

[Upload your file](#)[Review](#)[Give Feedback on Reviews](#)[Check scores](#)[Introduction](#)[List of reviews](#)[Read Review 1](#)[Give Feedback on 1](#)

View Reviews and Give Feedback on Reviews

During this phase you are to view the reviews your file has generated.

Warning: You need to submit feedback for this review.

Introduction

During this phase you are to view the reviews your file has generated. You should also grade these reviews and provide a comment on the reasoning behind your grading. Note that you should not provide personal information in these comments and you are anonymous for the student that reviewed your file. However, you are not anonymous to the teacher.

[Upload your file](#)[Review](#)[Give Feedback on Reviews](#)[Check scores](#)[Introduction](#)[List of reviews](#)[Read Review 1](#)[Give Feedback on 1](#)

Your reviews

- [Review # 0 \(You have not given feedback on this review\)](#)
 - [Review # 1 \(You have not given feedback on this review\)](#)
 - [Review # 2 \(You have not given feedback on this review\)](#)
 - [Review # 3 \(You have not given feedback on this review\)](#)
-

Review # 1 on your file

Comment on Clarity

For documentation/diagrams: 1. The language dose no come with any major flaws. It's clear and understandable and I haven't found spelling/grammar mistakes.

2. The notation seems ok for most parts, there aren't many associations between the classes. Consider adding attributes which will compliment the associations in the future. It's easier to follow and it gives the class a lot more character then just a name. The only one I can see is <<use>> and I'm not really sure that you're supposed to write it like that. I'll post a link down below for an example the TA gave our group about attributes, go down to the QueryBuilder example:

<https://www.uml-diagrams.org/association.html>

[Upload your file](#)[Review](#)[Give Feedback on Reviews](#)[Check scores](#)[Introduction](#)[List of reviews](#)[Read Review 1](#)[Give Feedback on 1](#)

3. I can't see any major defects that would break the rule of thumb. Good class names!

4. As I mentioned before there are barely any attributes and multiplicity is none existent. I posted a link regarding this which is worth checking out.

For code: 1. The formatting of blocks are structured good and correct. I can't see anything wrong so far.

2-3-4. Yes the naming of classes are good and they are using real words. But I can't see that many attributes that are included in the class diagram. Consider adding attributes which will compliment the associations in the future. It's easier to follow and it gives the class a lot more character then just a name. The code overall is readable and understandable, and so are the comments.

Continue. 4. Comments are lacking on some parts, consider adding more comments.

Grade

2. Sufficient - Written in english, proper naming, and mostly correct UML notation. It may have defects/or be wrong but it is readable.

Comment on Completeness

1. Yes it includes at least one UML class diagram.
2. Yes it includes at least two sequence diagrams.
3. Yes it contains the source code of the application.

[Upload your file](#)[Review](#)[Give Feedback on Reviews](#)[Check scores](#)[Introduction](#)[List of reviews](#)[Read Review 1](#)[Give Feedback on 1](#)

Consider also to make the member management a bit more viable instead of going into compact list to do changes. When I try to update the user it asks me to do the changes but when ever I'm done I can't go anywhere further. The information doesn't update.

I get null pointer exception when I'm in boat menu and tries to add a new boat directly to a member, I also get this when I try to delete boat in boat menu. If I want to change the information I also get null pointer exception.

When ever I want to see specific member data the program freezes.

Grade

1. Failed - The submission completely miss one or more of the artifacts of the workshop, or does not at all contain what it should.

Comment on Content

1. Yes there is a clear difference between model and view. All print outs are handled in view with either `System.out.print()` or calling for a method with the information that shall be viewable in view.
2. Is the design and implementation object oriented? *It seems like you've used associations to connect the classes. But consider to add that in the UML next time so I know where to look at. * Classes have high cohesion and are not too large. It seems like you've evened it out as best as you could. * Classes have low coupling and are not too connected to other entities. This part is good. * I haven't

[Upload your file](#)[Review](#)[Give Feedback on Reviews](#)[Check scores](#)[Introduction](#)[List of reviews](#)[Read Review 1](#)[Give Feedback on 1](#)

this and double check yourself then consider checking this out: <https://stackoverflow.com/questions/46616222/what-are-hidden-dependencies> *Encapsulation is implemented and I have another link regarding this if you want to double check:

https://www.tutorialspoint.com/java/java_encapsulation.htm

*There are some part of the domain model that are missing, but the structure seems to inspire the design. You have an abstract class inside the view and the main outside everything, not sure how you should implement this. Maybe it's the right way to approach things but I can't find any examples, consider looking into this just in case if you haven't already.

3. The design is not in sync with the code since the class diagram is missing attributes, methods and some associations.

Grade

2. Sufficient - A majority of points mentioned are ok.

Give feedback on the review:

You should respond to the review you are given.

[Upload your file](#)[Review](#)[Give Feedback on Reviews](#)[Check scores](#)[Introduction](#)[List of reviews](#)[Read Review 1](#)[Give Feedback on 1](#)

- A good review should be thorough and complete
- A good review may still be of a different opinion than yours.

A bad grade does not automatically mean that you or the reviewer gets a low grade, it is an indication of that something is not right. If you think something is wrong with the review, state your view in the comments. Be polite. You are anonymous to the other student but not to the teaching assistants. Remember different people have different views and may interpret the same information differently. Learn from this, how could you have written your file in a way that this reviewer would have liked?

Warning: This Feedback is not complete.

Comment on what you learned from this review also motivate your grading of this review.

- ☒ Not graded yet
- ☐ 0. Not even an attempt - The review has no comments and grading is wrong.
- ☐ 1. Failed - Most of the review is not correct and fail to motivate the grading.
- ☐ 2. Sufficient - The review is truthful and motivates grading but the text is sparse, or the review does not seem thorough.

[Upload your file](#)[Review](#)[Give Feedback on Reviews](#)[Check scores](#)[Introduction](#)[List of reviews](#)[Read Review 1](#)[Give Feedback on 1](#)

4. Excellent - The review is truthful, thorough and motivates the grading in a good way. Also the review was helpful and I learned how to write better artifacts(strategy/plan/cases/report) from it

Save review feedback

Peer review system

Developed in 2016 by Daniel Toll in order to allow students to share and review documents anonymously.

Contact

Emaildaniel.toll@lnu.se**Slack**<https://coursepress.slack.com/messages/>

© Untitled. Design: HTML5 UP.