**Rapport technique**



**Projet SoundHub**

**Auteurs**

Leroy, Maxime – 111 244 596

Longle, Henri – 111 244 597

**Équipe**

13

**Réalisé dans le cadre du cours**

GLO-2005 – Modèles et langages des bases de données pour ingénieurs

**Rapport présenté à**

L’Enseignant, Richard Khoury

**Remis le**

14 avril 2019

## Enonciation du problème et de ses exigences (2 points)

Aujourd’hui, YouTube est la plateforme dominante sur le marché du partage de contenus audiovisuel, et malgré sa croissance incessante, elle s’éloigne de plus en plus de sa communauté en axant davantage ses plans marketing sur des méthodes qui ont pour but de faire fructifier économiquement leur entreprise.

La communauté est donc mise de côté et c’est dans ce cadre qu’intervient notre application SoundHub, qui se veut être une plateforme novatrice utilisant les dernières technologies du web afin de combler les nouveaux besoins des utilisateurs de YouTube : un endroit où l’information mise en avant est de qualité.

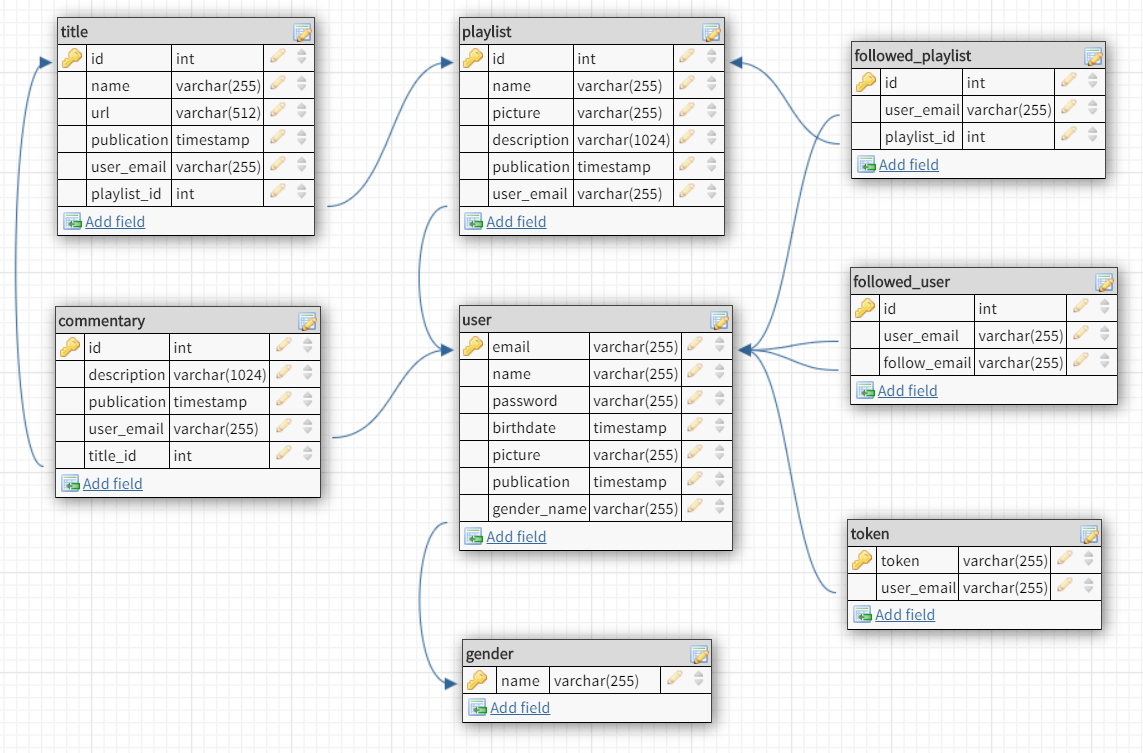
SoundHub est donc une application de partage de contenu musical de qualité, destiné à un cercle privé d’utilisateurs : il est impossible de consulter le contenu de la plateforme sans y être membre.

