

Component A (developed by pair of Qian Wang & Zicheng Xu)

reader: Qian Wang

recorder: An Wang and Jinyi Zhang

inspected components: **dashboard.js** and **dashborat.html** in app dashboard

problem 1:

In function JPTrader.drawChart, we missed some information that should be shown in the chart.

```
735     credits: {
736       enabled: false
737     },
738     title: {
739       text: 'ETF Quote'
740     },
741     xAxis: {
742       type: 'datetime',
743       tickPixelInterval: 15
744     },
745     yAxis: [{
746       title: {
747         text: 'Price'
748       },
749       plotLines: [{
750         value: 0,
751         width: 1,
752         color: '#808080'
753       }]
754     }, {
```

problem 2:

In function JPTrader.drawChart, the index of an iteration was wrong.

```
784     let i;
785     for ( i = -59; i <= 0; i += 1 ) {
786       data.push({
787         x: time + i * 1000,
788         y: 0
789       });
790     }
```

problem 3:

We didn't add {{{}} outside of variable email transmitted from views.py, which led to the string value of email cannot be read correctly.

```
117
118 <script type="text/javascript">
119   const _username = "{{username}}";
120   const _userid = {{userid}};
121   const _email = "{{email}}";
```

problem 4:

In html file, we added two quotations of jquery, which caused some functions cannot work. Also, the sequence of script belong was wrong.

```
4 <meta charset="utf-8">
5 <link rel="stylesheet" href="http://cdn.static.runoob.com/libs/bootstrap/3.3.7/css/bootstrap.min.css">
6 <script src="http://cdn.static.runoob.com/libs/jquery/2.1.1/jquery.min.js"></script>
7 <script src="http://cdn.static.runoob.com/libs/bootstrap/3.3.7/js/bootstrap.min.js">
</script>
```

problem 5:

In the function onmessage() of web socket, we missed some cases before the code inspection.

```

423
424     case "sold_message":
425
426         console.log('Received a sold_message...');
427         console.log(message);
428         // find corresponding handler from current orders array
429         const _currentOrders = JPTrader.currentOrders;
430
431         for (let i = 0; i < _currentOrders.length; i++) {
432             if (_currentOrders[i]["instrument_id"] == message["instrument_id"]) {
433                 _currentOrders[i].handler(message);
434                 break;
435             }
436         }
437
438         // update the aggregate view here.
439         JPTrader._updateAggregateView();
440
441         break;
442
443     case "resume_order":
444         JPTrader._resumeHandler( message );
445         break;
446
447     case "unfilled_order":
448
449         console.log('Received an unfilled message from server');
450         break;
451
452     case "canceled_order":
453         console.log('Received an canceled confirmation message from server');
454         JPTrader._canceledHandler(message);
455         break;
456
457     case "finished_order":
458         console.log('Received an finished message from server');
459         break;
460
461     default:
462         console.log('Data type not specified or not recognized');
463         break;
464 }

```

problem 6:

In function JPTrader.sendWithSocket(), the function ws.send() was called at wrong time, it should be called after the initialization of web socket.

```

224     let customize = function(){
225         const customizeRequest = {
226             "request_type": "customize_request",
227             "instrument_id": $("#stock_id").val(),
228             "order_size": $("#new_order_size").val(),
229             "order_discount": $("#new_discount").val(),
230             "order_duration": $("#new_duration").val(),
231         };
232
233         console.log(customizeRequest);
234
235         this.ws.send(JSON.stringify(customizeRequest));
236         $('#myModal').modal('hide');
237     }
238     document.getElementById("customize_btn").addEventListener("click", customize);

```

Component B(developed by pair of Jinyi Zhang & An Wang)

reader: Jinyi Zhang

recorder: Zicheng Xu

inspected components: sessionmanager.py in app sessionmanager

problem 1:

The time of sending email to inform trader was wrong. In addition, the content of the email should contain the information of order. And it should also check whether the trader choose to receive inform email before this line of code.

```

# Send email to inform trader that an the order is finished
send_mail('One of your order finished', 'One of your order finished.', 'wa

```

problem 2:

The function of setting bid window has some problem. It should be adjustable as said in relevant user story.

```

267     def set_bid_window(self):
268         """
269         a function that determines bid_window by remaining
270         """
271         execution_times = self.quantity // self.order_size
272         if self.quantity % self.order_size != 0:
273             execution_times += 1
274
275         self.bid_window = self.time // execution_times
276
277         # More child orders than remaining time
278         if self.bid_window == 0:
279             self.bid_window = 1 # Set to the minimum bid w

```

problem 3:

Although the session manager would handle diverse session, but they are actually handled in the same thread. Since different suborder should be treated in diverse threads, we need to modified it to multiple thread mode. Therefore, we need to deal with the locking and unlocking of different threads.

```

208         if self.session_manager:
209             for session in self.session_manager.values():
210
211                 session.set_init_timestamp(timestamp)
212                 if not session.saved:
213                     self.save_order(session)
214
215                 message_json = session.trade(price)
216
217                 if message_json:
218                     return_list.append(message_json)
219

```

problem 4:

We shouldn't minus the price by order_discount directly. Instead, we should treat order_discount as percentage value and minus the corresponding part.

```

209         else:
210             # If meets an order execution time
211             if self.step < self.bid_window:
212                 self.step += 1
213             else:
214                 self.step = 1
215
216             # Attempt to execute a sell order. By only referencing the last
217             order_size = min(self.quantity, self.order_size)
218             sell_price = price * (1 - float(self.order_discount) / 100)
219
220             message = self.execute_order(order_size, sell_price)

```

problem 5:

In function ws_message(message), we missed some elif condition as in dashboard.js.

```

414         if request_type == 'order_request':
415             # Initialize order parameters
416             quantity = int(content['quantity'])
417             order_size = int(content['order_size'])
418             order_discount = int(content['order_discount'])
419             total_time = int(content['sell_duration'])
420             email = content['email']
421             # Init a session instance
422             session = Session(instrument_id, quantity, order_size, o
423
424             # Add the session instance to the Session Manager
425             sm.add_session(instrument_id, session)
426
427         elif request_type == 'cancel_request':
428             sm.removeSession(instrument_id, 'canceled_order')
429
430         elif request_type == 'resume_request':
431             sm.resume_canceled_session(instrument_id)
432
433         elif request_type == 'customize_request':
434             order_size = int(content['order_size'])
435             order_discount = int(content['order_discount'])
436             total_time = int(content['order_duration'])
437
438             sm.session_manager[instrument_id].order_size = order_size
439             sm.session_manager[instrument_id].order_discount = order_discount
440             sm.session_manager[instrument_id].time = total_time;
441
442

```

Problem handling:

10 of the above 11 problems was fixed after code inspection. Problem 3 of component B has been fixed by now. The reason is that it is related to a lot of code we have written and it do need a lot of time to truly fix this problem. So we need more time. Fortunately, it doesn't influence the main function of our software.