

S3.01 - Développement d'application et Gestion de Projet

Enseignant tuteur : Nathalie VALLES-PARLANGEAU

Groupe n°1



SAE du Troisième Semestre

- Pôle Systèmes et Réseaux -

Rapport Final de la création d'une infrastructure de déploiement web



Application Web Responsive de gestion de clubs sportifs amateurs

Matis Chabanat | Titouan Cocheril

Arthur Le Menn | Ivan Salle

- - -

TD1 | TP1

BUT Informatique - Parcours A

Semestre 3

- 2022 | 2023 -

Table des matières

Infrastructure de déploiement d'application web	3
1. Description Générale application Sport Track	3
2. Expression des besoins	4
3. Schéma de l'architecture globale de l'application	5
Proof of concept (PoC)	6
1. Introduction	6
2. Schéma Architecture réseau PoC :	7
3. Adressage IP	8
Adressage IP Fixe	8
Configuration DHCP	9
4. Routage et règles IPTABLES	11
Routes	11
Iptables	12
5. Configurations des différents services déployés	12
Configuration du FTP	12
Configuration Apache2	13
Configuration MySql	14
Configuration Php	16
6. Mise en place de la PoC	17
1. Création de la base de données	17
Diagramme de classes UML correspondant à notre base de données	18
2. Peuplement de la base de données	19
3. Création des pages dans le dossier var/www/	20
4. Développement du site	21
Étape 1 : config.php qui réalise la connexion à la BD.	21
Étape 2 : index.php page d'accueil du site.	21
Étape 3 : playlist.php page d'affichage des musiques d'une playlist.	22
Étape 4 : musique.php page d'affichage de toutes les musiques.	23
Étape 5 : lieuMusique.php page qui affiche la/les playlist(s) dans laquelle se trouve une musique en particulier.	24
5. Résultats de tests effectués	26
Etape 1 : Capture d'écran de la page d'accueil	26
Etape 2 : Capture d'écran Page de la playlist sélectionnée qui renvoie les musiques qu'elle contient	27
Etape 3 : Capture d'écran de la page musique.php qui permet à l'utilisateur de sélectionner la musique dont il souhaite connaître sa ou ses playlists	27
Etape 4 : Capture d'écran Page de la musique sélectionnée qui renvoie les playlists qui la contiennent	28

Infrastructure de déploiement d'application web

1. Description Générale application Sport Track

Sport Track est une application web de gestion de club sportif amateur. Elle est responsive c'est-à-dire qu'elle s'adapte pour tous types d'appareils.

La cible de notre application sont les sportifs amateurs licenciés, les dirigeants et entraîneurs de clubs sportifs amateurs.

L'application se divisera en trois grandes fonctionnalités : une partie messagerie qui permettra aux acteurs de l'application de communiquer. L'entraîneur pourra communiquer avec ses joueurs et le dirigeant de son club, les joueurs pourront parler entre eux.

La seconde partie de l'application sera la partie calendrier : les joueurs pourront consulter leurs entraînements, matchs avec le lieu et l'horaire, l'entraîneur pourra planifier ses entraînements, le dirigeant pourra gérer l'accès à son club et son terrain/salle de sport.

La dernière partie de l'application sera une partie profil : statistiques, classement, palmarès et composition équipe mais aussi des équipes adverses.

On a décidé que l'application sera destinée pour 3 sports seulement : le foot, la basket et le hand qui sont des sports collectifs avec un grand nombre de licenciés avec 1,9 millions de licenciés pour le foot, 660 000 pour le basket et 550 000 licenciés pour le hand en France

En ajoutant tous les dirigeants de club et les entraîneurs, l'application serait donc destinée à un grand nombre de personnes.

On peut ajouter qu'on laissera accès à l'application qu'aux sportifs licenciés, entraîneurs et dirigeants confirmés pour créer un réel écosystème de sportif.

L'application leur servira d'outils pour faciliter leur épanouissement au sein du club



2. Expression des besoins

Il est important de mettre en place des mesures de sécurité adéquates lors du déploiement d'une application web comme Sport Track, qui accueille de nombreux utilisateurs et de grandes quantités de données.

Un aspect important de la sécurité est l'adressage, qui consiste à attribuer une adresse unique à chaque appareil connecté à un réseau. Cela permet de s'assurer que les données sont envoyées au bon appareil et de protéger le réseau contre les attaques de piratage.

Le routage, quant à lui, consiste à acheminer les données entre les différents appareils d'un réseau. Il est important de configurer correctement les règles de routage pour s'assurer que les données atteignent leur destination de manière rapide et sécurisée. Les règles IPTables permettent de contrôler les données entrantes et sortantes d'un ordinateur ou d'un réseau.

