

FICHE TECHNIQUE

Description de Whatsup

Diffusion non restreinte

DESCRIPTION

Ce document traite du projet “Whatsup” qui est une salle de discussions (ChatRoom) programmé en Java. Ce projet a été réalisé dans le cadre du cours de génie logiciel. Voici le lien du projet sur GitHub : https://github.com/ork14750/Projet_GL_Chat_Room . Les instructions pour l'utilisation de l'application peuvent se trouve ci-dessous ou sur le « read me » dans le GitHub.

TABLE DES MATIÈRES

1.	ARCHITECTURE	3
2.	SERVEUR	3
2.1.	Lancer le GUI serveur.....	4
2.2.	Configurer une base de donnée	4
2.3.	Configurer le port d'écoute et lancer le serveur	5
2.4.	Le nombre de clients connectés.....	5
2.5.	Les logs.....	6
3.	CLIENT	6
3.1.	Lancer le GUI client	7
3.2.	Joindre le serveur	7
3.3.	S'authentifier.....	8
3.3.1	S'inscrire	8
3.3.2	Se connecter.....	8
3.4.	Discussion	8
3.4.1	Envoyer un message à un utilisateur connecté.....	9
3.4.2	Envoyer un message à tous les utilisateurs connectés.....	9
3.4.3	Envoyer un message à un groupe	9
3.5.	Partage de fichier	9
3.5.1	Envoyer un fichier	9
3.5.2	Télécharger un fichier.....	9
3.6.	Groupe.....	10
3.6.1	Créer groupe.....	10
4.	BASE DE DONNEE	10
5.	DESIGN PATTERN	11
5.1.	Singleton.....	11
6.	LES PRINCIPES SOLID	11
7.	TESTS	11

1. ARCHITECTURE

L'architecture de whatsapp a été un choix décisif dans la conception de celui-ci, au vu de la problématique et l'objectif de l'application, nous avons décidé de nous tourner vers une architecture en deux grandes couches

- Une Couche CLIENT/SERVEUR
- Une Pseudo-MVC

Ce choix nous a permis de séparer l'application en deux projets indépendants, Un projet CLIENT et un projet Serveur, et chaque'un de ses projets est basé sur une architecture Pseudo-MVC.

