

CSCI 275, Deliverable 1

ENTITY TYPES:

1. User
 - a. Name
 - i. First name
 - ii. Middle Initial (or name)
 - iii. Last Name
 - b. Address
 - i. Street
 - ii. Zip code
 - iii. City
 - iv. State / Province
 - v. Country
 - c. SSN or SIN number
 - d. Identification
 - i. Driver's license verification
 - ii. Government ID verification
2. Securities
 - a. Bonds
 - b. Stocks
 - c. Options
 - i. Call
 - ii. Put
 - d. Futures
 - e. (all of these will have historical data as an attribute)
3. Order
 - a. Timestamp
 - b. Order size
 - c. Security Identifier (ticker)
 - d. Order type
 - i. Stock
 1. Long / short
 2. Market
 3. Limit
 4. Stop order (stop market order)
 5. Stop-limit order
 6. Trailing stop order (\$ and %)
 7. Conditional
 8. All or none

- ii. Option
 - 1. Call
 - 2. Put

4. Order Fulfillment Venue

a. Exchange

- i. Exchange Name (e.g. NYSE, Nasdaq, TSX, ASE, etc)
- ii. Exchange Type (interest rates, petroleum products, currencies , metals, agriculture, stock indices)
- iii. ASE
- iv. Electronic communications Network
- v. OTC
- vi. Current Rate
- vii. Share code
- viii. Currency ID
- ix. Darkpool
- x. Trading curb

b. ATS (Alternative trading system)

i. Dark pool

- 1. Type
 - i. N/A
 - ii. Broker/Dealer Owned
 - iii. Agency Broker, Or Exchanged Owned
 - iv. Electronic Market Makers
- b.Target Clients
- c. Order types
 - i. Long / Short
 - ii.Limit
- d. Pricing / Matching mechanisms
- e. Business / Revenue model
- f. Ownership structure

ii. ECN (Electronic Communications Network)

5. (Block Chain???) (Note: we may or may not continue researching blockchain)

a.Stocks

- i. BTCS
- ii. BTL group
- iii. DigitalX
- iv. 360 blockchain

- b. Shared ledger
- c. consensus algorithm
- d. Node application

6. Order Management System

- a. Vendors (Purchasing and Receiving)
- b. Order entry(sales order, quotes, credit memos)
- c. Marketing information(catalogs, promotions, pricing)
- d. Customer information(Names , addresses, order history)
- e. Product information(descriptions, inventory information)

<http://ezinearticles.com/?Features-of-an-Order-Management-System&id=1419622>

7. Account

- Account number
- Account Balance
- Account Type (Roth IRA, IRA, Standard Brokerage, etc.)
- Account name
- Access level
- Account PIN/password
- Account activity
- Status (Frozen account vs. Active account)

Project Overview

We will strive to create a rudimentary DBMS akin to that of a stock broker or cryptocurrency exchange (e.g. ETrade and Coinbase, respectively). The proposed application involves developing a GUI which allows the end-user to acquire financial securities (stocks or bonds, potentially derivatives and cryptoassets), lookup historical asset pricing data and manage their account. We will need to develop a thorough understanding of how stock brokers bridge the gap between investors and the financial markets.

The application we intend to develop will be very similar to a platform like Etrade Pro, which includes an interface for users to place orders, manage their assets and analyze the market. The application will interact with the underlying DBMS in numerous ways. Whenever a client clicks the “Buy” or “Sell” button in the GUI, the order details will need to propagate to the order management system, whereby the order will be processed. When a buy order is placed by a user, several constraints must be enforced (e.g. ensuring the user has a sufficient cash balance in their account to finance the purchase). If the constraints pertaining to a user’s account are met, then the appropriate quantity of cash should be deducted from the account balance once the securities for the order have been obtained. There will also need to be various constraints on each account (e.g. only the user can withdraw / deposit cash into their account). After an

order is placed and has been parsed by the OMS, the OMS will analyze the markets and decide how to fulfill the client's order.

