Reviewers: Fredrik Wällstedt • Meri Stakovska
Reviewee: Marcus Rosén

Peer Review

Your domain model is spacious and gives an good and clear overview of the conceptual classes you have chosen. It appears that you have used linguistic analysis to identify candidate classes based on the keywords supplied in the requirements.[1] You limit your modelling to the requirements and exclude irrelevant and out of scope features based on speculations in accordance with the guideline “think like a mapmaker”. [2] We think your model should pass the grade requirements after some minor changes.

You have used some concepts that are not implicitly mentioned in the requirements but are clearly motivated. One example is BoatType which is a “description” class that encapsulates the concept need to systemize the actual boats in the system.[3] A boat type exists in the system without any boats being added.

In our opinion the BoatImage class is an example of that not every noun or noun phrase make for good conceptual classes. In our opinion BoatImage is an attribute to a Boat and does not exist if a Boat is not registered it has a purely descriptive nature like a number or text. We agress with your decision to make Event a conceptual class, since it is made up of several primitive attributes.[4]

Larman states that it's not necessary to include all associations in the domain model however regarding the associations in your model we draw the conclusion that some representations of the requirements 11 and 12 are missing. A member and a secretary is to be able to view/list the calendar events - this would not be evident to a domain expert. According to Larman if there is a need to know of some memory of the relation between classes, an association is required.[5]

Association between Member and its Boat (regarding changes to the BoatDetails)
The model could benefit from adding some associations between classes. Classes are not restricted to having single associations to other classes, and some of the classes in the domain model could benefit from having associations to other classes added. E.g., the Member class has a relation to the Calendar class stipulated in requirement 12. This could be made clearer by adding an association.[6]

We are not convinced that there should be a 1-n relationship from Booking to Berth. Should a specific booking be able to refer to several berths?

As a developer it is easy to understand the majority of the associations in the model and the conceptual classes. However the interaction between some of the classes are not evident e.g. the relations Calender-Member, Calender-Secretary and the components of Boat.

---

[1] Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062, page 141
[2] Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062, page 145
[3] Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062, page 146 ff, 165
[4] Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062, page 146
[5] Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062, page 150-151
[6] Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062, page 155

According to Larman it can be beneficial to a model to include multiple associations between classes in some cases[7]. We think adding these in the places we have mentioned would add to the clarity of the model.

Perhaps you've taken for granted the authorization/login state requirement however since it is through authorization that the role of the user is assigned we therefore regard it as essential to the model. This would also inhibit us as developers to understand what it is that the domain experts wants.

And lastly a minor notation detail, the associations should be hyphenated e.g. "Allocated to" should be "Allocated-to".[8]

References

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

[7] Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062, page 155
[8] Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062, page 152