Table des matières

[1 Analyse préliminaire 1](#_Toc483380406)

[1.1 Introduction 1](#_Toc483380407)

[1.2 Objectifs 1](#_Toc483380408)

[2 Analyse 1](#_Toc483380409)

[2.1 Analyse concurrentielle 1](#_Toc483380410)

[2.2 Cahier des charges détaillé 1](#_Toc483380411)

[2.2.1 Définition de l’audience 1](#_Toc483380412)

[2.2.2 Définition du contenu et des fonctionnalités 1](#_Toc483380413)

[2.2.3 Maquette préliminaire 1](#_Toc483380414)

[2.2.4 Détermination des points faibles 5](#_Toc483380415)

[2.3 Stratégie de tests 5](#_Toc483380416)

[3 Conception 6](#_Toc483380417)

[3.1 Détermination de l’arborescence du site et des rubriques 6](#_Toc483380418)

[3.2 Définition de la charte graphique 6](#_Toc483380419)

[3.3 Maquette graphique 6](#_Toc483380420)

[3.4 Interrogation 9](#_Toc483380421)

[3.5 Conception du code 10](#_Toc483380422)

[3.6 Ressources 18](#_Toc483380423)

[4 Réalisation 18](#_Toc483380424)

[4.1 Dossier de réalisation 18](#_Toc483380425)

[4.2 Description des tests 19](#_Toc483380426)

[4.3 Erreurs restantes 23](#_Toc483380427)

[5 Conclusion 23](#_Toc483380428)

[6 Annexes 24](#_Toc483380429)

[6.1 Sources – bibliographie 24](#_Toc483380430)

[6.2 Procédure d’installation 24](#_Toc483380431)

[7 Table des figures 28](#_Toc483380432)

# Analyse préliminaire

## Introduction

Dans le cadre de mon travail pratique individuel je dois réaliser une application servant à écouter de la musique en streaming. Je dois aussi créer un site internet servant à gérer la base de données de cette application.

Objectifs de ce document :

* Présenter le processus d’élaboration du site web
* Documenter le code du site web
* Recenser les informations nécessaires à la reprise du projet

## Objectifs

Les objectifs du site web de gestion sont les suivants :

* Page de login ou création de compte
* Utilisateur
  + L’utilisateur peut uniquement consulter ses propres playlists
* Admin
  + Effacer un ou des utilisateurs
  + Supprimer les playlists d’un ou de plusieurs utilisateurs
  + Uploader des titres au format .mp3

# Analyse

## Analyse concurrentielle

Je n’ai pas trouvé grand-chose de ressemblant à ce site web. C’est selon moi juste un petit outil de gestion. Je n’ai pas besoin de beaucoup m’inspirer pour le réaliser.

## Cahier des charges détaillé

### Définition de l’audience

Le site web sera utilisé par les utilisateur de mon application ainsi que par les administrateurs qui pourrons gérer l’application depuis ce site.

### Définition du contenu et des fonctionnalités

* Les utilisateurs peuvent créer un compte.
* Les utilisateurs peuvent se connecter et choisir quelle playlist consulter
* Les administrateurs peuvent se connecter et gérer les utilisateurs, leurs playlists ainsi que les titres sur la base de données.

### Maquette préliminaire

Pour la réalisation des maquettes j’ai utilisé « Balsamiq online » disponible à cette adresse :

<https://webdemo.balsamiq.com/>

Voici les maquettes que j’ai réalisées avant de me lancer dans le projet :

Fenêtre de connexion :

L’utilisateur peut se connecter ou cliquer sur le lien « Inscription » pour ouvrir la page d’inscription.

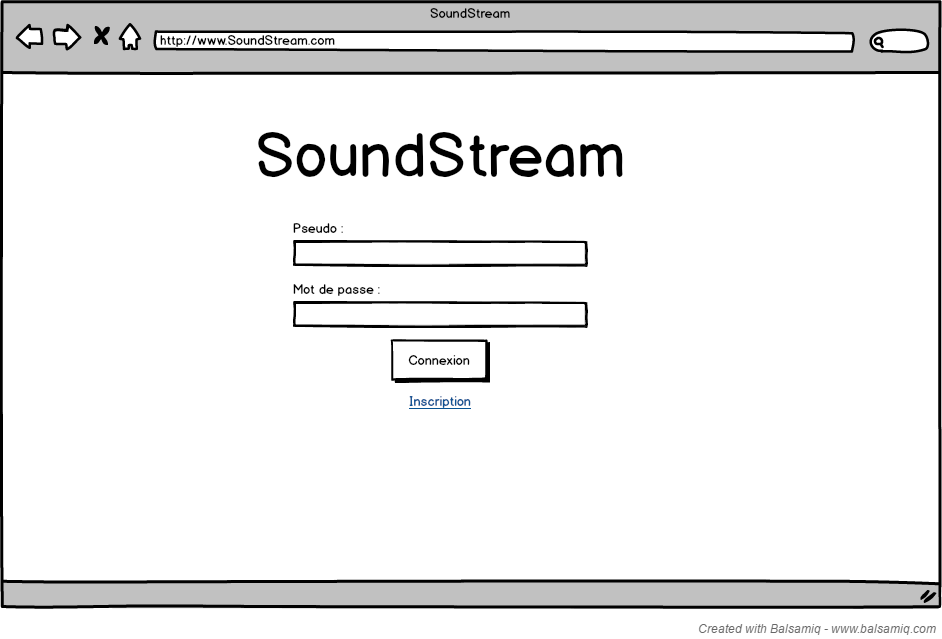


Figure 1 Fenêtre connexion

Fenêtre inscription :

L’utilisateur arrive depuis la fenêtre de connexion. Il peut soit s’inscrire soit retourner sur la fenêtre de connexion.

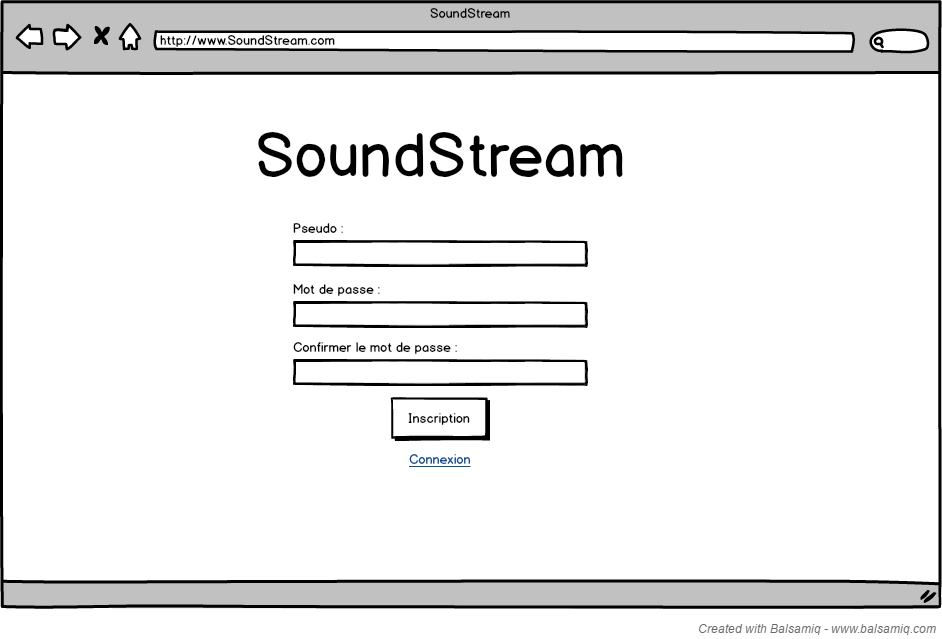


Figure 2 Fenêtre inscription

Fenêtre liste des playlists :

