Linnæus University
Course: Object Oriented Analysis and Design (1DV607, Autumn 2016)

Peer-Review of Workshop 1 (grade 2)

Reviewed work by: **Carl Johansen**, **Anton Lindmark** & **Fredrik Hall**
Reviewer: **Mitja Tim Rijavec Bruneus**

Overall evaluation

The presented Domain Model is simple and very easy to read and understand, probably even by a person not familiar with UML diagrams and without special knowledge of the modelled domain. However, this simplicity comes at a cost of not providing enough information to the implementer for certain planned use cases (e.g. berth assignment). The naming is somewhat inconsistent and the model is not entirely compliant with UML standards. Overall, the model is reasonably good representation of the domain but is in need of some tinkering.

Observations and recommendations

Coverage of the requirements and use cases:

The model covers most of the requirements/use cases in a very clear and simple way. The model does however not represent probably the most important requirement for the new system: the automatic berth assignment system (even payment tracking, but that is not covered in the use cases and could therefore potentially be omitted). According to the use case 4 (Register boat), it is the Member who registers a new boat and not the Secretary as indicated in your model (if you have other assumptions, they should be stated explicitly).

Choice of classes and associations:

The classes are chosen well, the associations between them are reasonable and the terms used to describe them come from the domain. The *Municipality* class might, according to my personal view, not really be necessary at this level of detail (no use case). The representation of the authentication process is not really clear (who authenticates who?). It might also be a good idea to represent class attributes (e.g. a *Boat* has a *type*, *size* …) relevant to the model.

Naming conventions:

The names of classes should usually be nouns in the singular form. Therefore, it might be better to rename your classes *Berths* > *Berth* and *Boats* > *Boat* (for some examples of class names, see Larman 2005, Table 9.1. Conceptual Class Category List). Association names should preferably be verbs in the "ClassName-VerbPhrase-ClassName format where the verb phrase creates a sequence that is readable and meaningful" (ibid., 9.14. Associations). *Member/Secretary/Treasurer Login Authenticate* does not comply with this guideline. The same goes for *Municipality Checks berth info Boats*. Some association names are in the 3rd person (e.g. *Assigns*) while most are not, try being consistent preferably using the 3rd person forms according to the readability guideline. Larman (ibid.) also states that the two common formats for association names are Association-name and AssociationName.

UML notation:

The associations should be indicated using simple lines and not arrows as the "association is inherently bidirectional, meaning that from instances of either class, logical traversal to the other is possible" (ibid., 9.14. Associations). A reading direction arrow can be used to indicate how to read the diagram, especially if the intended reading is not top-to-bottom or left-to-right (see ibid., Figure 9.12). To indicate that multiple objects of the class exist, the multiplicity can be indicated on the association. If there are several associations between two classes, multiple lines should be used (e.g *Remove/Change*, see ibid., Applying UML: Multiple Associations Between Two Classes).

Reference

Larman, C. (2005). Applying UML and patterns. 3rd ed. Upper Saddle River, N.J.: Prentice Hall PTR