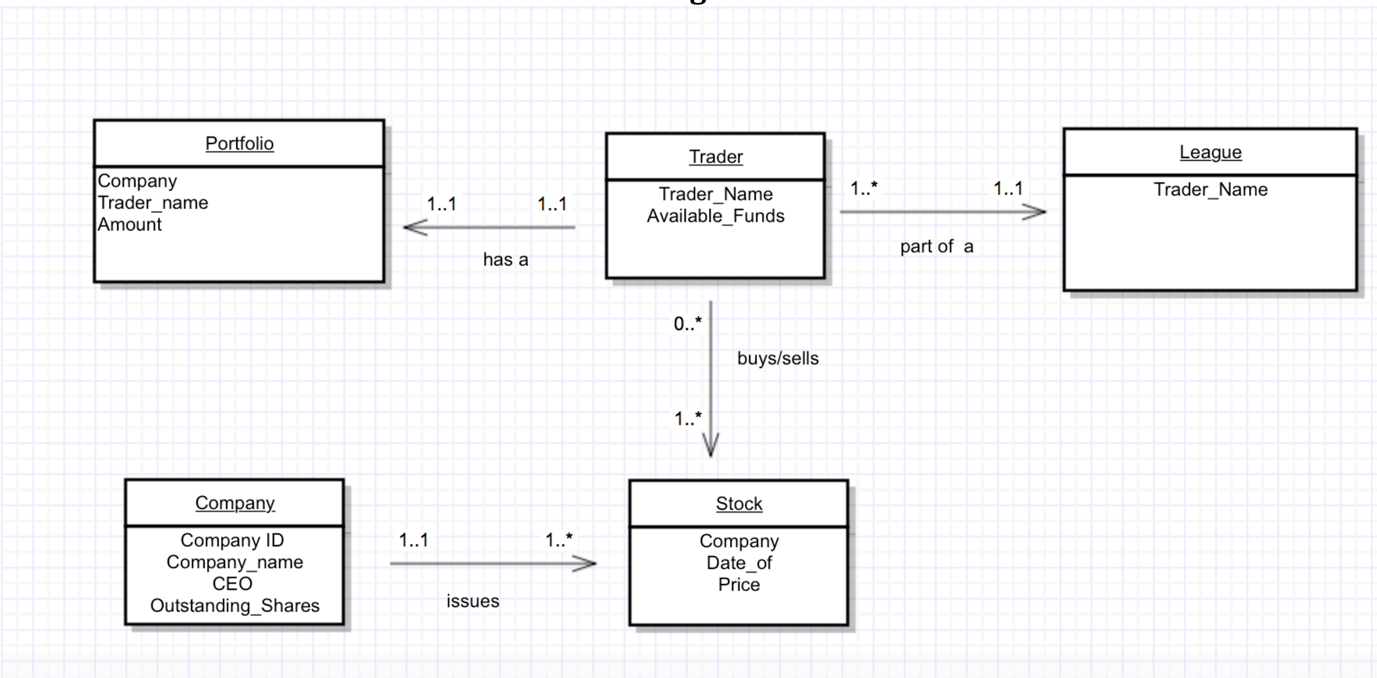
George Abinader, Robert Carney, Prannoy Mohan

FINAL PROJECT README:

There are two parts to this project, the stock market jar (front end) and the SQL file that contains the dump of the stock market database (back end). The user should have MySQL installed and then either run the script via MySQL, or run MySQL through the terminal and run the SQL script to import the database with procedures and functions.

The stock market jar should be run from the terminal application. Locate the directory that the jar was saved to. Once you are in that same directory use the command <java –jar CS3200-PROJECT.jar> to run the program. Make sure that you have at least JAVA SE Development Kit 8 and above.

