**Minance – High Level Design**

Controller

* Dashboard
  + View() – return the view.
  + Run() – starts up listeners and calls decisionService.

Model

* BasePriceModel
  + VAR\_ID - String
  + Symbol – String
  + Time – TimeDate
  + Price – Double
  + PercentageChange (Since app started)
* OrderHistoryModel
  + Symbol – String
  + Time – TimeDate
  + PurchasePrice – Double
  + SellPrice – Double
  + PercentGain – Double
  + madeProfit - Bool
* TradeVariableModel
  + VAR\_ID – Key
  + LowerLimit – percent to lose before sell
  + UpperLimit – percent to gain before sell

View

* Dashboard
  + Run – Button. Later will be start/stop.
  + Live ticker of coin price – later feature.

Service

* CandleStickService
  + CreateRecord() – saves record of basePriceModel calling repo.save().
* DecisionService
  + Extracts csv to a list of TradeVariableModel
  + Calls decisionEngine.start() per variableSet taken in as constructor params;
* OrderHistoryService
  + CreateRecord() – saves record of orderHistoryModel, calling repo.save();

Listener

* WebSocketListeners
  + LivePriceTicker – constant live price for btc.
  + 1minuteCandleStick – when bar final is true, calculate percentage gain on start price, call candlestickService.createRecord().

Logic

* DecisionEngine extends Thread
  + Run() – is used to start the code, calls prepare().
  + Prepare() – sets base price(global) and gets latest price(method calling listener).
  + Count() – counts to 3 and makes purchase, resets counter.
  + BuyTest() – generateLimits(upper and lower sell prices) and calls sellTest().
  + SellTest() – while the coin is owened, compared prices to detect highest price achievable. Once outside the threshold (best upper or lower), generateFees(), getPercentageChange(), store as OrderHistoryModel, setsCoinOwned to false, generatesReport().
  + GenerateReport() – calls orderHistoryService.createRecord().
  + GenerateFees() – calculates cost of transaction.
  + GeneratePercentageChange – calculates the gain or loss of the transaction.
  + ResetCounter() – sets value back to 0;
  + getLatestPrice() – gets price from listener.
  + coinOwened() – bool method.

Repository

* BasePriceRepository – Saves the base price to mongoDb.
* OrderHistoryRepository – Saves the orderHistoryModel to mongoDb.