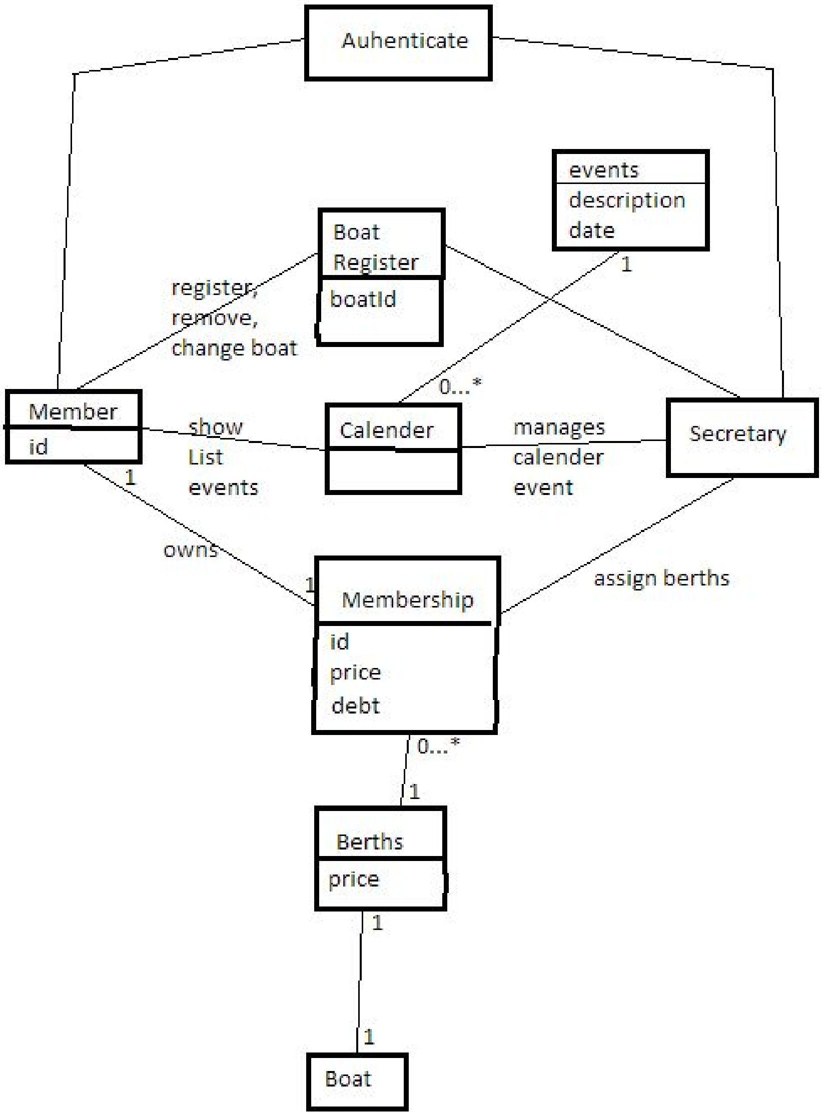

This domain model is good in a developer point of view, it is modeled with good association names and with multiplicity indicators between some classes, and only when needed. Most of the naming of conceptual classes and associations is relative to the real domain, which help both developers and the domain actors to understand the model.

The strong points of this domain model is the associations and the multiplicity indicators, the multiplicity indicators which is not strictly needed by the domain model [1, ch 9.14], but helps speeding up the understanding especially by the domain experts. The idea, about associations between the "Membership" conceptual class and the other classes which it is associated with, is really good and something I think not many have thought about.

One weakness with this model is the "Authenticate" class which feels like an irrelevant class in this domain. I should suggest it would be more like an association from a user class to the different roles in the domain. This would resemble the real world and not like it would in the software world which Tobias O. implies in his lecture about Domain Modeling [2, timestamp: 00:00-03:50].

The domain model is good and cover almost all of the user cases and the contradictory requirements. I would recommend to change the model in some cases with conceptual classes which implies classes of the real world. But other than that

1. Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062
2. Olsson T., An Online Lecture On Domain Modeling, 2016,
[https://youtu.be/Aeu_Bd738SM]