

## PROJET WEB : REALISATION

### SITE DE MUSIQUE

L3 – M.I.A.G.E. – Groupe n°8

Maxime CHABOISSIER  
Floris FOURNIER  
Antoine NOSAL  
Sofiane ZEGHADI

### Introduction

*Dans ce dossier, il s'agira de présenter sous différentes formes les choix qui ont été fait au cours de la réalisation d'un site de musique. Ainsi seront abordés :*

1. *La présentation du sujet*
2. *L'analyse des différents choix*
3. *Le guide utilisateur*
4. *Les limites de l'application*

*Vous trouverez en annexe la répartition des tâches finales, différentes de ce qui avait été convenu au cours de la phase de conception.*

*Nous avons décidé de donner un nom à notre application WEB : SeaDiLi.*



## 1. Présentation du sujet

Pour rappel, le but du projet est de développer un site de musique dédié à la création de *playlists* et à leur écoute. Le site devra être composé de deux éléments :

- ⇒ une page dans laquelle l'utilisateur pourra rechercher des titres et les inclure dans une *playlist* ;
- ⇒ une page où l'utilisateur pourra écouter des *playlists*.

Le site devra également permettre de consulter des informations sur les artistes associés aux titres pour aider les utilisateurs dans leur recherche et de sauvegarder des *playlists*.

Une base de données est fournie, son schéma est le suivant :

users	tracks	artists	playlists	playlists_tracks
- user_id - username - password - email	- artist_id - track_id - title - mp3_url	- artist_id - name - image_url - info	- user_id - playlist_id	- playlist_id - position - track_id

Ont été défini dans le cahier des charges du projet 3 niveaux de priorité. Vous trouverez dans le tableau suivant ces 3 niveaux et des commentaires sur leur réalisation.

Niveau	Cahier des charges	Réalisation
Niveau 1	Le site permet de <b>rechercher</b> des titres et de <b>consulter</b> les informations sur les artistes	Sur la page d'accueil de SeaDiLi, un utilisateur peut, via une barre de recherche, rechercher des titres et des artistes. Les informations sur les artistes sont disponibles via un simple clic.
Niveau 2	Le site permet de <b>construire</b> des <i>playlists</i> et de les <b>écouter</b>	Tous les titres affichés sur SeaDiLi peuvent être ajoutés à une <i>playlist</i> (dite « en création »). Une fois une <i>playlist</i> créée, un utilisateur enregistré et connecté peut y accéder et écouter tous les titres qui s'y trouvent.
Niveau 3	Le site permet de <b>créer des comptes</b> utilisateur et de <b>sauvegarder</b> les <i>playlists</i>	Une page dédiée permet à un utilisateur de rejoindre SeaDiLi en créant un compte. Grâce à ce compte, l'utilisateur pourra se connecter et créer des <i>playlists</i> qui lui seront associés.

## 2. Analyse des choix

Au niveau technique, il a été décidé d'utiliser deux patrons de conceptions pour construire l'application web demandé, le patron *MVC* et le patron *Active Record*.

Le patron *MVC* (*Modèle Vue Contrôleur*) suppose une séparation entre les données (le modèle) l'affichage (la vue). Cette séparation suppose quant à elle la présence d'autres classes qui définissent les différentes méthodes à exécuter en fonction des actions de l'utilisateur (le contrôleur).

Pour représenter les données, nous avons décidé d'utiliser le patron *Active Record*. Ce dernier implique que pour chaque table de la base de données, est créée une classe (*php*) correspondante. Ainsi, chaque tuple de la base de données pourra être chargé dans un objet de la classe qui correspond à la table dans laquelle il est contenu.

Afin de faciliter et d'optimiser la connexion à la base de données, nous avons décidé de créer une classe *DataBase* qui nous permet de récupérer facilement un objet représentant une connexion à la base. L'extension *PDO (PHP Data Objects)* est l'interface qui a été choisi pour accéder à la base. Elle a notamment l'avantage de pouvoir être utilisée indépendamment du *SGBD*. Un fichier de configuration (*seadili-db.ini*) permet de très facilement configurer les paramètres de la base.

Dans le tableau suivant, vous trouverez une présentation de l'architecture et des commentaires sur la réalisation, associée à chacune des parties.

