# File #1

## Review Clarity
### Language
- No issues regarding spelling or grammar. Good!

### Notation and layout
- Using class diagram, which is the correct UML for domain modeling. Good!

- The diagram is readable as it is not too small nor too big. Only remark is that it is not centered in the file, as of now it is aligned to the right.

- The diagram was easy to open as it was saved as a .pdf.

- The layout is good and is making use of the left-to-right, top-to-bottom reading direction in a good way (Larman, chapter 9, 9.12 figure, 2004), making it easy to see how the different classes are associated with one another. Good!

- Not using consequent format for names (big/small letters). Some classnames start with a lowercase letter and others start with a uppercase letter (see userDetails, membershipFees and boatDetails). All attributes start with an uppercase letter. According to Wikipedia (2017) the classnames in a class diagram should start with an uppercase letter and the attributes should start with a lowercase letter. This is also how the class diagrams in Larmans (2004) book are written.

### Naming of classes and associations
- The classes are all nouns, but all of them are not in singular form. userDetails, membershipFees, boatDetails and Events are all in plural.

- The associations are all verbs. Good!

### Not clear:
- The dashed line between the classes Berth and Booking. What is it for? Why is it not a solid line?

Review Completeness
You have provided one UML class diagram and it is a domain class diagram, as it should be. Good!

The assignment for grade 2 was to make a domain model for the requirements 1, 4, 5, 6, 8, 10, 11 and 12. All these requirements are considered in the domain model. These are my thoughts about the completeness:

Req. 1 - Partly done. The User class does not have any attributes, such as username and password to be able to be authenticated in the system.

Req. 4, 5, 6 - Partly done. In the class boatDetails the attributes size, type and image (with optional [0..1]) are missing. As of now the attributes are Model and Weight, which I can not find stated anywhere in the requirements or problem description.

The class Berth is also missing attributes like the fee for the Berth and location of the Berth.

The line between the classes Boat and Berth is missing an association. Add association Has-assigned or something else that makes it clear that the Boat has an assigned Berth after the registration (as stated in req. 4, step 6).

The domain model is also missing a line between Berth and MembershipFee. After getting an assigned Berth for the Boat, the MembershipFee is updated according to requirement 4, step 6.

The MembershipFee class does not have any attributes. In the problem description it states that the MembershipFee contains a fixed part and a variable part. The variable part is decided from how many assigned berths the Member has. Added attributes in the MembershipFee could then for example have the names fixedCost and variableCost.

The line between the classes Member and Booking is not needed. According to the requirement 4 the system assigned the registered Boat a Berth after completing the registration. If it is off-season or pre-season the Berth is assigned when the Secretary confirms the proposal (in your case named Booking).

Req. 8 - Done. Only remark is the dashed line between the classes Berth and Booking. As of now it is confusing. Why is it not a solid line?

Req. 10, 11, 12 - Partly done. According to the requirement 10 the Secretary manages events. I would switch places for the Calendar and Event classes, alternatively draw the Manages association line from the Secretary to the Event class.

The Event class does not have the attributes stated in the requirements: title, startDate, endDate.

## Review Content

The only unneeded class I can think of is the Treasurer, but I understand perfectly why it is in the domain model. Since both the other user roles, Member and Secretary, are present. Personally I would keep the Treasurer as it is not making the domain model hard to follow and it is a part of the domain, even though it is not part of the requirements for grade 2.

The information needed to realize the requirements are partly present in the model. Mainly the model is missing attributes, some associations and multiplicity.

The model is focused on problem understanding, as it does not contain any operations.

The three main concepts Member, Boat and Berth are a part of the model.

The reservation of a berth is handled in the model, but not the history of previous assigned berths.

Membership fees are handled in the model, but does not contain any attributes as of now.

**Suggestions for improvements**
- Center the diagram in the file, as of now it is aligned to the right.

- Use consequent format for names (big/small letters). Some classnames start with a lowercase letter and others start with a uppercase letter (see userDetails, membershipFees and boatDetails). All attributes start with an uppercase letter. According to Wikipedia (2017) the classnames in a class diagram should start with an uppercase letter and the attributes should start with a lowercase letter. This is also how the class diagrams in Larmans (2004) book are written.

- Rewrite the classnames for the classes userDetails, membershipFees, boatDetails and Events to make them all singular and not plural.

- Add attributes to the User class, such as username and password to be able to be authenticated in the system.

- Add attributes to the class BoatDetail, such as size, type and image (with optional [0..1]). As of now the attributes are Model and Weight, which I can not find stated anywhere in the requirements or problem description.

- Add attributes to the class Berth, such as fee for the Berth and location of the Berth.

- Add association Has-assigned between the classes Boat and Berth. Or something else that makes it clear that the Boat has an assigned Berth after the registration (as stated in req. 4, step 6).

- Add a line and an association between Berth and MembershipFee. After getting an assigned Berth for the Boat, the MembershipFee is updated according to requirement 4, step 6.

- Add attributes to the MembershipFee class, such as fixedCost and variableCost. In the problem description it states that the MembershipFee contains a fixed part and a variable part. The variable part is decided from how many assigned berths the Member has. Added attributes in the MembershipFee could then for example have the names fixedCost and variableCost.

- Remove the line between the classes Member and Booking. According to the requirement 4 the system assigned the registered Boat a Berth after completing the registration. If it is off-season or pre-season the Berth is assigned when the Secretary confirms the proposal (in your case named Booking).

- Make the dashed line between the classes Berth and Booking a solid line instead. As of now it is confusing.

- Draw the Manages association line from the Secretary to the Event class instead of the Calendar class. Since the Secretary according to requirement 10 manages the events.

- Add attribute to the Event class, such as title, startDate and endDate, as stated in the requirements 10, 11 and 12.

- Add multiplicity, so that a Member for example can own more than one Boat.


**Reference**
Larman, C. (2004). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development.* Boston: Addison Wesley Professional.

Wikipedia. (2017). *Class diagram.* Retrieved 2017-09-07, from https://en.wikipedia.org/wiki/Class_diagram.