

哈工大计算机学院系统安全

实验三

完整性访问控制系统设计与实现

学号：1160301011

姓名：宋新彤

指导老师：张玥

时间：2018 年 12 月 26 日

实验 3 完整性访问控制系统设计与实现

一、系统设计说明：

设计完整性访问控制系统，实现系统，并满足某商业公司的完整性访问控制需求。

- (1) 配合第 7 章，为商业公司设计系统，提出该应用系统的安全策略。
- (2) 配合第 9 章 为商业公司设计系统，应用系统满足完整性需求。
- (3) 具体指明是哪类应用系统，应用背景范围不限，可以是银行、股票等，符合商业系统完整性需求即可。

二、实验环境

Eclipse、java1.8、mysql5.7、Navicat12

ATM 网上移动银行系统

三、系统要求：

项目背景：

原先柜台式的交易已不满足当今社会的需求,因而开发一个满足社会需求的网上移动银行成为现在工作的重中之重。

系统目标

管理

方便客户对自己账户的管理（密码，个人资料的修改，绑卡，信用卡等）。

操作

可以使用储蓄卡进行生活缴费，信用卡商城消费，转账、取款等操作

查询

可以方便的实现客户以及管理员对消费历史记录和用户行为分析的查看。

关键技术：JDBC 架构

此银行管理系统分为三个层次：

数据层：主要是对原始数据(数据库)进行操作

运算层：对原始数据(数据库)进行运算，从而达到功能的实现

表示层：通过 java GUI 实现功能的图形可视化处理

- (1) 给出应用系统的安全策略文档。

本银行系统采用整体性原则，多级安全策略保障用户的安全。首先是用户的注册，一是密码的确认输入，通过两次输入密码，在第二次输入时若密码不同，则进行提示，及时进行修改，确保用户设置密码时的正确性。二是验证码的输入，确保能够准确识别机器与用户；其次是用户的登录，通过在数据库中对用户名和密码

的匹配实现登录功能,同时需要数据库系统返回该用户的状态,如果用户被锁定,则需要与系统进行交互,解除锁定。三是改密功能,在实现改密之前,不仅考虑用户的需求,更需要输入原始密码,实现系统的验证,修改成功会返回登录界面重新登录。同时,为了满足用户的需求,还提供了用户找回密码功能,再验证过用户的基本信息(包括姓名与身份证号)后,即可找回密码。其次是用户的取款转账工作,在用户实施该操作后,会插入转账取款记录,以供用户查询,同时可以查看用户的余额、用户的个人信息以及进行用户的账单查询。针对用户的操作,可分为几个模块,分别是转账取款、信用卡开通、生活缴费与日常消费、绑定储蓄卡以及解除绑定。在生活缴费与日常消费模块,需要区分储蓄卡以及信用卡,对于用户的生活缴费只可以使用储蓄卡,从而实现对用户权限的一定限制,提高系统的安全性。最后是超级管理员功能模块,对于此银行系统的超级管理员,只可以查看与银行系统功能相关的数据即用户的行为数据,比如统计消费总数、注册总数,转账取款的总金额数等等,但即使是超级用户,也不可以查看用户的个人信息,从而保障用户的隐私,提高系统的安全性。

(2) 提供交互界面,能够完成录入、查询等功能。

用户登录



用户注册界面：



The image shows a user registration window titled "注册界面" (Registration Interface). It features a menu bar with "操作" (Operation) and "更换皮肤" (Change Skin). The main area contains four input fields labeled "用户名：" (Username), "密码：" (Password), "确认密码：" (Confirm Password), and "验证码：" (Verification Code). Below the verification code field is a CAPTCHA image showing the text "I HyG". At the bottom is a "注册" (Register) button.

基于对系统安全的保障，以上信息不可以为空。

银行系统主界面：

进入之后首先点击账户管理——编辑用户信息（填写客户资料）
老用户已填写会自动从数据库访问显示之前填写的信息。
新用户则需要填写信息写入数据库。



进入查询界面查看客户个人信息：



找回密码界面：

有了客户信息才可以执行找回密码的功能

填写相关信息后 文本框会从数据库访问 显示你之前的密码 并更改账户状态，恢复正常



银行信息：



添加收款人界面：

这是手动添加收款人信息， 查询界面点击浏览就显示所添加的收款人信息。



当转账时，数据库自动检测你的收款人名单是否有该用户的信息，如果没有则自动添加用户信息到收款人名单中

绑定银行卡界面：

尊敬的用户: dk01

请填写信息

银行卡序列号: 1000

银行卡卡号: 1001

存息编号: 101

设置支付密码: 111111

活期金额: 10000

定期金额: 10000

注册 **返回**

消息
绑定成功!

dk01

增加 **解绑** **保存** **返回** **退出**

card_serial	uid	client_id	bank_id	int_serial	card_id	card_ps	card_status	car...	c
1000	27	27	1	101	1001	111111	1		2018-1

解绑银行卡界面:

输入卡号信息 和密码 进行解绑, 前提是 必须没有消费记录和转账取款记录。
否则无法解绑

尊敬的 dk02: 您好!

