

**ADRIANO Rémi**  
**THOUVEREZ Bastien**

**Licence Maths Info**  
**Université Claude Bernard LYON1**

TODO  
MCD  
utilisateur phpmyadmin  
suppression de playlist  
verifier xamp

# At The Moment Music



Université Claude Bernard



Lyon 1

# Sommaire

<b>Introduction</b> .....	3
Outils utilisés .....	3
Choix établis .....	3
 <b>Base de données</b> .....	4
 <b>Fonctionnalités</b> .....	6
Obligatoires .....	6
<i>Ajout de données</i> .....	6
<i>Création de playlist</i> .....	7
<i>Recherche</i> .....	7
<i>Statistiques</i> .....	7
<i>Importation de Clémentine</i> .....	8
<i>Suppression de données</i> .....	8
 Supplémentaires .....	9
<i>Gestion utilisateur</i> .....	9
<i>Partage de playlist</i> .....	9
 Non implémentées .....	10
<i>Profil utilisateur</i> .....	10
<i>Commentaires et notes</i> .....	10
 <b>Problèmes rencontrés</b> .....	10
 <b>Conclusion</b> .....	10
.	
 <b>Annexes</b> .....	11
Modèle conceptuel de données .....	11
Cinématique .....	12

# Introduction

L'application ATM à été développée dans le cadre du projet PHP de LIF4 de la licence informatique de l'Université Claude Bernard Lyon 1.  
Il s'agit d'une gestion de playlist développée en PHP.

## Outils utilisés

Pour développer cette application, nous avons utilisé plusieurs langages et outils :

- HTML 5 : langage de balisage pour la conception de pages web
- CSS3 : langage de mise en forme des pages web
- JavaScript : langage de script coté client
- jQuery : framework JavaScript
- PHP 5 : langage de scripts libre coté serveur pour ajouter du dynamisme aux pages web
- MySQL : système de gestion de base de données
- Laravel : framework MVC PHP 5.3, permettant de simplifier le développement de site web
- Netbeans 7.2.1 : environnement de développement intégré
- Subversion (SVN) : logiciel de gestion de version, simplifiant grandement le travail en groupe
- Google Code : hébergement de projet
- Highcharts : bibliothèque jQuery pour la création de graphiques

## Choix établis

Nous avons choisi de travailler avec Netbeans car il est doté d'un plugin SVN. Couplé avec l'hébergeur de projet Google Code, il permet d'utiliser simplement un gestionnaire de version et facilite le travail en groupe : SVN permet de gérer les conflits dans le code si plusieurs personnes travaillent sur le même fichier simultanément.

Nous avons choisi d'utiliser un framework (Laravel) puisque nous avons déjà de bonnes connaissances en développement Web. De plus, ce framework nous est familier car nous l'avons déjà utilisé auparavant (en stage ou personnellement).

Enfin, nous avons décidé de donner le nom d'ATM à notre projet qui signifie « At The Moment », qui peut être traduit de nombreuses façons en français, mais qui est toujours lié à l'instant présent. Notre but était de trouver un nom que l'utilisateur pourrait retenir facilement, et le sigle ATM s'est avéré correspondre à notre attente.

## Base de données

La base de données de l'application est constituée de 10 tables :

- une table user\_lif contenant les données sur les utilisateurs :

Nom	Type
<u>id_user_lif</u>	int(10)
username_user_lif	varchar(10)
password_user_lif	varchar(60)
last_name_user_lif	varchar(40)
first_name_user_lif	varchar(40)
date_creation	timestamp

Table user\_lif

On retrouve un identifiant id\_user\_lif qui s'incrémente automatiquement à chaque ajout (chaque table de la base en a un), un nom d'utilisateur, un mot de passe, un prénom, un nom, une date de création.

- une table adminPlaylist :

Nom	Type
<u>id_user_lif</u>	int(10)
<u>id_playlist</u>	int(10)
code_admin	char(1)

Table adminPlaylist

Cette table lie l'identifiant d'un admin avec celui d'une playlist afin de savoir qui administre les listes de lecture. Le « code\_admin » permet de gérer des niveaux de droits : il est représenté par un caractère lettre, ainsi un administrateur de rang A et un administrateur de rang I sur une même playlist n'auront pas les mêmes droits.

- cinq tables album, band, kind, playlist, song contenant respectivement les informations concernant les albums des artistes, les groupes, les genres de musique, les listes de lectures et les chansons.

