

# Backlog of changes

## Feedback code:

1. Naming inconsistencies among fields and methods. At one or two places the naming consists of only one letter words such as string p et cetera.
2. Model-View separation is not clear. The suggestion was to add a folder structure to the project to clarify this.
3. The Controller class and the MemberRegister class have functionality that should be handled by UI such as printing out messages.
4. Too many question such as "Is this your correct name (y/n) ". Which only added clutter to the program and not much other payoff.
5. The Controller class should be divided into couple of more classes by for example separate boat and member controller. This should make the code easy to follow and maintain.
6. No Jar file

## Implemented changes in the code:

1. We removed several unused variables to clean up the code. Furthermore all of the methods should have correct names to the naming convention rules. The names of the variables have also been update to make the code easier to understand and use.
2. To fix the Model - View separation we added a new folder structure which follows:  
**Model**, where the Boat, BoatInterface, BoatTypes, Member, MemberInterface, and the MemberRegister classes are located. **View**: View.java. **Main**: Main.java  
**Controller**: Controller.java.
3. We moved a lot of the printouts from the Controller, as well as the MemberRegister to the View class.
4. We rewrote the addMember() method in the Controller class now doesnt have so many yes-no questions since the peer reviewer thought it disrupted the flow.
5. We discussed this change as a group, however felt like splitting the Controller class in a BoatController, a MemberController as well as a FileHandler class would not be necessary and might be more confusing and not a better design choice. We preferred having one controller to handle the functionality of the program.
6. We now added the a jar file and updated the readme file to make it even more simple for the user to run our application.

## Implemented changes in the diagrams

The sequence diagrams and the class diagram has gone thru a few changes. The most obvious ones are the added packages for the class diagram, as well as a better structure and design which should make the diagrams easier to follow.

**Sequence diagram:** The sequence diagram has gotten rid of the clutter before the important functions starts, such as mainMenu() and it is also assumes that the user enters correct inputs to get a more clear structure, this was something that the feedback told us to fix.

For some updates the book "Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062" by Larman C was used. Inspiration from figure 15.7 (page 357/960) and for returns figure 15.8 with one

difference that the return message for getters wasn't written out. The texts over the arrows is more on an abstract level so it should be easy to understand what is being sent.

**Class diagram:** The Class diagram got fewer details such as removing help methods for Controller, getters and setters in most cases. The Viewer does not have any attributes or methods since the Viewer does not affect the program it is only visual help for the user when entering data to the program.