Peer review - Robin Karlsson

http://1dv607.rpkn.se/workshop/2015/10/07/Workshop-2.html

Architecture and Diagrams

Association role names are very good. But some attributes and operations could have been included in class diagram (1, p. 250 section 16.1). Still good that diagram is not cluttered with too much information.

Code Quality

It is difficult to give any proper review when we never have been using python ourselves. The code looks very compact and is self-documented. Good with understandable comments. However the naming of `MemberRegistry` could have been better since the name does not really tell that it is the main controller.

Design Quality

There is an association between `Model.Member` and `Model.BoatList` and a dependency between `View.Member` and `Model.Boat` which both probably could have been avoided if the controller got the boats belonging to the member instead.

Miscellaneous

There is no running instruction for Windows. The two links referring to source code is not the same code, it should be specified that it's the release (or other branch) that should be used. The latest code on GitHub dependence on bcrypt, which is not installed per default on all unix-based OS and can be quite tricky to install on Windows. And the password is not provided and the application crash since there is a problem with the salt.

Is the requirement of a unique member id correctly done?

Yes, each member gets assigned a unique member id.

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

Yes, it gives a clear overview of the architecture and the relationships between classes.

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

Strong separation between model and view. There is no domain rules in the views and there is no user interface responsibilities in the model.

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

Since we have no prior experience with python it's hard to tell. A minor weakness could be the, probably, unnecessary couplings between `Model.Member` and `Model.BoatList` and between `View.Member` and `Model.Boat`.

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

Yes, absolutely!

References

1. Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062