Nom du dossier	Commentaires
<b>0_model</b>	<ul style="list-style-type: none"><li>→ « <i>M</i> » de « <i>MVC</i> ».</li><li>→ Contient l'ensemble des classes d'<i>Active Record</i> qui permettent de manipuler facilement les données.</li><li>→ Contient l'objet <i>DataBase</i> et son fichier de configuration.</li></ul>
<b>1_view</b>	<ul style="list-style-type: none"><li>→ « <i>V</i> » de « <i>MVC</i> »</li><li>→ Contient une classe qui intègre toutes les méthodes d'affichage</li></ul>
<b>2_controller</b>	<ul style="list-style-type: none"><li>→ « <i>C</i> » de « <i>MVC</i> »</li><li>→ Contient une classe abstraite <i>Controller</i> de laquelle hériteront tous les contrôleurs. Elle facilite leur utilisateur et appelant la bonne méthode en fonction du contenu associé à « <i>a</i> » (comme « action ») dans le tableau <code>\$_GET</code></li><li>→ Contient la classe « <i>SiteController</i> » qui se charge d'appeler la bonne méthode la vue en fonction des actions de l'utilisateur</li><li>→ Contient les fichiers <i>php</i> appelés lors de la connexion, de la déconnexion, ainsi que de la création d'une <i>playlist</i>.</li></ul>
<b>3_content</b>	<ul style="list-style-type: none"><li>→ Contient l'en-tête (<i>header</i>), le pied de page (<i>footer</i>), ainsi que divers fichiers <i>html</i> et <i>php</i> de contenu.</li></ul>
<b>4_style</b>	<ul style="list-style-type: none"><li>→ Contient l'ensemble des feuilles de style</li></ul>
<b>5_images</b>	<ul style="list-style-type: none"><li>→ Contient toutes les images utilisées sur le site</li></ul>
<b>js</b>	<ul style="list-style-type: none"><li>→ Contient la librairie <i>audioplayer</i> utilisée pour l'affichage du lecteur audio <i>html5</i></li><li>→ Contient la librairie <i>easy-modal</i> utilisée pour afficher les informations des artistes</li></ul>

Pour plus d'informations sur *audioplayer*, nous vous invitons à visiter ce site :

<http://osvaldas.info/audio-player-responsive-and-touch-friendly>

Pour plus d'informations sur *easy-modal*, nous vous invitons à visiter ce site :

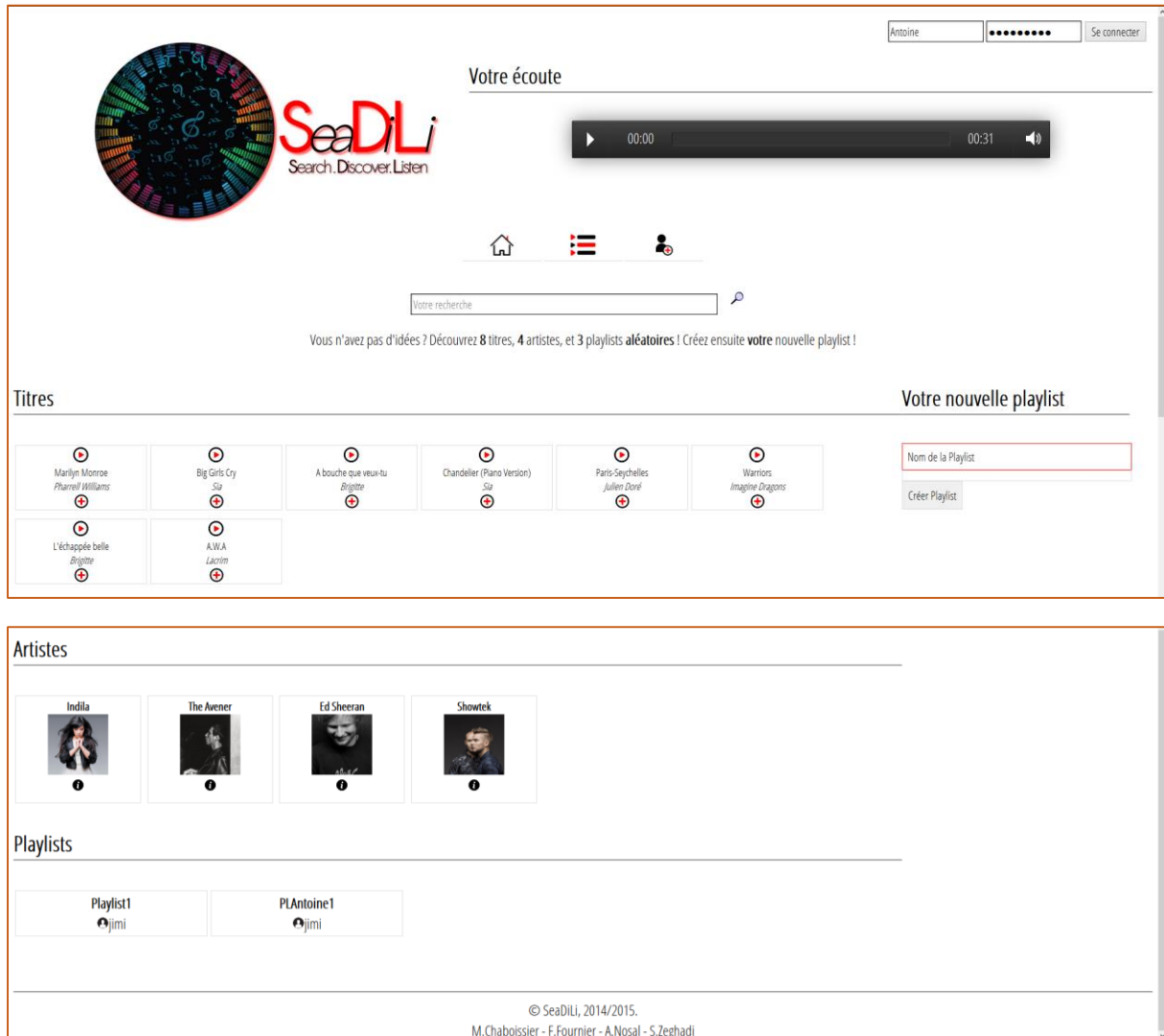
<http://flaviusmatis.github.io/easyModal.js/>

