

# Rapport

Antoine Lethimonnier - Alexandre Bec - Jérémy Rodrigues - Reuben Hattab

<b>I/ Organisation du projet</b>	<b>1</b>
<b>II/ Cahier des charges</b>	<b>1</b>
Exigences fonctionnelles et non-fonctionnelles	1
Diagrammes	3
Cas d'utilisation	3
Séquence	4
User/Constraint Stories	5
<b>III/ Conception</b>	<b>6</b>
Diagrammes de package et de classes	6
Patrons de conception	7
Principe SOLID	7
<b>IV/ GitHub</b>	<b>7</b>
<b>V/ Rapport de tests</b>	<b>7</b>

## I/ Organisation du projet

**Antoine Lethimonnier** ([WarningImHack3r](#)) : Structure du projet

**Alexandre Bec** ([4L3x7L](#)) : Lecteur de musiques et script de build

**Jérémy Rodrigues** ([SnKDaisuke](#)) : Tests

**Reuben Hattab** ([reuben-alt](#)) : Logger dans fichier

## II/ Cahier des charges

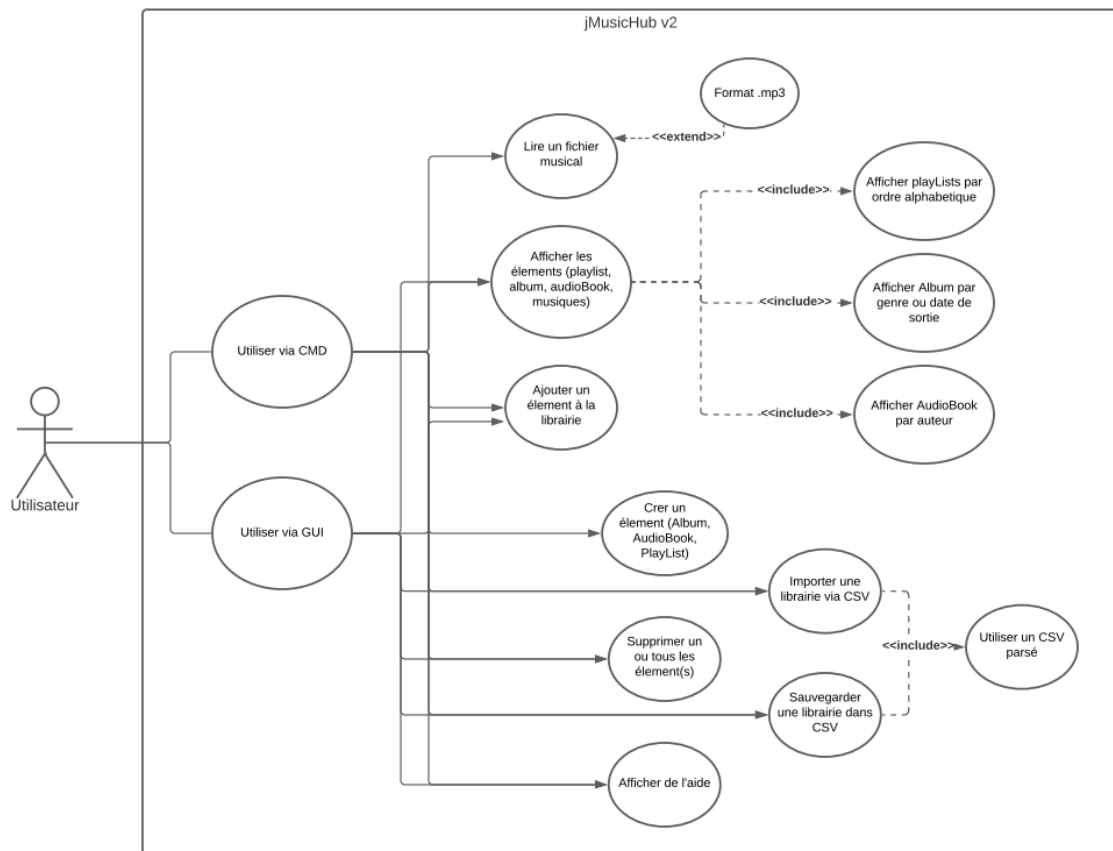
### Exigences fonctionnelles et non-fonctionnelles