When a stock order is placed, a broker can fulfill it in a variety of ways:

1. Order to the floor
 - They can have a human being execute the order on the floor of an exchange, say, the NYSE.
2. Internalize
 - The stock broker can internalize the order, that is, trade directly with the client by exchanging cash or the securities involved for that order.
3. Order to a third market maker
 - There exist 3rd party security dealers that are also willing to buy or sell stocks listed on exchanges at the publicly quoted prices. The 3rd party market makers act as the buyer when an investor is looking to sell securities, and he/she (the investor) is simply looking to make a small, short-term profit.
4. ATS (Alternative Trading System)
 - An ATS is a venue for matching buyers and sellers that is not regulated as an exchange. Legally, ATSs are registered as a broker-dealer.
 - Often times, institutional investors will utilize an ATS to fill large orders opposed to exposing their intentions on a large, public exchange.
 - ECN (Electronic Communications Network)
 - i. An ECN is a type of ATS (alternative trading system) that autonomously matches buyers and sellers of a financial instrument.
 - ii. Orders are publicly displayed in the consolidated quote stream
 - iii. ECNs must be FINRA members and registered with the SEC as a broker-dealer
 - iv. The primary benefit of ECNs is that they eliminate the need for a third-party to facilitate the trading of financial securities. This allows for greater global liquidity and for more after-hours trading between investors.
 - Dark pools
 - i. Dark pools are another type of ATS that act as a private exchange, allowing subscribers to trade securities without publicly sharing the details of their order (e.g. order size) until after the transaction has already occurred. Because of this, there is no order book, or window, into a dark pool to assess the supply and demand of securities. Transaction details are disclosed subsequent to the completion of the transaction after a specified delay.
 - ii. Dark pools provide large banks and institutional investors alike an advantage when trading with large orders because the lack of public knowledge about their order precludes price fluctuations that would

otherwise occur. For instance, if J.P. Morgan publicly announces they are selling 10 million shares of AAPL at price XYZ, the price of AAPL will decrease and it will be more difficult to fill those 10 million shares at XYZ. However, if J.P. Morgan utilizes a dark pool, the order information will not be disclosed until they have found adequate buyers and sellers to fill the order. This facilitates filling large order sizes.

iii. Types:

1. Broker-dealer
2. Agency Broker or Exchange Owned
3. Electronic Market Makers

5. Order to market maker

- Market makers consist of broker/dealer firms (usually large institutions) that assume the risk of holding a certain number of shares of security to facilitate the trading of said security.
- Order to market makers, occur when investors take advantage of the purchase & sale solutions from these firms to keep financial markets liquid
- Eg: (standard online brokerage firm, like Charles Schwab that allows for the filling of security orders quickly and efficiently.)

Order Management System:

The order management system (OMS) is the lynchpin for brokerage firms when it comes to fulfilling their client's orders. An OMS manages and processes transactions through something called the FIX protocol. It begins by accepting orders, parsing the details of the order (e.g. the ticker, order size, order type, etc.) and then quickly analyzing the market to determine if the order can be executed immediately, and if so, where it can be filled. Financial instruments are traded on dozens of different mediums, each of which having different prices, incentives and liquidities for each instrument.

Developing a virtual OMS will be challenging because it must analyze the pricing and liquidity of the desired asset on numerous exchanges and acquire it within the constraints of the order type. For instance, when processing a limit order, the OMS must attain the asset at a price equal to or better than the limit price specified in the order in addition to determining which exchange has the best price and sufficient liquidity to fill the order.

Investors can disguise the size of their order, which poses another problem for the OMS we have not discovered the solution to yet. How does an OMS deal with hidden order sizes? We will also need to research the technical details as to what protocol is used for different order types (i.e., we need to determine exactly how OMS treat market, limit, stop orders, etc.). We have discovered an API which provides real-time level 1 data from the IEX and Nasdaq exchanges. We are still in search of a viable level 2 API to more accurately simulate the markets for our OMS. A level 2 data stream would provide us with more information regarding the current

state of the market as it contains many bids and asks with their respective order sizes. We plan to utilize the live data feed to replicate the many alternatives the brokerage firm has when processing an order. The OMS will determine which security to look for, what price to execute at and so on upon receiving an order from a user. It will then analyze the various order fulfillment venues to ascertain where and under what conditions the desired security can be obtained. The OMS then purchases the requested security and makes the appropriate changes to the user's account.

