

## Advanced Software Engineering – Task 1, TaskID: MOD

**Abstract:** This document is used to collect all assumptions/considerations used to develop the necessary artifacts of this task and also acts as the result report.

The document is split according to the assignment in 2 sections; the initial section deals with the modelling exercise and interpretations based on the description and analysis of the wikipedia.org website. The second section deals with assumptions related to the implementation of the model in Java.

Assumptions and considerations are used in the following as synonymous, meaning an own interpretation of the actual situation has been applied

### Implementation Step a: “Model the Wikipedia system as described above”

*Task Details: Model the Wikipedia system as described above. The description contains information of varying quality. Think twice before you model it or before you decide to drop it. If you are uncertain or make certain assumptions, document your thoughts. Draw a diagram which represents the model. You can draw either simple “boxes and lines” using a pen and a piece of paper, UML Class Diagrams, or something similar.*

For the modelling, Eclipse IDE including EMF has been used in the currently available version (Eclipse Mars, available online <https://eclipse.org/modeling/emf/>).

1. As Wikipedia considers all elements described in the assignment as nodes in a network, a superclass is introduced called “Node” that considers all common aspects of the content and user sphere. This “Node” class has an attribute called nodeName representing the unique name for a specific node and an operation to encode the title in a URL, as well as an operation to retrieve the type prefix (as a getter from the attribute) from more specialized elements in the model.

Example: File:Great Wave off Kanagawa2.jpg

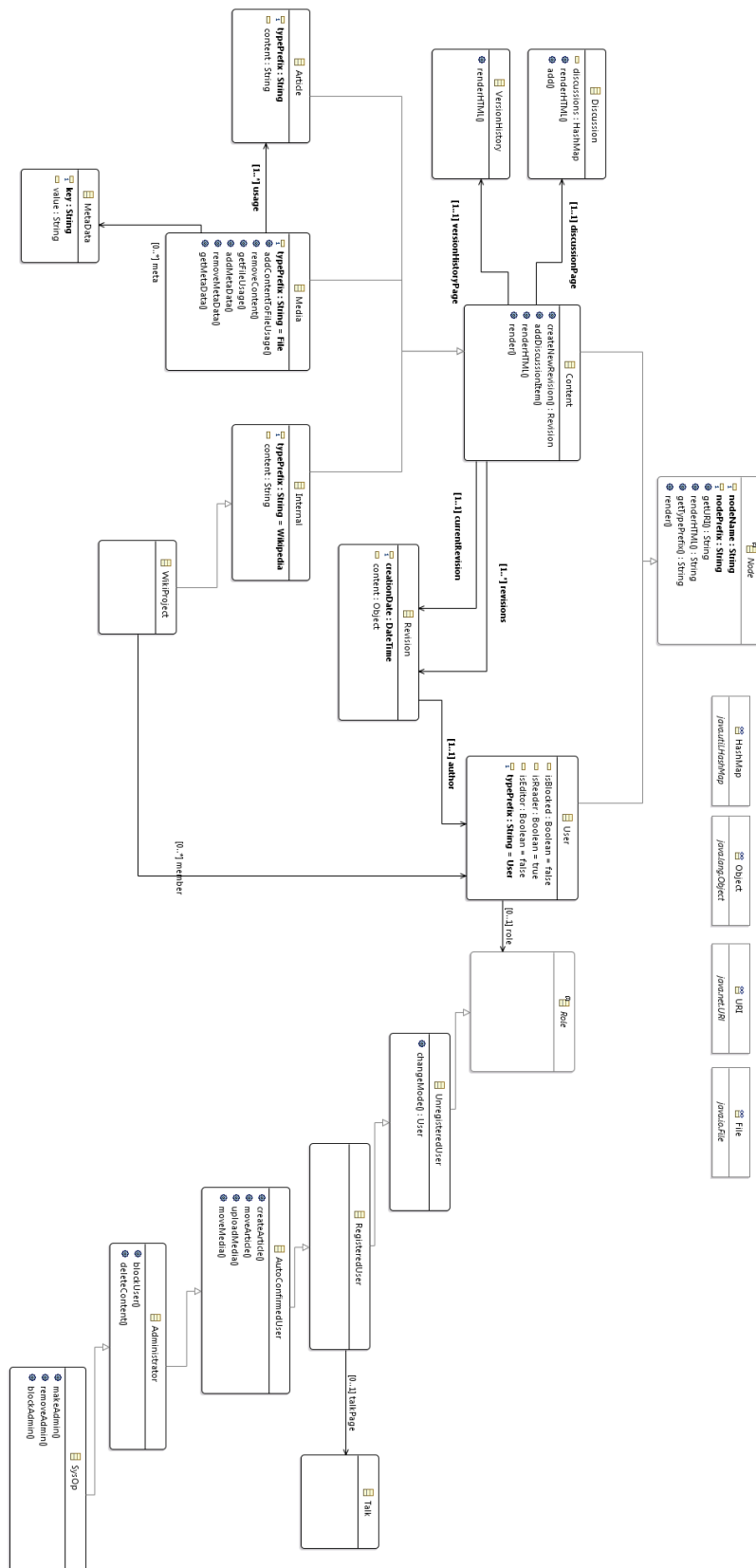
“File” -> return of getTypePrefix()

