

**FREE MUSIC ONLY**



**FMO**

**Adam – Lenny**

**2023 – 2024**

## Présentation du programme

Free music only est une application desktop au format mobile, cette application sert au partage et à l'écoute de musique libre de droit, on peut y créer des playlists ajouter et supprimer des musiques dans celle-ci, une recherche par nom de musique permet d'écouter davantage, dans notre application, on peut partager de nouvelles musiques sur l'application directement sur la page d'information de l'utilisateur.

Pour la réaliser, nous avons utilisé python, la partie graphique a été réalisée à l'aide de la librairie Kivy, nos données sont enregistrées dans une base de données sql, et les musiques et images sont enregistrées dans un google drive donc nous avons accès par appels api.



# Cahier des charges

| V1  | V2  |
|---|---|
| Interface utilisateur<br>(enregistrement, identification) | Accueil (tendance, music de même genre)     |
| Recherche musique (genre, artiste, nom)                   | Possibilité à l'artiste d'ajouter des music |
| Interface playlists                                       | Edit des playlists                          |
| Lecture de musique/playlists                              | Image pour les playlists/music/user         |
| Base de données (user, music, playlists, genre)           | Mise en ligne                               |
|   | Sécurité                                    |



## Répartitions des Taches

| Adam                   | Lenny           |
|------------------------|-----------------|
| Playlist               | Utilisateur     |
| Recherche              | Profil          |
| Graphique bibliothèque | Base de données |
| Play music(lecture)    | Sécurité        |
| Ajout de music         | Accueil         |

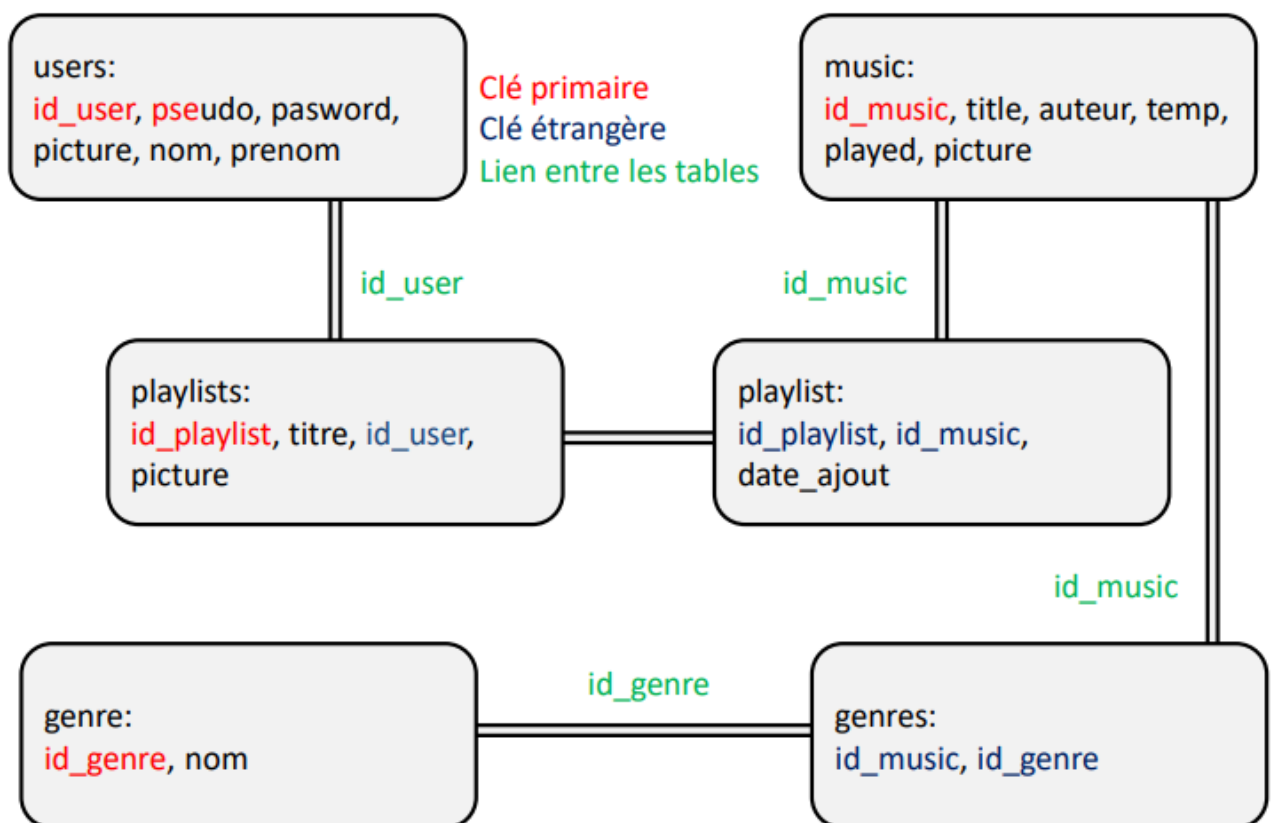


## Variables globales, Principales classes et Fonctions.

Les principales classes sont :

