FIT2107: Assignment 3

Part A: Checklist Preparation

Nicholas Chong (29808146)

1. Does the program fulfill the functionalities specified in the document "FIT2107-S2-2020 Assignment 2: Automated Unit Testing, Whitebox testing and Continuous Integration" in the section "The Calendar Module"?
2. Are the external modules and dependencies (e.g. Python, datetime, pickle) using the right versions and are they being kept consistent using some form of version management (e.g. requirements.txt)?
3. Is each function written within an appropriate length (less than 125 LOCs) and is easily understood within 10-15 minutes?
4. Does the commenting of the program align with the PEP8 style guide?
5. Does the naming convention of the program align with the PEP8 style guide?
6. Does the code layout of the program align with the PEP8 style guide?
7. Are there any global or shared variables and if so, are they carefully controlled?
8. Are there any hard-coded values in the program, and if so, are they made into named constants?
9. Has there been a testing module set up for each file in the module?
10. Does every function in the module have accompanying test functions that ensures the correct behaviour of the said function, with respect to black-box methods?
11. Does the code satisfy 100% line coverage and 100% branch coverage?
12. Does the program handle errors safely and if not, does it display error messages to the user for debugging?
13. Is the Calendar API used correctly, with respect to the Google Calendar API documentation?
14. Have proper continuous integration measures been set up for this program, including automatic testing and coverage calculations?
15. Has sufficient effort been made in preventing the pushing of any unwanted files, such as sensitive credentials, external library source code, caches, etc.?
16. Have mocks been properly utilized for calls to the Google Calendar API, returning dummy data and verifying the API's usage?
17. Does the user interface in the program contain no possibility of an infinite loop, exits gracefully, and handles correct and incorrect user input appropriately?
18. Is the code efficient with proper usage of the API, correct handling of data, proper handling of the user's input?