

Assumptions:

Boat can have a Register ID:

This is not in the requirements but I thought it could be nice to add in case the municipality needs to have access to the register in case of accidents as mentioned in Workshop1. However it is not a required field for the program, since I don't think that a canoe for example has a registration number so the program will work just fine when leaving this field blank.

The boat type motorboat is added:

Think that this is something that should be added. But since the boat types is an enumeration class the program will work fine just to add or subtract the boat types desired. However it will not work to load an XML that has a removed boat type in it.

View member list:

The normal view is shown at startup in a TableView, and by using a radio button you can switch to the verbose view. At start only the personal ID will be shown, but if mouse pointer is over a members first name, the detail of all members boat will be shown. This should be moved to hover over members boat count since that makes more sense. But because this is a integer there have been some implementation problems of that, so for now the user has to hover mouse pointer of members first name.

View specific member:

To view a specific member info you will have to use the edit member. There you can both see the specific member info, as well as change it. You can also add, edit and delete boats of a member.

Worth noticing is that any boat change will be changed even if you press the cancel button of the edit member view afterwards.

Testing:

The program has been pretty well tested but no structured document for this has been created. I cannot manage to make the program crash in this version, and all members and boats are saved properly. However, I think it could be nice with some test report to get a nice overview and by so be able to analyze if all aspect has been thought of.

Executable:

The executable .jar file should run on any computer that have a modern Java version that can run Java-FX. It's tested upon both MacOS Sierra and Windows 10.

Other Comments:

I am not really satisfied with the number of couplings, but I have decided that this will do for now. Hopefully I will learn patterns on how to solve this in workshop 3 so that I can make it better for the final submission for higher grades. I think that only adding a MVC pattern could help a bit. But that is if keeping the controller to one Class, maybe to keep high cohesion that would have to be split up and if doing so, it would be high coupling again.