**实验 4 完整性访问控制系统设计与实现**

学号：1170300421

姓名：贺宗磊

班号：1703202

邮箱：[zhiaixululuxu@126.com](mailto:zhiaixululuxu@126.com)

1. 给出应用系统的安全策略文档。

系统本身以角色赋予不同的权限，分为用户和管理员

用户拥有申请存取款功能，查询，更改密码等功能。用户在申请存取款后，会生成一个账单，提交到管理员处。待管理员同意后，才能执行操作。

管理员有查询任意用户和与普通用户一样的权限如，更改密码，注册申请等功能。

在登录界面可以进行两种操作——登录和注册。

取款和存款需要首先输入金额，然后生成一个账单。管理员登录后可以看见用户申请的账单信息，里面有用户的用户名和更改金额。管理员首先查询用户信息，检查是否为系统内用户，认证通过后，该账单被删除，该操作执行成功。如果管理员没有同意，则直接取消账单。注意：取款金额不能超过当前拥有的金额。

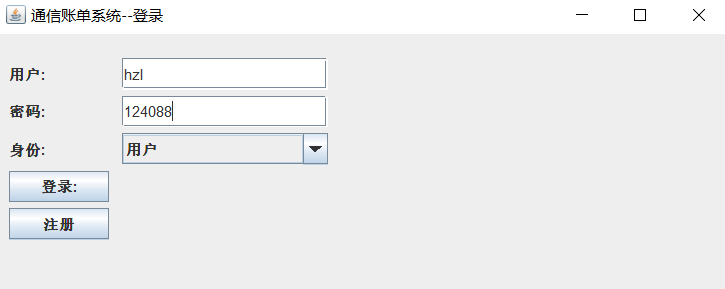
注，为了体现责任分离原则，模拟当存款少于5000时，由一个管理员审核，当大于5000时，除了管理员root审核外，还需要另一个管理员审核，体现出为责任分离。

