

“银行业务管理系统”

系统设计与实现报告

姓名：王章瀚

学号：PB18111697

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	4
2.3.1	数据库设计	4
2.3.2	逻辑数据库结构图	5
2.3.3	物理数据库模型	5
3	详细设计	6
3.1	客户管理模块	6
3.1.1	客户添加模块	6
3.1.2	客户查询模块	7
3.1.3	客户删除模块	8
3.1.4	客户修改模块	9
3.2	账户管理模块	9
3.2.1	账户添加模块	9
3.2.2	账户查询模块	10
3.2.3	账户删除模块	11
3.2.4	账户修改模块	12
3.3	贷款管理模块	12
3.3.1	贷款添加模块	12
3.3.2	贷款查询模块	13
3.3.3	贷款删除模块	14
3.3.4	贷款支付模块	15
3.4	业务统计模块	15
3.4.1	储蓄账户业务统计模块	15
3.4.2	贷款业务统计模块	16
3.5	登录管理模块	17
3.5.1	登录及登录状态验证	17
3.5.2	登出	18

3.7 用户输入校验模块	18
3.6 系统运行日志模块	20
4 实现与测试	20
4.1 实现结果	20
4.1.1 添加客户	20
4.1.2 查询客户	21
4.1.3 修改客户	23
4.1.4 删除客户	23
4.1.5 开户	24
4.1.6 搜索账户	25
4.1.7 修改账户	27
4.1.8 删除账户	28
4.1.9 新增贷款	29
4.1.10 查询贷款	30
4.1.11 查询贷款者	31
4.1.12 支付贷款	31
4.1.13 贷款删除	32
4.1.14 储蓄账户业务统计	33
4.1.15 贷款业务统计	36
4.1.16 登录与登出	37
4.1.17 其他琐碎部分	39
4.2 测试结果	39
4.2.1 添加客户	39
4.2.2 查询客户	40
4.2.3 修改客户	41
4.2.4 删除客户	42
4.2.5 开户	43
4.2.6 搜索账户	44
4.2.7 修改账户	45
4.2.8 删除账户	47
4.2.9 新增贷款	47
4.2.10 查询贷款	48
4.2.11 查询贷款者	50
4.2.12 支付贷款	51
4.2.13 贷款删除	52
4.2.14 储蓄账户业务统计	53
4.2.15 贷款业务统计	55
4.2.16 登录与登出	56
4.3 实现中的难点问题及解决	58
4.3.1 前端 Web 界面实现难点	58
4.3.2 业务统计模块的实现	58

4.3.3 后端开发过程中的测试	59
4.3.4 抛出异常的处理	59
5 总结与讨论	59

1 概述

1.1 系统目标

本系统旨在为某银行（下称憨憨银行）开发一个具有 Web 前端的银行业务管理系统。系统要求使用 MySQL 作为后端的 DBMS，并以 Flask 完成前后端通信。该系统需要满足甲方的无数奇奇怪怪的需求。

1.2 需求说明

此需求说明为甲方提供：

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。

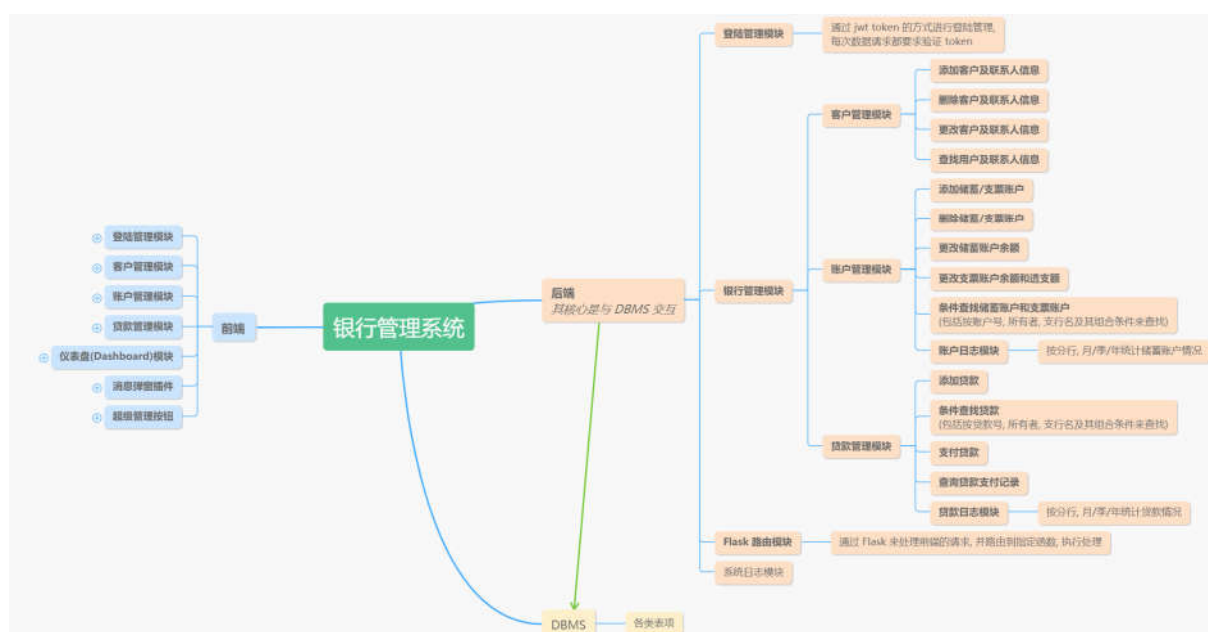
1.3 本报告的主要贡献

本报告详述了本银行管理系统的总体设计，数据库设计，前端设计，后端适配设计，前后端连接方式等，内容详尽，应有尽有。

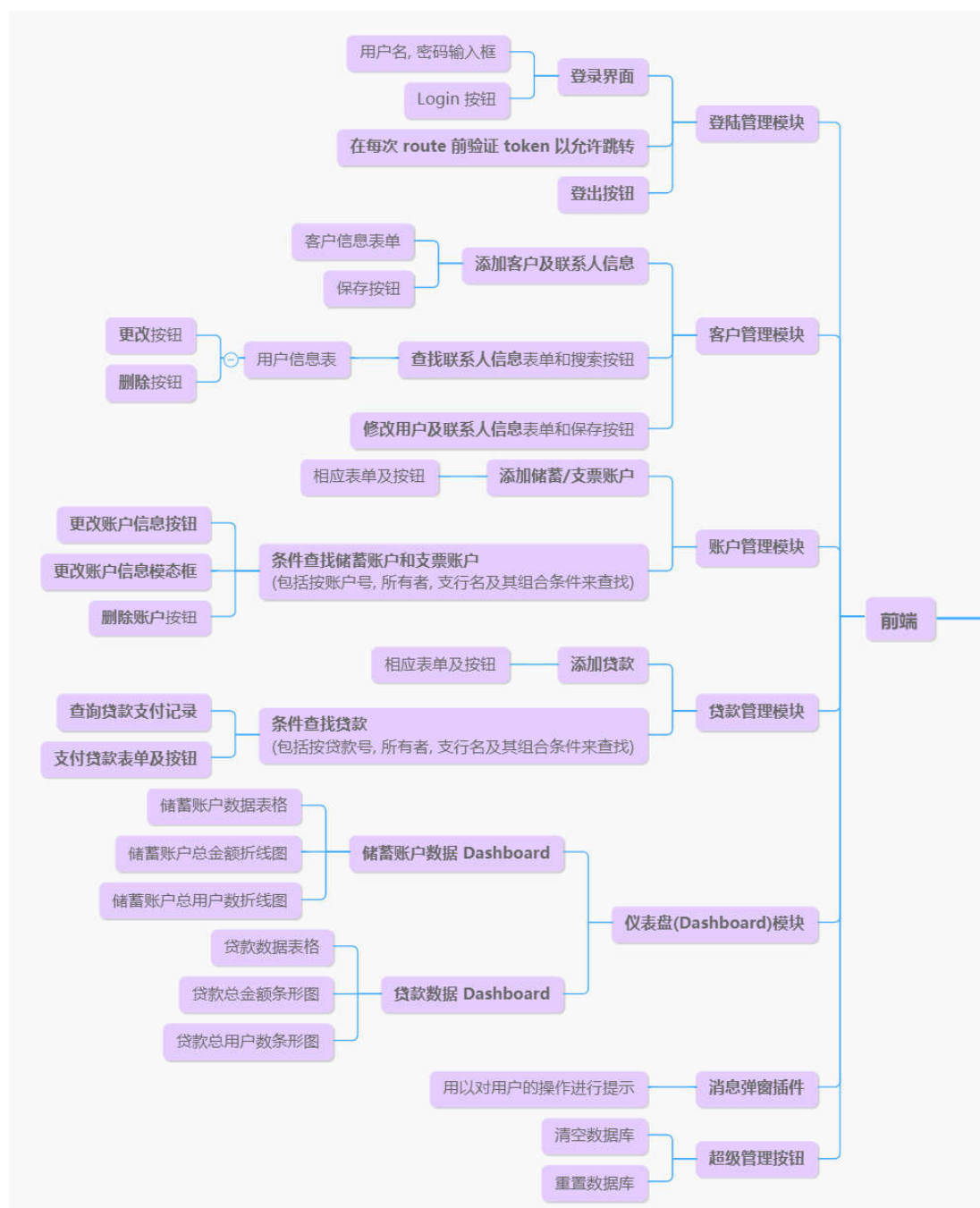
2 总体设计

2.1 系统模块结构

用 Processon 系统画出系统模块结构如下：



左侧前端的详细信息在上图中囿于空间，无法展示，故令给出下图：

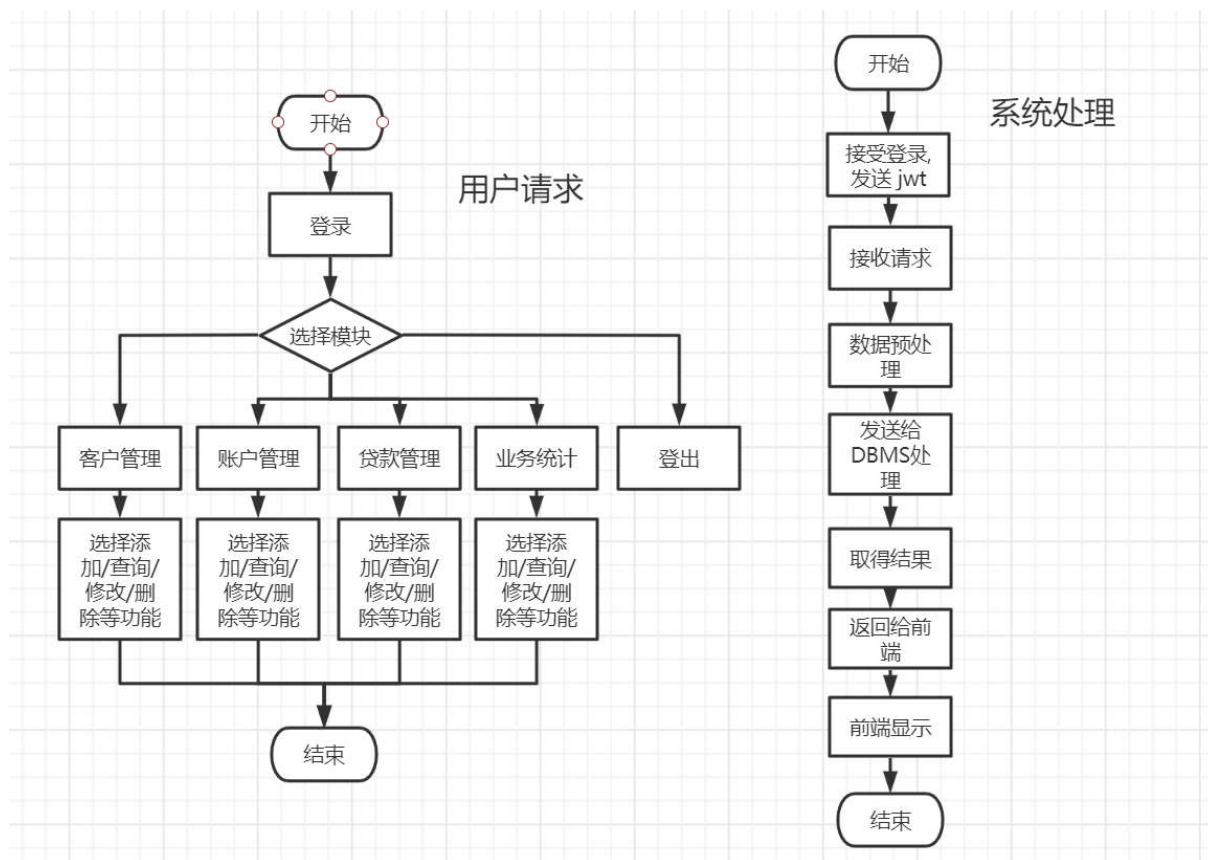


总体而言, 整个系统由前端服务器, 后端服务器及 DBMS 系统构成. 其中后端提供了前端查询所需要的各种操作. 前端所需功能在后端基本上都有对应的模块来处理, 前后端基本上是对应的. 但为了用户友好, 前端需要不同于后端地调整模块的层次.

此外, 为了满足 Dashboard 给出“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”的需求, 需要有日志来记录历史信息.

2.2 系统工作流程

主要是用户的请求和系统的处理:

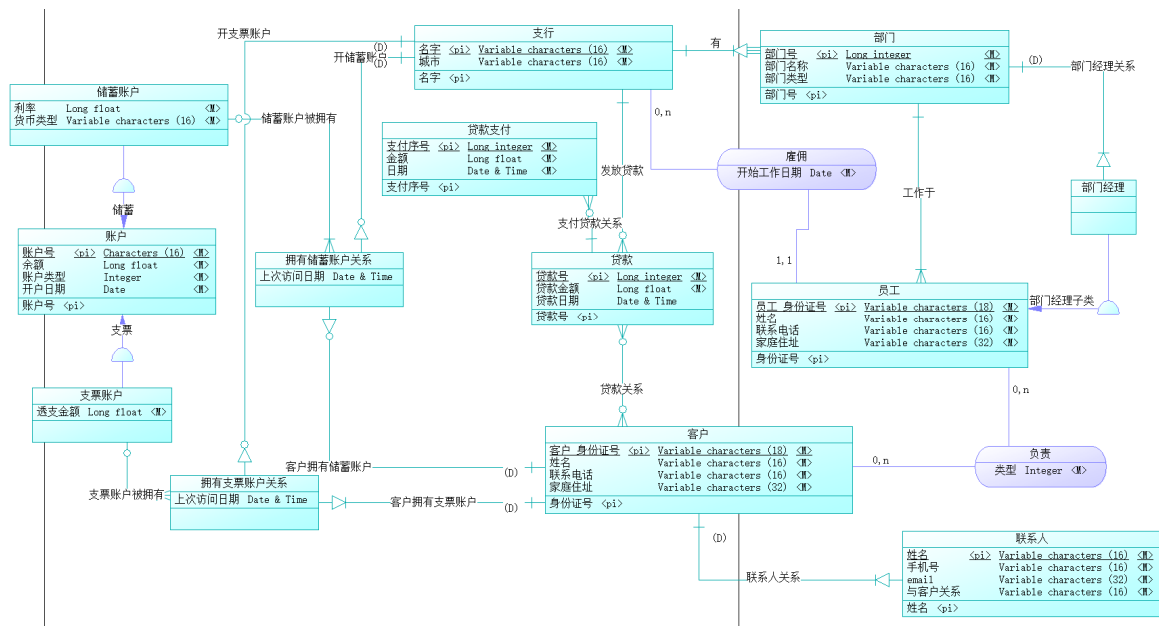


2.3 数据库设计

LX 说过: “一图胜过千言万语.” 下面的说明主要通过图给出.

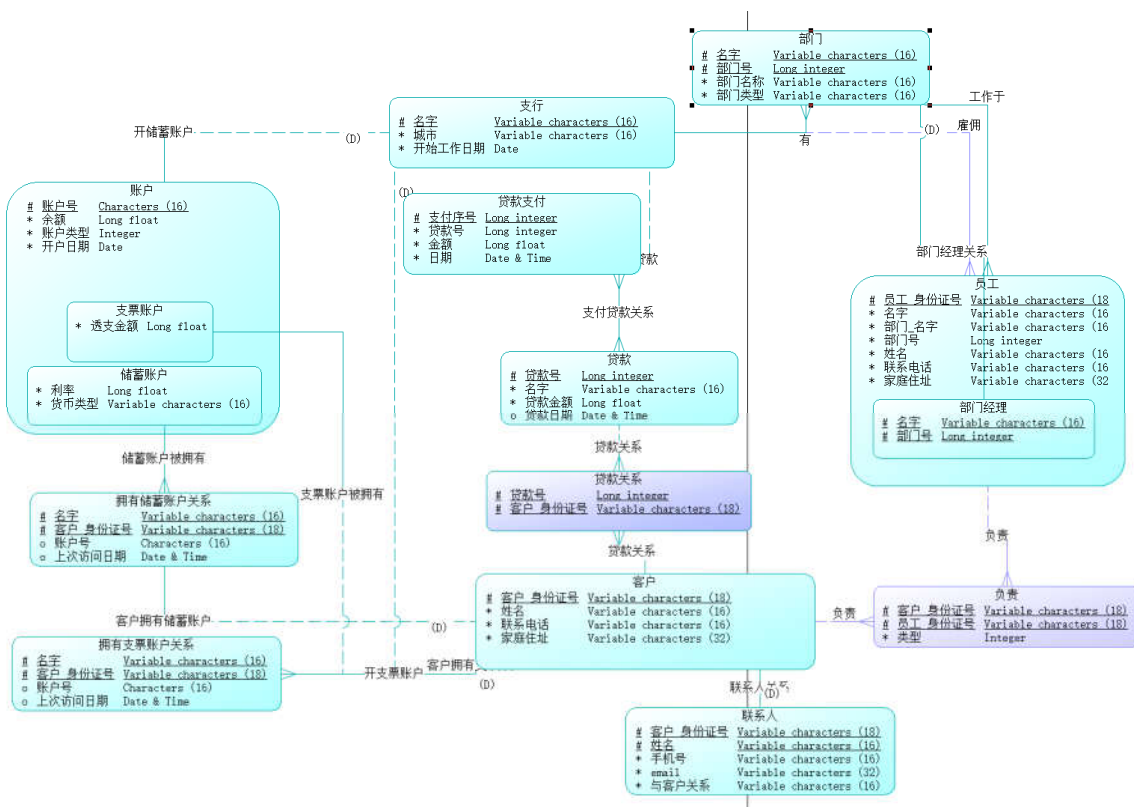
2.3.1 数据库设计

系统的 ER 图如下.



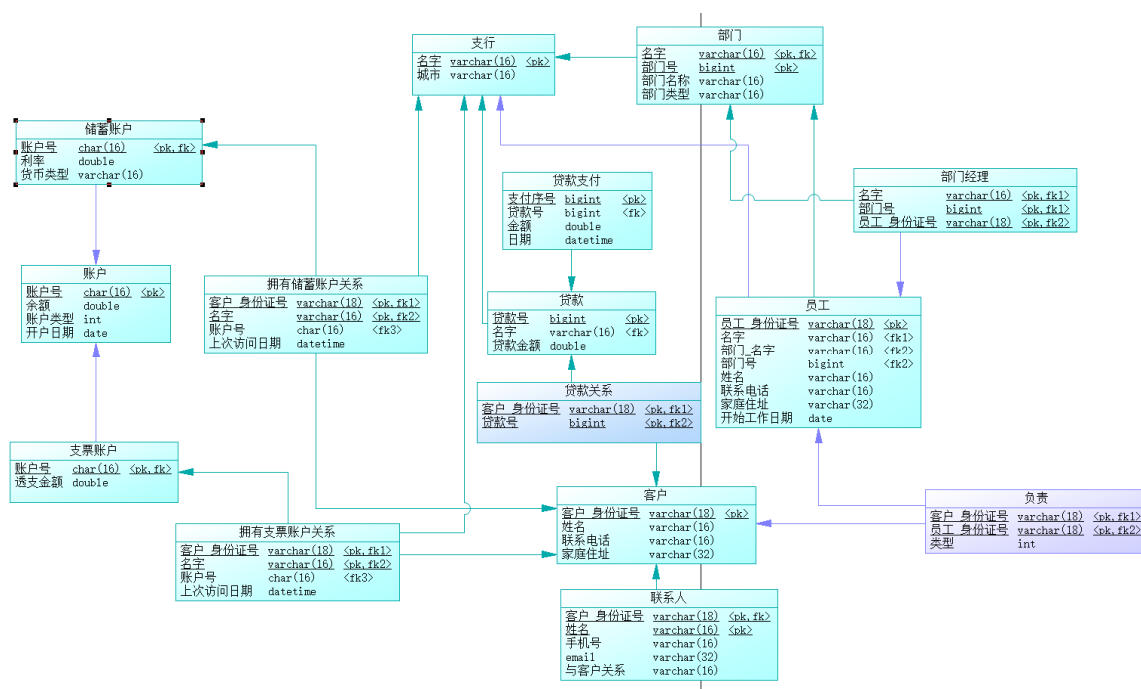
2.3.2 逻辑数据库结构图

逻辑数据库结构如下:



2.3.3 物理数据库模型

最终生成的物理数据库模型如下:



这其中物理模型里，对贷款号和贷款支付序号加了 Identity 标记，以使得最终生成的代码中，它们具有 auto_increment 属性。

3 详细设计

本节讲述详细的模块设计。这包含前端 webui 模块，后端 bankdb 模块及 DBMS 模块的略述。其中一些主要操作就把前后端合在一起描述，会比较方便。

3.1 客户管理模块

3.1.1 客户添加模块

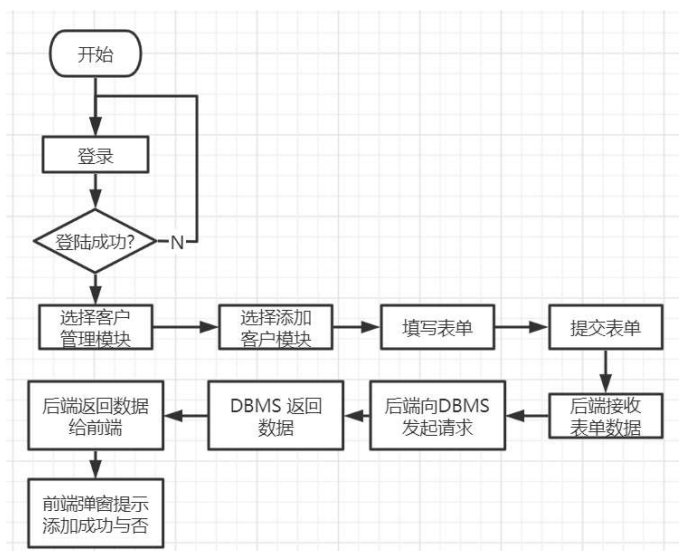
添加客户要求输入详尽的客户及其联系人资料。这包括：

1. 客户身份证号
2. 客户名
3. 客户电话
4. 客户住址
5. 客户联系人姓名
6. 客户联系人电话
7. 客户联系人邮箱
8. 客户与联系人的关系

此信息从前端传给后端后，后端尝试添加，如果成功，则向前端返回成功添加的信息。否则，可能抛出以下异常：

1. 管理员登录过期异常
2. 输入不符合格式异常
3. 重复添加异常(即客户身份证号重复)

流程图如下:



3.1.2 客户查询模块

查询客户要求输入查询条件. 可以包括:

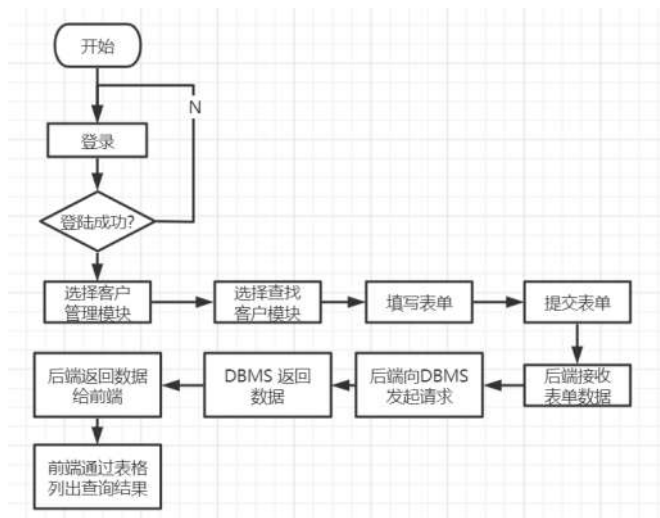
1. 客户身份证号
2. 客户名

此信息从前端传给后端后, 后端尝试查找并返回一个列表, 列表元素是字典, 其键值包括:

1. 客户身份证号
2. 客户名
3. 客户电话
4. 客户住址
5. 客户联系人姓名
6. 客户联系人电话
7. 客户联系人邮箱
8. 客户与联系人的关系

当登录过期时, 可能引起 “管理员登录过期异常”.

