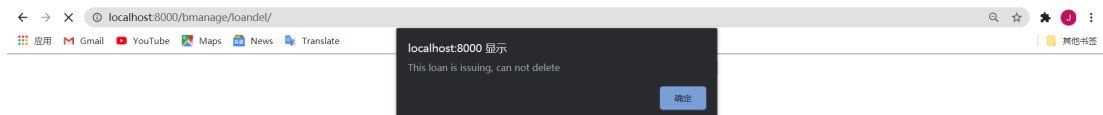
Loan ID: *

1113

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

- 测试结果



贷款查询：

- 测试用例

贷款信息查询

Loan ID: *

1111

Submit

Loan ID: *

1314

Customer ID: *

Bank Name:

Guiyang

Total Loan Amount: *

100000.0

Balance: *

100000.0

State:

贷款未发放

Loan ID	Bank	Total Loan Amount	Balance	State
1111	hefei	1000000.0	1000000.0	贷款未发放
1113	Hangzhou	200000.0	199000.0	贷款正在发放
1314	Guiyang	100000.0	100000.0	贷款未发放

搜索

Bank System, Copyright © 2020-2020 JarvisUSTC

• 测试结果

贷款信息查询

Loan ID: *

1111

Submit

Loan ID: *

1314

Customer ID: *

Bank Name:

Guiyang

Total Loan Amount: *

100000.0

Balance: *

100000.0

State:

贷款未发放

Loan ID	Bank	Total Loan Amount	Balance	State
1111	hefei	1000000.0	1000000.0	贷款未发放
1314	Guiyang	100000.0	100000.0	贷款未发放

搜索

贷款未发放

Bank System, Copyright © 2020-2020 JarvisUSTC

贷款信息查询

Loan ID: *

1111

Submit

Loan ID: *

1111

Customer ID: *

342425200110190534

Bank Name:

hefei

Total Loan Amount: *

1000000.0

Balance: *

1000000.0

State:

贷款未发放

Loan ID	Bank	Total Loan Amount	Balance	State
1111	hefei	1000000.0	1000000.0	贷款未发放
1113	Hangzhou	200000.0	199000.0	贷款正在发放
1314	Guiyang	100000.0	100000.0	贷款未发放

搜索

Bank System, Copyright © 2020-2020 JarvisUSTC

贷款发放:

- 测试用例

贷款发放

Loan ID: *

1111

Submit

Loan ID: *

1111

Customer ID: *

342425200110190534

Money: *

1000

Submit

Bank System, Copyright © 2020-2020 JarvisUSTC

- 测试结果（贷款查询的时候是未发放，这次发放之后变成正在发放，所以执行正确）

Loan ID	Bank	Total Loan Amount	Balance	State
1111	hefei	1000000.0	999000.0	贷款正在发放

4.2.4 业务统计

储蓄业务统计:

