Peer Review

Group: Johnny Pesola

Portfolio: http://www.pesola.se/webbprogrammerare/1dv607/Workshop1_grupp6.pdf

The first thing we noticed was the size of the domain model, and it felt more confusing than reflective. It gave us a slight headache. The domain model supersedes the requirements for this assignment, the extra payment domain model should be removed and keep to the specified Use Cases. Larman describes [1, section 9.1] your domain model perfectly in an example by stating that you should avoid "over-modeling".

We found that your domain model is centered around a class called "UserInterface". The domain model should not be focused on a software class, or user interface; it should be an actual depiction of how the company is operated. [Larman, Section 9.10]

The association name can be improved by using more descriptive names than "uses" [Larman, Section 9.16]

The model utilizes a lot of "relationship tables" (as in a database), the domain expert will most likely be confused by these. For instance a Boat is described in a BoatDirectory that consists of BoatDetails. The Boat should be described directly by the BoatDetail-class, what is the purpose of the BoatDirectory in this instance?

The entire model gives the impression of being a database model, not a domain model. The classes should focus on relationships in a more conceptual sense rather than specifying foreign keys with id-attributes. [Larman, Section 9.2, Section 9.14]

There are classes for Member and Secretary, but they do not have a relationship to a Person, how are they connected?

The model is very well made and follows UML-standards with arrows next to associations and the multiplicity expression on every class where needed. The model is not sloppy in any way, there has been very thorough work put into the assignment, but focused in wrong area.

We do not believe that the model has passed the grade 2 criteria as it is too much focused on software, not reality.

Sources
Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062