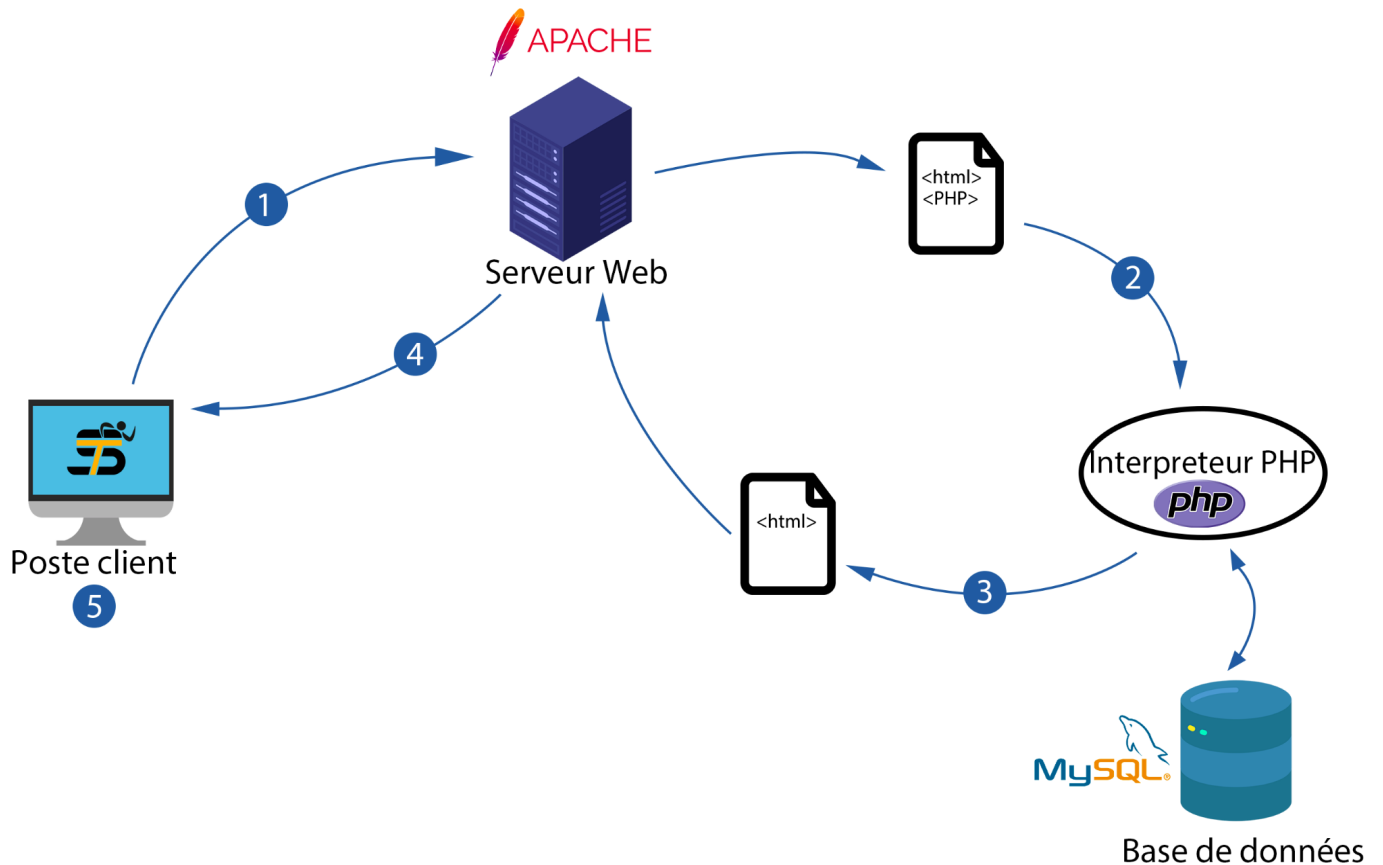
Il est important de configurer ces règles pour bloquer les connexions non autorisées et protéger le réseau contre les attaques.

Enfin, il est important de configurer les services nécessaires au déploiement de l'application, tels que le serveur web (Apache, PHP) et la base de données (MySQL), de manière à garantir leur disponibilité et leur sécurité. Cela peut inclure la mise en place de sauvegardes régulières et de mesures de sécurité supplémentaires pour protéger les données sensibles.

En résumé, il est essentiel de mettre en place une configuration de sécurité adéquate lors du déploiement de l'application Sport Track afin de protéger les données des utilisateurs et du réseau. Cela comprend l'adressage, le routage, les règles IPTables et la configuration des services nécessaires.

Dans le cadre de notre projet Sport Track, nous allons procéder à la simulation de la mise en place de l'infrastructure nécessaire au déploiement de notre application. Cette simulation vise à nous permettre de vérifier que tous les éléments sont en place et fonctionnent correctement avant de procéder au déploiement effectif de l'application.

3. Schéma de l'architecture globale de l'application



- 1) Envoi de la requête au serveur par le client.
- 2) Envoi de la page contenant du PHP et html à l'interpréteur php (ici php5) qui est connecté à la BD.
- 3) L'interpréteur renvoie au serveur la page html générée.
- 4) Envoi de la page au client par le serveur.
- 5) Lecture de la page par le navigateur du poste client.

Proof of concept (PoC)

1. Introduction

Nous allons mettre en place un proof of concept pour simuler le déploiement et l'hébergement de notre site sport track. Cela nous permettra de traiter les différents points importants qui permettront de garantir le bon fonctionnement de notre plateforme.

