

# Projet de WEBRAILS

*Plylst.it*

*Sacha Bron & Benoist Wolleb*





## Table des matières

Introduction.....	3
Cahier des charges.....	3
Description.....	3
Fonctionnement.....	3
Structure de la base de données.....	3
Cas d'utilisation et définition des droits des utilisateurs.....	5
Scénarios principaux.....	6
Structure du logiciel.....	7
Implémentation.....	9
Partie AJAX.....	9
Bibliothèques utilisées.....	9
Particularités.....	10
Gestion de projet.....	11
État des lieux.....	12
Conclusion.....	13
Annexes.....	14
Manuel d'installation.....	14
Manuel d'utilisation.....	15
Accueil anonyme.....	15
Affichage des listes d'un utilisateur.....	16
Lecture d'une liste.....	17
Création d'un compte.....	18
Connexion.....	18
Accueil d'un utilisateur connecté.....	19
Édition du compte.....	20
Affichage des playlists.....	21
Édition de playlist.....	22
Création d'une playlist.....	23
Ajout d'une chanson.....	24
Affichage des favoris.....	25
Affichage des amis.....	26
Envoi d'une invitation.....	27
Accueil d'un administrateur.....	28
Gestion des utilisateurs.....	29
Listes d'un utilisateur.....	30



## Introduction

Ce projet prend place dans le cours WEBRAILS sous la forme d'un petit site web qui, bien que rudimentaire, va nous permettre d'appréhender la plateforme Ruby on Rails. Afin qu'il soit un peu plus amusant, nous avons décidé de créer un site en rapport avec du multimédia : ici, des données audio.

## Cahier des charges

### Description

plylst.it (prononcé "playlist it") est un outil de création de liste de lecture musicale. Son atout principal est la conversion de musique à partir de vidéo YouTube, de morceau SoundCloud, et autres services de streaming, vers du HTML5.

### Fonctionnement

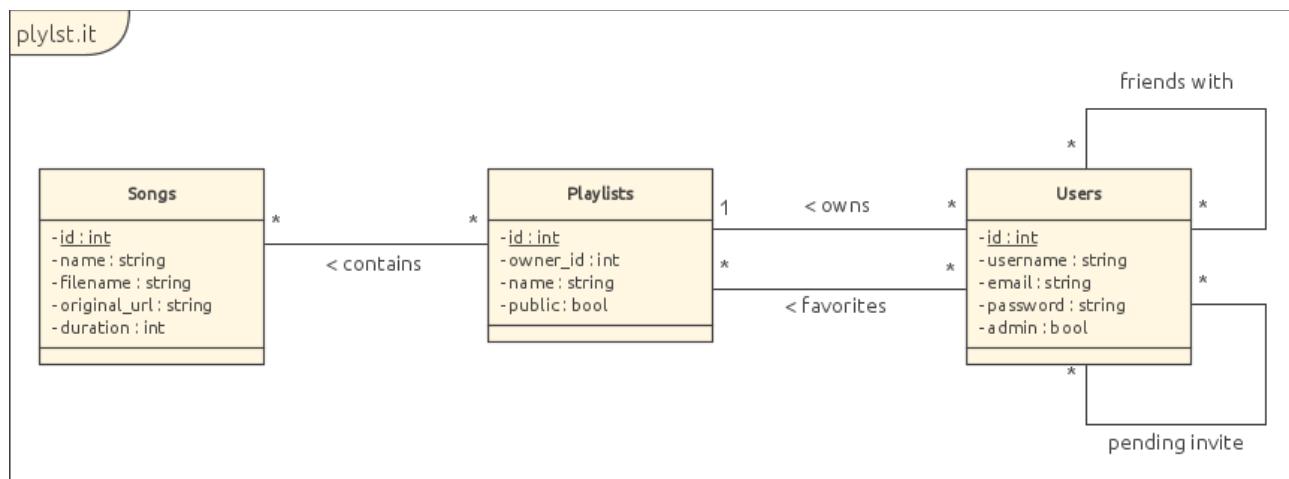
Un utilisateur inscrit peut créer une liste de lecture (publique ou privée) et y ajouter des morceaux. En pratique, il lui suffit d'envoyer le lien du média désiré et le serveur se charge de télécharger le fichier désigné. S'il s'agit d'une vidéo (lors d'utilisation de liens Youtube par exemple) la piste audio est extraite.

Les listes de lectures ainsi constituées peuvent être écoutées via l'interface web du site. Lors de la lecture, le navigateur du client récupère et lit un fichier musical à la fois, en récupérant dynamiquement le suivant de la liste une fois qu'une piste se termine.

Un système interne d'amitiés permet à des utilisateurs de se partager les listes et il est possible de mémoriser des listes en tant que favoris.

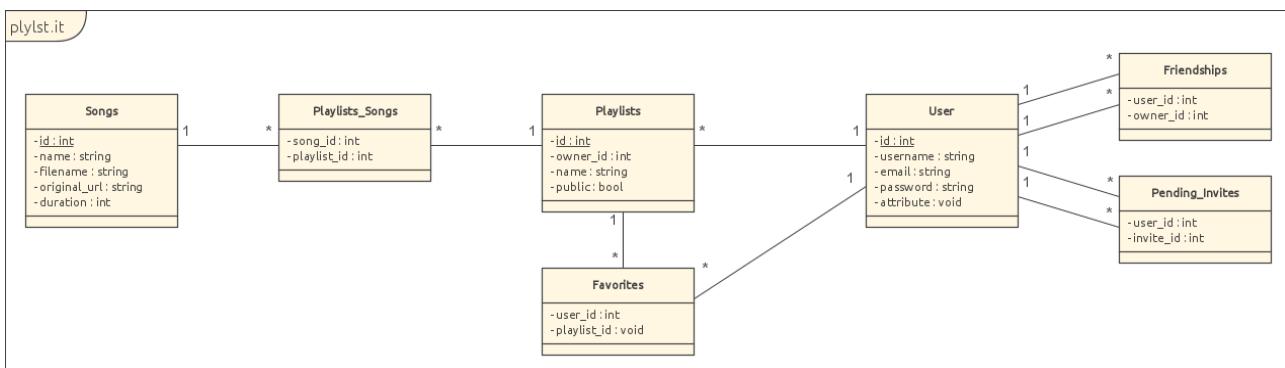
### Structure de la base de données

Le diagramme UML de la base de données est le suivant:





En y intégrant les tables d'association pour les relations de types N-N, on obtient:



Chaque enregistrement de la table Songs correspond à un fichier audio enregistré côté serveur dans un dossier public afin d'être disponibles lorsqu'un client souhaite les écouter. Une chanson est caractérisée par son titre, extrait automatiquement de la page d'origine, sa durée en secondes, et le nom du fichier auquel la chanson correspond. À noter que nous conservons l'URL d'origine du média, ainsi, si plusieurs utilisateurs ajoutent la même chanson à leurs listes de lecture, le fichier n'est véritablement téléchargé qu'une fois.