Une fois l’utilisateur, connecté il arrive sur une page avec son pseudo et une liste de playlists sur lesquelles il peut cliquer pour les consulter.

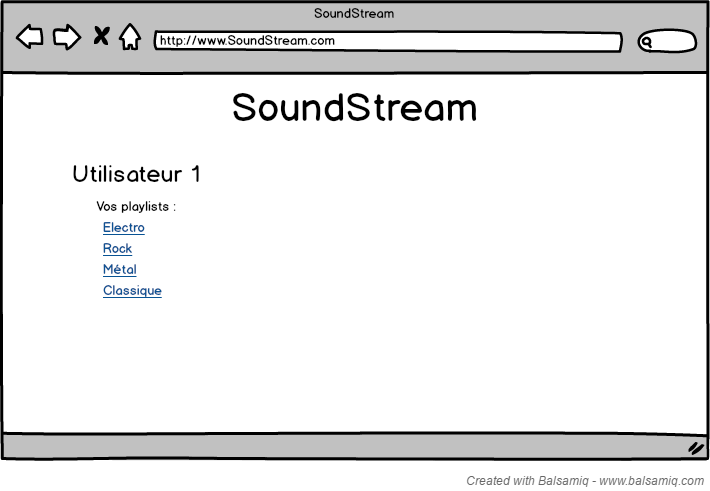


Figure 3 Fenêtre liste playlists

Fenêtre consultation d’une playlist :

Une fois que l’utilisateur a cliqué sur une playlist il arrive sur cette page contenant les informations sur la playlist qu’il a choisi.

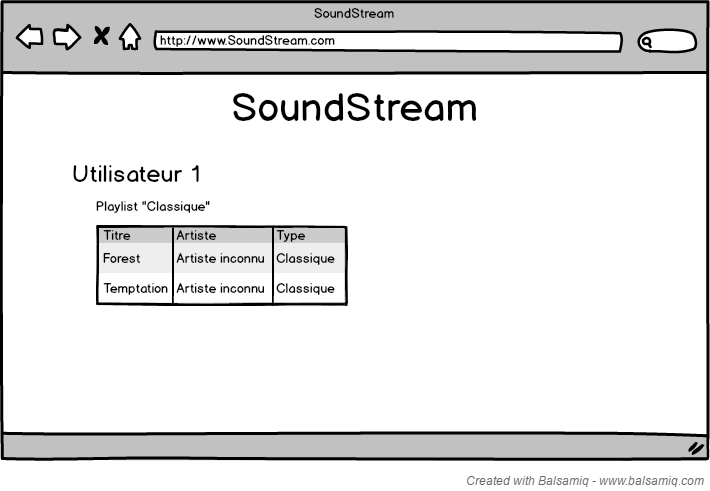


Figure 4 Fenêtre playlist

Fenêtre administration :

Une fois qu’un administrateur s’est connecté il arrive sur une page où il a deux liens, un pour gérer les utilisateurs et l’autre pour l’ajout de fichiers dans la base de données.

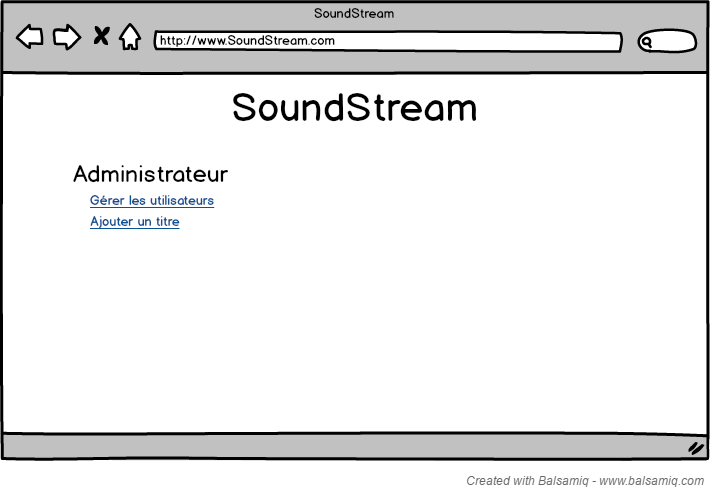


Figure 5 Fenêtre administration

Fenêtre gestion utilisateurs :

Une fois que l’administrateur a cliqué sur le lien « Gérer les utilisateurs » il arrive sur une page avec la liste des utilisateurs où il peut soit consulter leurs playlists ou cocher les utilisateurs qu’il veut supprimer.

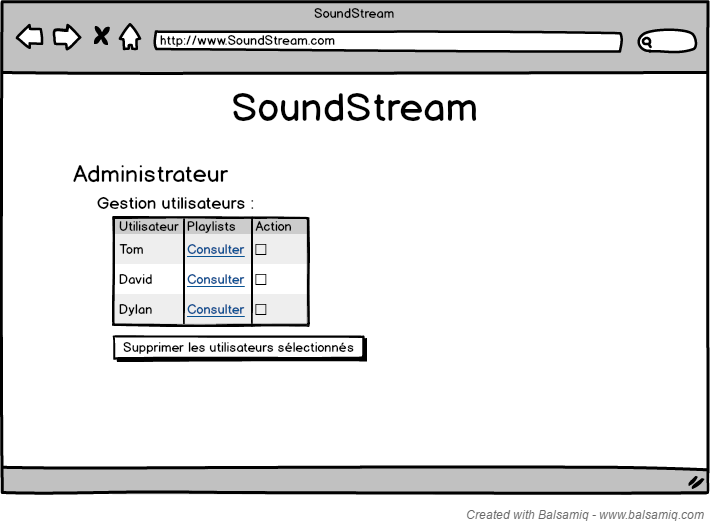


Figure 6 Fenêtre gestion utilisateurs

Fenêtre gestion d’un utilisateur spécifique :

Quand l’administrateur clique sur « Consulter » il arrive sur la page de gestion dédié à l’utilisateur sélectionné. Depuis cette fenêtre l’administrateur peut supprimer les playlists de l’utilisateur sélectionné ou le supprimer en personne.

