

```
1      /*
2          Author's Name:      Kenneth Larot Yamat
3
4          Purpose of Program:  To create a program that automatically creates
5                               trading tickets for a security, for example, buy
6                               and sell orders for shares of an exchange traded fund.
7
8          Date Due:           11:59 PM on March 4th, 2024
9      */
10
11     Project Proposal:  To create a program that automatically creates
12                       trading tickets for a security, for example, buy
13                       and sell orders for shares of an exchange traded fund.
14
15                       A user would only manually enter the first order, either to
16                       buy or sell a security, the program would populate and submit
17                       a new ticket based on the fulfillment of the previous ticket,
18                       the chain of tickets would continue until the user decided to
19                       cancel the chain.
20
21     A) Background and the needs:
22
23         This program is needed because there are many securities
24         that are difficult to trade because they are illiquid as
25         a result of large bid and ask spreads, or because they lack
26         volume.
27
28         The goal is to reduce spreads while increasing volume.
29
30         Another need is due to the fact that manually performing this task
31         is laborious and prone to error.
32
33     B) Function list:
34
35         getSecurityPrice
36         setSecurityPrice
37
38         setPurchasePrice
39         getPurchasePrice
40
41         setLiquidationPrice
42         getLiquidationPrice
43
44         setAvarageTrueRange
45         getAverageTrueRange
46
47         setBollingerBandWidth
48         getBollingerBandWidth
49
50         setAverageDirectionalIndex
51         getAverageDirectionalIndex
52
53     C) User interface (UI) design:
54
55         Step 1
56
57
```

```
58
59         Trade Ticket
60         =====
61
62         Security:      [User Input Element]
63         Buy or Sell:   [User Input Element]
64         Limit:         [User Input Element]
65         Quantity:      [User Input Element]
66         ATR:           [User Input Element]
67         BBW:           [User Input Element]
68         ADX:           [User Input Element]
69
70     Step 2
71
72
73     Your initial [Buy/Sell] Trade ticket for [Security] has been
74     submitted at the following price [Limit Price] and quantity [Quantity].
75
76     Subsequent orders will be automatically generated and submitted contingent
77     upon the fulfillment of the previous order, with buy limits and sell limits
78     based on the Average True Range, Bollinger Band Width, and Average Directional
79     Index entered on the initializing ticket.
80
81     Sell orders will be generated with a limit of  [Calculated Amount] above the previously filled ticket
82     Buy  orders will be generated with a limit of  [Calculated Amount] below the previously filled ticket
83
84     [  User Input Element  [Accept and Submit]  [Override and Submit]  [Start Over]  ]
85
86     Step 3
87
88     [Ticker Symbol]      [Buy/Sell]      [Order Quantity]      [Limit Price]      [Ticket Status]
89
90     [  HFH.P              Buy              1              86.86              Open              ]
91
92     Step 4
93     [Ticker Symbol]      [Buy/Sell]      [Order Quantity]      [Limit Price]      [Ticket Status]
94
95     [  HFH.P              Buy              1              86.86              Filled              ]
96
97     Step 5
98
99     [Ticker Symbol]      [Buy/Sell]      [Order Quantity]      [Limit Price]      [Ticket Status]
100
101     [  HFH.P              Sell             1              86.89              Open              ]
102
103     Step 6
104
105     [Ticker Symbol]      [Buy/Sell]      [Order Quantity]      [Limit Price]      [Ticket Status]
106
107     [  HFH.P              Sell             1              86.89              Filled              ]
108
109     Step 7
110
111     [Ticker Symbol]      [Buy/Sell]      [Order Quantity]      [Limit Price]      [Ticket Status]
112
113     [  HFH.P              Buy              1              86.87              Open              ]
114
```

```
115 Step 8
116
117 [Ticker Symbol] [Buy/Sell] [Order Quantity] [Limit Price] [Ticket Status]
118
119 [ HFH.P Buy 1 86.87 Filled ]
120
121 Notes: this sequence is based on + 00.03 to Sell orders and - 00.02 to
122 Buy orders for first issue preferred shares for the security HFH
123
124 D) Class diagram
125
126
127
128
129 [AverageTrueRange] [BollingerBandWidth] [AverageDirectionalIndex]
130 ↓ ↓ ↓
131 ↓ ↓ ↓
132 =====
133 ↓
134 ↓
135 ↓
136 ↓
137 [ExistingBidAskSpread] =====> [UserTicketBuySellDistance] =====> [Ticket] =====> [AutoTicket]
138 - bidAskSpread double ↑ ↑ - ticketQuantity int - sequenceMaximum int
139 ↑ ↑ - acquisitionPriceLimits double
140 ↑ ↑ - disposalPriceLimits double
141 ↑ ↑
142 [CurrentVolume] =====>↑ ↑
143 - currentVolume int ↑
144 ↑
145 [TimeHorizon] =====
146 - amountOfTime int
147
148
149
150 E) File and database design:
151
152 [Data Dictionary for Database Tables and Non-Database Files]
153
154 [File and Database Design]
155
156 data will be instantiated as an ArrayList and printed initially and stored as .txt files, a program will be created
157 to convert these .txt files into .xml and .csv files where and when appropriate.
158
159 [Data Dictionary]
160
161 the data dictionary will define the columns ticker symbol, buy/sell, order quantity, limit price, and ticket status.
162 the data dictionary will also contain the methods and classes that modify or control this data.
163
164 [Database Tables]
165
166 will be organized by column headers such as date, ticker symbol, order quantity, limit price, ticket status
167
168 [Non-Database Files]
169
170 will contain the initial trade authorization, and the user inputs, authorization for the subsequent auto trades
171 based on the other inputs, or, authorization for the automated trades based on user overridden inputs.
```

```
172
173     [Relational Database]
174
175     each row in the data file represents on ticket, all the data on that ticket is related to that particular ticket, each
176     ticket is related to the previous ticket
177
178     [Plain Text Files]
179
180     all data will initially be created as ArrayLists and converted into plain text files.
181
182
183
184 F) Expectations of project fulfillment:
185
186     a. [Ticket]             instantiates based on user input.
187
188     [AutoTicket]           instantiates based on fulfillment of previous ticket.
189
190     b. [Controller Classes] The Ticket.java class is the view class
191
192                             UserTicketBuySellDistance.java is the controller class, it sets
193                             the limits for automatically generated tickets
194                             these automatically generated distances are only suggestions,
195                             and can ultimately be overridden by the user in the Ticket.java
196                             class.
197
198                             AverageTrueRange.java BollingerBandWidth.java AverageDirectionalIndex.java
199                             ExistingBidAskSpread.java CurrentVolume.java TimeHorizon.java
200                             are model classes that feed into the UserTicketBuySellDistance.java
201                             controller class. these will be calculated, but the user will have the
202                             ability to override these values.
203
204     c. [GUI applications]
205
206     d. [Arraylist]           Arraylist will be used to log the sequence of trades
207
208     e. [Exception handling]  a user may enter alphabetical values in a field that requires an int or
209                             double, and vice versa, an invalid data message will prompt the user.
210
211     f. [Database]
212
213     g.
214     [Documentation]          very detailed and elaborate notes will be included in every program,
215                             class, method, and attribute regarding the purpose, design, development,
216                             and miscellaneous other notes as well. JavaDoc will see extensive use.
217
218 G) Project Report
219
220     a.
221     b.
222     c.
223     d.
224     e.
225     f.
226     g.
227     h.
228
```

229
230