

## Code Review Model

Moderator: Michael

Author: Rafi

- The code is mostly understandable to read.
- The code has some documentation, could use more regarding the description of each method.
- The purpose of the class is not described but the purpose of the class is understood.
- Method and variable names are descriptive enough to know the purpose of each.
- Naming conventions are consistent.
- The methods in the class do belong there, they access, add, and remove required data from the database.
- Methods that return values are returning the correct values.
- Exceptions are being checked in try/catch blocks in multiple methods.
- The methods are not returning any known errors or exceptions, though unknown errors may exist.
- The code is testable on its own.
- There are multiple test cases in a separate file, not all methods in the model are tested.
- There are an adequate number of cases for the methods that were tested, but an inadequate amount for the rest.
- Is the testing methodology the same as the rest of the project? If not, is there a reason why?
- The code is fairly clean and organized.
- Some code isn't being used since other features haven't been implemented yet, but all features are necessary.
- Variables are using classes that make sense for them.
- Appropriate libraries are being used.
- The code performs as well as it should.
- Methods have low time complexity in the case the database is small, there could be cases where it is higher.

- Space complexity is also low when the database is small, but could be much larger for some methods in the case of a large database.
- The style of code is consistent.
- The style of the code is similar to the rest of the backend code.