Eric Mercer & David Liu
November 15, 2012
Professor Yang
CompSci 316

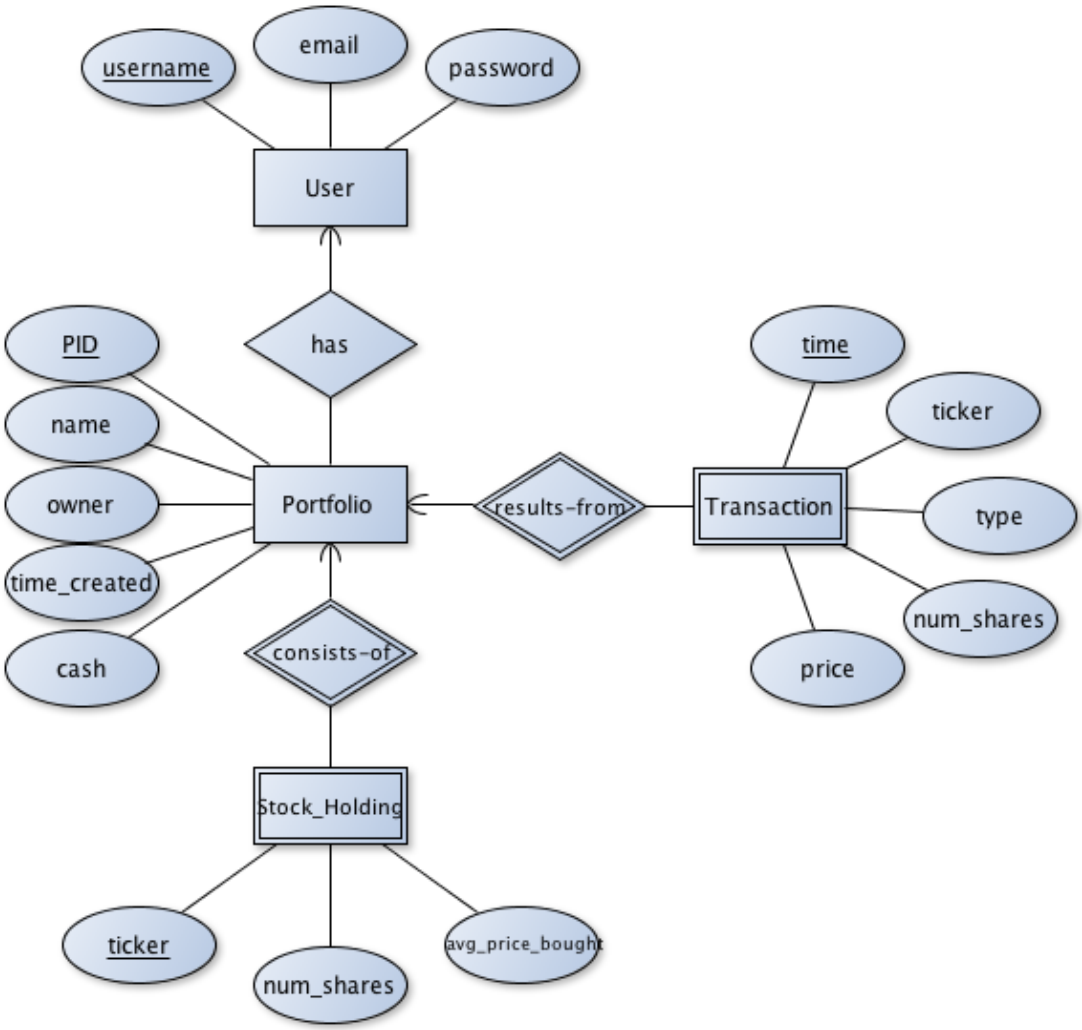
StockSim
Milestone 2

New Data Assumptions

No new assumptions.

Updated E/R Diagram

A `time_created` attribute was added to the `Portfolio` entity. Otherwise no changes were made beyond improving the names of a few attributes.



Updated Database Tables

```
User(username, email, password)
Portfolio(PID, name, owner, time_created, cash)
Stock_Holding(PID, ticker, num_shares, avg_price_bought)
Transaction(PID, time, ticker, type, num_shares, price)
```

*Note: avg_price_obught is well defined since it is only updated upon purchasing new shares of a stock. It is defined to be the weighted average of the price of the shares bought, and represents the value at which you would break even if you sold all your shares at a single price. The value is not changed upon selling any shares.

Platform Description

The StockSim platform uses a PostgreSQL database accessed via JDBC. In addition, Java is used as a host language for passing information between the top and bottom layers of the stack and processing the results of database queries and Yahoo! Finance queries. The website is built using a mixture of HTML, CSS, and JSP and is hosted on a Tomcat server.

Database Performance Tuning

No performance tuning seemed necessary as the most intensive database query is calculating the market value of all portfolios to determine the leaderboard. If we decide to only update the leaderboards every few hours or so, then it's not a query that we frequently need to run, and we can fully determine when it is run.

Other database queries all run very quickly so no tuning changes were made.

We may run into performance issues when querying Yahoo Finance in the future, but our current features don't require large queries from Yahoo and consequently there have been no performance issues so far.

Tasks Completed:

Tasks 1-8 have been completed and significant progress has been made on Task 9, including the creation of a user account, creation of portfolios, an overview of all portfolios owned by a user, and an in-depth view of the holdings of each portfolio and its market value.

Tasks to be Completed:

Implementing additional features including executing transactions, viewing transaction history, searching for stocks by ticker, and viewing a leaderboard.

Testing these additional features with a large dataset and tuning any performance issues.

Polishing the web interface.