查询是直接调用当前数据库中的数据，没有更改数据库中的信息，不需要管理员进行同意。

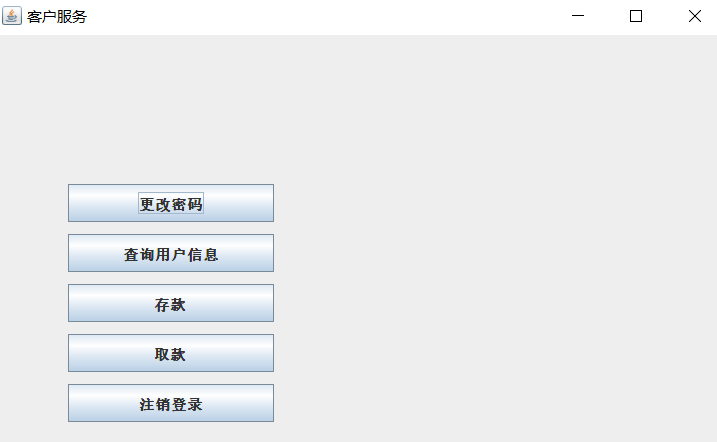
用户和管理员的每一步操作均被记录写入日志，方便最后审核复查。

2. 提供交互界面，能够完成录入、查询等功能。

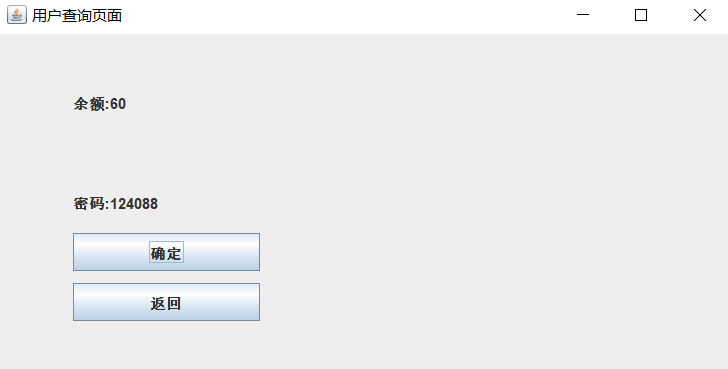
登陆界面：



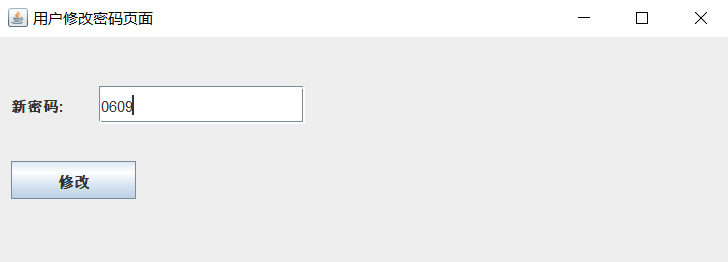
用户登录界面：

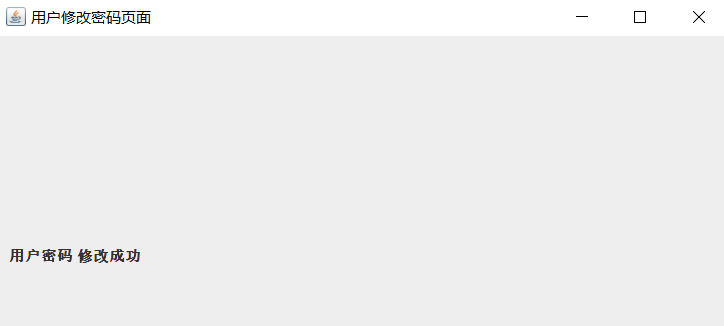


查询功能：

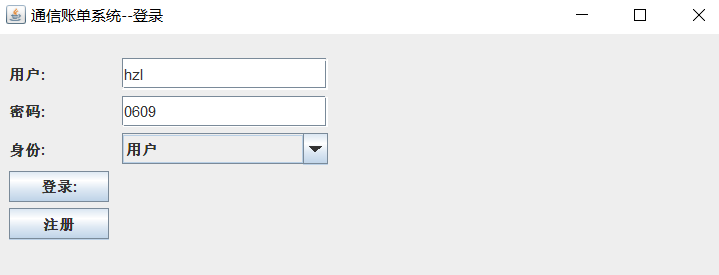


修改密码功能：

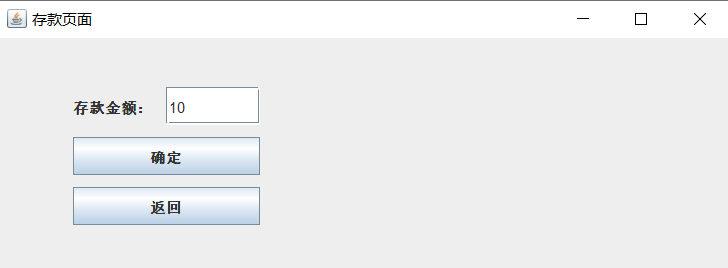




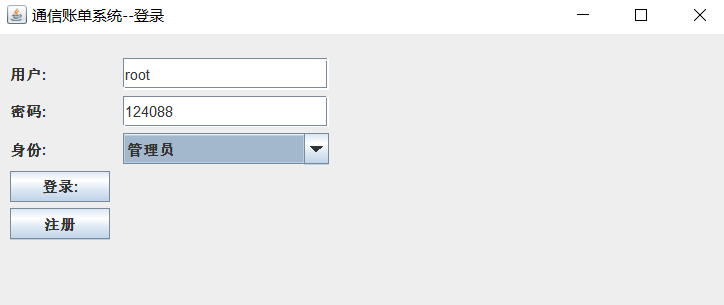
修改成功重新登录：



存款功能：



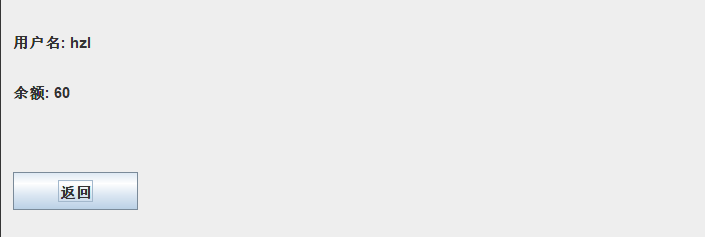
