Todo list

1 Functional requirements

The functional requirements are based on the non-functional requirements and aim to implement those. Additional requirements are derived from the technical environment and its specifications.
To be able to use the data of Wikidata, ArticlePlaceholder needs to base on Wikibase and MediaWiki. As MediaWiki is realized in PHP, the main part of the extension needs to make use of that language as well as the services provided by MediaWiki and Wikibase.
The programming language for scripts on Wikipedia is Lua. The default for the content pages present(s) the information and data in a useful way and make use of the images in Wikimedia Commons, which are connected to the item displayed.
To be conform with the technical specifications the extension is required to work in any mobile device.

Explain Lua

As described before, the requirements differ from the different stages in the deployment cycle. They are always built on top of each other but since the parts have different importance we focus on a very basic set up first and add up more features later on. Listed below are the requirements for the final system deployed to a Wikipedia.

1.1 Naming

The special page should have a short name that is easy to remember. Calling the special page "Article Placeholder" would be confusing due to people expecting it to generate a whole article in that case. Therefore it is called "AboutTopic".

More reasoning

1.2 Display the article placeholder

The pages displaying the content of a Wikidata item focus on a clear differentiation between the layout of the placeholder and what a user-written article would look like. They still contain elements of the well-known Wikipedia layout.

Therefore, the main image of the item is on the right side of the page where it would be in articles too.

The core part of the page are the statement groups. Those consist of a property, one or multiple values, their qualifiers and references.

The statement groups are each in a box with a black border and arranged in a tile layout. The amount of statement boxes per row differs depending on the width of the screen.

The label of the property is the header for each group which is separated from the body by a horizontal line. Multiple values for the same statement are also separated by a line. Their qualifiers are right under the corresponding value and in a smaller font.

References can be found in an own section at the bottom of the page in order to stick to the existing layout of Wikipedia.

The identifier of an item should be in a distinct place to be emphasized and distinguishable from the other statements at the same time. They are listed underneath the main image of the article to the right of the statement groups.

The described layout is the default but it is very important to give the user control over everything and therefore let every part be user editable. The renderer(s) are in Lua and every part has get and set functions to overwrite them on-wiki and adjust the design to the need of the communities.

1.3 Localisation

It is important to have the whole extension localized from the beginning. Not only would it be hard to adjust it in the end but mainly this is the most important part about it- the extension being used by many international communities. Since the data itself is translated in Wikidata, this focuses more on aspects like labels on buttons and the name of the special page itself.

rewrite this!!! shorter, more clear sentences!!!

1.4 Discover article placeholder

1.4.1 SpecialPage URL

A user is able to get to the article placeholder not only through the interface of the special page but also via a handy URL, that gets the item number as a parameter. Based on the defaults of mediawiki and Wikibase, here is an example which shows the use of the defaults used on mediawiki and wikibase: `Special:AboutTopic/Q5279`

1.4.2 Search

To be able to discover an article placeholder while looking for a certain topic the generated pages are integrated in the search on the Wikipedia. There are different places for the results in the search page. It can be attached on the bottom of the results or placed to the right of the regular search results. Here it is important to figure out the needs of the editors and readers and act accordingly.
In order to reduce the content to display only desired entries related to a certain topic the search looks for items with corresponding labels and aliases. The results in the search should be limited to notable items. It is important to remark, that not on Wikipedia not necessarily the same notability criteria applies as on Wikidata(try to understand your own sentence :p correct it yourself). Since it is hard to decide, which content is actually notable, the items appearing in the search should be limited to only the ones having at least two to three statements and either two sitelinks to the same project (like Wikipedia or Wikivoyage) or two to three statements over all.

The same criteria should apply when linking from a value in the statement groups to another placeholder. Every item, that has sitelinks to the Wikipedia, the placeholder is on, should link to the corresponding article.

rewrite this part

A lot of users will search for content of Wikipedia on search engines like google. It will be necessary to index the article placeholder. This is outside the scope of the thesis but will be an important point when thinking about the caching of the pages.

1.4.3 Red links

> A red link (...) signifies a link to a page that is either non-existent or deleted.

[?]

Connect to bib file (wiki:01)

Red links are used a lot in Wikipedia to indicate an article, which isn't there. Today it leads the user to a "create article" page. In the future it should rather bring them to an article placeholder, which offers them the option to create an article. Due to the high technical complexity of matching the titles of the red links to the Wikidata items, this topic is out of the scope for the thesis but will be an important part of the future development of the Article Placeholder extension.

1.5 Creation of an article from the article placehoder

Starting at the content page created by the extension the user might want to create an article for the sake of completeness of the ~~Wikipedia article~~.
The challenge is to encourage editors, to add a new article if appropriate and at the same time prevent vandalism, for example creating empty articles or stubs.
The title for the page might differ from the label of the item due to the local Wikipedia's standards. Therefore a user must be able to enter a title for the page that doesn't exist on that Wikipedia yet.
Using the existing JavaScript modules as provided by MediaWiki, there is a button on the top of the page asking for the creation of the article. A popup on the page opens and asks the user for a title for the new page. The default is the label of the item, which can be adjusted or a completely new title can be entered.
There should be an option for users with disabled JavaScript. But since the amount of users with JavaScript disabled is about 3% the priority is rather low.

Connect to bib file (wiki:02)

After entering the title of the new article, the user is redirected to an empty editing page, where they can add their content.
Using the resources available, the user should be able to translate the article from an existing article in another language on the same topic.
In the future, it is intended to add the data provided by Wikidata to the editing page so the user can use it while writing the article.

1.6 Sorting of statement groups

The data in Wikidata is not sorted. The article placeholder are aimed at reader, who want to get an overview over the data and therefore only need the most important data on a topic and shouldn't be overwhelmed with everything available.
To keep the control at the editors, there should be a list of properties maintained on-wiki. They are ordered by their importance and read by the renderers for the statement groups.
To be easier to edit the list doesn't contain only the property numbers (as P103) but also their label (in this case: "native language"). The editor can comment on each line.

Since this might be important for future development of other extensions and might be interesting for the Wikibase software itself as well at some point, the parser for the property list page is in PHP to be accessible.
Not only should the statement groups on a placeholder be sorted but also limited to a decent number.

1.7 Identifier

Since the layout treats the identifier differently they shouldn't be in the list of ordered statements either. There isn't a datatyp for identifier in Wikidata yet but it is planned to introduce one.
Until the datatyp is deployed it is sufficient to have a static list of property ids in the extension, that lists all identifier used in Wikidata, that can be exchanged later on with the proper handling of all statements with the identifier datatyp.

1.8 Caching

Since the generated content pages will be accessed frequently it will be necessary to implement some form of caching for them. With caching it would be possible to limit the amount of API requests to Wikidata, too, which will show an improvement in the performance.

1.9 Unit tests

Especially the main code in PHP should be covered completely by unit tests. It is necessary for the workflow to set up the continues integration according to the services used by and defaults of MediaWiki and Wikibase. The CI part is not part of the thesis but is done while working on the other parts of the project.