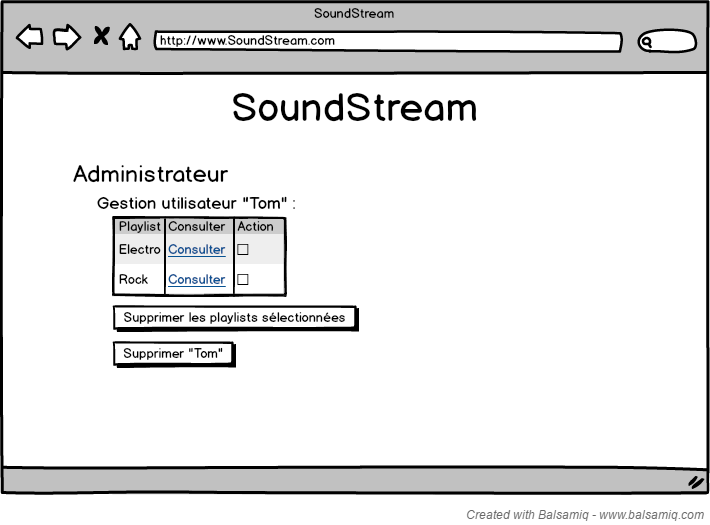


Figure 7 Gestion dédié utilisateur

Depuis la page d’administration l’administrateur peut cliquer sur « Ajouter un titre » pour arriver sur la page contenant le formulaire d’ajout de titre. Depuis cette page il lui suffit de remplir le formulaire et d’upload le fichier .mp3 correspondant.

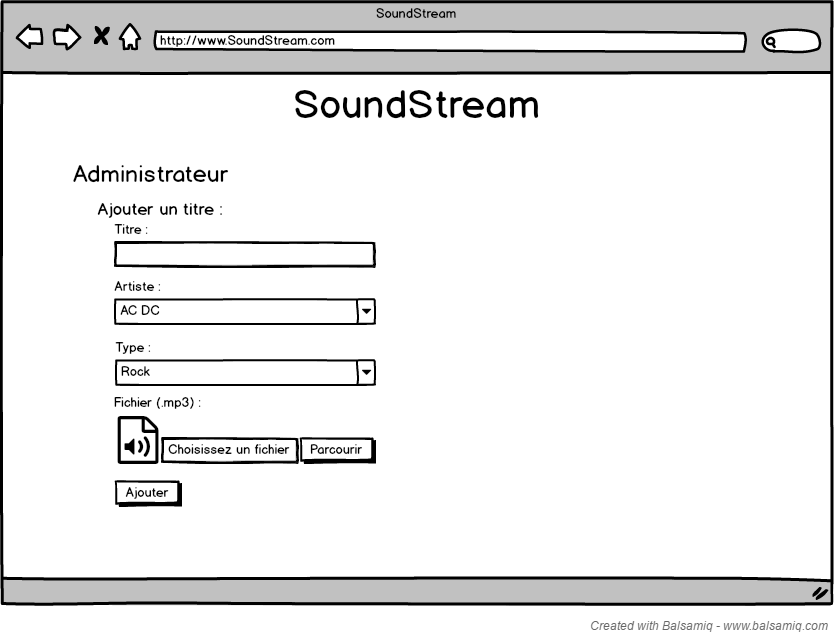


Figure 8 Fenêtre ajout de titre

### Détermination des points faibles

Pour ce projet j’ai choisi d’utiliser la librairie Bootstrap, je la connais déjà légèrement mais je devrai me rafraichir les idées.

## Stratégie de tests

Pour tester mon application, je vais réaliser des plans de tests ainsi que des rapports. Je donnerai aussi un plan à un de mes collègues pour qu’il le réalise vu qu’il n’a pas connaissance de mon application.

# Conception

## Détermination de l’arborescence du site et des rubriques

L’arborescence est séparée en trois parties :

1. La partie connexion/inscription qui est le départ du site
2. La partie utilisateur ou il pourra consulter ses playlists
3. Et la partie administration qui sert à gérer les morceaux ainsi que les utilisateurs

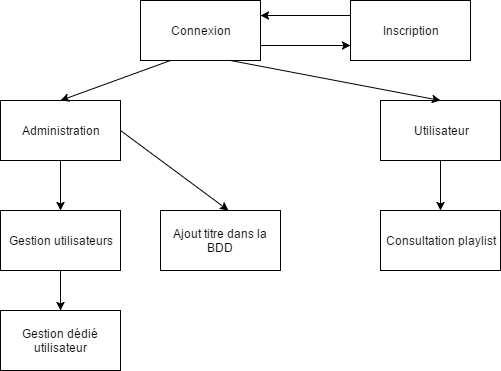


Figure 9 Arborescence du site

## Définition de la charte graphique

Concernant la charte graphique j’ai décidé de faire quelque chose de sobre tout comme l’application. De plus j’ai choisi d’utiliser la librairie Bootstrap car elle est très facile à utiliser.

## Maquette graphique

Voici les maquettes graphiques que j’ai réalisé en HTML dans le but de les rendre dynamiques avec du PHP.

Page de connexion :

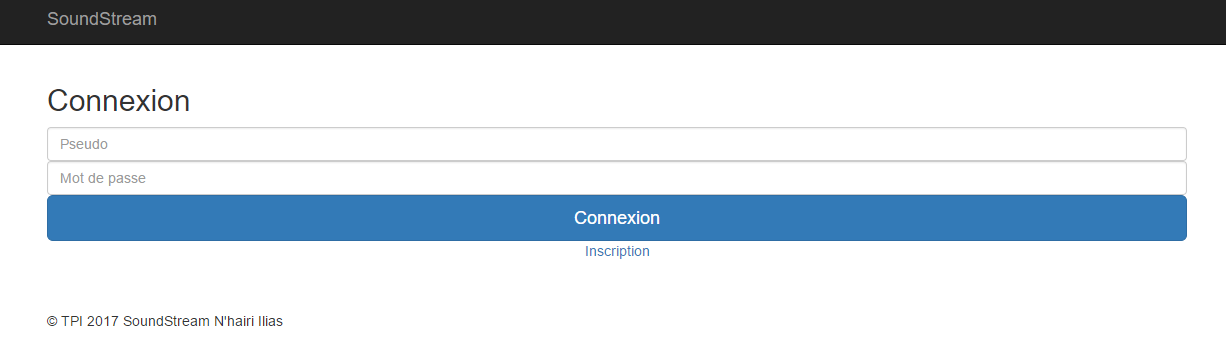


Figure 10 Page de connexion

Page d’inscription :

