

LyricsClassic

PROJET PROGRAMMATION POUR LE TEXTE II

COURS PROPOSE PAR : XANTHOS ARIS

Table des matières

1.	INTRODUCTION	3
1.1	CONTEXTE	3
1.2	SOLUTION	3
1.2.1	FONCTIONS	3
2.	MATERIEL	3
3.	SPECIFICATIONS	4
3.1	DIAGRAMME D'UTILISATION	4
3.2	DIAGRAMME D'ACTIVITE	4
4.	BESOINS FONCTIONNELS	5
5.	BESOINS NON FONCTIONNELS	5
6.	SOUS ENSEMBLE ET PRIORITE D'IMPLANTATION	5
7.	INFORMATION DE MAINTENANCE	5
8.	ANNEXES	6
8.1	MAQUETTES	6
8.2	REQUETE API	8
9.	BIBLIOGRAPHIE/SOURCES	9

1. Introduction

1.1 Contexte

L'université propose des cours de traitement textuel, dans ces cours l'étudiant est amené à composer un corpus de texte à analyser. Pour composer ledit corpus l'étudiant peut laisser libre choix à son imagination ou peut être imposé par l'enseignant. Le programme proposé par l'université, Orange3¹, possède des widgets permettant l'analyse textuelle et la composition de corpus. Composer, par exemple, un corpus avec des pièces de théâtre a été facilité par un widget spécialisé, « Théâtre Classic », mais qu'en est-il de la composition d'un corpus composé de paroles de chansons ?

1.2 Solution

Pour remédier à ça, LyricsClassic va permettre la composition de corpus basé sur des paroles de chansons uniquement, afin de faciliter l'accès aux étudiants à ce type de données. Pour plus de détails, se référer aux maquettes (point 8.1).

1.2.1 Fonctions

- Accéder à une API stockant des paroles de chansons.
- Afficher la liste des chansons proposées par l'API.
- Faire une sélection de chansons.
- Extraire les paroles des chansons sélectionnées uniquement.

2. Matériel

Le programme va être codé en utilisant le langage de programmation *python*. Le tout étant intégré au code source d'Orange3 lié aux packages Textable-Prototypes. Les paroles des chansons seront récoltées via une API : Genius².

¹ <https://orange.biolab.si/>

² <https://genius.com/>

3. Spécifications

3.1 Diagramme d'utilisation

Le diagramme d'utilisation nous permet de voir toutes les actions des différents utilisateurs. Dans la figure 1 ci-dessous on peut voir par exemple les actions de l'utilisateur : il pourra sélectionner des chansons et composer un corpus.

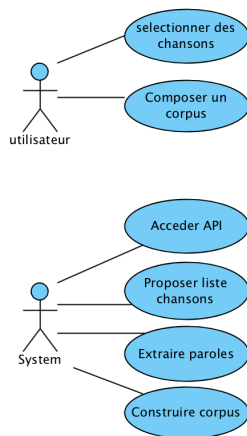


Figure 1: Diagramme d'utilisation de LyricsClassic

3.2 Diagramme d'activité

Le diagramme des activités (figure 2) transcrit visuellement les actions, à l'intérieur de l'application. Comment les actions de l'utilisateur s'exécutent en arrière plan.