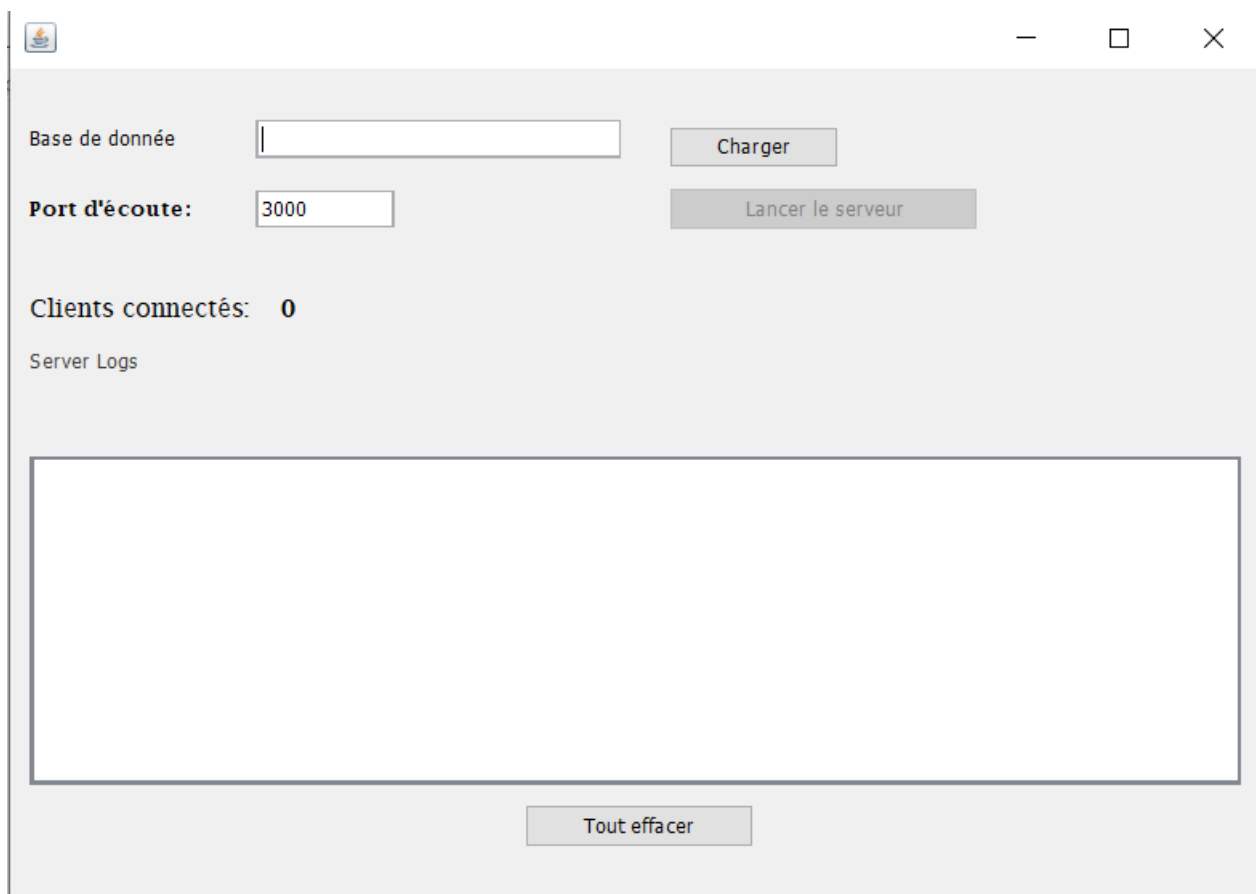
L'interaction entre ces deux parties se fait via des appels réseaux.

Ce choix d'architecture était selon nous l'une des meilleurs façon d'implémenter whatsapp, on pouvait facilement se répartir les tâches, la maintenance applicative était beaucoup plus simple.

L'un des points fort de ce choix est le fait de pouvoir partager seulement la partie CLIENT de notre application aux utilisateurs.

2. SERVEUR

Le serveur est le coeur de notre application. C'est lui qui va gérer la logique des échanges entre différents utilisateurs et groupes d'utilisateurs.



2.1. Lancer le GUI serveur

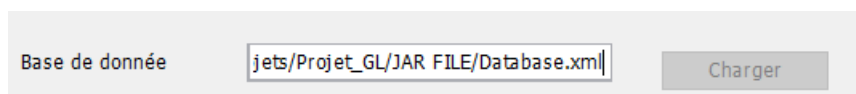
Exécuter l'application "Server.jar" contenu dans le dossier « JAR FILE » que vous avez préalablement télécharger.

Une fois exécuté, l'interface d'administration du serveur s'ouvre.

2.2. Configurer une base de donnée

Lorsque vous êtes sur le GUI SERVEUR, la première étape consiste à configurer une base de donnée de type XML.

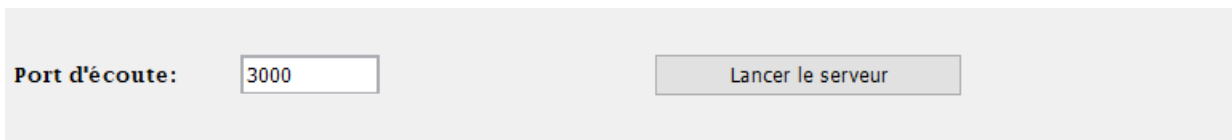
- Cliquer sur le bouton charger,
- Une fois le FileManager apparu, rendez vous dans le dossier « JAR FILE », puis choisissez le fichier Database.xml



La base sera automatiquement prise en compte, vous pourrez passer à l'étape suivante.