ID	Nom	Type	Description
JMH_EXI1	Fonctionnalité de base	F	L'application permettra d'écouter de la musique
JMH_EXI2	Favoris	F	Créer un playlist
JMH_EXI3	Disponibilité	F	Utilisation gratuite
JMH_EXI4	GUI	NF	Interface pour

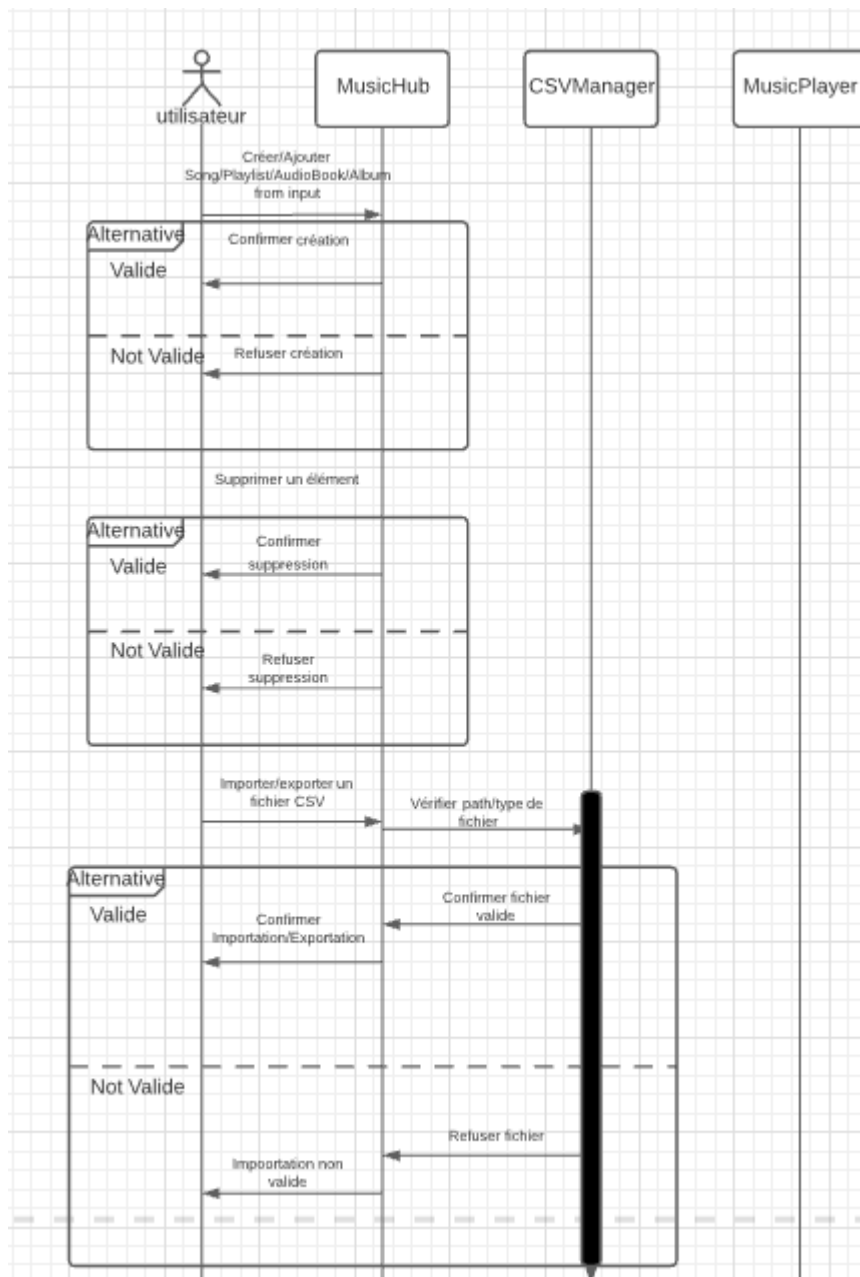
			utiliser l'app
JMH_EXI5	Lecture Musique	F	Lecture d'un fichier de musique au format .mp3
JMH_EXI6	PlayList	F	Création de playlist
JMH_EXI7	Filtre	NF	L'utilisateur peut filtrer les éléments de la librairie
JMH_EXI8	Aide	NF	L'utilisateur peut recevoir de l'aide si besoin (bouton)
JMH_EXI9	Suppression	NF	Supprimer toute la librairie ou un élément
JMH_EXI10	Affichage	F	Afficher les Musique, Playlist, AudioBook et Album
JMH_EXI11	Importation	F	L'utilisateur peut importer une librairie via un fichier CSV
JMH_EXI12	Sauvegarde	NF	Possibilité de sauvegarder la librairie dans un fichier CSV