解绑卡号:

卡号密码:

确认 返回

取款界面:

取款默认从银行卡里的定期余额进行扣款到用户的虚拟钱包中。输入要取款的金额 3000, 选择定期存款的时间 5 年, 利率可自己设置。

右侧会计算出到期的利息和本息, 再输入从哪张银行卡取出, 点击“取出”。显示成功。

尊敬的 dk01: 欢迎进入取款业务

操作 设置

存款利息计算

存款金额 元 到期利息

存款时间 年 到期本息

存款利率 % 到期时间

起始时间 银行卡号

取出 计算

存款利息计算

贷款金额 元 应还利息

贷款时间 年 应还本息

贷款利率 % 还款时间

贷款日期

计算

消息

取钱交易成功!

卡号: 1001

剩余余额为: 7000.00

您的钱包余额为: 5500.00

确定

显示取款记录界面

dk01						
增加 删除 保存 返回 退出						
wid_num	card_serial	int_serial	wid_pre_money	wid_interest	wid_get_money	card_balance
w00116187	1000	611	3000	1500	4500	7000

同行转账界面：

绑卡后就可以进行转账操作，首先进入同行转账界面，输入信息，要求输入密码

测试卡号： dk01 的卡号：1001 密码：111111

dk02 的卡号 :2001 密码：222222 1001 向 2001 转账

付款卡号：

收款卡号：

转账金额：

确认

返回

付款卡号：

收款卡号：

转账金额：

输入

?

请输入支付密码：

确定

取消

转账成功，同行无手续费

消息

i

转账交易成功!

剩余余额为: 9000.0

确定

跨行转账界面：

不同于同行转账的是 要输入对方姓名，并且有手续费

对方姓名:

dk02name

对方卡号:

2001

付款卡号:

1001

转账金额:

2000

确认

返回

消息

转账交易成功!

剩余余额为: 6998.0

手续费为:2.00

确定

跨行同行转账记录界面:

增加 删除 保存 返回 退出

trade_id	card_serial	client_id	int_out	to_cardid	
91112016	1000	28	2	2001	2018-
91116905	2000	28	2	1001	2018-
91119065	1000	27	2	2001	2018-
T91114314	1000	27	1	2001	2018-

存款界面:

账户的余额作为虚拟钱包，从钱包扣款存入到指定的储蓄卡卡号的活期、定期余额中。

存款金额:

1000

*限额100000

存款卡号:

1001

活期存入

消息

交易成功!

当前账户余额为: 3500.0

确定

存款金额:

1000

*限额100000

存款卡号:

1001

活期存入

消息

交易成功!

当前账户余额为: 4500.0

确定

缴费界面:

储蓄卡消费，进入缴费界面选择手机、交通、医保、生活缴费进行充值，扣款自动从输入卡号的活期余额进行扣款支付



信用卡开通界面:

账户 客户 ID 会自动从数据库中访问显示到界面, 免密支付当设置为 2 即开启, 消费就无需再输入密码

开通后, 查询信用卡账单, 可以查看 新开通的信用卡账单使用情况

The image shows a software window titled "尊敬的用户: dk01" (Respected user: dk01). The main area is titled "请填写信息" (Please fill in the information) in red. It contains several input fields: "您的账户 ID:" (Your Account ID) with value "27", "您的客户 ID:" (Your Customer ID) with value "27", "设置支付密码:" (Set Payment Password) with masked characters "....." and a red note "*请设置6位数密码" (Please set a 6-digit password), "还款账号:" (Repayment Account) with value "1001", "免密设置:" (Non-password setting) with a dropdown menu showing "1" and a red note "*1关闭, 2开启" (1 Close, 2 Open), "自动还款:" (Automatic repayment) with a dropdown menu showing "1" and a red note "*1关闭, 2开启" (1 Close, 2 Open), and "申请额度:" (Apply for limit) with value "20000". At the bottom is a green button labeled "开通" (Open). A "消息" (Message) dialog box is open, displaying "开通成功!" (Open successful!) and a "确定" (Confirm) button.