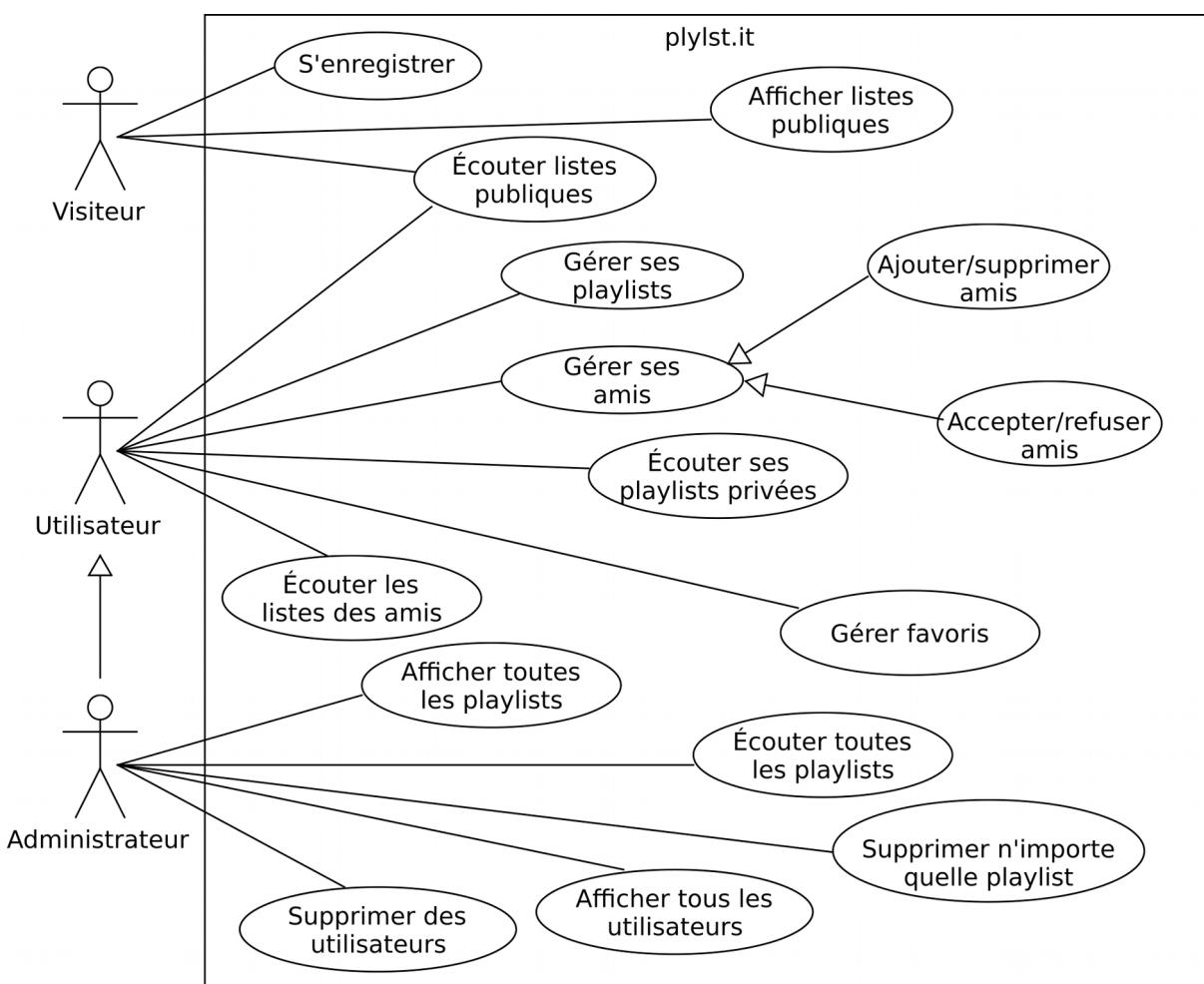
Chaque enregistrement de la table Playlist correspond à une liste de lecture créée par un utilisateur. Une liste a forcément un nom et un booléen qui indique si elle est publique (accessible par tous) ou privée (accessible uniquement à l'utilisateur et à ses amis). La table d'association Playlists\_Songs permet d'établir une relation N-N entre les listes et les chansons.

La table User contient les informations des utilisateurs enregistrés sur le site. On y retrouve les champs classiques d'une telle situation en stockant le pseudonyme, l'adresse mail, le mot de passe haché, etc... Notons ici la présence d'un booléen indiquant si l'utilisateur est administrateur ou non.

La table Favorite établit une relation N-N entre les utilisateurs et les listes, ce qui permet de mémoriser listes de lecture d'autres utilisateurs en favoris, ce qui permet, via la page adéquate, de les retrouver facilement. D'autre part, les favoris permettent de "classer" les listes de lecture par popularité.

Enfin, les deux tables Friendships et Pending\_Invites permettent de gérer les relations d'amitié entre les utilisateurs. Lorsqu'un utilisateur souhaite devenir ami avec un autre, une demande est enregistrée dans Pending\_Invites. L'utilisateur faisant objet de la demande peut alors l'accepter ou la refuser, après quoi l'enregistrement dans Pending\_Invites est supprimé et deux entrées sont créées dans la table Friendships. Ne stocker qu'un seul enregistrement fonctionne, mais il est plus simple dans notre cas de stocker l'amitié dans les deux sens, comme si la relation n'était pas symétrique ( $U1 \rightarrow U2$  et  $U2 \rightarrow U1$ ).

## Cas d'utilisation et définition des droits des utilisateurs



Un visiteur peut:

- Parcourir les playlists publiques
- Écouter les playlists publiques
- Créer un compte

Un utilisateur connecté peut:

- Se connecter / se déconnecter
- Créer / supprimer des playlists
- Ajouter / supprimer des morceaux dans ses playlists
- Écouter ses propres playlists privées
- Écouter les playlists privées de ses amis
- Ajouter / supprimer des amis
- Accepter / refuser des invitations d'amis
- Ajouter / supprimer des playlists en favoris



Un administrateur peut :

- Afficher les utilisateurs
- Supprimer les utilisateurs
- Supprimer n'importe quelle playlist
- Afficher toutes les playlists
- Écouter toutes les playlists
- Mettre en favori n'importe quelle playlist

### Scénarios principaux

1) Un client non inscrit sur le site arrive sur la page d'accueil. Cette dernière liste les listes de lecture publiques créées par des utilisateurs. Les listes sont affichées par ordre décroissant de popularité, caractérisé par le nombre de fois où la liste a été mise en favori. L'utilisateur non inscrit peut cliquer sur une de ces listes et écouter toutes les chansons qu'elle contient.

