COURSE PROJECT

1 INTRODUCTION

You have been hired to develop a prototype account management system for DebtsRUs State Bank. The system is to manage all savings, checking, and pocket accounts at the bank for customers of DebtsRUs. This includes the following tasks.

- Maintaining balance information for all customer accounts,
- Maintaining information on bank customers, the owners of the accounts,
- Processing transactions (deposits, withdrawals, payments, etc.),
- Generating monthly reports and updating accounts with monthly interest, and
- Providing a simulated ATM-App (Automated Teller Machine-Application) interface.

Your system is to be implemented within the Oracle DBMS environment using Java and the JDBC interface to the Oracle DBMS.

2 DETAILS

2.1 Accounts

An account is a repository of money owned by one or a set of customers. If there is more than one owner, one of them is designated as the *primary* owner (who, among other things, will receive monthly statements) and all other owners are *co-owners* who will have all owner's privilege except for receiving month statements. Associated with each account is a unique *account ID number* (an integer), and a list of transactions made during the month. Also associated with each account is a bank branch name in which the account is held.

Accounts come in three flavors: *checking*, *savings*, and *pocket*. There are two sub-flavors of checking accounts: *student* checking and *interest* checking. There is only one kind of savings accounts and one kind of pocket accounts.

The following rules apply to all accounts:

1. When a checking or savings account is first created, it must have a balance of at least $1,000 (this should be recorded in the transaction history as a deposit).
2. When a pocket account is created the customer must already have a checking or savings account with a positive balance. The customer also selects the account (checking or savings) to be linked to the pocket account.
3. No transaction can make an account balance to go below $0.00; every transaction that removes more than the available balance must fail.
4. Each transaction that makes a checking or savings account balance to $0.01 or less automatically *closes* the account. When an account is closed, the account is not removed from the database until after a final statement is generated (at the end of the month). No transactions (including deposits) are permitted on a closed account.
5. At the end of each month, all open accounts earn interest on their balances. The rate of interest varies with the type of account.

2.2 Checking Accounts

In addition to the general account rules, all checking accounts observe the following rules:

1. *Interest checking* has an initial 3.00% annual rate (for simplicity, the monthly interest rate is just the annual rate divided by 12) but *can be changed*. It should be simple to change these values in your prototype system (bank policies change quite often). *Student checking* earns no interests, i.e., has a fixed 0.00% annual rate.
2. The following transactions are valid on a checking account: *deposit*, *withdrawal*, *transfer*, *wire*, *write-check*, and *accrue-interest*.

2.3 Savings Accounts

Savings accounts observe the following rules in addition to the general account rules:

1. The initial annual interest rate is 4.80%.
2. The following transactions are valid on a savings account: *deposit*, *withdrawal*, *transfer*, *wire*, and *accrue-interest*. Note that no checks can be written for a savings account.

2.4 Pocket Accounts

Pocket accounts are used to make flexible payments (e.g., from cell phones) to vendors, or other customer's pocket accounts. The general account rules and the follow rules apply:

1. The interest rate is always 0.0%.
2. A flat $5 monthly fee is applied on the first transaction of the month. The monthly fee is waived if there are no transactions in the month,
3. The following transactions are valid on a savings account: *top-up*, *purchase*, *collect*, and *pay-friend*.

2.5 Customers

A customer is an individual with a name, a (unique) *tax identification number*, and an address. Every customer has a set of (jointly) owned accounts. A customer should be only kept in the system if he or she owns one or more accounts.

Also associated with each customer is a unique PIN (*Personal Identification Number*), which is a 4-digit string (i.e., leading 0 is acceptable). The PIN is used by the customer to access the accounts she/he owns using an ATM-App (interface). The PIN should be private data: it should not be possible to write a program (e.g., SQL) that reads the PIN value for a customer. Rather, only two functions should have access to an account's PIN, **VerifyPin**(*PIN*) and **SetPin**(*OldPIN*, *NewPIN*). The function **VerifyPin**(*PIN*) returns *true* if *PIN* is the correct PIN for the customer. **SetPin**(*OldPIN*, *NewPIN*) changes a customer's PIN to *NewPIN*, if *OldPIN* is the customer's current PIN. When a new customer is created in the system, the PIN is initialized to 1717.

2.6 Transactions

Transactions are actions that move money into and out of accounts, and from one account to another. A transaction can be generated by interaction of a customer with an ATM-App, or by an action taken by a bank teller.

The following transaction types are allowed in the system:

Deposit: Add money to the checking or savings account balance.

Top-Up: Move a specified amount of money from the linked checking/savings account to the pocket account.

Withdrawal: Subtract money from the checking or savings account balance.

Purchase: Subtract money from the pocket account balance.

Transfer: Subtract money from one savings or checking account and add it to another. A transfer can only occur between two accounts that have at least one owner in common. If the transfer was requested by a customer, she or he must be an owner of both accounts. Furthermore, the amount to be moved should not exceed $2,000.

Collect: Move a specified amount of money from the pocket account back to the linked checking/savings account, there will be a 3% fee assessed.

Pay-Friend: Move a specified amount of money from the pocket account to a specified customer's pocket account.

Wire: Subtract money from one savings or checking account and add it to another. The customer that requests this action must be an owner of the account from which the money is subtracted. There is a 2% fee for this action.

Write-Check: Subtract money from the checking account. Associated with a check transaction is a check number. (Note that a check cannot be written from all account types.)

