

File #0

Review Clarity

Language

- Only issue with spelling/grammar is the association between BoatType and Boat. Actual: Is of an, Expected: Is of a.

Notation and layout

- Using class diagram, which is the correct UML for domain modeling.
- The diagram is readable as it is not too small nor too big.
- The diagram was easy to open as it was saved as a .png.
- The layout is good. However, it would be even better if you switched the right column: Secretary, Calendar and Event, with the left column: Reservation, Berth and BoatType. These changes would make it easier to follow the associations, as it would make use of the left-to-right reading direction (Larman, chapter 9, 9.12 figure, 2004). It would then become clear that it is the Secretary that approves the Reservation and not the Reservation that approves the Secretary.
- You are using a consequent format for the names. Good!

Naming of classes and associations

- The classes are all nouns in singular form, thus following the rules of thumb. Good!
- The associations are all verbs. Good!

Not clear:

I could not figure out what the class Reservation was for. Is it the proposal on the allocation that the system provides the Secretary for confirmation? (See req. 8, step 2.) Or is it something else? The Reservation class feels redundant and should be removed. I think it would be enough just to have the Secretary Assigns Berth.

Alternatively keep the Reservation class and the association between the Secretary and the Reservation, but remove the association between Secretary and Berth, since the Secretary approving the Reservation automatically assigns Berth. Rename the association between Reservation and Berth from reserves to assigns.

Review Completeness

You have provided three UML class diagrams, but I am only reviewing the Grade 2_Model_0 since the other ones are for a higher grade.

The Grade 2_Model_0 is a domain class diagram, as it should. Good!

The assignment for grade 2 was to make a domain model for the requirements 1, 4, 5, 6, 8, 10, 11 and 12. All these requirements are considered in the domain model. These are my thoughts about the completeness:

Req. 1 - Done. Both Member and Secretary includes the attributes username and password. This information is repeated in both classes and could therefore be put in a separate class for the two to make use of (Larman, chapter 9, figure 9.9, 2004).

Req. 4, 5, 6 - Partly done. Missing the membership fee (see req. 4). The fee consists of a fixed part and a variable part, where the variable component is dependent on the number and the size of the boats. (see Problem Description last paragraph). The requirements state that the registered boat should be assigned a berth after the registration, as of now there is no relation between Boat and Berth. The attribute Picture should be made optional [0..1] (Larman, chapter 9, figure 9.20, 2004).

Req. 8 - Partly done. Includes contradicting attributes in Berth: IsAvailable and IsReserved. Choose one.

If it is available: IsAvailable = true. IsReserved = false.

If it is reserved: IsAvailable = false. IsReserved = true.

Req. 10, 11, 12 - Is fine the way it is. Only concern is that the Calendar might not need the attribute Title, since there is only one single Boat Club Calendar. The requirements state that the Secretary manages events, which might make it better to switch places between Calendar and Event in the domain model.

Review Content

Unneeded classes

- Reservation might be unneeded depending on what alternative for improvements you choose (as stated earlier in the review).

Missing information for realizing the requirements 4, 5 and 6

- Missing the membership fee (see req. 4). The fee consists of a fixed part and a variable part, where the variable component is dependent on the number and the size of the boats. (see Problem Description last paragraph).
- The requirements state that the registered boat should be assigned a berth after the registration, as of now there is no relation between Boat and Berth.

The model is focused on problem understanding.

The three main concepts Member, Boat and Berth is part of the model.

The reservation of a berth is handled in the model, but not the history.

Membership fees are partly handled in the model under Berth attribute YearlyAmount, but not fully handled as the fixed membership fee (not counting the berth cost) is not present.

Suggestions for improvements

- Fix the Spelling/grammar error: Association between BoatType and Boat.
Actual: Is of an, Expected: Is of a.
- Switch the right column: Secretary, Calendar and Event, with the left column: Reservation, Berth and BoatType. These changes would make it easier to follow the associations, as it would make use of the left-to-right reading direction (Larman, chapter 9, 9.12 figure, 2004). For example: It would then become clear that it is the Secretary that approves the Reservation and not the Reservation that approves the Secretary.
- Remove the Reservation class or alternatively keep the Reservation, but remove the association between Secretary and Berth, since the Secretary approving the Reservation automatically assigns Berth.
- Rename the association between Reservation and Berth from reserves to assigns.
- Make a separate class (maybe Person?) for the attributes username and password. This information is repeated in both Member and Secretary classes and could therefore be put in a separate class for the two to make use of (Larman, chapter 9, figure 9.9, 2004).
- Create a class for MembershipFee with attributes such as fixedCost and variableCost.
- Create an association between the classes Boat and Berth.

- Make the attribute Picture in the class Boat optional [0..1] (Larman, chapter 9, figure 9.20, 2004).
- Choose one of the attributes IsAvailable or IsReserved in the class Berth, since they are now contradicting.
- Remove the attribute Title in the class Calendar, since there is only one Boat Club Calendar.
- Switch places between Calendar and Event in the domain model, since the requirements state that the Secretary manages events.

Reference

Larman, C. (2004). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. Boston: Addison Wesley Professional.