CAB302 Assignment Requirements

Must haves

Trading Platform

- A trading platform to support many different resources that need to be efficiently distributed and shared across the organisation.

Electronic Credits

- Electronic credits will be used to buy resources.
- Each organisation will be assigned credits, each user will have access to their organisational unit's credits

Flexible Solution

- Solution needs to be flexible such that we can add/remove resources to trade easily.

Marketplace Model

- The software platform must facilitate trades via a marketplace model.
- buy/sell orders can be partly fulfilled if there is a valid order available, the new quantity available will then be displayed for the original order to carry on.

Buy Orders

- Organisational units who wish to buy resources can put in a buy order for certain things.

Sell Orders

- Organisational units who wish to sell resources can put in a sell order for a certain quantity/price or a resource.

Reconcile Outstanding Trades

- The software needs to periodically check and reconcile all outstanding trades.

Transaction at Lowest Price Offering

- When a buy order is made for the same resource with different price

  offerings, the transaction should be carried out using the lower price.
- Transactions should be carried out even if there is some difference between buy and sell price. Only if the sell price is lower than the offering buy price (e.g 'BUY 10 widgets for 20 credits' and 'SELL 10 widgets for 15 credits')

Client/Server Model

- The software must be implemented with a client server model where the client for the project hosts the server to keep track of every organisational unit's assets/credits/- trades, where the users will connect to this server to list trades.
- Have the client-server model run on 1 server, which clients connect to.

Scalable Solution

- No artificial limit will be made to the number of commodities in the database, number of trades listed or number of users using the system.

User Credentials

- Every user has to have their own username and password.

GUI

- Friendly GUI interface, not a CLI.

User Access

- Every user need access to the system through their own usernames/passwords. These users need to be trading using the credits available to the organisational unit they are a part of.

Overdraft Protection

- The software must check both credits and resources available to the organisational unit and prevent over-drafting so that they won't go into debt.
- This includes current orders, so a client can't make sell orders that total more than they own.

Restrict Access

- Only they have the authority to create new organisational units. Need authority to edit number of credits they have, each asset they have etc.
- Need to implement special types of user accounts that can do this from the GUI client.
- Things to be stored in a db:
  - User info (username, password, account type, org. unit).
  - Org. unit name, credits, assets, quantity of each asset.
  - Asset types.
  - Current trades (buy/sell, org. unit, asset name, quantity, price, date) – trade history (buy/sell, org. unit, asset name, quantity, price, date).
- No plain text passwords are to be sent over the network or stored in a database, passwords at least need to be hashed beforehand.

Should haves

- When listing a buy or sell order, users wants to see what current buy and sell offers are currently listed so that they don't over/under bid.
- Users should be able to remove an offer and then list it again at a different price. They should be able to see all currently standing offers from their organisational unit in the database and selectively remove/edit them.
- When buying and selling an asset, users need to see price history of that asset to see what it was sold/bought for in the past.
- Authorised users should be able to add/remove/edit quantity of assets connected to each organisation, add new users, assign users passwords and create new organisational units.
- Authorised users should be able to add new users with the same level of access, so they can give it to anyone who joins the admin team.

Nice to haves

- Nice to be able to change own password without asking IT team to do it for them.
- Implement a graph to show resource price history with x being date and y being price.
- When the software is open and a trade involving their unit is reconciled, they would like to have a little message alert somewhere to indicate this.
- Would like the client to read from some kind of configuration file to get the server IP address and port to connect to, and server should do the same thing to get its port number. This might be needed if the server is to move to as different computer, easiest way to get the new configuration information.