2) Un utilisateur inscrit se connecte sur le site. La page d'accueil affiche les listes de lecture publiques mais aussi les listes de lectures privées de ses amis. L'utilisateur peut cliquer sur une de ces listes et en écouter le contenu.

3) Un utilisateur inscrit écoute une liste de lecture publique ou privée à laquelle il a accès et met celle-ci en favori. La liste est ainsi mémorisée et affichée dans la page adéquate. Le nombre de favoris de la liste de lecture s'est incrémenté et la liste est montée sur la page d'accueil affichant les listes recommandées.

4) Un utilisateur inscrit crée une nouvelle liste de lecture et y poste des chansons. La liste peut, par la suite, être renommée, modifiée ou supprimée.

5) Un utilisateur inscrit envoie une invitation d'amitié à un autre utilisateur via le panneau adéquat. L'autre utilisateur confirme la relation d'amitié, et tous deux ont accès aux listes privées de l'autre. Les relations d'amitié peuvent par la suite être supprimées.

6) Un administrateur affiche tous les utilisateurs et, pour chacun, a accès à toutes leurs listes de lecture, y compris les privées. L'administrateur peut les écouter, les placer en favori, ou les supprimer.

7) Un administrateur affiche tous les utilisateurs via le panneau de gestion et peut supprimer n'importe lequel.

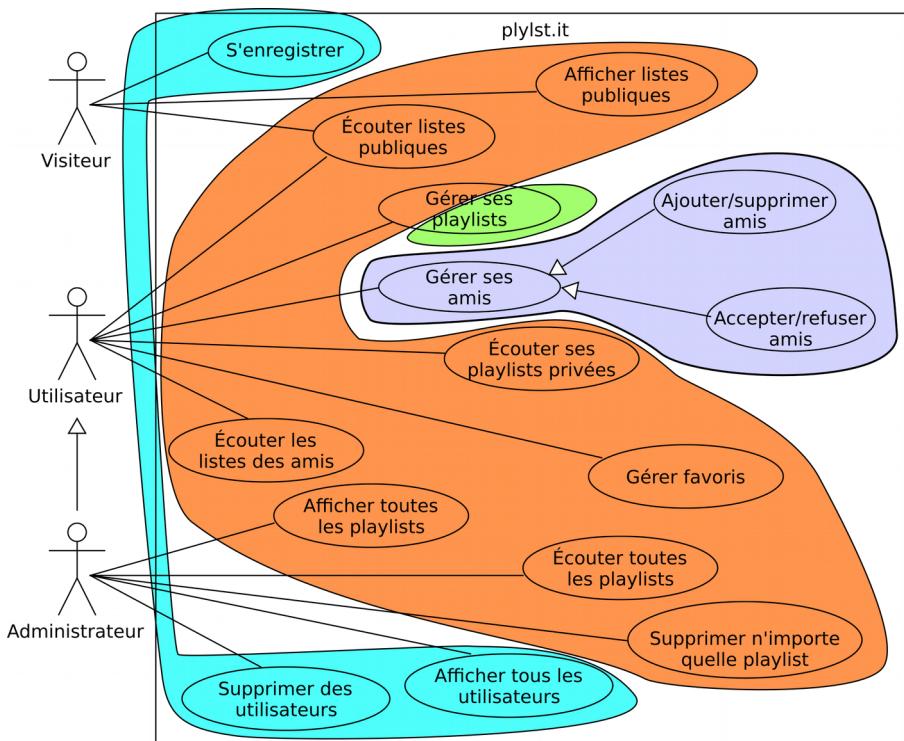


## Structure du logiciel

Notre projet comporte 4 contrôleurs :

- ■ Song : Principalement responsable de télécharger le média d'origine et d'extraire la piste audio si c'est la première fois que le lien est posté, sinon il faut simplement établir la liaison avec l'enregistrement déjà existant. Ce contrôleur est donc responsable de créer l'enregistrement de la chanson dans la base de données et aussi de créer l'enregistrement dans la table Playlists\_Songs pour les playlists que l'utilisateur a sélectionné lors de l'envoi. Cette tâche aurait pu se trouver dans le contrôleur de playlist, mais tout est fait dans la même fonction pour des raisons pratiques.
- ■ Playlist : Responsable de toutes les opérations qui se font sur les listes de lecture (ajout, suppression, modification) avec gestion des droits. Par exemple, ne peut avoir accès à une liste privée que son créateur ou un de ses amis. C'est aussi dans ce contrôleur qu'est géré l'enregistrement de favoris et l'affichage des recommandations de listes de lecture sur la page d'accueil par ordre de popularité. Enfin, c'est dans ce contrôleur que se gère la suppression d'une chanson dans une liste. En effet, la suppression d'une piste concerne principalement la suppression de l'enregistrement dans la table Playlists\_Songs, car il ne faut pas supprimer l'enregistrement de la chanson elle-même, au cas où d'autres listes y font appel.
- ■ User : Responsable simplement de la gestion de l'utilisateur avec contrôle d'accès pour les tâches d'édition et de suppression.
- ■ Friendships : Responsable de tout ce qui concerne les relations d'amitié. Ce contrôleur est une extension du contrôleur des utilisateurs et gère les invitations et les relations d'amitié dans la base de données.

Au final, par rapport aux cas d'utilisation, voici comment se découpent les rôles :



Enfin, une multitude de vues sont responsables du bon affichage de notre site. La plupart sont banales et correspondent aux actions de bases des ressources (créer, afficher, modifier, supprimer) mais quelques unes sont notables :

- **discover:** Cette vue affiche toutes les listes de lectures accessibles dans le contexte donné (listes publiques et listes privées des amis) par ordre de popularité, c'est à dire où les listes qui ont été placées le plus de fois en favori sont placées en premier. Cette vue affiche aussi un bref aperçu des pistes qui y sont contenues en affichant jusqu'à 3 chansons. Cette vue permet aussi à l'administrateur de supprimer une liste de lecture.
- **show:** Cette vue est celle qui permet de lire les morceaux d'une liste. Elle se compose d'un lecteur multimédia HTML5 qui propose quelques contrôles classiques (play, pause, précédent, suivant et une barre de progression) et qui lit successivement tous les morceaux de la liste. Les morceaux sont affichés et il est possible de sauter à une autre piste à tout moment en cliquant dessus. D'autre part, cette page permet de placer la liste actuelle dans ses favoris ou de l'en retirer, ce qui se fait via l'afficheur actif en forme de cœur. L'ajout ou la suppression du favori se fait dynamiquement en AJAX afin de ne pas interrompre la lecture.
- **show\_users:** Cette vue n'est accessible que par un administrateur et affiche tous les utilisateurs enregistrés sur le serveur. Via cet affichage, il est possible de supprimer un utilisateur ou d'afficher ses playlists.



