

The language is clear.

The notation and the layout are not clear, if the associations have no reading direction arrow beside, the diagram is read from left to right, top to bottom.

The association names should start with a capital letter, since an association represents a classifier of links between instances (Larman, 2004).

The verbs for associations should be in singular form.

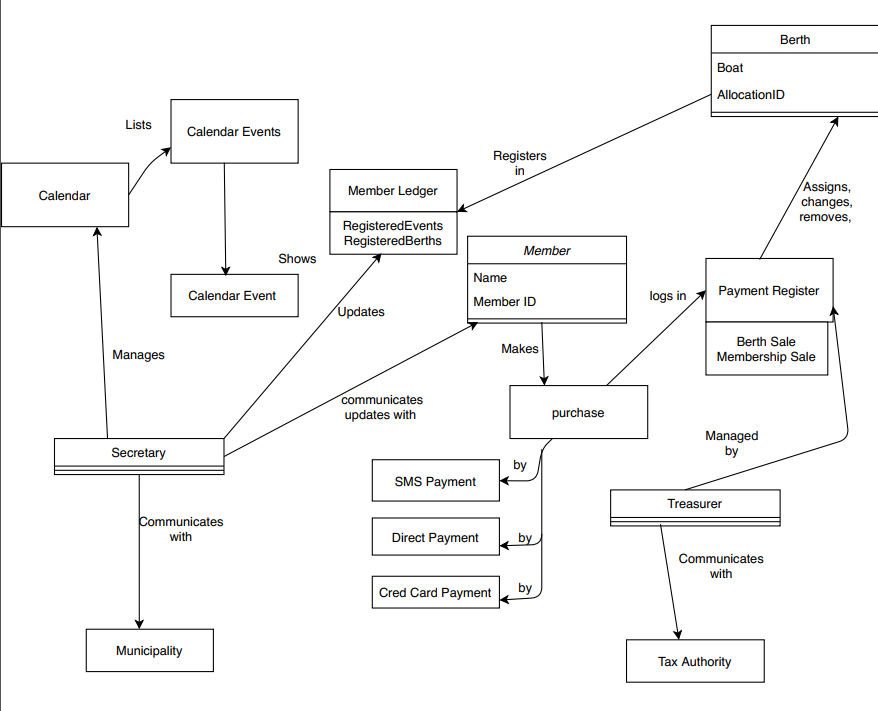
The association between *Secretary* and *Berth* is not needed, same as between *Secretary* and *Calendar* (Larman, 2004) (Ohlsson, T).

The association between *Member* and *CalendarList* is not needed, same as between *Member* and *Calendar* (Larman, 2004) (Ohlsson, T).

This model is more focused on problem understanding.

Event should not be an attribute of *Calendar*, it should be a separate conceptual class *Event*.

Multiplicity of the role between boat and berth should be multiple to multiple. As through history, a boat can be assigned to different berths, and vice versa (Ohlsson, T)



The language is clear.

The notation and the layout are not clear, the association should be represented by a line, the reading direction arrow should be placed beside association name.

