

## Peer review for

David Andersson, Mattias Gustavsson and Robin Stempa

by Pär Eriksson and Oskar Emilsson

Your domain model is very clean and easy to follow, with well named conceptual classes, which makes it easy to follow. It gives a good picture of the system and it's a good starting point for developers to understand the domain. There are, however, a few things that could make it even clearer as a domain model.

Larman says that domain models are meant to represent real world objects and actions that exists in the domain [1, p224], so the Login and authentication should not be coupled to the Member, as `Login` is not a real object. Perhaps login/authentication could be its own domain. In the real world, the authentication could simply be to have the key to a door, and might not be necessary to understand the domain at hand.

The associations could be a little bit clearer with regards to direction, and some association names seems code focused. For example, `Berths-read-Boat` is particularly unclear. Instead, maybe a boat could occupy a berth, or a berth could be booked for a boat. A conceptual object does not necessarily need associations to other objects in the domain. According to Larman, associations should exist where we “need some memory of a relationship” [1, p246].

As a last thought, the `eventList` in the calendar could possibly be derived from the information in the events.

But as the domain model is now, we think a domain expert, such as the secretary, would indeed understand the model, and your model may very well pass grade 2. Good job!

## References

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