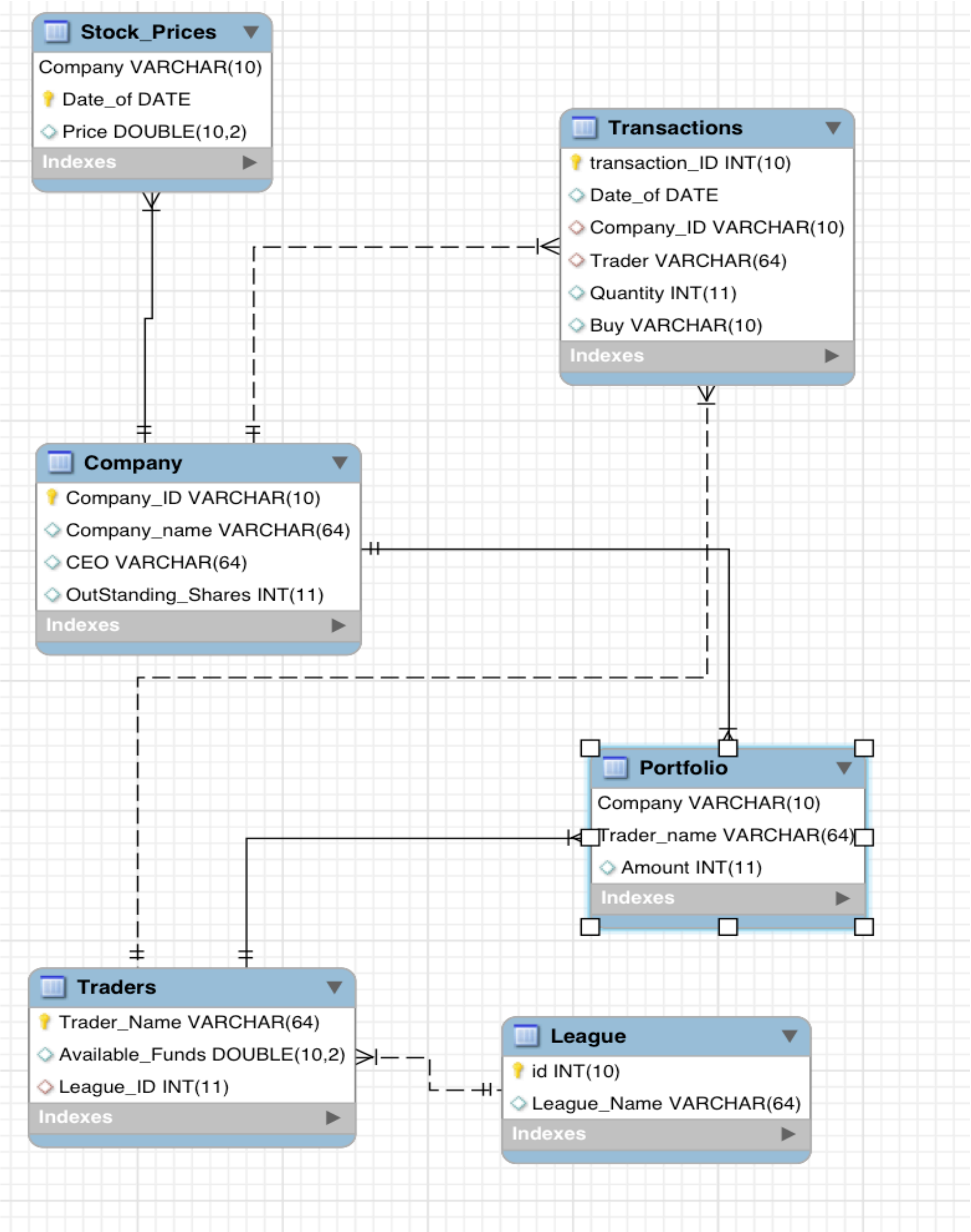
UML diagram:



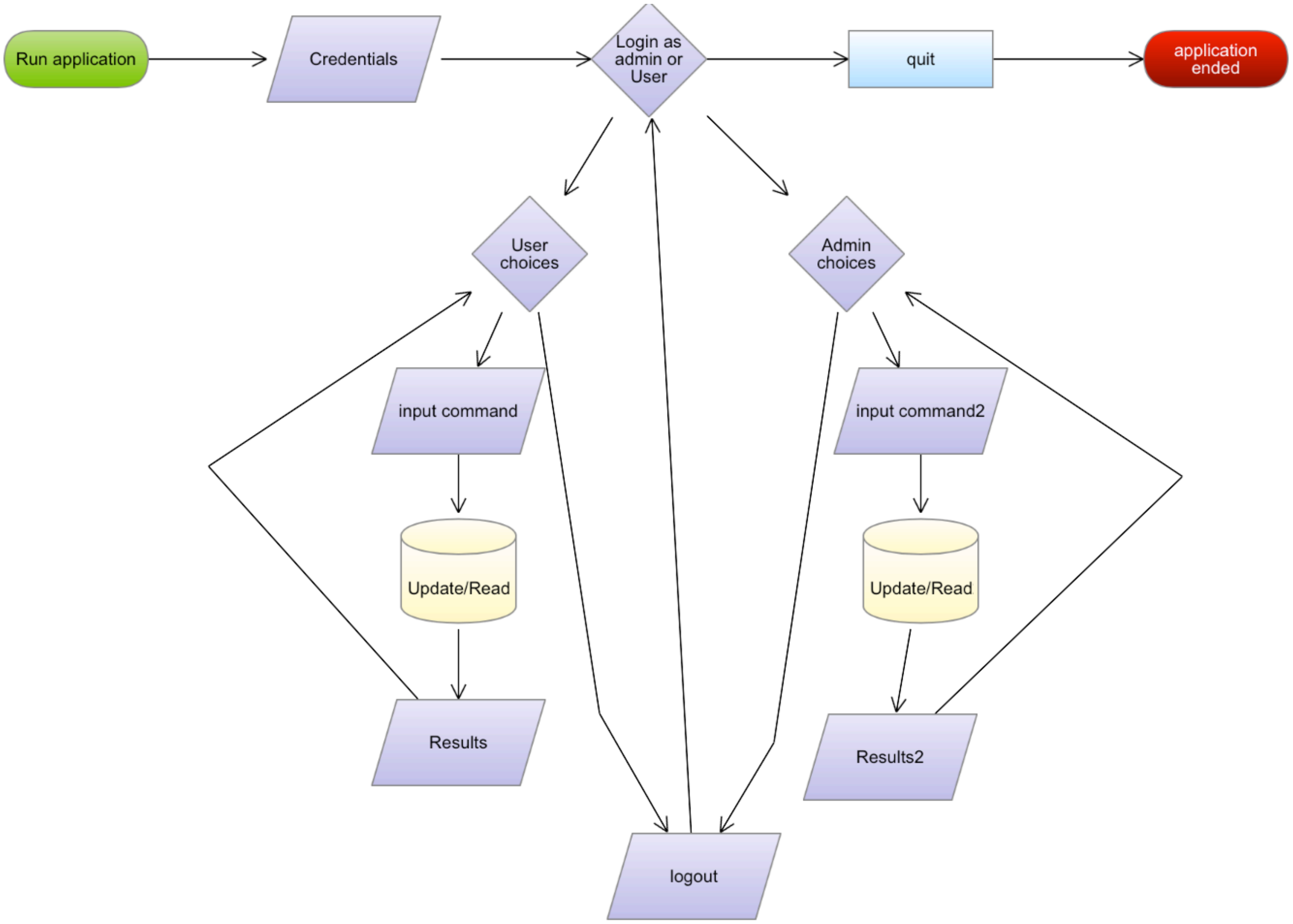
A trader has a portfolio which contains tuples pertaining to how many stocks in a certain company that trader owns (if he or she has previously owned stock in that company). The primary key is the company, and trader name (composite key). A trader is part of a league, and a league must contain at least one trader. A company can issue one or more of its stock, each stock is issued by only one company.

