## TheProposalAndTheSolution.txt

1	/*			
2		Author's Name	:	Kenneth Larot Yamat
3				
4		Purpose of Pr	ogram:	To create a program that automatically creates
5		·	J	trading tickets for a security, for example, buy
6				and sell orders for shares of an exchange traded fund.
7				and seem of delice for single of an exchange change full
8		Date Due:		11:59 PM on March 4th, 2024
9	*/	bace bac.		11755 TH OH Haren Telly 2021
10	,			
11	Projec	ct Proposal:	To create	a program that automatically creates
12	110100	ic iroposui.		ckets for a security, for example, buy
13				orders for shares of an exchange traded fund.
			and Sell C	orders for shares of all exchange traded fund.
14			A	. Id oul, would be the first order ofther to
15				ald only manually enter the first order, either to
16				l a security, the program would populate and submit
17				set based on the fulfillment of the previous ticket,
18				of tickets would continue until the user decided to
19			cancel the	e chain.
20				
21	A) Bad	kground and the	needs:	
22				
23			This progr	ram is needed because there are many securities
24			that are o	Hifficult to trade because they are illiquid as
25			a result o	of large bid and ask spreads, or because they lack
26			volume.	
27				
28			The goal i	s to reduce spreads while increasing volume.
29			8	
30			Another ne	ed is due to the fact that manually performing this task
31				ous and prone to error.
32			13 1000, 10	and profile to critor.
33				
34	B) Fur	nction list:		
35	b) rai	iccion iiijc.		
36			getSecurit	vPnice
37			setSecurit	
38			3ecsecui 10	ymice
39			setPurchas	raDni ca
			500.0.0.0.0.	
40 41			getPurchas	DEFITUE
41			co+1 * ~* -1-	ationDnico
42			setLiquida	
43			getLiquida	ICTOULLICE
44				
45			setAvarage	
46			getAverage	rueRange
47				- h.t.l.l
48				gerBandWidth
49			getBolling	gerBandWidth
50				
51				eDirectionalIndex
52			getAverage	DirectionalIndex
53				
54	C) Use	er interface (UI	) design:	
55	•	•	-	
56	St	tep 1		
57				

го													
58 59		Trade Ticket											
60		==========											
61			F										
62		Security:	[User Input Elem	-									
63		Buy or Sell:	[User Input Elem	ent]									
64		Limit:	[User Input Elem	ent]									
65		Quantity:	[User Input Elem	ent]									
66		ATR:	User Input Elem	entĪ									
67		BBW:	User Input Elem	-									
68		ADX:	[User Input Elem	-									
69		ADA:	[OSCI TIIPUC LICIII	circi									
70	Cton 2												
	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 automatica	lly generat	ed and submitted	d contingent							
77		Subsequent orders will be automatically generated and submitted contingent upon the fulfillment of the previous order, with buy limits and sell limits											
78			ge True Range, Bol										
79			he initializing ti		,	.60							
80		THUCK CHECKED ON C	ine initializing ti	ckee.									
81		Coll andone will b	o gononated with a	limit of	[Calculated Ame	ountl above the proviously fills	d ticket						
			e generated with a			ount] above the previously fille							
82		Buy orders will b	e generated with a	limit of	[Caiculated Amo	ount] below the previously fille	a ticket						
83			_										
84		[ User Input Ele	ment [Accept and	Submit]	[Override and Su	ubmit] [Start Over] ]							
85													
86	Step 3												
87	·												
88		[Ticker Symbol]	[Buy/Sell]	[Order	Quantity]	[Limit Price]	ſTicket	Status]					
89		[:reke: Symbol]	[50,75011]	[0.46.	Quantity]	[2226 11266]	Lizekee	5646451					
90		[ HFH.P	Punz		1	86.86		Onon	1				
		[ HEH.F	Buy		1	80.80		0pen	J				
91	614												
92	Step 4		/										
93		[Ticker Symbol]	[Buy/Sell]	[Order	Quantity]	[Limit Price]	[licket	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		[1=1110. 27004]	[,,]	[ 5. 46.	C J ]	[		~ 1					
101		[ HFH.P	Sell		1	86.89		0pen	]				
		[ HEH.F	2611		1	80.83		орен	J				
102	C±a.a. c												
103	Step 6												
104		<b></b>	r= /			F	F ·	a					
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		[ . Teker Jymoot]	[249/3011]	Lorder	eacher ch ]	[2110 11100]	LITCKEC	5 64 645 ]					
113		[ HFH.P	Dinz		1	86.87		Onon	1				
		l neu.r	Buy		1	00.0/		0pen	]				
114													

```
115
             Step 8
116
117
                        [Ticker Symbol]
                                         [Buy/Sell]
                                                           [Order Quantity]
                                                                                    [Limit Price]
                                                                                                             [Ticket Status]
118
                                                                                                                    Filled
119
                        [ HFH.P
                                                 Buy
                                                                                               86.87
120
                        this sequence is based on + 00.03 to Sell orders and - 00.02 to
121
             Notes:
                        Buy orders for first issue preferred shares for the security HFH
122
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
139
140
142 [CurrentVolume] ========>>↑
143
144
146
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
             [Data Dictionary]
159
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
                 will contain the initial trade authorization, and the user inputs, authorization for the subsequent auto trades
170
                 based on the other inputs, or, authorization for the automated trades based on user overridden inputs.
171
```

## TheProposalAndTheSolution.txt

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 b. [Controller Classes] 190 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 ExistingBidAskSpread.java CurrentVolume.java TimeHorizon.java 199 are model classes that feed into the UserTicketBuySellDistance.java 200 201 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] 212 213 214 [Documentation] very detailed and elaborate notes will be included in every program, 215 class, method, and attribute regarding the purpose, design, development, and miscellaneous other notes as well. JavaDoc will see extensive use. 216 217 218 G) Project Report 219 220 a. 221 b. 222 с. 223 d. 224 e. 225 f. 226 g. 227 h. 228

TheProposalAndTheSolution.txt

Monday, March 4, 2024, 11:23 PM