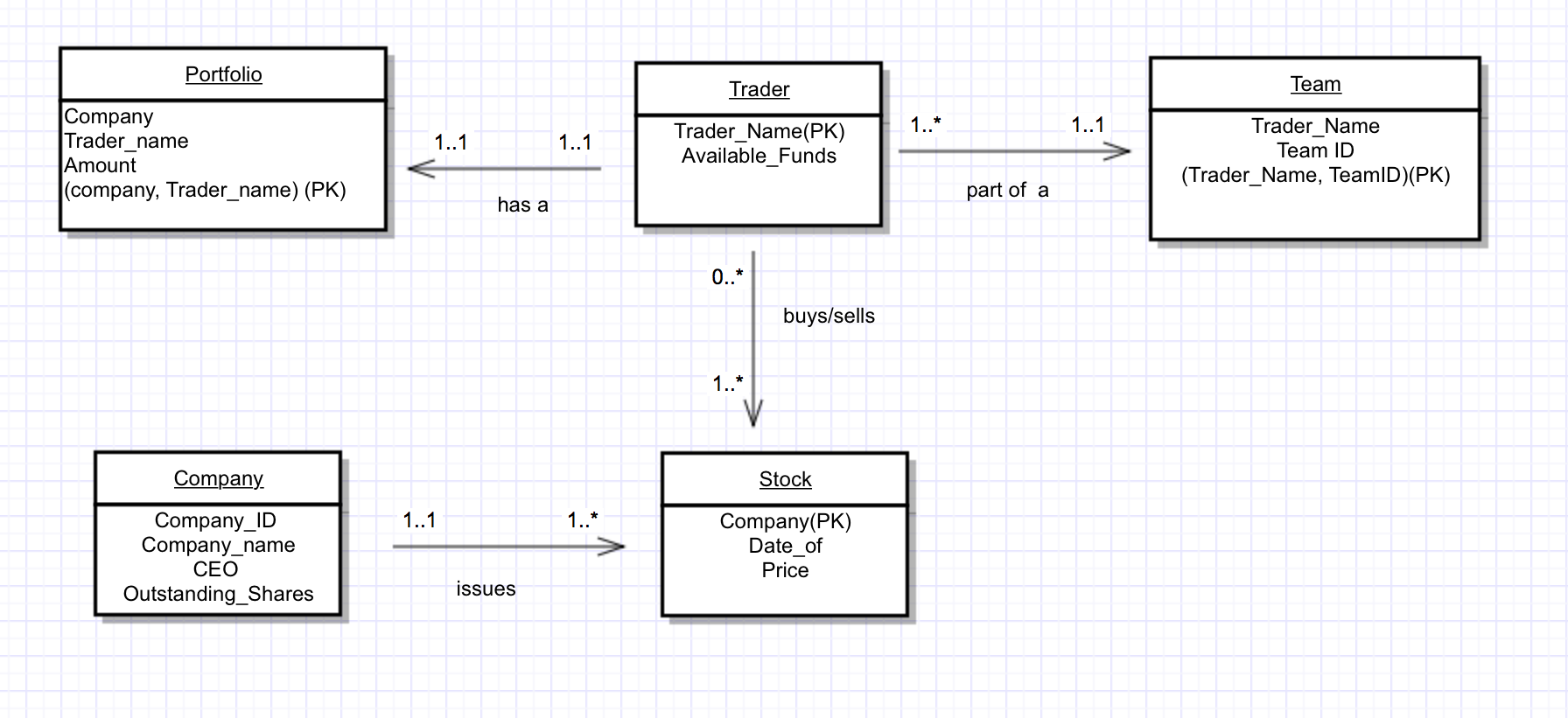
George Abinader, Robert Carney, Prannoy Mohan

README:

There are two parts to this project, the stockmarket jar and the sql file that contains the exported database. The user should mysql installed and then either run the script via MySQL Workbench or other GUI they use, or their terminal application. The stock market jar should be ran 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.

