

# Team WINY: Second Iteration Development

Irwin Li (izl2000)  
Wanlin Xie (wx2161)  
Yiming Sun (ys2832)  
Nancy Xu (nx2131)

## Responsibilities

Irwin and Wanlin: Leaderboard, News on homepage, Frontend, Stock information module  
Nancy and Yiming: Game (Registration, create Admin), Testing

## Git repository

<https://github.com/wanlinxie/ASE-marketgame>

Our second iteration is tagged as v1.2

## Static Analysis

<https://github.com/wanlinxie/ASE-marketgame/tree/master/static-analysis>

### Fixed:

Exactly one space required after comma  
No space allowed around keyword argument assignment  
Exactly one space required after :  
No space allowed after bracket  
No space allowed before bracket  
Unnecessary parens after 'if' keyword (superfluous-parens)  
Trailing whitespace (trailing-whitespace)  
Exactly one space required around assignment  
Final newline missing (missing-final-newline)  
Multiple imports on one line

### Not Fixed:

Line too long  
Found indentation with tabs instead of spaces (mixed-indentation)  
Wrong continued indentation  
Unused-import  
Missing function docstring  
Too many local variables  
Too many nested blocks  
Too many branches  
Too many statements  
Invalid-name  
Wildcard-import  
Wrong-import-order  
redefined-builtin

## Code Inspection

Reader: Irwin Li, Recorder: Wanlin Xie

We picked the `games()` function in the `server.py` file as a module to examine, since it is an especially long function. One major problem we noted was that `global_id` is updated for every result in the games database when the `games()` function is called. This is a race condition, since if a user accesses the games tab, and the internet suddenly freezes or the user purposely stops loading the page, the `global_id` would be set to the incorrect value. Instead, the `global_id` should be updated every single time the condition that checks for the proper registration time is satisfied. This ensures that the `global_id` will always be updated with the correct value, even if the user freezes the loading of the web page. This error was fixed during code inspection.

We also noted the complexity of the code as another potential error. Because we allowed the admin to create multiple games, there can be multiple games that exist in the database. Thus, we need to check whether the current time takes place during a registration period, a game period, or neither. This makes the code more complex. We considered reducing the number of games in the database to one, and deleting games when they are past their time period; however, since we decided we may need to utilize the information of past games, we decided to leave this code as is.

Reader: Yiming Sun, Recorder: Nancy Xu

We picked the `updater.py` as a module to examine.

One error we found was that we could condense lines 37 and 39 into one line. There was no need to separate them into two lines.

Another error we found was that the `scrape()` function was dead code, and should be removed. We removed the function from the code.

Another error we found was that in order to debug, we utilized print statements instead of logging statements.

### **Code Demo**

We demonstrated our second iteration features including creating Stock Market Games, creating groups to participate and register in games, and ranking teams based upon portfolio performance. We introduced a new module that scraped and constantly updated stock information. In addition, on the main page, we presented trending financial news and the top 5 performing stocks of the day. The stocks displayed on the homepage and listed inside both their and game portfolios allowed for the user to edit their number of shares by buying and selling.

There was a problem with the yahoo finance scraper that was used to display the top 5 performing stocks. One of the stocks was displayed incorrectly.