

Rapport de projet JEE

Thème: Implémentation d'une bibliothèque musicale en ligne



GESLOT Baptiste AMADOU Asmaou

XICOLA Charles GAZZOLA Francesco

Table des matières

Table des matières	2
Introduction	3
Diagrammes des classes	3
Diagramme de classe des servlets	3
Diagramme de classe des modèles	3
Modèle relationnel de la base de donnée	4
Implémentation technique	5
Conclusion	6

Introduction

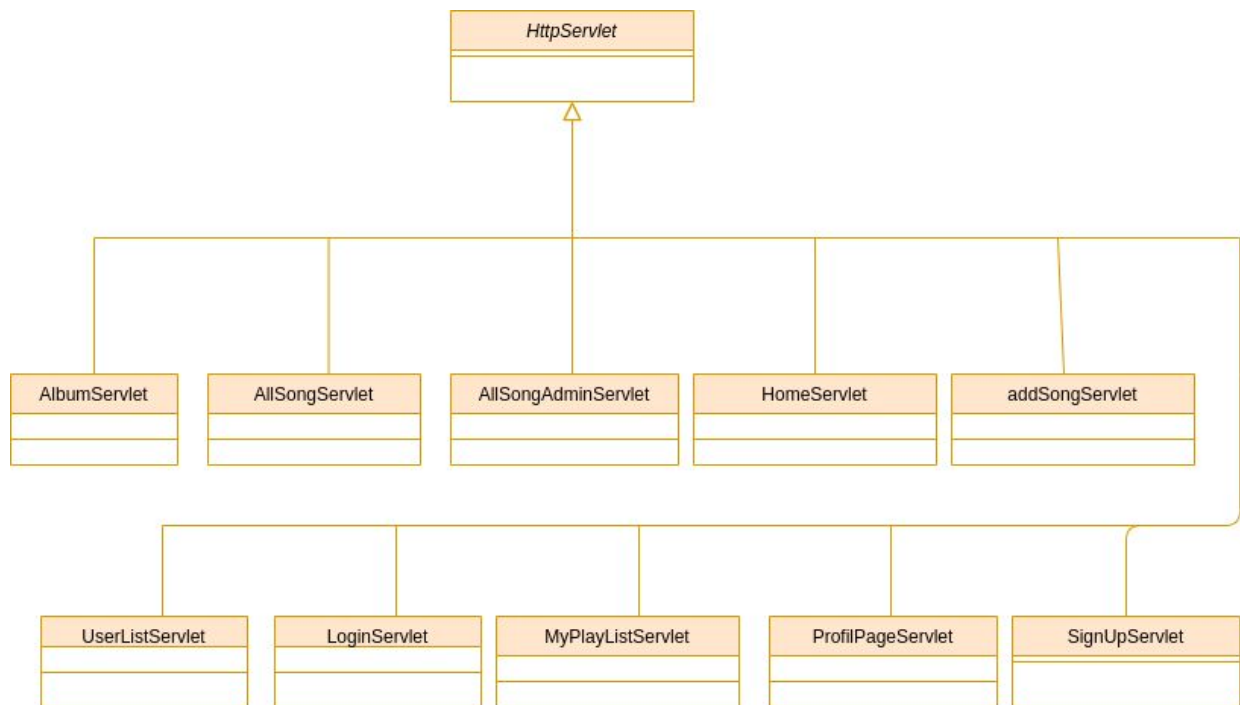
JEE (*Java Enterprise Edition*) est la version entreprise de la plate-forme Java qui se compose de l'environnement JSE ainsi que de nombreuses API et composants destinés à une utilisation côté serveur au sein du système d'information de l'entreprise.

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.