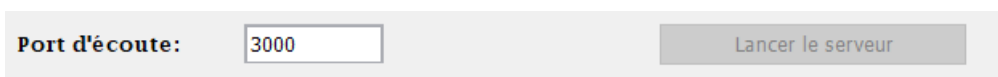
2.3. Configurer le port d'écoute et lancer le serveur

Le port d'écoute du serveur, par défaut est "3000", La modification de ce dernier est possible au niveau du champs "Port d'écoute".



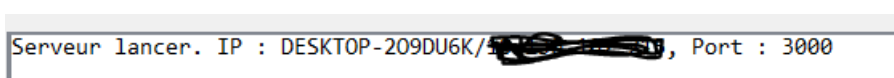
Port d'écoute: Lancer le serveur

Une fois le port configuré vous pouvez cliquer sur le bouton "Lancer le serveur"



Port d'écoute: Lancer le serveur

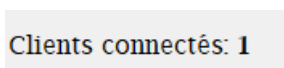
Dès lors que le serveur est lancé, un log apparaîtra dans la partie "Server logs" en bas de la page, vous indiquant que le serveur est lancé et prêt à accueillir des clients.



```
Serveur lancer. IP : DESKTOP-209DU6K/5.15.1.10, Port : 3000
```

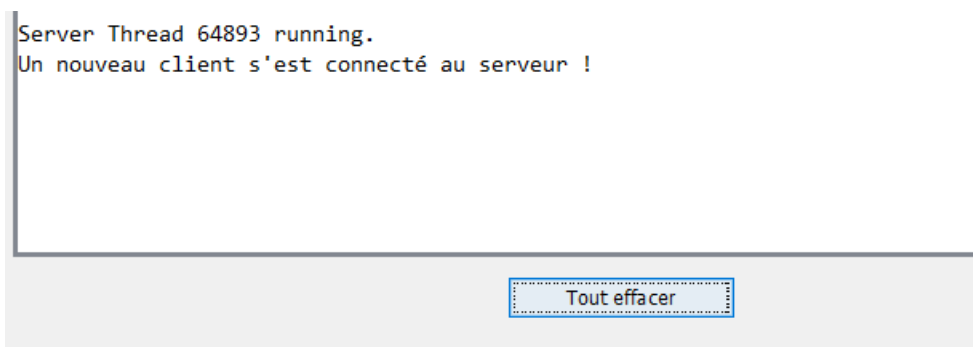
2.4. Le nombre de clients connectés

Vous avez la possibilité de voir le nombre de clients connectés au niveau de "Clients connectés" au milieu droit de la page.



Clients connectés: 1

Lorsqu'un nouveau client se connecte un log apparait dans "Server logs"



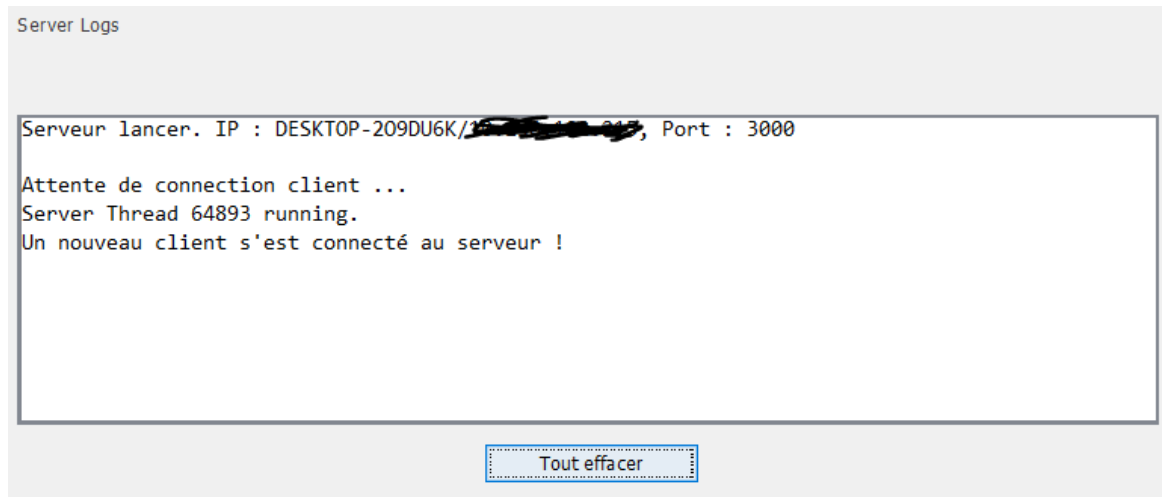
```
Server Thread 64893 running.  
Un nouveau client s'est connecté au serveur !
```