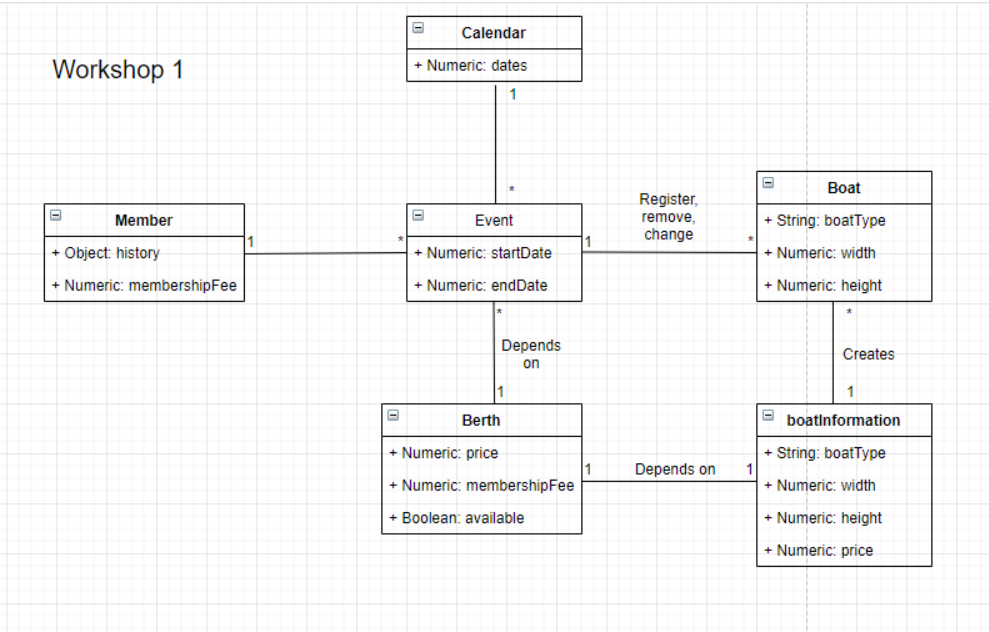
The association notation miss multiplicity to indicate the numerical relationship between instances of the classes (Larman, 2004).

The association between *Secretary* and *Calendar* is not needed (Larman, 2004) (Ohlsson, T).

This model is focused on problem solving, it should focused on problem understanding.

Therefore, some domain objects are a bit extra in this diagram, such as the association between *Calendar*, *Calendar Events* and *Calendar Event*, it only needs *Calendar* and *Event* as domain object. The domain objects should not look like software artifact, a domain model shows real-situation conceptual classes (Larman, 2004).

Boat should not be an attribute of *Berth*, it should be a separate conceptual class *Boat* (Larman, 2004).



The language is clear.

The notation and the layout are not clear, if the associations have no reading direction arrow beside, the diagram is read from left to right, top to bottom.

The verbs for associations should be in singular form.

The association between *Member* and *Event* is not needed (Larman, 2004) (Ohlsson, T).

The association between *Member* and *Boat* is missing (Ohlsson, T).

The association between *Event* and *Boat, Berth* are not needed, as the problem description states “Club’s calendar holds important event and meetings that are advertised”.

This model is a bit more focused on problem solving, it should focused on problem understanding.

The model does not have reservation of berth and the history of this handled.

Larman, C., 2005. *Applying UML And Patterns*. 3rd ed. Upper Saddle River (New Jersey): Prentice-Hall, p.Chapter 9.

Larman, C., 2004. *Applying UML And Patterns: An Introduction To Object-Oriented Analysis And Design And Iterative Development*. 3rd ed. Addison Wesley Professional, p.Chapter 9.

Review 0:

I think the association for *Secretary* and *Berth* is not needed, as there is no need to have an ongoing memory of the fact of a particular *Secretary* assigning particular *Berth*. Same as *Secretary* and *Calendar*, Member and Calendar, Event.

Historically, a boat can be assigned to different berths and vice versa. So the multiplicity of the role is multiple to multiple.

This review helped me to emphasize the reason of why I design the domain model in such way, which gives more understanding to the requirements from different perspectives.

Review 1:

I think the association for *Secretary* and *Berth* is not needed, as there is no need to have an ongoing memory of the fact of a particular *Secretary* assigning particular *Berth*. Same as *Secretary* and *Calendar*.

I think it is not needed to have some memory as “Register Boat”, it is more meaningful to indicates the connection with ownership instead of functionality.

Review 2:

I think the association for *Secretary* and *Berth* is not needed, as there is no need to have an ongoing memory of the fact of a particular *Secretary* assigning particular *Berth*. Same as *Secretary* and *Calendar*, Member and Calendar, Event.