Manon JACQUET – 3064101

**User Interface**

Explanations :

This app is an slang translator. It takes English slang word and give a translation in French and vice versa

Home page :

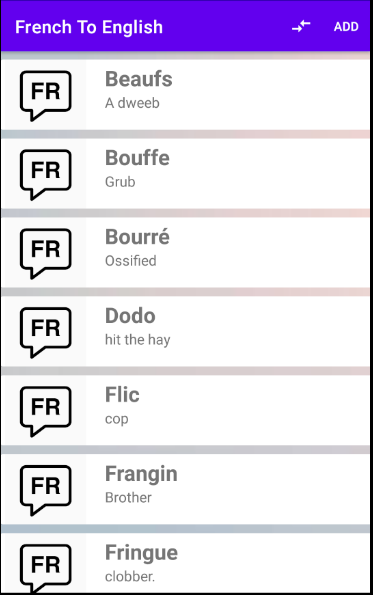
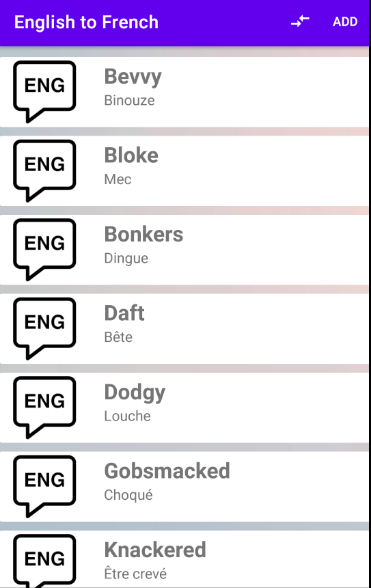
* This is the home page of the app.
* Tap on the screen to open the app

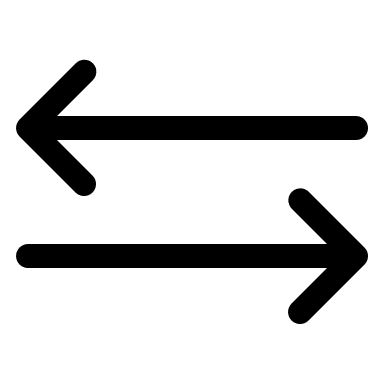
Then you can have access the app :

There are two parts in the application: The first one contains data that are already stored while the second one can be used to add your own data

Presentation of the first interface :

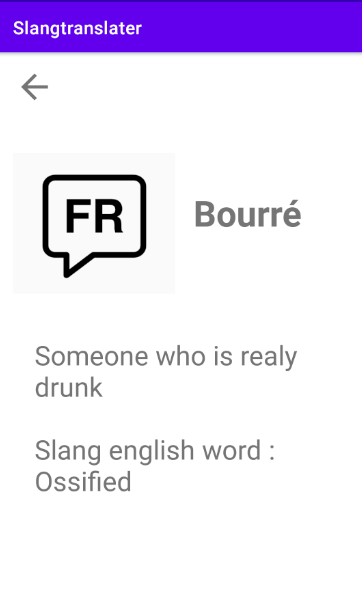
In this first interface, a list of pre-recorded slang words appears in alphabetical order, as observed in the following screen shots



²²

From the icon « double arrow » located in the top right corner of the screen, you can choose which language “French or English”, you want to translate

When you click on one of these words, a full definition is given in the other language . For instance :

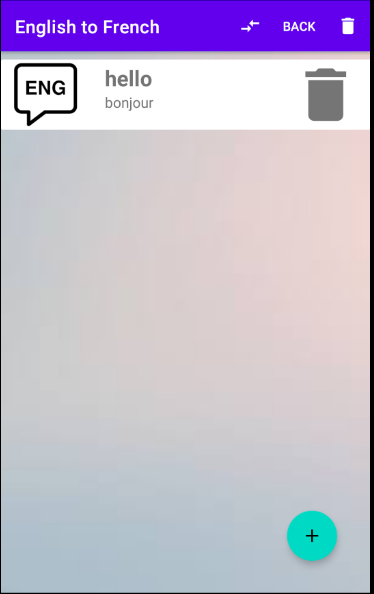


This is what you can see when you click on a given word.

