



1. Scraper
 - a. Collects data from Yahoo! Finance at every set interval
 - b. Updates the Database of each ticker
 - c. To improve efficiency, can attempt to pipeline by informing Strategy Calculators immediately after completion of each ticker.
2. Database
 - a. CSV/JSON form of all the tickers
3. Strategy calculators
 - a. Made up of many modules, each module uses their own set of indicators to calculate if a position should be made, and the TP/SL.
 - b. Extracts (read only) data from the Database to commence calculation.
 - c. Interface for Scraper:
 - i. Inform(String TickerName, String TimeStamp)
4. Analyser
 - a. Assumes at all position advices are taken regardless of Strategy weightage.
 - b. Tracks these positions to identify if it has succeeded or failed.
 - c. Successful positions will result in the improvement of strategy's weight in future decisions by the Comparator
 - d. Failed positions will result in the decrease in the strategy's weight in future decisions by the Comparator
 - e. Interface for Strategy Calculators:
 - i. StrategyOutput(Int BuySell, Float TP, Float SL)
5. Comparator
 - a. Each Strategy Calculator is given a rating due to their past performances
 - b. Outputs from the respective Strategy Calculators are imported given a weighted score based on their rating.
 - c. A high overall score could pass through and carried out, while low overall score will be disposed.
 - d. The ratings of the Strategy Calculators can be updated by the Analyser
 - e. Interface for Strategy Calculators:
 - i. StrategyOutput(Int BuySell, Float TP, Float SL)
 - f. Interface for Analyser:
 - i. UpdateWeightage(String StrategyCalculator, Int WeightageChange)
6. Executor
 - a. Performs the execution of the Comparator on the Brokerage Platform
 - b. Interface for Comparator:
 - i. Execute(Int BuySell, Float TP, FloatSL, Int Leverage)