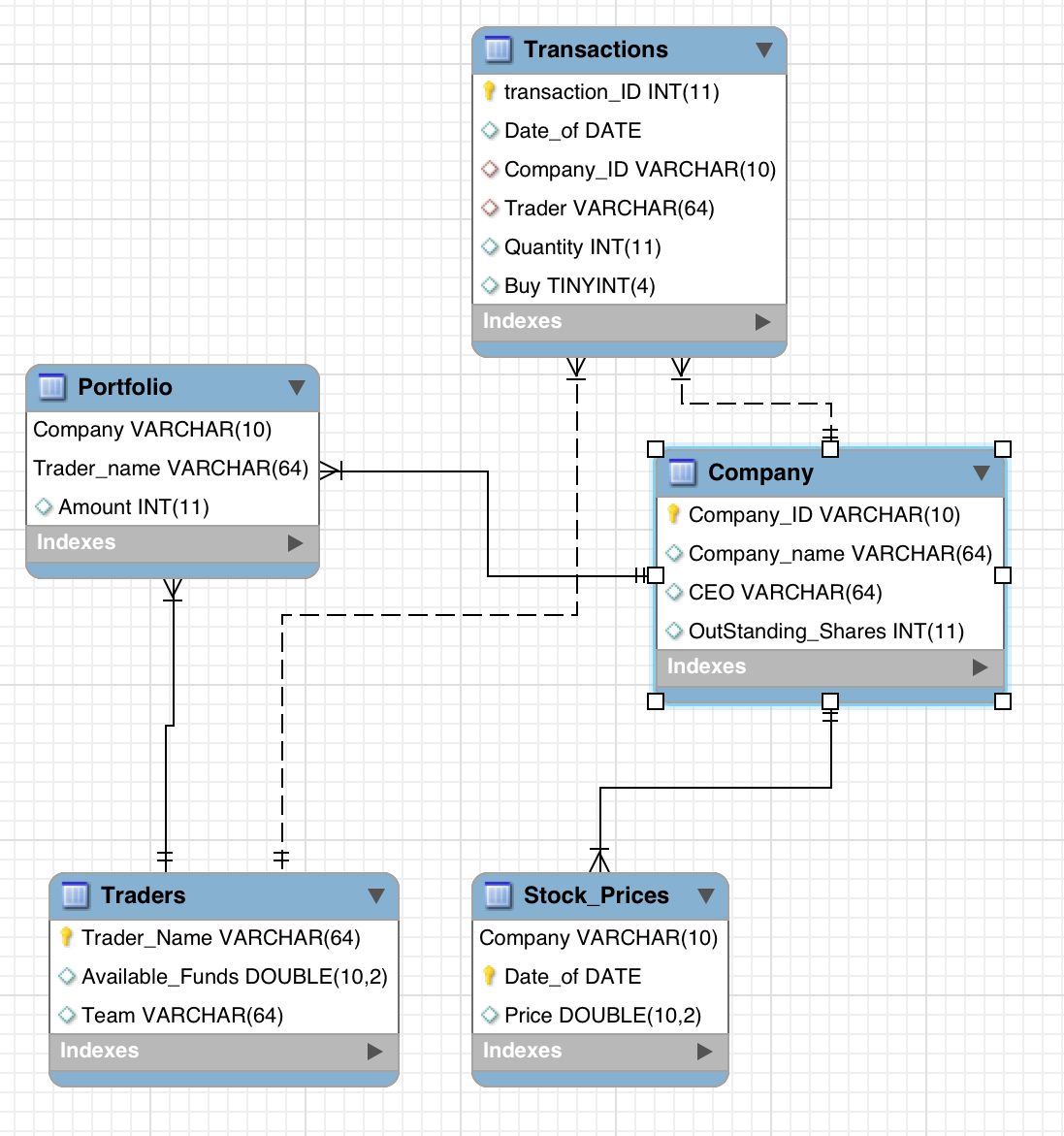
UML diagram



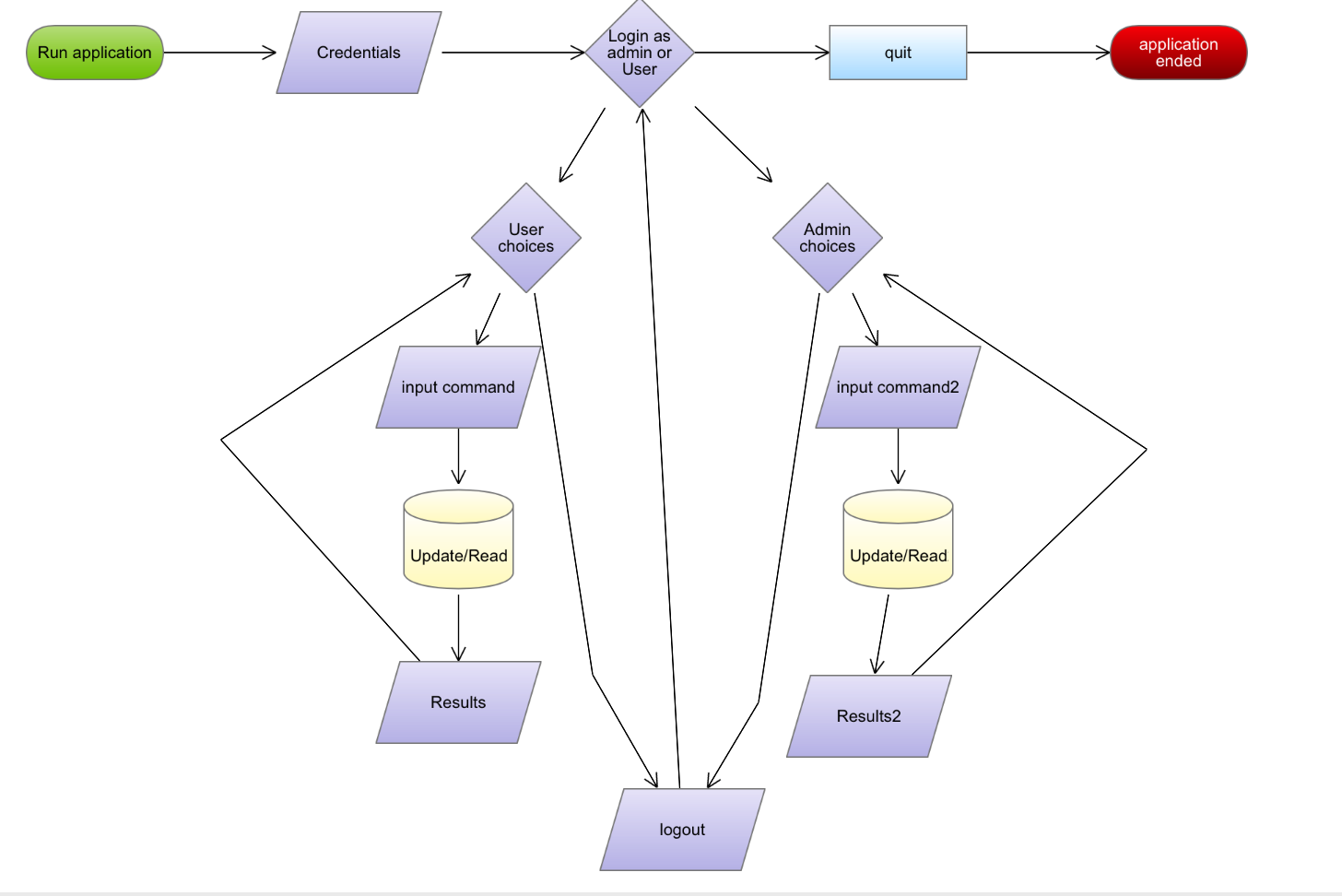
A trader has a portfolio where all the stocks he currently owns are kept along with its amount. The private key is the company, and trader name. A trader buys or sells 1 or more stock, and a stock doesn’t have to be owned by any trader or it could belong to many traders. A trader is part of one team, and a team must have at least trader. A company can issue 1 or more its stock, each stock is issued by only 1 company.

EER diagram

To convert from the UML to EER we made the team name a field of the trader, and kept track of how many stocks a trader buy/sells within a transactions table. The number of stocks a company issues/ are available is kept track as the field as outstanding shares. The transaction table is also kept so that the user can know how much the user bought or sold a stock for.



User Interaction



The user first starts by running the program within their terminal. They are then prompted to credentials for the data base. Until they enter valid credentials they will not be able to move onto the decision on whether they choose to login as an admin or user (trader). Based on if the person is an admin or user they will have different choices. At any type to view the choices the person has they just have to help into the console.

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

[lo] = logs you out

[ab] = returns how much money the user/ trader has left to spend

[cl] = displays all of the past transactions you have made

[help] = displays all of the commands above to choose from.

If the user enters a command not found above nothing will execute, will wait for the user to enter a valid command, and say enter help to view options.

The choices for admin are:

[ne] = create a new team. Will ask for a player and then y/n to confirm and then for done if the admin is done entering players, or for the next name

[up] = pulls the current realtime 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 and the admin is able to acess all the data of the trader and enter the commands for a user mentioned above.

[de] = asks for a trader/user to delete. Deletes all records of that trader.

[vl] = displays the teams within the database.

[vt] = displays all the traders in the database.

[pr] = displays the current rates of stock available

[cl] = displays the transaction log for all traders

[help] = displays all the commands above

If a user enters an invalid command not mentioned above, nothing happens and the console returns a warning message saying no such command exists and press help to view commands.

The user/admin can press command/ctrl c anytime to end the application and then start it by entering java –jar CS3200-PROJECT.jar

Lessons Learned:

We initially planned very ambitiously to implement a gui/ or use a web interface, but that required knowledge that we did not have and could not learn quickly enough.

Future Work:

We would like to implement better UI such as maybe a web-interface. We would also implement better security as some of the sql statements are written within the front end and that enables our database susceptible to sql injections. We would also like to switch to a DBMS for more scalability and for wider access, so that multiple users can login at the same time and issue their own transactions.