Obstacles:

- None of us have a comprehensive understanding of how stock brokers route and fill orders from their clients. This means we need to conduct research on the systematic way in which OMSs analyze market data and process orders.
- We would like to integrate live market data into our application. This will involve utilizing an API from a stock exchange or a financial data provider and necessitate the development of a system that dynamically analyzes current market conditions.
- We are unsure of the method we will employ to store historical asset data as new data is generated continuously when the markets are open. In theory, the amount of data we store will increase every day.

FIX: Financial Information Exchange Protocol

- FIX protocol was originally developed for equity trading between Solomon Brothers and Fidelity.
- FIX protocol is a language utilized to efficiently transmit financial data pertaining to stocks, bonds, options and futures between parties
- <https://www.fixtrading.org/implementation-guide/>

Small Order Execution System (SOES)

- May be advantageous to conduct more research on this system as it is used for order splitting which is essential for institutional investors.

Data feed:

We will attempt to simulate the various choices a stock broker has to fulfill a client's order by considering live stock market data from one or more exchanges. We have discovered a few APIs that we will attempt to integrate into our project with their respective SDKs.

There is a Yahoo Finance API (YQL) we can potentially utilize to integrate pseudo-live data from the markets with our DBMS

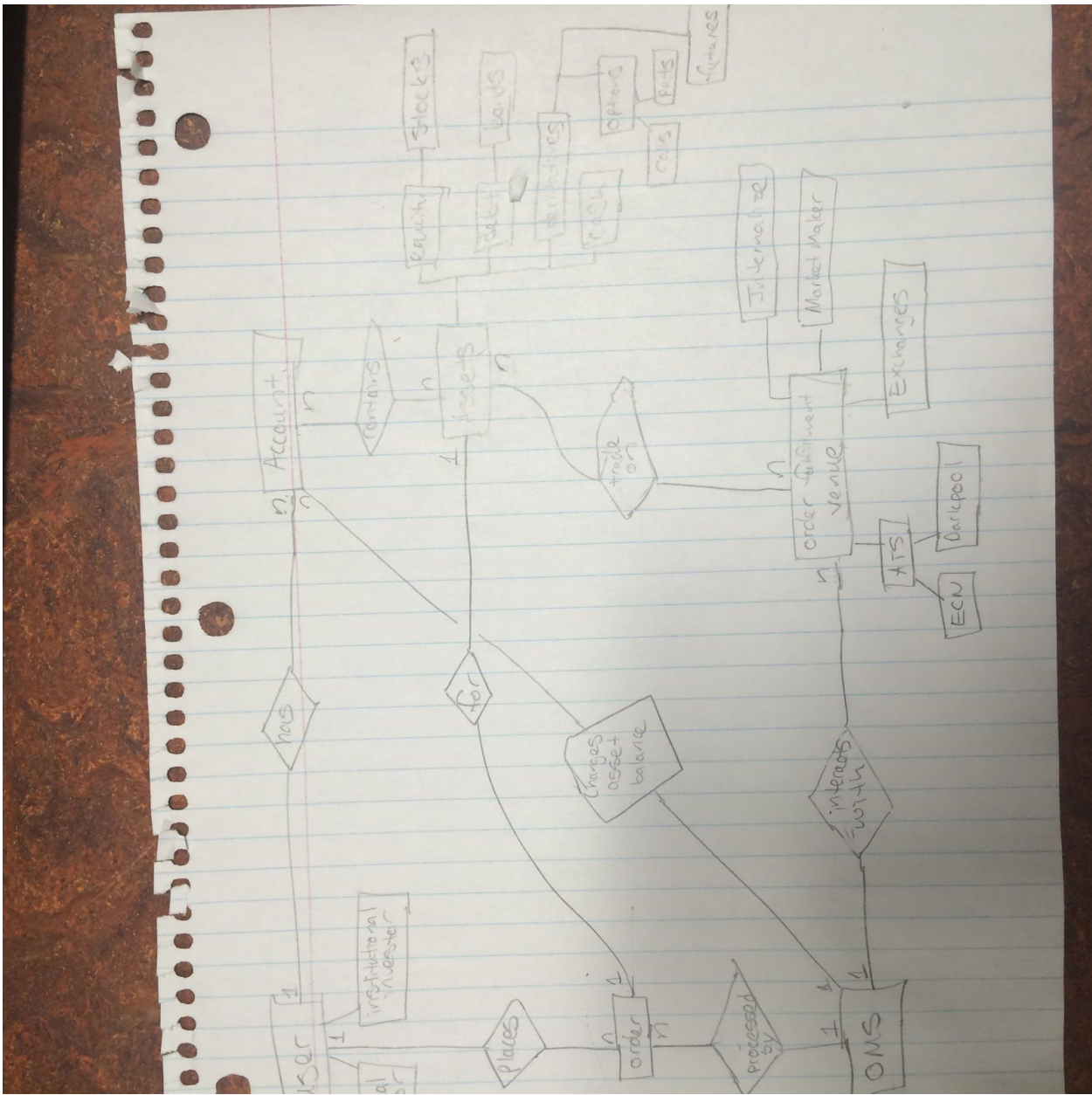
<https://stackoverflow.com/questions/14795726/getting-data-from-yahoo-finance>

