Peer Review for Athmar Mubark (am222xc)

by Pär Eriksson and Oskar Emilsson

You have unfortunatly not submitted a complete application for the peer review, the implementation expects a local MySQL server, and there are no instructions on how to it should be set up. We would recommend you to use SQLite instead, as that database is just one file you can easily ship with your application. But we will review the code and design to our best ability.

One of the libraries used for the database connection is also missing, namely the `com.mysql.cj.jdbc.` *`PrepareStatement`* which is imported in the DBManager model. The supplied .jar-file did not work either for the above reasons.

The diagrams are clean and easy to follow, although the class diagram might not be considered correct UML notation. Although the class diagram is not bloated (which is a good thing), it could have had package diagram notation (seen on page 316 in Larmans book `[1]`) to make it even clearer. The sequence diagrams are well notated, and looks like a clean way to implement, though classes are usually notaded with a colon (:) in front of it `[1, p348]`.

The implementation however does not seems to completely follow your design, as there are hidden dependencies in `IControl` for example, where there the `input_member` and `input_boat` from the view are used directly, and that isn't depicted in the class diagram. There are many inner classes in IControl that aren't visualized in the design (though the book does not go into detail on how to show that). Some classes are also nested which does not seem as a best practice for us.

Names of variables and methods inside the classes could be a lot clearer, names like `my_input2`, `the_view1` or `saveListener2()` does not tell a fellow developer what they are or what they should do. When naming a class prefixed with an `I`, such as `IConsole` or `IControl`, it usually implies that it is a class interface (not UI), which where a little confusing at first.

The relationship between boats and members does not seem to be strictly object oriented design. By the looks of the implementation their relationship seems to be saved only in the database, the member object has no knowledge of the boats they own. The enum in the boat model seems unnecessary, since the enum is just converted to an arraylist in the getter method.

The code is generally difficult to follow and hard to understand, much because of the naming conventions but also because of the many inner classes. Unfortunately we think that the design and implementation in its current state might have trouble to pass grade 2.

References

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