管理员登录界面：



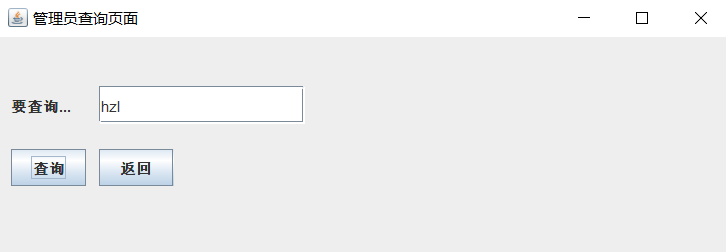
查看订单信息：

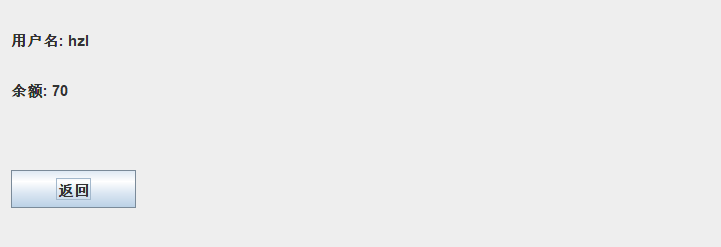


首先查看用户信息，输入用户名字查询



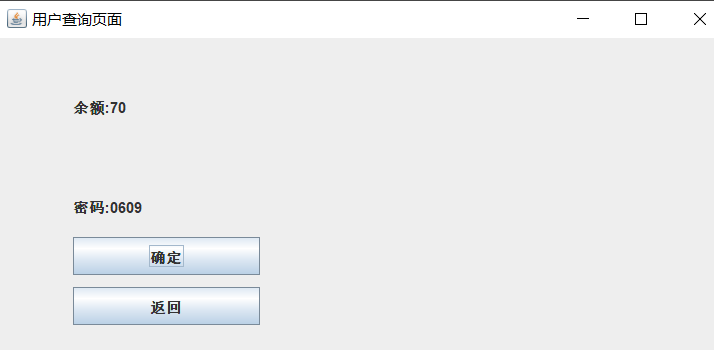
审核通过后管理员再次查询用户信息（同意或不同意）







用户查看自己的信息，可发现充值成功：



3. 满足责任分离原则。

在完整性策略定义的授权方式里面，责任分离原则禁止一个实体完成单独完成一个操作。我们的用户相当于一个实体，对于查询和退出登陆两种操作是用户自身的权力，而且不影响数据的完整性。

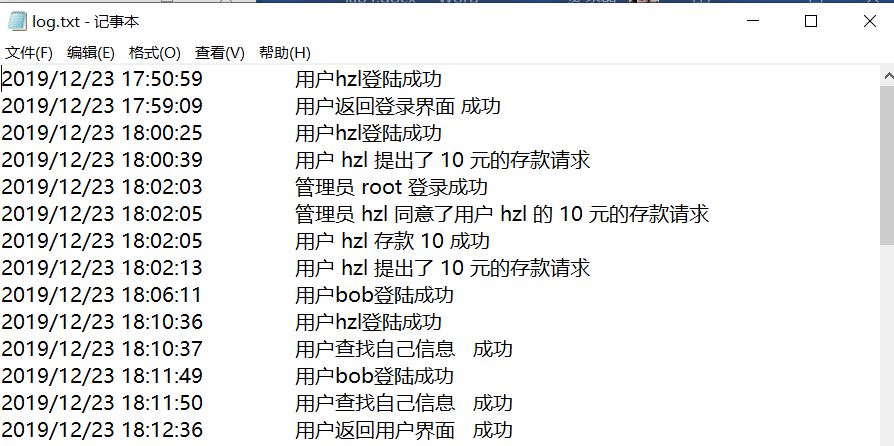
对于取款和存款两种操作，会对数据库中的内容进行修改，所以此时必须采用责任分离的原则。

为此，增加一个管理员身份的用户。单独有管理员时他不能做任何事情，单独有用户他只能查询余额，只有在用户申请这两种然后管理员同意时操作才能完成，符合责任分离的原则，同时也能保证数据的完整性。

