



Rapport de projet

LAUGEL Emile

4A IA2R, 2022/2023





Sommaire

1. Introduction	3
2. Description détaillée de l'application retenue	3
3. Description détaillée de l'architecture mise en place	4
4. Framework et librairies utilisées	5
5. Conclusion	6



1. Introduction

Dans le cadre cet enseignement, nous avons pour but de créer un service web muni d'une interface. Pour ce faire nous avons été introduits plusieurs outils, tels que Java EE, Spring Boot, etc.

Tout d'abord nous devons choisir un thème, et celui que nous avons choisi et qui a été retenu est un service communiquant avec une base de données et fournissant des informations choisies parmi celles disponibles sur une ou plusieurs musiques telles que son nom, son artiste, ses paroles, sa durée, etc.

Mais notre service ne s'arrête pas à fournir uniquement les informations. Il est aussi capable d'y ajouter des informations à travers l'interface fournie avec.

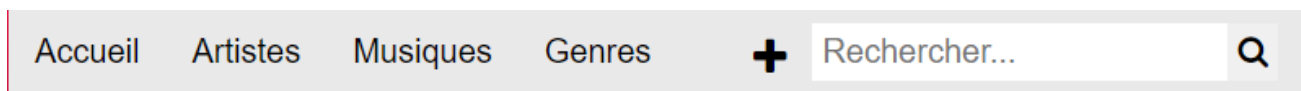
2. Description détaillée de l'application retenue

Pour commencer, l'application est composée d'une barre de navigation contenant 4 pages primaires et une barre de recherche avec son bouton correspondant. Une capture d'écran de la barre de navigation ci-dessous :



De plus, une barre rouge apparait en dessous de la page active ou si une sous-page de cette catégorie est active. Seule la page de recherche ni cette fonctionnalité ni une catégorie apparaissant sur la barre de navigation.

La barre de navigation ci-dessus correspondant à celle affichée uniquement dans la page Accueil et la page Recherche. Ci-dessous la capture d'écran de la barre de navigation affichée dans les autres pages :



Comme vous pouvez le remarquer, un nouveau bouton intitulé + s'est rajouté. Dans la suite nous allons expliquer pourquoi.

La page primaire Accueil sert à afficher l'objectif du projet, afin qu'un utilisateur puisse comprendre le but de ce projet sans avoir à lire tout ce rapport.

Les pages primaires Artistes, Musiques et Genres servent respectivement à afficher sous forme de bouton lien, les noms alternatifs des artistes, les noms de musiques et les noms des genres contenus



dans la base de données. Chaque bouton lien mène à une page contenant respectivement les informations de l'artiste, de la musique et du genre sélectionné. De plus, chacune d'entre elle est muni d'un bouton modifier et d'un bouton supprimer permettant d'effectuer ces actions dans la base de données.

Le bouton + présent uniquement dans les pages primaires Artistes, Musiques et Genres sert à ajouter respectivement un artiste, une musique ou un genre dans la base de données. En effet, celui-ci semble être un unique bouton mais en fait, ce sont trois boutons à actions différentes mais forme identique.

La barre de recherche sert à rechercher un contenu dans la base de données et son bouton sert à envoyer la requête. Le contenu est trié en fonction de la similarité, c'est-à-dire du plus ou moins similaire. Le contenu en lui-même est similaire à celui des catégories Artistes, Genres et Musiques ; c'est-à-dire que celui-ci est affiché sous forme de liens menant à la page de l'objet sélectionné.

3. Description détaillée de l'architecture mise en place

L'architecture mise en place est divisée en 2 parties : le front et le back

- Le back:

Afin d'utiliser notre projet sur un navigateur quelque conque car celui est quand même un service web, nous avons dû héberger un serveur. Et pour se faire, nous avons utilisé le programme Apache Tomcat que nous avons utilisé dans Java EE à travers l'IDE Eclipse. Après de longues configurations, le programme Tomcat nous a généré un serveur utilisable en local host sur un port spécifique.