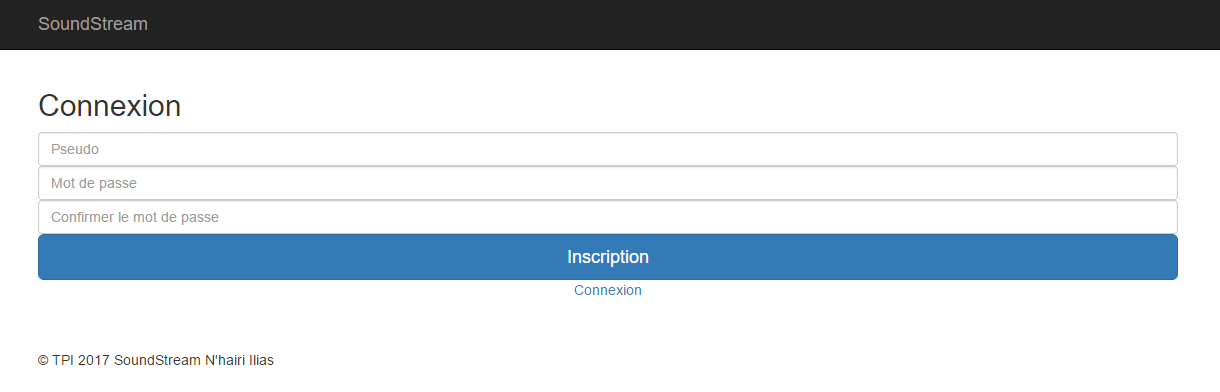


Figure 11 Page d'inscription

Fenêtre playlists de l’utilisateur :

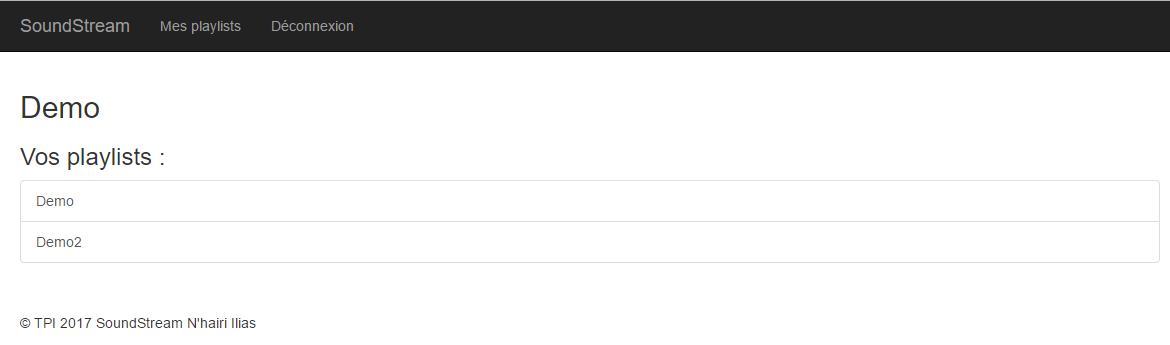


Figure 12 Playlists utilisateur

Fenêtre consultation d’une playlist :

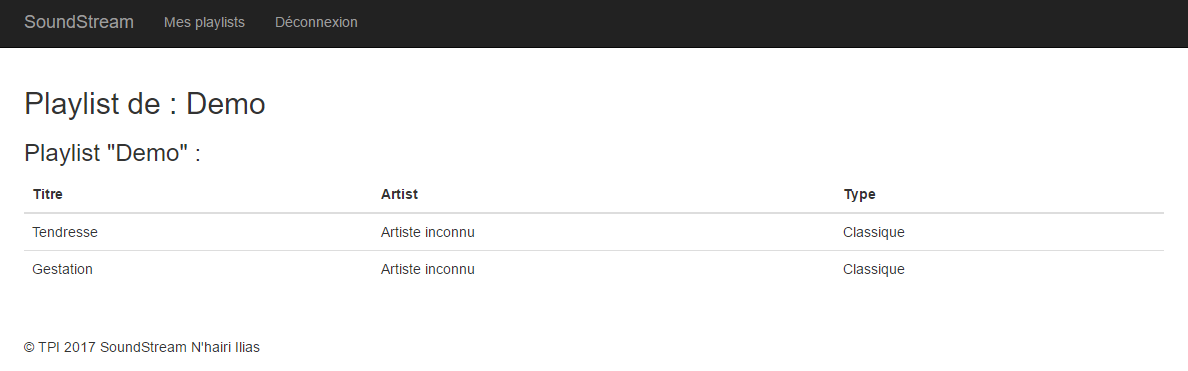


Figure 13 Playlist

Administration :

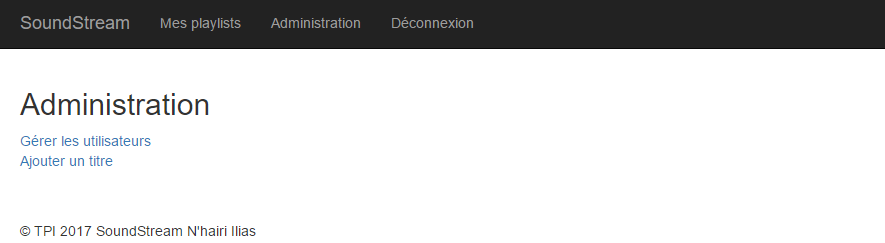


Figure 14 Administration

Fenêtre gestion des utilisateurs :

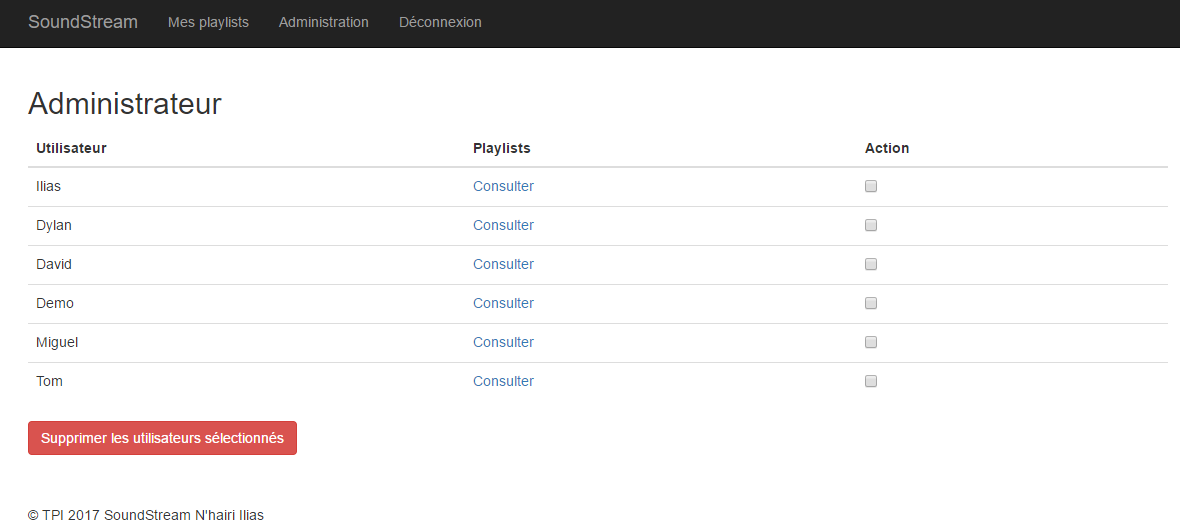


Figure 15 Gestion utilisateurs

Fenêtre gestion d’un utilisateur :