## Implémentation

### Partie AJAX

Comme dit précédemment, la petite partie de notre site faisant appel à AJAX est l'ajout et la suppression de listes de lecture en favori. La technologie AJAX permet de communiquer dynamiquement avec le serveur sans rechargement de page, ce qui permet d'effectuer des opérations en arrière-plan sans interrompre l'action en cours.

Ici, cette utilisation est pertinente car, notre site étant orienté multimédia, la liste sera en cours de lecture lorsque l'utilisateur va décider d'ajouter ou de supprimer son favori sur celle-ci. Or, un rechargement de page couperait la lecture des pistes. Ici, la communication AJAX nous permet de faire ceci sans interruption.

Anecdote : on aperçoit à l'oeil qu'il s'agit d'un mécanisme asynchrone en arrière-plan, car lorsque l'on clique sur le bouton en forme de coeur pour ajouter le favori, celui-ci ne devient rouge qu'un petit peu plus tard, marquant la confirmation que la liste est bien mémorisée.

Dans la vue, la requête est codée comme suit :

```
<%= link_to '♥', playlists_favorite_path(playlist:  
@playlist), id: "fav", class: (@playlist.users.include?  
(current_user) ? 'active' : ''), remote: true %>
```

On utilise le mécanisme interne de Rails pour faire des requêtes AJAX en spécifiant le paramètre *remote*.

### Bibliothèques utilisées

Afin de simplifier l'implémentation de différentes parties de notre application, nous avons utilisé le module Devise qui gère pour nous le système de comptes. En effet, ce module met en place automatiquement les opérations :

- Création de compte
- Connexion
- Rester connecté (via une checkbox sur la page de login)
- Gestion du mot de passe, haché dans la base de données
- Récupération du mot de passe
- Édition du compte



## Particularités

Le cœur de notre application, sa partie métier, constitue en télécharger des médias venant de grands sites de distribution de contenu et d'en extraire la piste audio. La faisabilité de cette partie définissait à elle toute seule la faisabilité de tout notre projet. En effet, les sites proposant du contenu multimédia à l'instar de Youtube, Dailymotion ou Soundcloud ne permettent pas, nativement, de télécharger le contenu qu'ils proposent. Au contraire, ils mettent même tout en œuvre pour l'empêcher. Dès lors, braver ces mécanismes de protection était un *challenge* annoncé.

Heureusement pour nous, nous ne sommes pas les seuls qui désirent opérer une extraction de contenu, loin de là. En effet, cela fait plusieurs années, quasiment depuis que les services médias sur le web sont apparus, qu'existent des sites permettant d'en extraire la sève. D'innombrables sites permettent, par exemple, de télécharger un fichier mp3 en postant un lien Youtube.

L'outil utilisé pour télécharger et extraire la piste audio est le logiciel libre youtube-dl, présent de base dans les dépôts des principales distributions Linux, qui supporte un nombre impressionnant de plateformes (795 officiellement) parmi lesquelles Youtube, Soundcloud, Dailymotion, Vimeo, etc...

Lorsqu'un utilisateur de notre site souhaite mettre une nouvelle chanson dans une de ses listes de lecture, notre interface lui propose un simple champ dans lequel un lien peut être posté. Comme expliqué plus haut, c'est le contrôleur Song qui se charge de cette tâche.

Celui-ci effectue un appel système vers le logiciel youtube-dl afin de télécharger le média, d'en extraire la piste et de récupérer les métadonnées, principalement le titre et la durée, pour être enregistré dans la base de données.

Le fichier final est renommé de manière générique, en utilisant la date et l'heure en format UNIX (nombres de secondes écoulées depuis le 1<sup>er</sup> janvier 1970) ainsi que les nanosecondes, afin de garantir que tous les fichiers porteront un nom distinct. Le fichier final est placé dans le dossier public, accessible à tous.

```
# Récupération et traitement des métadonnées
cmd = "youtube-dl -J " + song_params[:original_url] + "
2>> /dev/null"
IO.popen(cmd) { |f| data = f.gets }
metadata = JSON.parse(data)

# Construction du nom du fichier
filename = `date +%s%N | tr -d "\n"`

# Téléchargement et extraction avec youtube-dl
value = system("youtube-dl", song_params[:original_url], "-
x", "--audio-format", "mp3", "-o", Dir.pwd +
"/public/downloads/" + filename + ".%(ext)s")
```



## Gestion de projet

Nous avons géré notre projet en le découplant en plusieurs itérations afin de travailler de manière progressive. Les itérations que nous avions prévues au départ et que nous avons tenues sont les suivantes :

### ■ Itération 1 (10.05.2016) :

- Création de la base de donnée et des différentes vues (statiques)

### ■ Itération 2 (17.05.2016)

- Système de conversion et stockage des fichiers audio
- Création des contrôleurs de création/modification/suppression des playlists

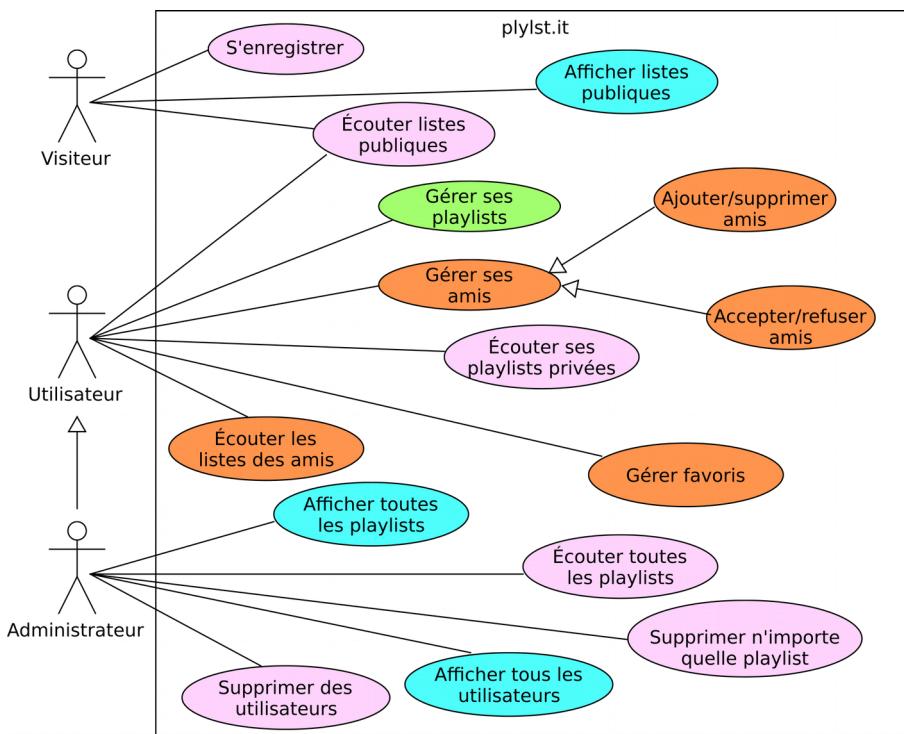
### ■ Itération 3 (31.05.2016)