### 3. Guide utilisateur

Afin de construire au mieux le guide utilisateur de notre application, nous avons décidé de nous baser sur des copies d'écran.

#### Page d'accueil

Vous trouverez ci-dessous la page d'accueil du site (en deux parties) :




Une fois sur cette page, l'utilisateur découvre alors 8 titres aléatoires, 4 artistes aléatoires et 2 *playlists* aléatoires.

Concernant les titres et les artistes, il peut alors faire plusieurs actions :

- lire un titre en cliquant sur 
- ajoute un titre à la *playlist* en création en cliquant sur 
- découvrir les informations d'un artiste en cliquant sur 

Concernant la recherche, il pourra :

→ recherche dans tous les titres et tous les artistes de la base en cliquant sur 

Concernant les *playlists*, il pourra :

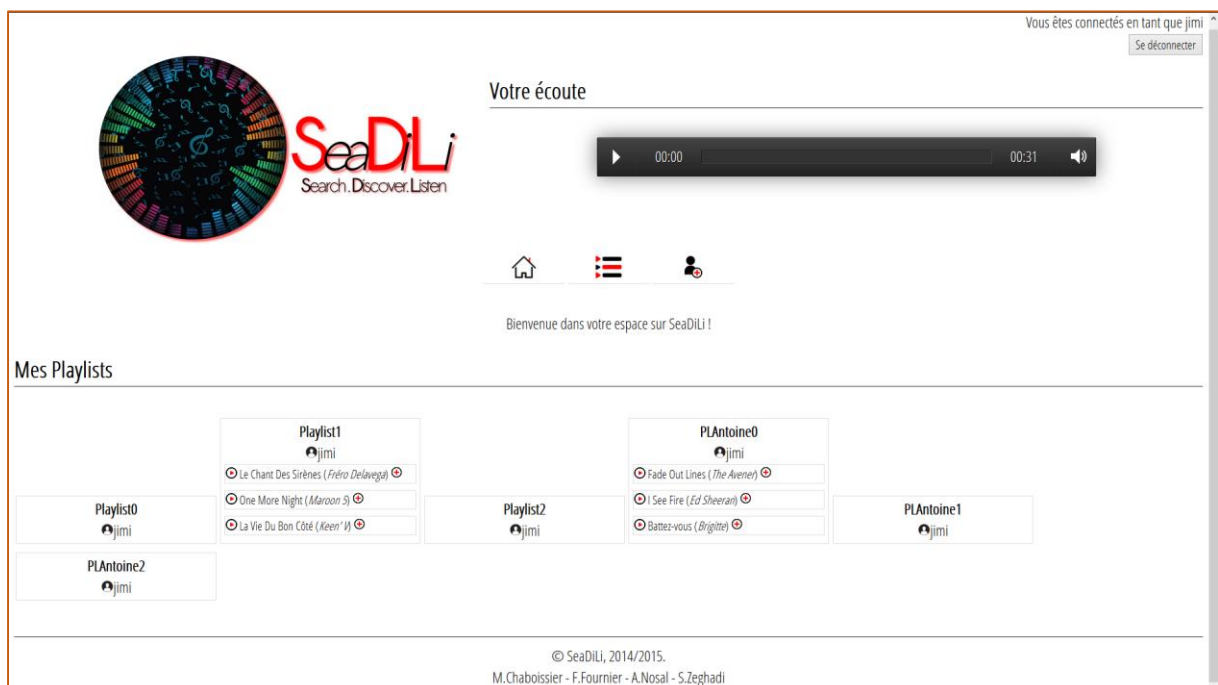
→ afficher la *playlist* en cliquant sur la partie contenant son nom :



