Report Grading

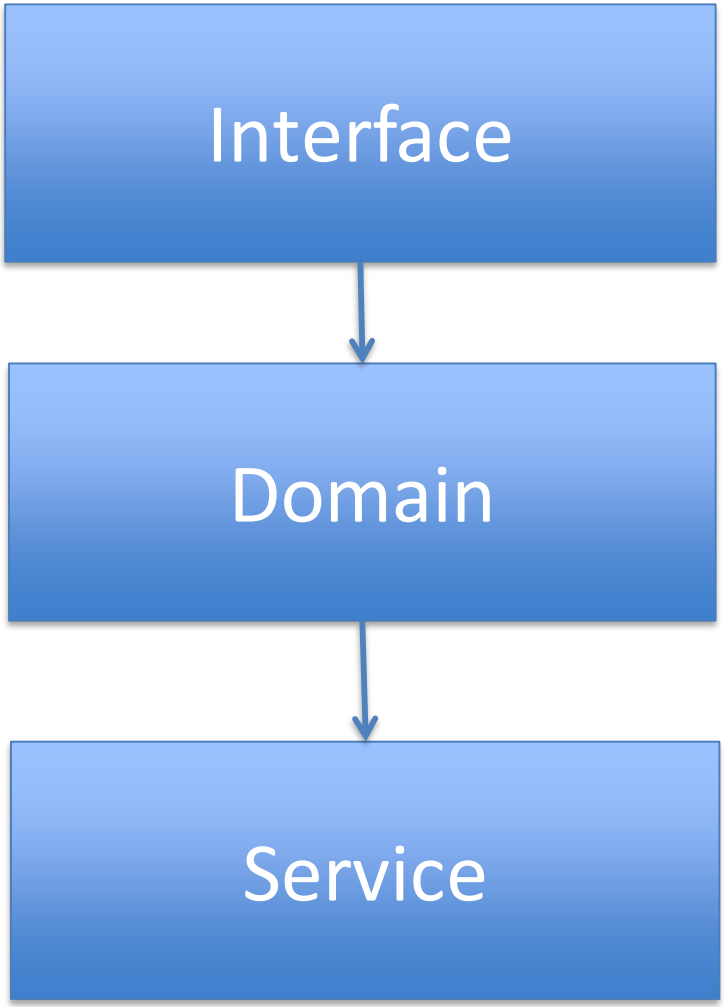
- For the report to pass, at least booking, check in, and check out shall be documented. Your report shall include these parts:
  - Pre-study
  - Requirements
  - Use cases / activity diagram
  - Domain model
  - Component diagram
  - Component sequence diagram
  - Class diagram
- Comments: Text shall be used when needed, to put the parts above into a coherent readable document. Text is also needed to establish relationships between the parts. See appendix for requirements on each part.

Extra points for more levels

This course focuses on the domain layer. You can, however, get extra points for including more layers, for example interfaces and services, such as databases, but only if they are modelled. We give no support for this in the course.

Nevertheless, this is no replacement for a well designed domain layer. You can only get extra points if the domain layer is of high quality.

See next page for risk discussion.



Risk when including more layers

- The domain layer shall be well designed using object oriented principles.
- In the past, a few groups have successfully included other layers. However, most good groups have had no extra layers and still gotten a high grade!
- Indeed, in the past, some weak groups have included several layers instead of a good domain layer. This is bad and will in the worse case result in the project not passing.

Extra Modelling

- You can also obtain points within the domain layer by doing extra modelling such as:
  - OCL
  - State machine
  - Deployment diagram

How to evaluate the text and diagrams

APPENDIX

Pre-study

A textual description of the problem domain, 3-8 pages	1 *	
Showing a good understanding of the problem domain	2 *	
A textual description of the system to be built, 2-6 pages	1 *	
Having several potential solutions, showing clear proof of creativity	2	
A good justification for the chosen solution	2 *	
Having a good business case for the chosen solution	2	
Included in appendix: mind-maps, drawings or other models	2 *	
Included in appendix: 2-5 interviews with people in the hotel business +	4	
Showing creative ways of obtaining knowledge of the problem domain	2	
Bonus point if the pre-study is exceptionally good	1	

+ The interviews need to be transcribed in an appendix. Each interview shall be at least 2 pages of text.