MainApp, User, Music, Connection\_Inscription, Interaction\_sql,  
et les classes de chaque pages

Les tables de la Base de Donnée :



## Présentation du déroulement du projet

Nous avons commencé par faire un cahier des charges, nous avons énormément d'idées beaucoup d'entre elles n'ont pas été réalisées, pas forcément par manque de temps, mais par manque d'envi après de nombreuses heures à se battre pour faire fonctionner cette appli, par exemple, nous voulions pouvoir mettre en place une page d'accueil qui proposerait des musiques en recommandant par rapport à ce qu'on écoute et nos préférences, c'est compliqué de faire ce genre d'algorithme de recommandation et à 2 ce n'était pas possible, ça demandait d'analyser chaque musique pour en trouver des similitudes on en était pas capables. On voulait aussi que la recherche soit plus poussée, que l'on puisse chercher par genre, ou par artiste mais ça demandait de faire plusieurs requêtes sql, et gérer chaque une d'entre elle graphiquement par onglets navigable, alors on a préféré faire seulement la recherche par nom de musiques.

Dans l'ensemble on a plutôt bien avancé ce qui nous a permis de ne pas être dérangé par le manque de temps, on avait plutôt bien anticipé ce qu'on voulait faire donc dès qu'on s'est réparties les tâches on a pu assez facilement travailler chacun de son côté, on a décidé de partager le projet sur GitHub et on avait donc aucun problème de gestion de nos fichiers. Après on communiquait par discord pour s'aider et montrer notre avancement.

Le projet a rencontré plusieurs gros problèmes, d'abord apprendre à utiliser la librairie Kivy n'a pas été facile j'ai eu beaucoup de mal à la prendre en main au début, par exemple les boutons de navigations entre les différentes pages et assez moche le dernier n'est pas aligné, les images sont aplatis, par contre quand plus tard j'ai fait les boutons du Player de la musique il fait plus beau tout est aligné, cependant j'ai pas du tout réussi à rendre les images dans les bonnes proportions pour qu'elles apparaissent correctement, si je ne les mettais pas comme elles le sont actuellement j'avais un carré jaune à la place, ce qui rend la compréhension de l'utilité du bouton encore plus difficile



qu'elle ne l'ait déjà. Pourtant, j'avais à l'avance fait des essais avec Kivy pour comprendre son fonctionnement puisque c'est notre librairie principale.

L'autre gros problème a été de jouer la musique, j'avais bien vérifié que Kivy avait un lecteur de musique intégré et je l'avais même essayé, cependant lorsque j'ai voulu mettre en pause la musique, augmenter le son, aller à un certain moment de la musique rien ne se produisait, je n'arrivais qu'à lancer la musique après de nombreuses heures à lire la documentation de Kivy concernant le jeu de musique, je ne comprenais toujours pas pourquoi ce que disais la documentation ne se produisait pas, j'ai donc directement été voir dans le code de Kivy, là où devait avoir été écrit les fonctions décrites par la documentation, et là j'y vis les fameuses fonctions avec pour toute comme seule instruction 'return 0', donc Kivy ne pouvait pas gérer la musique, j'ai donc décidé d'utiliser vlc, il était très facile et rapide à mettre en place et utiliser, cependant rapide ne veut pas dire efficace, puisque ce n'est que après que j'ai su qu'il fallait que vlc soit installé sur tous les ordinateurs où je voulais lancer le projet, du coup l'idée de transformer notre appli desktop en appli Android n'était plus possible, j'ai dû rajouter un vlc dans le projet et prévoir l'ajout d'un lien vers celui-ci dans les variables d'environnement de l'ordinateur pour qu'il soit utilisable.

Il a ensuite été facile de tout connecter à la base de données, on a rajouté le fait que les musiques soient enregistrées sur un google drive, et comme j'avais déjà prévu la classe pour gérer ça, ça a été facile de l'implémenter, on a un petit délai quand ils téléchargent mais pas trop long pour que ce soit trop gênant, d'ailleurs on voulait implémenter le fait qu'il télécharge au fur et à mesure la musique par paquet de 30s mais ça n'a pas été fait.



