Domain Model description

In our design we saw that the primary actors were key conceptual classes in our diagram. Starting from “member” we found the relationship to both the boat, and in making payments. The member will be both active in owning boats, and making payments according to the size and model to which they have registered their boat(s) to the club, thereby shown in our diagram. Branching from the Boat is the berth in which the boat is placed, following our logic to the physical destination of the boat. The secretary, another key actor, is in charge of managing the calendar in which different events are displayed for members. So logically in our diagram we have made that connection between the secretary and calendar events. The secretary is also in charge of booking the berths for each boat, so the appropriate connection was made from that. Lastly, the treasurer, who is also a primary actor, is in charge of checking the payments the members are making in regards to berth rental. Which is why, if you look in our diagram you see that the treasurer has a logical connection to the payments.

Every concept class is used to delimit the requirements in a way to efficiently design a domain model that will allow anyone (not just some software guy) to understand how the system should be structured and flow according to the yacht club’s desires (real world application).