

# Review – Christoffer Holmgren.

## Code Quality/Standards

- It exists 7 index.php files in your project which contain pretty much the same code.
- We found a class named Model that is empty with an empty constructor which is **dead code**. We also find code that is commented out in a few of the index.php files.
- Inconsequent naming, sometimes a boat club member is referred to as member and sometimes as user.

## Code Design

- There are multiple string dependencies between the index.php, the views and the controllers. An example is in the ListUsersView which defines names for the methods on Boat which is read in the BoatController.
- It is good that objects are connected using associations instead of keys/ids.
- In User and BoatList you have getters and setter, however the class member variables are all public.

## Diagram

### Class Diagram

- The class model shows that different views implement/realize View, but in your code they inherit/specialize view.
- We also found some classes that are modeled as a dependency but in reality is an association.

### Sequence Diagram

- Your sequence diagrams confused us a bit since many of them show that the class that start the application is referred to as User. The confusing part is that the actual User class is not what you intend in the diagram, rather it seems like you refer to the User folder that contains the index.php file that starts the application. This is just a specific example since this problem exists in all the sequence diagrams.
- In Compact List the ListUserView does a few important operations that is not shown in the diagram.
- Since the application displays both compact and detailed list to the user, you should replace the diagrams Compact List and Full List with only one diagram that in which the ListUserView calls both compactList and fullList.

## Note

The public version crashes on edit, remove and show user.