流程图如下:



3.1.3 客户删除模块

用户选择前面查询的结果，点击对应条目的删除按钮即可删除。

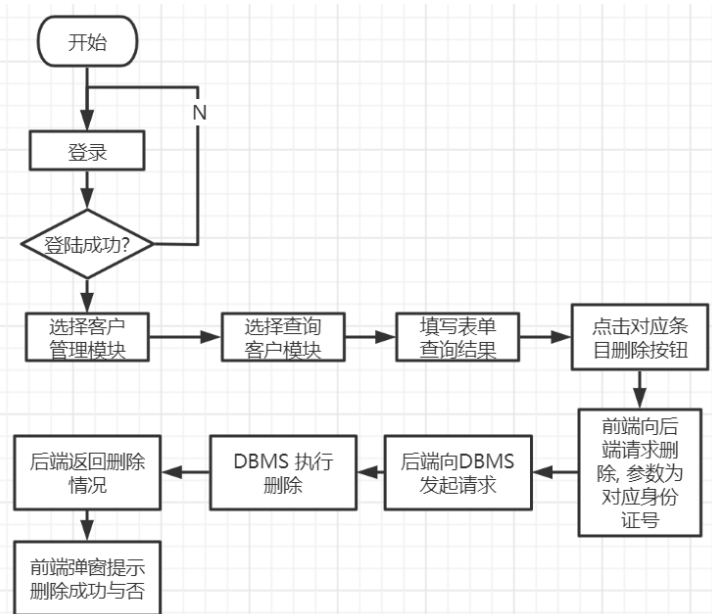
当用户点击删除按钮的时候，前端向后端发起请求，并传递用户的身份证号作为参数。

后端收到身份证号，即可删除对应条目，并向前端返回删除情况。

如果删除失败，可能抛出以下异常：

1. 管理员登录过期异常
2. 仍有储蓄账户异常
3. 仍有支票账户异常
4. 仍有贷款异常

流程图如下：



3.1.4 客户修改模块

用户选择前面查询的结果, 点击对应条目的修改按钮即可修改客户信息.

当用户点击修改按钮的时候, 前端切换到修改界面, 其中用户身份证号是不可修改的. 其余项均可修改.

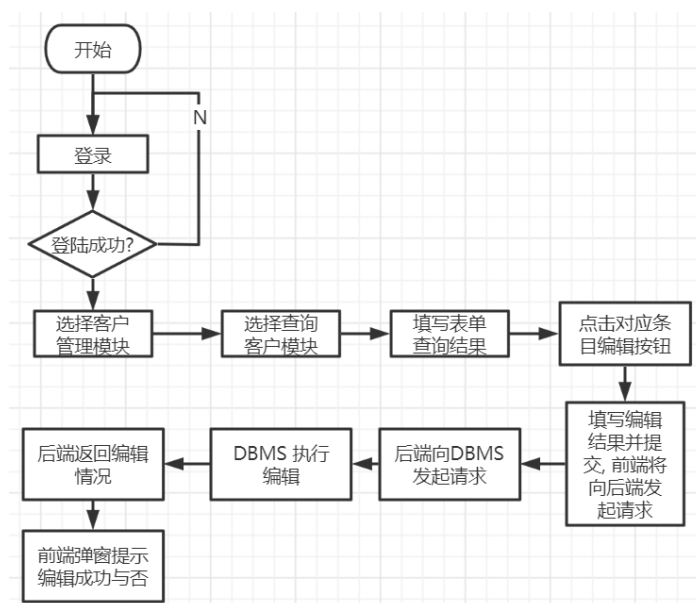
前端需要向后端发起修改客户请求, 并传递所有新信息以供后端修改.

后端收到修改参数后, 向 DBMS 发起请求即可修改对应条目, 并向前端返回修改情况.

如果修改失败, 可能抛出以下异常:

1. 管理员登录过期异常
2. 输入不符合格式异常

流程图如下:



3.2 账户管理模块

3.2.1 账户添加模块

添加账户要求输入详尽的账户和客户资料. 这包括:

1. 账户号
2. 客户身份证号
3. 支行名称
4. 账户类型
5. 对于储蓄账户
 - a) 利率
 - b) 货币类型

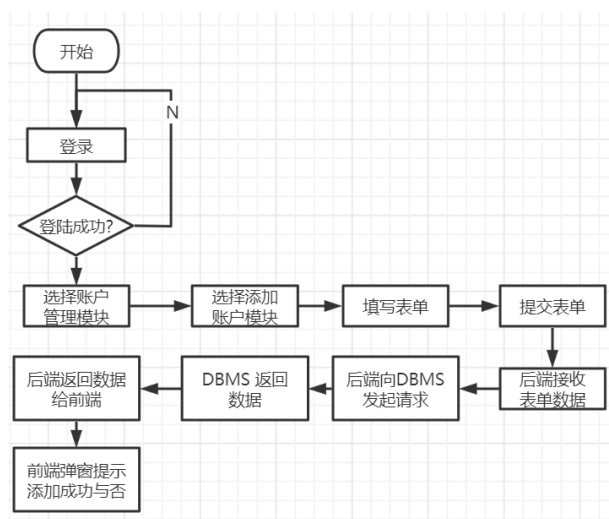
此信息从前端传给后端后, 后端尝试添加, 如果成功, 则向前端返回成功添加的信息. 否则, 可能抛

出以下异常:

1. 管理员登录过期异常
2. 输入不符合格式异常
3. 重复添加异常(即(账户号, 客户身份证号) 二元组重复)

注意我们允许相同账户号被多次添加客户绑定, 这是为了支持多客户共有账户.

流程图如下:



3.2.2 账户查询模块

查询账户要求输入查询条件. 可以包括:

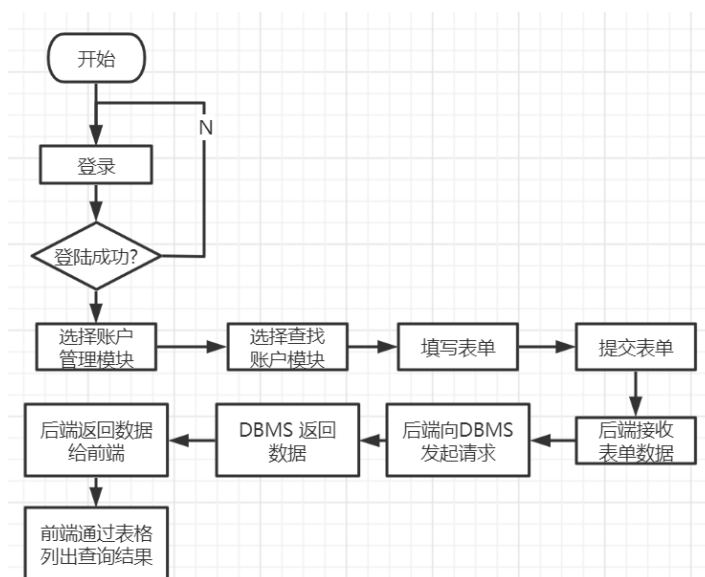
1. 客户身份证号
2. 账户号
3. 支行名称

此信息从前端传给后端后, 后端尝试查找并返回一个列表, 列表元素是字典, 其键值包括:

1. 账户号
2. 支行名
3. 客户身份证号
4. 账户类型
5. 账户余额
6. 对于储蓄账户
 - a) 利率
 - b) 货币类型
7. 对于支票账户
 - a) 透支量
8. 上次访问时间(针对于(账户号, 客户身份证号) 二元组而言)

当登录过期时, 可能引起 “管理员登录过期异常”.

流程图如下:



3.2.3 账户删除模块

用户选择前面查询的结果，点击对应条目的删除按钮即可删除。

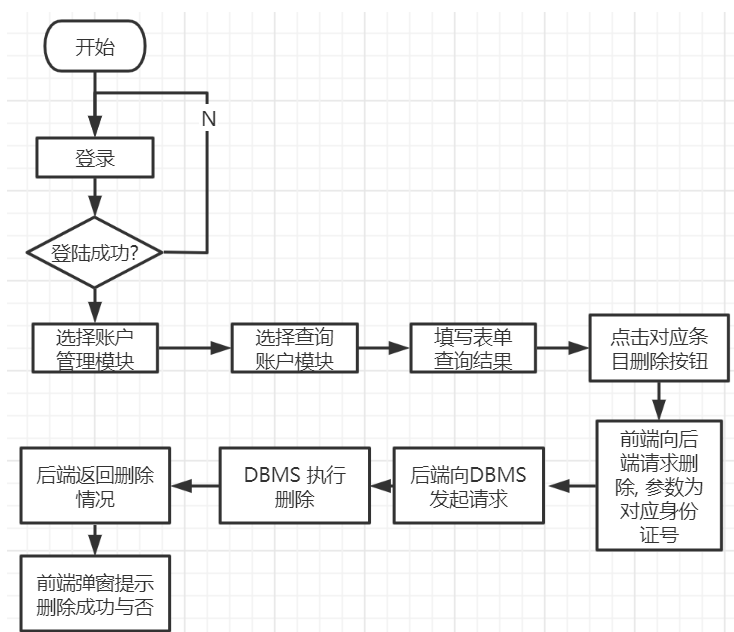
当用户点击删除按钮的时候，前端向后端发起请求，并传递账户号作为参数。

后端收到账户号，即可删除对应条目，并向前端返回删除情况。

如果删除失败，可能抛出以下异常：

1. 管理员登录过期异常
2. 余额非零异常
3. 对于支票账户：
 - a) 透支量非零异常

流程图如下：



3.2.4 账户修改模块

用户选择前面查询的结果, 点击对应条目的修改按钮即可修改账户信息.

当用户点击修改按钮的时候, 前端弹出账户修改的模态框, 其中用户身份证号, 账户支行, 账户类型等信息是不可修改的. 其余项包括余额, 利率, 货币类型, 透支额等均可修改.

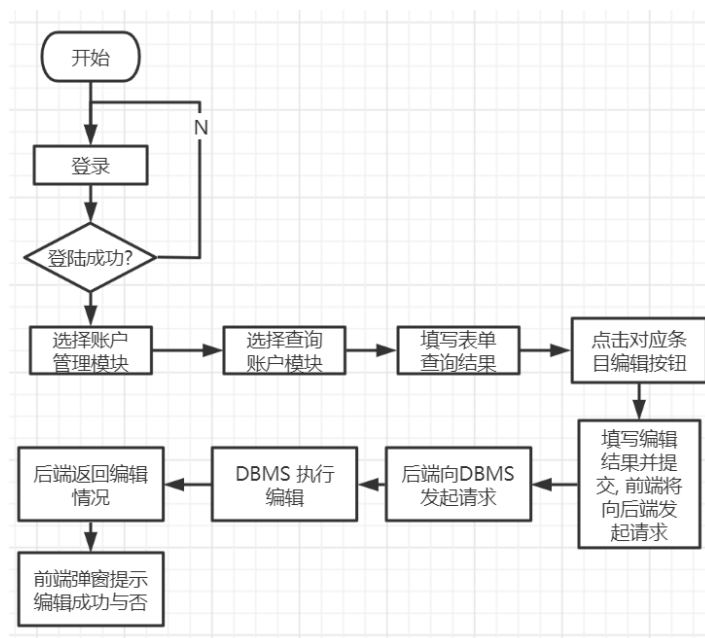
前端需要向后端发起修改客户请求, 并传递所有新信息以供后端修改.

后端收到修改参数后, 向 DBMS 发起请求即可修改对应条目, 并向前端返回修改情况.

如果修改失败, 可能抛出以下异常:

1. 管理员登录过期异常
2. 输入不符合格式异常

流程图如下:



3.3 贷款管理模块

3.3.1 贷款添加模块

添加账户要求输入详尽的贷款和客户资料. 这包括:

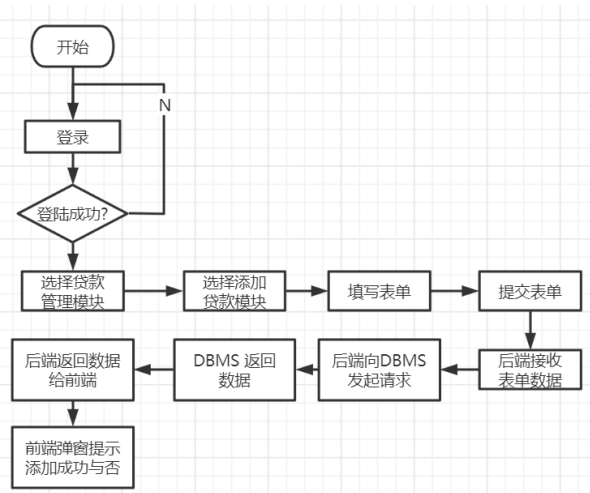
1. 支行名
2. 贷款额
3. 客户号(可以输入多个)

此信息从前端传给后端后, 后端尝试添加, 如果成功, 则向前端返回成功添加的信息. 否则, 可能抛出以下异常:

1. 管理员登录过期异常

2. 输入不符合格式异常

流程图如下:



3.3.2 贷款查询模块

查询贷款要求输入查询条件. 可以包括:

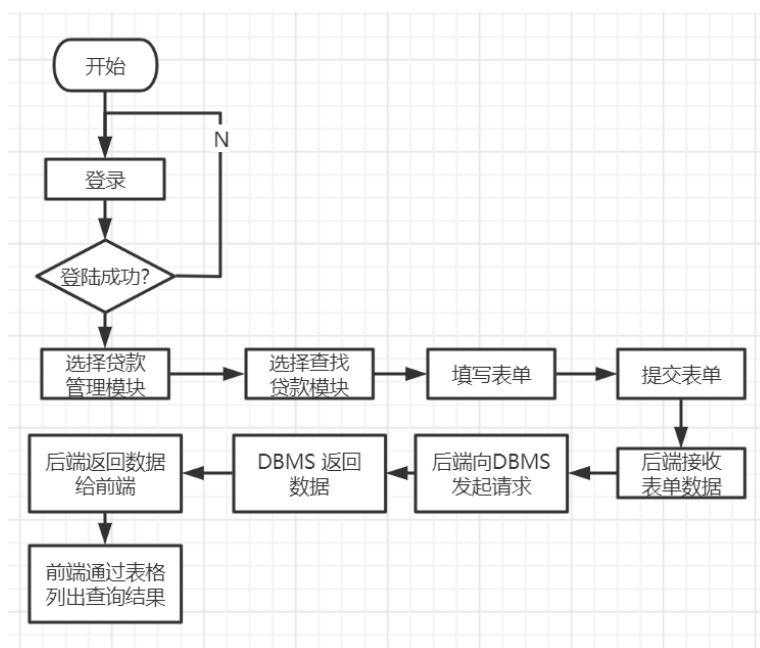
1. 客户身份证号
2. 贷款号
3. 支行名称

此信息从前端传给后端后, 后端尝试查找并返回一个列表, 列表元素是字典, 其键值包括:

1. 贷款号
2. 支行名
3. 贷款量

当登录过期时, 可能引起“管理员登录过期异常”.

流程图如下:



3.3.3 贷款删除模块

用户选择前面查询的结果，点击对应条目的删除按钮即可删除。

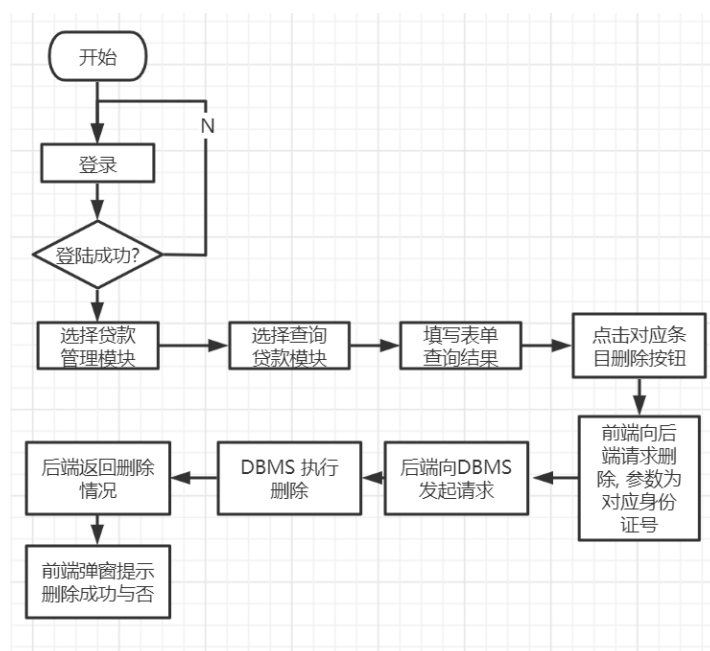
当用户点击删除按钮的时候，前端向后端发起请求，并传递贷款号作为参数。

后端收到账户号，确认贷款已经付清后，即可删除对应条目，并向前端返回删除情况。

如果删除失败，可能抛出以下异常：

1. 管理员登录过期异常
2. 贷款正在支付中异常

流程图如下：



3.3.4 贷款支付模块

用户选择前面查询的结果, 点击对应条目的修改按钮即可查看并修改贷款的支付信息.

当用户点击修改按钮的时候, 前端切换到贷款详情选项卡, 其中贷款号, 支行名, 贷款量, 已付总额度等息是不可修改的. 而支付量可修改. 并提供一个支付按钮

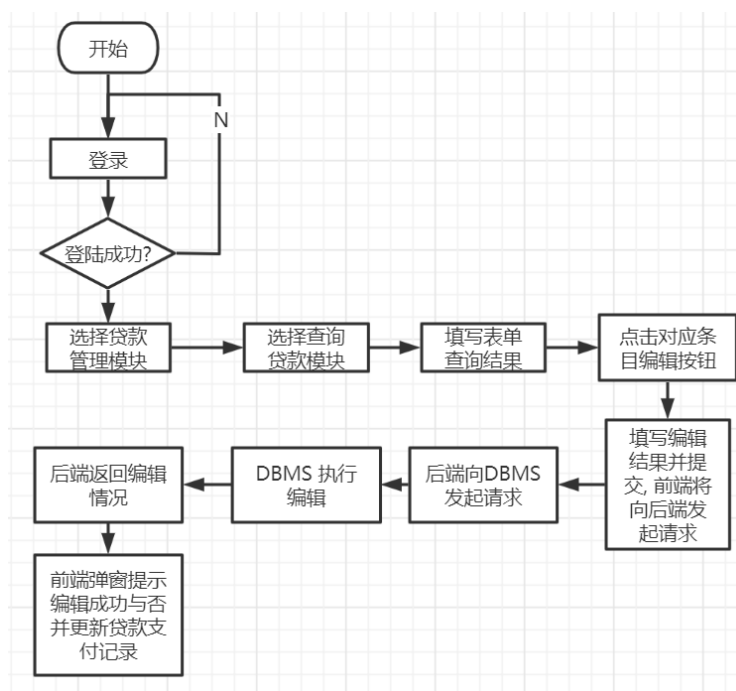
前端需要向后端发起修改客户请求, 并传递所有新信息以供后端修改.

后端收到修改参数后, 向 DBMS 发起请求即可修改对应条目, 并向前端返回修改情况.