Nom	Type	Nom	Type
<u>id_band</u>	int(10)	<u>id_song</u>	int(10)
<u>id_user_lif</u>	int(10)	<u>id_user_lif</u>	int(10)
name_band	varchar(80)	title_song	varchar(100)
date_form_band	int(4)	date_song	int(4)
date_disband_band	int(4)	length_song	varchar(4)
		path_song	varchar(80)
		track_song	int(2)

Table band

Table song

Ces tables contiennent des données qui leur sont propres, et une clé étrangère sur id\_user\_lif afin de savoir quel utilisateur a ajouté les différents éléments via l'application.

- 3 tables de lien songalbum, songkind, songplaylist

Nom	Type	Nom	Type	Nom	Type
<u>id_song</u>	int(10)	<u>id_song</u>	int(10)	<u>id_playlist</u>	int(10)
<u>id_album</u>	int(10)	<u>name_kind</u>	varchar(50)	<u>id_song</u>	int(10)
Table songAlbum		Table songKind		<u>id_user_lif</u>	int(10)
				date_added	timestamp
				Table songPlaylist	

TODO verifier ce que je raconte...

Ces tables regroupent les clés étrangères pour les liaisons entre les tables. Ainsi, par exemple dans la table songPlaylist, si un utilisateur à l'identifiant 78 crée une playlist qui porte l'identifiant 42 et y ajoute 20 chansons, on trouvera dans la table 20 lignes du type :

42	id_song1	78	Date
42	id_song2	78	Date
42	id_song3	78	Date
...	...	...	...

Cette gestion est nécessaire pour ajouter une chanson à plusieurs playlist, car dans le cas d'une gestion à l'aide d'un champ id\_playlist dans la table song, la chanson serait limitée à appartenir à une seule playlist, ce qui n'est pas ce que nous recherchons. Aussi, le fait de découper les tables de cette façon amène plus de clarté lors du développement. La contrepartie de cette gestion est qu'elle prend plus de place dans la base de donnée.

# Fonctionnalités

## Concept global de l'application

L'application ATM est une application multi utilisateur permettant de créer des listes de lectures, de rechercher, d'ajouter et de supprimer du contenu (chansons, albums, artistes). Toutes les chansons contenues dans la base de donnée sont accessibles pour tous les utilisateurs via la recherche et chaque utilisateur peut ajouter du contenu pour lui, qui sera ensuite accessible pour tous les autres.

## Fonctionnalités obligatoires

L'application ATM comprend un certain nombre de fonctionnalités demandées dans le cadre du projet. Toutes les fonctionnalités sont accessibles via le menu déroulant de l'application qui est toujours visible, et divisé en quatre parties : recherche, playlist, outils et utilisateur.



Accès aux fonctionnalités de l'application

- **Ajout de données (artiste, album, chanson)**

L'utilisateur peut aussi ajouter ses propres chansons, albums et groupes via le menu 'Outils', il est simplement amené à remplir des formulaires pour rentrer les données qui seront ajoutées dans la base.

Formulaire d'ajout d'album

Par exemple, l'ajout d'album présente une liste déroulante de tous les groupes existants et demande à l'utilisateur de saisir le nom de l'album et son année de production. Par conséquent, il faut d'abord créer un artiste avant d'ajouter un album, et de même, il faut avoir créé un artiste et un album afin d'y ajouter des chansons.

- **Création de playlist**

Le menu 'Playlist' recense les playlist de l'utilisateur, et permet d'accéder à la gestion des playlist. Ainsi, la partie 'Administration' propose un bouton pour ajouter une playlist qui amène sur un simple formulaire demandant le nom que veut donner l'utilisateur à sa liste.

- **Recherche**

La recherche de chanson consiste en un formulaire présentant un champ de texte. Lorsque l'utilisateur clique sur 'Go', l'application va rechercher dans la base de données toutes les chansons, groupes, albums et genres correspondant à la recherche de l'utilisateur.

[PLAYLIST](#)
[OUTILS](#)
[DANA](#)

**4 résultats de la recherche pour "korn"**

**Chansons**

Titre	Artiste	Album	Genre	Ajouter
Hollow life	KoRn	Untouchables	Metal	
Shoot and ladders	KoRn	Untouchables	Metal	gougoule
Did my time	KoRn	Untouchables	Metal	tiagooooo
Alone I Break	KoRn	Untouchables	Metal	heygo

Résultat d'une recherche et ajout à une playlist

Après une recherche, l'utilisateur peut alors ajouter les chansons qu'il souhaite dans une playlist précédemment créée. Lorsqu'il clique sur 'Ajouter à', une fenêtre déplaçable qui affiche la liste des playlist de l'utilisateur, il clique alors sur la playlist dans laquelle il veut ajouter sa chanson.