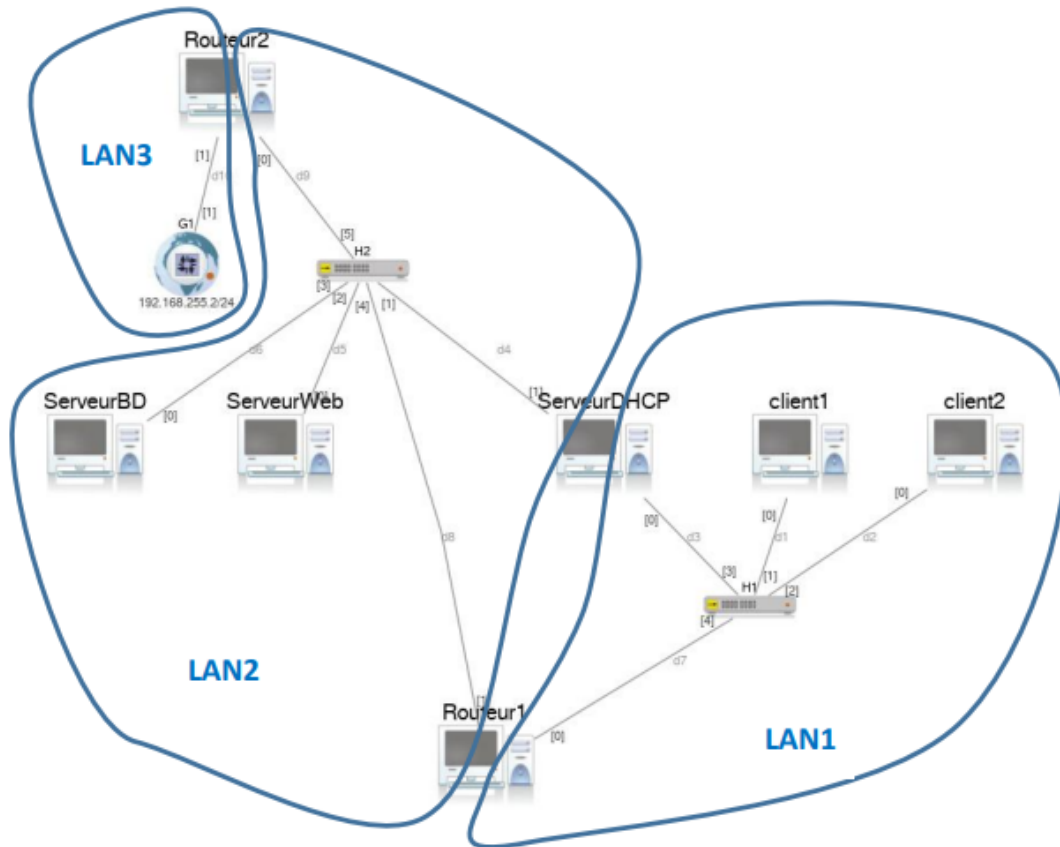
Parmi ces points, nous allons notamment travailler sur l'adressage et le routage afin de permettre à nos utilisateurs d'accéder facilement à notre site. Nous allons également nous occuper de la configuration des iptables, qui jouent un rôle important dans la sécurité de notre plateforme.

Nous allons procéder à l'installation et à la mise en place de différents services qui seront nécessaires au bon fonctionnement de notre site.

Enfin, nous allons mettre en place un site web simplifié pour tester les différentes configurations et services mis en place sur le navigateur. Nous pourrons donc nous assurer que toutes les fonctionnalités seront opérationnelles une fois notre plateforme mise en ligne.

Vous trouverez ci-dessous les différentes étapes énoncées. Les commentaires qui sont en lien avec Sport Track seront représentés par des commentaires en bleu de ce type : [/Commentaire Sport Track/](#) .

2. Schéma Architecture réseau PoC :



Adresses IP de toutes les machines du réseau de déploiement :

Nom Machine	Numéro LAN	@IP / Masque CIDR
client1	LAN 1	192.168.0.11/24
client2	LAN 1	192.168.0.12/24
DHCP eth0	LAN 1	192.168.0.1/24
Routeur1 eth0	LAN 1	192.168.0.254/24
Routeur1 eth1	LAN 2	192.168.1.254/24
DHCP eth1	LAN 2	192.168.1.1/24
ServeurBD	LAN 2	192.168.1.11/24
ServeurWeb	LAN 2	192.168.1.12/24

Routeur2 eth0	LAN 2	192.168.1.253/24
Routeur2 eth1	LAN 3	192.168.255.254/24
G1	LAN 3	192.168.255.2/24

L'architecture de notre application Sport Track sera assez similaire à celle-ci.

En effet pour réaliser l'application nous allons utiliser un serveur WEB, un serveur DHCP, un serveur de Base de Données. Des switchs seront utilisés pour relier les différentes machines. Les sous-réseaux séparent les machines connectées localement (les clients) et les serveurs (de la même façon que sur le schéma) grâce à des routeurs.

3. Adressage IP

Adressage IP Fixe

Les routeurs, les 2 serveurs ainsi que le DHCP seront adressés en IP Fixe donc :

nano /etc/rc.local et ajouter :

-> Routeur1

```
ifconfig eth0 192.168.0.254/24
ifconfig eth1 192.168.1.254/24
```

-> Routeur2

```
ifconfig eth0 192.168.1.253/24
ifconfig eth1 192.168.255.254/24
```

-> ServeurDHCP

```
ifconfig eth0 192.168.0.1/24
```



```
ifconfig eth1 192.168.1.1/24
```

-> ServeurWeb

```
ifconfig eth0 192.168.1.12/24
```

-> ServeurBD

```
ifconfig eth0 192.168.1.11/24
```

Configuration DHCP

-> ServeurDHCP

1) **nano /etc/default/isc-dhcp-server**, modifier la dernière ligne en mettant :

```
INTERFACE = 'eth0 eth1'
```

2) **nano /etc/dhcp/dhcpd.conf** et ajouter :

```
#LAN1 :  
  
subnet 192.168.0.0 netmask 255.255.255.0{  
range 192.168.0.11 192.168.0.30;  
option broadcast-address 192.168.0.255;  
option routers 192.168.0.254 ;  
default-lease-time 600 ;  
max-lease-time 7200 ;  
}
```

```
#LAN2 :  
subnet 192.168.1.0 netmask 255.255.255.0{  
range 192.168.1.20 192.168.1.30;  
option broadcast-address 192.168.1.255;  
option routers 192.168.1.253,192.168.1.254 ;  
default-lease-time 600 ;  
max-lease-time 7200 ;  
}
```