- Création du système de comptes et d'administration
- Création des contrôleurs de création du compte/connexion/déconnexion
- Création du système d'écoute de playlists

### ■ Itération 4 (07.06.2016)

- Création du système d'amitiés
- Ajout de la fonctionnalité de favoris
- Tests

Ce qui, dans le schéma des cas d'utilisation, correspond à :





## État des lieux

Dans l'état actuel, notre site est fonctionnel et offre toutes les fonctionnalités prévues au départ. Cependant, comme rien n'est parfait, les points suivants peuvent être améliorés :

- Meilleure gestion des accès : Nous avons porté une attention particulière à la gestion des accès. Ainsi, un utilisateur n'ayant pas les autorisations pour une liste privée ne pourra pas en forcer l'accès, en mettant le chemin directement dans la barre d'adresse par exemple. Effectuer ceci aura pour effet de rediriger la page sur le login si l'utilisateur n'est pas encore enregistré. Cependant, il reste probablement des situations où le contrôle d'accès peut être amélioré.
- Meilleure gestion des erreurs : Il reste probablement quelques coquilles sur certaines pages du site. Un bug connu, notamment, est de poster un URL incorrect ou d'un site non reconnu par youtube-dl lors de l'ajout de morceaux.
- Gestion des playlists : Actuellement, il n'est possible d'ajouter qu'un seul média à la fois. Cependant, youtube-dl supporte le téléchargement de listes complètes sur certains sites. Il serait donc pertinent de développer la fonctionnalité afin que l'utilisateur puisse ajouter toute une liste en un seul coup.
- Lecture plus riche : Il serait de bon ton de pouvoir ordonner les pistes dans une liste de lecture ou de proposer un mode de lecture aléatoire.
- Meilleure gestion des doublons : Comme dit précédemment, à l'ajout d'un morceau, le contrôleur va d'abord vérifier si ce lien a déjà été utilisé par un autre utilisateur et va simplement utiliser le fichier déjà existant. Cependant, pour ce faire, nous effectuons simplement une recherche de chaîne de caractères sur les URL stockées dans la table Songs. Or, il est possible que la cible soit la même pour deux URL différents, par exemple les trois adresses suivantes indiquent la même vidéo :

Lien original: <https://www.youtube.com/watch?v=sOnqjkJTMaA>

Lien avec paramètre: <https://www.youtube.com/watch?v=sOnqjkJTMaA?t=6m39s>

Lien raccourci: <https://youtu.be/sOnqjkJTMaA>

- Meilleure gestion des amis : Actuellement, la page qui affiche les amitiés est assez rudimentaire. On ne peut qu'ajouter ou supprimer des amis. Il serait probablement utile, et dans l'air du temps, d'insérer des fonctionnalités "sociales" comme l'envoi de messages entre utilisateurs. D'autre part, l'ajout d'amis se fait via un affichage brutal de tous les utilisateurs, ce qui n'est pas une solution viable lorsque beaucoup d'utilisateurs existent. Il faudrait probablement créer un champ de recherche.
- Autres méthodes de connexion : La plupart des gens aiment, à tort, utiliser le même compte pour tout leur environnement internet. Afin de ne pas "gêner" l'utilisateur en le forçant à se créer un énième compte, nous pourrions implémenter un système nous permettant d'utiliser un compte Facebook, Google ou encore OpenID.



## Conclusion

Ce projet nous a permis de découvrir la plateforme Ruby on Rails et nous sommes satisfaits du travail accompli. Ruby est une plateforme web fiable et mûre et c'est avec plaisir que nous l'avons utilisée. En effet, dans notre formation, nous avons eu un aperçu d'autres plateformes qui poursuivent le même but, comme JavaEE ou Express.js, et découvrir celle-ci complète le tableau des possibles alternatives.





## Annexes

### Manuel d'installation

Durant le développement du projet, nous avons travaillé sur la version 5 de Ruby on Rails qui nécessite Ruby en version  $\geq 2.2.2$ , le Gemfile contient donc des versions de dépendances très récentes, ce qui peut poser problème en cas d'environnement avec Ruby on Rails en version 4 ou d'une version de Ruby trop ancienne.

Prérequis :

- Disposer de Ruby en version  $\geq 2.2.2$
- Disposer de Ruby on Rails en version 5
- Disposer du logiciel youtube-dl dans le PATH système
- Disposer d'un serveur de base de données compatible (MySQL / MariaDB par exemple)

Sur un système d'exploitation GNU / Linux basé Debian, ces prérequis sont installés comme suit :

```
> sudo apt-get install ruby ruby-dev youtube-dl mysql-server  
libmysqlclient-dev zlib1g-dev  
> sudo gem install rails
```

Les paquets ruby-dev, libmysqlclient-dev et zlib1g-dev sont nécessaires afin que Ruby puisse compiler les extensions natives dont il a besoin.

Puis exécuter les tâches suivantes :

- 1) Dans le dossier du projet, installer les dépendances en exécutant la commande : bundle install
- 2) Créer la base de données playlist\_it dans le SGBDR choisi
- 3) Éditer le fichier config/database.yml pour paramétriser les accès à la base de données
- 4) Charger le schéma de la base de données : rake db:schema:load RAILS\_ENV=development
- 5) Lancer le serveur avec la commande : rails s
- 6) Le site est actif à l'adresse : <http://localhost:3000>



## Manuel d'utilisation

### Accueil anonyme

À l'arrivée sur le site, l'utilisateur non inscrit arrive sur cette page :

The screenshot shows the homepage of [plylst.it](#). At the top right are links for "Register" and "Login". On the left, there's a "Discover" section with three items:

- Best public playlist — Benoit — 1 ❤**  
Michael Jackson - Earth Song  
Elton John - Sacrifice
- ben2 public playlist — Ben2 — 0 ❤**  
Michael Jackson - They Don't Care About Us  
Michael Jackson - Thriller  
Survivor - Eye Of The Tiger
- ben3 public playlist — Ben3 — 0 ❤**  
Michael Jackson - Earth Song  
Michael Jackson - Black Or White

Below these is another section:

- My other public playlist — Ben2 — 0 ❤**  
Rihanna - Needed Me  
MIKA - Hurts (Remix - Official Video)

A large image of a DJ at a turntable is visible on the right side of the page.

Le bandeau au sommet de la page est présent sur toutes les pages du site et affiche des liens vers des pages spécifiques qu'il est possible d'atteindre selon le contexte. Ici, avec un utilisateur anonyme, peu d'actions sont possibles :

- Créer un compte
- Se connecter
- Découvrir les listes disponibles (page d'accueil)

Cette dernière permet d'avoir un aperçu des listes disponibles. Celles qui sont affichées ici sont toutes publiques et classées par ordre de popularité (selon le nombre de favoris). Chaque liste est caractérisée par :

- Son nom, un clic mènera vers la page d'écoute
- Son auteur, un clic mènera vers l'aperçu de toutes les listes accessibles de l'utilisateur
- Le nombre de fois que la liste a été mise en favori

En dessous s'affiche un aperçu de 3 morceaux au maximum que la liste contient.