当前程序	责任人
登录	业务员 1
注册	业务员 2
找回密码	业务员 3
查看银行信息	业务员 4
超管后台管理	后台进程
缴费	业务员 5
手机缴费	业务员 6
生活缴费	业务员 7
商城购物	业务员 8
交通缴费	业务员 9
绑定储蓄卡	业务员 10
避免重复开通	业务员 11
解绑储蓄卡	业务员 12
存款	业务员 13
跨行转账	业务员 14
同行转账	业务员 15
取款	业务员 16
信用卡还款	业务员 17
查询	业务员 18
个人信息编辑	业务员 19
添加收款人	业务员 20
修改用户密码	业务员 21

2018-12-27 10:23:15dk01已成功登录, 当前责任人: 业务员1
2018-12-27 10:23:17当前程序: 查看银行信息; 责任人: 业务员4
当前银行名称:中国ATM银行
银行地址:黑龙江省哈尔滨市
当前程序: 查询; 责任人: 业务员18
查询余额
2018-12-27 10:24:06当前程序: 查询; 责任人: 业务员18
查看个人信息
2018-12-27 10:24:34当前程序: 查看银行信息; 责任人: 业务员4
当前银行名称:中国ATM银行
银行地址:黑龙江省哈尔滨市
2018-12-27 10:25:25当前程序: 超管后台管理; 责任人: 后台进程
截止2018-12-27 10:25:25.0
转账总金额:9007.00

日志信息如上所示: 在每个事务中, 都至少有两个责任人, 一个是用户, 一个是系统业务员, 某些业务还包括超管后台。如果因为用户的账号名、密码、验证码、取款金额过大等输入错误引起的问题, 业务的责任人为用户, 如果用户的密码、用户名等各项输入均无错误, 事务出现了问题, 责任人为当前业务员, 通过查询日志, 可以清楚得知是哪个进程出现了问题, 进而改正。

(4) 保存审计日志。

日志信息如下:

其中包含当前操作时间, 此操作的用户以及内容、当前的责任人是业务员还是用户等, 从而可以清晰地实现责任分离原则

2018-12-27 10:23:15dk01已成功登录, 当前责任人: 业务员1
2018-12-27 10:23:17当前程序: 查看银行信息; 责任人: 业务员4
当前银行名称:中国ATM银行
银行地址:黑龙江省哈尔滨市
当前程序: 查询; 责任人: 业务员18
查询余额
2018-12-27 10:24:06当前程序: 查询; 责任人: 业务员18
查看个人信息
2018-12-27 10:24:34当前程序: 查看银行信息; 责任人: 业务员4
当前银行名称:中国ATM银行
银行地址:黑龙江省哈尔滨市
2018-12-27 10:25:25当前程序: 超管后台管理; 责任人: 后台进程
截止2018-12-27 10:25:25.0
转账总金额:9007.00

(5) 遵循 Clark-Wilson 模型, 定义应用系统的完整性限制条件。

Clark-Wilson 模型

系统中, 执行操作之前、之后, 系统一直处于一致性状态

运行良好的事务：使系统从一个一致性状态转移到另一个一致性状态

模型考虑如下几点：

- 1) 主体必须被识别和认证
- 2) 客体只能通过规定的程序进行操作
- 3) 主体只能执行规定的程序
- 4) 必须维护正确的审计日志
- 5) 系统必须被证明能够正确工作

实体包括：

CDI：受限数据项

数据要受完整性约束

UDI：非受限的数据项

数据不受完整性约束

TP：交易过程

将系统从一个有效状态转移到另一个有效状态的过程

IVP：完整性验证过程

检查 CDI 遵守完整性限制的过程

在本银行管理系统中，用户为主体，所以用户在登录此系统时，必须输入用户名、密码以及验证码进行验证。客体作为超管用户登录到系统中，只可以执行部分操作，对于此银行系统的超级管理员，只可以查看与银行系统功能相关的数据即用户的行为数据，比如统计消费总数、注册总数，转账取款的总金额数等等，但即使是超级用户，也不可以查看用户的个人信息，从而保障用户的隐私，提高系统的安全性，即客体只能通过规定的程序进行操作。用户只可以执行部分操作，像修改密码，查看银行信息、查看用户个人信息，存钱，取钱，绑定储蓄卡，解绑储蓄卡，实现跨行转账，实现同行转账，开通信用卡，进行生活缴费等。同时，在本银行系统中，添加了详细的访问日志，在访问日志中，针对每一个进程以及当前进程的责任人进行了详细记录，在一个事务的每一个进程中，都有着详细的责任人记录，从而实现系统的完整性。由上面的结果截图可知，系统能够正常运行。