Tout effacer

2.5. Les logs

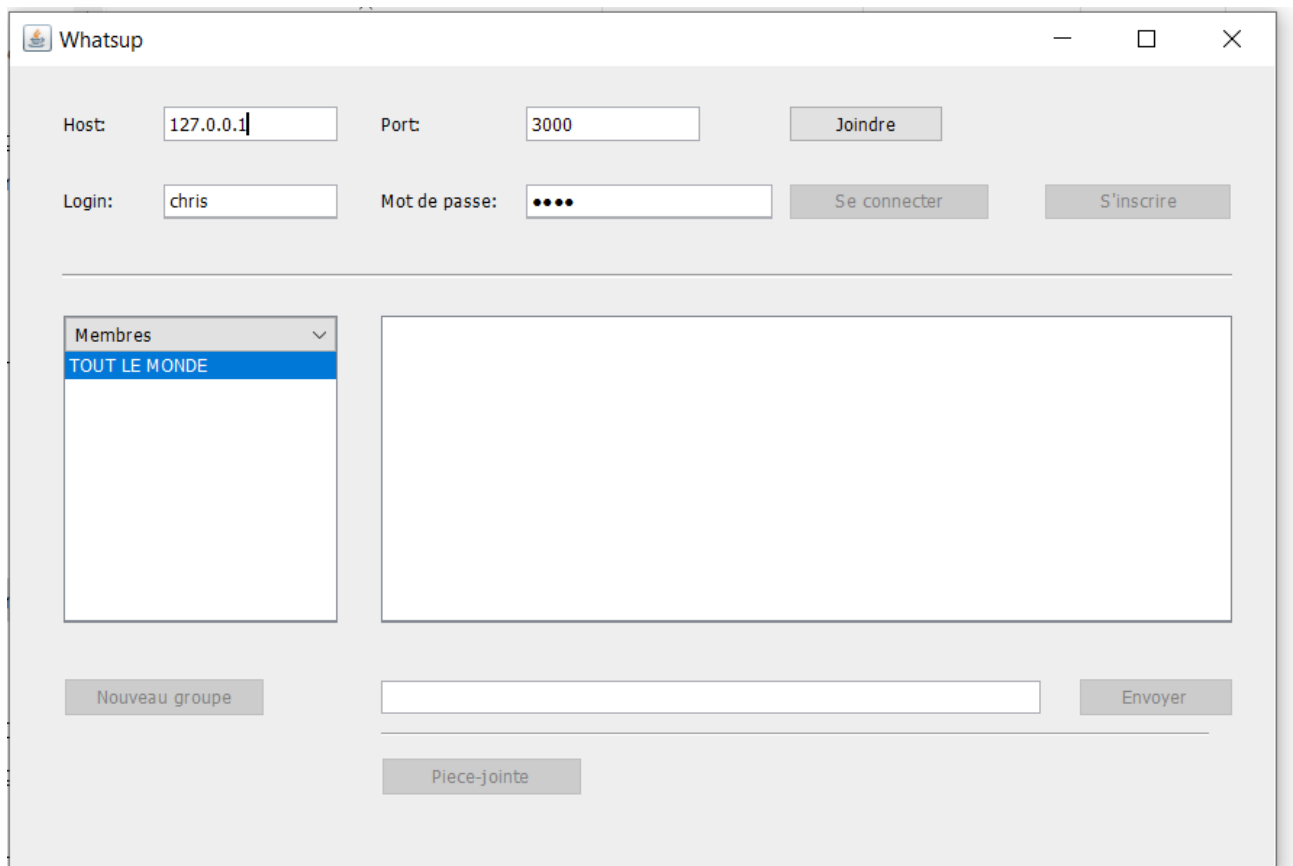
Dans la partie en bas de notre application se trouve le logger, dans lequel vous êtes notifié des différents événements qui se déroulent au sein du serveur :