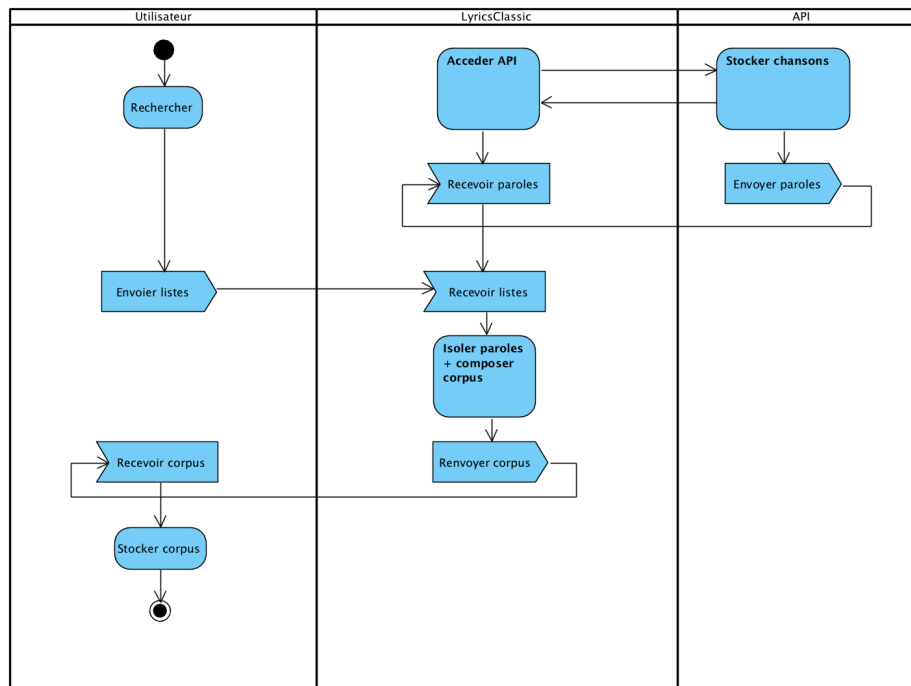


Figure 2 : Diagramme d'activité LyricsClassic

Dans la figure 2 on peut voir le chemin parcouru par l'information à l'intérieur du widget. L'utilisateur fait une recherche qui envoie une requête au Widget, qui questionne l'API, qui envoie les paroles, puis LyricsClassic isole uniquement les paroles et compose le corpus avant de le renvoyer à l'utilisateur.

4. Besoins fonctionnels

- LyricsClassique doit pouvoir accéder à l'API : Genius.
- LC³ doit pouvoir fournir une liste alphabétique de tous les artistes présents sur l'API.
- LC doit pouvoir fournir la liste de toutes les chansons de l'artiste présentes sur l'API.
- LC doit contenir les rubriques suivantes :
 - A, B, C, [...], X, Y, Z.
 - Listes par lettres des artistes.
 - Listes des chansons par artistes.
- LC doit permettre d'extraire ces paroles au format « .txt ».
- LC doit contenir la rubrique suivante :
 - exporter au format « .txt ».

5. Besoins non fonctionnels

- L'interface devra être basée sur des concepts simples, évidents à comprendre et à utiliser par tous.
- Les résultats affichés seront illimités dans le nombre disponible de chansons fourni l'API.
- LyricsClassic requiert une clé API Genius.
- LC requiert une connexion à internet.

6. Sous ensemble et priorité d'implantation

L'application ne sera pas disponible tant que toutes les rubriques ne seront pas complètes et opérationnelles. Par complètes on entend : Affichage total de la visualisation du package. Et par opérationnelles on entend : permettre l'exportation d'un fichier « .txt ».

