

# Rapport du projet SE et SD



A L'ATTENTION DE Mme BRUNSCHWIG

Realisé par

YAHY LOTFI

DOHA ELACHAB

# Problématique :

Dans ce projet, nous avons été amenés à implémenter un bar en utilisant des systèmes d'exploitation et des systèmes distribués. Nous devons être en mesure de suivre la simulation du bar via une interface, commençant par les clients et allant jusqu'aux fournisseurs.

## Réalisation :

Ce projet contient plusieurs parties, et chaque partie devra être réalisée en respectant certaines conditions.

Nous avons choisi de commencer par la partie fournisseur et le processus de commande qui devront être réalisés en JAVA, et se connecter via la technologie de communication JavaRMI. Nous avons donc téléchargé les fichiers sur elearn (fournisseur, bière et lbère) afin de créer le serveur, une fois le serveur créé et publié, on a créé le processus de commande qui va se connecter sur ce serveur afin de pouvoir avoir accès à toutes les fonctions de l'objet Bière, même à distance.

Sur le processus de commande nous devons permettre au barman de passer des commandes une fois une alerte d'un fût de bière vide est arrivé. Cette alerte est envoyée par le processus contrôle, pour ce, on communique via le protocole UDP.

Une fois la socket créée sur le processus commande, nous attendons un message de la part du processus de contrôle, qui vérifie et s'assure de l'intégrité du bar. Une fois cette alerte est arrivée, on affiche au barman sur la console le choix de pouvoir commander et de voir la liste.

```
try {}  
    DatagramSocket serverSocket = new DatagramSocket(port);  
    byte[] buffer = new byte[1000];  
    byte[] receiveBuffer = new byte[BUFFER_SIZE];  
    byte[] sendBuffer = new byte[1024];  
    DatagramPacket packet = new DatagramPacket(buffer, buffer.length);  
  
    System.out.println("Server listening on port " + port);  
  
    while (true) {  
        DatagramPacket receivePacket = new DatagramPacket(receiveBuffer, receiveBuffer.length);  
        System.out.println("Waiting for message...");  
        serverSocket.receive(receivePacket);  
  
        String message = new String(receivePacket.getData(), 0, receivePacket.getLength());  
        System.out.println("Message received: " + message);  
    }
```

- le processus contrôle :

Comme indiqué dans le sujet, le barman contient le processus contrôle, main, communication et commande. Nous avons donc créé un seul fichier barman qui contient le processus main ainsi que le processus de contrôle. En effet, le processus de contrôle n'est que le fils du processus main qui vérifie à chaque fois si un fut de bière est vide ou non dans la mémoire partagée. Pour la mémoire partagée, nous avons déclaré une structure qui représente un fut de bière avec sa couleur etc... ensuite nous l'avons rattaché à une mémoire partagée qui est déclarée au début du main et qui est supprimé à la fin de ce fichier.

#### Processus de communication et le client :

Ensuite nous avons le client qui est représenté par un fichier client et le processus de communication qui est représenté par un fichier communication. Ces deux communiquent via TCP, le processus de communication envoie au main les informations obtenues via un Pipe.