Accrue-Interest: Add money to the checking or savings account. The amount added is the monthly interest rate times the average daily balance for the month (e.g., an account with balance $30 for 10 days and $60 for 20 days in a 30-day month has an average daily balance of $50, *not* $45!). Interest is added at the end of each month.

Associated with every transaction is the date of the transaction and the account(s) involved (in addition to any information specific to the transaction; e.g., check number). This information will be included in the monthly statement for each account.

Transactions may fail for various reasons. For example, a transaction fails if any of the accounts involved are closed or if more money is deducted than is available in the account.

All successful transactions on an account should be recorded for the account and printed in the monthly statement for the account.

2.7 ATM-App Interface

Your system needs to provide a simulated ATM-App (Automated Teller Machine or mobile application) interface. (For simplicity, these two interfaces are combined for this project.) The ATM-App interface should query for a PIN. If the PIN is successfully verified, the ATM-App should allow the customer to make any of the following transactions:

- Deposit, top-up,
- Withdrawal, purchase,
- Transfer (between accounts with a common owner only), collect,
- Wire, pay-friend.

If the customer owns more than one account, she should be prompted for the account(s) the transaction should access.

Optionally, you may want to include a "quick cash" option, which automatically withdraws some pre-selected amount(s) from a pre-selected account. If there is no pre-selected amount or account, no withdrawal should occur. Similarly, a "quick refill" option can be provided for pocket accounts.

2.8 Bank Teller Interface

The Bank Teller Interface allows bank employees to manage customer accounts. The following options should be available:

Enter Check Transaction: Submit a check transaction for an account.

Generate Monthly Statement: Given a customer, do the following for each account she owns (including accounts which have closed but have not been deleted): generate a list of all transactions which have occurred in the last month. This statement should list the names and addresses of all owners of the account. The initial and final account balance is to be included. If the sum of the balances of the accounts of which the customer is the primary owner exceeds $100,000, a message should be included in the statement to warn the customer that the limit of the insurance has been reached.

List Closed Accounts: Generate a list of all accounts which have closed in the last month.

Generate Government Drug and Tax Evasion Report (DTER): By federal law, deposits over $10,000 for a single customer in all (owned or jointly owned) accounts within one month must be reported to the government. Generate a list of all customers which have a sum of deposits, transfers and wires during the current month, over all owned accounts (active or closed), of over $10,000. (How to handle joint accounts?)

Customer Report: Generate a list of all accounts associated with a particular customer and indicate whether the accounts are open or closed.

Add Interest: For all open accounts, add the appropriate amount of monthly interest to the balance. There should be a record in your database that interest has been added this month. So a repeated "Add Interest" transaction would report a warning and do nothing else.

Create Account: Given an account type and other necessary information (e.g. owners, initial balance), create a new account with the specified characteristics. Note that this operation may introduce new customers to the bank. You may consider a create customer operation, but as far as the bank operations are concerned, customer creation is a part of account creation.

Delete Closed Accounts and Customers: Remove from the database all closed accounts and remove all customers who do not own any accounts (because their accounts have closed).

Delete Transactions: Delete the list of transactions from each of the accounts, in preparation for a new month of processing.

2.9 Set Dates and Interest Rates

The following operations should also be provided in your system. They may not a functional part of your system but they are needed to test and debug your system.

- Set a new date, you can assume that the new date is later than all dates recorded in the database, and
- Set a new interest rate for a given type of accounts.

You may assume that the bank is open every day.

3 REQUIREMENTS

Your prototype system should have user interface(s) for the ATM-App and Bank Teller interfaces. It is not necessary to have your interfaces accessible from a web browser, even though access through the web seems logical and desirable. In designing the GUIs of your system, keep in mind the principle of "simple" and "functional". You will not earn extra credits with fancy GUIs. On the contrary, if your system does not function as specified, you will lose points.

You must store all information of your system in a database system managed by the Oracle DBMS. That is when your system is not running or crashed or shutdown, all data are in the database and nothing is stored in any files. When your system restarts, all previous actions done on your system must be remembered. Again, you should not use files. Your system should be implemented in Java using JDBC to connect to your database managed by the Oracle DBMS.

The course project is to be completed by each group consisting of 2 students. Everyone in the group is expected to know all details of the implementation up to the level of being able to answer questions concerning design decisions.

4 EARLY PROJECT REPORT

Each group should submit an early project report by Monday, November 4. The report has to address the issues in making major design decisions. In particular, you should discuss the following points that will help understanding the requirements of the project and main steps towards completing the project.

1. Identify as many integrity constraints as you can on your initial ER diagram. You may describe the constraints in English.
2. Design an ER diagram for the application described in the project and express as many integrity constraints you have identified as possible.
3. Translate the ER diagram into relation schemas and do not forget the integrity constraints you have identified.
4. Indicate which integrity constraints that your relational database schema is able to incorporate; identify additional integrity constraints if possible.
5. Discuss briefly how you will deal with a violation of each of the integrity constraints identified.
6. Provide an initial system design. You can choose appropriate means to describe your system design, e.g., class diagrams, diagram of functional components, etc.
7. List the task divisions and list each member's responsibility.

Please typeset your report (you may draw figures by hand), a revised version of your report will be a part of the final project report.

5 Project Evaluation

Project evaluation of all teams will be conducted in the 10th week, i.e., during the week of December 2. Details will be announced later.