Refactor Document – Team 2

Refactor #1: Long method

As mentioned in TA's comments for iteration 2, the method initFrame in GymManageMainFrame class has more than 60 lines. Although the code is stable, we don't want this method to be too long. So, we decide to refactor the method and divide the methods into multiple smaller methods. We tried our best to maintain the functionality after refactoring. However, we can't minimize initFrame method within 20 lines. Since the purpose of initFrame is to create the interface after logged in. We found it might be more convenient for us to concatenate all creations of buttons into one method.

It took us a long time to complete this refactoring. Since multiple buttons are linked to this method, we have to relink the buttons after we break initFrame. As a result, we decide to leave initFrame as more than 20 lines since the code is already stable.

NOTE: Some other methods for interfaces are more than 20 lines as well. Since the code is already stable, we decide to not modify them since as discussed in lectures, extra modifications lead extra risks which sounds unworthy for now. We prefer stable code although the code might smell a little bad.

Refactor #2: Missing Getters and Setters

In iteration 3, we implement equipment page which enables the administrator to view equipment status. At the beginning of iteration 3, we forgot to create getters and setters for equipment class. The attributes stored in this class thus was inaccessible. This is why whenever we attempt to modify database, the updated information is never visible on the interface. Missing getters and setters make the code smells bad. We refactor the code by adding getters and setters for all the attributes in equipment class. As a result, the problem gets resolved.

Refactor #3: Inaccurate Comments

There are multiple comments we wrote during iteration 2 that are not accurate enough. Wrong comment makes code smell bad. Take the initFrame method in class GymManageMainFrame as an example. We originally commented this method as "input username" while a better comment should be "Administrator operation page, this class extends from KeyFrame and should be able to allow administrator to do some operations", It is easy for the one who creates this method to understand but other team members will be definitely confused. Sami was trying to test the functionality of interface. He got confused since he can't find the method that creates administrator operation page. Thus, we decided to walk though and refactor all comments. As a result, we rewrite comments for most of the classes and the comments are direct and accurate now.