Concernant la *playlist* en création, une fois tous les titres souhaités ajoutés, l'utilisateur peut la créer en cliquant sur « *Créer Playlist* ». Deux cas de figure se présentent alors :

- Si l'utilisateur est connecté, la *playlist* lui sera associé, et il la retrouvera en se connectant et en se rendant dans la page « *Playlists* ».
- Si l'utilisateur n'est pas connecté, la *playlist* sera associé à un utilisateur dans la base qui correspond au « compte commun ». La *playlist* n'est associée à personne et « appartient » donc à tout le monde. Cela permet de partager des *playlists* tout en gardant le modèle de la base de données imposé. Cette fonctionnalité n'a que très peu été approfondie lors de la réalisation.

### Page Playlists



Une fois connecté, l'utilisateur peut accéder aux *playlists* qu'il a créées via cette page. Il pourra ainsi écouter chacun des morceaux qu'il a ajoutés et profiter de l'application.

## Page d'enregistrement



Sur cette page, l'utilisateur a la possibilité de se créer un compte utilisateur. Il devra respecter certaines conditions pour que son compte soit effectivement créé. Si ce n'est pas le cas, des messages s'afficheront afin qu'il puisse correctement remplir le formulaire.

## 4. Limites de l'application

L'application qui a été développée tente de répondre au mieux au cahier des charges fourni (sujet du projet). Cependant, on peut lui trouver des limites.

- Le lecteur audio unique ne devrait pas être rechargé lors du changement de page.
- L'utilisateur devrait pouvoir manipuler des *playlists* (les modifier).
- L'utilisateur devrait pouvoir choisir de partager ou non une *playlist*.
- L'utilisateur devrait pouvoir modifier les paramètres de son compte.
- L'utilisateur devrait pouvoir rechercher dans la base des *playlists*.
- L'utilisateur devrait pouvoir partager ses créations sur les réseaux sociaux.

Ces nouvelles fonctionnalités auraient bien entendu pour but d'améliorer l'expérience utilisateur.

L'application SeaDiLi, en l'état, reste néanmoins utilisable.

## Conclusion

*Ce projet de programmation WEB nous a permis de mettre en pratique les connaissances acquises lors des cours et des TDs mais aussi d'utiliser des connaissances précédemment acquises. A travers un sujet très intéressant et qui nous parle, nous avons pu créer une application complète, répondant à un cahier des charges précis et prête à être utilisée.*

*Le projet a été développé grâce à l'outil Git et est hébergé sur GitHub. Voici le lien du dépôt :*

<https://github.com/Tixinoo/Seadili>

## Annexe 1 : Répartition finale des tâches

Vous trouverez ci-après la répartition finale des tâches :

