

## Code Review

Moderator: Rafi

Author: Michael

- The code is very understandable to read.
- There is plenty of documentation. Although, there could be some for the global variables, and a little more detail in the allShifts (Schedule.java) method, like how the string is parsed, and why the string starts with allShifts
- Although its easy to tell what the class is doing there is no description of the utility of the class.
- The naming conventions are pretty consistent, except for a few of the parameters. Like shift\_id in the editShifts method (Schedule.java) could be shiftID for consistency.
- The methods in the class do belong there.
- The methods do return the correct values.
- Error and exceptions are handled with in the class. It would be better if the caller received some error or exception when something happens to make debugging easier and save more time.
- Errors and exceptions are not being thrown or returned.
- The code could be tested using the allShifts method for validation.
- There are no tests.
- There are no adequate number of tests.
- There are no methodologies.
- The code is clean. Good and consistent spacing and indentation. The braces are 95% consistent.
- Some of the code could be removed, like getShiftsbyWeek (Schedule.java) because it would be faster to have it be done in the client if they always have a copy of the shifts.
- The times could be integers this would help prevent errors when calling the methods.
- I don't think there are any other classes that could be used to increase efficiency of the class. Although, I'm not sure if the Date class is being any more useful if the date is already in the form "YYYY-MM-DD".
- The code performs well enough.
- The time complexity of the code is low.
- The space complexity of the code is low.
- The style of the code is pretty consistent with the back end.

- The naming convention is consistent through the project. But the indentation spacing, and brace placement is inconsistent.