### Affichage des listes d'un utilisateur

Sur la page précédente, en cliquant sur l'auteur, on accède à la liste de ses listes accessibles dans le contexte actuel, donc ici, uniquement les listes publiques :

The screenshot shows a DJ, Ben2, in a dark room, wearing a t-shirt with a jigsaw puzzle graphic. He is operating a DJ setup with turntables and a mixer. The website interface is visible on the left, displaying his public playlists.

Discover      Register      Login

# plylst.it

## Playlists created by Ben2

**ben2 public playlist — Ben2 — 0 ❤**

Michael Jackson - They Don't Care About Us  
Michael Jackson - Thriller  
Survivor - Eye Of The Tiger

**My other public playlist — Ben2 — 0 ❤**

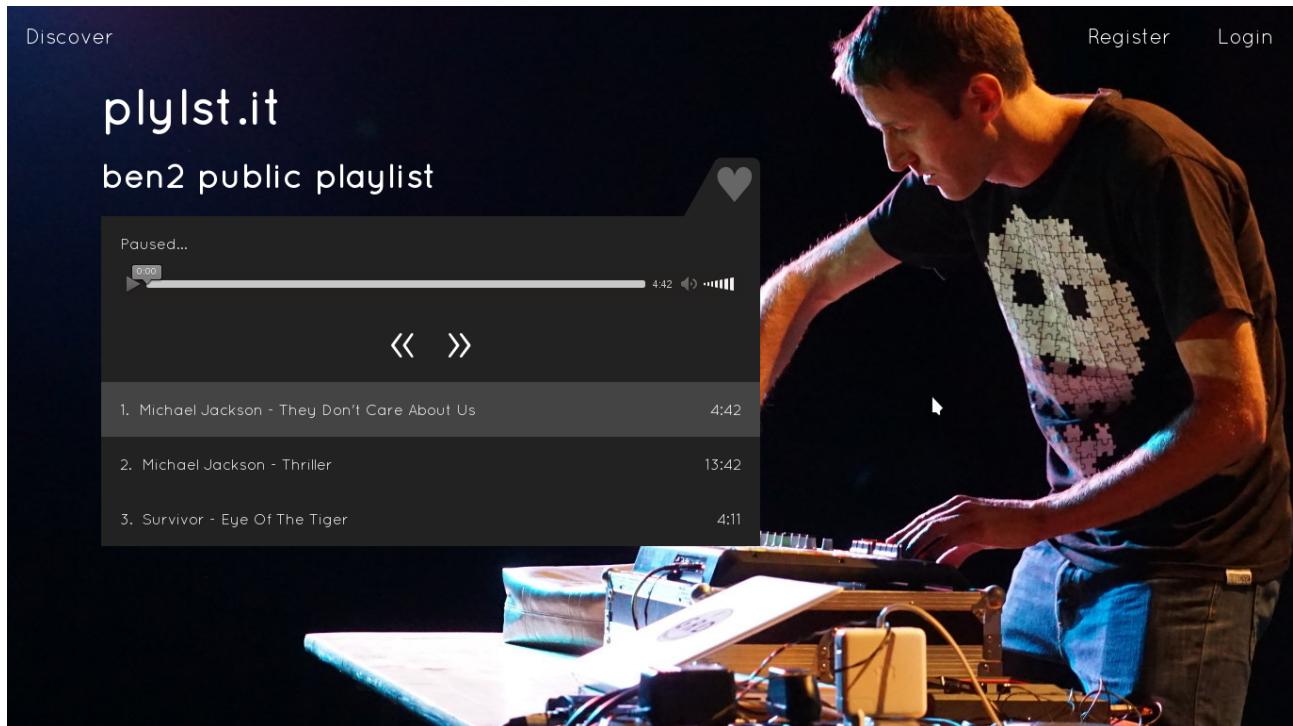
Rihanna - Needed Me  
MIKA - Hurts (Remix - Official Video)

Cette page est très semblable à celle de la page d'accueil, sauf qu'elle n'affiche les listes que de l'utilisateur sélectionné. Les listes s'affichent selon leur disponibilité dans le contexte actuel, ici par exemple, on a accès aux deux listes privées de Ben2, mais sa liste privée ne s'affiche pas.



### Lecture d'une liste

Tout comme sur la page d'accueil, en cliquant sur le nom d'une liste, on peut accéder à son contenu :



Voici la fonctionnalité principale de notre site. Ici, il est possible d'écouter les morceaux qui composent la liste actuelle. Le lecteur possède les contrôles de base permettant de gérer la lecture :

- Un afficheur qui montre l'activité actuelle et le nom de la piste s'il y en a une à l'écoute
- Un bouton play / pause qui permet de commencer ou arrêter la lecture
- Une barre de progression permettant de visualiser et modifier la position de lecture
- Un contrôle de volume
- Deux boutons précédent / suivant

Tous les morceaux de la liste sont affichés en dessous et il est possible de lire directement un morceau en cliquant dessus. Lorsqu'une chanson se termine, la suivante est lue automatiquement.



### Création d'un compte

Lorsque l'utilisateur veut se créer un compte, la page accessible via le lien "Register" s'affiche :

Discover

Register Login

# plylst.it

## Sign up

Username

Email

Password (6 characters minimum)

Password confirmation

### Connexion

Si l'utilisateur possède un compte, le lien "Login" mène vers la page de connexion :

Discover

Register Login

# plylst.it

## Log in

Login

Password

Remember me



## Accueil d'un utilisateur connecté

Un utilisateur connecté aura une page d'accueil similaire à un utilisateur anonyme :

The screenshot shows the homepage of the [plylst.it](#) website. At the top, there are navigation links: Discover, Friends, Favorites, Add Song, My Playlists, Edit Account, and Logout. Below the navigation, the word "Discover" is prominently displayed. Three playlists are listed:

- Best public playlist — Benoist — 2 ❤**  
Michael Jackson - Earth Song  
Elton John - Sacrifice
- ben3 public playlist — Ben3 — 1 ❤**  
Michael Jackson - Earth Song  
Michael Jackson - Black Or White
- ben3 private playlist — Ben3 — 0 ❤**  
Ibrahim Maalouf - Baikal (Extrait de la bande originale du film "Dans Les Forêts De Sibérie")

On the right side of the screen, there is a photograph of a man with short hair, wearing a black t-shirt with a graphic of a dollar sign made of puzzle pieces, and blue jeans. He is sitting at a DJ setup with various equipment, including a turntable and a mixer, and appears to be adjusting knobs or buttons.