如果修改失败, 可能抛出以下异常:

1. 管理员登录过期异常
2. 输入不符合格式异常

流程图如下:



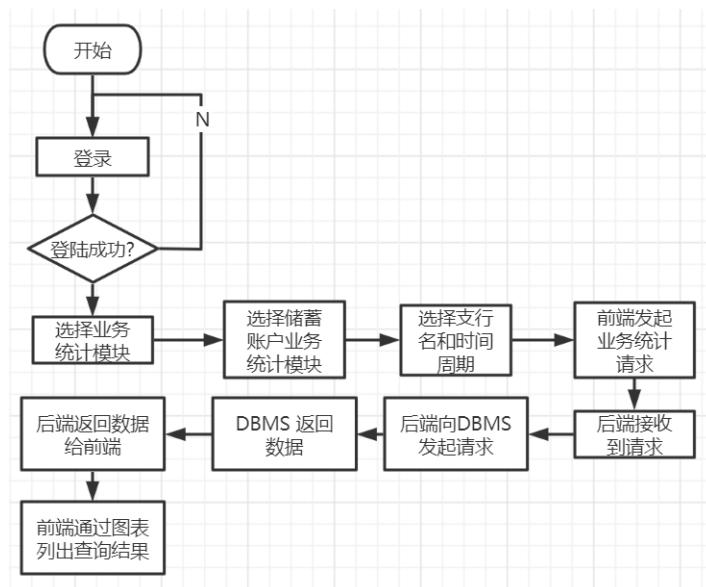
3.4 业务统计模块

3.4.1 储蓄账户业务统计模块

该模块前端提供一个刷新按钮. 每当点击刷新按钮或重新进入这个页面, 前端都会向后端发起请求统计业务, 后端将业务统计完成后就返回给前端, 前端将通过支行选择条件和时间周期条件来渲染数据, 同时以表格和折线图或条形图的形式呈现出来. (可以说是非常酷炫了)

该模块只需要输入支行名和时间周期，即可输出详细的业务统计信息，包括但不限于储蓄量和客户数。

流程图如下：

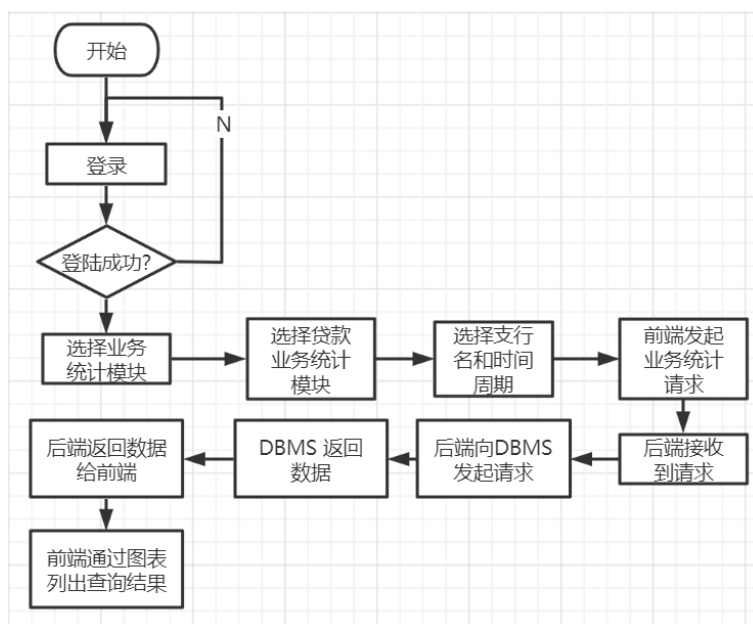


3.4.2 贷款业务统计模块

该模块前端提供一个刷新按钮。每当点击刷新按钮或重新进入这个页面，前端都会向后端发起请求统计业务，后端将业务统计完成后就返回给前端，前端将通过支行选择条件和时间周期条件来渲染数据，同时以表格和折线图或条形图的形式呈现出来。(可以说是非常酷炫了)

该模块只需要输入支行名和时间周期，即可输出详细的业务统计信息，包括但不限于贷款量和客户数。

流程图如下：



3.5 登录管理模块

3.5.1 登录及登录状态验证

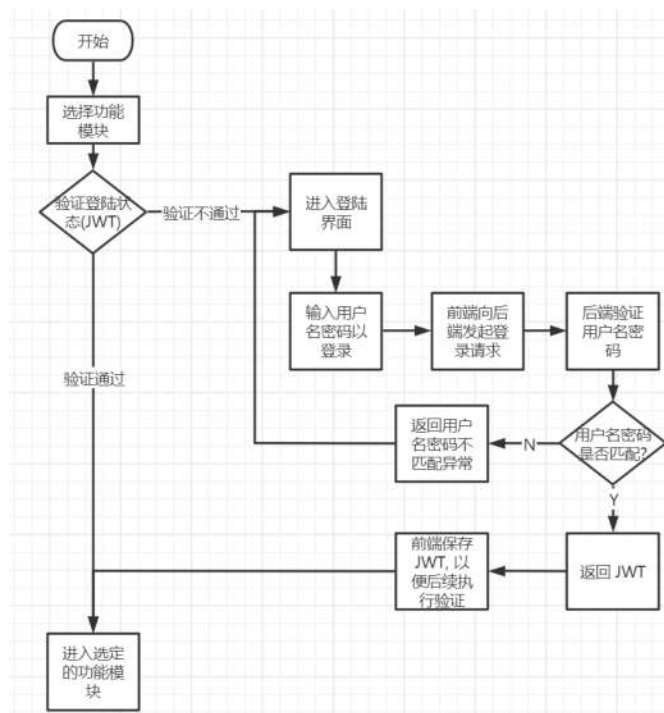
该模块要求用户输入用户名和密码，并点击登录。此后，前端会发送用户名和密码到后端去验证，后端若验证通过，则返回一个 JWT (JSON Web Tokens)。前端接收到 JWT 并保存起来，以后每次请求数据都要将 JWT 同时发出，以便后端验证。

后端每次接收到请求，都要验证 JWT 是否正确(即是否由给定密钥签名而成)，若不正确，则要求用户重新登录，若正确，则正常返回数据。

整个过程可能抛出的异常有：

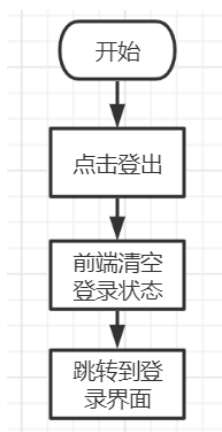
1. 管理员登录过期异常
2. 用户名密码不匹配异常
3. JWT 过期异常
4. JWT 验证失败异常

流程图如下：



3.5.2 登出

前端点击登出按钮, 即可清空 token 信息, 并跳转到登录界面.



3.7 用户输入校验模块

对于用户的输入, 后端应予以判断, 以确认是否满足要求. 例如, 客户身份证号位数是否正确, 账号号位数是否正确, 余额是否大于等于零, 账户类型是字符串表示还是整型数字表示等. 虽然一切不妥的修改可以通过日志恢复, 但及早避免错误的产生将更有利于系统的运行.

为了处理这个问题, 所有从前端传到后端的数据首先都会经过一层预处理函数, 这个函数将遍历输入数据, 并根据键(key)来对值(value)进行预处理, 这包括:

1. 客户身份证号:

-
- a) 应为 18 位
 - b) 应为数字
 2. 客户姓名
 - a) 不超过 16 个字符
 3. 客户电话
 - a) 不超过 16 个字符
 - b) 数字组成
 4. 客户住址
 - a) 不超过 32 个字符
 5. 客户联系人姓名
 - a) 不超过 16 个字符
 6. 客户联系人电话
 - a) 不超过 16 个字符
 - b) 应为数字
 7. 客户联系人邮箱
 - a) 不超过 32 个字符
 - b) 满足邮箱格式
 8. 客户和联系人的关系
 - a) 不超过 16 个字符
 9. 账户号
 - a) 恰为 16 个字符
 - b) 数字组成
 10. 账户类型
 - a) 若是字符串, 则为 STORE 或 CHECK
 - b) 若是数字, 则为 0 或 1
 11. 账户余额
 - a) 浮点数
 - b) 至少是 0
 12. 支票账户透支额
 - a) 浮点数
 - b) 至少是 0
 13. 储蓄账户利率
 - a) 浮点数
 - b) 至少是 0
 14. 储蓄账户货币类型
 - a) CNY 或 USD
 15. 贷款额
 - a) 浮点数
 - b) 至少是 0
 16. 支行名
 - a) 不超过 16 个字符

3.6 系统运行日志模块

使用 Python 的 logging 模块定制化设计了系统的日志模块. 包括有 debug, info, warn, error 等级别的日志信息, 并且同时在终端及文件输出. 有了日志模块才使得银行系统的一切操作得以被监控.

本日志配置文件模块自动读取 etc 目录下的 logger.yaml 配置文件, 其格式类似下面这样:

```
1 bankdb:
2   level: info
3   filename: '../log/bankdb.log'
4   fmt: '%(asctime)s [bankdb] %(filename)s [line:%(lineno)d]: %(levelname)s %(message)s'
5 web:
6   level: info
7   filename: '../log/web.log'
8   fmt: '%(asctime)s [web] %(filename)s [line:%(lineno)d]: %(levelname)s %(message)s'
```

通过这样的简单配置, 可以使得整个系统使用的 logger 比较统一.

4 实现与测试

4.1 实现结果

4.1.1 添加客户

添加客户的界面如下:

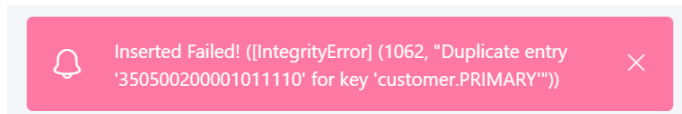
The screenshot shows a web application interface for managing customers. On the left is a sidebar with a teal-to-blue gradient, containing navigation icons and labels: DASHBOARD, CUSTOMER (selected), ACCOUNT, LOAN, Kiosk, and TYPEGRAPH. The main area is titled 'CUSTOMER' and has tabs for 'Insert', 'Search', and 'Edit'. The 'Insert' tab is active, showing a form titled 'Insert Profile'. The form has several input fields: 'Customer Id' (35050020001011110), 'Customer Name' (小小憨憨), 'Customer Phone' (123456), 'Customer Address' (地址), 'Contacts Name' (小小憨憨), 'Contacts Phone' (181111), 'Contacts Email' (888@hanhan.com), and a 'Relation between Customer and Contacts' dropdown (情侣). A green 'Save' button is at the bottom of the form. On the right is a customer profile card for '小小憨憨' (Customer), showing their ID, phone, address, and contact information. The footer of the page includes 'HAN-HAN BANK · ABOUT ME' and a copyright notice: '© 2021 made with by Vink.Z.H.King'.

非常直观地可以看出, 用户只需要填写表单信息, 然后点击 Save 按钮即可完成添加客户的操作. 同时, 屏幕右侧还会有客户名片来友好地展示客户信息.

如果成功添加客户, 则屏幕上方会弹出添加成功的消息,

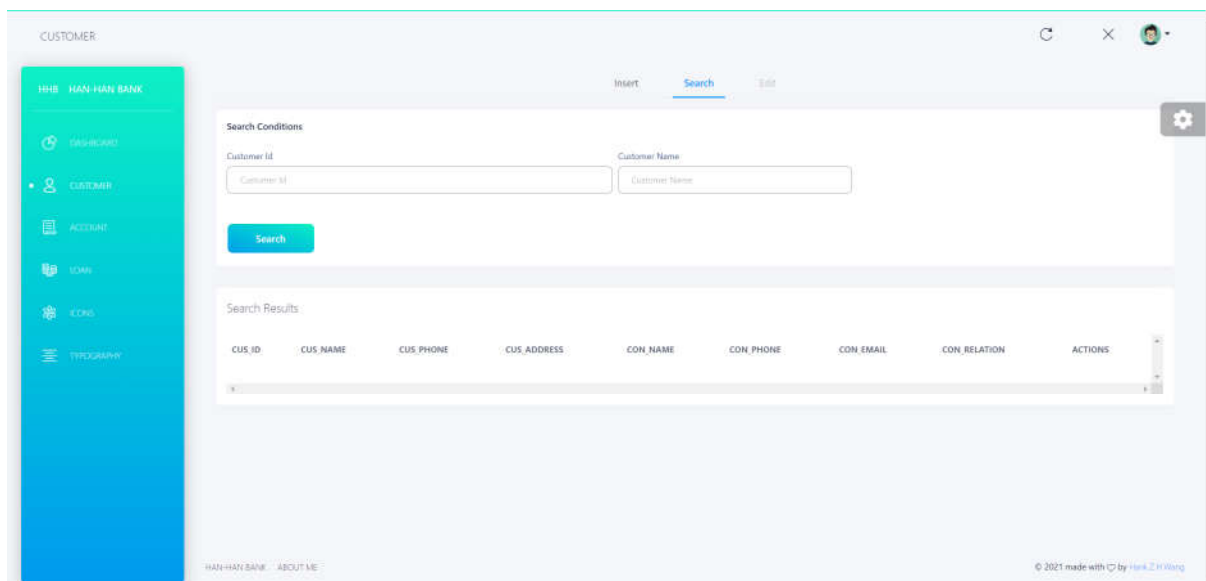


否则会提示异常的原因.

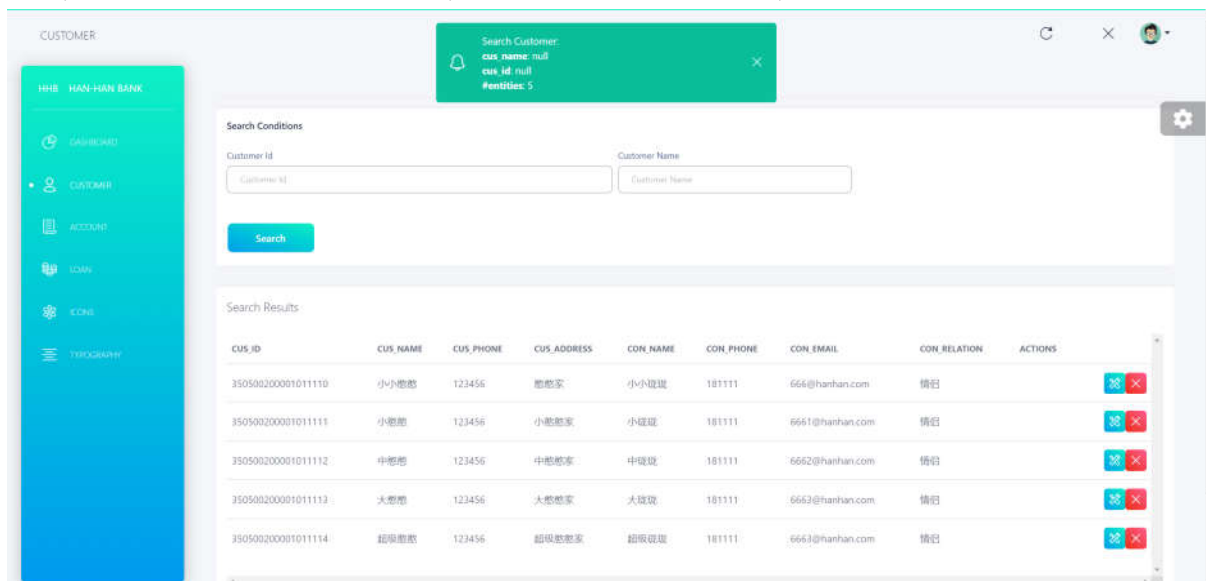


4.1.2 查询客户

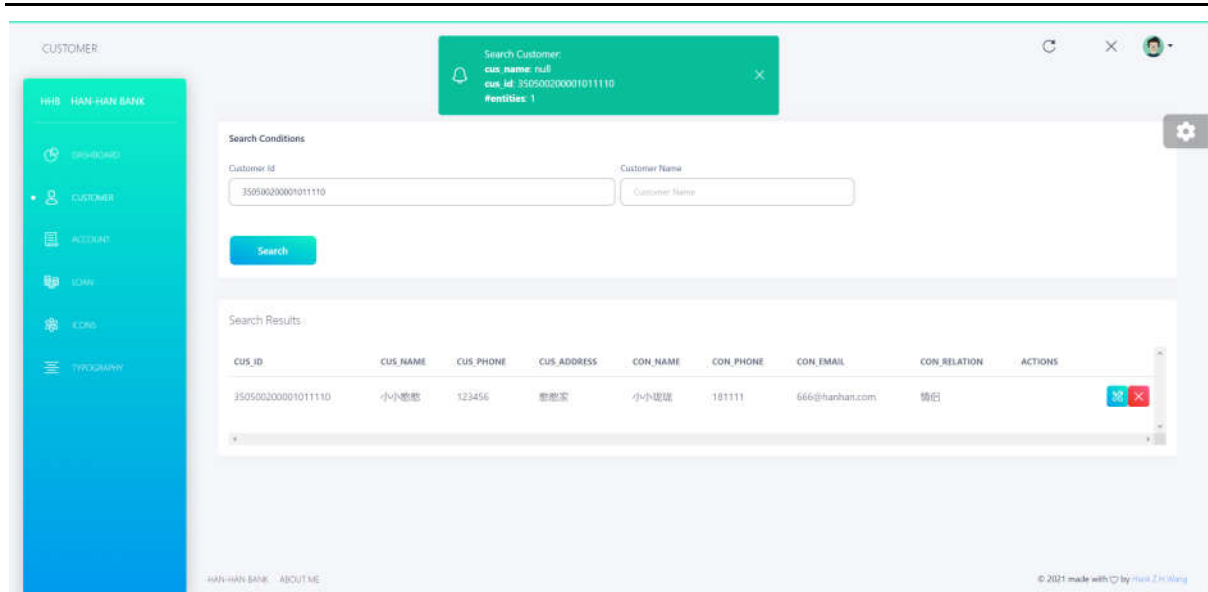
查询客户的界面如下:



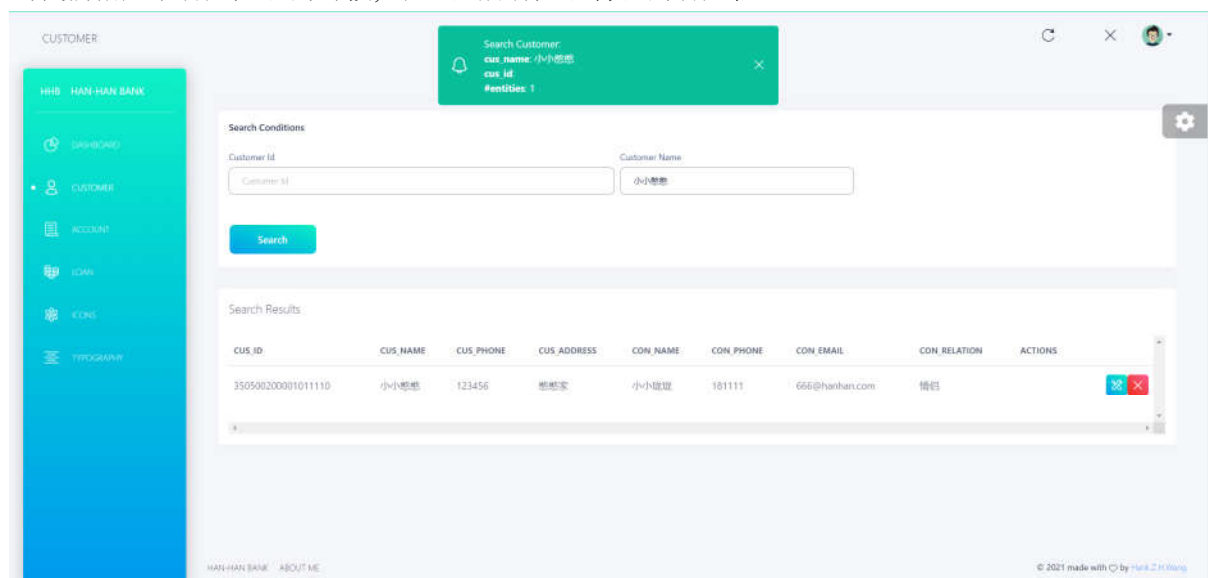
当用户需要搜索的时候, 只需要输入相应信息即可. 这里表单信息取 与 的关系, 然后执行查询. 例如, 当我们不指定任何信息的时候, 就会查出所有客户的信息, 并列之于表中:



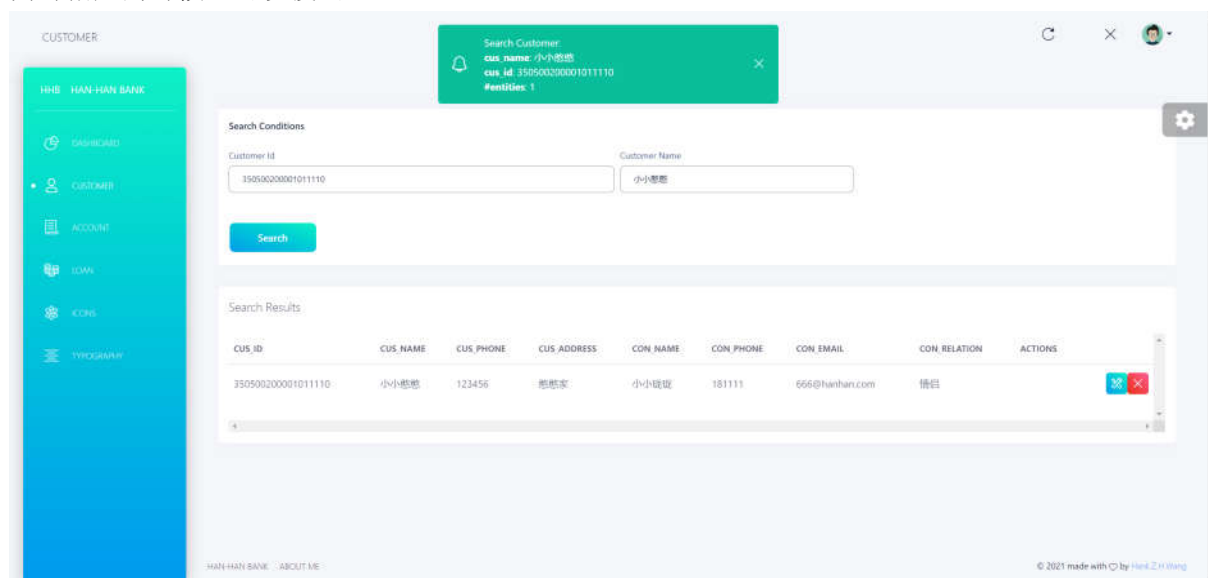
当我们指定了客户身份证号时, 则返回该客户信息:



当我们指定了客户姓名的时候，就返回所有姓名符合的客户信息：



同时指定的时候也可以搜出：



4.1.3 修改客户

如果要修改客户信息，则点击客户条目后面的工具按钮即可跳转到修改选项卡：

The screenshot shows the 'Edit Profile' form for a customer. The form is titled 'Edit Profile' and has a 'Save' button at the bottom. The form contains the following fields:

- Customer ID: 350500200001011110
- Customer Name: 小小憨憨
- Customer Phone: 123456
- Customer Address: 憨憨家
- Contacts Name: 小小萌萌
- Contacts Phone: 181111
- Contacts Email: 666@hanhan.com
- Relation between Customer and Contacts: 情侣

我们将客户地址修改后点击保存：

The screenshot shows the 'Edit Profile' form after the customer address has been updated. A notification banner at the top says 'Updated Customer: 小小憨憨'. The form contains the following fields:

- Customer ID: 350500200001011110
- Customer Name: 小小憨憨
- Customer Phone: 123456
- Customer Address: 小小憨憨家
- Contacts Name: 小小萌萌
- Contacts Phone: 181111
- Contacts Email: 666@hanhan.com
- Relation between Customer and Contacts: 情侣

重新搜索就会发现客户信息已经被修改：

350500200001011110	小小憨憨	123456	小小憨憨家	小小萌萌	181111	666@hanhan.com	情侣	
--------------------	------	--------	-------	------	--------	----------------	----	--

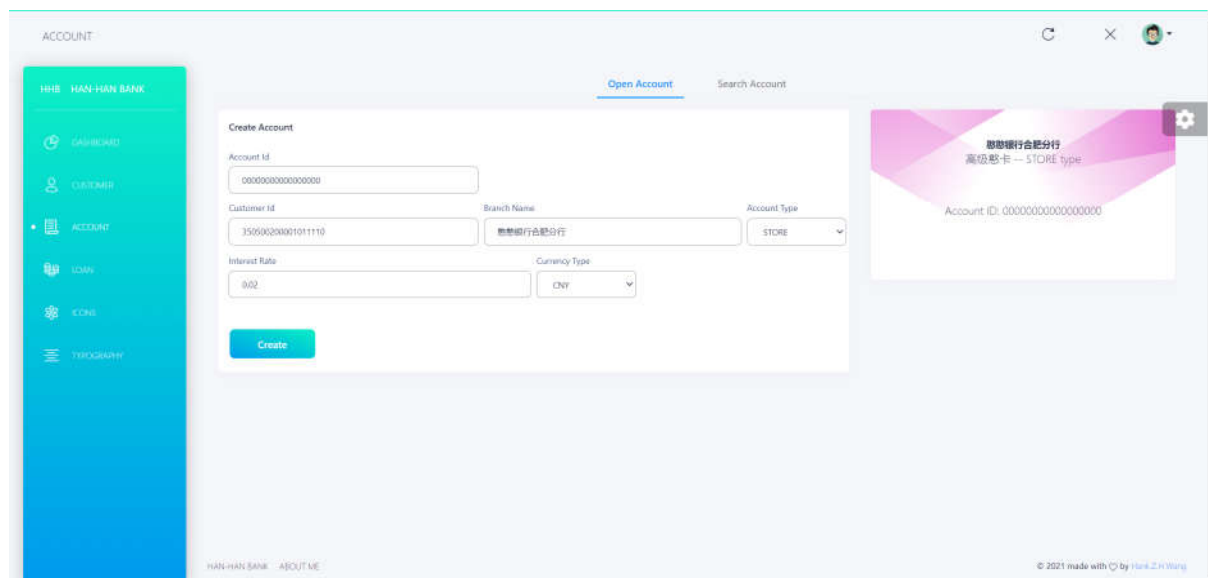
4.1.4 删除客户

点击客户条目最后面的删除按钮即可删除客户。

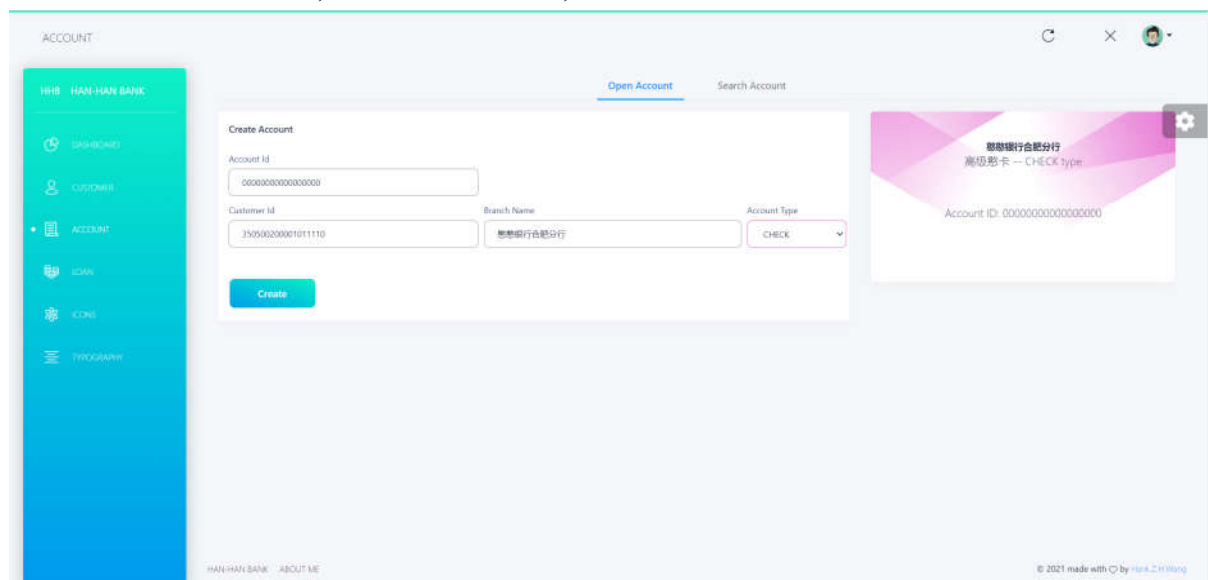


4.1.5 开户

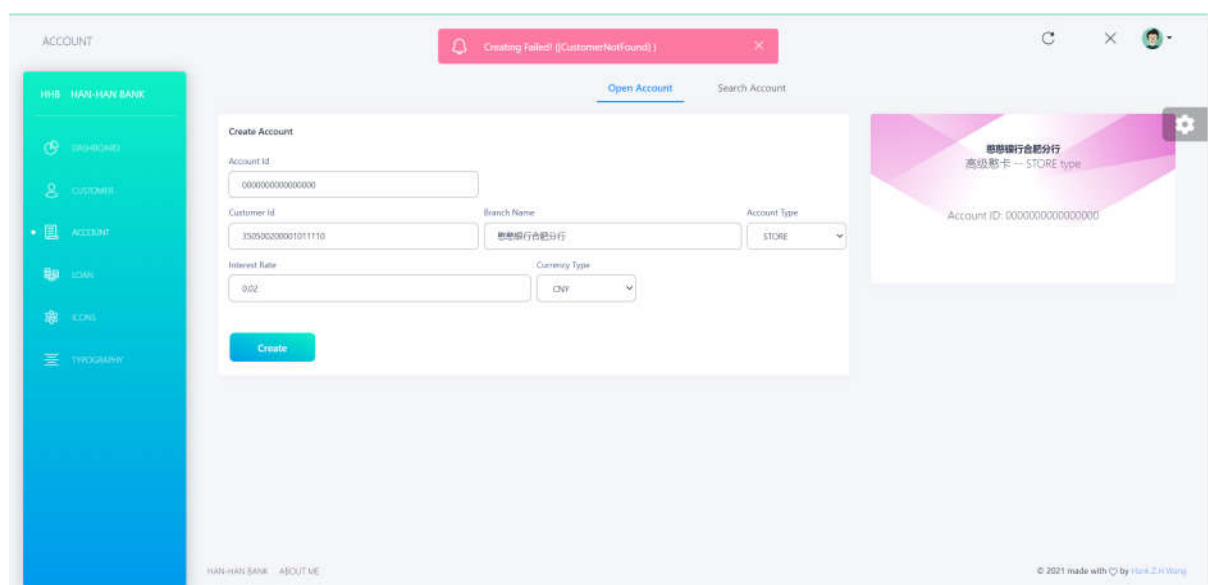
开户界面如下:



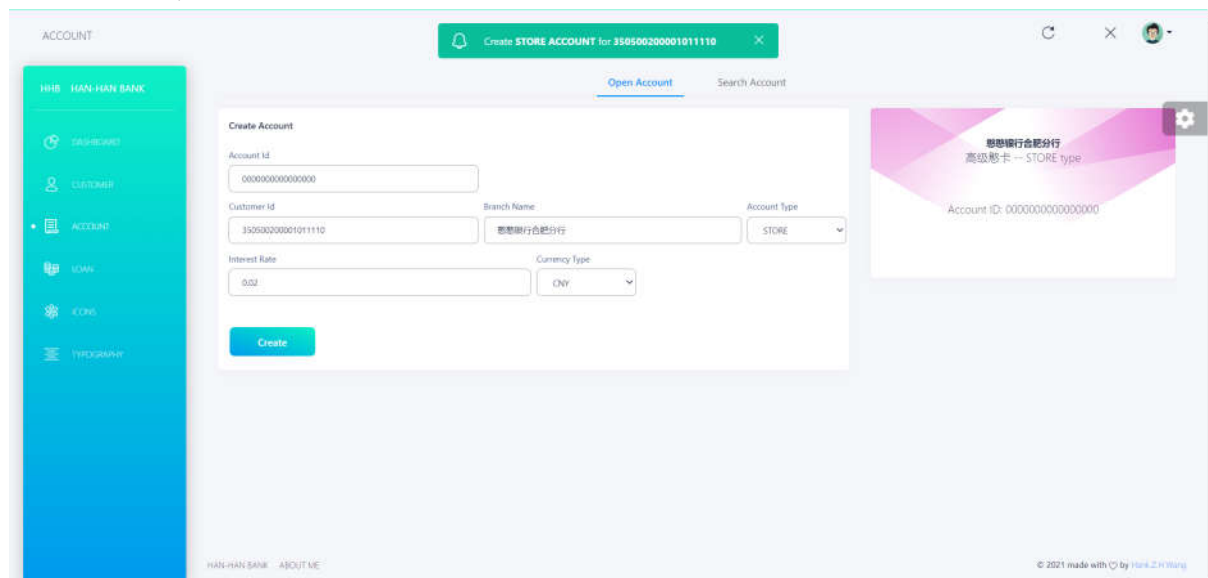
显然, 如果要开户, 只需要输入相应的信息即可. 屏幕右侧会显示将产生的电子银行卡的预览图片. 上图是开储蓄账户的界面, 如果要开支票账户, 则是如下表单:



点击 create 后可以开户, 但若客户不存在, 则会报错:



如果一切顺利，将提示开户成功：



4.1.6 搜索账户

要搜索账户，只要选中对应选项卡，并填写表单，点击搜索按钮即可：

ACCOUNT

Open Account Search Account

Search Conditions

Account ID:

Customer ID:

Branch Name:

Search

Search Results(Store)

ACC_ID	BRA_NAME	CUS_ID	ACC_TYPE	ACC_BALANCE	STO_INTEREST_RATE	STO_CURRENCY_TYPE	STO_LAST_VISIT_DATE	ACTIONS

Search Results(Check)

ACC_ID	BRA_NAME	CUS_ID	ACC_TYPE	ACC_BALANCE	CHE_OVERDRAFT	CHE_LAST_VISIT_DATE	ACTIONS

如果什么都不填写, 默认返回所有账户信息:

ACCOUNT

Search Succeeded!

Search

Search Results(Store)

ACC_ID	BRA_NAME	CUS_ID	ACC_TYPE	ACC_BALANCE	STO_INTEREST_RATE	STO_CURRENCY_TYPE	STO_LAST_VISIT_DATE	ACTIONS
000000000000000000	慧慧银行合肥分行	350500200001011110	STORE	0.00	0.02	CNY	Fri, 18 Jun 2021 13:56:48 GMT	
000000000000000003	慧慧银行合肥分行	350500200001011113	STORE	6,666.00	0.06	CNY	Tue, 26 Jan 2021 09:00:00 GMT	
000000000000000004	慧慧银行惠州分行	350500200001011114	STORE	666.00	0.06	CNY	Thu, 15 Apr 2021 09:00:00 GMT	


Search Results(Check)

ACC_ID	BRA_NAME	CUS_ID	ACC_TYPE	ACC_BALANCE	CHE_OVERDRAFT	CHE_LAST_VISIT_DATE	ACTIONS
000000000000000040	慧慧银行惠州分行	350500200001011114	CHECK	77.00	0.00	Thu, 15 Apr 2021 09:00:00 GMT	

否则只返回对应账户信息:

The screenshot displays a web application for managing accounts. On the left is a sidebar with a teal-to-blue gradient, containing a menu with options: HOME, LOGIN/LOGOUT, CUSTOMER, ACCOUNT (selected), CASH, EXCH, and TRANSFERS. The main area is titled 'ACCOUNT' and features a 'Search Conditions' section at the top with a green 'Search Succeeded!' notification. Below this are input fields for 'Account ID' (with a hint 'Account ID(16 digits)'), 'Customer Id' (containing '350500200001011114'), and 'Branch Name'. A blue 'Search' button is positioned below these fields. The search results are divided into two sections: 'Search Results(Store)' and 'Search Results(Check)'. Each section contains a table with columns: ACC_ID, BRN_NAME, CUS_ID, ACC_TYPE, ACC_BALANCE, STO_INTEREST_RATE, STO_CURRENCY_TYPE, STO_LAST_VISIT_DATE, and ACTIONS. The 'Store' table has one row with ACC_ID '000000000000000004', BRN_NAME '惠康银行星州分行', CUS_ID '350500200001011114', ACC_TYPE 'STORE', ACC_BALANCE '666.00', STO_INTEREST_RATE '0.06', STO_CURRENCY_TYPE 'CNY', and STO_LAST_VISIT_DATE 'Thu, 15 Apr 2021 09:00:00 GMT'. The 'Check' table has one row with ACC_ID '0000000000000000040', BRN_NAME '惠康银行星州分行', CUS_ID '350500200001011114', ACC_TYPE 'CHECK', ACC_BALANCE '77.00', CHE_OVERDRAFT '0.00', and CHE_LAST_VISIT_DATE 'Thu, 15 Apr 2021 09:00:00 GMT'. Each row in the tables has an 'ACTIONS' column with edit and delete icons.

4.1.7 修改账户

要想修改账户，只需要点击账户条目右侧的  按钮。例如，我们想要修改账户号为 000000000000000004 的储蓄账户，点击对应条目的修改按钮后，将会弹出模态框：

The 'Update account' modal form is displayed over the search results. It contains the following fields and controls: 'Account ID' (a text box with '000000000000000004'), 'Type' (a text box with 'STORE'), 'Account Balance' (a text box with '666'), 'Interest Rate' (a text box with '0.06'), and 'Account Type' (a dropdown menu with 'CNY' selected). At the bottom are two buttons: a purple 'Save' button and a dark blue 'cancel' button.

模态框内的账户号和账户类型是写定的，不可修改的。可以修改的是账户余额，利率和货币类型。例如，我们想要修改账户余额为 6666，就直接修改并且点击保存即可。

Update account

Account ID

0000000000000004

Type

STORE

Account Balance

6666

Interest Rate

0.06

Account Type

CNY

Save

cancel

可以看到, 账户表中的信息也将会自动更新:

ACCOUNT

HAN HAN BANK

Dashboard

Customer

Account

Loan

Loan

Transaction

Search Conditions

Update Succeed!

Search Succeed!

Account ID

Branch Name

Search

Search Results(Store)

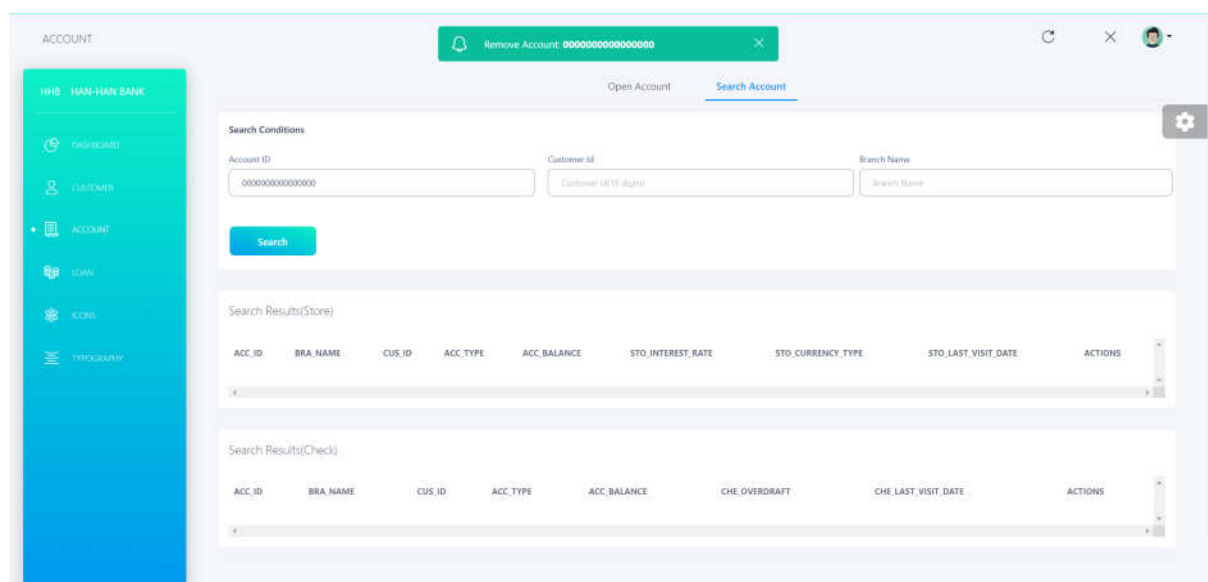
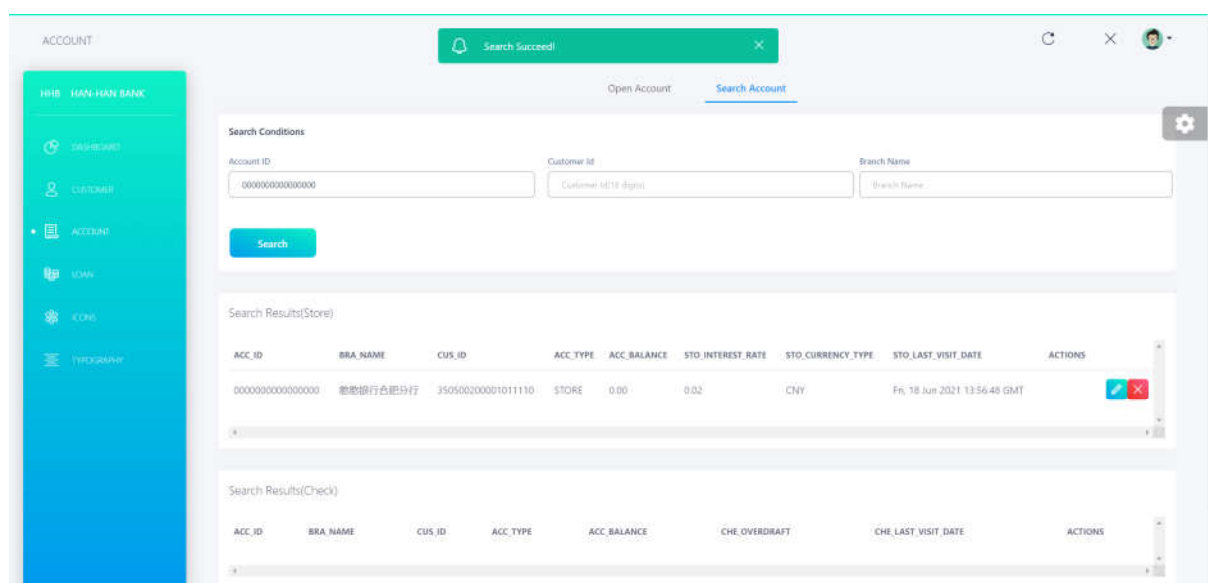
ACC_ID	BRA_NAME	CUS_ID	ACC_TYPE	ACC_BALANCE	STO_INTEREST_RATE	STO_CURRENCY_TYPE	STO_LAST_VISIT_DATE	ACTIONS
0000000000000004	德惠银行泉州分行	350500200001011114	STORE	6,666.00	0.06	CNY	Fri, 18 Jun 2021 14:01:01 GMT	<div></div>

Search Results(Check)

ACC_ID	BRA_NAME	CUS_ID	ACC_TYPE	ACC_BALANCE	CHE_OVERDRAFT	CHE_LAST_VISIT_DATE	ACTIONS
0000000000000004	德惠银行泉州分行	350500200001011114	CHECK	77.00	0.00	Thu, 15 Apr 2021 09:00:00 GMT	<div></div>

4.1.8 删除账户

如果要删除账户, 也只需要点击相应的删除按钮即可, 例如我们可以删除刚刚添加的账户 0000000000000000:



这样就完成了账户删除。

4.1.9 新增贷款

新增贷款的界面如下：

LOAN

Insert Loan Search Loan Pay Loan

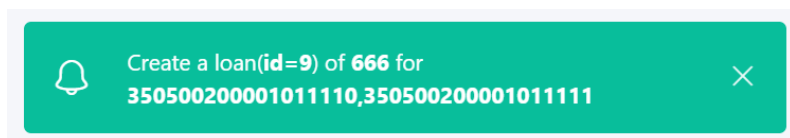
Branch Name: 慧恩银行合肥分行

Loan Amount: 666 Formatted Loan Amount: 666.00

Customer ID: 350500200001011110, 350500200001011111

Create

显然，只要输入支行名，贷款额和客户身份证号并点击 **Create** 按钮即可添加贷款。添加成功后，会给出如下提示：



4.1.10 查询贷款

查询贷款的界面如下：

LOAN

Insert Loan Search Loan Pay Loan

Search Conditions

Branch Name: Branch Name Loan ID: Loan ID

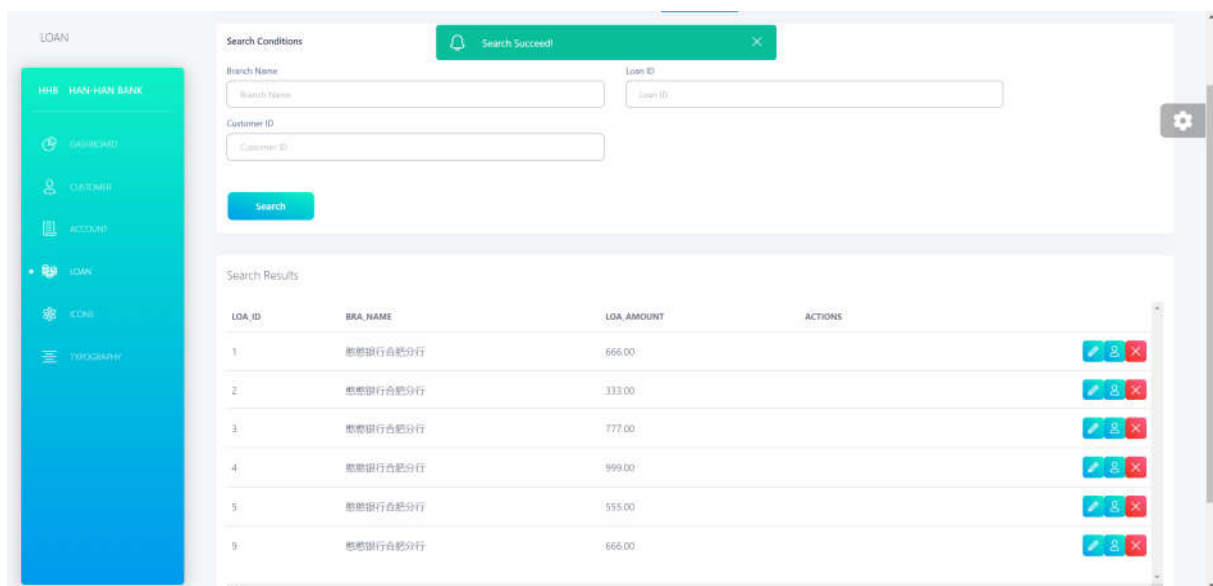
Customer ID: Customer ID

Search

Search Results

LOA_ID	BRA_NAME	LOA_AMOUNT	ACTIONS
1			

为了查询贷款，可以输入支行名，贷款号及客户号的组合条件。如均为空，默认返回所有贷款信息：



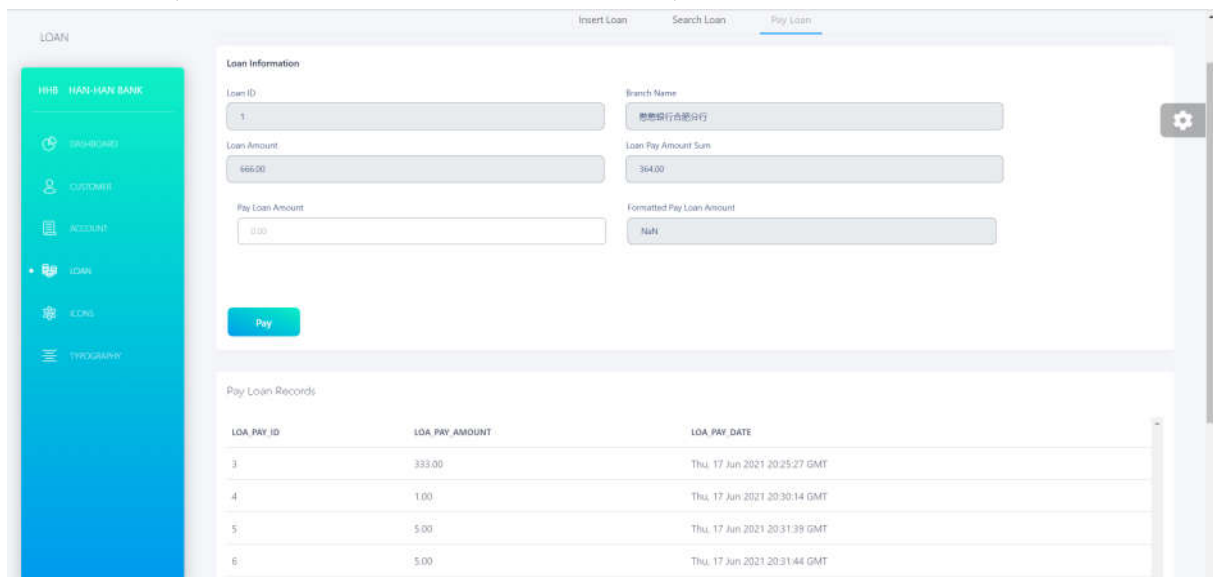
4.1.11 查询贷款者

如果要进一步查询贷款所有者，可以点击右侧的人物按钮，此时则会弹出贷款所有人的身份证号：



4.1.12 支付贷款

若要支付贷款，同样点击相应贷款条目右侧的铅笔按钮，即可跳转到支付贷款的选项卡：



该选项卡展示了贷款的支付记录及情况. 且会显示贷款已支付总额.
若要支付贷款, 只需要输入支付额, 并且点击 **Pay** 按钮即可.

The screenshot shows a 'Loan Information' form with the following fields and values:

Field	Value
Loan ID	1
Branch Name	慈德银行合肥分行
Loan Amount	666.00
Loan Pay Amount Sum	394.00
Pay Loan Amount	30
Formatted Pay Loan Amount	30.00

A green 'Pay' button is located at the bottom left of the form.

例如上述输入点击了 **Pay** 之后, 就会自动新增条目:

11	30.00	Fri, 18 Jun 2021 14:10:11 GMT
----	-------	-------------------------------

4.1.13 贷款删除

当贷款支付完成, 可以删除贷款:

The screenshot shows the 'Loan Information' form with the following fields and values:

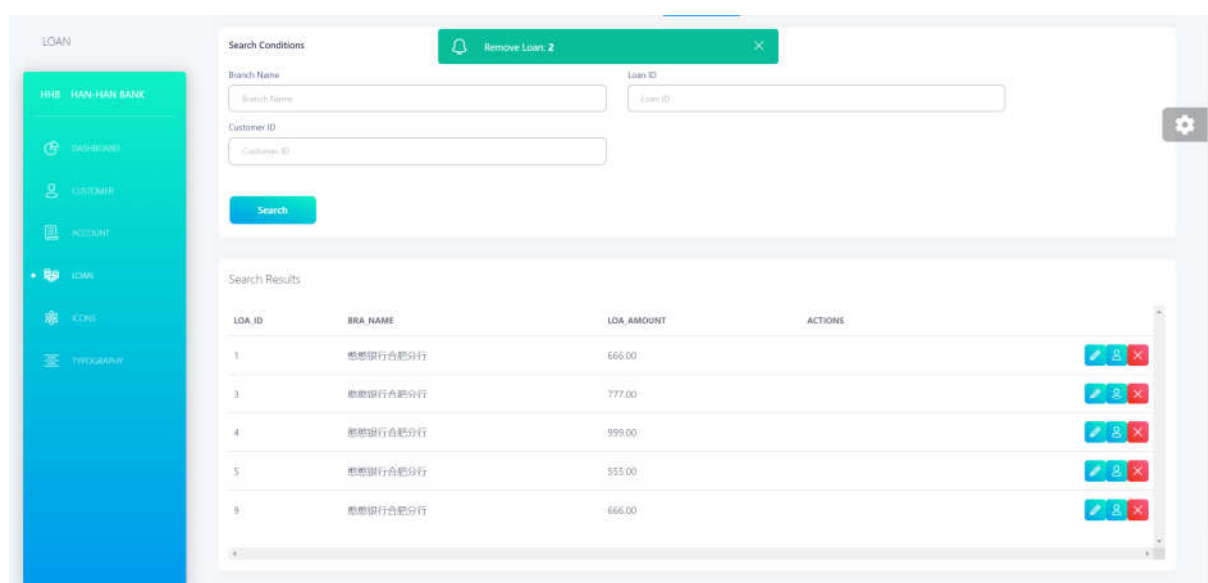
Field	Value
Loan ID	2
Branch Name	慈德银行合肥分行
Loan Amount	333.00
Loan Pay Amount Sum	333.00
Pay Loan Amount	333
Formatted Pay Loan Amount	333.00

A green 'Pay' button is located at the bottom left of the form.

Below the form is a 'Pay Loan Records' table:

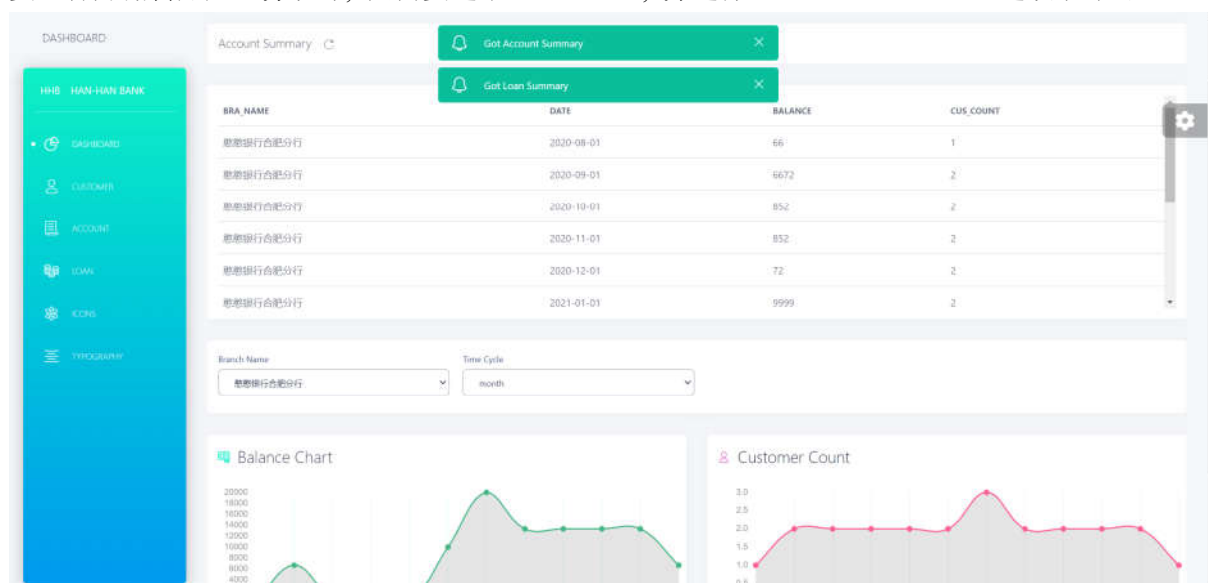
LOA_PAY_ID	LOA_PAY_AMOUNT	LOA_PAY_DATE
12	333.00	Fri, 18 Jun 2021 14:10:49 GMT

进入搜索选项卡点击删除即可:



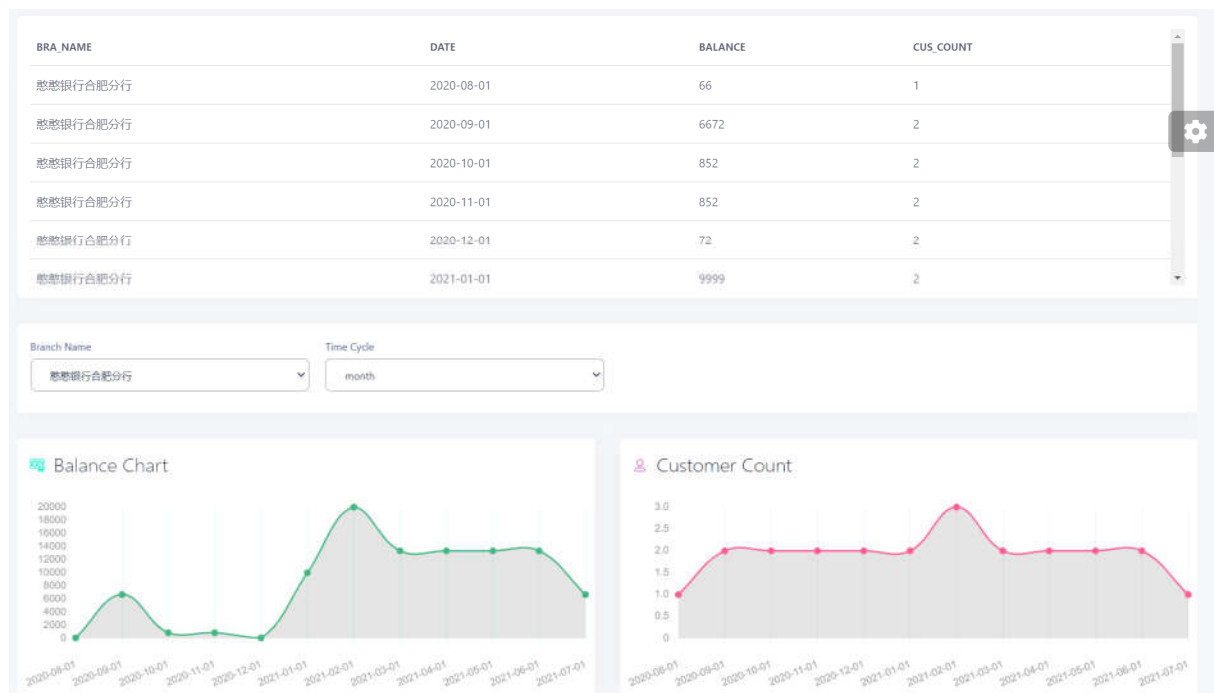
4.1.14 储蓄账户业务统计

要想打开储蓄账户业务统计，只需要选中 Dashboard，并选择 Account Dashboard 选项卡即可：

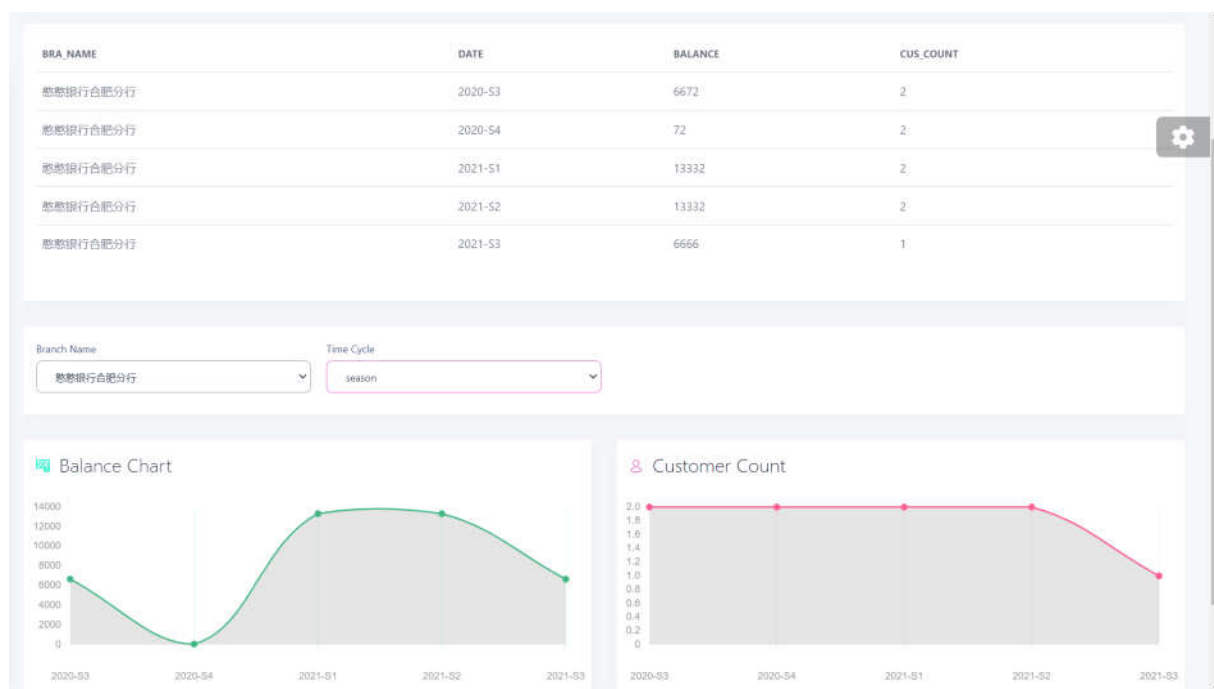


这里可以按支行名和时间周期来显示结果。

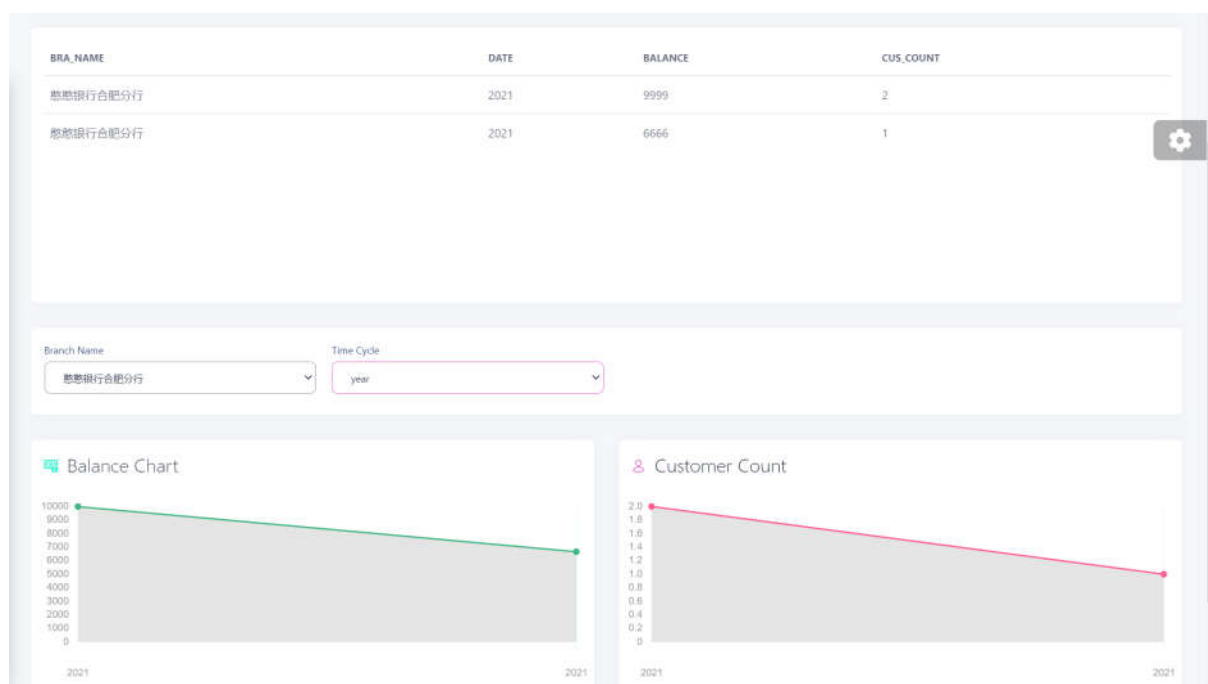
例如按月：



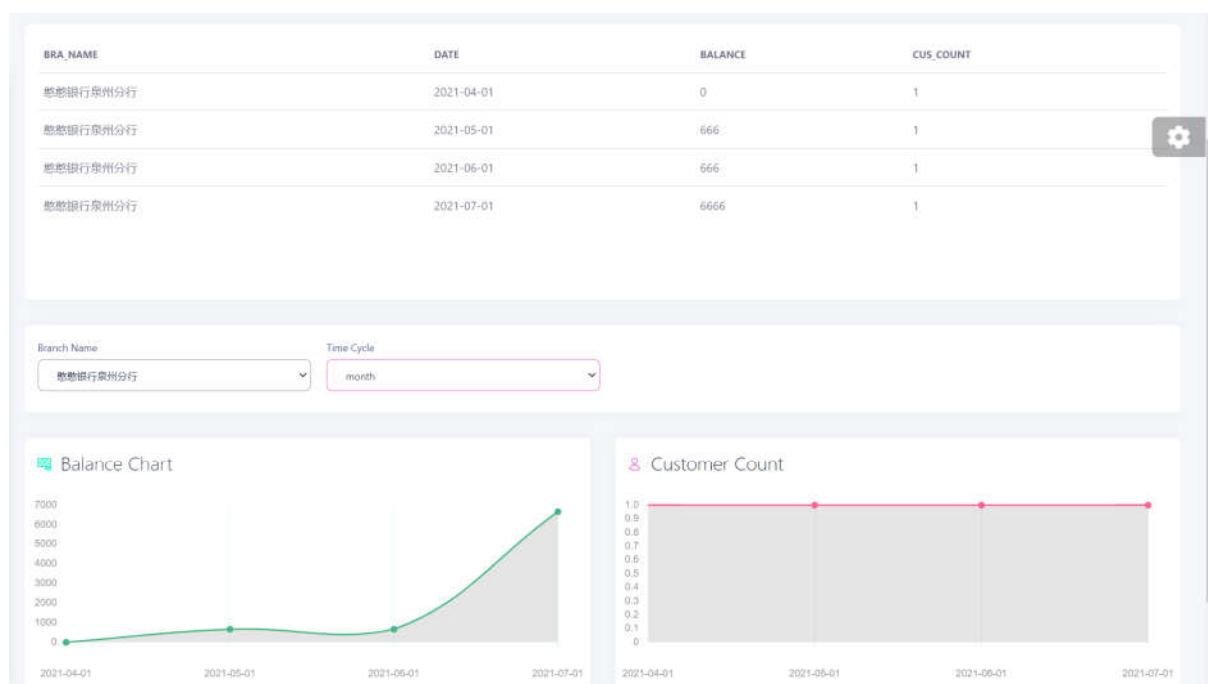
按季:



按年:



同样对于其他支行也是一样的操作:

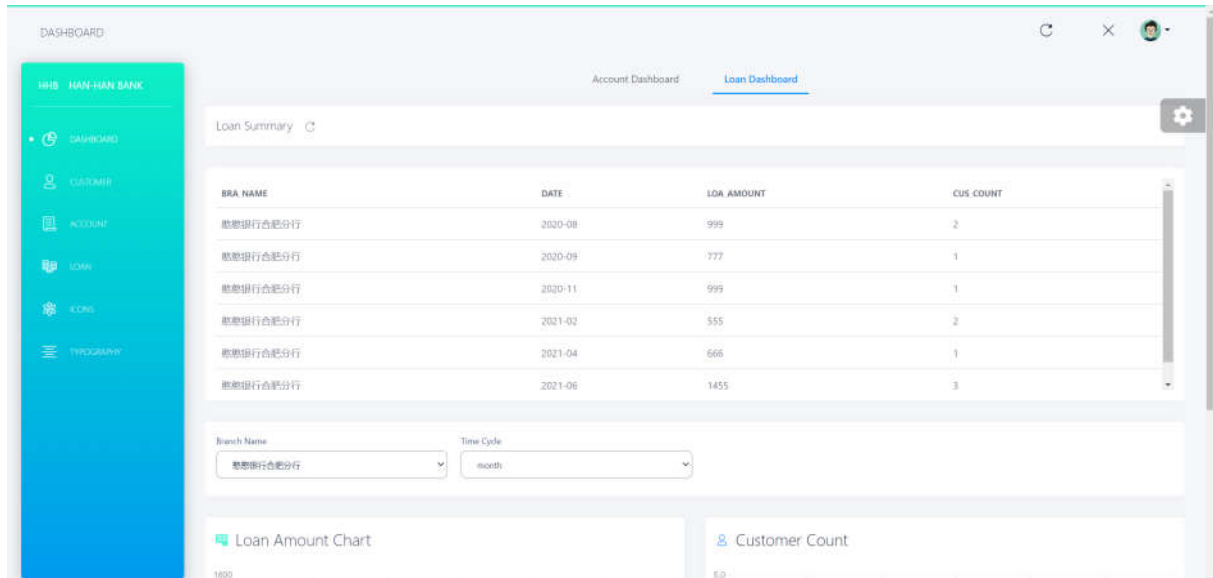


这里面每一个数据点都可以通过鼠标悬停展示详细信息:



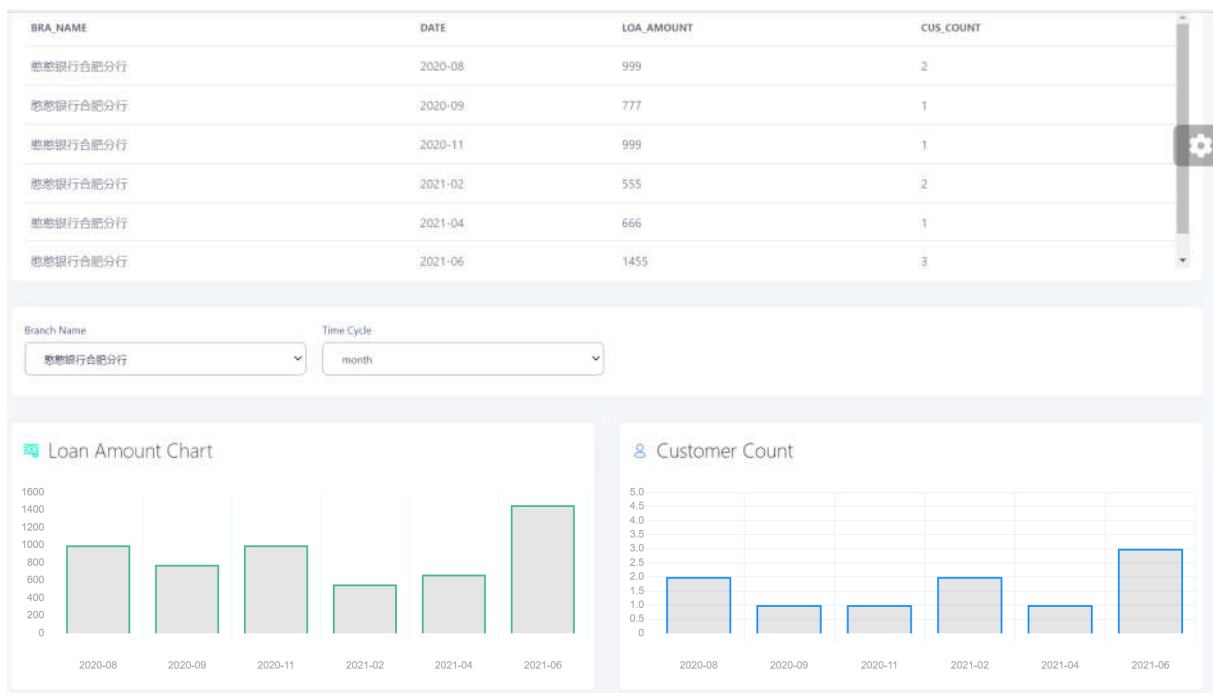
4.1.15 贷款业务统计

要想查看贷款业务统计，同样进入 Dashboard 后，点击 Loan Dashboard 选项卡：

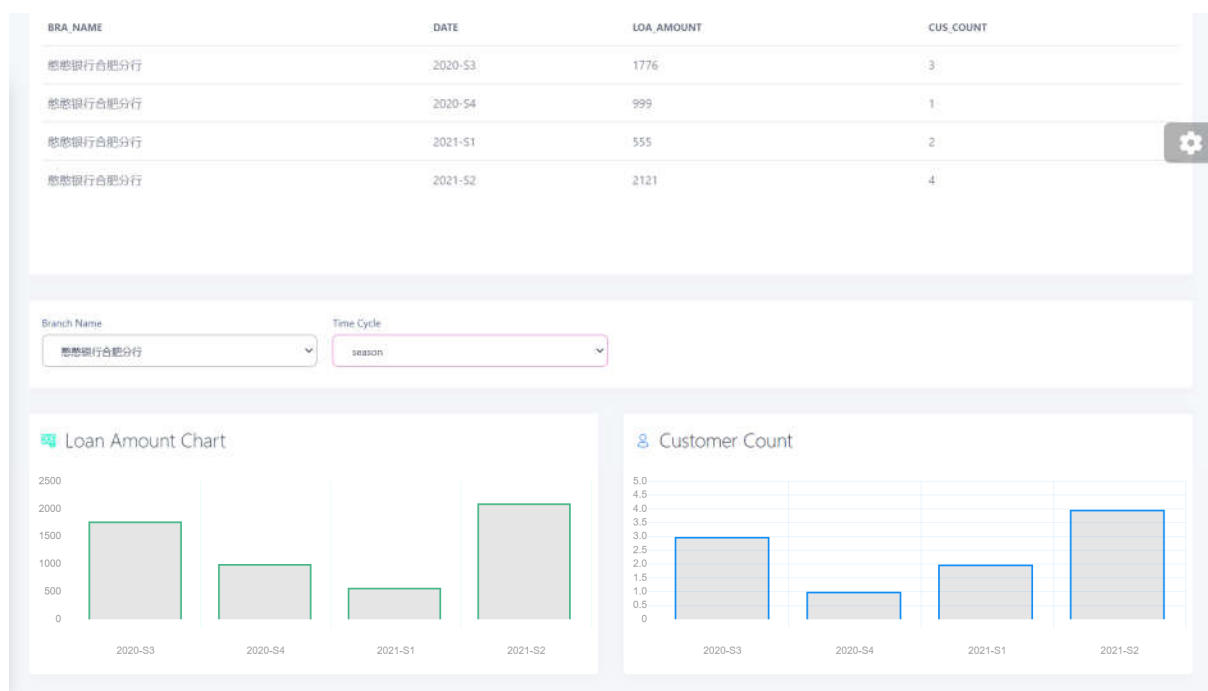


这里同样也列出了详细信息。

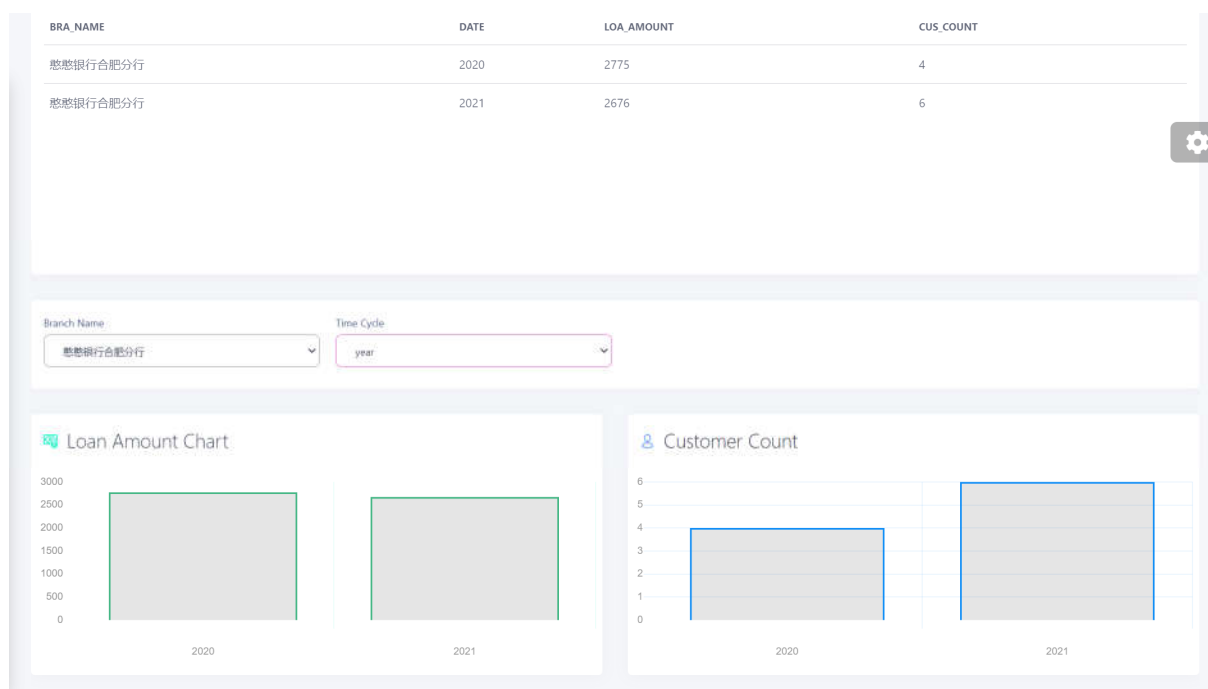
如按月：



按季：

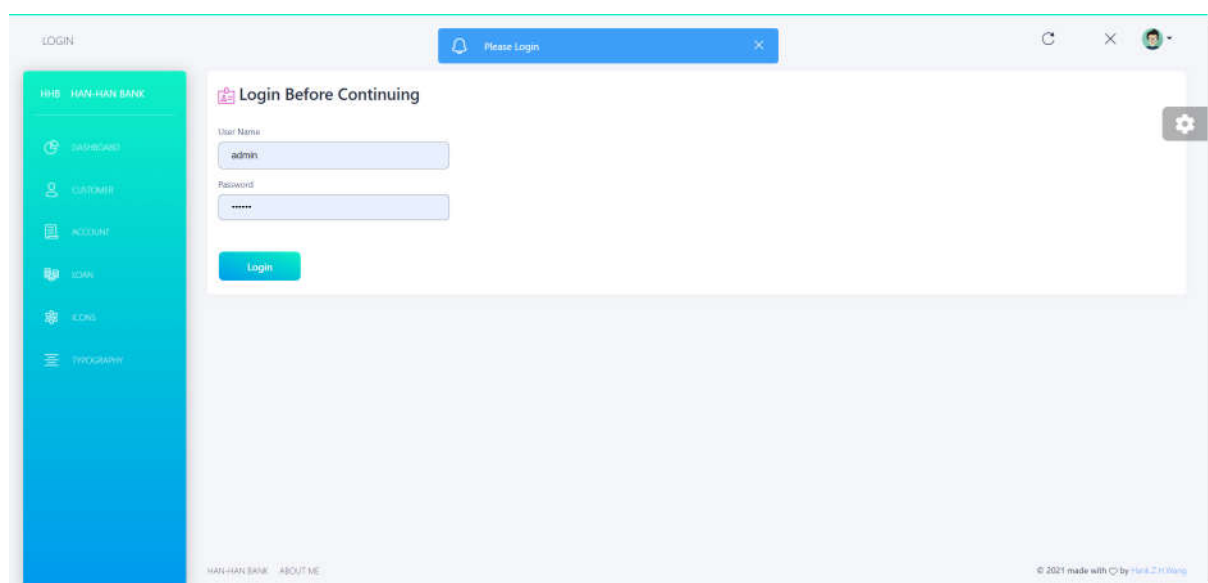


按年:

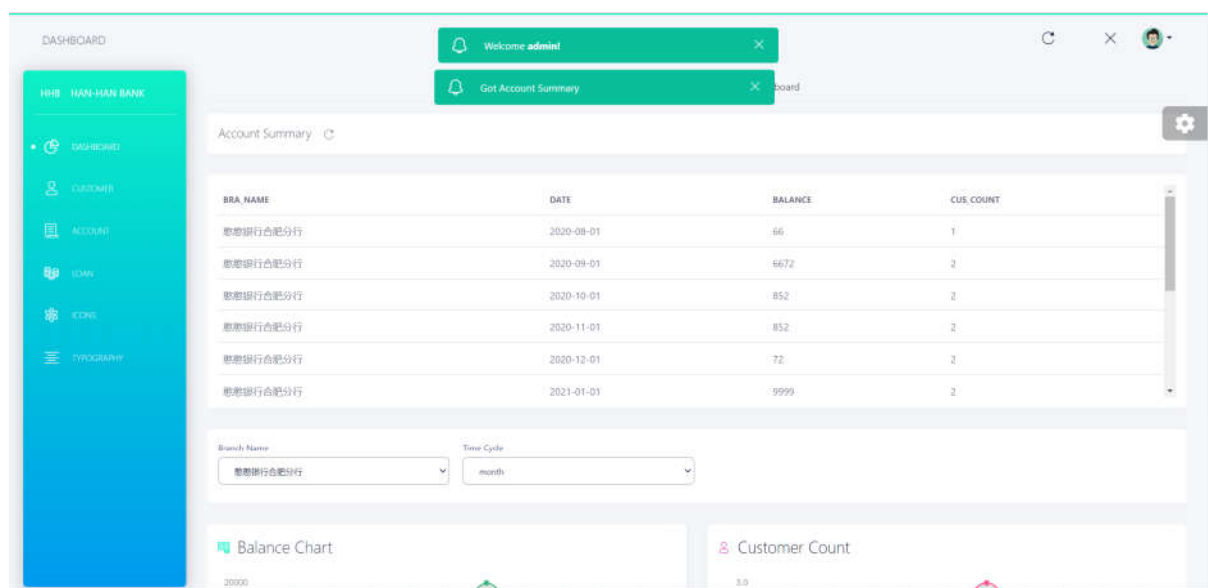


4.1.16 登录与登出

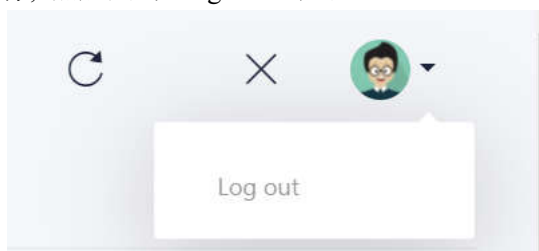
若管理员未登录, 则无法进入各个模块, 因为会被重定向到登录界面:



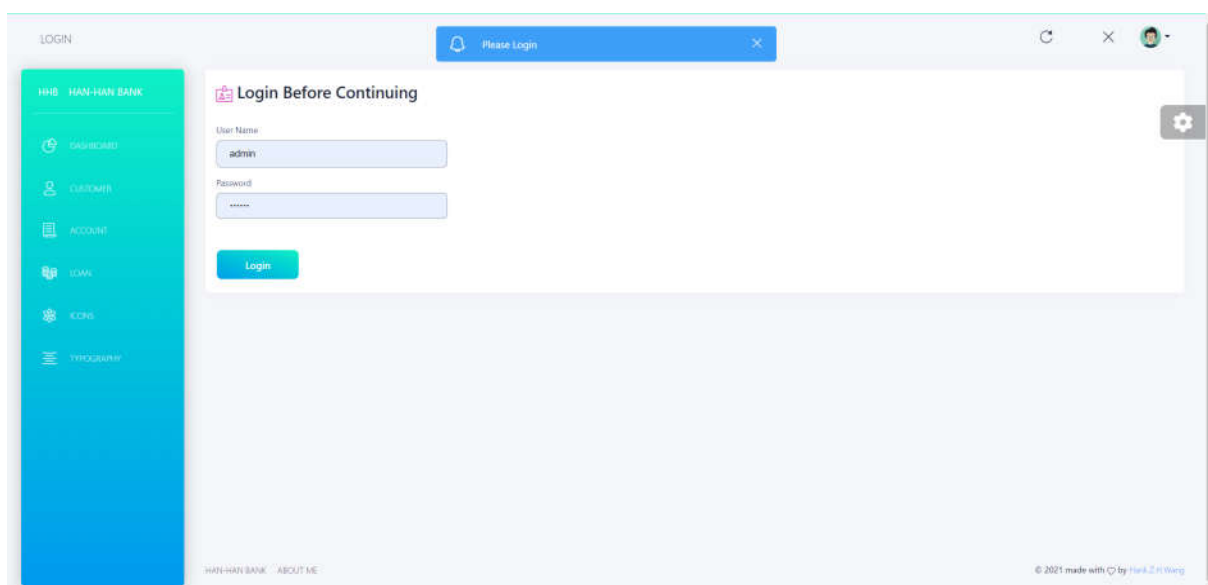
输入用户名密码后点击 **Login** 即可登录, 前端同时会提示欢迎信息:



若要登出, 点击右上角的头像, 然后点击 **Log Out** 即可:

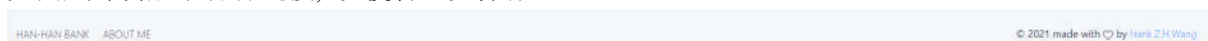


整个界面会重新回到登录界面:

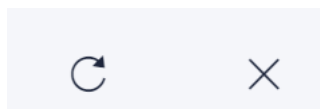


4.1.17 其他琐碎部分

页面底部会展示各种链接，以便管理员操作：



页面右上角有重置和清空数据库的按钮，便于开发者调试：





4.2 测试结果

4.2.1 添加客户

1. 成功的测试样例

Customer Id	Customer Name	Customer Phone
<input type="text" value="350500200001011110"/>	<input type="text" value="小小憨憨"/>	<input type="text" value="123456"/>
Customer Address		
<input type="text" value="憨憨家"/>		
Contacts Name	Contacts Phone	Contacts Email
<input type="text" value="小小珑珑"/>	<input type="text" value="181111"/>	<input type="text" value="666@hanhan.com"/>
Relation between Customer and Contacts		
<input type="text" value="情侣"/>		

 Inserted Customer: 小小憨憨 

2. 失败的测试样例(客户身份证号重复):

Customer Id	Customer Name	Customer Phone
<input type="text" value="350500200001011111"/>	<input type="text" value="小小憨憨"/>	<input type="text" value="123456"/>
Customer Address		
<input type="text" value="憨憨家"/>		
Contacts Name	Contacts Phone	Contacts Email
<input type="text" value="小小珑珑"/>	<input type="text" value="181111"/>	<input type="text" value="666@hanhan.com"/>
Relation between Customer and Contacts		
<input type="text" value="情侣"/>		



Inserted Failed! ([IntegrityError] (1062, "Duplicate entry '350500200001011111' for key 'customer.PRIMARY'"))



4.2.2 查询客户

客户查询的测试样例及结果:

1. 按客户身份证号查询

Search Customer:
cus_name: null
cus_id: 350500200001011110
#entities: 1

Search Conditions

Customer Id

Customer Name

Search

Search Results

CUS_ID	CUS_NAME	CUS_PHONE	CUS_ADDRESS	CON_NAME	CON_PHONE	CON_EMAIL	CON_RELATION	ACTIONS
350500200001011110	小小憨憨	123456	憨憨家	小小珑珑	181111	666@hanhan.com	情侣	

2. 按客户名查询

Search Customer:
cus_name: 小小憨憨
cus_id:
#entities: 1

Search Conditions

Customer Id

Customer Name

Customer Id

小小憨憨

Search

Search Results

CUS_ID	CUS_NAME	CUS_PHONE	CUS_ADDRESS	CON_NAME	CON_PHONE	CON_EMAIL	CON_RELATION	ACTIONS
350500200001011110	小小憨憨	123456	憨憨家	小小璐璐	181111	666@hanhan.com	情侣	<div></div>

3. 按身份证号及客户名查询:

Search Customer:
cus_name: 小小憨憨
cus_id: 350500200001011110
#entities: 1

Search Conditions

Customer Id

Customer Name

350500200001011110

小小憨憨

Search

Search Results

CUS_ID	CUS_NAME	CUS_PHONE	CUS_ADDRESS	CON_NAME	CON_PHONE	CON_EMAIL	CON_RELATION	ACTIONS
350500200001011110	小小憨憨	123456	憨憨家	小小璐璐	181111	666@hanhan.com	情侣	<div></div>

4.2.3 修改客户

修改客户的测试用例及测试结果:

1. 修改客户地址

Edit Profile

Customer Id

Customer Name

Customer Phone

350500200001011110

小小憨憨

123456

Customer Address

小小憨憨家

Contacts Name

Contacts Phone

Contacts Email

小小璐璐

181111

666@hanhan.com

Relation between Customer and Contacts

情侣

Save



2. 修改客户电话号:

Edit Profile

Customer Id: 350500200001011110

Customer Name: 小小憨憨

Customer Phone: 18080808080

Customer Address: 小小憨憨家

Contacts Name: 小小璐璐

Contacts Phone: 181111

Contacts Email: 666@hanhan.com

Relation between Customer and Contacts: 情侣

Save



3. 修改客户联系人邮箱:

Edit Profile

Customer Id: 350500200001011110

Customer Name: 小小憨憨

Customer Phone: 18080808080

Customer Address: 小小憨憨家

Contacts Name: 小小璐璐

Contacts Phone: 181111



Contacts Email: 666666@hanhan.com

Relation between Customer and Contacts: 情侣

Save



以上结果重新搜索后均能反映出来:

CUS_ID	CUS_NAME	CUS_PHONE	CUS_ADDRESS	CON_NAME	CON_PHONE	CON_EMAIL	CON_RELATION	ACTIONS
350500200001011110	小小憨憨	18080808080	小小憨憨家	小小璐璐	181111	666666@hanhan.com	情侣	 

4.2.4 删除客户

删除客户的测试用例和测试结果:

1. 删除仍有账户的客户

350500200001011111 小憨憨 123456 小憨憨家 小璐璐 181111 6661@hanhan.com 情侣

Removing Failed! ([StillHasAccount])

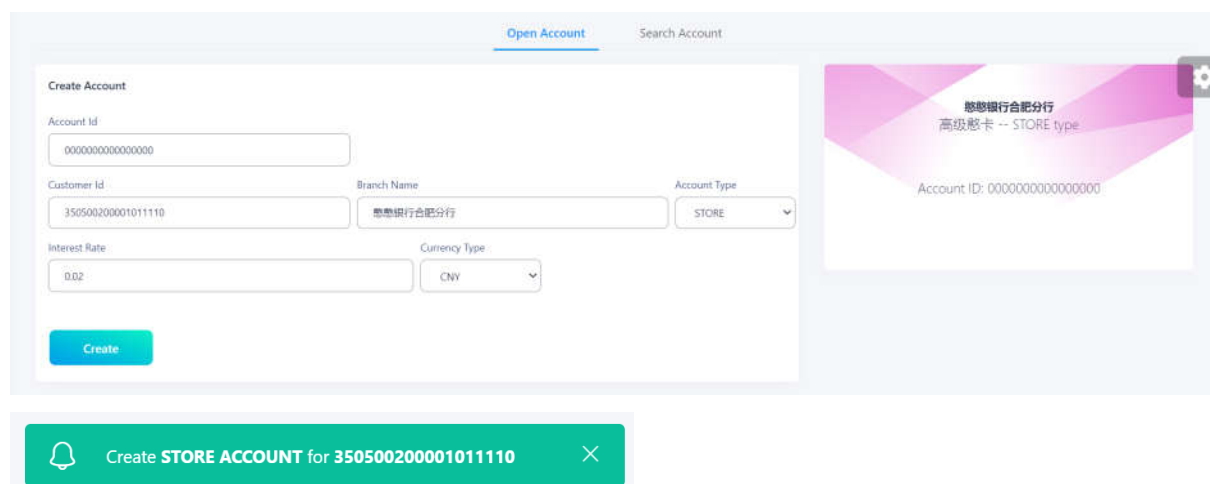
2. 删除没有账户和贷款的客户:



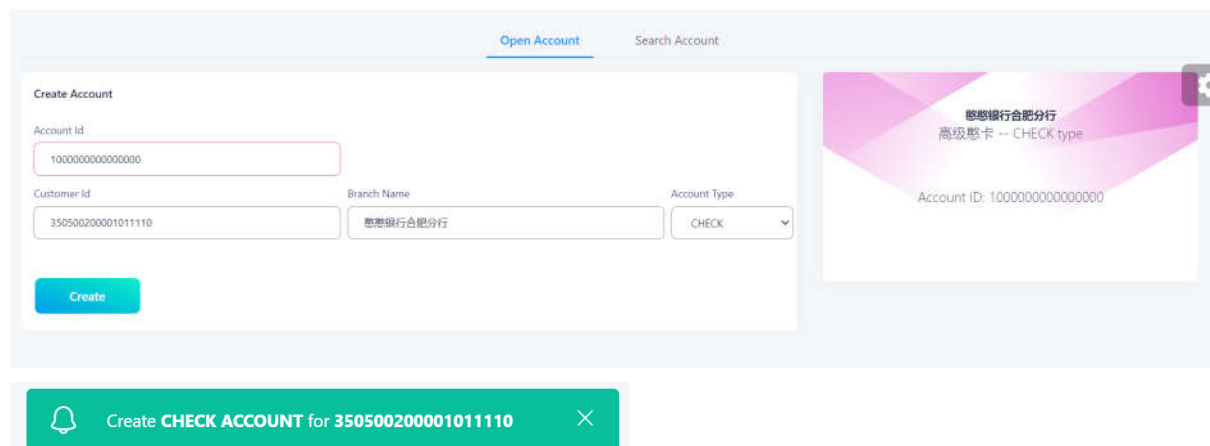
4.2.5 开户

开户的测试用例和测试结果:

1. 开储蓄账户:



2. 开支票账户



3. 开户账户号重复:



The screenshot shows a web application interface for creating an account. At the top, a pink error message box states: "Creating Failed! (IntegrityError) (1062, 'Duplicate entry '350500200001011110-慈慈银行合肥分行' for key 'have_check_account.PRIMARY')". Below the error, the "Create Account" form is visible. It contains three input fields: "Account Id" (with value 0000000000000000), "Customer Id" (with value 350500200001011110), and "Branch Name" (with value 慈慈银行合肥分行). There is also a dropdown for "Account Type" set to "CHECK". A green "Create" button is at the bottom left. On the right, a pink banner for "高级联卡 -- CHECK type" is displayed, showing the "Account ID: 0000000000000000".



4.2.6 搜索账户

搜索账户的测试用例和结果:

1. 按客户身份证号:

The screenshot shows the search results for accounts by customer ID. At the top, a green success message box states: "Search Succeeded!". Below the message, the "Search Conditions" section shows the search criteria: "Account ID" (empty), "Customer Id" (350500200001011110), and "Branch Name" (empty). A green "Search" button is below the criteria. The results are displayed in two tables: "Search Results(Store)" and "Search Results(Check)".

ACC_ID	BRA_NAME	CUS_ID	ACC_TYPE	ACC_BALANCE	STO_INTEREST_RATE	STO_CURRENCY_TYPE	STO_LAST_VISIT_DATE	ACTIONS
0000000000000000	慈慈银行合肥分行	350500200001011110	STORE	0.00	0.02	CNY	Fri, 18 Jun 2021 15:10:01 GMT	 

ACC_ID	BRA_NAME	CUS_ID	ACC_TYPE	ACC_BALANCE	CHE_OVERDRAFT	CHE_LAST_VISIT_DATE	ACTIONS
1000000000000000	慈慈银行合肥分行	350500200001011110	CHECK	0.00	0.00	Fri, 18 Jun 2021 15:10:22 GMT	 

2. 按账户号:

Search Succeed!

Search Conditions

Account ID

0000000000000000

Customer Id



Customer Id(18 digits)

Branch Name

Branch Name

Search

Search Results(Store)

ACC_ID	BRA_NAME	CUS_ID	ACC_TYPE	ACC_BALANCE	STO_INTEREST_RATE	STO_CURRENCY_TYPE	STO_LAST_VISIT_DATE	ACTIONS
0000000000000000	惠惠银行合肥分行	350500200001011110	STORE	0.00	0.02	CNY	Fri, 18 Jun 2021 15:10:01 GMT	 

Search Results(Check)

ACC_ID	BRA_NAME	CUS_ID	ACC_TYPE	ACC_BALANCE	CHE_OVERDRAFT	CHE_LAST_VISIT_DATE	ACTIONS
--------	----------	--------	----------	-------------	---------------	---------------------	---------

3. 按支行名:

Search Succeed!

Search Conditions

Account ID

Account ID(16 digits)

Customer Id



Customer Id(18 digits)

Branch Name



惠惠银行泉州分行

Search

Search Results(Store)

ACC_ID	BRA_NAME	CUS_ID	ACC_TYPE	ACC_BALANCE	STO_INTEREST_RATE	STO_CURRENCY_TYPE	STO_LAST_VISIT_DATE	ACTIONS
0000000000000004	惠惠银行泉州分行	350500200001011114	STORE	666.00	0.06	CNY	Thu, 15 Apr 2021 09:00:00 GMT	 

Search Results(Check)

ACC_ID	BRA_NAME	CUS_ID	ACC_TYPE	ACC_BALANCE	CHE_OVERDRAFT	CHE_LAST_VISIT_DATE	ACTIONS	
0000000000000040	惠惠银行泉州分行	350500200001011114	CHECK	77.00	0.00		Thu, 15 Apr 2021 09:00:00 GMT	 

4.2.7 修改账户

修改账户的测试用例和测试结果:

1. 修改储蓄账户余额

Search

Update account









Account ID: 0000000000000000 Type: STORE



Account Balance: 6666

Interest Rate: 0.02 Account Type: CNY

Save cancel

Search Results(Store)

ACC_ID	BRA_NAME	CUS_ID	RATE	STO_CURRENCY_TYPE	STO_LAST_VISIT_DATE	ACTIONS
0000000000000000	慈惠银行合肥分行	3505002		CNY	Fri, 18 Jun 2021 15:10:01 GMT	 
0000000000000001	慈惠银行合肥分行	3505002		CNY	Sat, 30 Jan 2021 09:00:00 GMT	 
0000000000000003	慈惠银行合肥分行	3505002		CNY	Tue, 26 Jan 2021 09:00:00 GMT	 
0000000000000004	慈惠银行泉州分行	3505002		CNY	Thu, 15 Apr 2021 09:00:00 GMT	 

0000000000000000 慈惠银行合肥分行 350500200001011110 STORE 6,666.00 0.02 CNY Fri, 18 Jun 2021 15:13:06 GMT  

2. 修改支票账户余额

Update account







Account ID: 1000000000000000 Type: CHECK



Account Balance: 66

Overdraft: 0

Save cancel

Search Results(Check)

ACC_ID	BRA_NAME	CUS_ID	CHE_OVERDRAFT	CHE_LAST_VISIT_DATE	ACTIONS
1000000000000000	慈惠银行合肥分行	3505002	0.00	Fri, 18 Jun 2021 15:10:22 GMT	 
0000000000000020	慈惠银行合肥分行	3505002	66.00	Thu, 15 Oct 2020 09:00:00 GMT	 
0000000000000040	慈惠银行泉州分行	3505002	0.00	Thu, 15 Apr 2021 09:00:00 GMT	 

1000000000000000 慈惠银行合肥分行 350500200001011110 CHECK 66.00 0.00 Fri, 18 Jun 2021 15:13:39 GMT  

3. 修改支票账户透支量

Update account







Account ID: 1000000000000000 Type: CHECK



Account Balance: 66

Overdraft: 99

Save cancel

Search Results(Check)

ACC_ID	BRA_NAME	CUS_ID	CHE_OVERDRAFT	CHE_LAST_VISIT_DATE	ACTIONS
1000000000000000	慈惠银行合肥分行	3505002	0.00	Fri, 18 Jun 2021 15:13:39 GMT	 
0000000000000020	慈惠银行合肥分行	3505002	66.00	Thu, 15 Oct 2020 09:00:00 GMT	 
0000000000000040	慈惠银行泉州分行	3505002	0.00	Thu, 15 Apr 2021 09:00:00 GMT	 

1000000000000000 慈惠银行合肥分行 350500200001011110 CHECK 66.00 99.00 Fri, 18 Jun 2021 15:13:59 GMT  



4.2.8 删除账户

删除账户的测试用例和测试结果



1. 删除仍有余额的账户:

Removing Failed! (StillHasBalance) X

Search Results(Store)

ACC_ID	BRA_NAME	CUS_ID	ACC_TYPE	ACC_BALANCE	STO_INTEREST_RATE	STO_CURRENCY_TYPE	STO_LAST_VISIT_DATE	ACTIONS
000000000000000001	慈慈银行合肥分行	350500200001011111	STORE	6,666.00	0.06	CNY	Sat, 30 Jan 2021 09:00:00 GMT	 

2. 删除无余额无透支的账户:

000000000000000000	慈慈银行合肥分行	350500200001011110	STORE	0.00	0.02	CNY	Fri, 18 Jun 2021 15:14:47 GMT	 
--------------------	----------	--------------------	-------	------	------	-----	-------------------------------	---

Remove Account: 0000000000000000 X

4.2.9 新增贷款

新增贷款的测试用例和测试结果

1. 正常新增多人共有贷款:

Create a loan(id=9) of 99 for 350500200001011110,350500200001011111 X

Insert Loan Search Loan Pay Loan

Insert Loan

Branch Name:
慈慈银行合肥分行

Loan Amount:
99

Customer ID:
350500200001011110
350500200001011111

Formatted Loan Amount:
99.00

Create

2. 新增单人所有贷款:

Create a loan(id=10) of 996 for 350500200001011110

Insert Loan Search Loan Pay Loan

Insert Loan

Branch Name
慈惠银行合肥分行

Loan Amount
996

Formatted Loan Amount
996.00

Customer ID
350500200001011110

Create

4.2.10 查询贷款

查询贷款的测试用例和测试结果

1. 无条件搜索所有贷款:






















Search Conditions Search Succeed!

Branch Name Loan ID

Customer ID

Search

Search Results

LOA_ID	BRA_NAME	LOA_AMOUNT	ACTIONS
1	慈惠银行合肥分行	666.00	  
2	慈惠银行合肥分行	333.00	  
3	慈惠银行合肥分行	777.00	  
4	慈惠银行合肥分行	999.00	  
5	慈惠银行合肥分行	555.00	  
6	慈惠银行合肥分行	666.00	  
9	慈惠银行合肥分行	99.00	  

2. 指定支行名

Search Conditions

Search Succeed!

×

Branch Name

慈惠银行合肥分行

Loan ID

Loan ID

Customer ID

Customer ID

Search

Search Results

LOA_ID	BRA_NAME	LOA_AMOUNT	ACTIONS
1	慈惠银行合肥分行	666.00	<div><div></div><div></div><div></div></div>
2	慈惠银行合肥分行	333.00	<div><div></div><div></div><div></div></div>
3	慈惠银行合肥分行	777.00	<div><div></div><div></div><div></div></div>
4	慈惠银行合肥分行	999.00	<div><div></div><div></div><div></div></div>
5	慈惠银行合肥分行	555.00	<div><div></div><div></div><div></div></div>
6	慈惠银行合肥分行	666.00	<div><div></div><div></div><div></div></div>

3. 指定客户身份证号

Search Conditions

Search Succeed!

×

Branch Name

Branch Name

Loan ID

Loan ID

Customer ID

350500200001011111

Search

Search Results

LOA_ID	BRA_NAME	LOA_AMOUNT	ACTIONS
1	慈惠银行合肥分行	666.00	<div><div></div><div></div><div></div></div>
2	慈惠银行合肥分行	333.00	<div><div></div><div></div><div></div></div>
6	慈惠银行合肥分行	666.00	<div><div></div><div></div><div></div></div>
9	慈惠银行合肥分行	99.00	<div><div></div><div></div><div></div></div>

4. 指定贷款号

LOA_ID	BRA_NAME	LOA_AMOUNT	ACTIONS
5	慈惠银行合肥分行	555.00	[edit] [delete] [share]

4.2.11 查询贷款者

查询贷款者的测试用例和测试结果:

1. 查询刚刚添加的多人共有贷款:

ID	Branch Name	Amount	Actions
2	慈惠银行合肥分行		[edit] [delete] [share]
3	慈惠银行合肥分行		[edit] [delete] [share]
4	慈惠银行合肥分行		[edit] [delete] [share]
5	慈惠银行合肥分行		[edit] [delete] [share]
6	慈惠银行合肥分行	666.00	[edit] [delete] [share]
9	慈惠银行合肥分行	99.00	[edit] [delete] [share]
10	慈惠银行合肥分行	996.00	[edit] [delete] [share]

2. 查询刚刚添加的单人所有贷款:

ID	Branch Name	Amount	Actions
2	慈惠银行合肥分行		[edit] [delete] [share]
3	慈惠银行合肥分行		[edit] [delete] [share]
4	慈惠银行合肥分行		[edit] [delete] [share]
5	慈惠银行合肥分行	555.00	[edit] [delete] [share]
6	慈惠银行合肥分行	666.00	[edit] [delete] [share]
9	慈惠银行合肥分行	99.00	[edit] [delete] [share]
10	慈惠银行合肥分行	996.00	[edit] [delete] [share]

4.2.12 支付贷款

支付贷款测试用例和测试结果:

1. 正常支付

The screenshot shows a web application window with a green notification bar at the top stating "Search Succeed!". Below the notification bar are three tabs: "Insert Loan", "Search Loan", and "Pay Loan", with "Pay Loan" being the active tab. The "Loan Information" section contains several input fields: "Loan ID" (1), "Branch Name" (惠惠银行合肥分行), "Loan Amount" (666.00), "Loan Pay Amount Sum" (333.00), "Pay Loan Amount" (333), and "Formatted Pay Loan Amount" (333.00). A green "Pay" button is located below these fields. Below the "Pay Loan" section is a "Pay Loan Records" table with the following data:

LOA_PAY_ID	LOA_PAY_AMOUNT	LOA_PAY_DATE
2	333.00	Fri, 18 Jun 2021 15:23:24 GMT

2. 超额支付

The screenshot shows the same web application window, but with a pink notification bar at the top stating "Search Failed! (PayTooMuch)". The "Loan Information" section shows the same data as the previous screenshot, but the "Pay Loan Amount" is now 6666, and the "Formatted Pay Loan Amount" is 6,666.00. The green "Pay" button is still present.

3. 支付已付清贷款

Search Failed! (LoanAlreadyDone)

Insert LoanSearch LoanPay Loan

Loan Information

Loan ID

1

Branch Name

徽商银行合肥分行

Loan Amount

666.00

Loan Pay Amount Sum

666.00

Pay Loan Amount

1

Formatted Pay Loan Amount

1.00

Pay

Pay Loan Records

LOA_PAY_ID	LOA_PAY_AMOUNT	LOA_PAY_DATE
2	333.00	Fri, 18 Jun 2021 15:23:24 GMT
3	333.00	Fri, 18 Jun 2021 15:23:43 GMT

4.2.13 贷款删除

贷款删除的测试用例和测试结果:

1. 删除已付清贷款

Loan Information

Loan ID

3

Branch Name

徽商银行合肥分行

Loan Amount

777.00

Loan Pay Amount Sum

777.00

Pay Loan Amount

777

Formatted Pay Loan Amount

777.00

Pay

Pay Loan Records

LOA_PAY_ID	LOA_PAY_AMOUNT	LOA_PAY_DATE
5	777.00	Fri, 18 Jun 2021 15:31:51 GMT

3

徽商银行合肥分行

777.00

Remove Loan: 3

2. 删除未付清贷款

Loan Information

Loan ID

4

Branch Name

徽商银行合肥分行

Loan Amount

999.00

Loan Pay Amount Sum

0.00

Pay Loan Amount

0.00

Formatted Pay Loan Amount

NaN

Pay

Pay Loan Records

LOA_PAY_ID	LOA_PAY_AMOUNT	LOA_PAY_DATE
------------	----------------	--------------



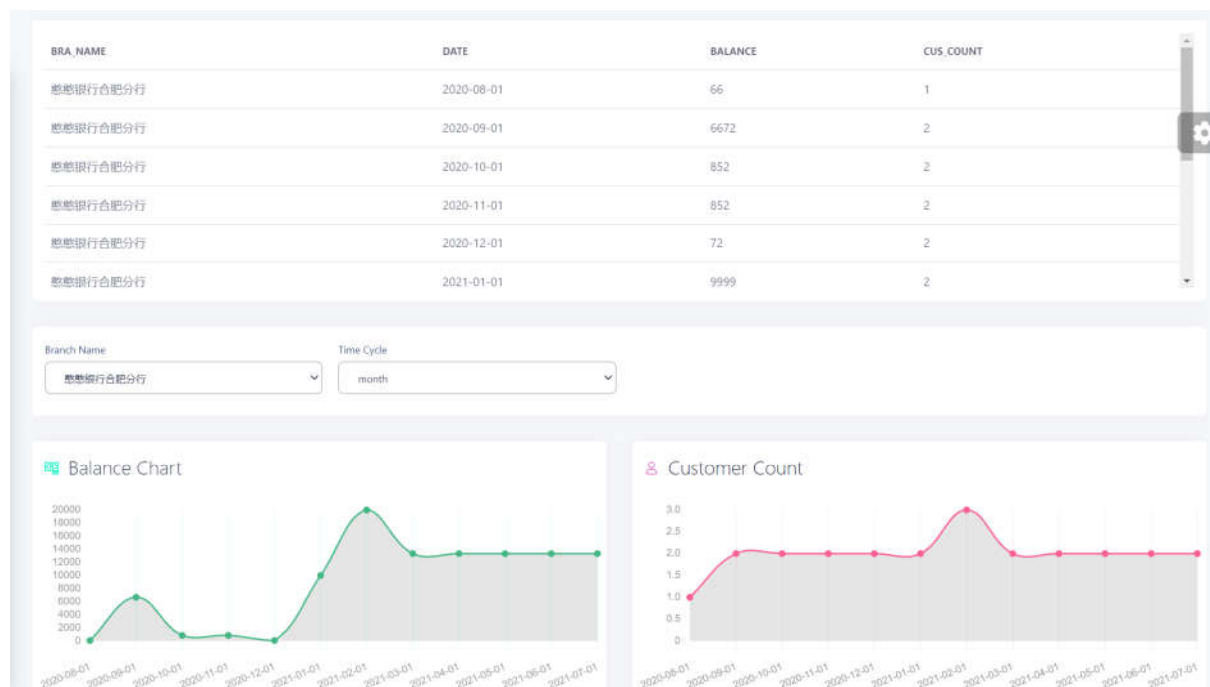
Removing Failed! ([LoanBeingPaid])



