

**Rapport du projet de LTW2**

***Share The Sound***



**DAITA Jordan** – M1 WIC

**DROUET Rémy** – M1 WIC

**LI Ling** – M2 DCISS RS

LTW2 – Enseignants : ATENCIA Manuel & ESPINOSA Javier

Année universitaire : 2014-2015

**Table des matières**

[**Introduction** 3](#_Toc417058949)

[**I.** **Conception de *Share the Sound*** 4](#_Toc417058950)

[**I.1. Description du *player*** 4](#_Toc417058951)

[**I.2. Architecture 3-tiers** 4](#_Toc417058952)

[**I.3. Synchronisation des chansons** 4](#_Toc417058953)

[**II.** **Réalisation du prototype** 4](#_Toc417058954)

[**II.1. Technologies utilisées** 4](#_Toc417058955)

[**II.2. Principes ergonomiques** 4](#_Toc417058956)

[**III.** **Répartition du travail** 4](#_Toc417058957)

[**IV.** **Révolutions** 4](#_Toc417058958)

[**Conclusion** 5](#_Toc417058959)

[**Annexe** 6](#_Toc417058960)

# **Introduction**

Dans le cadre du cours LTW2, il nous a été demandé de réaliser un projet de groupe au sujet d’un *music player* collaboratif. C’est ainsi que nous nous sommes lancés dans cette aventure de la conception et le développement de l’application web *Share the Sound*.

Afin de mener bien ce travail, nous nous sommes basés sur la description du projet fourni à son démarrage. Notre conception est issue de l’architecture 3-tiers, y compris le client, le serveur et la base de données. L’application doit être capable de fonctionner comme une radio. C’est-à-dire que la musique se lance automatiquement en fonction du nombre de votes pour chaque chanson (celle qui a le plus de votes sera la chanson suivante). En plus du système de votes, deux interfaces sont attendues : *playlist* et *catalogue*. C’est en nous basant de cela que nous avons pu émettre notre hypothèse et réaliser notre solution.

Nous allons présenter dans ce rapport les étapes de conception et de réalisation de notre travail pour aboutir à notre résultat. Pour se faire, nous allons présenter le fonctionnement de notre *music player*, l’architecture de notre site web, comment nous avons émis l’hypothèse sur la synchronisation des chansons, avant de passer sur les aspects techniques et ergonomiques concernant la réalisation de notre site web.

# **Conception de *Share the Sound***

## **I.1. Description du *player***

(…)

**Système de votes**

## **I.2. Architecture 3-tiers**

(…)

Diagramme de classes

## **I.3. Synchronisation des chansons**

(…)

# **Réalisation du prototype**

## **II.1. Technologies utilisées**

Afin de rendre notre application de *music player* dynamique, interactive et collaborative, nous nous sommes servis des technologies et des langages web suivantes :

***Html5*, *css*, *bootstrap*** :

Les deux interfaces, « playlist » et « catalogue », ont été conçus grâce aux langages *html* et *css*, en plus du framework css *Bootstrap*. En effet, nous disposons d’une page d’index avec les codes scripts qui dirige les deux interfaces « playlist » et « catalogue ». L’utilisation du *Bootstrap* nous a permis de normaliser la présentation du site avec un ensemble homogène de styles. Il a également facilité le travail de la mise en forme des pages. C’est grâce à ce framework que nous avons mis en place des boutons esthétiques, la barre de navigation, etc.

***JavaScript, Angular JS, JQuery*** **et *Ajax***

L’utilisation du *JavaScript*, langage de programmation de scripts orienté objet, est primordiale dans notre projet. Ce langage nous a permis d’écrire du code source en créant des fonctions et des évènements côté client afin de dynamiser nos pages *html*. Nous avons également utilisé *jQuery* et *Ajax* qui sert à faciliter l’écriture des scripts.

Nous avons particulièrement employé le framework *JavaScript* *AngularJS*, *introduit dans le cours,* pour organiser efficacement les pages. Il nous a permis de construire une architecture MVC, et de manipuler le DOM au moyen de directives.

***Node.js* et *nano*** :

(…)

***jSon* et *CouchDB***:

Une base de données est nécessaire dans notre projet pour stocker les musiques (*albums, tracks, proposed Tracks*). Nous avons choisi *CouchDB* qui a été introduit dans le cours. Il s’agit d’une nouvelle engeance de système de gestion de base de données. Nous voulions profiter de cette occasion pour découvrir cette base NoSQL. Pour se faire, nous avons organisé une collection des données sous forme Json.

***Deezer API****:*

Tout au long du projet, nous nous sommes souvent rendus sur le site *Deezer developers* pour exploiter la présentation des données de musiques sur Deezer dans les onglets API et SDK. Etant donné que les musiques proposés dans le catalogue de notre application viennent de Deezer, nous devions pouvoir créer des fonctions pour récupérer et ajouter des chansons, mettre à jour la playlist, etc. Nous avons ainsi pu créer des requêtes d’actions *get/put/post/delete*.

## **II.2. Principes ergonomiques**

(…)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Police** | **Taille** | **Couleur** |
|  |  |  |  |
|  |  |  |  |

La réalisation de notre application a été effectuée sous les navigateurs *FireFox* et *Chrome*.

# **Répartition du travail**

(...)

# **Révolutions**

Version mobile ??

(...)

# **Conclusion**

(…)

**Annexe** : Code source