As a developer would the model help you and why/why not?

As a developer this Domain model could be useful but we think it is too in depth for this task. It could be clarified by simplifying the classes.

What are the weaknesses of the model, what do you think should be changed and why?

UserDescription is too in depth. UserDescription and UserRole could easily be replaced by one single “class”. UserDescription “class” is confusing, should e.g. be called UserLogin instead.

The “class” Membership seems unnecessary, it could be included in the “class” Member. Quote [1] “*Do not add things that are not there”.*

The calendarEvent and BoatClubCalendar could be replaced by a single “class” e.g. called Calendar.

Do you think a domain expert (for example the Secretary) would understand the model why/why not?

The secretary could understand the model though some points could be clarified and simplified to make the understanding of the model easier and more readable for the secretary.

What are the strong points of the model, what do you think is really good and why?

It covers all available scenarios. The presentation is quite easy to follow. They used correct names for the classes Secretary, Member and Treasurer. Quote [2] *“Use the existing names in the territory. For example, if developing a model for a library, name the customer a "Borrower" or "Patron" the terms used by the library staff.”*

Do you think the model has passed the grade 2 (passing grade) criteria?

The domain model is in some scenarios too in depths, but its most certainly is a pass. This model also represent payment, which was not included in grade 2.

Reference Section

Quote [1 & 2] Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062, Chapter 9.10: Think like a mapmaker; Use Domain Terms.