|  |  |
| --- | --- |
| Use case ID | backtesting-backtest |
| Use Case Name | backtest |
| Goal | Do the backtesting of a specific strategy |
| Primary Actors | programmer |
| Pre-Conditions | Programmer formulated a strategy and prepared candle stick data of chosen commodities |
| Post-Conditions | Simulating all of the strategy behaviors and get the history of all trades on selected commodities. Getting a statistic result of the trading history. |
| Related Actions | All of the usecases related to strategy |
| Basic Flow | 1. Fetch candle stick data from chosen commodities by the programmer 2. Simulate all actions of the strategy by vectorized method and store all of the history actions 3. Calculate the statistics from the history 4. Return all the statistics and history by dictionary |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | Backtesting-walkforward-analysis |
| Use Case Name | Walk forward analysis |
| Goal | Divide all of the candle stick data by sliding window and do backtest in each sliding window |
| Primary Actors | programmer |
| Pre-Conditions | Programmer formulated a strategy and prepared candle stick data of chosen commodities |
| Post-Conditions | Simulating all of the strategy behaviors and get the history of all trades on selected commodities in each sliding window. Getting a statistic result of the trading history window by window. |
| Related Actions | All of the usecases related to strategy |
| Basic Flow | 1. Fetch candle stick data from chosen commodities by the programmer 2. Simulate all actions of the strategy by vectorized method and store all of the history actions within a windowed data 3. Calculate the statistics from the windowed history 4. Repeat step 1 until no more windowed data 5. Return the statistics and history by dictionary windowed by windowed |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | backtesting-monkey-test |
| Use Case Name | monkey test |
| Goal | Do the monkey test of a specific entrance/exit strategy |
| Primary Actors | programmer |
| Pre-Conditions | Programmer formulated a strategy and prepared candle stick data of chosen commodities |
| Post-Conditions | Simulating all of the strategy behaviors and get the history of all trades on selected commodities. Getting a statistic result of the trading history. |
| Related Actions | All of the usecases related to strategy |
| Basic Flow | 1. Fetch candle stick data from chosen commodities by the programmer 2. Simulate all actions of the strategy by vectorized method and store all of the history actions 3. Calculate the statistics from the history 4. Return all the statistics and history by dictionary |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | backtesting-show-statistic-result |
| Use Case Name | Show statistic result |
| Goal | Show the backtest statistic result of a strategy |
| Primary Actors | programmer |
| Pre-Conditions | Programmer gained a result from backtest usecase |
| Post-Conditions |  |
| Related Actions |  |
| Basic Flow | Show the statistics in a formulated table |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | backtesting-show-result-diagram |
| Use Case Name | Show result diagram |
| Goal | Show the backtest statistic result and history of a strategy in diagrams |
| Primary Actors | programmer |
| Pre-Conditions | Programmer gained a result from backtest usecase |
| Post-Conditions |  |
| Related Actions |  |
| Basic Flow | Show the statistics in a formulated table and trading histories diagrams |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | backtesting-parameter-optimization-using-search-based-method |
| Use Case Name | Parameter optimization using searched based method |
| Goal | Using searched based method to search through a given range of numbers and optimize based on a given benchmark performance metric. Finally return the optimal number of the optimized parameter |
| Primary Actors | programmer |
| Pre-Conditions | Given a strategy |
| Post-Conditions | Return the optimal number or ratio |
| Related Actions | All of the usecases related to strategy and backtest usecase |
| Basic Flow | Run multiple backtest usecase and return the number which has the best(max or min) performance metric number |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | Frontend-show-statistics-of-current-strategy |
| Use Case Name | Show statistics of current strategies |
| Goal | Show all the current statistics of all of the running strategies |
| Primary Actors | programmer |
| Pre-Conditions | Given multiple running strategies |
| Post-Conditions |  |
| Related Actions |  |
| Basic Flow | Show statistic of strategies and separate each of them into multiple pages |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | Frontend-show-trading-history |
| Use Case Name | Show trading history |
| Goal | Show all the current trading history of all running strategies |
| Primary Actors | programmer |
| Pre-Conditions | Given multiple running strategies |
| Post-Conditions |  |
| Related Actions |  |
| Basic Flow | Show trading histories of strategies and separate each of them into multiple pages |
| Alternative Flow |  |

