# jMusicHub - OOP Project Report

*Aimeric ADJUTOR*

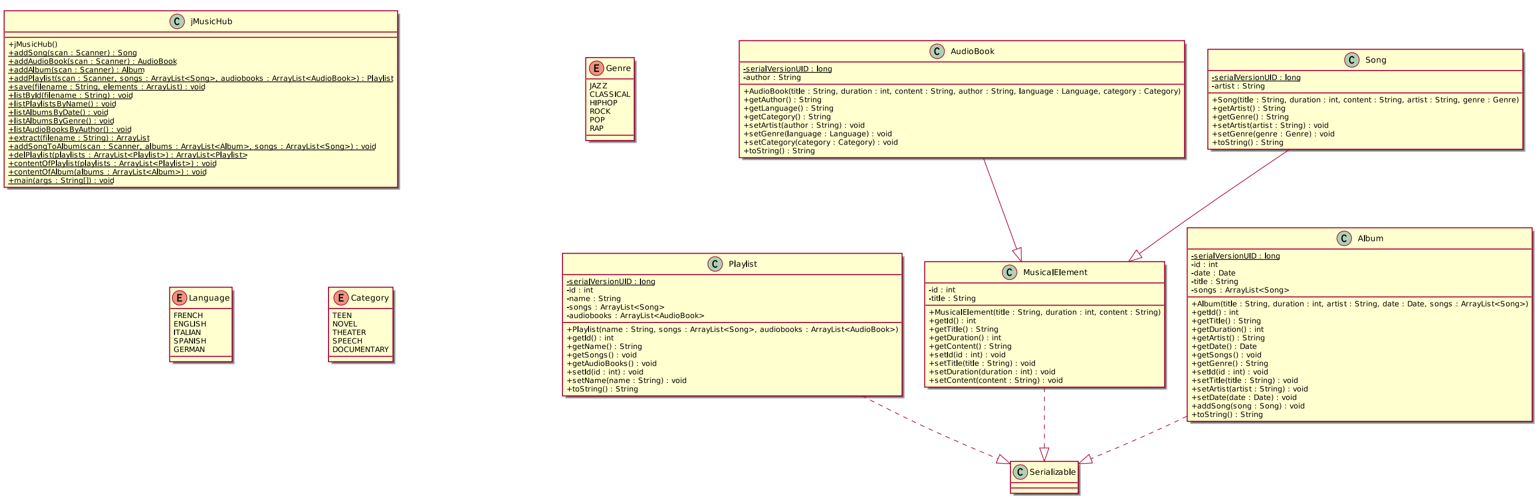
ESIEA

*ABSTRACT*

This document comes with a javadoc and files for a program named jMusicHub. Here, I'll try to state why I did some stuff the way I did, the tools I used, the difficulties, and so on..

## 1. UML Diagram

You can zoom-in easily since the quality of the diagramm is great



## 2. Student contribution

Since I did it alone, every part of the code has been done by myself.

I used different kinds of help when I was confronted to some errors and also to learn new things because I'm new with the Java programming language.

My main sources of knowledge comes from Stackoverflow, javaTpoint, Oracle and w3schools.

## 3. Work done

### 3.1. Tools

For starters, the tools I used for this project are :

- Neovim, as my text editor alongside some plugins and snippets to make my life easier.
- Void linux, as my OS.
- Deoplete, ultisnips and vim-javacomplete2 as my plugins.
- Plantuml and groff to, respectively, generate my UML diagram and my report.

### 3.2. Code

I coded with some freedom as the specifications were sometimes not clear. I suppose it was written this way to mimic specifications done by a client.

My "main" program is jMusicHub.java, it's basically the app you'll want to run to do anything. The other files are just the different objects that we create thanks to jMusicHub.

The whole thing can actually be launched by using the run file (./run). It's just a simple bash program that compile (javac) and launch (java).

Lots of commands are available so a quick peek at the help page (h) is recommended.

### 3.3. Abstraction and Interface

The abstract class is MusicalElement which is instanciated by Song and AudioBook.

The interface used in this project is the Serializable one. It is used by MusicalElement (therefore, Song and AudioBook), Album and Playlist. It's used to store the objects created into files. One for each types (songs, audiobooks, albums and playlists).

### 3.4. Exceptions

Most exceptions are handled the same way because they appear on the same patterns.

We basically abort the tasks or create empty objects when necessary and display something to the user to keep him up-to date.

## 4. Difficulties

I honestly had difficulties everytime I got back on my code to build new stuff. The websites, given at the beginning, were really helpful.

The first big difficulty was to understand what was the Serialization and how to use it. It took me some time but thanks to websites and indications given by some classmates I managed to implement it in my code.

Other difficulties were managed by, somehow, finding a solution online or aborting my idea and using other ways to achieve what I was looking for.