(6)遵循 Clark-Wilson 模型的证明规则和实施规则，并在设计报告中有所体现。

Certification rule (CR1) 当任意 IVP 运行时，它必须保证所有的 CDI 处于有效状态

TP 的功能：转换

CR2 对相关联的 CDI，一个 TP 必须将这些 CDI 从一个有效状态转到另一个有效状态

一个特定的 TP 和几个相关 CDIs 相关联

实施规则 1 and 2

TP \rightarrow CDI：保证具有关联关系

Enforcement rule (ER1) 系统要维护关联关系，保证经过验证的 TP 操作相应的 CDI

user \rightarrow TP \rightarrow CDI：关联关系

ER2 TP 操作 CDI 时，保证操作用户有权对相应 CDI 做操作，TP 所代表的用户是 CDI 的真实用户

三元组 { user, TP, {CDI set} }

证明规则 3、实施规则 3

满足责任分类原则

CR3 系统执行操作时，符合责任分离原则

模型需要保证用户身份和执行代码身份一致

验证用户身份

ER3 系统执行 TP 前，应验证用户身份

验证客户身份，登录系统的操作员身份

实施规则 4

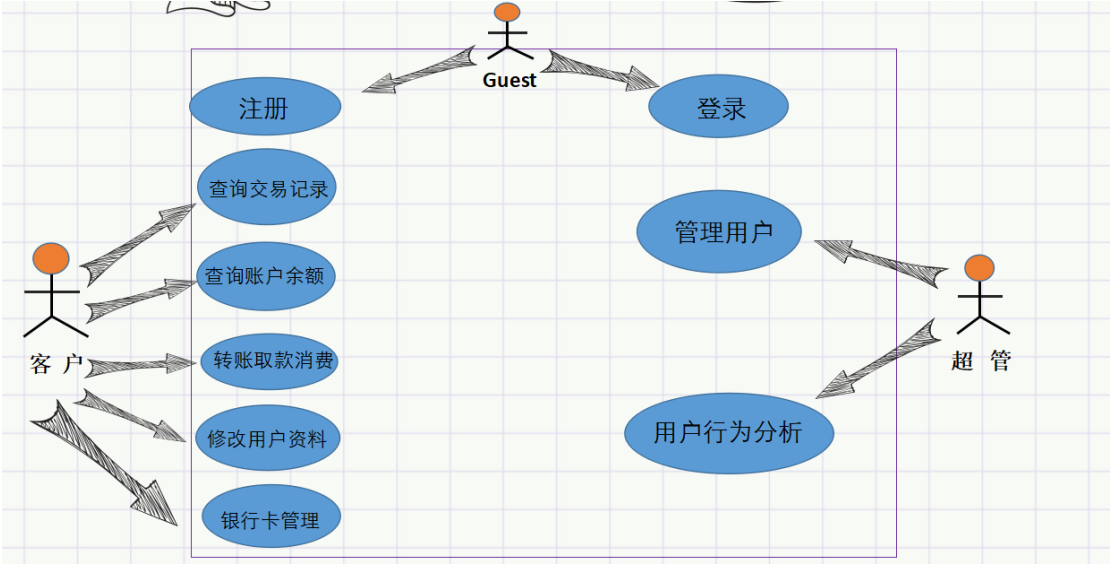
谁可以执行 TP——需被授权

ER4 只有可以授予 TP 访问规则的主体才能修改列表中相应的表项，授权主体不能执行 TP 操作。

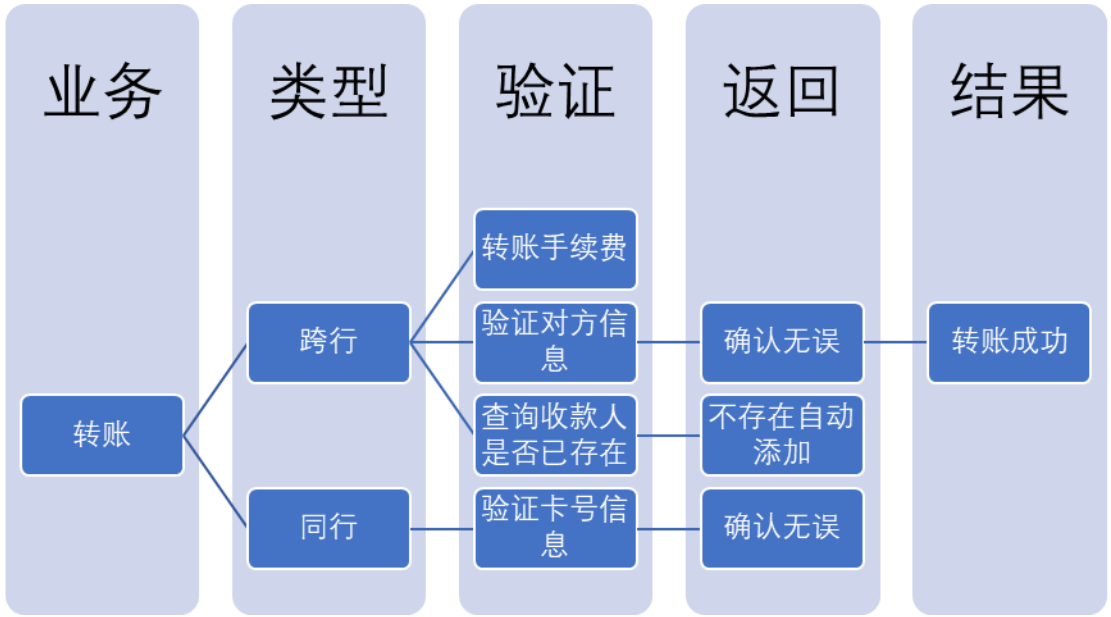
在本银行系统中，有表示层、运算层以及数据层三个层次，当用户信息输入无误后，则从表示层跳至运算层，调用、修改数据库信息，进而跳入数据层，ER1 得证。当用户输入无误后，系统切换当前责任人为业务员，相当于提供了授权，ER2

