

Peer review for dv222bk by jr222er, lm222ix and aa222zh

Runnable version

The runnable version of the application passes all of the requirements and did not crash while being tested.

Implementation and diagram correlation

In the MemberList you are instantiating BoatList and should therefore have a dependency between them.

As stated by Larman in [16.11 p. 396] dependencies should be noted in the class diagram between the User class and Member class considering the Member class is used in some parameters of methods within the User class. However, we feel that the diagram is already quite complex and perhaps adding further dependencies such as this one would cloud the image.

The dependency between Console and Boatlist is unnecessary considering that you are just using the Member class to access the Boatlist, BoatList is never directly used from Console.

There are some additional dependencies to add from User to Member and User to Boat. They are used as parameters in multiple methods in User.

Apart from this the class diagram is well documented and shows many of the dependencies and associations that we would not have found so easily while just looking at the source code.

The sequence diagrams looked good and correlates well with the implementation.

Architecture

The application follows a Model-View separation according to the two principles stated by Larman in [13.7 p. 392].

Code quality

The source code follows a good naming practice and contains comments with enough information. The source code also follows the principles of encapsulation.

Design quality

Objects are connected using associations and not using ids. The classes have a high cohesion and the elements within the classes are strongly related to each other. The application may suffer from some high coupling considering there are quite many dependencies.

The classes that are found in the application and the domain model from Workshop 1 have a strong correlation to each other.

Conclusion

As developers the diagrams would give us a much clearer view of how the application works. Some confusion would probably occur due to the missing dependencies we noted above, the Enums typed as separate classes and the dependencies that should be removed.

Strongpoints

- + The sequence diagrams show a strong correlation to the implementation and are easy to follow.
- + The class diagram even though has some faults is well documented and easy to follow
- + The source code follows a good practice when it comes to naming, commenting and encapsulation.
- + The application succeeds on following a clear Model-View separation.

Weakpoints

- + The class diagrams has some unnecessary dependencies and are also missing one dependency as stated in the review.
- + The classes may suffer from some high coupling.

We believe that the application has passed grade 2. The application follows a Model-View separation, contains good sequence diagrams. It may suffer from some high coupling but this is not enough to not pass grade 2. Regarding the issue of high coupling, we could not find an example of what to regard as too much coupling. However, in our opinion many of the dependencies could have been avoided and as such we consider the application to suffer from issues with high coupling.

References

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