**Search Stock**

*Variation 1:* Stock Found

* User enters the symbol of the stock they wish to view.
* The system searches for the stock by the given symbol with a call to a YQL open datatable.
* The stock is found, and its data is placed into a StockQuote object.
* The system displays the stock on screen.

*Variation 2:* Stock not Found

* User performs Search Stock.
* The provided stock is not found in the YQL open datatable.
* The system informs the User that the given stock cannot be found.

**Place Buy Order**

* The user carries out Search Stock for the stock they wish to buy.
* The user selects what stock they wish to buy.
* The user enters the minimum price they would pay for the stock.
* The user enters the maximum price they would pay for the stock.
* The user presses the place order button.
* The system places the order in the “Pending Orders” StockList.

**Place Sell Order**

* The user indicates what stock they wish to sell.
* The user enters the price and quantity they would like to sell.
* The system places the sell order.

**Change Buy/Sell Order**

* The user carries out Place Buy Order or Place Sell Order.
* The user carries out View Portfolio, variation 2 or 3.

*Variation* 1: Buy/Sell More Order

* The user selects the Order in the list that they would like to change.
* The user enters the quantity amount that they would like to change the order quantity by.
* The user presses the “Buy More” or “Sell More” button.
* The system updates the quantity field in the pending order.

*Variation* 2: Cancel Buy/Sell Order

* The user selects the Order in the list that they would like to cancel.
* The user presses the Cancel Order button.
* The system cancels the order and removes it from the list of pending orders.

**View Portfolio**

* The user presses the view portfolio button.
* The system reads the user’s portfolio.
* The system displays the user’s portfolio.

*Variation* 1: Currently Owned

* The user carries out View Portfolio.
* The user selects the “currently owned” option from the drop down menu.
* The system displays the user’s currently owned stocks.

*Variation* 2: Pending Buy

* The user carries out View Portfolio.
* The user selects the “pending buy” option from the drop down menu.
* The system displays the user’s stocks that are pending purchase.

*Variation* 3: Pending Sell

* The user carries out View Portfolio.
* The user selects the “pending sale” option from the drop down menu.
* The system displays the user’s stocks that are pending sale.

**Update All Stock Data**

* The user presses the “update” button in the window.
* The system retrieves the most recent stock data it can, or the last update if the market is closed..
* The system updates all stock lists with the newly retrieved values.
* The system executes all pending orders if their conditions are met.
* The system displays the updated stock list.

**Execute Pending Orders**

* The user carries out Place Buy Order or Place Sell Order.
* The user carries out the Update All Stock Data.
* The system iterates through the updated lists (Buy Orders and Sell Orders).

*Variation #1: Conditions of an order are not met*

* The system checks the conditions of the pending order.
* The system finds that at least one order’s conditions have not been met.
* The system does nothing to the order.
* //The system executes only the orders whose conditions have been met.
* //The system updates the “Currently Owned” list in the Portfolio.

*Variation #2: Conditions of an order are met*

* The system checks the conditions of the pending order.
  + For “Buy Order,” the system adds the stock to the user’s portfolio and adjusts the user’s virtual wallet according to the cost of executing the Buy Order.
  + For “Sell Order,” the system removes the stock from the user’s portfolio and adjusts the user’s virtual wallet according to the cost of executing the Sell Order.
* The system updates the “Currently Owned” list in the Portfolio.

**Save All Data**

* The user carries out any of the other use cases.
* The user presses the Save button.
* The system serializes all user data.
* The system saves serialized data into “StockTrader.ser”
* Upon opening the program the next time, the new data will be input into the system.