3) **nano /etc/rc.local** pour configurer le démarrage du serveur DHCP. Ajouter :

```
service isc-dhcp-server start
```

-> client1, client2

4) **nano /etc/network/interfaces** et modifier le contenu en y ajoutant les 3 lignes suivantes :

```
auto eth0  
allow-hotplug eth0  
iface eth0 inet dhcp
```

5) **nano /etc/rc.local** et y ajouter :

```
ifup eth0
```

Pour réaliser le réseau de notre application, il sera utile de savoir adresser les serveurs avec une adresse IP fixe. De même, nous allons configurer le serveur DHCP afin d'adresser les nouvelles machines pour le réseau de notre application.

4. Routage et règles IPTABLES

Routes

Afin de configurer le routage, il faut commencer par activer le routage dans les routeurs. Pour cela, **nano /etc/rc.local** et ajouter la ligne :

-> Routeur1, Routeur2

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Nous allons réaliser un routage exhaustif, cela signifie que tout le monde peut communiquer avec tout le monde. Pour cela : **nano /etc/rc.local** et ajouter

-> client1, client2

```
route add default gw 192.168.0.254
```

-> ServeurWeb, ServeurBD

```
route add -net 192.168.0.0/24 gw 192.168.1.254  
route add default gw 192.168.1.253
```

-> Routeur1

```
route add default gw 192.168.1.253
```

-> Routeur2

```
route add -net 192.168.0.0/24 gw 192.168.1.254  
route add default gw 192.168.255.2
```

Iptables

-> Serveur BD

```
iptables -P INPUT DROP  
iptables -I INPUT -s localhost -j ACCEPT  
iptables -I INPUT -s 192.168.1.12 -j ACCEPT
```

Sur le réseau de notre application, la sécurité du réseau est primordiale étant donné qu'il sera connecté à internet. Cette sécurité passe d'abord par le découpage en sous-réseaux pour séparer les infrastructures plus importantes comme les serveurs. Ensuite, les règles Iptables seront configurées de sorte à restreindre l'accès des utilisateurs de l'application.

5. Configurations des différents services déployés

Configuration du FTP

Le serveur FTP sera hébergé sur le ServeurWeb. Pour procéder à cette configuration, **nano /etc/rc.local** et ajouter la commande suivante :

-> ServeurWeb

```
ftpd -D
```

Cette commande va permettre de lancer le FTP.

Ensuite, **nano /etc/passwd** et ajouter la ligne suivante :

```
ftp::1000:1000:Anonymous:/home:/bin/bash
```

Cette ligne permet de créer un utilisateur anonyme et lui définir un répertoire par défaut.

Maintenant le FTP est mis en place. Afin de tester si le FTP est fonctionnel nous allons sur une machine qui à accès au ServeurWeb et nous dans la console nous tapons la commande ftp suivie de l'IP du ServeurWeb donc :

```
ftp 192.168.1.12
```

Cette commande va permettre à la machine de se connecter au FTP dans ServeurWeb.

Nous rentrons comme login le login de l'utilisateur précédemment créé dans ServeurWeb donc Anonymous et en mot de passe nous pouvons rentrer n'importe quel mot de passe. Une fois connecté, nous avons accès aux dossiers de ServeurWeb : le FTP est fonctionnel.

Après la réunion du 9 novembre 2022, il nous avait été conseillé de créer un utilisateur non anonyme pour renforcer la sécurité. En voici donc les étapes à suivre pour créer un utilisateur non anonyme.

-> ServeurWeb

```
# Création d'un utilisateur du nom de toto
useradd -d /home/toto -m toto

# Définition du mot de passe de l'utilisateur créé
passwd toto
```

Lors de cette dernière commande, le terminal nous demande la saisie d'un mot de passe. Vous n'avez donc plus qu'à saisir le mot de passe que vous souhaitez (dans notre cas, nous avons choisi **toto** en nom d'utilisateur et **toto** en mot de passe afin de nous simplifier la tâche et gagner en rapidité).

Configuration Apache2

Installation Apache2

```
aptitude install
```

Cette commande va permettre d'installer toutes les applications dont on a besoin dont Apache2.

Démarrage apache2

Par défaut, Apache2 se lance automatiquement au démarrage de la machine.

Sinon il existe quelques commandes

```
sudo service apache2 start //Pour lancer apache2
sudo service apache2 stop //Pour arrêter apache2
sudo service apache2 restart //Pour redémarrer apache2
sudo service apache2 reload //Pour recharger la config
```

Configuration des ports

```
/etc/apache2/ports.conf
```

On ne modifiera pas les ports d'écoutes par défaut qui sont 80 pour les requêtes HTTP et 443 pour les requêtes HTTPS

Il sera essentiel pour notre application de savoir configurer un serveur Web, avec notamment un logiciel comme Apache afin de déployer notre application web.