(the above usecases are going to integrate into the same page)

|  |  |
| --- | --- |
| Use case ID | Frontend-stop-strategy-from-trading |
| Use Case Name | Stop strategy from trading |
| Goal | Stop designated running strategy |
| Primary Actors | programmer |
| Pre-Conditions | Given a running strategy |
| Post-Conditions |  |
| Related Actions | All of the usecases related to strategy |
| Basic Flow | Close all of the positions and Kill the strategy |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | Frontend-start-strategy-to-trade |
| Use Case Name | Start strategy to trade |
| Goal | Start trading the designated strategy |
| Primary Actors | programmer |
| Pre-Conditions | Given a strategy |
| Post-Conditions |  |
| Related Actions | All of the usecases related to strategy |
| Basic Flow | Start the strategy to trade |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | Frontend-show-current-funding-size |
| Use Case Name | Show current funding size |
| Goal | Show the current and equity curve of the fund |
| Primary Actors | programmer |
| Pre-Conditions | Given an account funding size from broker |
| Post-Conditions |  |
| Related Actions | Fetch funding price from broker |
| Basic Flow | Show the funding size |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | Accounting-manage-stop-loss/profit |
| Use Case Name | Manage stop loss/profit |
| Goal | Define or design the stop loss and profit loss that needed by the strategy |
| Primary Actors | programmer |
| Pre-Conditions | Given a strategy |
| Post-Conditions |  |
| Related Actions |  |
| Basic Flow | Design the details of the specific stop loss/profit |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | Accounting-manage-comodities |
| Use Case Name | Manage comodities |
| Goal | Add or delete the commodities of a strategy |
| Primary Actors | programmer |
| Pre-Conditions | Given a strategy |
| Post-Conditions |  |
| Related Actions |  |
| Basic Flow | Add or delete commodities |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | Accounting-position-sizing |
| Use Case Name | Position sizing |
| Goal | Adjust the position sizing of a strategy |
| Primary Actors | programmer |
| Pre-Conditions | Given a strategy |
| Post-Conditions |  |
| Related Actions |  |
| Basic Flow | Design a position sizing method |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | Accounting-manage-fund |
| Use Case Name | Manage fund |
| Goal | Increase/decrease the fund of a strategy and increase/decrease the amount of fund by buy/selling positions |
| Primary Actors | programmer |
| Pre-Conditions | Given a strategy |
| Post-Conditions |  |
| Related Actions |  |
| Basic Flow | Increase/decrease the funding |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | Accounting-update-indicators |
| Use Case Name | Update indicators |
| Goal | Update new indicator values that needed by a strategy |
| Primary Actors | programmer |
| Pre-Conditions | Given a strategy |
| Post-Conditions |  |
| Related Actions | Update indicators usecase |
| Basic Flow | Add new candle stick data to accounting module and delete unnecessary data that is out-dated Then calculate new values of indicators |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | Accounting-manage-current-positions |
| Use Case Name | Manage current positions |
| Goal | Add/delete/accounting/storing positions of a strategy |
| Primary Actors | Programmer |
| Pre-Conditions | Given a strategy |
| Post-Conditions |  |
| Related Actions |  |
| Basic Flow | Add/delete/accounting/storing positions of a strategy |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | Accounting-manage-current-positions |
| Use Case Name | Manage current positions |
| Goal | Add/delete/accounting/storing positions of a strategy |
| Primary Actors | Programmer |
| Pre-Conditions | Given a strategy |
| Post-Conditions |  |
| Related Actions |  |
| Basic Flow | Add/delete/accounting/storing positions of a strategy |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | Accounting-update-candle-sticks |
| Use Case Name | Update candle sticks |
| Goal | Update new candle stick data the needed by a strategy |
| Primary Actors | Programmer |
| Pre-Conditions | Given a strategy |
| Post-Conditions |  |
| Related Actions |  |
| Basic Flow | Add new candle stick data to accounting module and delete unnecessary data that is out-dated |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | Broker-incubation |
| Use Case Name | incubation |
| Goal | Incubate a strategy |
| Primary Actors | Programmer |
| Pre-Conditions | Given a strategy |
| Post-Conditions |  |
| Related Actions |  |
| Basic Flow | Start the strategy but only simulate its behavior to trade and do the accounting related to the strategy |
| Alternative Flow |  |

|  |  |
| --- | --- |
| Use case ID | Broker-take-data-flow |
| Use Case Name | Take data flow |
| Goal | Take the data feed by the specific data source or broker |
| Primary Actors | Programmer |
| Pre-Conditions | Given a strategy |
| Post-Conditions |  |
| Related Actions |  |
| Basic Flow | Collect the data feed from the data source |
| Alternative Flow |  |
| Use case ID | Broker-place-order |
| Use Case Name | Place order |
| Goal | Place the order by what the strategy want |
| Primary Actors | Programmer |
| Pre-Conditions | Given a strategy |
| Post-Conditions |  |
| Related Actions |  |
| Basic Flow | Place the order by strategy |
| Alternative Flow |  |