first part:

The naming is very clear with good notations. It’s good that obvious names for the associations are left out instead of having a lot of has, on all associations. Also, multiplicity is used only when necessary which makes it easy to follow. I think the attributes could be stripped down a bit, not needed obvious attributes such as first name, and last name.

If commenting on small details all classes start with a capital letter except "user", and also "Berths" has a different type than the other classes. This is nothing that bothers me too much since often the domain model is drawn on a whiteboard and as soon as it’s set not used again. But now when it’s a clear nice looking model it makes you wonder if it could have special meaning.

second part:

Overall I would say that this class diagram is a domain model, however, it’s two classes that sound more like software design rather than domain modeling. The first is the System which has a very central role in the diagram. I would prefer it to be stripped of the design and find the associations that are the domain model associations rather than designing a system. The second class is the user, which I believe could be Role instead and maybe the Role could be a super class to Member and Secretary.

I also think Non-Member could be stripped of since it’s not part of the requirements. Or if you think it’s important for reasons that have been missed in the requirements I think you should have a text file together with the domain model explaining the reason for it.

third part:

I believe efforts has been put down trying to make this model but maybe the focus has been on problem-solving instead of problem understanding. Which is probably the reason why there is a class named "System" in the model. As Larman states that ”In the UP, the term ”Domain Model” means a representation of real-situation conceptual classes, not of software objects”[1, p134].

I have a bit hard to follow what it means that the Berths class has two attributes, one is ”Berth”, and one is ”List of Berths incl. sizes”. I think this could be split up into separate classes to get more clarity and maybe make good use of description classes as is advocated by Larman[1,p147-149]. In the same fashion, I think that the Event attribute in Calendar class should be changed to an attribute as well. Larman writes a Guideline that states that ”If we do not think of some conceptual class X as a number or text in the real world, X is probably a conceptual class, not an attribute”[1, p146]. The association names between Member and Boat also shows that a software concept has been thought of when modeling instead of real life associations and conceptual classes.

I think this model right now is just barely over passing grade, but I would definitely try to change the model to follow the rules of a domain model to be sure that the teacher will think the same. On the good side, I think it's a good base and just by making it a proper domain model will make it a very good model.

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