Peer Review Workshop 2 – Kristoffer Lind, Hannes Ljusås & Anton Andersson

Problems/bugs

- The second valid format for a social security number might be missing a number (only 9 numbers), as well as validating 000-00-0000 as a social security number.
- A person can register by using numbers as a name.

Try to compile/use the source code using the instructions provided. Can you get it up and running? Is anything problematic? Are there steps missing or assumptions made?

Result:

- No instructions as of how to get the program running. We figured it out though.
- Without looking at the source code it took us a while to figure out how to log in to the system.

Does the implementation and diagrams conform (do they show the same thing)? Are there any missing relations? Relations in the wrong direction?

Answer: Yes, the diagrams look good and show a proper amount of information regarding the specific event.

Is the requirement of a unique member id correctly done?

Answer: Yes.

What is the quality of the implementation/source code?

- **Code Standards**
  - The code looks good and is consistent.
- **Naming**
  - Naming in general is done in a good way. Minor inconsistencies with naming of methods found, hardly worth mentioning (capitalization, word separation).
- **Duplication**
  - We could not find any duplicated code.
- **Dead Code**
  - No dead code was found.

What is the quality of the design? Is it Object Oriented?

- **Objects are connected using associations and not with keys/ids.**
  - Yes.
- **Classes have high cohesion and are not too large or have too much responsibility.**
  - Yes.
- **Classes have low coupling and are not too connected to other entities.**
  - Yes.
- **Avoids the use of static variables or operations as well as global variables.**
  - Yes.
- **Avoids hidden dependencies.**
  - Yes.
- **Information should be encapsulated.**
  - Yes. Minor deviations are noted and taken into account by the use of comments.
- **Inspired from the Domain Model.**
  - Yes.
- **Primitive data types that should really be classes.**
  - No.

As a developer would the diagrams help you and why/why not?

Answer: Yes. The diagrams are informative enough and give a good overlook of how the system is intended to work.

What are the strong points of the design/implementation, what do you think is really good and why?

Answer: Good use of MVC structure, consistent coding, proper naming and a very user-friendly interface. The program contains almost perfect validation for all input events. We believe these things makes it easy for another developer to continue to work with the code, and also make it easy to get a grip of how it is supposed to work.

What are the weaknesses of the design/implementation, what do you think should be changed and why?

Answer: On a small note, the member controller could be divided into both a member controller and a boat controller to separate responsibilities even more.

Do you think the design/implementation has passed the grade 3 criteria?

Answer: Yes, most definitely.
Add a simple search function and you should be able to reach grade 4. To reach grade five you should simply have to address the validation issues and tweak the search function a bit.