Advanced Software Engineering
Exception Handlers: Team Assignment 7
Gabrielle Taylor: gat2118
Jackie Lin: jl4162
Leon Song: ls3233
Aaron Ong: ao2591

Second Iteration Development and Code Inspection

Demo: December 7, 2016 with Lin

Static Analysis Output

django-lint

See all djano-lint output in docs/django-lint.txt.

Stocks module

```
$ django-lint stocks/
************* Module stocks.models
W: 22,12:ParentOrder: quantity: blank=False is implicit
W: 24,11:ParentOrder: is_sell: blank=False is implicit
W: 24,11:ParentOrder: is_sell: BooleanField with default=True will not be reflected in
database
W: 25,17:ParentOrder: time_executed: blank=False is implicit
W: 25,17:ParentOrder: time_executed: Uses superceded auto_now or auto_now_add
W: 26,8:ParentOrder: user: ForeignKey missing related_name
W: 47,12:ParentOrder: progress: blank=False is implicit
W:229,1:ParentOrder.__str__: Use __unicode__ instead of __str__
W:173,1:ParentOrder.trade: Method should come after standard model methods
W:234,16:ChildOrder: parent_order: ForeignKey missing related_name
W:235,12:ChildOrder: quantity: blank=False is implicit
W:236,17:ChildOrder: is_successful: blank=False is implicit
W:237,9:ChildOrder: price: blank=False is implicit
W:238,17:ChildOrder: time_executed: blank=False is implicit
W:238,17:ChildOrder: time_executed: Uses superceded auto_now or auto_now_add
W:240,1:ChildOrder.__str__: Use __unicode__ instead of __str__
```

These issues were all stylistic, so we did not address them.

Authentication Module

```
$ django-lint authentication/
```

No problems found.

Main Module

```
$ django-lint priceIsRight/
************* Module priceIsRight.settings
W:  1,0: Missing required field 'MANAGERS'
W:  1,0: Missing required field 'ADMINS'
W:  1,0: Missing required field 'TEMPLATE_DEBUG'
W:  1,0: Missing required field 'MIDDLEWARE_CLASSES'
```

These issues were not relevant to our project so we did not address them.

JSlint

See all JSlint output in docs/JSlint.txt.

Progress.js

```
22.1 Use spaces, not tabs.
      case 'C': //completed
22.11 Use double quotes, not single quotes.
      case 'C': //completed
43.11 Unexpected trailing space.
      });
45.18 Unexpected TODO comment.
      return r; // TODO: can't actually return r like this, need a callback function
75.58 Unclosed string.
      "<tr class=\"" + child_class + "\"> \
```

Issues 22.1, 22.11 and 43.11 were fixed. Line 75.58 is a false positive; the linter does not recognize escape characters. We could not implement the feature mentioned in issue 45.28. Note: issues in lines 22.1 and 22.11 were throughout the entire document, but we did not want to paste 200+ lines of “use spaces, not tabs”. See docs/jslint.txt in the repository for the entire contents.

Flake8

See all Flake8 output in docs/Flake8.txt.

Flake8 highlighted around 500 issues. Most of these were use of tabs instead of spaces (as preferred by PEP8). After fixing all tab related issues, 166 issues remained. More than half of these were issues such as:
./stocks/models.py:122:12: E221 multiple spaces before operator

```
./stocks/models.py:128:13: E122 continuation line missing indentation or outdented
./stocks/models.py:202:1: W293 blank line contains whitespace
./stocks/tests.py:1:11: E401 multiple imports on one line
./stocks/tests.py:11:47: E231 missing whitespace after ','
./stocks/tests.py:47:73: E261 at least two spaces before inline comment
./stocks/urls.py:7:2: W292 no newline at end of file
./stocks/views.py:3:1: F401 'django.http.Http404' imported but unused
./stocks/views.py:25:27: E251 unexpected spaces around keyword / parameter equals
./stocks/views.py:30:35: E262 inline comment should start with '# '
./stocks/views.py:81:74: W291 trailing whitespace
./stocks/views.py:82:9: E122 continuation line missing indentation or outdented
./stocks/views.py:85:1: E302 expected 2 blank lines, found 1
./stocks/views.py:95:5: F841 local variable 'parent' is assigned to but never used
./stocks/views.py:97:80: E501 line too long (89 > 79 characters)
```

Issues with unused imports, and extraneous inline comments were mostly fixed. The others were left as is since PEP8 is a very strict style guide, and these issues were all stylistic (not related to control flow or unused imports, which may have indicated some problem). Going forward, we will keep in mind these issues, but it was not worth making those changes retroactively at this stage.

Code Inspection

Component A: stocks/views.py -- integrating backend model with frontend UI

Reader: Jackie Lin
Recorder: Gabrielle Taylor

- Overview of stocks/views.py by Jackie:
    - Essentially a view is the middlepoint between backend models and frontend UI
    - It takes the model data and updates the html/javascript
    - It also synchronizes front end actions (user clicks button) to backend functions (execute trade)
- Main functions:
    - Accepting (and cleaning) forms to create ParentOrder models
    - Initiating trades
    - Retrieving order information for display
    - Functions to add/pause/resume/cancel orders
    - Javascript polling helper functions
- Known Problems
    - Stat calculation messy, should be moved to models - FIXED
- New problems discovered during inspection
    - Resuming orders!
        - Resuming an order starts a **new** thread to execute a trade, instead of continuing the old one.

        This is problematic because each trade() function keeps track of several variables e.g. count for previous failures. These variables impact the behavior of the trading algorithm, for example if we fail many times then we ought to shrink the order size. Resuming a function causes us to lose those counters and revert to default trading behavior, when in fact the algorithm ought to be more nuanced. - FIXED, handled in stocks.models

    - Errors in form input redirect you to ugly HTTPResponse pages - FIXED

Component B: stocks/models.py -- database representation and methods for orders

Reader: Aaron Ong
Recorder: Leon Song

- Overview of stocks/models.py by Aaron:
    - Stocks/models.py is the backend of the stock exchange software.
    - It defines the models and methods required in order for trades to store and execute orders.
- Main functions:
    - Definition of ParentOrder and ChildOrder model classes.
    - Executing the trading algorithm (TWAP).
    - The backend interface between the traders, their ParentOrders, and ChildOrders.
    - Handles the continuous execution of the trading algorithm.
    - Supports views.py to add/pause/resume/cancel orders.
- Known Problems
    - Does not handle integer overflow well. - FIXED, handled in stocks.views
    - Resuming paused orders often resulted in selling an excess of ChildOrders. - FIXED
- New problems discovered during inspection
    - Resumed orders contained a branch of code that did not handle resumed orders. well. The trade algorithm was remodelled to account for previously sold ChildOrders before the ParentOrder was paused. - FIXED