Strong Points

The model is using good names that are easy to understand and follow. Larman states on page 239 that one guideline is to use existing names in the territory. There is no such thing as a right domain model, many different domain models about the same problem can all be right. However Larman states on page 267 for a domain model to be useful it should capture the essential information to understand the domain model in the context of the requirements. I believe that this domain model fulfils those requirements.

Would this help me as a developer?

When I look at the domain model I can get a good overview about the problem. I think this can really help me as a developer, especially to visualize the problem.

Weakness of the model

I believe that software objects are used in this model, and that is something that should be avoided. I consider interface as a software object and not a real object. Larman states on page 224 that a domain model is a visualization of things in real-situation not software object.

The lack of associations might possible make it harder to understand the concepts and how all objects relate to each other. Also on the associations that are done, it can potentially be replaced with something else. Larman states on page 249, that associations names like has are usually poor and does not enhance to understand the domain.

Do you think the model has passed the grade 2 (passing grade) criteria?

I think the model is almost there, just some minor fixes and it should pass the criteria for grade 2.

Reference Section

1. Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 9780131489066 (https://aanimesh.files.wordpress.com/2013/09/applying-uml-and-patterns-3rd.pdf)