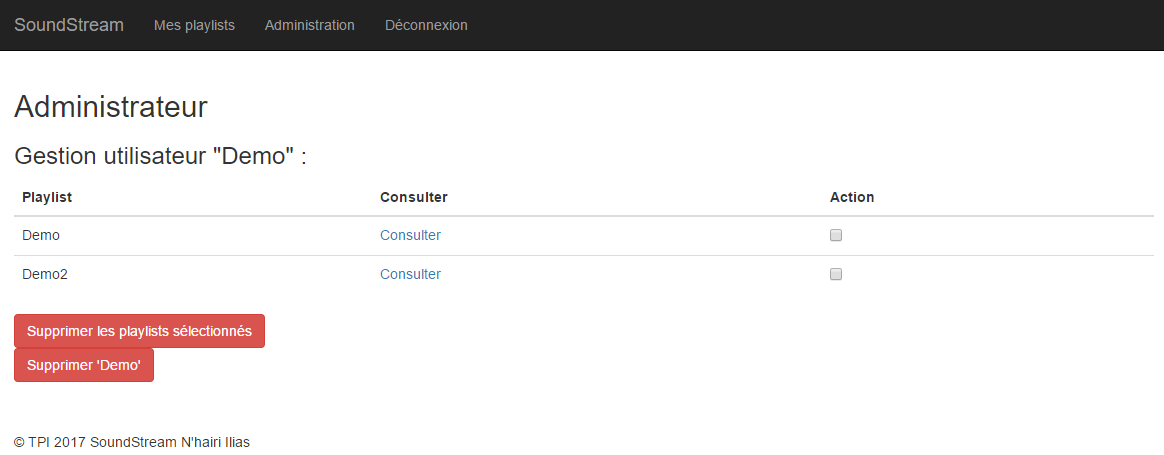


Figure 16 Gestion utilisateur

Fenêtre ajout musique :

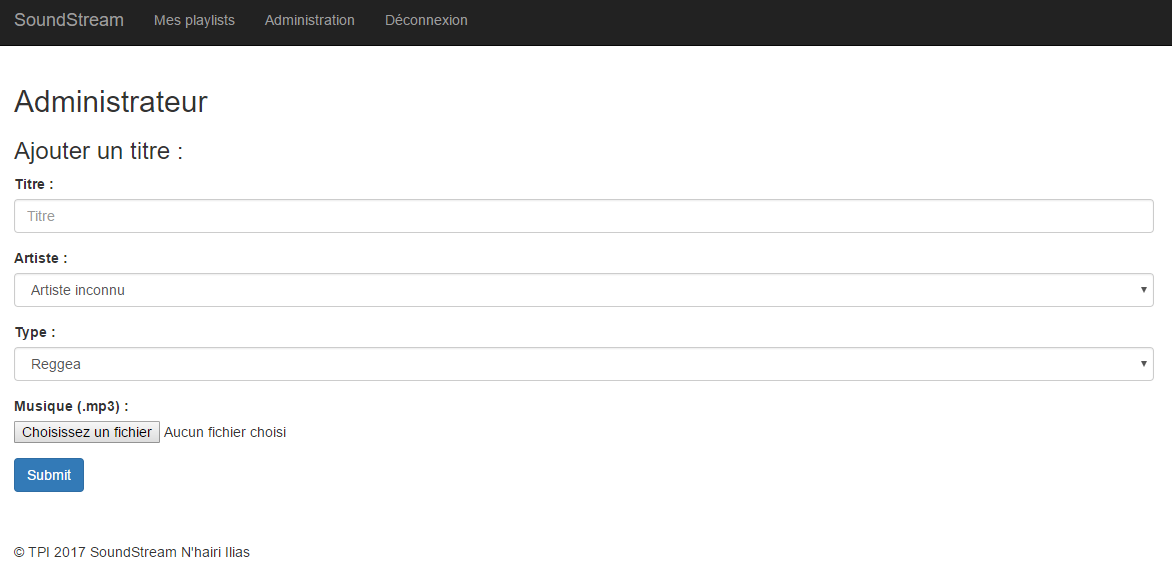


Figure 17 Ajout musique

## Interrogation

Voici les requêtes SQL que j’ai utilisé dans le site web avec une petite explication.

* SELECT \* FROM artists  
  Récupère les artistes dans le but de créer une liste déroulante.
* SELECT \* FROM types  
  Récupère les types dans le but de créer une liste déroulante.
* DELETE FROM playlists WHERE idPlaylist = :id  
  Cette requête me permet de supprimer un playlist et toutes les entrées correspondantes dans la table «Contain». Cependant pour que cela fonctionne en une requête il faut changer les contraintes en « ON DELETE CASCADE ».
* DELETE FROM users WHERE idUser = :id  
  Cette requête me permet de supprimer un playlist et toutes les entrées correspondantes dans la table « playlists » et «Contain». Cependant pour que cela fonctionne en une requête il faut changer les contraintes en « ON DELETE CASCADE ».
* SELECT \* FROM users WHERE pseudoUser = :pseudo AND passUser = :pass  
  Vérifie les informations de connexion fournies par l’utilisateur dans le but de la connexion.
* INSERT INTO users (pseudoUser,passUser,privilegesUser) VALUES (:pseudo,:pass,0)  
  Ajoute un utilisateur dans la base de données en fonctions des données fournies par lui-même.
* SELECT \* FROM users WHERE idUser = :id  
  Récupère les informations d’un utilisateur dans un but d’administration.
* SELECT \* FROM playlists WHERE idUser = :id  
  Récupère les playlists d’un utilisateur dans un but d’administration.
* SELECT \* FROM users WHERE privilegesUser != 1  
  Récupère tous les utilisateurs qui ne sont pas administrateur dans un but d’administration.
* INSERT INTO musics (titleMusic,idType,idArtist,fileName) VALUES (:title,:idType,:idArtist,:file)  
  Ajoute une musique dans la base de données.
* SELECT \* FROM playlists WHERE idPlaylist = :idPlaylist AND idUser = :idUser  
  Vérifie qu’une playlist appartient à un utilisateur. Cette requête est utilisée pour vérifier qu’un utilisateur non admin consulte une de ses playlists.
* SELECT namePlaylist,pseudoUser FROM playlists, users WHERE users.idUser = playlists.idUser AND playlists.idPlaylist = :idPlaylist  
  Récupère les information d’une playlist dans le but d’y afficher dans la page.
* SELECT \* FROM musics,playlists,types,contain,artists WHERE musics.idType = types.idType AND contain.idPlaylist = playlists.idPlaylist AND
* contain.idMusic = musics.idMusic AND playlists.idPlaylist = :idPlaylist AND artists.idArtist = musics.idArtist  
  Récupère toutes les musiques d’une playlist afin d’y afficher dans un tableau.
* SELECT \* FROM playlists WHERE idUser = :id  
  Récupère les playlists d’un utilisateur.