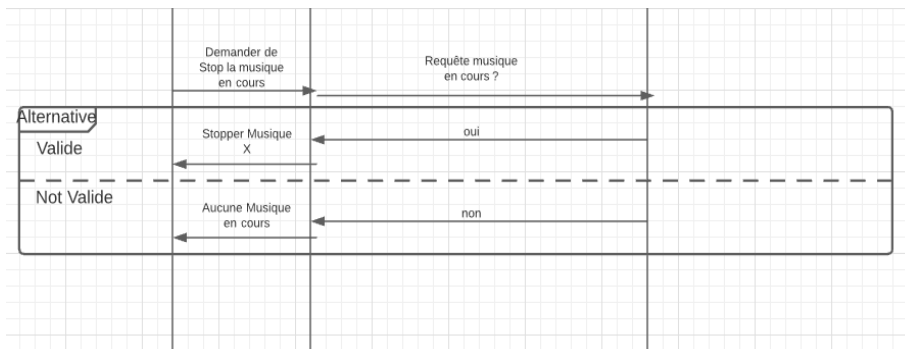
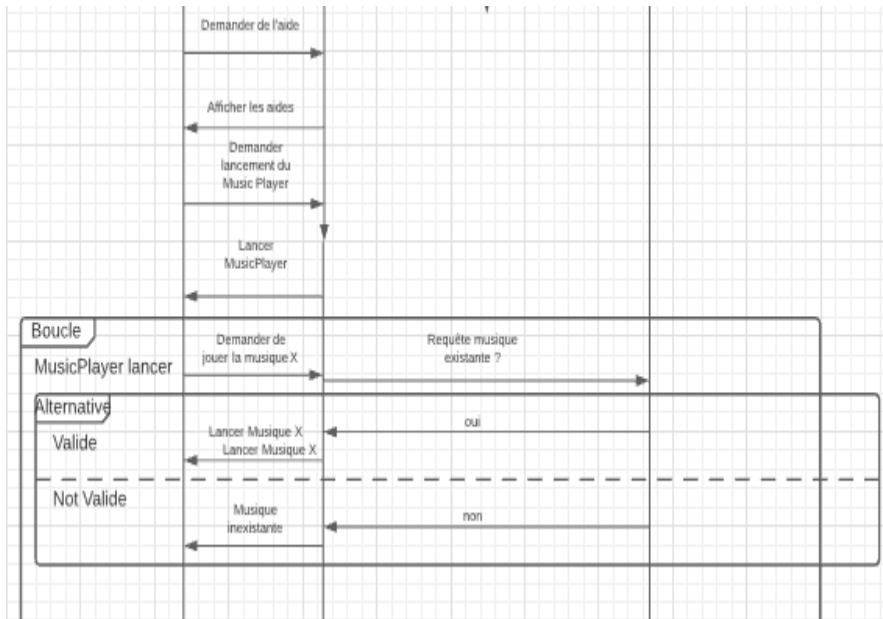
# Diagrammes

