# NTIC Project Notes

Lexicon updates (daily) -> create a web interface that we can access these lexicons with.

More or less 10 languages (5 or 6 active, FR EN IT DE GR)

FR - EN for example, the pairs

Huge lexicons, thousand and thousands for each language

\*\* Up to now the oberon-based software (local to PC) was used. The aim of course is to \*\* Update this software and adapt it to the web-interface (including management and visualizations)

The problem with the current interface is that the environment is old and might disappear soon. Additionally, there's a requirement to use ODBC to contact with Oracle, which could disappear as well. The aim is to go away from the local environment towards the cloud (web-based interface)

- The interface will strictly be web-based and computer-centric. The focus is on utilizing it via computers/desktops

• The model is pretty clear (text fields will be the same, same functionality, etc)

\*\* Difficulty is that there is a lot of information, and depending on the category of the word, meaning that data structure is quite complex. As well as using the right tool for the web-based.

• Flask was suggested as a framework (written in Python), we also of course have a tendency to Javascript

\*\* database is based in oracle and in SQL (meaning that connection to this database should be a consideration point for us when we're choosing a framework or so what)

• sets are used, encoded in one integer, sets of 32 elements maximum, and encoded in binary. there is an encoder/decoder process to manage the data and display

LOOKING AT THE INTERFACE:

- i.e: noun, horse, or verb (horse around), the bilingual will connect the base form. There's a large set.

- Verb arguments (number of arguments, subject and target) i.e: to give something to someone, would have 3 arguments, the subj obj and the indirect obj (and the argument order is important to, ie in english give to or give something to, mapping to donner a)

- internal numbers are stored and displayed as well (the ID number, arbitrary, stored in the database)

- bilingual setting is only correspondence between two lexemes

- for the monolingual, we have category, base form, and some properties about this word, interface will change a little bit depending on the relevance (i.e: DE has more genders than FR, so 3 will go to 2)

- phonetics, inflections, and you can have buttons to change

- adding new words -> check the data, drop down menus, choices, and insert!

- features (semantic), properties that we can think about (ie: human, count, etc...) and these features could be important to change the semantic meaning of the word in some contexts

- update button will send changes to the database

- words without translation are still valid (no translation) if you want a correspondence you will have to add in the other language and then link them

- variants are also considerations and those are built into the interface as well

\*\* interface on the left side, has the inflected, and the right side has the lexeme (properties of the word), this is due to the fact that we have two tables, and they are separated based on such, with connections between the inflected and the lexeme

//so i think since we are making an UI that is exactly the same as the current, we will probably just be able to do a boiler plate that contains all the same fields, and then connect the thing in the backend. so should not be too bad, a question of learning FLASK layout and connecting it with database

//we of course need access to the original interface (without ability to manage it preferably) so that we can do a one to one transfer of UI

translation system will always take the highest preference, so what is the point of ordering the correspondence? (because of the connection to a separate web-based extension that the wrote for translation of words in context)

//so far bilingual, monolingual, and collocations

collocation example: cold case, where case takes the same meaning but cold changes its meaning. Info on collocations should give you to monolingual database to look at. in the monolingual too, it should show the collocations. collocations can have translation as well, and so this also reflected in the bilingual

//start with monolingual -> bilingual -> collocations

//suggestion: enter inflection differently, so we dont have to develop it, but rather keep it to a dropdown list to connecting to words already in the database (rather than add new entirely)

<http://blackboxframework.org/index.php?cID=home,en-us>

//we might have to look at the blackbox framework detail notes to keep in mind some framework performance and proposed features so that we make sure we are up to snuff

//encode decode, and decode will allow us to have the repository, file open, and big text file, and decode will let the system do its work, but allows the system to create the subdirectories and stuff