Sur la page d'accueil sont affichées les listes n'appartenant pas à l'utilisateur, ce qui lui permet d'en découvrir de nouvelles. Les listes de l'utilisateur Ben2 ne sont donc plus affichées. D'autre part, étant ami avec l'utilisateur Ben3, on voit que l'on peut accéder à ses listes privées, ce qui n'était pas le cas avant.



## Édition du compte

Comme tout système de compte, il est possible d'éditer le compte de l'utilisateur, en accédant à la page via le lien "Edit Account" du bandeau :

The screenshot shows a user profile editing interface. At the top, there are navigation links: Discover, Friends, Favorites, Add Song, My Playlists, Edit Account, and Logout. The main title is "Edit User". The form contains the following fields:

- Username: ßen2
- Email: benoist.wolleb@la-cave.ch2
- Password (leave blank if you don't want to change it): [empty input]
- Password confirmation: [empty input]
- Current password (we need your current password to confirm your changes): [empty input]

Below the form are two buttons: "Update" and "Cancel my account". Under "Cancel my account", there is a link "Unhappy?". At the bottom left is a "Back" button.



### Affichage des playlists

Évidemment, il est possible à l'utilisateur d'afficher les listes qu'il a déjà créées :

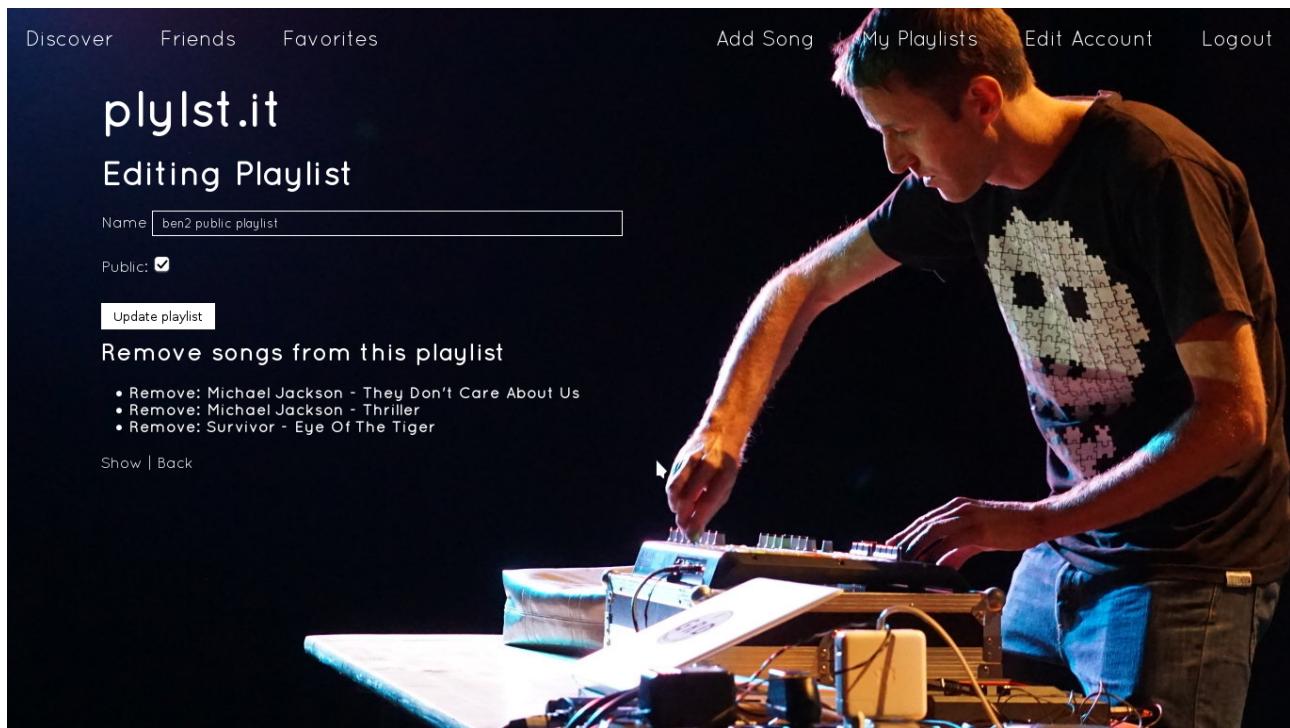
The screenshot shows a web application interface for managing playlists. At the top, there are navigation links: Discover, Friends, Favorites, Add Song, My Playlists, Edit Account, and Logout. Below this, the title "plylst.it" is displayed, followed by "My playlists". Underneath, there are two entries: "ben2 public playlist" and "ben2 private playlist", each with "Edit" and "Destroy" links. A "Create a new playlist" button is located below these entries. The background of the page features a photograph of a DJ in a black t-shirt with a dollar sign graphic, working at a turntable setup.

Il est possible depuis cette page d'accéder à ses propres playlists pour les écouter, de les éditer (changer leur nom par exemple) ou de les supprimer. Le dernier lien permet d'en créer de nouvelles.



## Édition de playlist

La page d'édition de la playlist, accessible depuis la page précédente, se présente ainsi :



The screenshot shows a web application interface for editing a playlist. At the top, there are navigation links: Discover, Friends, Favorites, Add Song, My Playlists, Edit Account, and Logout. The main title is "plylst.it" followed by "Editing Playlist". Below the title, there are input fields for "Name" (set to "ben2 public playlist") and "Public:" (with a checked checkbox). A "Update playlist" button is present. To the right of the form, there is a photograph of a man (Benoist Wolleb) wearing a t-shirt with a dollar sign graphic, sitting at a turntable and mixer. Below the form, there is a section titled "Remove songs from this playlist" with a list of three songs: Michael Jackson - They Don't Care About Us, Michael Jackson - Thriller, and Survivor - Eye Of The Tiger. At the bottom left, there are "Show" and "Back" buttons.

Il est possible de modifier son titre ainsi que sa visibilité. La partie inférieure de la page permet de supprimer des pistes actuellement présentes.



### Création d'une playlist

Accessible depuis la page d'affichage des listes de l'utilisateur, la création se présente ainsi :

The screenshot shows a web application for creating a new playlist. At the top, there are navigation links: Discover, Friends, Favorites, Add Song, My Playlists, Edit Account, and Logout. The main title is "New Playlist". Below it, there are two input fields: "Name" and "Public:" with a checkbox. A "Create playlist" button is located below these fields. At the bottom left is a "Back" link. The background of the page features a photograph of a DJ wearing a t-shirt with a dollar sign graphic, operating a DJ setup with turntables and mixers.

On notera ici la présence de la checkbox permettant de régler la visibilité de la playlist. Les listes sont par défaut privées.