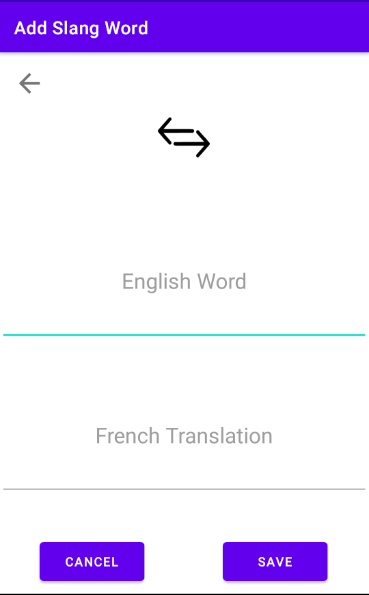
The arrow at the top left of the screen returns the user to the previous screen

The “ADD” button is an access to a new screen in which you can add your own translation, if you are not satisfied by the one which is given.

Second Part of the app:



Here we can see many elements:

* + - First, we can still find the double arrow. It has the same function as the previous one described in the first part. By clicking on it, you can have access to the second activity “French to English”
    - The trash at the top right of the screen can be used to delete all the words you have registered in our activity
    - The “Big” trash icon, next to the entered word is used to delete a word that contains a spelling mistake or that you do not want to see anymore
    - The “BACK” button returns the user to the first interface
    - The “+” button at the bottom left of the screen can be used to add a new word
    - You can still click on the word to get it in a new *activity*
    - This is what you have on the screen when you have clicked on the ‘+’ button to add a new word
    - When the user clicks on the button back or cancel, he comes back to the previous activity
    - The save button allows the user to add a word in the database.

**Documentation**

As I said before, my app contains two different parts, but this was not my first idea. At the beginning I wanted to create a list of multiple buttons. Some of them would have been visible and the others not. The invisible button would have been here to add new data. Thus, Thus, these invisible buttons would have been in the app to add new information . Then, when a user wanted to add a word, I would like to put a new information in one of these neutral button ( button with no information) and set their visibility to “true”.

I have started to use a “Scrolling View” to set up my first idea and then different buttons were added. But when I achieved that, I did not really like how it looked like. Moreover, I was not satisfied by the fact that people would not be able to add any new words they learnt and to enter as much data as they wanted. Then I changed my mind.

During my research to improve my app, I found the “*Recyclerview*” that looked very well adapted for what I wanted to do. Thus, I begun to work on the first part with the data which were already stored. This task can be subdivided in 2 activities : on one hand, the part with English word translated into French and on the other hand the other way around from French to English. Here, I mainly describe how I worked on the first activity, from English to French, but I applied a similar approach to develop the French to English part.

First, I created different <string-array> in the strings.xml file, and then I build my *main activity* with a “Recyclerview”. To use this recycler view and add the data I wanted, I needed 2 things: first a layout which would be the “visible part” of the app and then an “adapter” to link the layout and the recyclerview.

The layout was the easiest part of the app because it was just an XML program. But I spent much more time to the understand the “recycler View” how it works and what information it will have to provide. Thus, I created an adapter class and a “sub-class” named “MyView “. This sub-class allows to pick up each element from my layout. Then, these selected elements are used in my “main” class. Thanks to this sub-class, I can set each element of my layout with the elements of my array. Moreover, I created a “clickable” layout for the user : when he clicked on one of these layout, he has access to more information. Once the adapter was finished, I had just to call it in my Recycler view activity to add each data in this RecyclerView. Similar steps were applied to build the “French to English” activity.

The second part of the application was created to give users the opportunity to add their own words. At the beginning, I imagined the user could add his words to the same place as those already stored. But to do that, one should have used the same adapter as the one used for the data already stored. But, because I did not use an array but a database, nor a basic string but an editView this action was not possible .

Thereby, I decided to create two new activities based on the first ones. I will describe one of them but both of each have the same operation. I thought to do two activities with data that could be directly stored in the existing database. However to prevent already recorded data from being overwritten by new ones, I finally thought it was more simple to separate the activities, instead of doing two other new tables.