Le projet répond aux exigences suivantes :

* Créer un compte
* Se connecter
* Se déconnecter
* Afficher une liste de playlists
* Afficher les informations d’une playlist
* Afficher les titres associés à une playlist
* Afficher les informations d’un titre
* Afficher les commentaires associés à un titre
* Afficher sa liste d’abonnement aux utilisateurs
* Afficher les playlists que l’on suit
* Lire un titre via YouTube
* Lire un titre via un lien direct vers un mp4
* Editer une playlist
* Editer un titre
* Editer un profil
* Supprimer une playlist
* Supprimer un titre
* S’abonner à une playlist
* S’abonner à un utilisateur
* Se désabonner d’une playlist
* Se désabonner d’un utilisateur

## Modèle entité-relation du système (3 points)

## Modèle relationnel du système (3 points)



Légende :

*  : clé primaire
*  : clé étrangère

## Implémentation et fonctionnalités du niveau serveur de BD (4 points)

## Indexation des données, normalisation des relations, et optimisation des requêtes (4 points)

## Implémentation et fonctionnalités de la logique d’affaire (3 points)

## Implémentation et fonctionnalités de l’interface utilisateur (3 points)

## Sécurité du système (2 points)

Afin d’éviter que certains utilisateurs n’accèdent à des ressources auquel ils n’ont pas accès, Soundhub authentifie ses utilisateurs à chaque appel à la base de données via le système d’authentification OAuth. Quand un utilisateur se connecte, il entre son email et son mot de passe, le serveur vérifie les données et lui donne en échange un token d’authentification permettant d’accéder aux ressources du site. Ce token est unique et lié à l’utilisateur. Aujourd’hui OAuth est le protocole d’authentification le plus utilisé par les services web, de nombreuses entreprises tels que Facebook, Twitter ou bien même Google utilisent ce système.

Le mot de passe est haché grâce à la libraire Bcrypt. Bcrypt est une librairie de cryptage qui est basé sur l’algorithme de chiffrement par blocs « Blowfish ».

La librairie utilise un système de salage de mot de passe, en effet, pour éviter que deux données identiques possèdent le même hachage, des données supplémentaires sont volontairement rajoutées dans le hachage. De plus, afin de s’adapter au fil des années et aux nouvelles puissances de calcul des machines qui pourrait attaquer par force brute les informations, Bcrypt propose d’effectuer plusieurs passes sur le hachage. Dans Soundhub, Bcrypt fait 10 passes et le sel est généré par la libraire et n’est jamais sauvegarder ou divulgué.

La base de données est protégée des injections car toutes les recherches de données se font sur les clés primaires via l’ORM « SQLAlchemy » et aucune requête n’est donc construite à la volée.

Pour éviter l’insertions de mauvaises données dans la base de données, plusieurs systèmes de sécurité ont été mis en place. Dans un premier temps il y a une vérification des données par le JavaScript qui vérifie les données du formulaire entré. Par la suite l’api Python vérifie que les données ne sont pas vides et répondent bien aux différents critères. Finalement, c’est la base de données SQL qui à travers différents trigger (avant l’insertion et avant la modification) valide les différents champs, de plus des contraintes (clef externes, élément unique, élément non null) ont été mises sur les différentes tables.

## Organisation, gestion de l’équipe, et division des tâches (1 points)

L’organisation a été simple puisque tout le travail réalisé a été fait en la présence de la totalité des membres sur toute la durée du projet : du premier au dernier commit inclus.

La gestion de l’équipe s’est faite entièrement via un système de branches et de pull requests sur un github public créé avec le compte laval github de Maxime (le lien est précisé dans la section revue de code).

La division des tâches s’est faite de manière naturelle en rapport avec les facilités et points faibles de chacun, à savoir :

* Maxime: logique back end API (Python, Flask)
* Henri : configuration et installation de docker (Docker)
* Maxime et Henri : logique front end (HTML, CSS, JavaScript), logique back end SQL (MySQL)

## Revue du code (5 points)

## **Lien du code source**

<https://github.com/maximeleroylaval/bdd/tree/master/projet>