CS174A COURSE PROJECT

1 INTRODUCTION

You have been hired to develop a software system for the StarsRUs Brokerage. The brokerage allows its customers to buy and sell stocks of movie stars and directors in the market. In general, the stock price of a movie star or a director depends on the future potential of obtaining movie contracts by the actor or the director. A number of factors that affect the future potential include: past and current contracts, awards, reviews, total box office sales of the movies the actor/director has played/directed, and of course trading in the market. Your system should be able to manage all customer accounts and to support online trading and transactions by its customers. In particular, your system will provide the following functions:

- Maintain balance information for all customer accounts,
- Maintain information on customers,
- Process transactions (deposits, withdrawals, buy, sell, etc.),
- Generate monthly reports and updating accounts with monthly interest, and
- Provide an interface for online trading and transactions, accessing necessary information about movies, actors and directors, the market, and necessary trading methods.

Your system is to be implemented within the MySQL DBMS environment using Java and the JDBC interface to the DBMS. You have to demonstrate your system and the main functionality to the teaching staff of this course on CSIL computers (which run the Linux operating system).

2 DETAILS

2.1 Customer Profile

Each customer who wants to trade stocks on the StarsRUs system has to register on the system with the following information: *name*, *state* (a two-letter code for tax purposes), *phone number* (10 digits), *email address*, and a *tax identification number* (Tax ID). Each customer must also choose a username and password on the StarsRUs system for obtaining online services. The StarsRUs system must ensure that usernames are unique among all customers.

2.2 Accounts

An account is a repository of money or stocks owned by a customer. Associated with each account is a unique *ID number* (an integer), and a *list of transactions* made during the current month.

Accounts come in two main flavors: *Market* and *Stock*. Account balances can never go below 0; therefore any transaction that would result in a negative balance must fail. The account types are explained below.

Market accounts:

Each customer must have a market account with a positive balance before she can buy stocks. Your system may create a market account for each new customer when she registers. The customer must make an initial deposit of at least $1,000 immediately after the market account is opened. The brokerage pays a fixed 3% annual interest on the balance of the market account (see details in the transaction section). At the end of each month, each account earns interest on the balance. The following transactions are valid on a market account: *deposit*, *withdraw*, *accrue-interest*, *buy*, and *sell* (explained in the transaction section).

Stock accounts:

A stock account keeps the balance (in terms of shares) of each individual stock owned by one customer. A stock account for a customer is automatically created at the first time the customer buys that stock. A stock account should also keep track of the buying and selling prices for shares traded. Stock account balances do not have to be whole numbers, e.g., they can have fractions such as "110.715", "21.02", etc. However, it is not necessary to store more than 3 digits after the decimal point. The following transactions are valid: *buy* and *sell*.

2.3 Stocks

Each stock is associated with an actor (director) and each actor (director) has only one stock. Each stock has a three-letter symbol. The following information is stored for every stock: the daily closing prices and the current price.

2.4 Actor (Director) Profiles

The system also stores for each actor (or a director) a profile that includes: the *actor name*, the *stock symbol*, the *date of birth*, and for each *movie contract* signed by the actor/director, the *movie title*, *role* (actor or director or both), *year*, and the *total value* (payment to the actor/director) of the contract.

2.5 Movies

Your system does not manage (store) the movie information. Instead, your system will access the movie information external to your system. Movie information includes: title, production year, reviews, ranking, etc. of each movie. Details of the information about accessing movie information will be available at a later time.

2.6 Transactions

Transactions are actions that move money into and out of market accounts, and stock shares into and out of stock accounts. A transaction can be generated by interaction with a customer through the online trade interface, or by an action taken by a manager in the brokerage.

The following types of transactions should be suported by your StarsRUs system:

Deposit:

Add money to the market account balance.

Withdraw:

Subtract money from the market account balance.

Buy:

Acquire a specified number of shares of a specified stock at the current price. Each buy transaction costs $20 (commission). The customer must pay the total price for the shares and the commission with the money in the market account. (The transaction fails if there is not enough money in the market account.)

Sell:

Sell a specified number of shares of a specified stock at the current price. For tax purposes, the customer has to specify the original buying prices of the stocks to be sold so that the system can compute "earnings".

For example, a customer owns 100 shares originally bought at $20 per share and 250 shares acquired at $25.50 per share. If the customer sells, at the current price of $24 per share, 50 shares from the 100 shares, and 100 shares from the 250 shares, the customer will make $(24 - 20) \times 50 + (24 - 25.5) \times 100 = 50$ dollars.

Similar to buy transactions, each sell transaction costs $20 of commission. The money from selling the stock will be deposited into the market account.