- **Statistiques**

L'application permet aussi d'accéder à quelques statistiques de la base de données présentées sous forme de graphiques réalisés grâce à la bibliothèque jQuery nommée HighCharts.

- **Importation de Clementine**

L'application ATM permet d'importer un script SQL pour l'ajout de données dans la base. Elle consiste en un formulaire avec un bouton de type 'file' pour rechercher un fichier sur le disque dur, et un bouton de type 'submit' pour effectuer l'import. Tout fichier importé qui n'est pas un script SQL entraînera une erreur 500.

- **Suppression de données**

La suppression de contenu à proprement parlé dans la base de données n'est pas disponible dans l'application car elle est multi utilisateurs, et la suppression par exemple d'une chanson par un utilisateur entraînerait la suppression de la chanson dans toutes les playlist des autres utilisateurs qui l'auront ajouté, ce qui n'est pas ce que l'on souhaite ici. Par contre, un utilisateur qui crée une playlist peut supprimer les chansons qu'il y ajoute (les chansons seront donc seulement supprimées de la playlist, mais pas de la base de donnée), et peut supprimer la playlist elle-même. Cette suppression est possible grâce à un bouton 'Supprimer' que l'on trouve à côté de chaque chanson dans les playlist, et à côté de chaque playlist dans l'outil 'Administration' de 'Playlist'.

tiagooooo			
Titre	Artiste	Album	
Touch of Blessing	Evergrey	A Night to Remember (disc 2)	Supprimer
Porno Star	Buckcherry	Time Bomb	Supprimer

Affichage d'une playlist nommée Tiagooooo avec suppression de chansons



## Fonctionnalités supplémentaires

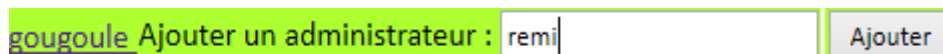
ATM dispose aussi de fonctionnalités supplémentaires :

- **Gestion d'utilisateur**

A l'ouverture de l'application, l'utilisateur est dirigé vers un écran de connexion. S'il n'est pas inscrit, il doit passer par la phase d'enregistrement via le menu de l'application : un utilisateur non enregistré ne peut pas utiliser l'application ATM.

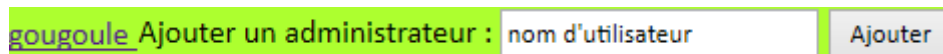
- **Partage de playlist**

Un utilisateur qui crée une playlist peut la partager à un (ou plusieurs) autre(s) utilisateur(s) s'il connaît son(leurs) nom(s) d'utilisateur.



Invitation d'utilisateur à une playlist

Pour se faire, le créateur de la playlist doit saisir le nom de l'utilisateur qu'il veut inviter à sa playlist dans le champ de texte prévu à cet effet présent à côté de chaque playlist. S'il souhaite le supprimer à l'avenir, il doit cliquer sur le nom de sa playlist, alors apparaîtront tous les utilisateurs qu'il a invité avec un bouton muni d'une croix permettant la suppression.



remi X

Suppression d'utilisateur à une playlist

## Fonctionnalités non implémentées

Enfin, d'autres fonctionnalités ont été prévues au début du projet mais n'ont pas eu le temps d'être développées : la base de données et le site sont prêts à les accueillir mais elles ne sont pas encore implémentées.

- **Profil utilisateur**

L'utilisateur devrait être en mesure de modifier son profil : son nom d'utilisateur, son mot de passe et les autres informations relatives au compte. Cette fonctionnalité serait accessible via le menu utilisateur, et serait doté de formulaires permettant simplement de modifier les données de l'utilisateur dans la base de données.

- **Gestion des commentaires et notes**

Nous voulions implémenter un système de commentaires de chansons avec une note associée pour chaque commentaire. Une table com est déjà présente dans la base de données afin de gérer cette fonctionnalité :

Nom	Type
<u>id_com</u>	int(10)
id_user_lif	int(10)
id_song	int(10)
vote_com	int(1)

Table com

Cette table contient simplement un id pour le commentaire, l'id de l'utilisateur qui commente, l'id de la chanson commentée et une note pour la chanson.