Une fois le serveur en place, il nous fallait des pages accessibles depuis un navigateur. Nous avons ainsi suivi la description du cours que nous avons eu pour ce projet au début de l'année. Dans celui-ci, les pages étaient créées en format JSP (Java Server Pages) et étaient accessible sur le navigateur grâce aux classes Servlet qui sont en vérité le vrai format de la page. Le fichier JSP sert uniquement à donner une allure à la page, c'est-à-dire le front.

Une fois capable de créer une page, nous devons l'utiliser dans son contexte. Par exemple la page de la catégorie Artistes qui affiche les noms alternatifs des artistes sous formes de liens. Pour afficher ceux-ci, il fallait au préalable les récupérer. Pour ce faire, nous avons utilisé une base de données en SQL avec le programme MySQL. Nous avons créé de nombreuses classes nous permettant de se connecter à celle-ci, de vérifier son état et de la modifier.

Nous l'avons nommé javaee et y avons créé 3 tables : artistes, musiques et genres correspondant respectivement aux informations que nous comptons stocker.



- Le front:

Afin de voir et modifier les informations que nous avons sous la main, il nous fallait faire de la programmation dans les fichiers JSP énoncés ultérieurement. Ceux-ci ont été programmés en HTML couplé à du CSS et des framework pour la beauté visuelle du service web.

Pour communiquer avec le Servlet gérant la page, nous avons nommé les variables que nous comptons récupérer dans les fichiers JSP afin de pouvoir les récupérer dans le Servlet. Toutes les pages sont rédigées en HTML et CSS, avec certains contenant du Javascript. Nous avons créé un fichier menu.jsp que nous importons dans chacune des pages. Celui-ci contient toutes les méthodes CSS et javascript qu'une des pages pourrait utiliser pour afficher son contenu.

4. Framework et librairies utilisées

Dans ce projet nous avons utilisé les framework et librairies suivantes :

- Font awesome :

Font awesome est une librairie proposant des icônes internet permettant de rendre plus agréable la page html à l'œil.

Dans notre projet, nous l'utilisons pour le design du bouton + et le design du bouton lié à la barre de recherche car ceux-ci prennent très peu de place.

- Apache Commons Codec :

Apache Commons Codec est une librairie contenant de simples encodeurs et décodeurs pour une variété de format tel que l'hexadécimal et Base64.

Dans notre projet nous l'utilisons car notre base de données MySQL manque une fonction SQL nommée DIFFERENCE. Celle-ci sert à évaluer la similarité entre deux chaînes de caractères et renvoie une valeur comprise entre 0 et 4. Plus la valeur est élevée, plus les deux chaînes de caractères sont similaires. Nous l'utilisons pour faire le classement de similarité dans les résultats de la recherche.

- MySQL Connector :

MySQL Connector est un pilote permettant de faire la connexion entre Java à travers Eclipse et les bases de données dans MySQL.

Dans notre projet, il nous sert à accéder à la base de données en effet, mais après de nombreuses configurations telles que la création des tables, le port d'accès, etc.

- Jakarta Servlet jsp jstl :

Jakarta Servlet JSP (Java Server Page) JSTL (Java Standard Tag Library) est une librairie qui encapsule les fonctionnalités de base à de nombreuses applications JSP. La JSTL prend en charge les tâches structurales courantes telles que l'itération et les conditionnels, les balises de manipulation des documents XML, les balises d'internationalisation et les balises SQL.



Dans notre projet, elle nous sert à créer des boucles ForEach pour l'affichage de liste de boutons.

5. Conclusion

Le projet a atteint son objectif final. Tous les boutons, et méthodes fonctionnent à part une seule : le chargement de l'image. Pour une raison que nous ignorons, l'image que nous voulions charger ne pouvait pas l'être. Après de nombreuses recherches sur Google de tutoriels de chargement d'image avec Java EE, nous avons réussi à ne faire fonctionner aucun.

Cependant le service web reste complet dans son ensemble. Si nous avions eu du temps supplémentaire, nous aurions pu probablement trouver une solution au chargement d'image.