得证，在用户登录注册时，存在验证码，身份信息等(提供信息过少时，不能进行注册)在以管理员身份登录时，存在超管口令等，ER3 得证。

(7) 系统设计功能流程图



转账业务查询



(8) 部分代码演示如下，数据库演示如下：

注册界面:

```
JLabel lblNewLabel_2 = new JLabel("确认密码 :");
lblNewLabel_2.setFont(new Font("新宋体", Font.BOLD, 18));
lblNewLabel_2.setBounds(70, 211, 101, 28);
contentPane.add(lblNewLabel_2);

JLabel validcode = new JLabel("验证码 :");
validcode.setFont(new Font("新宋体", Font.BOLD, 18));
validcode.setBounds(74, 275, 101, 28);
contentPane.add(validcode);

vcode.setBounds(197, 315, 123, 30);
contentPane.add(vcode);

u_name = new JTextField();
u_name.setBounds(187, 85, 123, 24);
contentPane.add(u_name);
u_name.setColumns(10);

u_password = new JPasswordField();
u_password.setBounds(187, 153, 123, 24);
contentPane.add(u_password);

u_password_1 = new JPasswordField();
u_password_1.setBounds(187, 215, 123, 24);
contentPane.add(u_password_1);

jtf_code = new JTextField();
jtf_code.setBounds(187, 275, 123, 24);
contentPane.add(jtf_code);

JButton OKButton = new JButton("注册"); //注册按钮设置
OKButton.setFont(new Font("新宋体", Font.BOLD, 20));

        } else {
            SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");//设置日期格式
            string=string+df.format(new Date());
            string=string+"当前程序: 注册; 当前责任人: 用户"+ "\n";
            string=string+u_name.getText()+" 未输入确认密码! "+ "\n";
            JOptionPane.showMessageDialog(null, "未输入确认密码! ");
        }

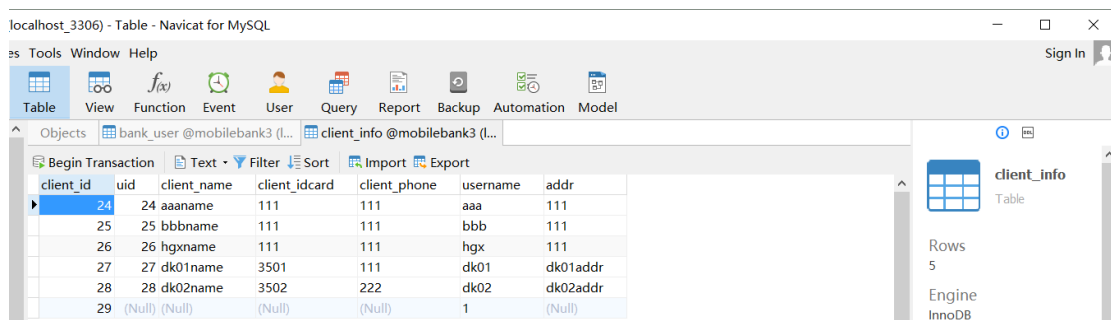
        } else {
            SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");//设置日期格式
            string=string+df.format(new Date());
            string=string+"当前程序: 注册; 当前责任人: 用户"+ "\n";
            string=string+u_name.getText()+" 未输入密码! "+ "\n";
            JOptionPane.showMessageDialog(null, "未输入密码! ");
        }
    }
    } else {
        SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");//设置日期格式
        string=string+df.format(new Date());
        string=string+"当前程序: 注册; 当前责任人: 用户"+ "\n";
        string=string+u_name.getText()+" 未输入用户名! "+ "\n";
        JOptionPane.showMessageDialog(null, "未输入用户名! ");
    }
}
}else{
    SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");//设置日期格式
    string=string+df.format(new Date());
    string=string+"当前程序: 注册; 当前责任人: 用户"+ "\n";
    string=string+u_name.getText()+" 您输入的验证码有误! "+ "\n";
    JOptionPane.showMessageDialog(null, "您输入的验证码有误! ");
}
}else{
    SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");//设置日期格式
    string=string+df.format(new Date());
    string=string+"当前程序: 注册; 当前责任人: 用户"+ "\n";
```

```

if(!jtf_code.getText().equals("")){
    if(!vcode.getCode().equals(jtf_code.getText())) {
        if (!u_name.getText().equals("")) {