除此之外，系统对对不同数目金额采用不同审核，当数目小于5000时，由一位管理员审核，反之由两位管理员审核，由此区分企业用户和普通用户，降低银行责任风险。

4. 保存审计日志。

所有的日志，都保存在 log.txt 内，部分信息内容如下：



5. 遵循Clark-Wilson模型，定义应用系统的完整性限制条件。

首先由定义可知：

Clark-Wilson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

如果把管理员看作主体，用户为客体，则管理员登陆界面即为认证过程，管理员的用户和密码只有自己才能知道的，满足1）。

作为客体的用户只能进行取款、存款、查询、返回四种操作，满足2）

主体管理员只能进行认证和用户信息的审核，不能单独进行存款取款等操作，满足3）

证明4）、5）

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。

证明规则1：

当任意 IVP 运行时, 它必须保证所有的CDI处于有效状态

当用户登录时，在没有管理员同意的情况下，不能对数据库中的数据进行操作。只能提交账单申请，待管理员同意后，才能更改数据库。

证明规则2：

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

在管理员同意后，账单就会被删除。这个交易过程，会将用户的申请状态，变为完成状态或错误状态（被管理员拒绝或读取/存入数据库失败）。

证明规则3：

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。所以需要验证身份。

这里设计的验证身份就是“登录”。

实施规则1：

系统要维护关联关系，保证经过验证的TP操作相应的CDI

在用户提出存取款申请后，管理员同意，就代表该账单已经被验证。被验证的这个账单可以对数据库中，相应的存款金额进行更改。

实施规则2：

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

经过验证的账单，即管理员同意后，可以对数据库中的 CDI（即用户的存款金额）进行更改。

实施规则3：

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

满足责任分离原则，用户和管理员都不能单独对存款金额进行更改，只有用户申请，管理员同意后，才能进行操作。

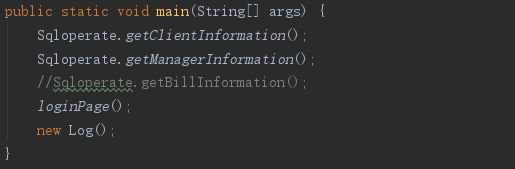
实施规则4：

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

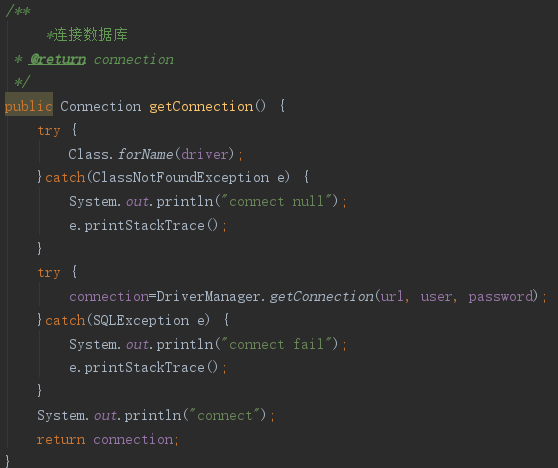
只有用户提出申请，才能进行 TP 操作。授权的管理员，没有执行 TP 操作的能力。

7.部分代码（完整代码见附件）

主函数：

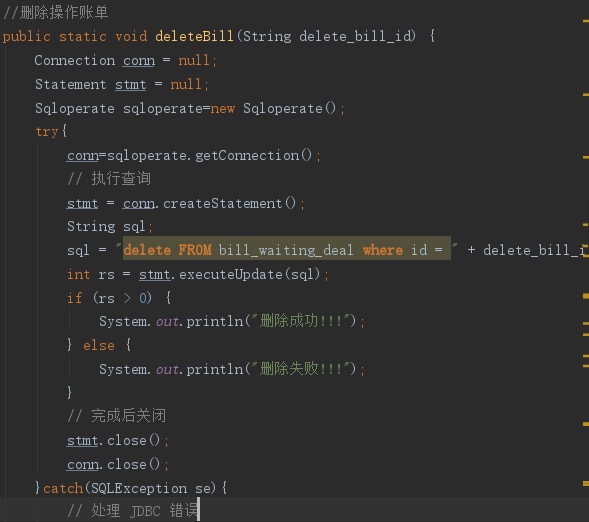


数据库的部分操作函数：

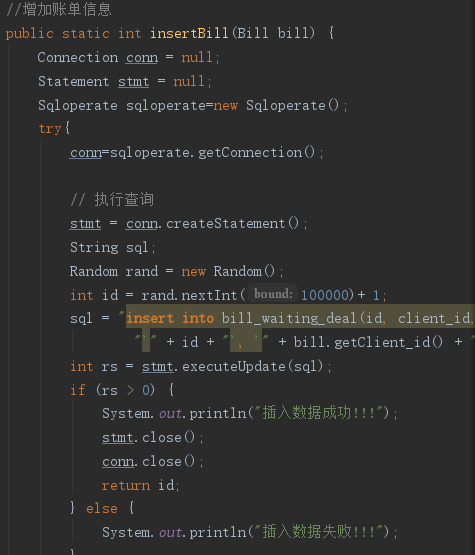




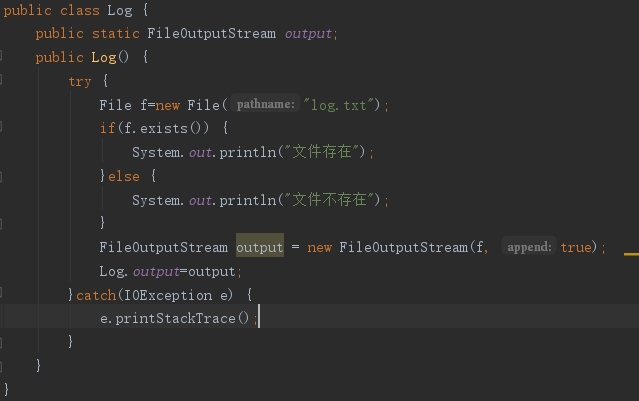
订单删除：



订单增加：



日志记录：



完整代码见附件

8.心得体会

通过本次实验首先进一步增强了编程能力，对项目的流程更加熟悉，并且对责任分离、功能分离原则，Clark-Wilson模型，定义应用系统的完整性限制条件等增强系统安全性的方法加深了了解，并在项目中尝试了运用，另外，通过对交互界面的书写，增强了对java的GUI操作的能力，总体来说，学习了很多。