Peer review To: Lars wöldren (lw222ii)

Compiling and usage

The models, implementation and all the documentations are easy to understand and to follow as it is presented.

The runnable version of the application works fine but there are some issues to be fixed.

Does the implementation and diagrams conform

Yes, the implementation and the diagram shows the same thing.

Is the architecture ok

There is model view separation

Yes, there is model view separation.

According to Larman C[1, p 329] model-view separation principle non-UI objects should not be directly connected to UI objects.

Is the model coupled to the user interface?

No, the model is not directly coupled with the user interface

Is the requirement of a unique member id correctly code

It is hard to say, because we could not be able to understand the unique member id code.

What is the quality of the implementation/source code

- Code standards

The code is well indented and easy to follow.

- Naming

The classes and the methods have good names and easy to understand what they supposed to do.

- Duplication

There is no duplication. All the methods and the classes are where they supposed to be.

What is the quality of the design? Is it object oriented?

Since objects are connected using association and classes have high cohesion we can say that the design is object oriented.

As a developer would the diagram help and why/why not?

Yes, since the diagram presents the requirements and has the needed conceptual classes it is easy to understand.

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

The strong points are :- Model view separation which allow separate development of the model and user interface layers. In Addition, model view separation allow easy porting of the model layer to another user interface framework.[1, p 330]

A Sequence diagram shows sequence or time ordering of messages.

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

There are some issues that should be changed, for example

A person can register

- without any personal number
- Can register with the same personal-Id and name as many times as he likes

Edit information (add boat, delete member,update information) does not update the one we wanted to update, instead it only updates, add or delete the member on the top. Moreover the boat size is only limited to 20 could not understand why?

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

Except that the issue we mentioned earlier, the design has the necessary requirement we think that it will pass grade 2 criteria.

References

[1] Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN, 0-13-148906-2