            if (!u_password.getText().equals("")) {
                if (!u_password_1.getText().equals("")) {
                    if (u_password.getText().equals(u_password_1.getText())) {
                        if (bu==null) { // !u_name.getText().trim().equals(user.getName())
                            SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");//设置日期格式
                            string=string+df.format(new Date());
                            string=string+"当前程序：注册；当前责任人：业务员2"+'\n';
                            string=string+u_name.getText()+" 注册成功,可以继续登录! "+'\n';
                            asi.addUser(u_name.getText(),u_password.getText());
                            asi.addClientinfo2(u_name.getText());
                            setVisible(false);
                            LoginFrame frame = new LoginFrame();
                            frame.setVisible(true);
                            JOptionPane.showMessageDialog(null, "注册成功,可以继续登录!");
                        } else {
                            SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");//设置日期格式
                            string=string+df.format(new Date());
                            string=string+"当前程序：注册；当前责任人：用户"+'\n';
                            string=string+u_name.getText()+" 用户名已经存在! "+'\n';
                            JOptionPane.showMessageDialog(null, "用户名已经存在!");
                        }
                    } else {
                        SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");//设置日期格式
                        string=string+df.format(new Date());
                        string=string+"当前程序：注册；当前责任人：用户"+'\n';
                        string=string+u_name.getText()+" 密码确认不符! "+'\n';
                        JOptionPane.showMessageDialog(null, "密码确认不符!");
                    }
                }
            }
        }
    }
}

```

数据库：



localhost_3306) - Table - Navicat for MySQL

Tools Window Help Sign In

Table View Function Event User Query Report Backup Automation Model

Objects bank_user @mobilebank3 (l... client_info @mobilebank3 (l...

Begin Transaction Text Filter Sort Import Export

client_id	uid	client_name	client_idcard	client_phone	username	addr
24	24	aaaname	111	111	aaa	111
25	25	bbbnname	111	111	bbb	111
26	26	hgxbname	111	111	hgxb	111
27	27	dk01name	3501	111	dk01	dk01addr
28	28	dk02name	3502	222	dk02	dk02addr
29	(Null)	(Null)	(Null)	(Null)	1	(Null)

client_info Table

Rows 5

Engine InnoDB

(9) 源代码

GitHub:<https://github.com/1160301011/database>

(10) 心得体会

通过本次实验，我对于数据库有了初步的了解，同时也了解到了如何使用 java 关联数据库进行开发，同时加深了对于完整性的理解，了解了责任分离制度，构造多个 CDI 相互之间的关联关系，用户授权的重要意义以及审计日志在项目开发中的重要作用，感觉受益匪浅。