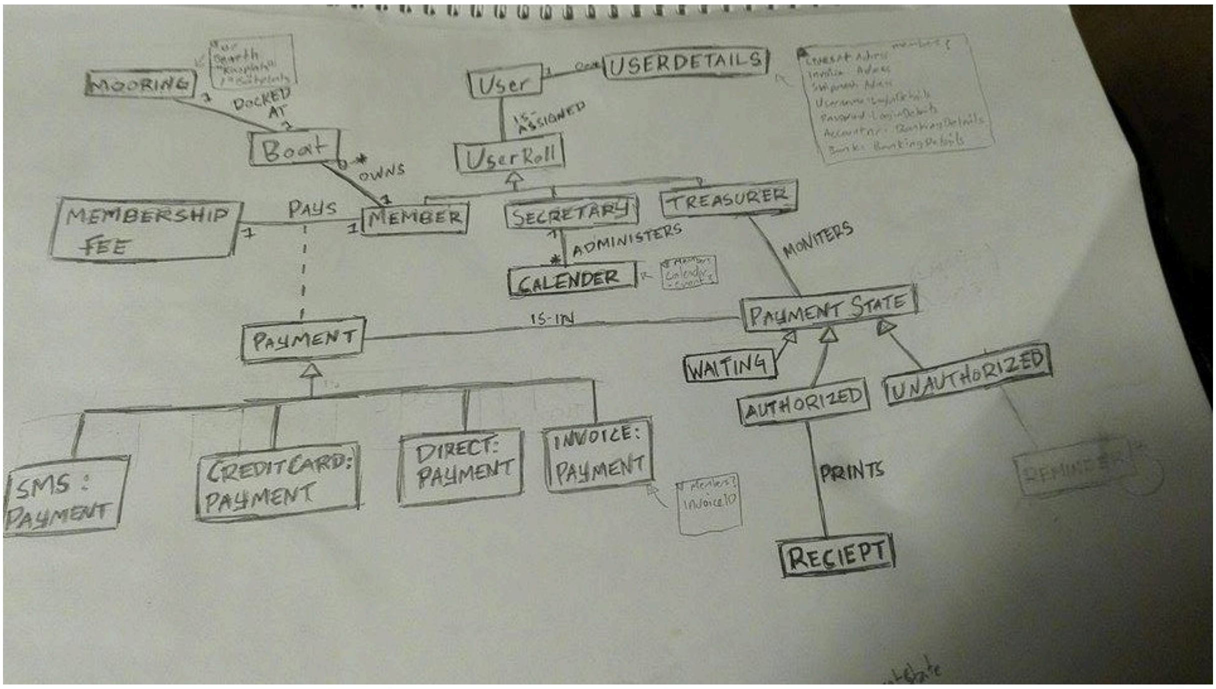
Peer review

For model by Henric Gustafsson hg222dv
Link: https://github.com/henceee/1DV607



In general the domain model gives a good overview of the domain area. It does include too much information/areas for the grade 2 requirements though, which makes it a bit harder to judge just for the passing grade alone. But I will only take the parts belonging to the grade 2 requirements in consideration for this review. In other words I will discard the classes “Treasurer”, “Membership fee”, “Payment” and their associated classes.

It is good that you have defined multiplicities for the associations where interesting, as between “Boat” and “Member”, as Larman describes [1, p154] “the multiplicity communicates a domain constraint that we care about being able to check in software”. By this specific multiplicity we learn that one can be a member even if you don’t own a boat (maybe a member had a boat before but not anymore, and the system should then still keep track of the member’s history of boats, payments etc). One concern I have though is the multiplicity between “User” and “User details” (1 to 0..1 if I see correct); should a user be able to exist without user details? Perhaps so, but maybe this would be best leaving out for the software implementation to decide about later on, as Larman indicates [1, p154], otherwise the software will need to try to obey to these conditions.

The partition of “User Role” into the subclasses “Member” and “Secretary” is done well in accordance to the guidelines described by Larman [1, p508]: “2. The subclass has additional associations of interest”, which “Secretary” and “Member” has in this case.

A problem I see is in the association between “Boat” and “Mooring”. As Larman states [1, p519] an association class is useful if “there is a many-to-many association between two concepts and information associated with the association itself”. This can very much be applied on the said association, since a mooring most likely can have different boats assigned to it in different years, and vice versa. It’s also probably interesting to list the current years berth assignments (or last year’s assignments before deciding on current year’s).

Another issue I find is if it shouldn’t be an association between the classes “Secretary” and “Mooring” (or rather to the association class between “Boat” and “Mooring” as described above). Larman addresses this in the guidelines for associations [1, p151]. According to the Use case 8 “Assign Berths” the secretary should be presented the available berths and also make the assignments of berths, which should make it logical to show this association.

Regarding the naming of the classes and associations I think it is done well, with well describing associations which makes it easy to understand without being a domain expert.

All in all the domain model is easy to understand and read, and should be easily translated by a developer into the basic models of the software, even though it might miss some details like the berth assignment association class.

I think this model has passed the grade 2 criterias.

References

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