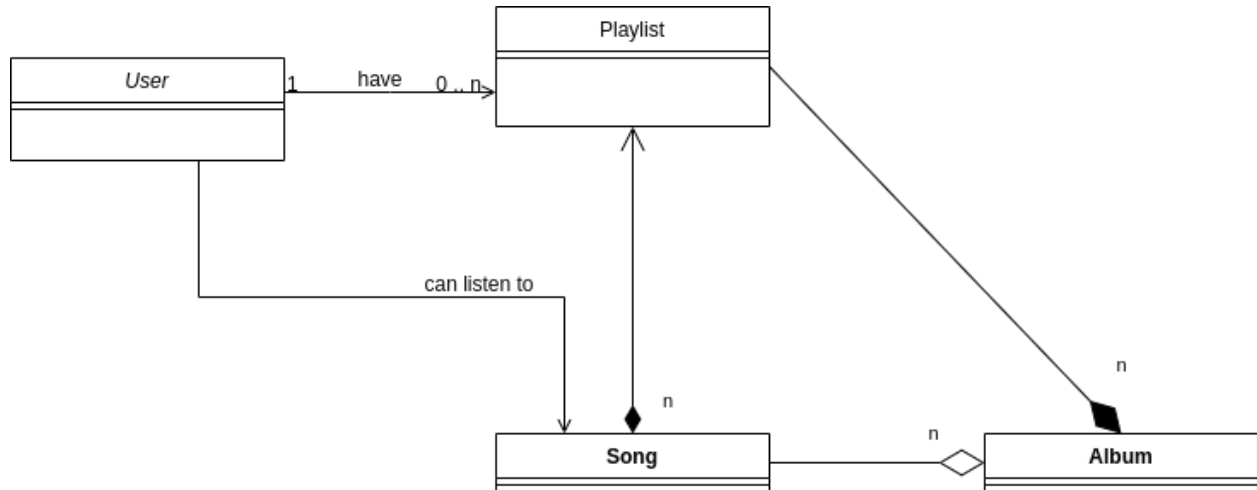
Diagrammes des classes

Diagramme de classe des servlets



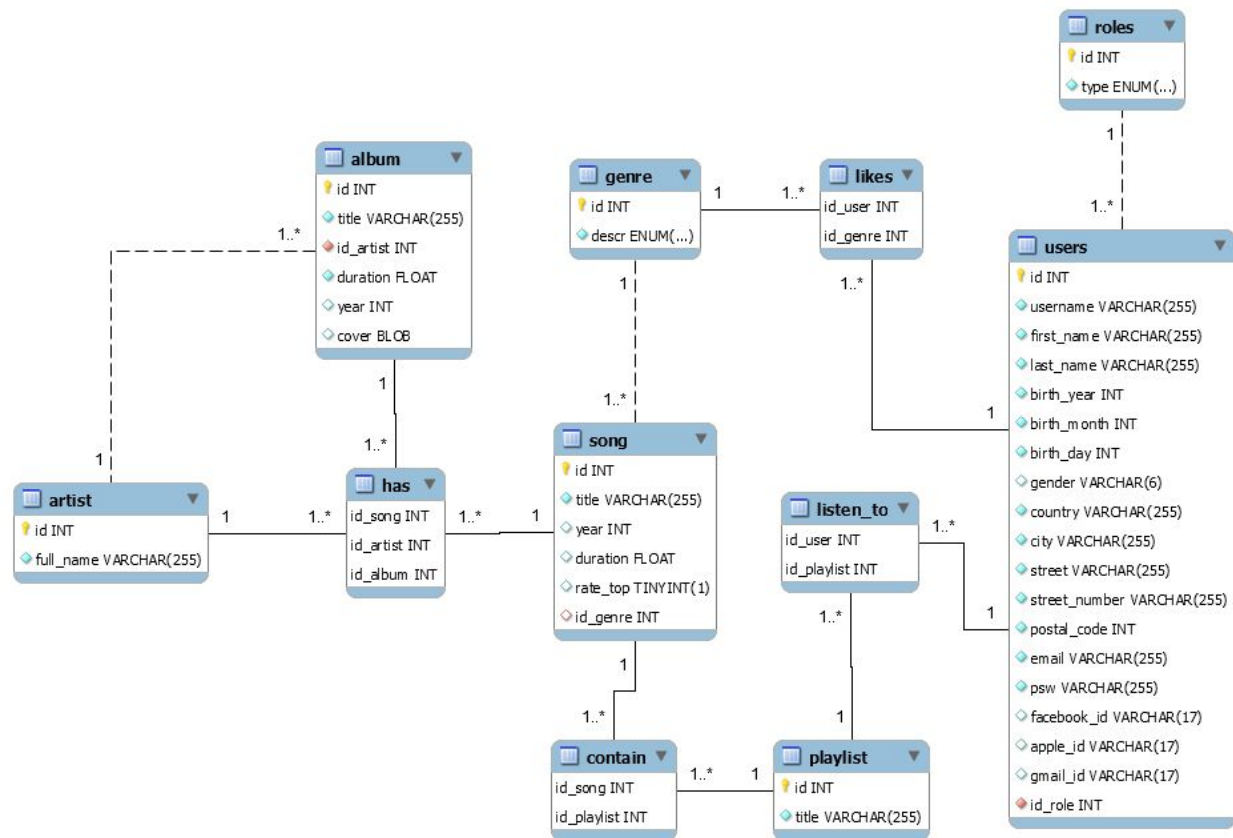
Toutes ces classes héritent de `HttpServlet` afin de créer une dynamique des données au sein du serveur HTTP.

