Report:

Note: Despite the deadline on March 10, for group 2 to send us their repo, we never received any communication from them. We emailed them on Monday March 13, reminding them to send their repo to us, but we did not get a response. We only received the repo link in the afternoon of March 15, after involving the professor to contact the other group. Because of this we missed out on 4-5 days of testing and severely limited our coverage.

Manual Testing:

- We read over their user stories to see what features their application would have, and what to expect. While using the application, we found some missing features and reported them as issues like the Inventory system that was supposed to be in iteration 1 but still doesn't seem to exist in iteration 2. Along with some smaller bugs like values not updating properly.

Code Review:

- We analysed the code base for any smells. We had some difficulties understanding their architecture pattern with the Data Access Object (DAO). We referred to their system architecture sketch however it was not relevant. Due to this and the limited time we had, our review capabilities were limited.
- We also found it difficult to grasp how their application worked because the UML class diagrams seemed to not match the actual project.

In addition, we noticed there were frequent uses of the "Object" type in Java throughout the entire code base. While we suspect a larger architectural design flaw with this app that led to the use of this. We didn't have enough time to fully understand their implementation.

Manual Story Testing Notes

- I don't know what happened to the inventory
    - Was supposed to be in Iteration 1, I assumed it got delayed and then was never added in iteration 2.
- BMI doesn't update automatically, or at least I can't find how to calculate it
    - Again, this was supposed to be in iteration 1, and still doesn't exist in iteration 2
- When logged in as a member, in the 'class schedule' tab you can see all the other classes taken by other members and modify THEIR coach. Not sure if seeing who else is taking what course is a bug, but I don't think User A should be allowed to change the coach for User B.
    - Actually, I don't even know if the members should be allowed to change who their coach is
    - This one seems like an actual issue.
- I can 'buy' a course even if there is not enough money in my account.
- Courses can have negative cost
    - I mean that could be a real thing but I don't know.
- The renovate buttons should be called refresh
    - When signed in as a member in 'class schedule' tab
    - I mean it's pretty nitpicky but I also didn't know what it did
- When signed in as a coach, I cannot figure out what the 'work info' tab does.
    - The refresh/renovate button doesn't do anything, even when the coach should have a class to instruct.

Static Analysis Notes

In bean.Member.java in the Member constructor with the parameters, the variable 'Fund' should probably be lower case like 'this.fund'. Right now it looks like Fund is a class.

In bean.Schedule.java the public constructor that takes no parameters doesn't seem to be used anywhere. Recommend removing it.

- Same for all of the other classes under the bean package

In view.baseview.KeyFrame.java on line 21, I would discourage the use of magic numbers.

- Same goes for in view.keyframe.GymManagerFrame.java in lines 100 and lines 109

In various classes in the view package, variable naming should be more descriptive. Naming variables "jtf1", "jtf2", "jtf3", "jtf4", etc. This substantially hurts maintainability as it doesn't tell the developer any information about the variable. More evident as there are comments in the initFrame() method (in AddMemberFrame.java) which describe which each variable is.

In various classes in the view package, all the jPanel.add calls are written line by line. It would make more sense to store the components in an array and use a for loop to add them to the jPanel.

In WorkPanel.java in the renovateAction(String account) method. The typing Object[][] is extremely dangerous as the return type can be anything, this defeats the advantages of object types. I suggest changing the return type of getWorkArrayListByName() to something more defined.