Workshop 2 Peer Review of ss223ck

Architecture:

There is a model view separation. The view uses the model for read only stuff, as asking for a list of users E.G.

The model does not return formatted strings, it sometimes return a bool to the controller.

We feel that the code standard is okay as far as naming goes, it's clearly stated what the functions actually do. However, there is a lot of duplication especially when it comes to initiating objects such as UserDAL, CrudBLL etc. This is done locally in methods instead of being done ONCE and kept in a field in a class that is supposed to use it. This breaks DRY and makes it very hard to be able to see what classes use what classes since it does not show up at all in a class diagram.

Boats are selected using the Primary key from the database.

GRASP isn't being followed as the view handles system events, such as creating objects and sending to the controller. Instead the controller should fetch the data, create an object and pass it on.

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

They would be helpful. They clearly show how the program executes the requirement.

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

Good naming, simple and easy to read code.

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

The separation of Memberinfornmation and User seems uneccessery as "User" could contain everything, the properties of a user and a list of boats.

As we've mentioned the locally initiated objects of classes is a problem as it does not clearly show the dependencies and breaks DRY.

An example:

When a Member is to be updated, a list of members is fetched and is only used for checking if the personal number exists anywhere in the list of members. Instead of picking out the member object that is to be changed, the view is called and a completely new User is created. It could've been connected more clearly by picking out the Member and sending it to the view, makes for more readable code and connects the classes more clearly, in our opinion.

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

We don't think so at the moment. We feel that some responsebility should be moved from the view to the controller (initiating class objects as mentioned above), add fields to classes and initiate objects in a constructor (dependency) to show clearly what class is connected to which classes and to remove a lot of duplication.