Team 10
EECS 448
04/24/2022

Code Review

In order to do the code review, the team met up together on one day and decided to use the walkthrough method. With that in mind, the medium-size code segment we decided to review was two classes of our main script file: the Server and Channel classes, responsible for distinguishing different places where users can chat. We would have wanted to review more code than that for this assignment, but the requirement was a medium-size segment, and since the script we have consists of a handful of classes, reviewing more than two might have made up at least half the code.

The walkthrough method consists of two stages:

1. Preparation, where participants are expected to review the artifact before review.
2. Team analysis of the artifact.

In order to do the preparation, after having agreed on what code segment to review, each team member read through the intended code silently and individually for a few minutes, and agreed that we then we all share our thoughts on what faults we thought we found. However, to make sure everybody was on the same page about what a fault was, we all had to refer to a previous lecture's notes that described the definition of faults, different types of them, and why each one of those is a fault. This would make the walkthrough participant-driven, not document-driven.

For the second step, after finishing reviewing the code individually, we each brought up a list of potential faults, why we thought they were such, and how to fix them. At first, the team had their differences on some of the proposed ideas, as we were unsure if they explicitly met the definition of a software fault. Therefore, we had a back-and-forth discussion on the matter, while referring to the aforementioned lecture notes.

After having settled on what the actual faults in our code were, the team came up with the following list of dispensables which were the primary faults we found in the Server class:

- The use of the variable department in the constructor, which we never actually utilized, so it was an unused variable whose absence would make the code cleaner.
- The getter method associated with department.
- The following getter methods: name, channels, and users. Those are all methods that we initially thought we would use to access the class's member variables, but ended up not using at all.
- The method setName was also not used because we did not implement or intend a feature to edit servers' names.
- The methods addUser and removeUser. The process of adding or removing a user from a server now happens automatically as soon as the user joins or leaves, without the use of these methods.

In the Channel class, we found the following dispensables:

- The variable category which we had not utilized at all, its getter method, and its setter.
- The getter methods name and logs.
- The method editChat, since we did not implement editing messages.

Jasem Ali was assigned to fix these faults simply by deleting the unused variables/methods in the classes and modifying the instantiation of these classes to exclude the removed variables from the constructor.