## Conception du code

Dans ce chapitre je vais détailler le code du site web. Tout est rédigé en HTML + PHP + SQL.

Pour se connecter l’utilisateur envoie ses informations puis elles sont vérifiés.  


Figure 18 Vérification connexion

Pour vérifier les informations j’exécute une requête avec celle-ci et si une seule ligne est retournée alors je stock les informations dans la session puis je redirige l’utilisateur en fonction de ses privilèges. La suite pour un utilisateur normal est simple j’affiche ses playlists dans une liste de lien.

Pour la page « playlist.php » c’est un peu plus compliqué. Premièrement je vérifie si la playlist appartient à l’utilisateur et que l’utilisateur en question n’est pas administrateur. Car les administrateurs peuvent consulter les playlists de tout le monde. De manière à ce que les administrateurs peuvent voir toutes les playlists et les utilisateurs que les leurs. Une fois cette vérification faite je récupère les informations ainsi que les musiques de la playlist.



Figure 19 Vérification pour la consultation des playlists

Pour la suppression d’utilisateurs ou de playlist c’est le même fonctionnement.



Figure 20 Suppression utilisateurs

Je récupère un tableau contenant les id des playlists ou utilisateurs à supprimer puis je les supprime à l’aide du requête SQL. J’ai eu des soucis pour faire ceci à cause des contraintes de clés étrangers. J’ai donc modifié les contraintes en « ON DELETE CASCADE » de manière à supprimer toutes les clés étrangères liées à la première.

Concernant l’ajout de titre je récupère les informations sur le nouveau morceau. Une fois cela fait je vérifie que le fichier est un .mp3 et lui génère un nom unique. Pour terminer j’ajoute tout dans la base de données après avoir sauvegardé le fichier dans le dossier « Music ». J’ai aussi eu besoin de modifier quelques lignes dans le fichier « php.ini ». Ces lignes permettent de modifier la taille maximum d’envoi de fichier par la méthode « Post ».

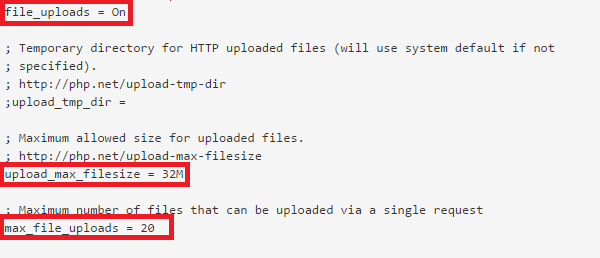


Figure 21 Php.ini 1

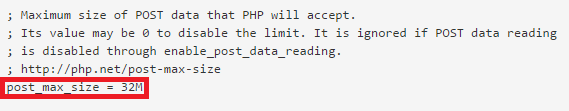


Figure 22 Php.ini 2



Figure 23 Ajout musique

Mon fichier « functions.php » contient quelques fonctions utiles. La première « Database » sert à se connecter à la base de données et à créer une instance PDO.



Figure 24 Connexion BDD

Une fonction qui m’a été très utile durant mon travail est la fonction « Debug » que j’ai créée dans le but de pouvoir consulter le contenu de mes variables.



Figure 25 Debug

C’est une petite fonction mais elle est très utile.

J’ai aussi écrit deux fonctions qui ont pour but de créer des listes déroulantes (pour les artistes et types).



Figure 26 Création listes déroulantes

## Ressources

Pour ce projet je n’ai pas eu besoin de beaucoup de ressources externe hormis la librairie Bootstrap.

<http://getbootstrap.com/>

J’ai aussi trouvé le bout de code servant à créer un nom unique sur un forum à cette adresse :

<http://stackoverflow.com/questions/36030621/create-a-unique-name-for-the-uploaded-image>

# Réalisation

## Dossier de réalisation

* http
  + css
    - Contient les fichiers .css de Bootstrap
  + Fonts
    - Contient les fichiers de police de Bootstrap
  + Js
    - Contient les fichiers javascript de Bootstrap
  + Music
    - Contient toutes les musiques que l’application utilisera
  + Php
    - Functions.php
      * Contient mes fonctions
  + addTitle.php
    - Contient le formulaire pour l’ajout de titres
  + Admin.php
    - Contient les liens des pages d’ajout de titre ou de gestion d’utilisateurs
  + Deconnexion.php
    - Contient quelques lignes servant à déconnecter l’utilisateur
  + delPlaylists.php
    - Contient le code servant à supprimer les playlists
  + delUsers.php
    - Contient le code servant à supprimer les utilisateurs
  + Index.php
    - Contient la page de connexion et un bout de code d’authentification
  + Inscription.php
    - Contient le formulaire d’inscription ainsi que le code d’ajout à la base de données
  + manageUser.php
    - Contient les formulaires de gestion de l’utilisateur et de ses playlists
  + manageUsers.php
    - Contient le formulaire de gestion des utilisateurs.
  + newMusic.php
    - Contient le code d’ajout de morceau à la base ainsi que la sauvegarde du fichier
  + Playlist.php
    - Contient l’affichage des playlists en fonction de l’id
  + Template.php
    - C’est la maquette que j’ai réalisée en premier pour me baser dessus pour chaque page.
  + User.php
    - La page affichant les playlists d’un utilisateur en fonction de son id dans la session.

Librairies externes :

* Bootstrap

## Description des tests

Pour tester mon site je vais réaliser un plan de test, les résultats seront dans le rapport de test. Le plan de tests doit être réalisé avec la base de donnée version 3 (tpi\_v3.sql).