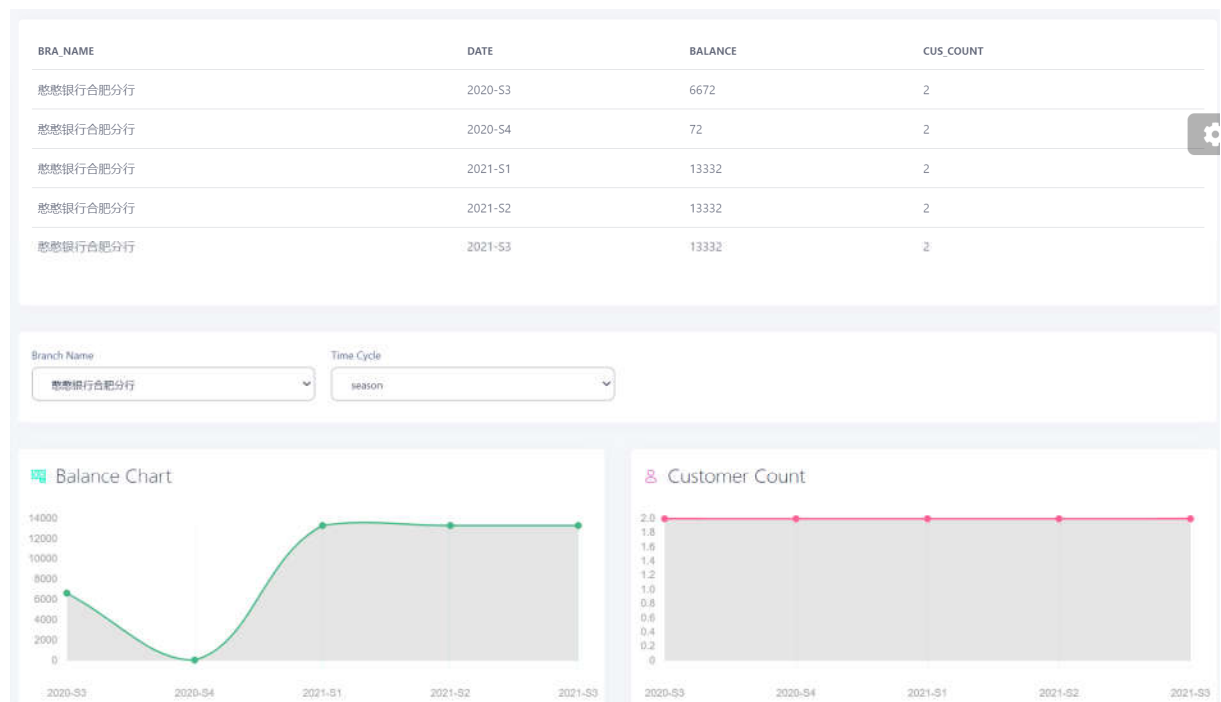
4.2.14 储蓄账户业务统计

储蓄账户业务统计测试用例和测试结果

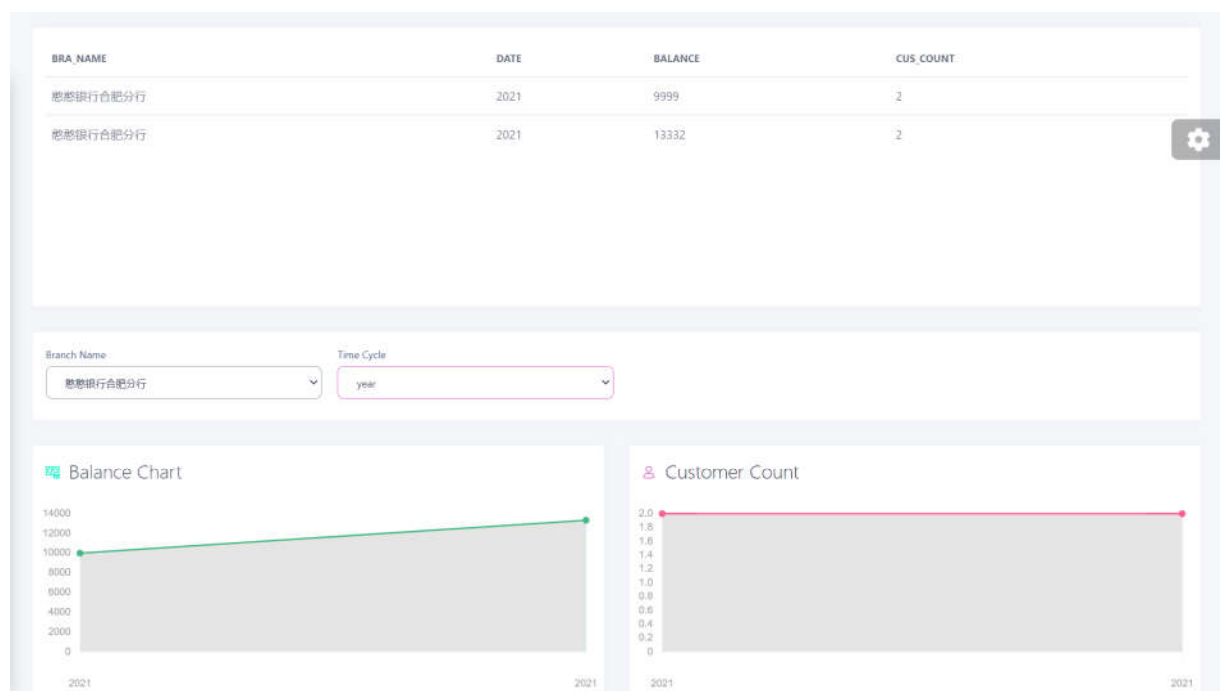
1. 按月



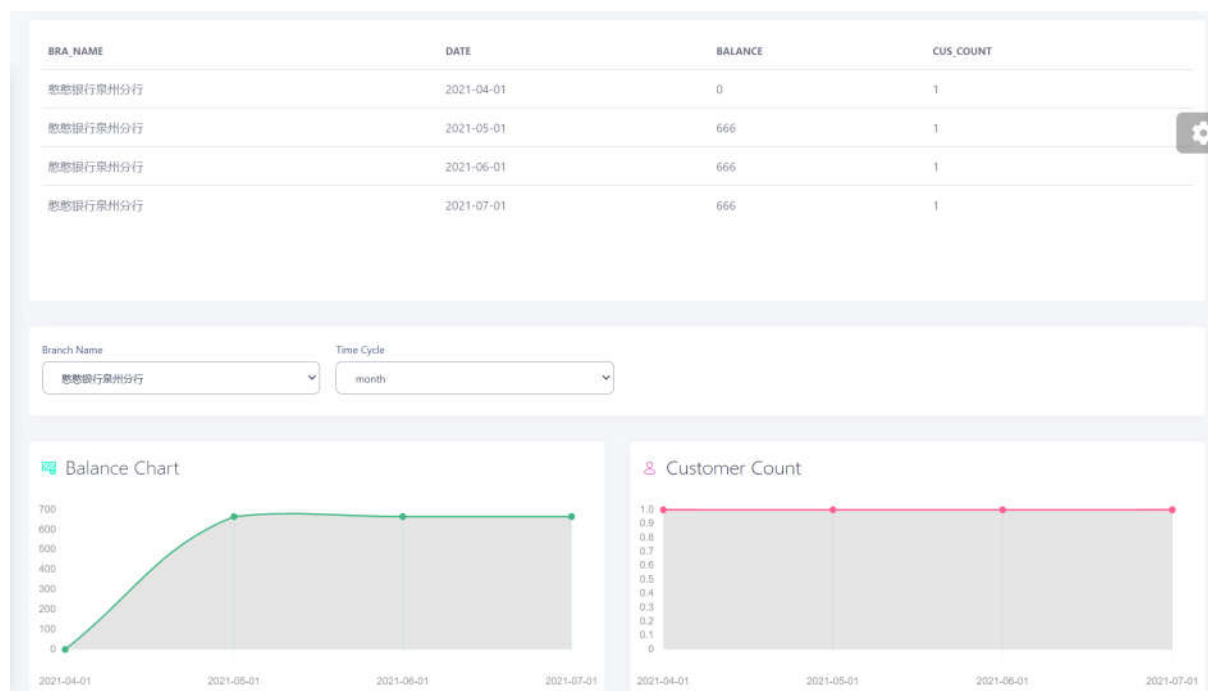
2. 按季:



3. 按年:



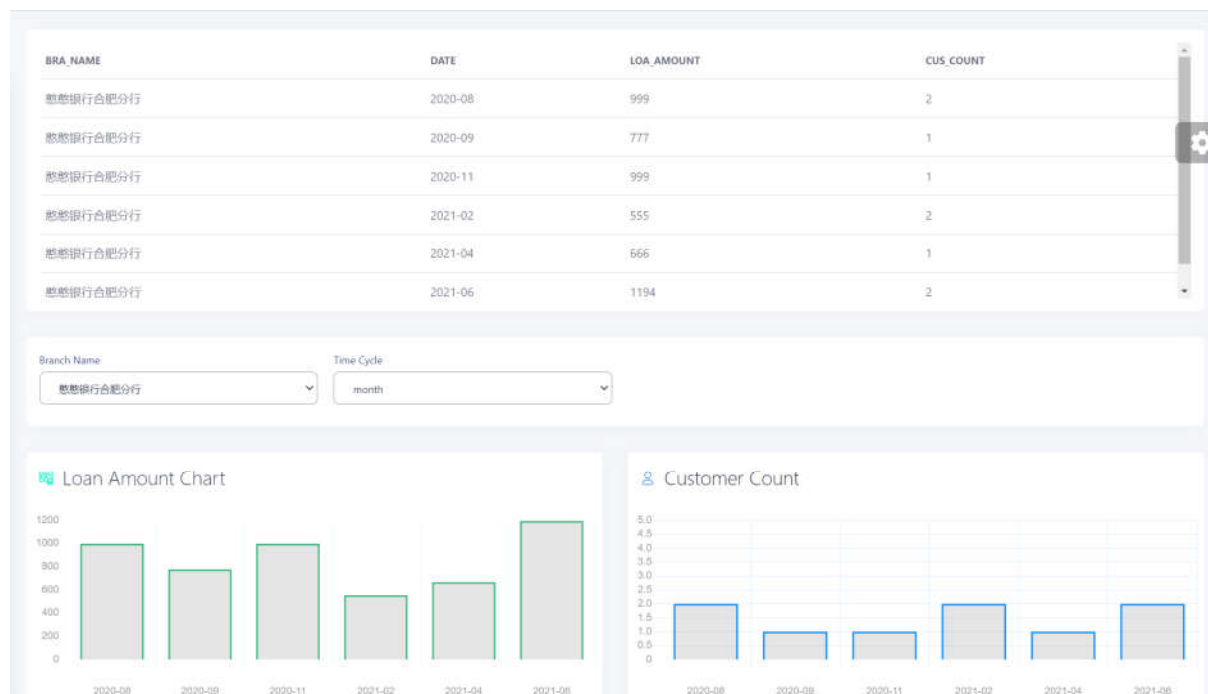
4. 不同支行



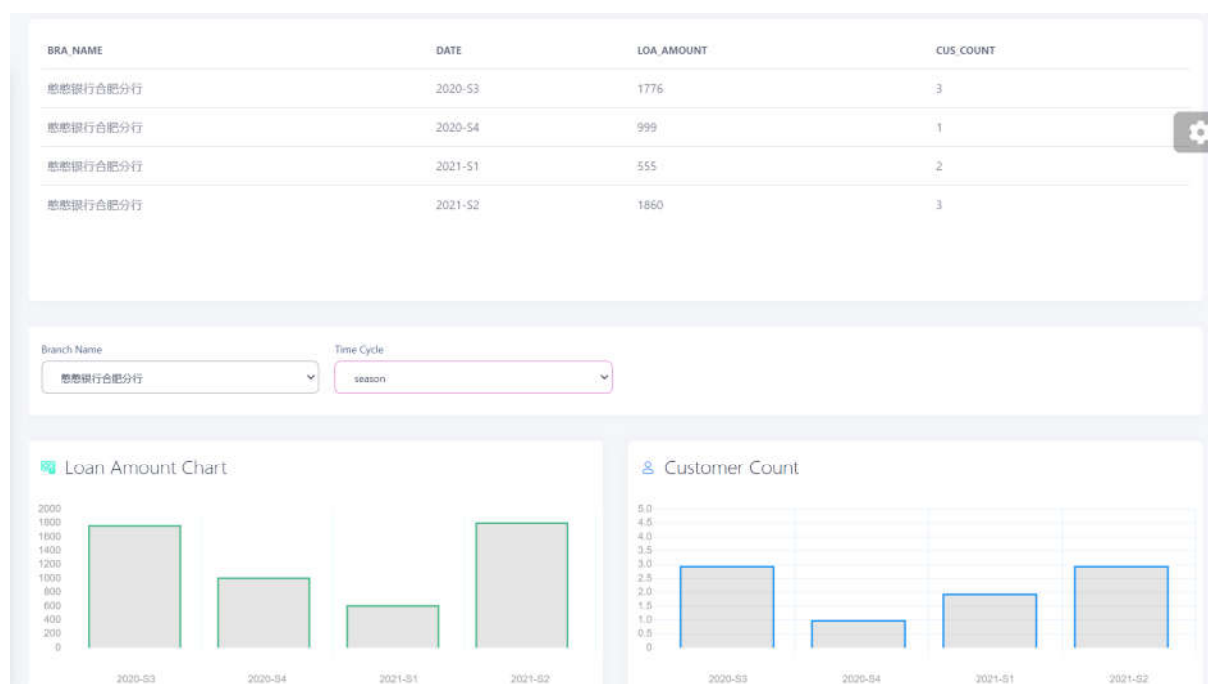
4.2.15 贷款业务统计

贷款业务统计的测试用例和测试结果:

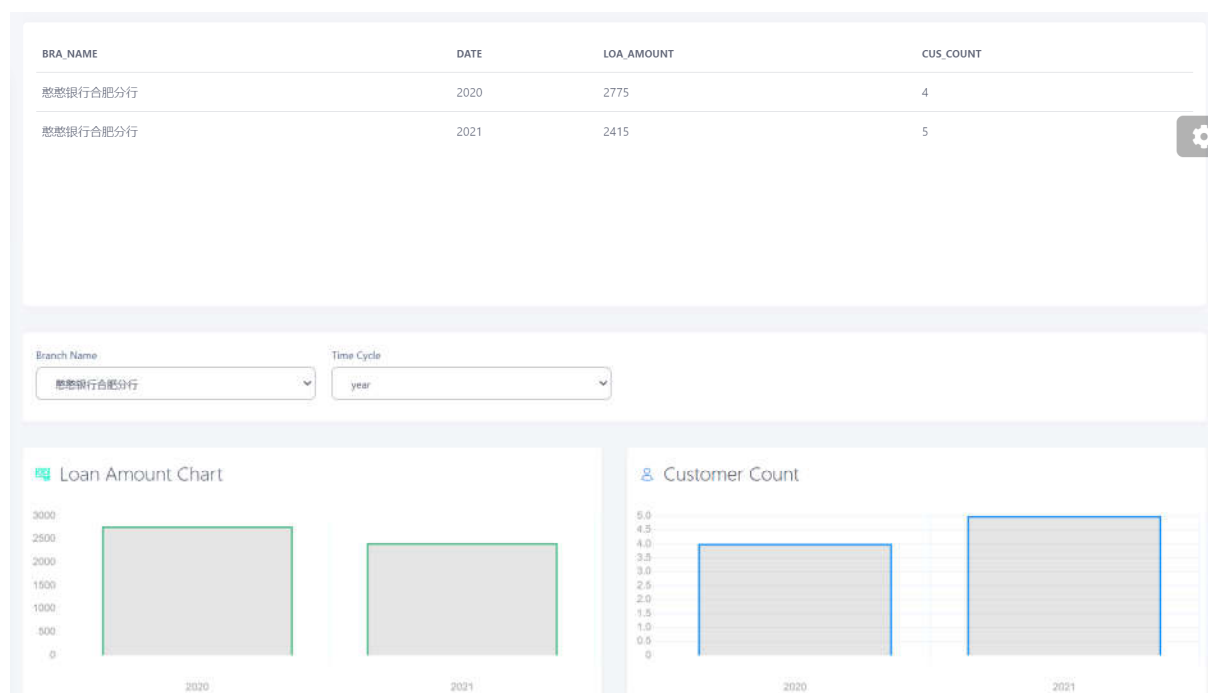
1. 按月



2. 按季



3. 按年




4.2.16 登录与登出

登录与登出的测试用例和测试结果:

1. 登录

Please Login

 **Login Before Continuing**

User Name

admin

Password

.....

Login

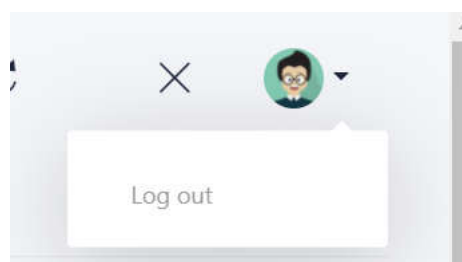
Welcome admin!

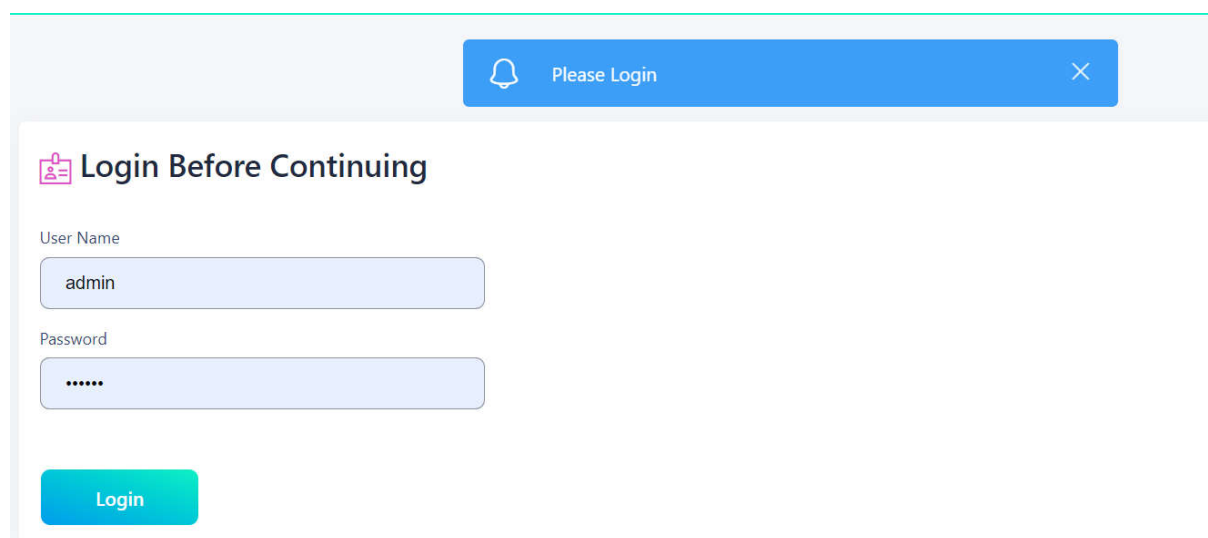
Account Dashboard Loan Dashboard

Account Summary

BRA_NAME	DATE	BALANCE	CUS_COUNT
憨憨银行合肥分行	2020-08-01	66	1
憨憨银行合肥分行	2020-09-01	6672	2
憨憨银行合肥分行	2020-10-01	852	2
憨憨银行合肥分行	2020-11-01	852	2
憨憨银行合肥分行	2020-12-01	72	2
憨憨银行合肥分行	2021-01-01	9999	2

2. 登出





4.3 实现中的难点问题及解决

4.3.1 前端 Web 界面实现难点

在前端 Web 界面的实现中, 使用 JQuery 来操作 DOM 是比较麻烦且冗长的. 以往的项目经验告诉我想要用纯粹 JQuery 实现这样一个比较复杂的银行管理系统并不容易. 因此本次实验中, 我花了大量时间来学习 Vue 框架. Vue (读音 /vju:/, 类似于 view) 是一套用于构建用户界面的渐进式框架. 与其它大型框架不同的是, Vue 被设计为可以自底向上逐层应用. Vue 的核心库只关注视图层, 不仅易于上手, 还便于与第三方库或既有项目整合. 另一方面, 当与现代化的工具链以及各种支持类库结合使用时, Vue 也完全能够为复杂的单页应用提供驱动. 因此对于本系统的实现, Vue 框架无疑是一个比较好的选择.

4.3.2 防止 SQL 注入攻击

SQL 注入攻击是非常著名的一种攻击手段. 根据 Wikipedia 的定义, 它也称 SQL 注入或 SQL 注码, 是发生于应用程序与数据库层的安全漏洞. 简而言之, 是在输入的字符串之中注入 SQL 指令, 在设计不良的程序当中忽略了字符检查, 那么这些注入进去的恶意指令就会被数据库服务器误认为是正常的 SQL 指令而运行, 因此遭到破坏或是入侵.

为了防止 SQL 注入攻击, 我在所有需要用到用户输入的 SQL 请求中都使用了 pymysql 自带的 mogrify 方法来生成合法(合乎开发者想法)的 SQL 请求. 因此非常简单地避免了 SQL 注入风险.

4.3.3 业务统计模块的实现

如果按照原本最开始的设想, 整个银行数据库只管理当前数据, 是难以实现业务统计的. 因此为了解决这个问题, 我额外加入了几个表来记录业务操作, 从而每次需要业务统计的时候能够方便地通过日志表来完成. 这其中日志表项包括账户更新日志, 账户(开/闭)日志, 贷款日志.

基于这些日志表,就可以方便地按月给出各项银行业务的统计信息,包括总金额和总客户数.然后可以通过月信息去聚合出按季度,按年份的信息,从而满足了甲方的要求.

4.3.4 后端开发过程中的测试

刚开始肯定难以同时开展前后端的工作,因此我先考虑去完成后端的代码.但是由于没有前端的交互,后端代码难以得到及时测试,不断累积下来就会变得非常难 DEBUG.

为了解决这个问题,我借助了 Python 的**单元测试模块 unittest**.有了这个模块的帮助,每次完成任何一小段代码片段,都可以及时地进行测试.

4.3.5 抛出异常的处理

整个系统的开发会抛出很多异常,包括但不限于重复添加异常,仍有账户异常,仍有贷款异常等.这些异常处理如果在各个函数中写各种 `try...except...` 子句,则会使代码变得非常冗长.

因此这里采用了 Flask 的 `errorhandler`(错误处理器)来全局地处理异常.这个方法甚至能给前端的编码带来方便.因为通过这个错误处理器,Flask 能够直接向前端返回指定错误信息,并且抛出异常到前端,这样前端的 `axios` 模块也能接收到异常,直接进入异常处理,而不必走正常的执行流程,这避免了大量的 `if...else...` 判断.

5 总结与讨论

本系统开发过程中,从前端到后端学习了不少框架,模块和语言,例如 HTML, JavaScript, Vue, Flask, pymysql, MySQL DBMS 等.这是大学期间少有的全栈开发工程经历,非常有意义.它不仅锻炼了系统设计能力,也考验了我的工程实现能力,更磨砺了我的耐心.不得不说,收获很大.