7. Information de maintenance

ASAP

³ LC = abréviation pour LyricsClassic

8. Annexes

8.1 Maquettes

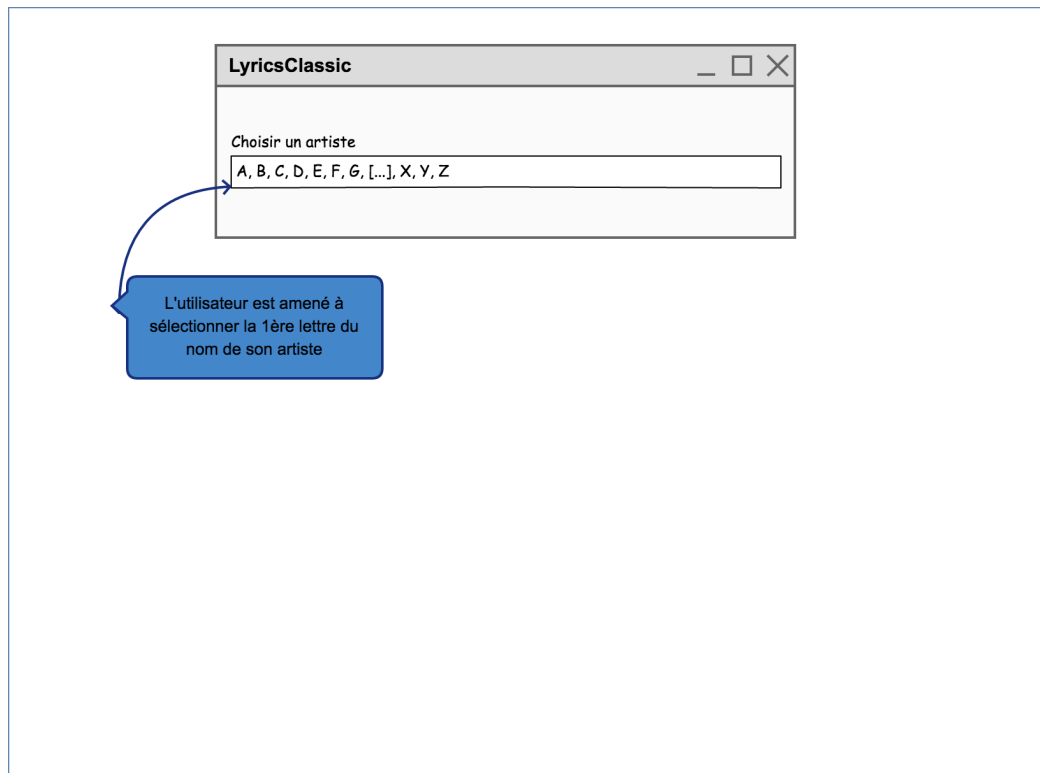


Figure 3: Base du widget LyricsClassic.

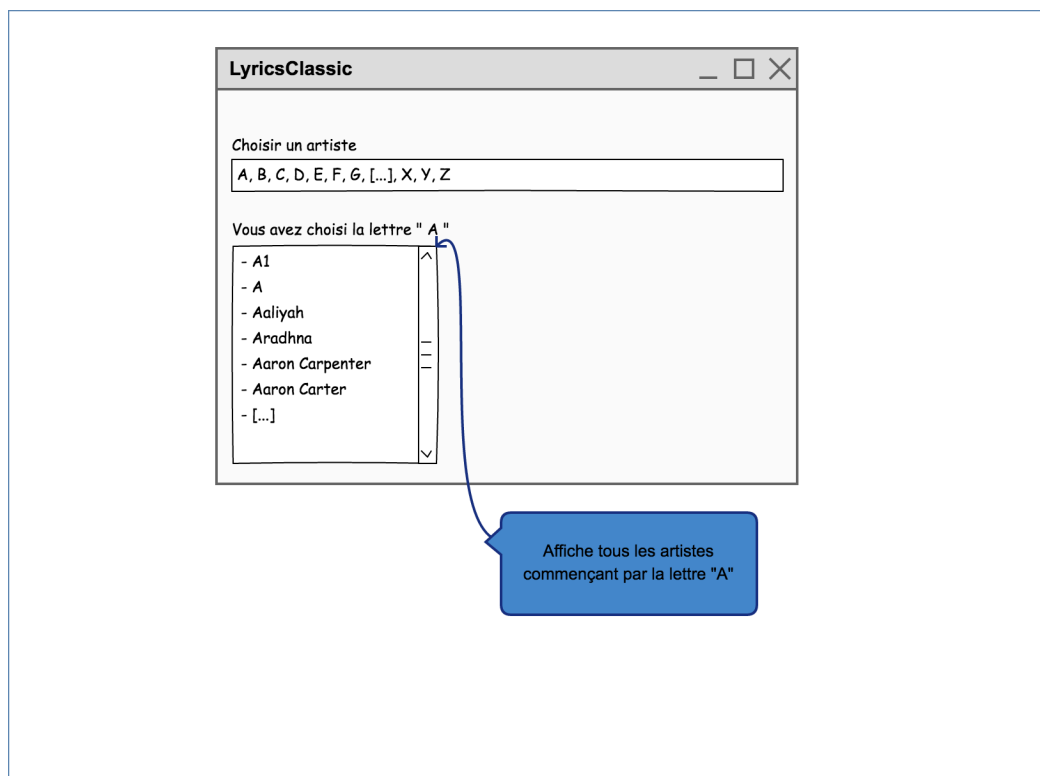


Figure 4: Base du widget LyricsClassic plus affichage du choix d'artiste commençant par la lettre A.