Plan de tests :

|  |  |  |  |
| --- | --- | --- | --- |
| Numéro test | Auteur | Scénario | Résultat attendu |
| 1 | Ilias N'hairi | Se rendre sur la page index.php puis cliquer sur inscription | La page d'inscription doit s'ouvrir |
| 2 | Ilias N'hairi | Créer le compte test avec le mot de passe "Super" qu'il faudrait confirmer | La page index.php s'affiche avec un message de confirmation de création de compte |
| 3 | Ilias N'hairi | Se connecter sur la page index.php avec le compte "Demo" avec le mot de passe "Super" | L'utilisateur doit être redirigé sur la page contenant ses playlists (il doit y en avoir 2) |
| 4 | Ilias N'hairi | L'utilisateur clique sur la première playlist "Demo" | Une page avec deux musiques s'affiche |
| 5 | Ilias N'hairi | L'utilisateur change le paramètre "id" dans le lien en "1" (playlist d'un autre utilisateur) | L'utilisateur n'a pas accès il est redirigé |
| 6 | Ilias N'hairi | L'utilisateur clique sur "Déconnexion" | Il est redirigé sur la page de connexion |
| 7 | Ilias N'hairi | Se connecter avec le compte "Admin" avec le mot de passe "Super" | L'utilisateur est redirigé sur la page admin.php |
| 8 | Ilias N'hairi | L'utilisateur clique sur "Gérer les utilisateurs" | La liste des utilisateurs s'affiche |
| 9 | Ilias N'hairi | Cliquer sur "Consulter" à côté de l'utilisateur "Demo" | Les 2 playlists de l'utilisateur s'affichent |
| 10 | Ilias N'hairi | Cliquer sur "consulter" à côté de la playlist "demo" | Une liste avec deux musiques doit s'afficher |
| 11 | Ilias N'hairi | Cliquer sur "back" du navigateur. Sélectionner les deux playlists et cliquer sur "Supprimer les playlists sélectionnées". Après la redirection cliquer sur "Consulter" à côté de l'utilisateur démo | Il ne doit plus y avoir de playlists |
| 12 | Ilias N'hairi | Cliquer sur "Supprimer 'Demo'" | Après la redirection sur la page de gestion des utilisateurs l'utilisateur "Demo" ne doit plus y être. |
| 13 | Ilias N'hairi | Sélectionner l'utilisateur "Ilias" puis cliquer sur "Supprimer les utilisateurs sélectionnés" | L'utilisateur "Ilias" ne doit plus se trouver dans la liste |
| 14 | Ilias N'hairi | Cliquer sur "Administration" puis sur "Ajouter un titre" remplissez les champs avec les informations d'une musique et fournissez le fichier .mp3 | La page administration doit s'afficher sans erreur et la musique doit être ajoutée dans la base de données. Le fichier doit avoir été sauvé dans le dossier "Music" avec un nom unique. |
| 15 | Ilias N'hairi | L'administrateur clique sur "Déconnexion" | L'utilisateur doit être redirigé sur la page de connexion |
| 16 | Ilias N'hairi | Se connecter avec le compte "test" avec le mot de passe super puis essayer d'accéder aux pages : admin.php manageUsers.php ManageUser.php addTitle.php | L'utilisateur doit toujours être redirigé vers la page user.php |

Rapport de tests par N’hairi Ilias le 19.05.2017 :

|  |  |  |  |
| --- | --- | --- | --- |
| Numéro de test | Auteur | Résultat | Fiche anomalie |
| 1 | N'hairi Ilias | OK | / |
| 2 | N'hairi Ilias | OK | / |
| 3 | N'hairi Ilias | OK | / |
| 4 | N'hairi Ilias | OK | / |
| 5 | N'hairi Ilias | OK | / |
| 6 | N'hairi Ilias | OK | / |
| 7 | N'hairi Ilias | OK | / |
| 8 | N'hairi Ilias | OK | / |
| 9 | N'hairi Ilias | OK | / |
| 10 | N'hairi Ilias | OK | / |
| 11 | N'hairi Ilias | OK | / |
| 12 | N'hairi Ilias | OK | / |
| 13 | N'hairi Ilias | OK | / |
| 14 | N'hairi Ilias | OK | / |
| 15 | N'hairi Ilias | OK | / |
| 15 | N'hairi Ilias | OK | / |

## Erreurs restantes

Je n’ai pas trouvé d’erreurs.

# Conclusion

Pour conclure je vais résumer le projet en quelques lignes. L’application « SoundStream » a été réalisée complétement en suivant le cahier des charges. Le site web a aussi été réalisé. Tous les objectifs ont été atteint.

Points négatifs :

Selon moi l’interface pourrai être beaucoup mieux, cependant nous n’avons pas reçu la formation nécessaire à cela.

La recherche n’est pas optimisée car si l’utilisateur ne connait pas le titre d’une musique il n’y a pas d’option pour découvrir des morceaux.

Difficultés :

Je n’ai pas trouvé ce projet très difficile. Je codais sans vraiment avoir de problèmes. Le projet avançait de manière régulière et continue.

Suite du projet :

J’ai beaucoup d’idées de chose à améliorer concernant ce projet en voici une liste :

* L’ajout d’artistes et de types depuis le site web. Cette option n’a pas été implémentée car je me suis concentré sur les objectifs principaux du cahier des charges.
* Une page servant à modifier les informations d’un compte (Mot de passe, nom d’utilisateur)
* Ajouter une fonction qui fait que quand une musique se termine la suivante se lance.
* Une interface plus ergonomique.
* Une fonction de découverte de musiques.
* L’amélioration la plus utile serai de faire migrer la base de données ainsi que le serveur web sur un serveur distant, afin que tout le monde puisse utiliser l’application via internet.
* Pouvoir écouter les musiques depuis le site web.

J’ai beaucoup apprécié travailler sur ce projet et j’aimerai beaucoup le continuer par la suite.

# Annexes

## Sources – bibliographie

Liens utiles :

* <http://getbootstrap.com/>
* <http://php.net>
* <https://dev.mysql.com>
* <https://docs.microsoft.com/fr-fr/dotnet/articles/csharp/language-reference/index>
* <https://www.developpez.net/>
* <https://coderwall.com>
* [www.google.ch](http://www.google.ch)
* <https://github.com>

Remerciement Dylan Schito pour les tests.

## Procédure d’installation

Installation de la base de données :

Tout d’abord il vous faudra télécharger « EasyPHP Devserver » à l’adresse suivante :

<http://www.easyphp.org/save-easyphp-devserver-latest.php>

Suivez les étapes d’installations.

Une fois l’installation terminée lancez EasyPHP en mode administrateur. Puis démarrez les deux serveurs à cette adresse : <http://127.0.0.1:1111/>

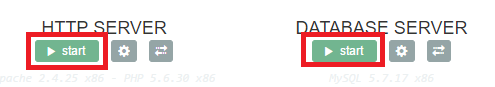


Figure 27 Installation serveur

Après cela ouvrez phpMyAdmin.

