We have used the PDF version of the book which you can find on the following link.

[**https://aanimesh.files.wordpress.com/2013/09/applying-uml-and-patterns-3rd.pdf**](https://aanimesh.files.wordpress.com/2013/09/applying-uml-and-patterns-3rd.pdf)

**As a developer, would the model help you? Why/why not?**

We feel like this domain model would help us understand the main concept of this specific domain. This is because of a lot of conceptual classes have been identified. But sometimes we feel that the association between the conceptual classes is a bit vague and abstract.

**Do you think a domain expert (For example, the secretary), would understand the model? Why/Why not?**

We currently do not think an domain expert would understand this model. This is because there are a lack of information regarding what goes on between the different sets of conceptual classes, as well as their naming.

**What are the strong points of the model?**

Since the domain model contains a lot of valuable conceptual classes mentioned in the requirements, we feel like the diagram is covering big parts, which gives an overview of the domain.

**What are the weak points of the model?**

After discussion regarding your domain model, we have concluded the following when comparing your UML notations different sets of components with the book ‘Applying UML and patterns’ by C. Larman.

One notation that we feel is missing from your diagram is the *multiplicity notation*. According to Larman [1, p249] *multiplicity defines how many instances of a class A can be associated with instance of a class B* (For example, *see figure 9.13*).

Another thing we have noticed with the diagram is the lack of descriptive verbs on the lines representing associations between the conceptual classes. Like Larman shows [1, p246] on figure 9.11, there should be some explanation of the associations which should consist of verb(s).

We also feel that the diagram needs arrows that explains the reading direction. Just like Larman writes [1, p248] *An optional “reading direction arrow” indicates the direction to read the association name; it does not indicate direction of visibility or navigation. If the arrow is not present, the convention is to read the association from left to right or top or bottom, although the UML does not make this a rule*. If one follow this rule and applies it to your diagram, the ‘Boat Register’ controls ‘Tax authority’.

One last thing that we had a hard time to grasp was the conceptual class ‘boat register’. What we find hard is what it actually represents. According to your numeric list, it represents a boat, the registration of a boat, and the registration of a payment of a boat. You can read more about identifying conceptual classes in the book [1, p230].

**Do you think the model has passed the grade 2 (Passing Grade) Criteria?**

Based on the points in the previous section, we do not feel like this is structured enough to reach the passing grade. With some re-tinkering like removing the list, and the list notations, replace with verbs explaining the conceptual classes, add arrows explaining the reading flow, and preferable add multiplicity.

**References**

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