## Partie personnelle

Ma partie consistait majoritairement à faire la partie graphique et la musique pour cela j'ai dû utiliser la librairie Kivy et python-vlc, j'ai écrit les classes suivantes :

MainApp : class principales du projet, gère la navigation et l'initialisation de toutes les pages

Music : classe qui gère la musique, la queue, play/pause, volume, etc.

User : classe qui gère les données de l'utilisateur et les garde dans un json

Inscription : classe de la page d'inscription

Connexion : classe de la page de connexion

Home : classe de la page home avec le bouton qui mène a la page de l'utilisateur

Library : classe de la page librairie gère l'affichage et la création de playlists

Search : classe qui gère la recherche de musique et l'ajout dans une playlist

User\_Page : classe info de l'utilisateur et création de nouvelles musiques





## Algorithme :

```
def play(self):
    music = self.queue.pop(0)

    if music['url'] in os.listdir('download_music'):
        media = vlc.Media('download_music/' + music['url'])
    else:
        id_music_file = self.google_api.search_file_by_name(music['url'])[0]['id']
        self.google_api.download_file(id_music_file, 'download_music/' + music['url'])

        media = vlc.Media('download_music/' + music['url'])

        id_img_file = self.google_api.search_file_by_name(music['img'])[0]['id']
        self.google_api.download_file(id_img_file, 'download_img/' + music['img'])
        music['img'] = 'download_img/' + music['img']

    self.media_player.set_media(media)

    self.media_player.play()
    self.media_player.audio_set_volume(self.volume)
    self.current_music = music
    self.playing = True
    self.paused = False
```

Cet algorithme pas très compliquer est la manière par laquelle je joue la musique, pour récupérer la musique je récupère la première présent dans la queue et l'enlève on ne peut donc pas revenir à un musique précédente,

La musique est un dictionnaire qui contient c'est info grâce au faite que je l'ait avait déjà récupérer dans la classe user dans ses playlist, je vérifie que la musique n'est pas déjà télécharger en vérifiant que celle-ci n'est pas déjà dans le répertoires des musiques télécharger je vérifie pas le nom de la playlists, il aurait été plus judicieux que le nom de la musique soit son id de manière à ne pas avoir de doublons, si c'est le cas je lance la musique grâce à vlc, sinon je la télécharge via google drive en récupérant d'abord son id puis le fichier, par la même occasion je télécharge l'image de la musique pour l'afficher graphiquement, ensuite le lance la musique et ajuste le volume, je finis par mettre a jour certaine variable qui vont influencer sur certain boutons graphique et leur fonctionnement.



## Conclusion Personnelle

Je suis plutôt content du résultat, certes on a pas pu réaliser tout ce que l'on souhaitait mais en même temps ce n'était pas réalisable, ce très grand projet ne pouvait pas être parfait à 2 en seulement 2 mois, on s'est beaucoup inspiré de Deezer et Spotify pour la réaliser et eux ont conçu leur applis en plusieurs années avec plusieurs milliers de développeurs, alors pour ce qu'on a réussi à faire je suis plutôt satisfait, on a pu utiliser toutes les librairies qu'on voulait pour avoir un rendu je trouve plutôt agréable à voir, bien plus que si ça avait été une page tkinter, dans l'ensemble l'appli permet pas mal de chose et j'en suis content, cependant il y a énormément de possibilité d'amélioration de l'appli que ce soit dans le code, en fonctionnalité ou graphiquement, déjà laisser tomber vlc pour ffmpeg, retirer tous les endroits où on ne respecte pas l'encapsulation, mieux gérer le nom de fichier, faire une page home, permettre à l'utilisateur de se déconnecter, le bouton user de la page home qui disparaît quand on clique sur le Player de musique et qu'on revient à la page home, donc voilà beaucoup d'amélioration son possible.

Pour conclure on peut être satisfait de notre rendu qui dans l'ensemble permet de faire tout ce qu'on souhaitait, on a pu découvrir de nouvelles librairies qui sont sympas mais pas forcément les plus simples à utiliser, on a énormément de pistes d'amélioration possible pour avoir une application plus belle esthétiquement que dans son code, on peut en apprendre que python n'est pas fait pour ce genre d'application et que d'autres langages sont sûrement plus adaptés. Ce projet a été très plaisant à coder et l'utilisation des bases de données surtout quand elles sont accessibles en ligne comme là permet vraiment de faire des choses sympas sans trop alourdir le projet d'information dont il n'a pas besoin.