Diagramme de classe des modèles



Une playlist est composée de plusieurs sons. un album contient plusieurs playlists. Un utilisateur peut avoir plusieurs playlists. Il peut choisir d'écouter un son.

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 playlists. Ces playlists sont uniquement accessibles en lecture pour l'utilisateur et en lecture et écriture pour l'administrateur. Chaque playlist a un identifiant en plus de ses caractéristiques.
- **listen_to**: C'est une association entre une playlist et un user. Cette table permet de savoir quelle playlist est associée à quel utilisateur.
- **song**: Cette table permet de regrouper les musiques. Chaque musique a un identifiant en plus de ses caractéristiques.

-
- album : Cette table contient les différents albums. Les albums possèdent un identifiant et elle contient l'identifiant l'artiste en plus de ses autres caractéristiques.
 - contain : Cette table permet de récupérer quelle song est associée à quelle 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 (listant les types de sons) ainsi qu'un identifiant.
 - likes : C'est une table qui contient le genre de son choisi par un utilisateur ainsi que son identifiant..
 - 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 pages sont écrites en HTML .De plus, nous avons utilisé des fichiers CSS pour styler les pages, de façon à obtenir un affichage en front aussi plaisant que possible. Cette partie nous a pris beaucoup de temps.

Back : Le côté back de l'application a été réalisé en Java, grâce notamment à aux classes servlet . Nous avons créé un servlet pour chaque JSP afin d'avoir un code plus fluide. Les différentes pages web s'affichent grâce aux méthodes doGet pour envoyer les requêtes et doPost pour récupérer des requêtes, en fonction des boutons et des liens sur lesquelles l'utilisateur appuie en front. Nous avons utilisé les cookies afin de pouvoir mémoriser les identifiants d'un utilisateur, ce qui lui permettra de naviguer entre les pages sans avoir à s'authentifier à chaque fois. Pour avoir accès au composants Java à travers les JSP nous avons utilisé les Expression Language. Nous avons aussi utilisé servlet-mapping afin d'associer une servlet à une URL dans le fichier web.xml

En ce qui concerne la base de données, nous avons créé la classe DBUtils qui contient toutes les fonctions nous permettant de manipuler des requêtes SQL. Nous avons aussi MyUtils qui permet de gérer les cookies. MySQLConnUtils qui permet la connexion au serveur MySQL via le fichier jar de MySQL connector. Enfin ConnectionUtils pour gérer la fermeture du serveur.

Etat d'avancement

Pour l'utilisateur : Il peut créer un compte et s'authentifier avec son email et son mot de passe. une fois connecté le username est récupéré(grâce au expression language) et affiché sur la page au niveau du profil. Nous avons pour cela créer une fonction 'findUser' dans DBUtils. Il a accès à ses playlists et à toute la bibliothèque musicale uniquement en mode lecture. Il peut aussi consulter les albums.

Pour l'administrateur : il peut gérer (ajouter, modifier ou supprimer) les playlists, les musiques ou les albums. Il peut aussi avoir accès à la liste des clients.

Nous avons aussi mis en place une mise à jour automatique de la page d'accueil car les tops du mois sont directement chargés depuis la base de données. Pour cela nous avons utilisé Expression language et des fonctions SQL.

Éventuelle amélioration :

La page de profile étant totalement fini, nous n'avons pas eu le temps d'écrire la fonction SQL permettant de récupérer les données de l'utilisateur et de l'afficher sur cette page.

Permettre à l'utilisateur de se connecter avec son username et son mot de passe car à ce stade il ne peut qu' utiliser son email et son mot de passe.

Implémenter une bar de recherche permettant à l'utilisateur de pouvoir effectuer une recherche sur la bibliothèque musicale.

Ajouter une fonction de hachage pour la gestion des mot de passe. Mettre en place une gestion de la liste des clients. Permettre à l'utilisateur de gérer ses playlists.

Optimisation de la base de données et l'implémentation de la récupération du mot de passe du client en cas d'oubli.

Conclusion

Nous avons à travers ce projet découvert et mis en pratique différents outils vus lors de ce module, ce qui nous a permis de développer des connaissances en modélisation UML et la programmation en JEE, HTML, CSS et JavaScript. 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.

Nous sommes globalement satisfaits du travail accompli, néanmoins avec un peu plus d'implication des deux autres membres du groupe nous aurions pu atteindre les objectifs fixés et avoir un résultat plus optimal.