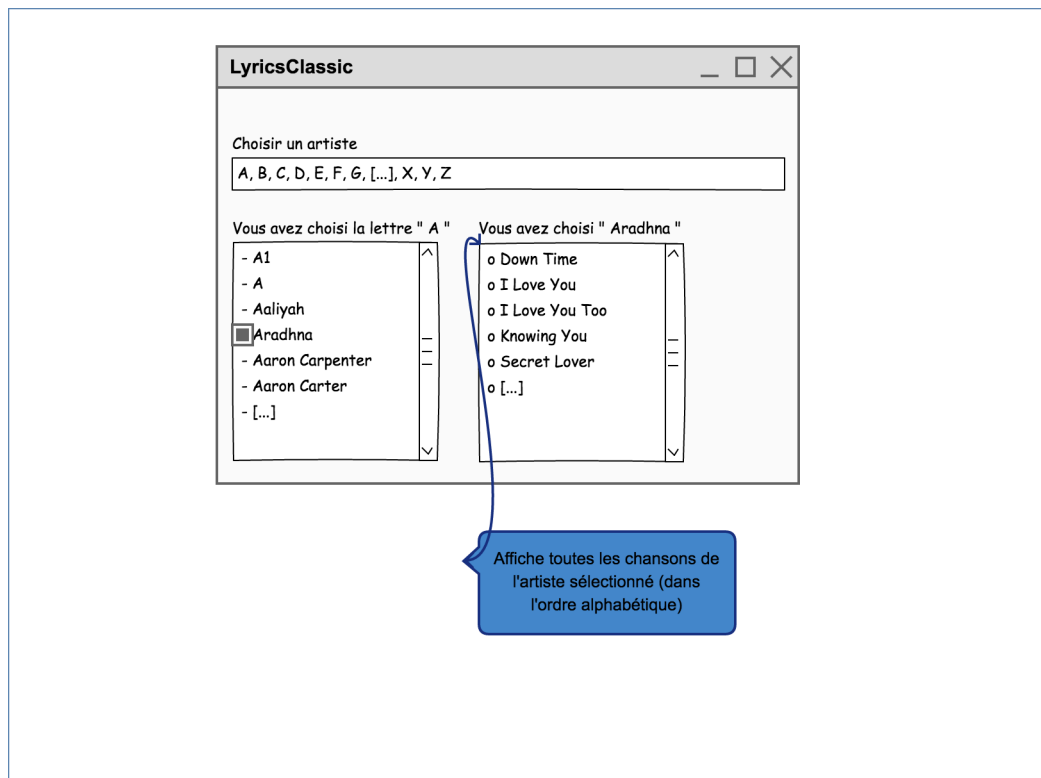


Figure 5: Base du widget LyricsClassic plus affichage des chansons disponibles après sélection de l'artiste.

