

## **Review for Daniel Svensson**

### **By Rasmus Sjostrom, John Herrlin**

The architecture is good and simple, with a few minor issues. One thing that we think is the most significant mistake with the application is the fact that the controller is dependent on the view. We feel that even though the model is not coupled to the view, having the controller being dependent on it could make things complicated when it comes to for instance changing the view to a completely different one, say a GUI.

Other than that, the diagrams and the relationships between the classes are easy to understand and loosely coupled. One thing to consider is to transfer some of the responsibility in the Member class over to the controller; the add/delete boat functionality for instance [1, section 25.2]. The boat class is a great example of proper responsibility and good encapsulation.

One of the other minor issues is the naming of variables, which are at some parts good and some other worse. There are plenty of variables that are being named with just a letter, which might not be a huge problem in this particular application, but a variable name should always indicate the intent of its use [2, Variables].

The code in general are however very clean and easy to read, which is a big plus. As mentioned previously we had no problem understanding the relationships between the classes, using the provided diagrams and source code.

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

Yes, as they are very clear and simple they would help in the implementation of the application.

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

A really strong point of the application is that it does what it is supposed to do and nothing else, with low duplication and loose couplings. That makes it easy to read and maintain.

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

We think that the controller should be completely independent as well, since that makes the application more flexible when it comes to further developing it.

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

Yes. The design meets the requirements and is well done.

#### References:

1. Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062
2. [https://en.wikipedia.org/wiki/Naming\\_convention\\_%28programming%29#Java](https://en.wikipedia.org/wiki/Naming_convention_%28programming%29#Java), online, last checked: 9 oct, 2015.