## Cas d'utilisation



## Séquence





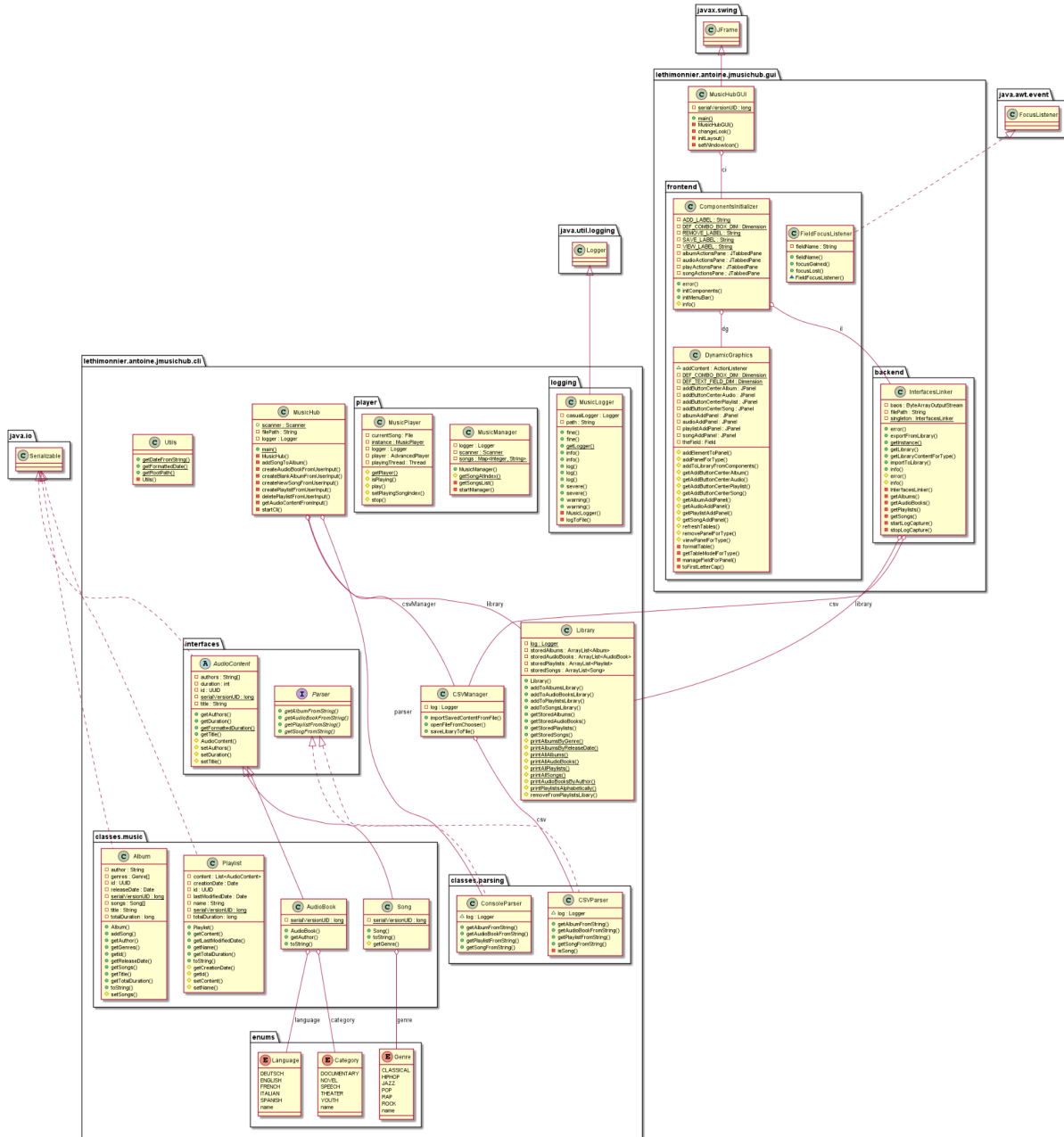
## User/Constraint Stories

Disponible dans Trello : <https://trello.com/b/eT7H9HGr/musicub-v2>

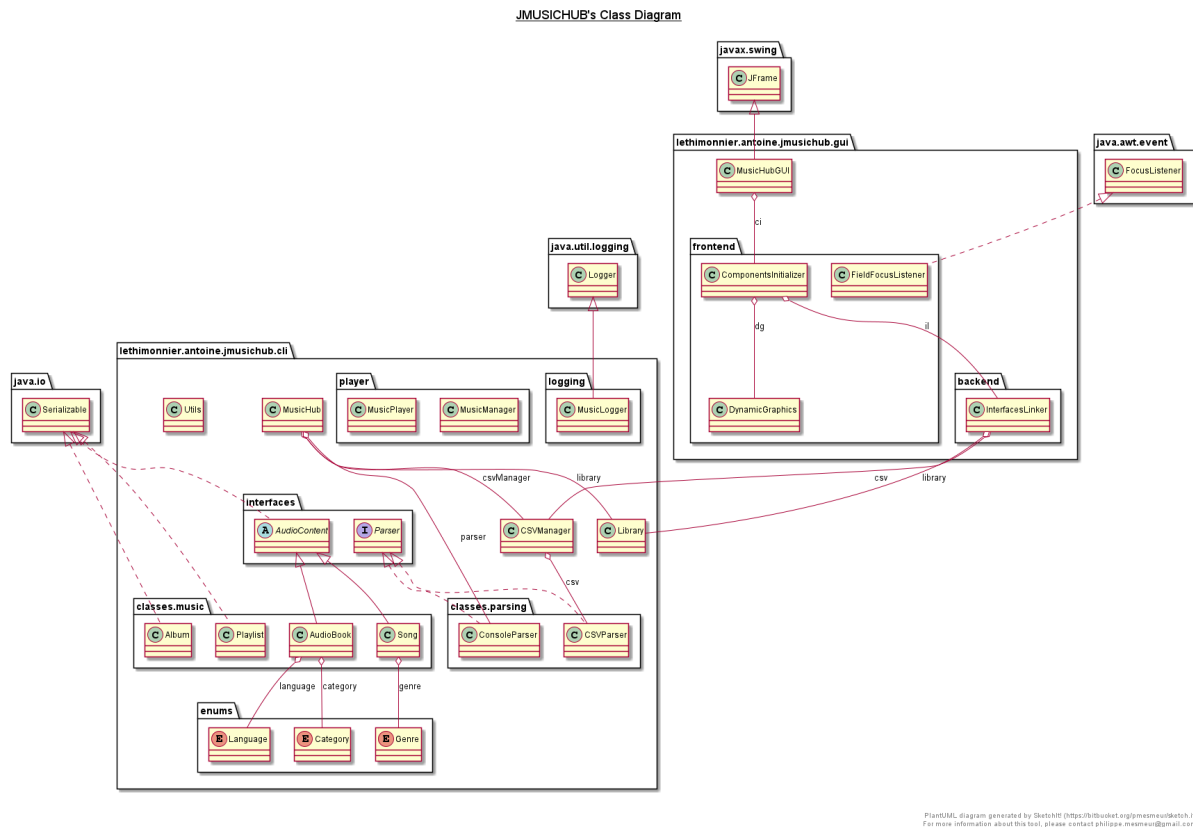
# III/ Conception

## Diagrammes de package et de classes

JMUSICHUB's Class Diagram



# Patrons de conception



## Principe SOLID

1. Single Responsibility : Chaque classe fait bien une et une seule chose, tout est séparé en classes individuelles.
2. Open/Closed : Toutes les variables et méthodes sont en private ou protected, sauf dans les cas où un public est nécessaire.
3. Liskov Substitution : Le principe est respecté
4. Interface Segregation : Les classes sont séparées
5. Dependency Inversion : Le principe est respecté

## IV/ GitHub

<https://github.com/WarningImHack3r/JMusicHub> (Repository privée)

## V/ Rapport de tests

- Tests d'acceptance exécutés :
  - Albums :  
Vérification des getters du constructeur Album sur les retours qu'ils effectuent lors de leur appel. (Réussi)
  - AudioBook :

Vérification des getters du constructeur AudioBook sur les retours qu'ils effectuent lors de leur appel. (Réussi)

- Playlist :

Vérification des getters du constructeur Playlist sur les retours qu'ils effectuent lors de leur appel. (Réussi)

- Song :

Vérification des getters du constructeur Song sur les retours qu'ils effectuent lors de leur appel. (Réussi)

- Library :

Vérification des getters du constructeur Library sur les retours qu'ils effectuent lors de leur appel. (Réussi)

- CSVManager :

Vérification du bon fonctionnement de l'import des fichiers CSV en testant que le fichier soit lu correctement et que les données soient correctement intégrées dans la Library. (Réussi)

- Liste des bugs :

Aucun problème rencontré