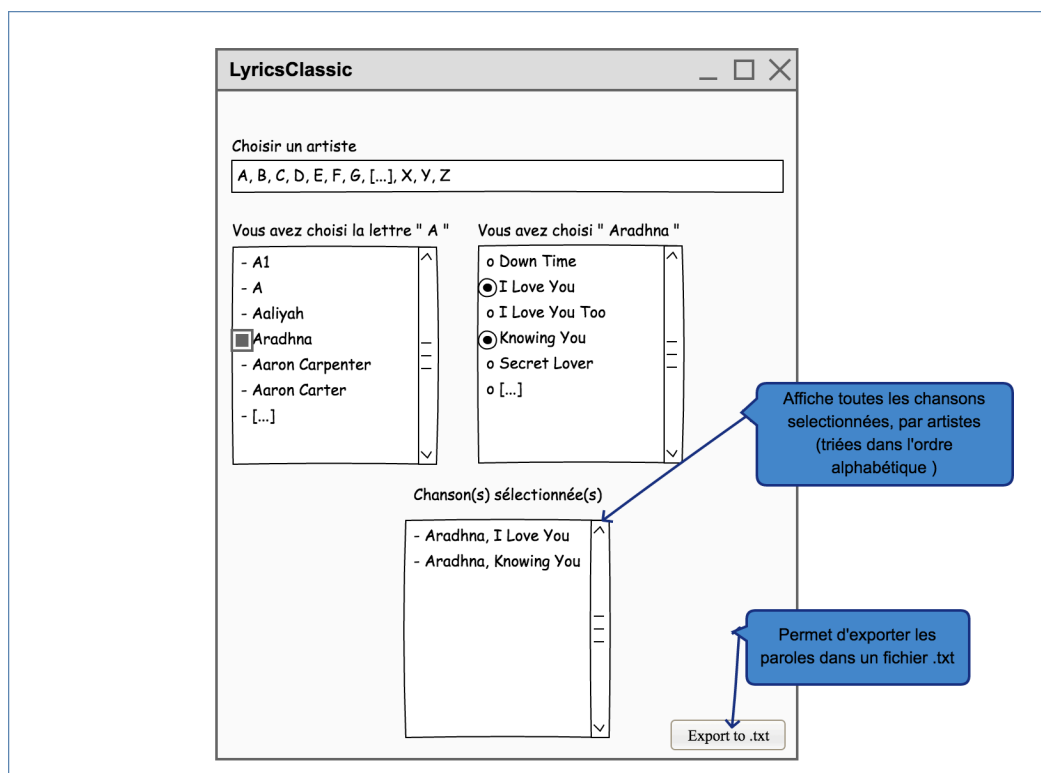


Figure 6: Base du widget LyricsClassic plus chansons sélectionnées depuis l'artiste ainsi que le bouton permettant l'exportation vers un fichier ".txt".

8.2 Requête API

```
>>> for result in body:
...     title = result["result"]["title"]
...     print(title)
...
HUMBLE.
m.A.A.d city
Swimming Pools (Drank)
DNA.
Money Trees
Bitch, Don't Kill My Vibe
XXX.
Poetic Justice
King Kunta
Backseat Freestyle
>>> for result in body:
...     path = result["result"]["path"]
...     print(path)
...
/Kendrick-lamar-humble-lyrics
/Kendrick-lamar-maad-city-lyrics
/Kendrick-lamar-swimming-pools-drunk-lyrics
/Kendrick-lamar-dna-lyrics
/Kendrick-lamar-money-trees-lyrics
/Kendrick-lamar-bitch-dont-kill-my-vibe-lyrics
/Kendrick-lamar-xxx-lyrics
/Kendrick-lamar-poetic-justice-lyrics
/Kendrick-lamar-king-kunta-lyrics
/Kendrick-lamar-backseat-freestyle-lyrics
```

Figure 7: Requête sur l'API Genius, recherche de l'artiste "Kendrick".

```
>>> path = "/Kendrick-lamar-humble-lyrics"
>>> page_url = "http://genius.com" + path
>>> page_url
'http://genius.com/Kendrick-lamar-humble-lyrics'
>>> page = requests.get(page_url)
>>> html = BeautifulSoup(page.text, "html.parser")
>>> [h.extract() for h in html('script')]
>>> lyrics = html.find("div", class_="lyrics").get_text()
>>> lyrics.replace('\n', '\n')
>>> print(lyrics)

[Video Intro]
Wicked or weakness?
You gotta see this
Waaaaay (yeah, yeah!)

[Album Intro]
Nobody pray for me
It's been that day for me
Waaaaay (yeah, yeah!)

[Verse 1]
Ayy, I remember syrup sandwiches and crime allowances
Finesse a nigga with some counterfeits, but now I'm countin' this
Parmesan where my accountant lives, in fact I'm downin' this
D'USSÉ with my boo bae tastes like Kool-Aid for the analysts
Girl, I can buy yo' ass the world with my paystub
Ooh, that pussy good, won't you sit it on my taste bloods?
I get way too petty once you let me do the extras
Pull up on your block, then break it down: we playin' Tetris
A.M. to the P.M., P.M. to the A.M., funk
Piss out your per diem, you just gotta hate 'em, funk
If I quit your BM, I still ride Mercedes, funk
```

Figure 8: Requête sur l'API Genius, recherche des paroles de la chanson "humble" de Kendrick Lamar.

9. Bibliographie/sources

AUDIBERT Laurent, « UML 2.0 (IUT, département informatique, 1ere année).

FOURNIER Jean-Pierre, « Génie logiciel : cahier des charges et spécifications », 2002.

Visual Paradigm for UML Performing textual analysis, 2010.

Visual Paradigm for UML QUICK START, 2010.