Peer review regarding Regarding Roy Nilssons group. https://github.com/rn222cx/1dv607-OOP

Test the runnable version of the application in a realistic way. Note any problems/bugs.

No problems or bugs. Works fine. It is possible to add new member with empty credentials. This can be considered as a bug.

Try to compile/use the source code using the instructions provided. Can you get it up and running? Is anything problematic? Are there steps missing or assumptions made?

Code compiles without errors. Instructions are good.

Does the implementation and diagrams conform (do they show the same thing)? Are there any missing relations? Relations in the wrong direction? Wrong relations? Correct UML notation?

The design model is very good and easy to read. Correct UML notation. All relations seems to be in right and in the correct directions. Very good!

The sequence diagrams are clear and very easy to understand and follow, beautifully rendered and with correct object destruction messages(1, p.230).

Is the Architecture ok?

The software architecture has a correct model view (and controller) separation of classes, with no model classes coupled to the view.

Is the requirement of a unique member id correctly done?

The new unique member id is correctly done.

What is the quality of the implementation/source code?

Outstanding quality of implementation and code. Easy to understand naming, no unnecessary repeating or dead code (1,2).

What is the quality of the design? Is it Object Oriented?

Excellent quality of the design - object oriented with object associations. Classes are not overly bloated. Dependencies are easy to understand and straightforward. The GRASP patterns are followed(3, p291).

As a developer would the diagrams help you and why/why not?

It would definitely. It describes flow of application very well.

What are the strong points of the design/implementation, what do you think is really good and why?

Good separation of model/view responsibilities. Focus is on model, and it seems easy to reuse. The part of application with View is good enough. It can be improved, but it is obvious that the focus was not on that. Implementation is very demanding. Developer should know very much about C# and .net in order to understand design/implementation.

One example: When adding one boat for existing user it should be good to see simple member list before we need to choose existing Member ID.

What are the weaknesses of the design/implementation, what do you think should be changed and why?

There are no visible weaknesses. The model is done better then the view, but that was stated in requirements.

Do you think the design/implementation has passed the grade 2 criteria?

Yes, very much so.

References:

1. CSharp coding conventions, https://msdn.microsoft.com/en-us/library/ff926074.aspx
2. Naming Guidelines, https://msdn.microsoft.com/en-us/library/ms229002(v=vs.110).aspx
3. Larman C. Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062