Configuration MySQL

1. Connexion à MySQL :

```
mysql -u root -p
```

-u : indique le nom de l'utilisateur qui veut se connecter

-p : indique au serveur de demander à l'utilisateur de saisir un mot de passe

2. Création d'une base de données :

```
CREATE DATABASE bd_sportTrack ;
```

3. Création d'un compte utilisateur de cette bd

```
CREATE USER 'utilisateur'@'%' identified by "utilisateur";  
# utilisateur peut se connecter depuis n'importe quelle  
# machine grâce au mot clé '%'
```

4. Attribution du droit de sélection (lecture) de la base de données à l'utilisateur

```
GRANT select ON bd_sportTrack.* TO 'utilisateur'@'%' ;
```

5. Mise à jour des privilèges

```
FLUSH PRIVILEGES ;
```

6. Utilisation de la bd créée précédemment

```
USE bd_sportTrack ;
```

7. Création de la table playlist

```
CREATE TABLE playlist(id INT PRIMARY KEY, titre VARCHAR(100);
```

8. Peuplement de la table playlist

```
INSERT INTO playlist VALUES (1234, "playlist1");
```

9. Modification du fichier de configuration du serveur mysql

```
nano /etc/mysql/my.cnf
```

10. Modification de la commande 'BIND' pour rendre le serveur mysql accessible depuis toutes les machines

```
bind-address = 0.0.0.0
```

11. Redémarrage du serveur mysql

```
service mysql restart
```

L'application Sport Track nécessite une base de données pour stocker les informations de tous les utilisateurs et les données des compétitions sportives. Pour cela, nous utilisons une base de données MySQL.

Configuration Php

Php est installé grâce à la commande :

```
sudo apt-get install php5
```

Et il sera configuré grâce à ces différentes étapes :

1. activation du module PHP sur le serveur apache

```
a2enmod php
```

2. redémarrage du serveur apache

```
service apache2 restart
```

3. création du fichier index.php dans le dossier var/www/

```
cd var/www/  
touch index.php
```

4. programmation d'un script php qui teste le lien entre php et mysql

```
<?php  
phpinfo();  
?>
```


5. Modification du fichier de configuration php

```
nano /etc/php5/apache2/php.ini
```

6. Affichage des erreurs php (facultatif, utile pour déboguer)

```
#Error Handling & logging  
  
display_errors = On  
  
display_startup_errors = On
```

Notre application va utiliser PHP 7.4 pour générer les pages web. Il nous faudra configurer l'interpréteur PHP de la même façon pour cacher les messages d'erreurs éventuels aux utilisateurs.

Enfin, nous accèderons à la base de données de la même façon en exécutant des requêtes php.

6. Mise en place de la PoC

Afin de mettre en place le POC, nous avons décidé de créer un petit site web dynamique comportant 4 pages :

- Une première page représentant la page d'accueil contenant le menu déroulant demandé qui redirige sur la page playlist une fois une playlist sélectionnée.
- Une deuxième page contenant la playlist sélectionnée (donc le listing des musiques contenues dans cette playlist).
- Une troisième page comportant toutes les musiques de la base de données, lorsqu'on sélectionne une musique, cela renvoie sur une quatrième page la/les playlist(s) dans laquelle elle se trouve.

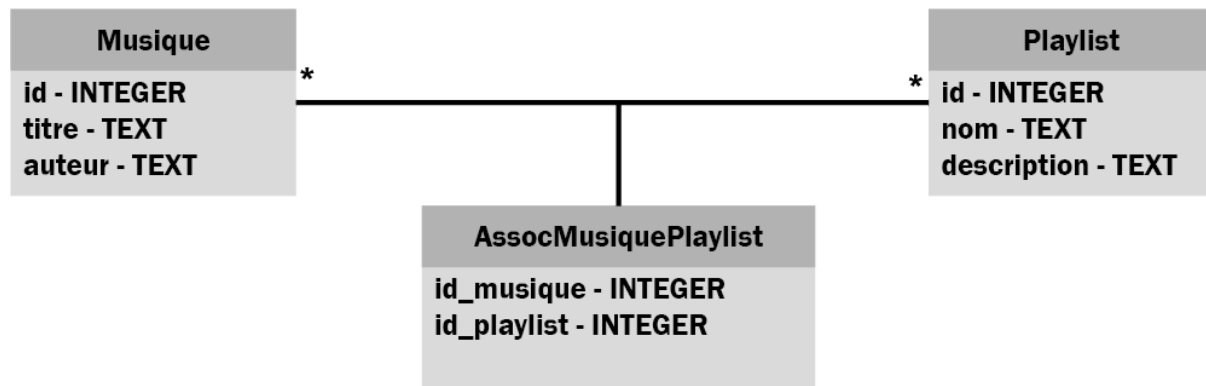
1. Création de la base de données

Notre base de données contiendra donc une table

- **Playlist** dotée des propriétés id, nom, description.
- **Musique** dotée des propriétés id, titre, auteur

- **AssocMusiquePlaylist** qui est la classe d'association entre une musique et une playlist (une musique peut se trouver dans plusieurs playlists). Cette association possède les propriétés `id_musique`, `id_playlist`. Ce sont des clés étrangères qui représentent les clés primaires des deux tables.