Accrue-Interest:

Add money to the market 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 the simple average of $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., stock symbol, number of shares, and price etc.). This information will be included in the monthly statement for each account.

Transactions may fail for various reasons. For example, a transaction fails if it results in a negative balance of an account. All successful transactions on an account should be recorded for the account and printed in the monthly statement for the account.

2.7 Trader Interface

Your system should provide an *Trader Interface*. The interface allows a new customer to register. After the registration process is successfully completed, the customer must login into your system with a valid username and a correct password. If the login is successful, the interface should allow the customer to make the following transactions and queries:

- Deposit,
- Withdrawal,
- Buy,
- Sell,
- Show balance for his/her market account,
- Show the transaction history for his/her stock account,
- List current price of a stock and the actor profile, and
- List movie information, which shows the detailed information of a given movie (obtained from an external source). Your system should display the information in a user-friendly manner. In addition, your system should also be able to provide only the following information upon request:

  Top movies: Given a specified time interval (for example, 1970-1980), list the titles of the movies that were rated 5 stars by some (any) organization.

  Reviews: Display all the reviews for a given movie.

2.8 Manager Interface

The Manager Interface allows StarsRUs employees to manage customer accounts. The following options should be available:

Add Interest:

For all market accounts, add the appropriate amount of monthly interest to the balance. This is usually done at the end of a month.

Generate Monthly Statement:

Given a customer, do the following for each account she/he owns: generate a list of all transactions that have occurred in the current month. This statement should list the name and email address of the customer.

The initial and final account balance is to be included, so are the total earning/loss (including interest) this month and the total amount of commissions paid.

The statement will be displayed in your interface.

List Active Customers:

Generate a list of all customers who have traded (buy or sell) at least 1,000 shares in the current month.

Generate Government Drug & Tax Evasion Report (DTER):

According to the law, each customer who earns more than $10,000 within one month must be reported to the government. Generate a list of all customers who have made more than $10,000 in the last month, including earnings from buying/selling stocks and interest. The residence state of each customer should also be listed.

Customer Report:

Generate a list of all accounts associated with a particular customer and the current balance.

Delete Transactions:

Delete the list of transactions from each of the accounts, in preparation for a new month of processing. (Actually they will be archived, but we will not worry about it in this project.)

2.9 Test, Debug, and Demo Operations

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

- Open market for the day,
- Close market for the day,
- Set a new price for a stock, and
- Set a new date to be today's date.

You may choose particular ways these operations are done and include the operations in the Manager Interface or in a separate interface. You can assume that the market is open every day. Note that your system should not use the system date (time), and also, the current date and time information should be stored in your database so that if your system shuts down and restarts, it will resume from the stored current date and time.

3 REQUIREMENTS

Your prototype should have functional user interface(s) for the online trade and manager interfaces. It is not necessary to have your interfaces accessible from the web and/or a mobile device. In designing the interface(s) of your system, keep in mind the principle of being "simple" and "functional".

You must store all information of your system in a database system managed by the MySQL DBMS. That means when your system is not running, all data are in the database and *nothing is stored in any files*. During the demo, your system may need to shut down and restart and all previously completed operations in your system must have their effects recorded in your system. Your system should be implemented in Java using JDBC to connect to your database.

The course project is to be completed by each group consisting of 1 to 2 students. Although both the difficulty and amount of work are suitable for either one or two students, it is strongly encouraged that you work with a partner as a group. In case of completion by two 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 MILESTONES

There are two milestones for the project:

- An early project design report (see Section 5), and
- A demonstration of your system to the teaching staff of this course.

The project deadline is at your scheduled demo. The regular demos will be arranged during the last week of instruction (the 10th week, i.e., the week of December 4 to 8).

As an option, demos can be arranged during the 9th week, (typically the second half). Each such "early demo" will be given a 10% extra credit (of the total project score). You have to choose in advance between regular and early demos and no groups will be allowed to have two demos. The details about the demos will be announced at a later time.

5 EARLY PROJECT REPORT

Each group should submit an early project report. The report has to address the issues concerning 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 (the 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. Draw a functional architecture of your planned overall system design. You may use a class design (diagram) along with a brief explanation of what each class is expected to do.
7. List the task divisions and 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. (You may also want to make a copy of your ER diagram so you can continue your project after submitting the early report.)

Early project report should be submitted no later than Monday, November 6.

6 DEMO AND WHAT TO TURN IN

At the demo time, you will turn in a revised design report (hardcopy), a listing of all JAVA source code (electronic), schema definitions (hardcopy), and SQL queries (hardcopy) used in your program. In addition, you will give a half-hour demo of your system. We will provide a sample dataset (including users, devices, and a list of transactions). This data should be entered into your database before your demo (by you).