Translate evulpo

The upcoming conceptual design is planned and structured by Alexander Schygula.

Following tools have been used to create this paper:

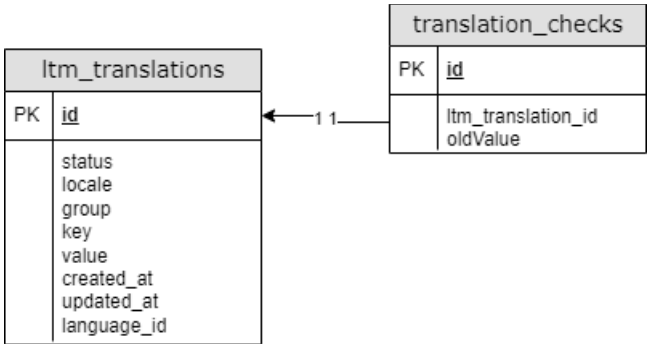
draw.io, a platform that allows you to create all types of business related graphics.

Introduction into the topic

To translate a system you have to prepare and consider only a few important things. On a technical perspective they are much easier. Modern libraries allow you to choose the „tech“ you want behind of it. So what you choose should be based more about what do you want to implement. Do you have a professional translator you may need to implement other interfaces than you would if you want to make it automatically. Due to the given techstack I assume, that a automatical translation in absence of a professional translator is prefered. But even for this case it is recommentable to do some quality checkups.

Good implementation in the backend

To translate the data we have to know what has to be translated and whether the already translated tasked are verified as „good enough quality to publish it“ in the businesscase of a learning platform the standards should be as high as possible to improve the user experience and create a feeling of „they know what they're doing“. For this a verified flag in the translation table just for checkups could be a possiblity. That will always be flagged to flase when the translatable part itself has been changed. A four eyes verification process for a change would maybe be the correct thing to do. To reduce redundant data, we could also create a new table with just a field „id“, „translation_id“ and „oldvalue“ to reset to the old translation if there is something wrong with the new one. After the new value has been confirmed you can delete the database field in the translation_checks table.



After setting up the infrastructure you can figure out what has to be translated. For a whole plattform translation it is everything you have in the main developing language (the language that gets the first and initial input from marketing) I guess in your case it will be german or english. After fetching a chunk of data that you want to translate since translating everything at once could bring some issues due to run time of the script or memory usage (in php laravel). After a chunk has been fetched you call the deepl api for the amount of chunks we had (I gave it a 4 minutes look and maybe I missed a bulk upload, sry for that). And then save the new translations with the target language_id and locale. In addition to this for each entry there should be created a new translation_checks entry with empty oldvalue parameter.

To run multiple requests at once a asynchronous javascript nodejs implementation could make sense, php will take a bit longer but for sure has to manage the translatable part anywhere so it makes sense to have everything translation related in the same place for debugging reasons.

Every given timeframe (24 hours, 12 hours, 7 days or on demand) there should be a lookup for new translation_check entries and responsible persons should be informed by a teamshook or automated email(makes sense if the platform is growing to keep track of changes and reduce required communication between departments). The responsible marketing people/translators should now have a interface to approve the changes one by one. In the frontend should also be a possiblity to make changes to the given translation to fix some errors made by the api as soon as they show up.

Frontend

The frontend page for the translator or marketing responsible doesnt have to be fancy, maybe a little http login for authentication or implemented in the software itself (if your staff is logging into a admin system or so). A little textbox with the new translation, the old translation and on demand the original value it has been translated from. Changes can be done in the textbox immediately. On a side or top nav you may also can select between languages to limit the translations to a domain if you have different responsibles for different language domains.

Possible Workflow