IEX API:

<http://help.intrinio.com/data-feeds/how-to-iex-real-time-stock-prices-via-websocket-and-firehose>

<https://intrinio.com/data/nasdaq-basic-realtime-stock-prices>

<https://www.investopedia.com/articles/financialcareers/06/mmakertricks.asp>



Automated Trading Systems (ATS):

- An automated trading system is just what it sounds like - it is a computer program that autonomously places trades financial securities based upon market data (fundamental and technical analysis) and the trading strategy underlying it.
- As of 2014, 75% of trades made on the NYSE originated from ATSs¹.
- ATS interact with several third parties to fulfill the orders they would like to place (e.g. dark pools, ECN and other automated exchanges).

Additional Information (not entirely relevant to our proposal):

The Decimalization of the markets:

The US markets were decimalized in April 2001. This means shares began trading on the penny. Shares traded in terms of sixteenths of a dollar (i.e. 6.25 cents) prior to the decimalization of the markets. Back in the day, “blocks of stock” were traded in increments of $\frac{1}{8}$ of a dollar with a size of 10,000 shares. The SEC initiated the change from sixteenths of a dollar to one-hundredths of a dollar because the smaller spreads would increase competition between market makers, ultimately benefiting market participants. The SEC also believed decimalization was necessary to remain competitive with the international markets that were already decimalized.

The decimalization of the markets brought about less liquidity which adversely affects market participants and increases the cost of trading shares. The cost of trading increases with decimalization because it necessitates many more trades to fill large orders. There's less liquidity because before there were eight possible price points within the dollar, but now there are 100 different price points. This induces less volume at each price point, which is why numerous orders are required to fulfill large orders.

However, the slimmer spreads often times lead to a misalignment in order sizes. For instance, if bank ABC is trying to buy 5000 shares of Company XYZ and the spread is 1 penny, the seller may only be selling 3000 shares; bank ABC must look elsewhere for the remaining 2000 shares.

¹ https://en.wikipedia.org/wiki/Automated_trading_system

Thomas Ciha: x2016onq@stfx.ca

Ethan Murphy: x2015jpp@stfx.ca

Jacob Burke: x2015lbl@stfx.ca

Role of a Market Maker:

<https://www.investopedia.com/university/electronictrading/trading2.asp>

Limit Order Information System (LOIS):

<https://www.investopedia.com/terms/l/limit-order-information-system-lois.asp>

Links:

[ht://www.investopedia.com/investing/basics-trading-stock-know-your-orders/](http://www.investopedia.com/investing/basics-trading-stock-know-your-orders/)

<http://www.aaii.com/journal/article/how-your-buy-and-sell-orders-get-filled.touch>

<https://www.investopedia.com/study-guide/series-55/commissions-and-trade-complaints/broker-vs-dealer/>

tp://doc.algotrader.ch/html/Java_Environment.html

<https://www.investopedia.com/articles/01/022801.asp>

[Https](#)

<https://www.forbes.com/2009/03/09/decimalization-uptick-rule-intelligent-investing-volatility.html#2c53eaae4dea>

MARKET DATA API:

<https://iextrading.com/developer/>

<https://intrinio.com/data/us-equities-level-1-realtime-stock-prices>

<http://www.aaii.com/journal/article/how-your-buy-and-sell-orders-get-filled.touch>