“Great Wave off Kanagawa2.jpg” -> nodeName

“File:Great\_Wave\_off\_Kanagawa2.jpg” return of getURI()

An additional operation is introduced to render each possible node into HTML “renderHTML”

2. On the level below Node, a distinction is made between content and users, in accordance with the assignment. The prefix logic still applies as described above. A specific username attribute is only modelled for registered users.
3. For the content elements (of different types) the hierarchical structure has been depicted as assigned, a detailed content-type specific representation (attributes and references) is based on an educated guess (see Metadata for Media, fileUsage for Media, content for Article/Internal)
4. For internal content, a distinction is made between material and projects as an interpretation of this sentence “Content belonging to internal Wikipedia material or projects forms special groups and must be clearly marked as such.” and research on wikipedia.org ([https://en.wikipedia.org/wiki/Wikipedia:Manual\\_of\\_Style](https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style) vs. [https://en.wikipedia.org/wiki/Wikipedia:WikiProject\\_AP\\_Biology\\_2008](https://en.wikipedia.org/wiki/Wikipedia:WikiProject_AP_Biology_2008)). The differentiation is that projects have members.
5. The language versions (and their interdependencies/cross-referencing) are not considered in the current model.



## Implementation Step b: “Model the Wikipedia system as described above”

*Task Details: Subsequent to creating the model diagram, create class stubs for all relevant classes and provide a full implementation for the content classes in Java [1] and an instance with sample data thereof. If you have to make your own assumptions, document them in a separate file. Use the Eclipse IDE [2] (including modeling tools) so you get used to it for future assignments. We strongly recommend not using any tools not mentioned by us, since we will be working with them during the course, but all complete and working solutions will be accepted.*

1. The implementation follows the approach above, whereas a focus is given to the content classes of the model, for the other parts, only stubs with needed operations/parameters are made available. The classes for the model are implemented in the package, in accordance with the hierarchy defined. The functional implementation is available for the elements Article, Media and Internal; a focus is given to provide functionality on abstract level and specialize/override where needed.
2. For the application, 2 applications have been realized to work with the model and create, view, modify content in this wiki-like application:
  - a. Command-line application: as a user interface the command line is used; the application can be operated with your keyboard only – start with **WikiApplicationConsole**
  - b. Swing UI application: as an alternative, a swing UI has been implemented.
3. No consideration was given on the user/community aspect. For both application, you can either enter a username (that will instantiate a User element) or your system user is retrieved and used
4. All functionality for content manipulation is encapsulated (UI independent) in the ContentManager class.
5. Basic persistency is implemented by serializing/de-serializing all content manager by the ContentManager class.

