Manuel d’utilisation de l’application :  
VS Spotify

# Description

Dans le cadre du cours programmation distribuée (JavaSocket) de la formation Bachelor of Science en informatique de gestion de la HES-SO Valais Wallis, 4ème semestre, nous avons dû réaliser une application Spotify fonctionnant en Peer to Peer.

Le but est d’avoir un serveur central (Scanner) sur un réseau privé, auquel tous les utilisateurs (Users) peuvent accéder via son ip & port connus.

Une fois que un utilisateur se connecte au scanner, les informations de l’utilisateur y sont envoyées.

Les autres utilisateurs se connectant au scanner peuvent demander la liste des utilisateurs connectés eux aussi au scanner et ainsi obtenir leurs informations tel que ip, port, musiques à dispositions.

Ainsi, je peux me connecter à n’importe que autre utilisateur et streamer chez moi les musiques qu’il me met à disposition, et n’importe qui peut se connecter à mon PC pour écouter les musiques que je mets à disposition.

Durant le cours, nous avons appris à utiliser les sockets sur java, et le but de ce projet est donc d’exploiter un maximum la communication entre des machines différentes via des sockets.   
De plus, du multithreading a du être implémenté afin de subvenir à nos besoins pour pouvoir faire toutes les tâches nécessaires en parallèle.

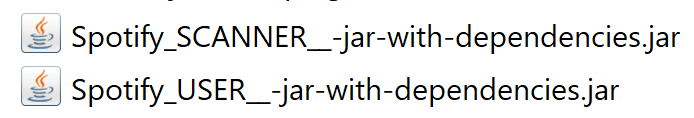
# Configuration requise

* Variable d’environnement appelé « **VSSPOTIFY** »
  + Peut indiquer le chemin d’accès au bureau par exemple.  
    Indique le chemin d’accès au répertoire ou l’on veut créer nos dossiers contenants les fichiers que l’on met à disposition des autres utilisateurs & les fichiers de LOG qui permettront d’avoir un historique des activités.
* Java version jdk-17.0.1 **Minimum**. Ou plus récent.

# Installation de l’application

## Option 1 :

Dans le rendu du projet en « .zip », dans le dossier ApplicationRendu , il vous sera fourni deux applications nommées :

* Scanner.jar
* User.jar

Une image contenant texte

Description générée automatiquementUne image contenant texte, équipement électronique, capture d’écran

Description générée automatiquementCes 2 applications ont chacun un fichier Scanner.bat & User.bat dans le même dossier. Il suffit de créer un raccourci de ces 2 fichier « .bat » et de placer ce raccourci ou bon vous semble afin de pouvoir lancer l’application depuis n’importe ou.

## Option 2 :

Le projet se trouve sous le gitLab (LIEN ICI), peut être clôné et packagé grâce à maven. Les 2 JAR se trouvent dans le dossier target.

# Manuel Résumé

## Info pratique

L’IP sert à identifier les utilisateurs, il sera unique à chaque machine.

Scanner : Sera toujours en écoute sur le port 45'000.

User : sera toujours en écoute sur le port 50'000.

## Résumé Scanner

Le scanner est en écoute constante de nouvelles connexions.

Une image contenant texte

Description générée automatiquement

Il enregistre les informations du client qui se connecte.

Une image contenant texte, capture d’écran, écran

Description générée automatiquement

## Résumé Client

On se connecte au scanner, puis le menu principal s’offre à nous.

Une image contenant texte

Description générée automatiquement

On peut afficher la liste des users connectés au scanner

Une image contenant texte

Description générée automatiquement

On peut mettre à jour nos informations chez le scanner (si on enlève ou rajoute des musiques de notre dossier)

Une image contenant texte

Description générée automatiquement

On peut se connecter à un autre utilisateur et jouer de la musique.Une image contenant texte

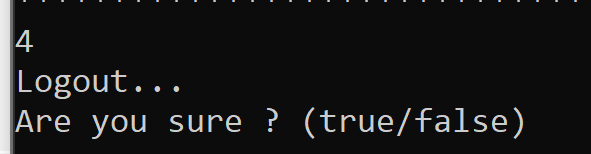
Description générée automatiquement

On peut contrôler la musique

Une image contenant texte

Description générée automatiquement

On peut se déconnecter



# Manuel Détaillé

## Scanner

### Descriptif

Le scanner sert de serveur central à un réseau. Tous les utilisateurs s’y connectent pour s’enregistrer.