Diagramme de classes UML correspondant à notre base de données



Requête SQL correspondant à la création des tables

```

-- Créer la table Playlist
CREATE TABLE Playlist (
  id INTEGER PRIMARY KEY,
  nom TEXT,
  description TEXT
);

-- Créer la table Musique
CREATE TABLE Musique (
  id INTEGER PRIMARY KEY,
  titre TEXT,
  auteur TEXT
);

-- Créer l'association AssocMusiquePlaylist
CREATE TABLE AssocMusiquePlaylist (
  id_musique INTEGER,

```

```

id_playlist INTEGER,
PRIMARY KEY (id_musique, id_playlist),
FOREIGN KEY (id_musique) REFERENCES Musique(id),
FOREIGN KEY (id_playlist) REFERENCES Playlist(id)
);

```

2. Peuplement de la base de données

Requête SQL correspondant au peuplement des tables

```

-- Peupler la table Musique
INSERT INTO musique (id, titre, auteur)
VALUES (1, 'Blinding Lights', 'The Weeknd'),
       (2, 'Savage', 'Megan Thee Stallion'),
       (3, 'Dance Monkey', 'Tones and I'),
       (4, 'Señorita', 'Shawn Mendes et Camila Cabello'),
       (5, 'The Box', 'Roddy Ricch'),
       ...
       (50, 'Bad Liar', 'Imagine Dragons');

-- Peupler la table Playlist
INSERT INTO Playlist (id, nom, description)
VALUES (1, 'Pop Hits', 'The best pop songs of 2020'),
       (2, 'R&B Classics', 'Timeless R&B tunes'),
       (3, 'Rock Anthems', 'Rock classics that defined a
generation'),
       (4, 'Summer Jams', 'Feel-good summer songs'),
       (5, 'EDM Bangers', 'Electronic dance music hits');

-- Peupler l'association AssocMusiquePlaylist
INSERT INTO AssocMusiquePlaylist (id_musique, id_playlist)
VALUES (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1),
       (8, 1), (9, 1), (10, 1),
       (11, 2), (12, 2), (13, 2), (14, 2), (15, 2), (16, 2), (17,
2), (18, 2), (19, 2), (20, 2),

```

```
(21, 3), (22, 3), (23, 3), (24, 3), (25, 3), (26, 3), (27, 3), (28, 3), (29, 3), (30, 3),  
(31, 4), (32, 4), (33, 4), (34, 4), (35, 4), (36, 4), (37, 4), (38, 4), (39, 4), (40, 4),  
(41, 5), (42, 5), (43, 5), (44, 5), (45, 5), (46, 5), (47, 5), (48, 5), (49, 5), (50, 5);
```

3. Création des pages dans le dossier var/www/

```
cd var/www/  
touch config.php  
touch accueil.php  
touch playlist.php  
touch musique.php  
touch lieuMusique.php
```

Sur la page d'accueil se trouvera un message de bienvenue ainsi qu'une description du site et en pied de page nos noms pour prouver que nous sommes les auteurs. Nous pourrons bien évidemment retrouver le menu déroulant qui permettra à l'utilisateur de sélectionner sa playlist qu'il souhaite écouter.

Sur la page playlist, on retrouvera le titre de la playlist ainsi que sa description (s'il y en a une) et enfin le listing de toutes les musiques contenues dans cette playlist. Chaque musique sera représentée par son titre et l'auteur de la musique.

Sur la page musique, on retrouvera le listing de toutes les musiques contenues dans la base de données. L'utilisateur aura la possibilité de cliquer sur chaque musique afin de savoir dans quelle(s) playlist(s) elle se trouve.

Afin de pouvoir se connecter à la base de données depuis un fichier php, il faut utiliser des commandes "Mysqli". Or, nous avons plusieurs fichiers php dans notre arborescence. Afin d'éviter la duplication de code, nous avons créé un fichier

config.php qui réalisera la connexion à la BD. Enfin, ce fichier config.php sera importé sur chaque page php qui nécessite une connexion à la BD en donnant le lien de connexion.

4. Développement du site

Afin de ne pas surcharger le document de rendu, nous avons décidé de ne mettre que le code PHP. Nous avons utilisé les méthodes mysqli pour faire des requêtes sur notre base de données.

Étape 1 : config.php qui réalise la connexion à la BD.

```
<?php

$bdd= "bd_sportTrack";
$host= "192.168.1.11";
$user= "utilisateur";
$pass= "utilisateur";

function getConnection() {
    global $host, $user, $pass, $bdd;
    $link=mysqli_connect($host,$user,$pass,$bdd) or die("Impossible de
se connecter à la base de données");
    return $link;
}

?>
```

Nous avons donc créé ici une fonction qui fait le lien à la BD et qui le retourne pour les autres fichiers.

Étape 2 : index.php page d'accueil du site.

```
<?php

include 'config.php';

$nomTable= "Playlist";

/* Connection bdd */
$link = getConnection();
```

```

$query = "SELECT * FROM $nomTable";
$result= mysqli_query($link,$query);

if (mysqli_connect_errno()) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
    exit();
}

/* Affichage des données */
while ($donnees=mysqli_fetch_assoc($result)) {
    $id=$donnees["id"];
    $nom=$donnees["nom"];
    $desc=$donnees["description"];
    echo "<option>$nom</option>";
}
?>

```

Étape 3 : playlist.php page d'affichage des musiques d'une playlist.

```

<?php
    include 'config.php';

    /* Connection bdd */
    $mysqli = getConnection();

    //Récupération de la donnée playlist
    $nomPlaylist = $_POST['playlistNom'];

    //Requête pour récupérer l'id
    $query = "SELECT id FROM Playlist WHERE nom='$nomPlaylist'";
    $result = $mysqli->query($query);

    if($result === false) {
        echo "Requete Echouee";
    } else {
        $result = $result->fetch_assoc();
        $idPlaylist = $result['id'];
    }