### Ajout d'une chanson

Lorsque l'utilisateur a créé des listes, il faut y mettre des morceaux via la page "Add Song" disponible depuis le bandeau :

The screenshot shows a web application interface for adding a song. At the top, there's a navigation bar with links for 'Discover', 'Friends', 'Favorites', 'Add Song' (which is the active tab), 'My Playlists', 'Edit Account', and 'Logout'. Below the navigation, the title 'plylst.it' is displayed, followed by 'Add a Song'. There's a text input field labeled 'URL from YouTube/Soundcloud/etc.' and two checkboxes: 'ben2 public playlist' and 'ben2 private playlist'. A large 'Add' button is centered below the checkboxes. At the bottom left, there's a 'Back' link. In the background, there's a photograph of a DJ at a turntable.

Ici, le champ de texte invite l'utilisateur à y coller le lien du média qu'il souhaite importer. Les listes de lecture sont affichées en dessous et chacune est munie d'une case à cocher, ce qui signifie qu'une seule chanson peut être ajoutée dans plusieurs listes à la fois.



### Affichage des favoris

Lorsque l'utilisateur souhaite accéder à ses favoris, le lien "Favorites" les affiche s'il y en a :

The screenshot shows the 'Favorites' section of the plylst.it website. At the top, there are navigation links: 'Discover', 'Friends', 'Favorites', 'Add Song', 'My Playlists', 'Edit Account', and 'Logout'. Below the navigation, the title 'plylst.it' is displayed in a large, stylized font, followed by 'Favorites'. There are two sections of playlists:

- ben3 public playlist — Ben3 — 1 ❤**
  - Michael Jackson - Earth Song
  - Michael Jackson - Black Or White
- Best public playlist — Benoist — 2 ❤**
  - Michael Jackson - Earth Song
  - Elton John - Sacrifice

On the right side of the page, there is a photograph of a man (Benoist) sitting at a DJ setup, wearing a t-shirt with a jigsaw puzzle graphic.

Les listes de lecture mises en favori s'affichent de la même manière que sur la page d'accueil et permettent donc un accès direct à leur auteur et leur contenu.



### Affichage des amis

Pour gérer les amitiés, l'utilisateur doit utiliser la page accessible via le lien "Friends" du bandeau :

The screenshot shows the 'Friends' section of the plylst.it application. At the top, there are navigation links: 'Discover', 'Friends', 'Favorites', 'Add Song', 'My Playlists', 'Edit Account', and 'Logout'. Below the navigation, the 'Friends' section is titled 'Having friends allow you to see their private playlists and share yours.' It displays a list of friends: 'Ben3' with a 'Remove' link, and 'Ben4' with an 'Accept' link. A large image of a DJ at a turntable is displayed on the right side of the page.

Ici, on voit que l'utilisateur est déjà ami avec Ben3 et qu'un autre utilisateur, Ben4, lui a envoyé une invitation. Il est possible d'accepter cette dernière en cliquant sur le lien à droite de l'invitation.

Sur cette page il est aussi possible d'envoyer des invitations à être ami à d'autres utilisateurs via le lien "Add a friend".

Envoi d'une invitation

La page permettant d'envoyer une invitation à devenir ami avec un autre utilisateur se présente comme suit :

The screenshot shows a web application interface for 'plylst.it'. At the top, there are navigation links: 'Discover', 'Friends', 'Favorites', 'Add Song', 'My Playlists', 'Edit Account', and 'Logout'. Below the navigation, the title 'Add a Friend' is displayed. A list of email addresses is shown: benoist.wolleb@la-cave.ch, benoist.wolleb@la-cave.ch2, benoist.wolleb@la-cave.ch3, and benoist.wolleb@la-cave.ch5. In the background, there is a photograph of a DJ in a black t-shirt with a white puzzle-piece graphic on it, operating a turntable setup.

Il s'agit d'une simple liste affichant tous les utilisateurs du site. Le clic sur un de ces utilisateurs permet de lui envoyer une invitation.



## Accueil d'un administrateur

Lorsqu'un administrateur se connecte, sa page d'accueil possède quelques différences notables :

The screenshot shows the 'Discover' section of the plylst.it admin interface. It lists several playlists:

- ben2 public playlist — Ben2 — 0 ❤**  
Michael Jackson - They Don't Care About Us  
Michael Jackson - Thriller  
Survivor - Eye Of The Tiger  
Destroy
- ben2 private playlist — Ben2 — 1 ❤**  
Michael Jackson - They Don't Care About Us  
Michael Jackson - Thriller  
Destroy
- ben3 public playlist — Ben3 — 1 ❤**  
Michael Jackson - Earth Song  
Michael Jackson - Black Or White  
Destroy
- My other public playlist — 0 ❤**  
Rihanna - Needed Me  
MIKA - Hurts (Remix - Official Video)

In the background, there is a photograph of a DJ wearing a t-shirt with a dollar sign made of puzzle pieces, operating a DJ mixer.

Premièrement, un administrateur a accès à toutes les listes de tous les utilisateurs, même si elles sont privées et qu'il n'est pas ami avec. On aperçoit donc ici la liste "ben2 private playlist" alors qu'ils ne sont pas amis. De plus, l'administrateur a accès à un lien en dessous de chaque playlist permettant de détruire celle-ci.



### Gestion des utilisateurs

L'administrateur a accès à une page spéciale via le bandeau permettant de gérer les utilisateurs, en cliquant sur le lien "Manage users" :

Username	Admin
Benoist	true
Ben2	
Ben3	
Ben4	
Ben5	

Il s'agit d'une simple liste des utilisateurs permettant de les afficher et de supprimer leur compte. D'autre part, un clic sur le nom d'un utilisateur mènera à la liste de ses playlists.



### Listes d'un utilisateur

Lorsque l'administrateur accède à la liste des playlists d'un utilisateur, une page similaire à la page d'accueil s'affiche mais ne contenant que les listes de l'utilisateur sélectionné :

The screenshot shows a web application interface for managing playlists. At the top, there is a navigation bar with links: Discover, Friends, Favorites, Add Song, My Playlists, Exit Account, Manage users, and Logout. Below the navigation bar, the title "playlst.it" is displayed in large letters, followed by "Playlists created by Ben2". There are three listed playlists:

- ben2 public playlist — Ben2 — 0 ❤**  
Michael Jackson - They Don't Care About Us  
Michael Jackson - Thriller  
Survivor - Eye Of The Tiger  
Destroy
- ben2 private playlist — Ben2 — 1 ❤**  
Michael Jackson - They Don't Care About Us  
Michael Jackson - Thriller  
Destroy
- My other public playlist — Ben2 — 0 ❤**  
Rihanna - Needed Me  
MIKA - Hurts (Remix - Official Video)  
Destroy

On the right side of the interface, there is a photograph of a man with short hair, wearing a black t-shirt with a graphic of a jigsaw puzzle piece, sitting at a DJ setup with turntables and a mixer.

Il est aussi possible depuis cette vue d'accéder à une liste pour l'écouter ou la mettre en favori, ou de la supprimer avec le lien "Destroy" en dessous de chaque liste.