Le scanner est en écoute constante de nouveaux users, et dès qu’il reçoit un demande de connexion, il délègue le travail de traitement du user à un UserHandler afin que le scanner continue son écoute sur le réseau pour de nouvelles connexions.

Chaque UserHandler est crée dans un nouveau thread.

Le scanner permet de :

* Ecouter les nouvelles tentatives de connexions
* D’envoyer mes informations en tant que client
* Récupérer la liste de tous les clients actuellement connectées, avec leurs informations tel que
  + Ip client
  + Port du serveur client
  + Musiques à disposition
* On peut se déconnecter du scanner et ainsi disparaitre de la liste des utilisateurs connectés.
* Générer des LOGS sur les activités.

## User

### Descriptif

Le user sera l’interface utilisateur qui permet à chacun de pouvoir interagir avec le système et de pouvoir écouter de la musique en streaming.

Le user aura 3 threads principaux.

* Thread User, il démarre l’application et vérifie que ses 2 sous-threads soient en vie.
* Thread Client, ce sera l’interface utilisateur. C’est le client qui se connecte au scanner et qui communiquera avec le UserHandler du scanner.
  + Quand je me connecte à un serveur d’un autre client, un nouveau sous-thread est crée pour le AudioPlayer. Ainsi le thread principal a les commandes PLAY PAUSE STOP sur la musique, et la musique écouté en stréaming est geré par le sous thread AudioPlayer.
* Thread Serveur, ce sera un thread qui écoute constamment si un autre utilisateur cherche à se connecter sur ma machine afin d’écouter une de mes musiques. Si un client se connecte à moi, le serveur délègue a gestion du nouveau client à un ClientHandler qui lui enverra la musique demandé depuis PC. Un ClientHandler est crée dans un nouveau thread pour chaque nouveau client qui se connecte, ainsi plusieurs clients peuvent écouter la même musique de mon PC.

L’application peut :

* Se connecter à un scanner
  + Lui envoyer ses informations en tant que client
    - Ip
    - Port de son serveur sur lequel il distribue ses musiques
    - Liste des musiques mis à disposition
  + Lui demander la liste des utilisateurs connectés au scanner avec leurs informations.
* Afficher la liste des utilisateurs connectés au scanner
* Envoyer une MAJ des musiques que je dispose
  + On regarde le contenu de mon dossier & on sérialise les informations pour les envoyer au scanner qui mettra à jour nos informations.
* Se déconnecter du scanner.
* Ecouter une musique
  + En se connectant à un autre user
  + Et commander la musique.
* Générer des logs sur l’activité du user.

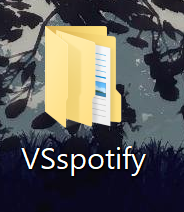
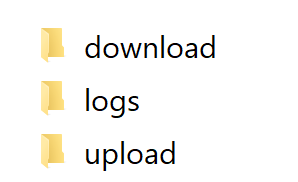
# Scénario d’utilisation

Pour commencer, lancer l’application du scanner. Double cliquer sur le raccourci en « .bat ».

Une image contenant texte

Description générée automatiquement

Figure 1: SCANNER

Que vous lanciez l’application USER ou SCANNER, un dossier va être crée à l’emplacement ou vous avez défini votre variable d’environnement.

Le scanner est maintenant prêt à être utilisé par les utilisateurs.

## Démarrer en tant qu’utilisateur

Double cliquer sur le raccourci de l’application pour le User afin de lancer le programme.

Une image contenant texte

Description générée automatiquement

Figure 2: USER A

Quand on lance l’application du user (USER A & B seront nos 2 personnages durant ce scénario), on voit que on lance 2 threads, un pour l’interface utilisateur qui sera le client, et l’autre qui sera le serveur.