EER diagram:

To convert from the UML to EER, we made the league name a field of the trader, and kept track of how many stocks a trader buy or sells within a transactions table.



Client Interaction:



The client first starts by running the program within their terminal. The client is then prompted for credentials for the database. When valid credentials are entered, the client is prompted by the program asking whether they choose to login as an admin or user. Based on the choice of user type, a client will have different choices. At any prompt for a command regardless of the user type, entering ‘help’ will display all available commands.

The choices for user are:

[bu] = Buy stock. The user must then specify what stock and for what amount. The procedure checks if the user has enough money to buy before committing the change. A new entry is made in the transactions table if the command is successful.
[se] = sell stock. The user must then specify what stock and what amount. The procedure checks if the user has enough stock to sell before committing the change. A new entry is made in the transactions table if the command is successful.
[st] = displays in descending order who is worth the most money by totaling their funds remaining and the value of the current stock they hold.
[in] = displays the traders portfolio, with how many of which stock they own
[pr] = displays the current cost of all stocks
[ab] = returns how much money the trader has left to spend
[lo] = logs you out
[help] = displays all the available commands above to choose from.

If a user enters a command not found above, nothing will execute, the user will be shown a helpful message and prompted to try again.

The choices for admin are:

[ne] = create a new league. Will ask for traders to populate it, one at a time.
[up] = pulls the current real time prices for the stocks from google finance and updates the current prices within the database.
[re] = resets the trader table by deleting all traders.
[pl] = plays the stock market game as a trader.
[de] = asks for a trader to delete. Deletes all records that refer to that trader.
[vl] = displays the leagues within the database.
[vt] = displays all the traders in the database and their respective leagues.
[pr] = displays the current rates of stock available.
[cl] = displays the transaction log for all traders in the database.
[dl] = deletes a league from the database and all traders in that league.
[help] = displays all the available commands above to choose from.

If an admin enters a command not found above, nothing will execute, the admin will be shown a helpful message and prompted to try again.

CRUD Operations:

Admin

Create: create league
Read: view league, view traders, price, check log
Update: update
Delete: reset, delete trader, delete league

User

Create: buy stock, sell stock
Read: standings, inventory, price, available balance
Update: [empty set]
Delete: [empty set]

Lessons Learned:

We initially planned very ambitiously to implement a graphical user interface or use a web interface, but that required knowledge that we did not have and could not learn quickly enough. We learned how to write a java program that interacts with a database integrating the use of SQL commands, queries, procedures, and functions. We also learned how to use existing APIs for a SQL database. All functionality is implemented and working.

Future Work:

We would like to implement nicer user interface such as a web interface. This program is a prototype so we would include password security when logging in as a trader or admin. We would also implement better security as many of the SQL statements are written within the front end and that enables our database to be susceptible to SQL injections. We would also like to switch to a different DBMS with better scalability. We would also like to implement other functions specific to stock such as options or shorting a stock, this would make the program even more realistic and provide more variety.