## Problèmes rencontrés

La prise en main d'un framework n'est pas une chose aisée. L'un de nous connaissait bien le framework Laravel pour avoir travaillé avec à plusieurs reprises mais l'autre n'en avait jamais utilisé : il nous aurait fallu plus de temps de projet afin de bien le prendre en main pour pouvoir développer les fonctionnalités manquantes et passer plus de temps sur la partie graphisme de l'application.

Le principal problème que nous avons rencontré a donc été le manque de temps. Aussi, le fait que nous n'avions pas de planning précis a joué sur l'organisation du développement du projet. En avoir un nous aurait probablement permis de développer les deux fonctionnalités non implémentées.

TODO compléter si besoin

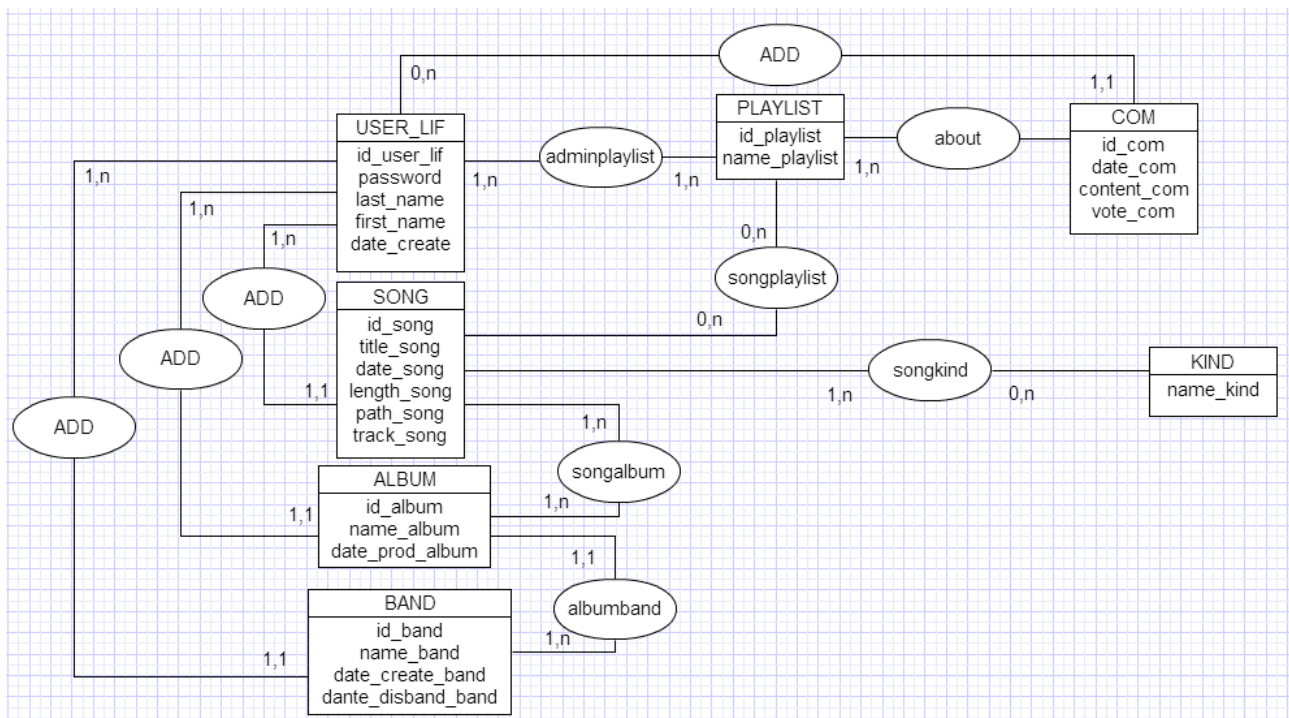
## Conclusion

Ce projet nous a permis de renforcer encore nos connaissances en HTML, CSS et PHP et nous a permis de nous familiariser avec des outils et bibliothèques du web comme le framework PHP Laravel ou la bibliothèque jQuery HighCharts qui nous a permis de faire les graphiques pour les statistiques.

TODO compléter, remplir du vide, brasser de l'air (technique du brassage d'air que j'ai appris en stage;))

## Annexes

- **Modèle conceptuel de données (MCD)**



- **Cinématique du projet**