Maintenant, on doit entrer les informations IP & PORT du scanner pour s’y connecter. On part du principe que l’on connait ces informations. (La personne Utilisant le scanner peut lancer la commande « ipconfig » sur Windows pour connaitre son ip.)

Une image contenant texte

Description générée automatiquement

Figure 3: USER A

On voit maintenant l’écran principal du User s’afficher, c’est le MENU principal.

Une image contenant texte

Description générée automatiquement

Figure 4: USER A

Lors de notre connexion, on peut voir que le scanner a bien obtenu nos informations lors de la connexion.

Une image contenant texte, capture d’écran, écran

Description générée automatiquement

Figure 5: SCANNER

Maintenant que nous sommes sur le menu principal, voyons voir l’option 1, voir les utilisateurs & leurs musiques mis à dispositions sur leurs serveurs.

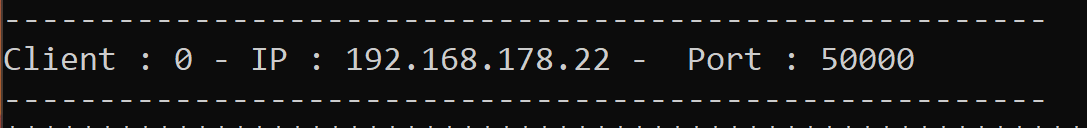
Une image contenant texte

Description générée automatiquement

Figure 6: USER A

On voit que en tapant l’option 1, il nous a affiché la liste des clients connectés. Actuellement, je suis le seul utilisateur connecté, on y voit mon adresse IP, mon PORT sur lequelle j’accepte que les clients se connectent moi, et je n’ai mis aucune musique à disposition.

Ajoutons des fichiers dans le dossier « upload » afin de mettre des musiques à disposition, et on re-affiche la liste des users depuis l’app.

Une image contenant texte

Description générée automatiquement

Figure 7: USER A

Figure 8: USER A

On voit que les musiques sont dans le dossier, mais pas encore mis à jour sur la liste du scanner.

On va donc mettre à jour nos fichiers sur la liste du scanner, on tape l’option 3.

Une image contenant texte

Description générée automatiquement

Figure 9: USER A

Une image contenant texte

Description générée automatiquement

Figure 10: USER A

On voit maintenant que la liste s’est mis à jour sur le scanner !

Maintenant, connectons un nouvel utilisateur au scanner :

Une image contenant texte

Description générée automatiquement

Figure 11: USER B

Une image contenant texte

Description générée automatiquement

Figure 12: USER A

Maintenant que un autre user s’est connecté, on peut voir la liste des users.

Le User B propose 2 musiques, et son ip est affiché si je veux me connecter à lui.

Connectons nous à lui pour lancer une musique, l’option 2 :

Une image contenant texte

Description générée automatiquement

Figure 13: USER A

On entre les informations pour se connecter à l’autre utilisateur

Une image contenant texte

Description générée automatiquement

Figure 14: USER A

On donne le nom de la musique qui nous intéresse

Une image contenant texte

Description générée automatiquement

Figure 15: USER A

Une image contenant texte

Description générée automatiquementEt on voit que l’on peut contrôler la musique alors qu’elle joue en même temps.

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Je décide de me déconnecter, l’autre utilisateur verra que j’e disparais de la liste des users connectés. Option 4.

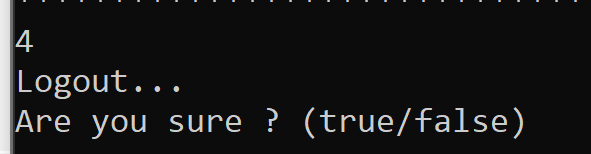


Figure 16: USER A

Une fois déconnecté, la fenêtre se ferme.

Une image contenant texte

Description générée automatiquement

Figure 17: USER B

Maintenant, l’User B ne voit plus que lui-même dans la liste des utilisateurs connectés.