- 测试用例

储蓄业务统计

Bank: *

Guiyang

Year: *

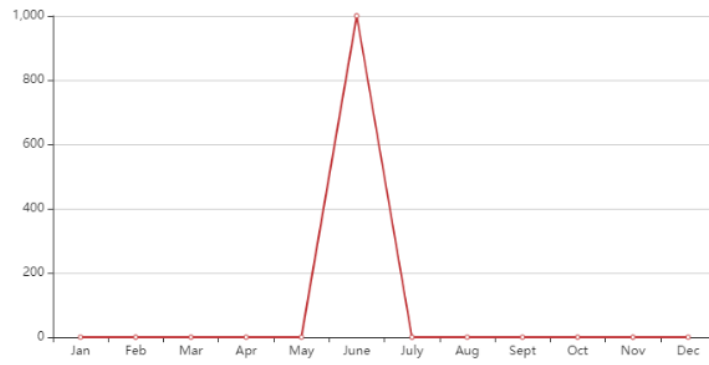
2020

Submit

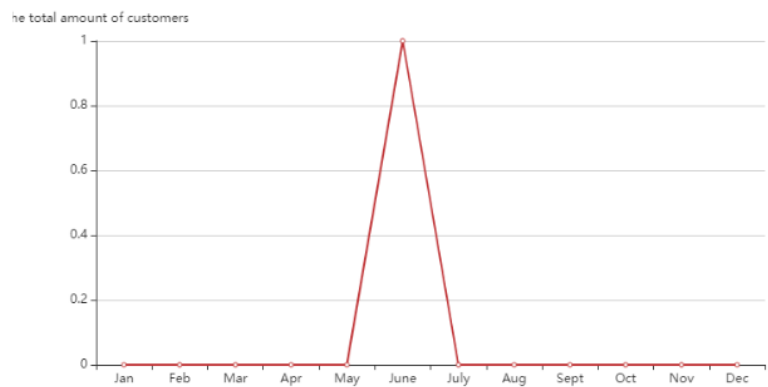
- 测试结果

Month Season Year

The total amount of business

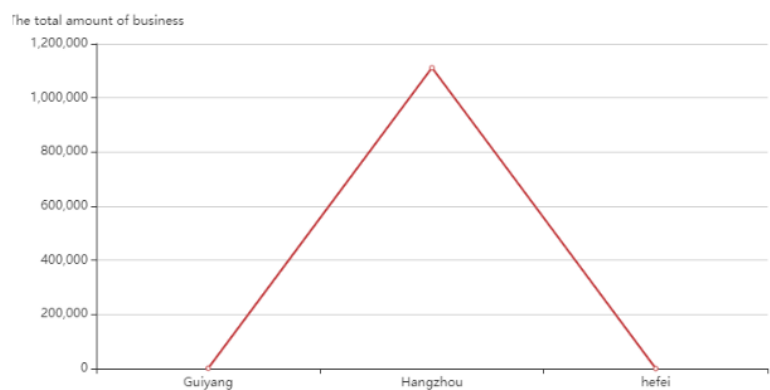


Time	Total Amount of Business
Jan	0
Feb	0
Mar	0
Apr	0
May	0
June	1000.0
July	0
Aug	0
Sept	0
Oct	0



Time	Total Amount of Customer
Jan	0
Feb	0
Mar	0
Apr	0
May	0
June	1
July	0
Aug	0
Sept	0
Oct	0
Nov	0
Dec	0

Month
Season
Year



Bank	Total Amount of Business
Guiyang	1000.0
Hangzhou	1111111.0
hefei	0

贷款业务统计：

- 测试用例

贷款业务统计

Bank: *

Hangzhou

Year: *

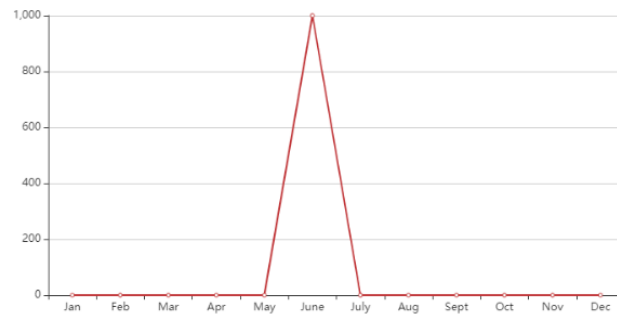
2020

Submit

• 测试结果

Month Season Year

The total amount of business

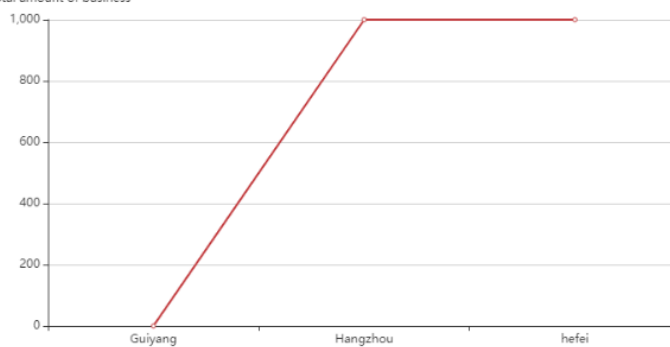


Time Total Amount of Business

Jan	0
Feb	0
Mar	0
Apr	0
May	0
June	1000.0
July	0
Aug	0
Sept	0

Month Season Year

The total amount of business



Bank Total Amount of Business

Guiyang	0
Hangzhou	1000.0
hefei	1000.0

he total amount of customers



5 总结与讨论

这次数据库实验不同于以往的课程实验，以往的课程实验专注于本课程知识点的练习和实践，而本次实验融合了更多工程的元素，不仅仅考虑数据库的知识，还需要考虑前端的设计、后端对数据库的操作、前后端的交互，可以说是个小型的软件工程实践。总的来说，本次实验全面的考察了我们在软件开发方面的综合能力。

在这次实验中，我使用了软件工程课程学习的Django框架进行开发，虽然之前接触过一次，但是对此框架还是很不熟练，实现的过程中逐渐加深了对此框架的理解，了解了工作流程和设计方法。此外前端没有用传统的Html, css直接设计，而是学习了Bootstrap 4前端框架，可以设计出更美观的页面，对于表格和曲线图，我也是使用了Echarts插件。在这次实验中，确实学到了很多数据库之外的知识，培养了学习能力、综合能力。

对于一个系统来说，如何处理报错确实是很重要的部分，在这次实验中，我尽可能的通过前端限制输入规则来简化后端的错误处理，不过也有很多未知的错误可能出现，还是需要花费很多精力去考虑。

综上所述，本次实验还是极大地锻炼了我的综合能力，既对数据库知识有了充分的练习，也对开发流程的把控进行了训练，我觉得是很不错的一次开发实战。