Peer Review of Alexandre Driaguine group ad222kr

Written by Frida Holmström fh222dt

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

Work well, no bugs noted. I have added members, added boats, changed members and list them as well.

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?

Works to compile according to the instructions, but all of the breakpoints needed to be removed before I could get started ☺.

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?

I assume the class diagram is reversed engineered in VS, and it looks good when compared to the code. A bit cluttered maybe with all attributes and operations opened. [1, p252, ch16.4]

The sequence diagrams are well done as well. They show the different events for each use case clearly. [1, p177, ch10.5]

Is the Architecture ok?

The overall architecture is well made. Perhaps you have a too high cohesion with very little code spread over many classes. For instance you could put all of the boat views in one class, as well as the member views, instead of the class inheritance. [1, p291, ch17.8]

Is the requirement of a unique member id correctly done?

Yes, you check what numbers has been used and add to that.

What is the quality of the implementation/source code?

The code has good standard. Not much duplication and good naming. [2]

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

The code uses GRASP in a correct way. The classes have high cohesion and low coupling. [1, p277, ch17.4]

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

Yes they help me understand the application much quicker than to look directly at the code. [1, p176, ch10.3]

What are the strong points of the design/implementation, what do you think is really good and why?
All objects are connected by association. This makes the code more understandable. The use of GRASP is also well done. The project is well modeled and then implemented in code. [1, p277, ch17.5]

What are the weaknesses of the design/implementation, what do you think should be changed and why?
Too many classes, as I mentioned above. To me it would be even easier to understand the code if it wasn't this many classes to consider. [1, p314, ch17.14]

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

References

1. Larman C., Applying UML and Patterns 3rd Ed, 2012, ISBN: 978-81-7758-979-5
   (This is not the same edition as listed by the teacher, therefor I have also added chapter references.)
2. C# Coding Conventions, https://msdn.microsoft.com/en-us/library/ff926074.aspx