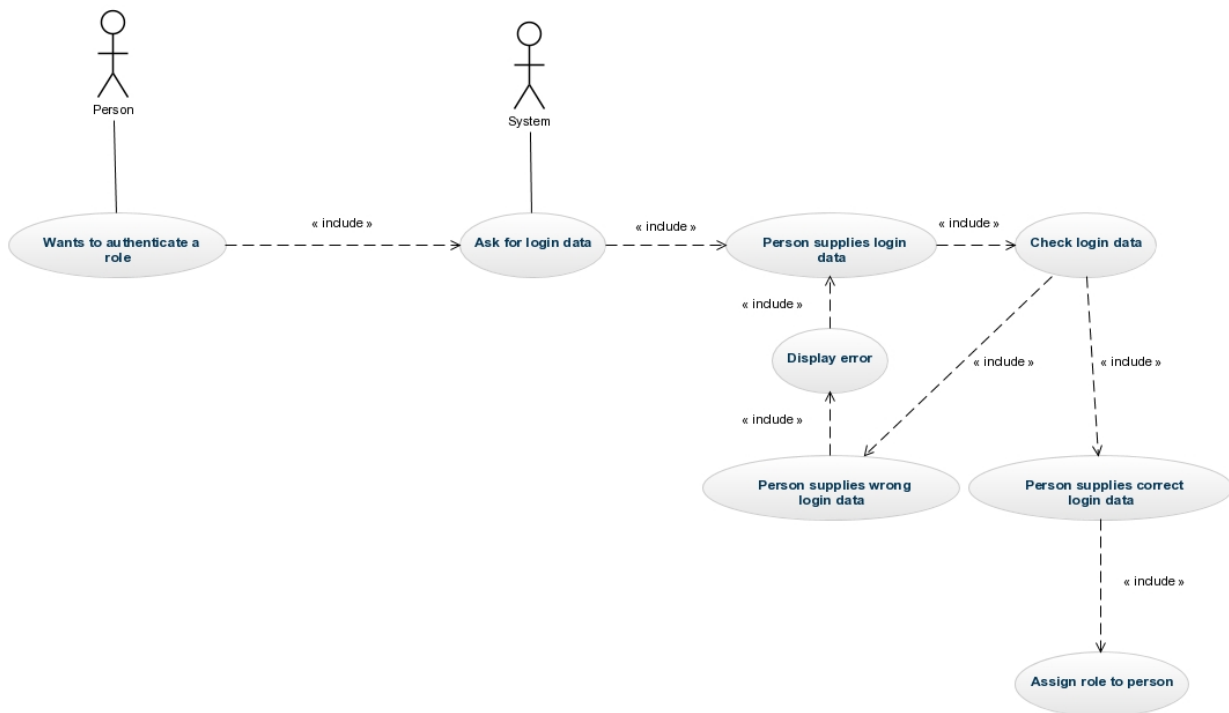
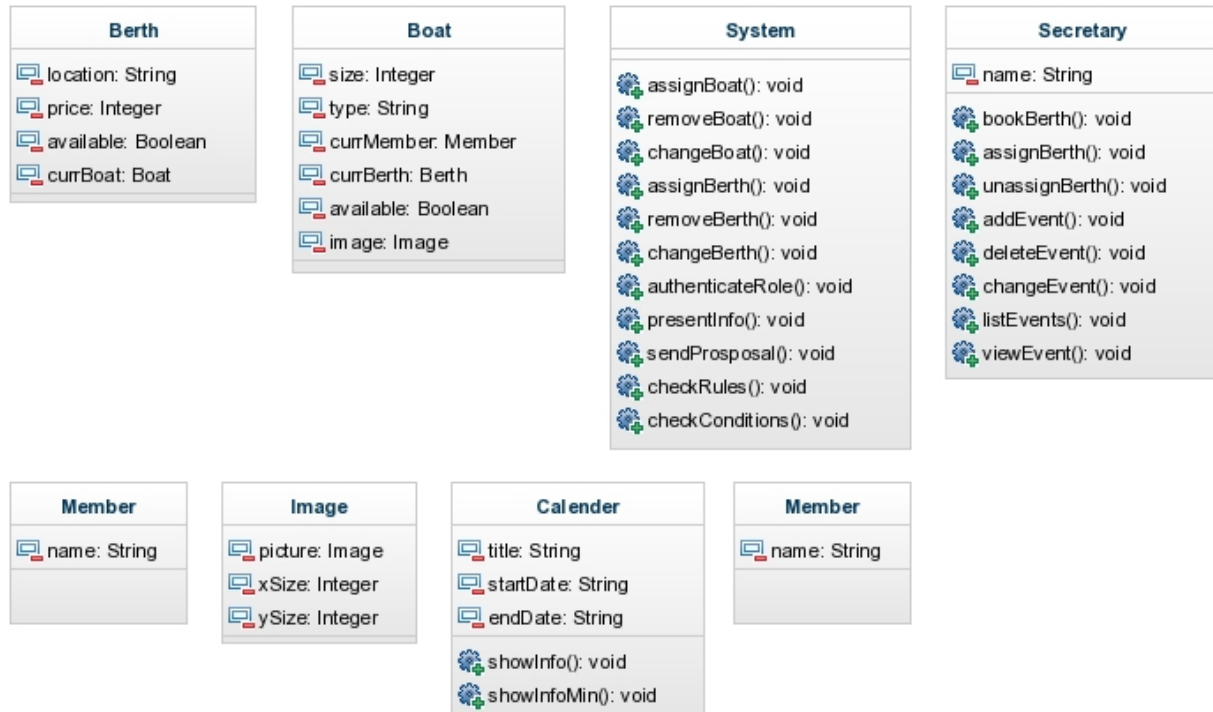
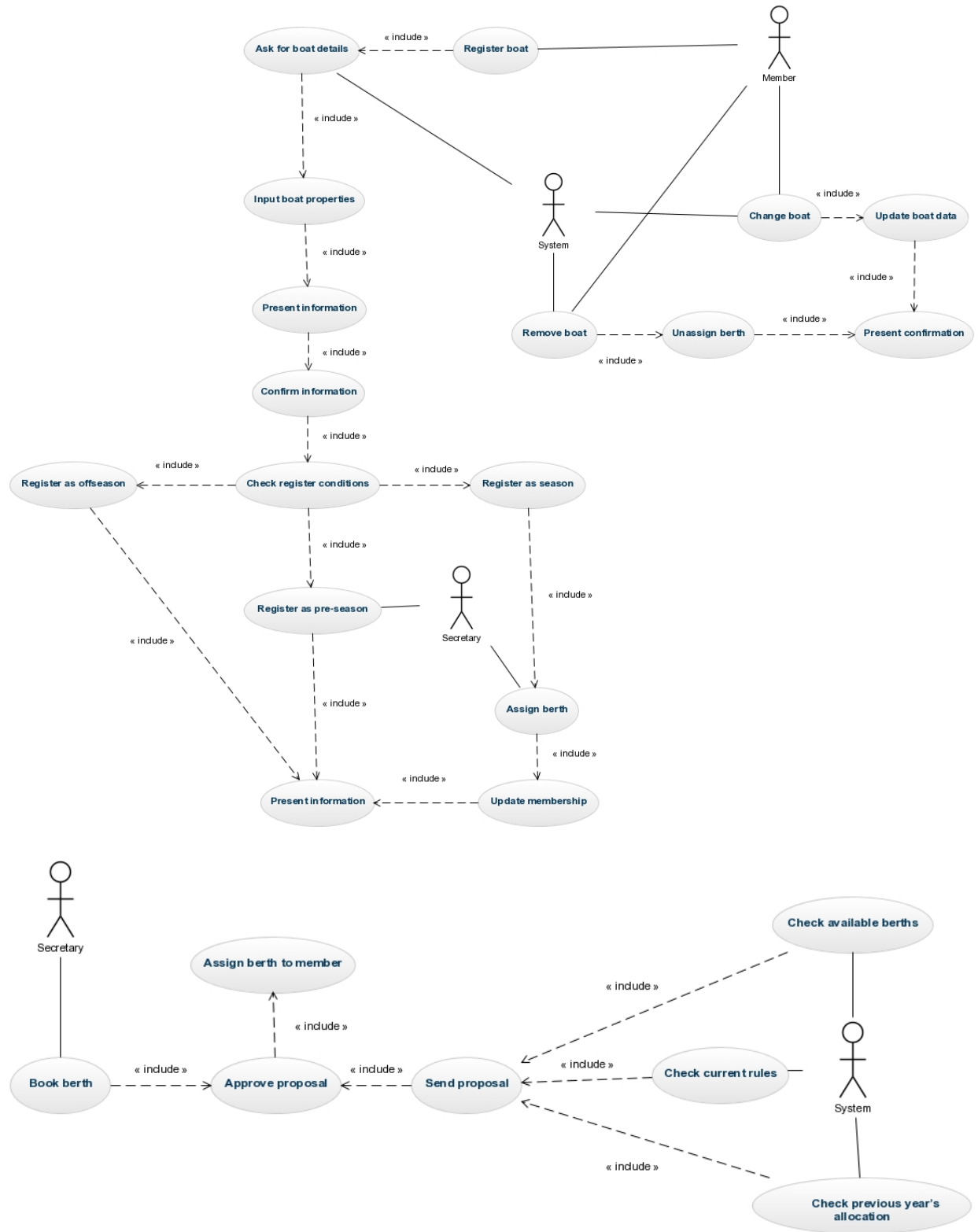
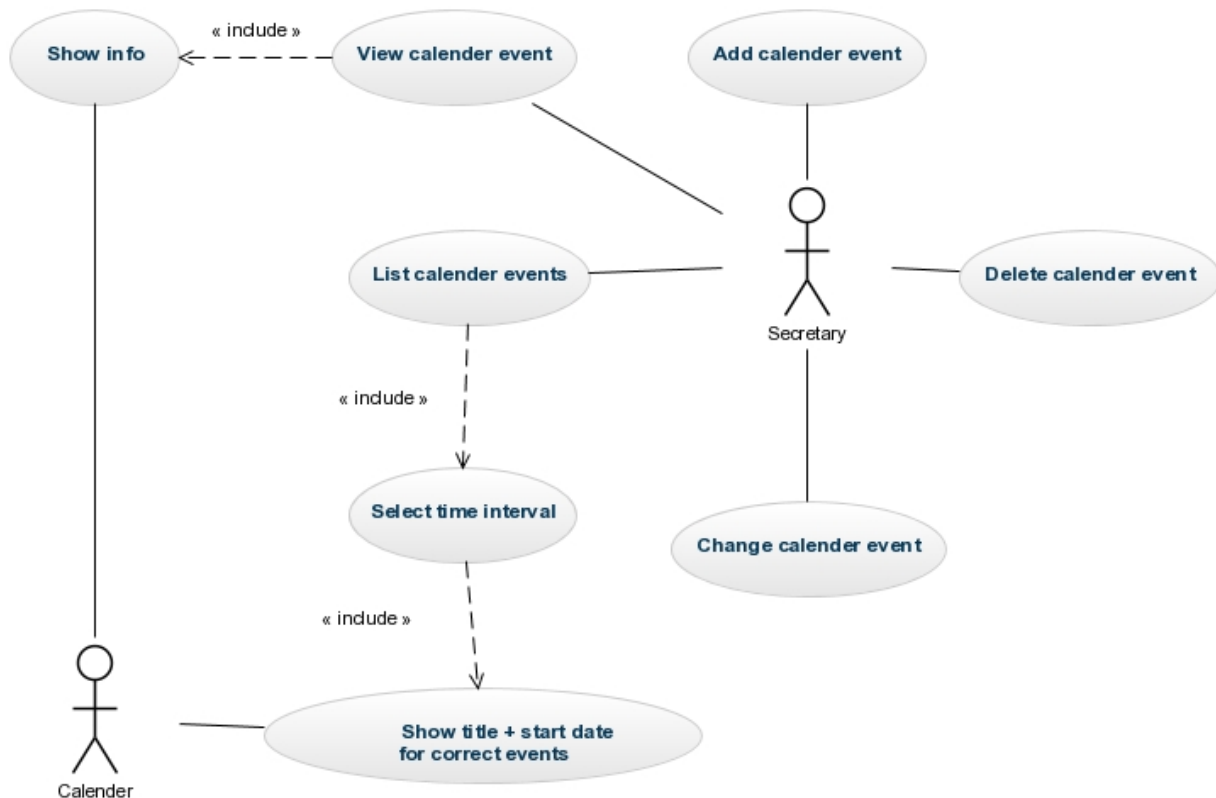


Robert Hallink

<https://github.com/DevRobDev/1DV407/tree/master/Workshop%201/Files/Diagrams>







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

The model would help us not only understand the domain, but also implement a solution due to its high level of detail. However, in some cases it might restrict the developer.

Do you think a domain expert (for example the Secretary) would understand the model why/why not?

The domain expert would probably understand the model thanks to good naming and clear structure. However, the complexity could make it hard to grasp and take longer time to get a clear overview. The associations might be hard to understand since they all are marked as “include”.

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

The naming is good and makes the model understandable. Every part of the system is covered. The model is object-oriented. The attribute types are clearly defined in the class diagram.

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

The model is very complex and unwieldy and is focused on software rather than reality. The model would benefit from being simplified. The associations should be more defined. The model would be easier to understand if the classes were merged rather than split up into different figures.

Do you think the model has passed the grade 3 criteria?

All required use cases are implemented in a well structured, object-oriented analysis and therefore we believe the model has passed the criteria. However, the model is far too complex and software-focused to be a domain model and needs to be simplified.