```

```

    //Récupération de la liste des chanson
    $query = "SELECT Musique.titre, Musique.auteur FROM
AssocMusiquePlaylist JOIN Musique ON AssocMusiquePlaylist.id_musique =
Musique.id WHERE id_playlist=$idPlaylist";
    $result = $mysqli->query($query);
    // Fermeture de la connexion
    $mysqli->close();
?>

#code HTML entre les deux balises php

<?php
    $i = 1;
    while ($row = $result->fetch_assoc()) {
        echo "<p>".$i.". ".$row['titre']." - ".$row['auteur']."</p>";
        $i++;
    }
?>

```

Étape 4 : musique.php page d’affichage de toutes les musiques.

```

<?php
    include 'config.php';

    /* Connection bdd */
    $link = getConnection();

    $nomTable= "Musique";

    $query = "SELECT * FROM $nomTable";
    $result= mysqli_query($link,$query);

    if (mysqli_connect_errno()) {
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
        exit();
    }

    /* Affichage des données */
    while ($donnees=mysqli_fetch_assoc($result)) {
        $id=$donnees["id"];
    }

```

```

        $titre=$donnees["titre"];
        $auteur=$donnees["auteur"];
        echo "<a href='lieuMusique.php?id=$id'><p>$id. $titre -
$auteur</p></a><br>";
    }
?>

```

Étape 5 : lieuMusique.php page qui affiche la/les playlist(s) dans laquelle se trouve une musique en particulier.

```

<?php
    include 'config.php';

    // Récupération de l'id de la musique choisie
    $idMusique = $_GET['id'];

    /* Connection bdd */
    $link = getConnection();

    //Requete numéro 1 qui récupère les informations de la musique
choisie (pour l'affichage)
    $recupMusic="SELECT titre,auteur FROM Musique WHERE id=$idMusique";
    $resultatRecup=mysqli_query($link,$recupMusic);

    if (mysqli_connect_errno()) {
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
        exit();
    }

    /* Affichage de la musique choisie */
    while ($donnees=mysqli_fetch_assoc($result)) {
        $titre=$donnees["titre"];
        $auteur=$donnees["auteur"];
        echo "<p>La musique $titre de $auteur que vous avez sélectionné
se trouve dans :</p>";
    }

    //Requete numero 2 qui affiche la ou les playlists qui possèdent la
musique choisie

```



```

$query = "SELECT nom,Playlist.description FROM Musique JOIN
AssocMusiquePlaylist ON Musique.id=AssocMusiquePlaylist.id_musique JOIN
PlayList ON AssocMusiquePlaylist.id_playliste=Playlist.id WHERE
Musique.id = $idMusique";

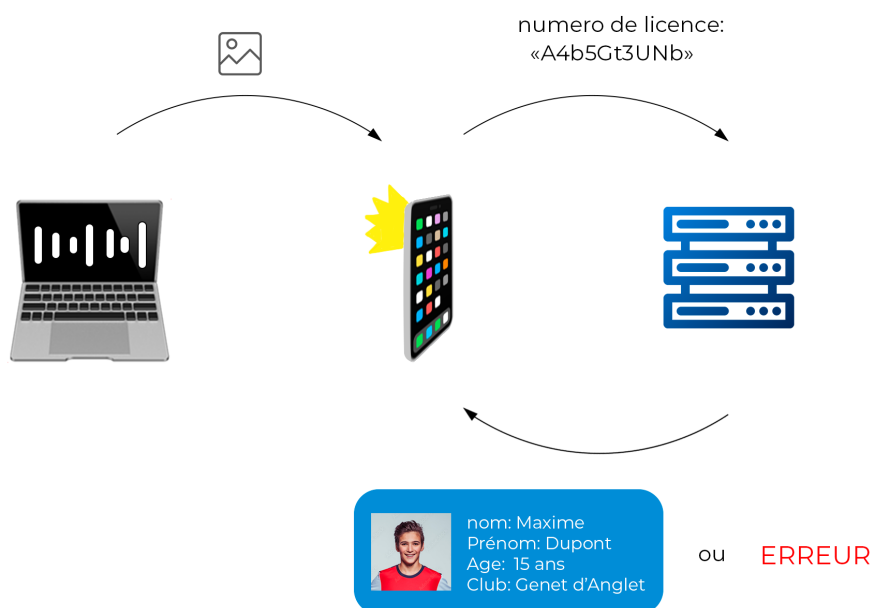
$result= mysqli_query($link,$query);

if (mysqli_connect_errno()) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
    exit();
}

/* Affichage des données */
while ($donnees=mysqli_fetch_assoc($result)) {
    $nom=$donnees["nom"];
    $desc=$donnees["description"];
    echo "<p>+ <strong>$nom</strong> - $desc</p><br>";
}
?>

