Use Case: Run Forward-tester

Brief Description

This use case defines the forward test function.. It is able to test trading strategies by buying and selling positions as defined by a strategy. The strategy defines the buying conditions and the selling conditions. If a condition is satisfied, the system buys/sell according to the condition, and the quantity of the position that is bought/sold is defined by the user.

Flow of Events

**Basic Flow**

The use case starts when the market opens and is repeated every 10 seconds while the market is open.

1. The system is turned on at 9:30am
2. The system obtains all strategies saved in the portfolio.
3. The system takes in data (the price of the stock/coin, the target contract, and the time). The system performs steps 4 and 5 for each strategy
4. The system checks to see if any buying condition is met
   1. Buying conditions can be related to the stock price or the target contract’s price
   2. If a buying condition is met, the system enters the **Buy Contract Flow**
   3. The system cannot move on to step 5 until all contracts are bought for each strategy
5. The system checks to see if any selling conditions are met
   1. If a selling condition is met, the system enters the **Sell Contract Flow**
6. Save the date, time, and portfolio value to the database to retrieve later
   1. You should also have saved what positions you have, what their current price is, and what the purchase price is
7. At 3:55PM, calculate the percent gain/loss for the day. Send an email to me with the results

**Buy Contract Flow**

This use case begins when the systems decide to buy a contract

1. The system checks a variety of preconditions. If a precondition is not satisfied, this flow is terminated
   1. The system checks to see if there isn’t an *InProgress* order status
   2. The system checks to see if enough days has passed since the last purchase
   3. The system checks to see if the strategy is not using more than its maximally defined allocation
   4. The system checks to see if there is enough buying power
   5. The system checks the user-created buying conditions, usually involving price movement or change in position’s price
2. The system initiates a buy action (deducting/increasing buying power) and creates an order with an order status of *InProgress*.
   1. The type of contract and number of contracts is defined by the user
3. The system iteratively checks the order status. If the order isn’t filled in 30 seconds, the order is cancelled, and this flow is terminated
4. After the order is filled, an email is sent with the following information
   1. What was sold
   2. The time the order was initiated
   3. The time the order was filled
   4. Current positions (and their unrealized p/l)
   5. Day performance (p/l gain)
   6. Week performance (p/l gain)
5. The new position is stored in a database along with the price it filled at

**Sell Contract Flow**

This use case begins when the systems decide to sell a contract

1. The system checks a variety of preconditions. If a precondition is not satisfied, this flow is terminated
   1. The system checks to see if there isn’t an *InProgress* order status
   2. The system checks to see if enough days has passed since the last sale
   3. The system checks to see if the strategy is not using less than its minimally defined allocation
   4. The system checks to see if there is enough buying power (assuming the sell action takes a debit)
   5. The system checks the user-created selling conditions, usually involving price movement or change in position’s price
2. The system initiates a sell action (deducting/increasing buying power) and creates an order with an order status of *InProgress*.
   1. You can sell a number of contracts or a percent of your holdings for the strategy
3. The system iteratively checks the order status. If the order isn’t filled in 30 seconds, the order is cancelled, and this flow is terminated
4. After the order is filled, an email is sent with the following information
   1. What was sold
   2. The time the order was initiated
   3. The time the order was filled
   4. Current positions (and their unrealized p/l)
   5. Day performance (p/l gain)
   6. Week performance (p/l gain)
5. The new position is stored in a database along with the price it filled at