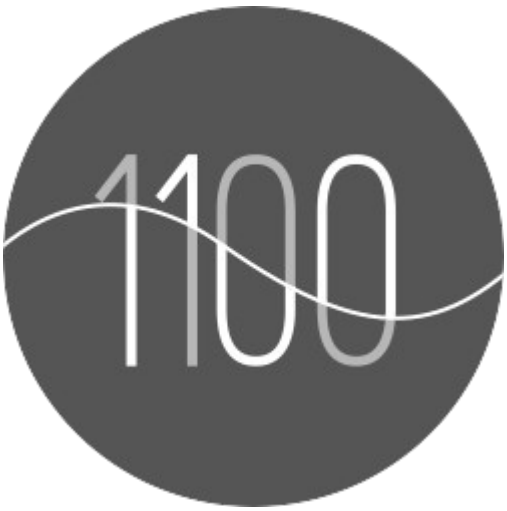
1100 Group

Sprint Report 05



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 TeqQueryWidget.py is easy to understand as he's separated the main components of his code into various methods. For instance, components such as exporting the data, populating the table, and running the query.
- 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 was responsible for creating the interfaces for adding/removing preset queries and adapting the teq interface to be able to create graphs. Victor uses obvious names to represent elements in his interface, and through the use of layout methods makes it very easy to understand and even modify. Victor also includes comments to help readability, but more comments to instruct how to connect each element to the API would be helpful. Docstring is not needed in this case, and the interface code does not contain any unnecessary additions or additions which break our design principles.

Victor:

I reviewed Kelvin’s code. Kelvin was responsible for converting the data from the iCareTemplate database into graphs. In his file graph.py, he gave descriptive and easy to understand names to all of his variables. His code was well documented and easy to follow. The only reminder I had to give him was to remember to remove the if __name__ == “__main__” statements at the end of this files.

Kelvin:

I was responsible for reviewing Andrei’s code. Andrei worked on the API function for preset query and the user interface for preset query and graph generation. Overall, his code is well-structured and well-documented. But one thing that can be improved is to generate docstring in each function to enhance the readability by giving the functionality, arguments and return of the function.

Jiayi:

I was responsible for reviewing Dhairya’s code. From the last code review, Dhairya has moved from QMainApplication to QWidget, which makes integrating his code into the main application near seamless. His code is still well-structured and works. It can run as a standalone application, which is good for testing. Some changes he could make is decouple database related functionality from the GUI code.