Tâche	Description détaillée (Résultat attendu)	Développeur(s)
<b>Créer l'en-tête de la page (HTML)</b>	Il s'agit de créer l' <i>header</i> du site. Il devra contenir une barre de connexion, un titre/logo, et les liens vers les trois pages du site ( <i>accueil, playlists, s'enregistrer</i> )	Maxime CHABOISSIER
<b>Mettre en forme l'en-tête de la page (CSS)</b>	Il s'agit de mettre en forme l' <i>header</i> de la page, en respectant au mieux les croquis présentés.	Sofiane ZEGHADI
<b>Créer la page d'enregistrement (HTML)</b>	Il s'agit de créer la page <i>s'enregistrer</i> du site. Elle devra contenir l' <i>header</i> , des zones de saisies ainsi que deux boutons, pour la confirmation et l'annulation. Les vérifications sur les champs devront également être programmées.	Floris FOURNIER
<b>Mettre en forme la page d'enregistrement (CSS)</b>	Il s'agit de mettre en forme la page <i>s'enregistrer</i> du site, en respectant au mieux les croquis présentés.	Antoine NOSAL
<b>Créer et mettre en forme la zone de recherche (HTML, CSS)</b>	Il s'agit d'intégrer à la page <i>accueil</i> une zone de recherche, autrement dit un champ de saisie et des cases à cocher correspondant aux critères de sélection. La mise en forme doit respecter au mieux les croquis présentés.	Maxime CHABOISSIER
<b>Créer et mettre en forme la zone des résultats dans la page d'accueil (HTML, CSS)</b>	Il s'agit d'intégrer à la page <i>accueil</i> une zone pour afficher des suggestions ou des résultats. Pourront être affichés des titres, des artistes ou des playlists. Ces éléments seront rassemblés par type et mis en forme selon les croquis fournis. Pour les artistes en particulier, la clique et/ou le survol devront permettant d'accéder aux informations de l'artiste.	Antoine NOSAL
<b>Créer et mettre en forme la partie « playlist en création » (HTML, CSS)</b>	Il s'agit d'intégrer à la page <i>accueil</i> une zone pour afficher la playlist en cours de création. Elle présente la liste des titres ajoutés et un bouton pour chacun d'eux, permettant de les enlever de la liste. Un champ de saisie pour le nom et un bouton de validation devront être ajoutés. La mise en forme respectera les croquis présentés.	Antoine NOSAL
<b>Créer et mettre en forme la partie « mes playlists enregistrées » (HTML, CSS)</b>	Il s'agit d'intégrer à la page <i>playlists</i> une zone listant des playlists d'un utilisateur. Initialement, on ne voit qu'un rectangle contenant le nom de la playlist, le nombre de morceaux qu'elle contient et la durée totale. En cliquant sur ce rectangle, la playlist s'affiche et l'utilisateur pourra lire les titres ou l'ensemble des titres.	Antoine NOSAL
<b>Créer et mettre en forme la partie « Découvrir » (HTML, CSS)</b>	Il s'agit d'intégrer à la page <i>playlists</i> une zone listant les playlists créées par les utilisateurs non-connectés.	Sofiane ZEGHADI
<b>Intégrer les données de la base (modèle Active Record)</b>	Selon le modèle <i>Active Record</i> , des classes PHP correspondant aux tables seront créées. Ainsi, chaque objet PHP manipulé correspondra à une ligne dans un table de la base.	Antoine NOSAL
<b>Implémenter la recherche</b>	Le lien avec les données étant programmé, il faudra désormais que la recherche soit fonctionnelle et que l'on puisse trier les résultats affichés dans la <i>zone de résultat</i> .	Antoine NOSAL
<b>Intégrer la création de playlist dynamique</b>	Il s'agit de rendre fonctionnelle la création des playlists et de les stocker dans la base. Dans un premier temps, sans aucun lien avec l'aspect compte utilisateur. Une playlist étant forcément rattachée à un utilisateur (contrainte du schéma fourni), un utilisateur « fictif » sera créé et permettra de stocker les playlists des utilisateurs non-connectés.	Antoine NOSAL
<b>Rendre fonctionnel l'enregistrement</b>	La page <i>s'enregistrer</i> devra en effet permettre d'enregistrer un utilisateur sur la base.	Sofiane ZEGHADI
<b>Rendre fonctionnelle la connexion</b>	La <i>barre de connexion</i> devra permettre de se connecter en tant qu'un utilisateur enregistré sur la base.	Antoine NOSAL
<b>Modifier la création de playlist pour qu'elle intègre l'aspect utilisateur</b>	Il s'agit d'ajouter l'aspect compte utilisateur à la création des playlists. Ainsi, si un utilisateur est connecté et qu'il crée une playlist, cette dernière lui sera associée et il pourra la retrouver dans la zone <i>mes playlists</i> de la page <i>playlists</i> .	Antoine NOSAL
<b>Afficher les playlists utilisateurs</b>	Il s'agit de lister les playlists d'un utilisateur connecté dans la page <i>playlists</i> dans la zone <i>mes playlists</i>	Antoine NOSAL
<b>Rédaction du rapport</b>	Il s'agit de rédiger un dossier/rapport de réalisation du projet.	Antoine NOSAL
<b>Préparation de la soutenance</b>	Il s'agit de préparer les diaporamas, supports de la soutenance, ainsi que de préparer cette dernière.	Maxime CHABOISSIER Antoine NOSAL Sofiane ZEGHADI

Nous tenons à préciser que l'étudiant Floris FOURNIER, à participer à une réunion de conception, et n'a réalisé qu'une seule tâche de la phase de développement (très légère). Il n'a ensuite plus donné de nouvelle. Il n'a donc en aucun cas véritablement participé à la phase de réalisation, et son implication dans la conception reste minime.