```

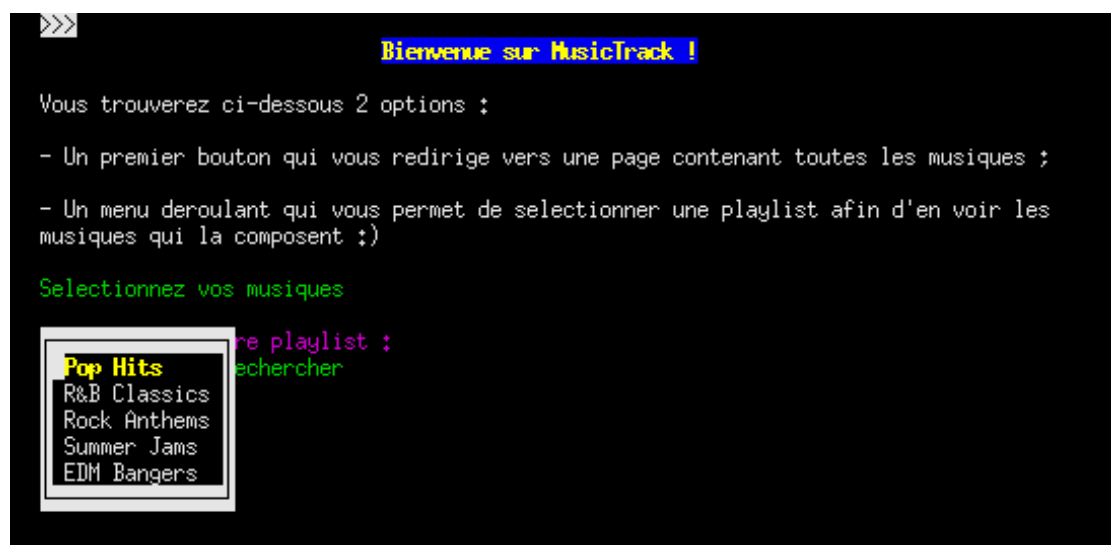
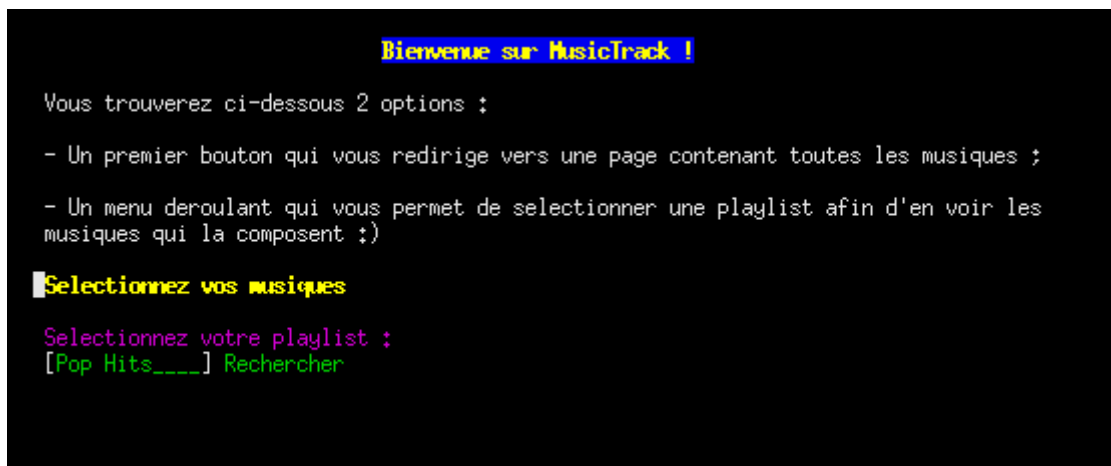
La PoC réalisée est assez similaire à ce que nous comptons réaliser sur SportTrack. Par exemple, l'une des fonctionnalités proposées est un scanner qui prendra en photo un code, le convertira en chaîne de caractères qui correspond à la licence d'un joueur. Le serveur exécutera des requêtes à la base de données pour afficher des informations sur un joueur qui sont stockées dans la base de données en fonction de cette licence.



Le schéma ci-dessus montre le fonctionnement du scanner. Cette fonctionnalité nécessite des échanges entre le Serveur et l'utilisateur de l'application, d'où l'importance de bien configurer les services proposés.

5. Résultats de tests effectués

Etape 1 : Capture d'écran de la page d'accueil



Cette deuxième capture représente l'action de rentrer dans la liste déroulante pour y choisir la playlist que l'on souhaite.

Etape 2 : Capture d'écran Page de la playlist sélectionnée qui renvoie les musiques qu'elle contient

```
<<<>>>
                               Bienvenue sur MusicTrack !

Voici les musiques de la playlist que vous avez selectionne !

Playlist Pop Hits - The best pop songs of 2020

1. Blinding Lights - The Weeknd
2. Savage - Megan Thee Stallion
3. Dance Monkey - Tones and I
4. Senorita - Shawn Mendes et Camila Cabello
5. The Box - Roddy Ricch
6. Roses Imanbek Remix - SAINT JHN
7. Say So - Doja Cat
8. No Guidance - Drake feat. Chris Brown
9. Circles - Post Malone
10. Good As Hell - Lizzo
```

Etape 3 : Capture d'écran de la page musique.php qui permet à l'utilisateur de sélectionner la musique dont il souhaite connaître sa ou ses playlists

```
<<<
                               Bienvenue sur MusicTrack !
                               (p1 of 3)

Vous trouverez ci-dessous toutes les musiques disponibles !

Selectionnez une musique pour voir dans quelle(s) playlist(s) elle apparait :)

1. Blinding Lights - The Weeknd
2. Savage - Megan Thee Stallion
3. Dance Monkey - Tones and I
4. Senorita - Shawn Mendes et Camila Cabello
5. The Box - Roddy Ricch
6. Roses Imanbek Remix - SAINT JHN
7. Say So - Doja Cat
8. No Guidance - Drake feat. Chris Brown
9. Circles - Post Malone
10. Good As Hell - Lizzo
11. High Hopes - Panic! At The Disco
12. Truth Hurts - Lizzo
13. Someone You Loved - Lewis Capaldi
14. Sunflower - Post Malone & Swae Lee
15. 10,000 Hours - Justin Bieber feat. Dan + Shay
16. Dont Start Now - Dua Lipa
17. Adore You - Harry Styles
```

Etape 4 : Capture d'écran Page de la musique sélectionnée qui renvoie les playlists qui la contiennent