Requirements

The specification FR are justified in the pre-study	1 *	
The FR have good coverage, the system is not too simple	1 *	
The FR capture in a good way what is expected of the system	2 *	
The specification of FR is easy to understand	1 *	
The FR have a unique id	1 *	
At least 3 good NFR are included	2	
At least 10 FR showing domain insight, i.e. not only trivial FR such as add room or change room	2 *	
The FR are easily testable in the final system, e.g. are precise enough	2 *	
Bonus point if the FR and NFR are exceptionally good	1	

FR= Functional requirements
NFR = non-functional requirements, constraints

Domain Model

Points will only be given for a domain model expounding domain knowledge. There shall be at least 10 concepts with attributes, associations, multiplicities and role names.

The domain model is a good representation of the problem domain	1 *	
The domain model is not too restricted	1 *	
The domain model contains good constraints using multiplicities	2 *	
The domain model is free from misplaced concepts, such as software artefacts	1 *	
The concepts contain the most important attributes	1 *	
The domain model contains the most important associations	1 *	
The vocabulary of concepts is described	1 *	
Good concept names	1 *	
Good role names	1 *	
Bonus point if the domain model is exceptionally good	1	

Use cases

There are 3 use cases, covering booking, check in, and check out	3 *	
2-5 more use cases	1 *	
2-5 more complex use cases, affecting the architecture	2	
The use cases have names, main flows, pre- and post-conditions	1 *	
The use cases contain some alternative flows	1 *	
The use cases contain all important alternative flows	2	
High quality pre- and post-conditions	2	
Good structure of the use case specification, e.g. using patterns	2	
The vocabulary from the domain model is used in the use case speciation	2	
The FR included are covered by the use cases	1 *	
At least one use case is specified by an activity diagram	1 *	
Bonus point if the use cases are exceptionally good	1	

Activity Diagram

Points will only be given for an activity diagram adding domain insight. There shall be at least a start node, an end node, a number of activities and a few branches.

The activity diagram is easy to understand	1 *	
Good activity names	1 *	
Good guard names	1 *	
Bonus point if the activity diagram is exceptionally good	1	

Component Diagram

The component diagram contains at least 2-3 components	1 *	
Interfaces are defined with clear names and responsibilities	1 *	
All interface operations have an informal contract defined	1 *	
All interface operations have a semi-formal contract defined, using the domain model for pre- and post-conditions	2	
The dependencies between components are shown	1 *	
The component diagram is easy to understand, at least in combination with a textual description	1 *	
The component diagram contains test components	1	

Component Sequence Diagram

The component sequence diagrams shall at least show the most complex collaborations in the system. If all your sequence diagrams contain less than 3 components, your system is probably to simple, or not divided in a correct way. It is okay to include test components.

At least 3 non-trivial sequence diagrams	3 *	
The sequence diagrams show alternative flows	2	
The sequence diagrams are easy to understand	1 *	
Good method names	1 *	
All method parameters are included	1 *	
Return values are shown	1 *	
Bonus point if the component sequence diagram is exceptionally good	1	

Class Diagram

The class diagram is a good representation of the system to be implemented	1 *	
The class diagram follows a good object oriented style	1 *	
Good class names	1 *	
The classes contain the most necessary attributes	1 *	
Good attribute names	1 *	
The model contains all the important associations	1 *	
The associations have correct multiplicities	1 *	
Good role and association names	1 *	
Modularity and extensibility are addressed adequately, e.g. by applying suitable design patterns.	4	

Grades

- 3/G
  - To pass the course you need 48 of the 59 star-marked points
- 4 (Chalmers only)
  - For a 4 you need an additional 13 points
- 5/VG
  - And for a 5 and VG you need yet 10 more points

Extra Points

- You can only get points for extra work if you have passed
- You can get at most 8 points for extra modelling work:
  - Points can be obtained for:
    - OCL
    - State machine
    - Deployment diagram
  - For a good model of the interface you can get 4 points
  - For good models used in the service layer, such as of a database, you can get 4 points as well

OCL

- You can get up to 4 extra points if you have four OCL constraints that clearly add value to the overall understanding/design of the system.

State machine

You can only get points for the state machine if it adds value to the overall understanding/design of the system, and contains at least 4 states and a number of transitions. You can model the states of either a class, a component, a use case, or the whole system.

The state machine is easy to read	1	
Good state names	1	
Good event names	1	
Bonus point if the state machine is exceptionally good	1	

Deployment Diagram

The deployment diagram must show a realistic case (what the finished system will look like), where there are several client and server nodes. It shall also show how the nodes communicate, as well as which components are allocated on each node.

The deployment diagram is easy to understand	1	
Good node names	1	
Associations between nodes are shown	1	