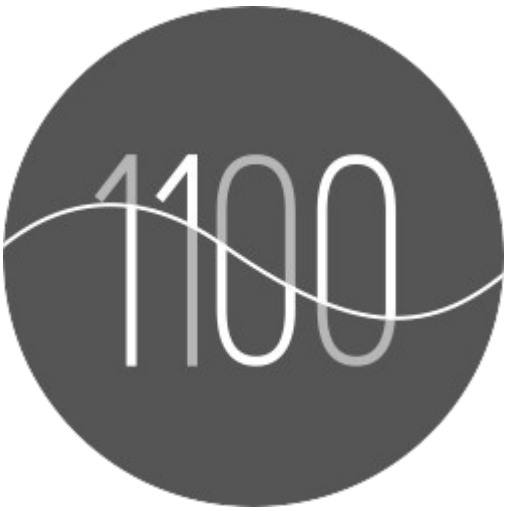
1100 Group

Code Review



CSCC01
Thierry Sans
Dhairya Dave, Andrei Grumazescu, Victor Lee,
Wing Kin Tsang, Jiayi Zhao
Fall 2018

Code Review Strategy

There are 4 main questions each team member will attempt to answer as they perform the code review.

1. Is the code easily understandable?
2. Is the code well-documented?
3. Is there duplicate code?
4. Are functions too big?

The purpose of these questions is to provide team members with a guide on what to look for when analyzing another team member's work. If there can be improvements made to a team member's code, then constructive criticism will be provided. Specific lines of code will be referenced as to where improvements can be made.

Code Review Summary

Dhairya:

I reviewed Jeff's code.

- Overall, Jeff's code in agency_ui is easy to understand as he's separated the main components of his code into various methods. For instance, components such as setting up the widget, selecting an excel file, and also submitting the iCARE template data.
- Another good thing that I have noticed in Jeff's code is that he has also created a gui_helper which he can call whenever he needs to use the functionality of displaying an error or notice message.
- It was a good idea to have a separate file that contains the code that is needed in order to setup and insert data into the database.
- One thing I would like to point out is that Jeff should consider commenting his code more often as that would improve overall readability of his code.
- To end off, Jeff's code is well structured across all of his files that he has worked on and last but not least there is no duplication of code.

Andrei:

I reviewed Victor's code. Victor's code is very to the point and readable. The API is easily testable which is helpful. A good choice of variable names and comments helped remove ambiguity or confusion from the use of imported methods that a reader might not understand. More so comments regarding the use of imported methods would be a helpful addition at the bottom where a large bulk of imported methods are used. A docstring would also be helpful. Those additions aside the code is well done.

Victor:

I reviewed Kelvin's code. His work was understandable and well-documented. Functions were not too large and they did what they were supposed to do. There was no duplicate code to be found in any of his work. One of my recommendations for Kelvin was that he should add a docstring for his printDB function, located on line 56 of query.py. He included docstrings for all of his other functions, which made them easy to understand. My other recommendation was that he should remember to remove the if __name__ == "__main__" statements on lines 55-70 of query.py and lines 29-34 of exportFile.py before pushing these files into the master branch.

Kelvin:

I reviewed Andrei's code. His code is easy to understand and well-documented. Proper and detailed comments are written for every major steps in the functions, appropriate name convention is used to make variables name understandable. There is no duplicated code, dead code or unnecessary debug statement in his work. However, some changes could be made to further improve the documentation, docstring can replace the function comment in the code, which not only states the use of the functions but also gives description for parameters and return value. Also, for long import module name, 'as' can be used to save time and enhance the readability of code, for example, 'import mysql.connector as sql', and use 'sql' to call the module in the code.

Jiayi:

I was responsible for reviewing Dhairya's code. First off, his code is well-structured and works. It can run as a standalone application, which is good for testing. Some changes that can be incorporated is to change his current GUI class to inherit from QWidget rather than QMainApplication. QMainApplication are hard to integrate into our current application and a QWidget is much easier to include anywhere in the main application. Other things that can be improved is to include an sql file that can be used as reference for storing user profiles in the database and commenting code for readability.

Code Review Debriefing Meeting: https://youtu.be/q93FxsOJpMU