

**Rapport de projet JEE**

**Thème:** Implémentation d’une bibliothèque musicale en ligne

ligne horizontale

# 

# 

**GESLOT Baptiste AMADOU Asmaou**

**XICOLA Charles GAZZOLA Francesco**

# 

# Table des matières

[**Table des matières**](#_heading=h.tyjcwt) **[2](#_heading=h.tyjcwt)**

[**Introduction**](#_heading=h.1t3h5sf) **[3](#_heading=h.1t3h5sf)**

[**Diagrammes des classes**](#_heading=h.4d34og8) **3**

[Diagramme de classe des servlets](#_heading=h.2s8eyo1) [3](#_heading=h.2s8eyo1)

[Diagramme de classe des modèles](#_heading=h.17dp8vu) [3](#_heading=h.17dp8vu)

[**Modèle relationnel de la base de donnée**](#_heading=h.lnxbz9) **4**

[**Implémentation technique**](#_heading=h.1ksv4uv) **5**

[**Conclusion**](#_heading=h.44sinio) **6**

# 

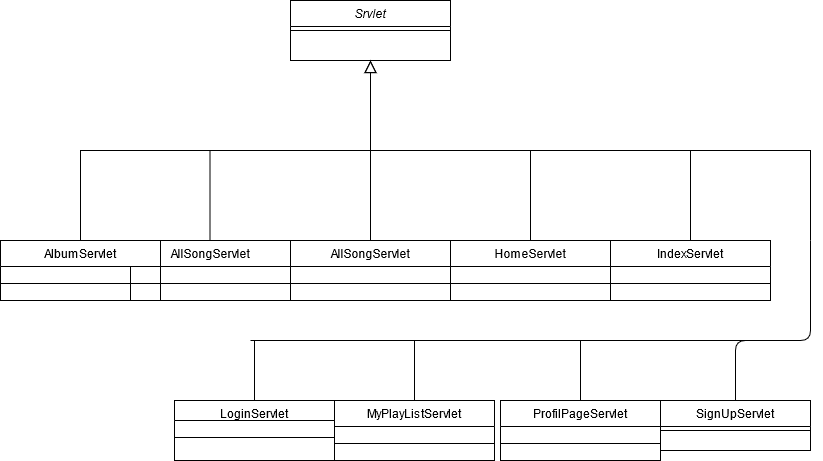
# Introduction

JEE (*Java Entreprise Edition*) est la version entreprise de la plate-forme Java qui se compose de l'environnement JSE ainsi que de nombreuses API et composants destinés à une utilisation côté serveur au sein du système d'information de l'entreprise.

Dans le cadre de l’apprentissage de cette technologie, nous avons développé une application web au nom de Musiqua. Cette application est une bibliothèque musicale. Ensemble, nous avons implémenté cette application, aussi en utilisant une base de données MySQL.

# Diagrammes des classes

## Diagramme de classe des servlets



# Modèle relationnel de la base de donnée

# 

La liaison entre la database et le serveur est faite à travers le module JDBC dans le RequestHandler. La base de donnée est composée de dix tables:

* users: ​ Cette table regroupe les différentes informations des utilisateurs qui se sont inscrits sur le site. Chaque utilisateur a un identifiant en plus de ses informations personnelles.
* playlist : Cette table contient les playlistes qui sont liées à un user ainsi qu’à des songs. Chaque playliste a un identifiant en plus de ses caractéristiques.
* listen\_to : C’est une association entre une playliste et un user.
* song : Cette table permet de regrouper les musiques. Chaque musique à un identifiant en plus de ses caractéristiques.
* album : Cette table contient les différents albums. Les albums possèdent un identifiant et on l'identifie de leur artiste en plus de ses autres caractéristiques.
* contain : C’est une association entre une song et une playlist.
* artist : Cette table regroupe les différents artistes. Chaque artiste a un identifiant en plus de ses autres attributs.
* genre : Le genre est une table qui contient une énumération de genre ainsi qu’un identifiant.
* likes : C’est une association entre un genre et un utilisateur.
* roles : C’est une énumération de rôles qui permettent de définir si un user est un client, un administrateur client ou musical.

# Implémentation technique

Front : Le Front-end a été réalisé avec la technologie jsp. Les différentes informations envoyées depuis le back sont récupérées en jsp. De plus, nous avons utilisé des fichiers CSS, de façon à obtenir un affichage en front aussi plaisant que possible.

Back : Le Back-end de l’application a été réalisé en Java, via des classes servlet​ . Nous utilisons les méthodes doGet et doPost pour afficher les pages.

Autres : Nous avons utilisé MySQL Workbench pour créer la base de données.

# Conclusion

Nous avons à travers ce projet découvert et mis en pratique différents outils vues lors de ce module, ce qui nous a permis de développer des connaissances en modélisation UML et la programmation en JEE. Ayant déjà travaillé sous MySQL et avec Java dans de précédent projet, c’est plutôt l’utilisation de la classe Servlet, le serveur TomCat ou encore les JSP qui étaient nouveaux. Le projet s’est déroulé sans incident, nous sommes globalement satisfaits du travail accompli, néanmoins avec un peu plus de temps nous aurions pu améliorer certains tels que :