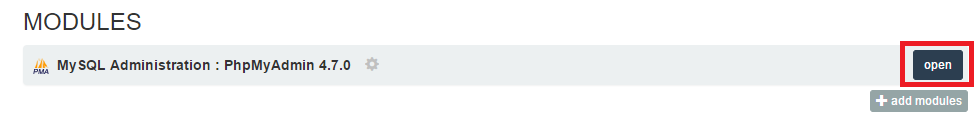


Figure 28 Activation des serveurs

Cliquez sur « Importer ».

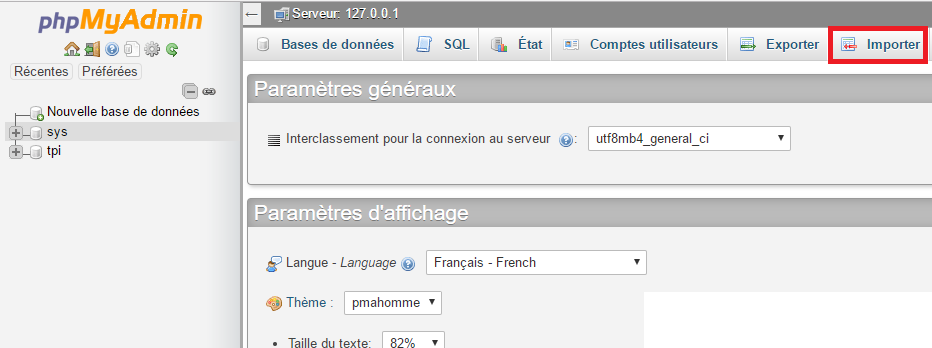


Figure 29 Importation BDD

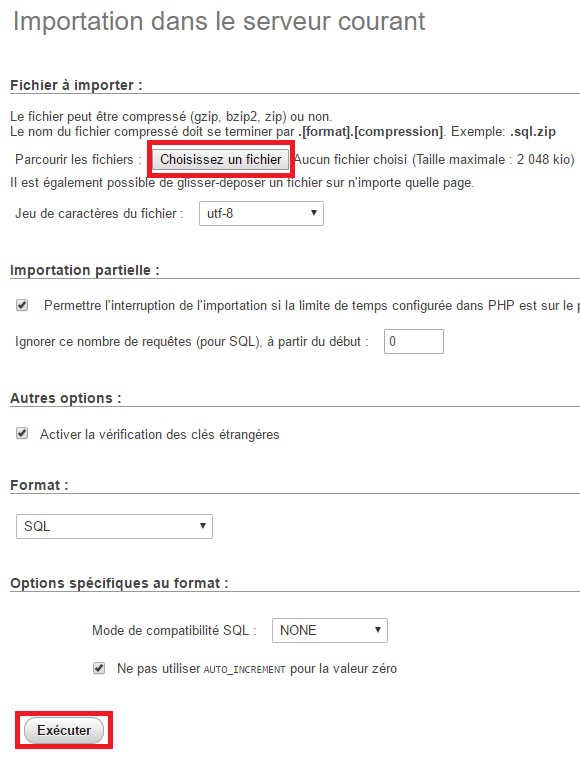
Choisissez le fichier install.sql dans le dossier « Install » puis cliquez sur exécuter.  


Figure 30 Importation Bdd suite

Il vous suffit maintenant d’exécuter le fichier « SoundStream.exe » qui se trouve dans le dossier « App » dans le dossier « Install ».

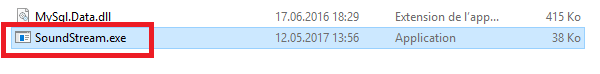
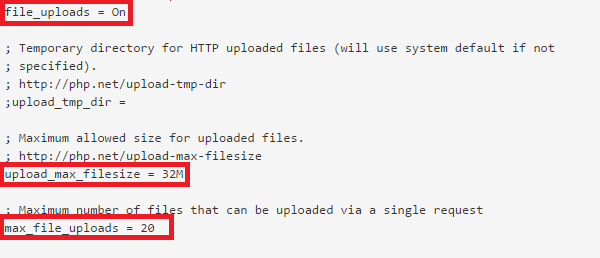


Figure 31 Lancement application

Puis copiez le dossier « TPI » du dossier « Install » dans le répertoire de votre serveur web. Ne pas oublier de modifier le fichier « php.ini », sinon l’upload de fichiers de fonctionnera pas. Modifier les ligne mise en évidences de la même manière que sur la figure suivante.



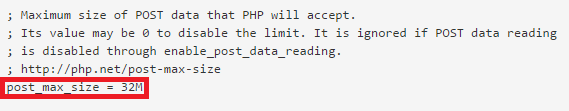


Figure 32 Configuration "php.ini"

# Table des figures

[Figure 1 Fenêtre connexion 2](#_Toc483309982)

[Figure 2 Fenêtre inscription 2](#_Toc483309983)

[Figure 3 Fenêtre liste playlists 3](#_Toc483309984)

[Figure 4 Fenêtre playlist 3](#_Toc483309985)

[Figure 5 Fenêtre administration 4](#_Toc483309986)

[Figure 6 Fenêtre gestion utilisateurs 4](#_Toc483309987)

[Figure 7 Gestion dédié utilisateur 5](#_Toc483309988)

[Figure 8 Fenêtre ajout de titre 5](#_Toc483309989)

[Figure 9 Arborescence du site 7](#_Toc483309990)

[Figure 10 Page de connexion 7](#_Toc483309991)

[Figure 11 Page d'inscription 8](#_Toc483309992)

[Figure 12 Playlists utilisateur 8](#_Toc483309993)

[Figure 13 Playlist 8](#_Toc483309994)

[Figure 14 Administration 9](#_Toc483309995)

[Figure 15 Gestion utilisateurs 9](#_Toc483309996)

[Figure 16 Gestion utilisateur 9](#_Toc483309997)

[Figure 17 Ajout musique 10](#_Toc483309998)

[Figure 18 Vérification connexion 12](#_Toc483309999)

[Figure 19 Vérification pour la consultation des playlists 15](#_Toc483310000)

[Figure 20 Suppression utilisateurs 15](#_Toc483310001)

[Figure 21 Php.ini 1 16](#_Toc483310002)

[Figure 22 Php.ini 2 16](#_Toc483310003)

[Figure 23 Ajout musique 17](#_Toc483310004)

[Figure 24 Connexion BDD 18](#_Toc483310005)

[Figure 25 Debug 18](#_Toc483310006)

[Figure 26 Création listes déroulantes 19](#_Toc483310007)

[Figure 27 Installation serveur 25](#_Toc483310008)

[Figure 28 Activation des serveurs 25](#_Toc483310009)

[Figure 29 Importation BDD 25](#_Toc483310010)

[Figure 30 Importation Bdd suite 26](#_Toc483310011)

[Figure 31 Lancement application 27](#_Toc483310012)

[Figure 32 Configuration "php.ini" 27](#_Toc483310013)