*Outline how you would approach the tasks of creating the architecture for the Beer Web Store. Consider, e.g., which steps would you take and in which order?*

I would first do a brainstorm to list possible issues relevant to the system. Thereafter i would use the brainstorm as basis for making a list of important quality attribute scenarios written on the form proposed by [Bass et al. 2007]. The quality attributes in the list should then be prioritised, especially if some of them are non-orthogonal to the others. This is all done in order to reveal non-functional requirements.

The quality attribute scenarios should then be used as input for attribute-driven design where I iteretively choose a module or a part of the system to decompose, and then refine that model according to the architectural drivers found via the quality attributes. Architectural drivers are requirements with high priority that have an effect on the system architecture.

The above steps will result in a architectural description which is best described in different views. I will use the proposed viewpoints from [Christensen et al.], Module, Component and connector and an Allocation viewpoint.



Figure Top level module view



Figure Decomposed module view showing the Domain Model

*Give concrete examples of what elements, relations, and structures as defined in [Bass et al., 2007] could be in relation to an architecture for the Beer Web Store.*

*Apply [Perry and Wolf, 1992]’s model of software architecutre as*

*Software architecture = {Elements, Form, Rationale}*

*to the Beer Web System.*

*Reflect on what happens if the words “software” and “computing” are removed from the definition:*

**Bass et al definition:**

*“The [software] architecture of a program or [computing] system is the structure or structures of the system, which comprise [software] elements, the externally visible properties of those elements, and the relationships among them.”*

Without the words “software” and “computing” the definition of architecture becomes much more general and can be used as a definition of e.g. building architecture.

[Perry and Wolf] samanlíkna arkitektur í vanligan forstan d við software arch

**[Alexander et al. 1977] (about patterns in building architecture):**

*“Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.”*

*The architect decides to create a full architecture description before embarking on any implementation of the system. Discuss pros and cons of taking that approach.*

Pro

* Makes the architect think about the whole solution in detail before taking the architecture in any direction.
* Makes it possible to discuss the proposed architecture with others architects and other stakeholders without wasting time implementing code that might change at a later time.
* All stakeholders will have well defined roles. Developers will know exactly how to implement the system.
* Easier to achieve the right balance between different qualitites.

Cons

* Difficult to understand any non-trivial system well enough to lay out the full architectural description in detail.
* Requirements change as time passes, hence a well planned system might be obsolete by the time it’s implemented.

References

[Bass et al. 2007] Bass, L., Clements, P., and Kazman, R. (2007). *Software Architecture in Practice*, Addison-Wesley, second edition.

[Alexander et al. 1977] Alexander, C., Ishikawa, S., Silverstein, M. (1977). *A Pattern language*. Oxford University Press.

[Fielding 2000] Fielding, R. T., (2000) *Architectural Styles and the Design of Network-based Software Architectures, ch 1.* http://www.ics.uci.edu/~fielding/pubs/dissertation/software\_arch.htm