Thus, I created a new activity with a “recyclerview” and I used the same principles as the ones described above, but adapted to my database. I still used the same layout (XML file described above) but the adapter changed. I still had my sub-class with the same elements but in my “primary” class, compared to the first adapter, I had to take the text, not in the array, but in the database, and I had to create a special part for the “delete” button. I made a link between the database and the delete button. Thanks to the position of each word in the database I can delete the right word. This button is invisible for the data already stored. Nevertheless, this adapter was almost the same as the first one.

The implementation of the database was the biggest part of this project. First of all, I created a table with three columns: the ID, the English word, then the French translation. Then I created a method which allows to read and to pick up these data. Once these functions were done, the database was created. In this database, I wanted to do three actions : add a word, delete one word and delete all the words. Thus, I implemented these 3 functions. Delete and delete all are link to the garbage which is on the principle layout. I can delete one element, or more, thanks to the ID.

“*Add*” transfers the text written by the user in the “edit view”, into the “Addword” activity. The first “add” was a pop-up but I changed for a new page because I did not like the final result. This activity changes the text in “editview” in string and transfers this text to the add function in the database.

I created some “menu” buttons. These buttons link each activity and allow to use all activities of the app. One of these (double arrow) is here to switch between “French to English” and “English to French” activity. When you clicked on it the activity requested is launched. The menu buttons “ADD” and “BACK” work the same way, but between the activities “add your own words” and “ have access to the data already stored”. To finish, in the activity where you can add your own word, the trash button is linked to the database and delete all added words. I made an “Alert dialog” as safety, in the case where the user clicks on the garbage inadvertently.

Finally, my only regret is about the “sort” function that did not work properly. Indeed, I wanted to use the “by order” command. The idea was to first add the word and then sort the database, but with the same “onclicklistener” (add button). The function “sort” was called but nothing happened, although I used the exact same way as the oneI used in the second tab(second activity with French to English word).

To sum up, It was really exciting to improve and develop my project step by step and I think I can say that I am pretty proud of what I have done!

**References**

**Tutorial Video:**

# Difference Between Linear, Relative and some other layouts in android : https://www.youtube.com/watch?v=7OfSevZlvM0

**Android Studio - Pop Up Window : <https://www.youtube.com/watch?v=4GYKOzgQDWI>**

# SQLite + Android - Insert Data in Database Table (Book Library App) : <https://www.youtube.com/watch?v=RGzblJuat1M&list=PLSrm9z4zp4mGK0g_0_jxYGgg3os9tqRUQ&index=3>

# Android adding values using a button in ListView – Tutorial : <https://www.youtube.com/watch?v=ws_p8LJ4Uq8>

# Android Dynamic Views Part 1 | Adding & Removing Views From Layout Using Java Code In Android Studio : <https://www.youtube.com/watch?v=EJrmgJT2NnI&t=75s>

# Options Menu with Sub Items - Android Studio Tutorial: <https://www.youtube.com/watch?v=oh4YOj9VkVE&t=230s>

# CREER UNE APPLICATION ANDROID : <https://www.youtube.com/watch?v=8CR40Dp1srw&list=PLMS9Cy4Enq5JnwAxe6Ao74qSTxxXjiw7N>

# How to Change the App Icon in Android Studio (With Adaptive Icons) : <https://www.youtube.com/watch?v=ts98gL1JCQU>

# RecyclerView + CardView - LAYOUTS AND CUSTOM OBJECTS - Android Studio Tutorial : <https://www.youtube.com/watch?v=Nw9JF55LDzE>

# Other sources :

# Tutoriel Android – TP : <https://hal.archives-ouvertes.fr/cel-01082588v1/document>

# OpenClassrooms lessons : <https://openclassrooms.com/fr/courses/2023346-creez-des-applications-pour-android/2023968-votre-premiere-application>

# Javapoint : <https://www.javatpoint.com/sql-order-by>

# Stack overflow : <https://stackoverflow.com/questions/21542825/java-sql-delete-row/21542836>