**Architecture Design Peer Review Form** (version B)

Please create a single document in which you address the following aspects of the architecture design. Please note that you can give praise if a design is nice/good and give constructive feedback if there is room for improvement. It is fine if your response to each of these sections is succinct; we are not asking for long pieces of prose.

**Section 1: Graphical Appearance**

Please comment on the graphical appearance of the architecture models.

Do the diagrams have a clear and consistent layout? Is space used well? If the font readable?

If layering has been applied, is this layouted nicely? Are dependencies directed from top to bottom?

**Section 2: Naming**

Do components have clear names? Do the names clearly suggest the responsibility of the components?

**Section 3: Explanation**

1. Does the document contain clear explanation of the architecture design? Is there enough explanation or is something missing?
2. Are important design decisions explained?
3. Are important assumptions stated?

**Section 4: Quality of the Design**

1. Decomposition & Responsibilities
	1. Completeness: Is all needed functionality present in the system?
	2. Do all components have a clear stereo-type responsibility as proposed by Rebecca Wirfs-Brock (see explanation supplied on the next page)?

Are the responsibilities of the different components clearly scoped?

* 1. Is the decomposition done systematically? i.e. are all components at the same level of abstraction?
	2. Is the decomposition at the top-level a proper functional decomposition or is there also a mix of implementation-decomposition?
1. Design Principles
	1. Separation of Concerns: Do all components have a clear single responsibility that is different from the responsibilities of the other components?
	2. Coupling: Are there any components that you suspect have too high a coupling?
	3. Cohesion: are there any sub-components put together in one component that do not seem to belong together

**Section 5: Any other feedback**

**ROLE STEREOTYPES**

In this section we briefly explain the notion of role-stereotypes. Rebecca Wirfs-Brock proposed a design-approach based on the notion that each software subsystem (or component and even class) should have a well-defined *responsibility* in order to play one of a few generic roles in a system’s design. Wirfs-Brock classified the roles of software objects into six stereotypes:

* (CT) Controller: objects designed to make decisions and control complex tasks,
* (CO) Coordinator: objects that do not make many decisions, but in a rote or mechanical way, delegate work to other objects
* (IH) Information holder: object designed to know certain information and provide that information to others.
* (IT) Interfacer: objects that transform information and requests between distinct parts of a system. It can be a user interfacer object that interacts with users. An interfacer can communicate with external systems or between internal subsystems.
* (SP) Service provider: objects that perform work and offer services to others on demand.
* (ST) Structurer: objects that maintain relationships between objects and information about those relationships. Structurers might pool, collect, and maintain groups of objects.

This taxonomy aims for orthogonal non-overlapping categories. However, there may be situations where a component can play multiple roles towards different collaborators.



If you want to read more about this, you can find the paper ‘Characterizing Classes’ available via the courses page on Canvas. Alternatively, you can read about this in the book: Object Design: Roles, Responsibilities, and Collaborations by Rebecca Wirfs-Brock, Alan McKean, Addison-Wesley, 2003, ISBN 0201379430.