The architecture

- There is a good separation between MVC.
- The view is handling the interface.
- The model contains a little bit of interface(with the exceptions), but it's still ok MVC.
- There are no domain rules in the View.
- The requirement of unique id is having some flaws but still working. You should create a new random number if the id already exists.
- The quality of your code is good. Naming is really great. No duplication or dead code.
- Your code is object orientated and using grasp correctly.
- Your classing have high cohesion, low coupling, not using any static variables and avoiding hidden dependencies - that's good.
- You have done well with encapsulated information.
- Your program is inspired from your Domain Model(from workshop 1).

Your diagram would not help us as a developer, because it's hard to understand.

Strong point with your code:

- Good namning.
- Great overall structure and code.

Weak point with your code:

- Randomize the id isn't perfect.
- Can't delete boats

Aside of the deleteboat-method not working we still think that you have PASSED 2 criteria!

Your diagram: [Reference 1.]

- Hard to know what the colors means in the class-diagram
- Your class-diagram Menu doesn't have an dependency with Member, boatType, but in your code Menu is getting a memberlist, a boattype and other objects.
- The boat doesn't have any association with menu, maybe you wrote the arrow on the wrong way?
- The member-class have a association with a list of boats.
- Your DAL have a list of members but no association to the class Member.

1. Vaughn Vernon - https://vaughnvernon.co/?page_id=31 (2015-10-08)