- Lancement du serveur
- Lancement d'un Thread
- Connexion d'un client
- ...



3. CLIENT

La partie client est une l'application indépendante dédiée aux utilisateurs. C'est elle qui permettra à un utilisateur d'échanger avec d'autres, en communiquant avec le serveur via des appels réseaux.



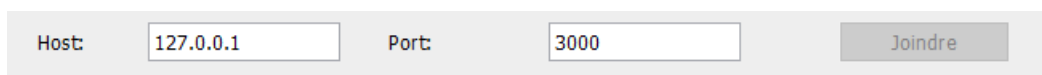
3.1. Lancer le GUI client

Une fois que le serveur est lancé :

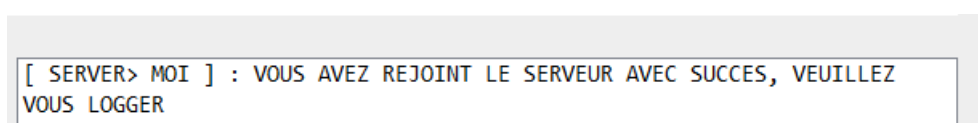
- Exécuter l'application "Client.jar" contenu dans le dossier « JAR FILE » que vous avez préalablement télécharger.
- Une fois executé, l'interface Client s'ouvrira.

3.2. Joindre le serveur

Pour se connecter au serveur, il faut entrer l'host du serveur ainsi que le port au niveau des champs "Host" et "Port" situés en haut de la fenêtre. Puis cliquer sur joindre.

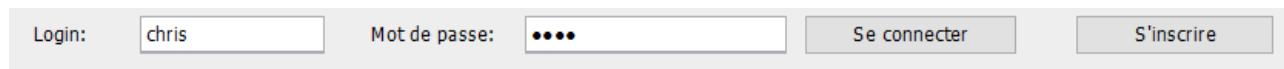


Un log apparaîtra si le serveur a été atteint avec succès, dans le logger en Bas à gauche.



3.3. S'authentifier

Une fois que le serveur rejoint, vous pouvez vous authentifier en remplissant “Login” et “Mot de passe”, en haut de la fenêtre.



Login: Mot de passe:

Vous aurez alors deux actions disponibles “Se connecter” et “S’inscrire”.

3.3.1 S'inscrire


Lorsque vous cliquez sur le bouton “s’inscrire”, un nouveau compte sera créé si le login est disponible. Vous pourrez alors commencer à discuter.

3.3.2 Se connecter

Si vous avez déjà un compte cliquer sur “Se connecter”. Si les identifiants sont correctes, vous pourrez alors commencer à discuter.

3.4. Discussion

Vous avez la possibilité d’échanger en entrant votre message dans le champs situé en bas de la fenêtre et l’envoyer à un destinataire en cliquant sur le bouton “Envoyer”

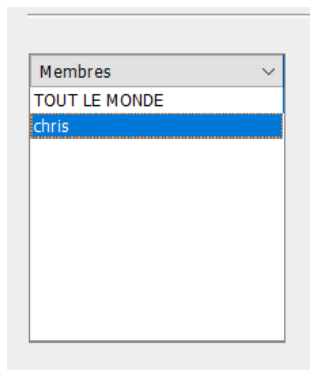


Une fois le message envoyé vous verrez la trace sur la console de log.

```
[ SERVER> MOI ] : VOUS AVEZ REJOINT LE SERVEUR AVEC SUCCES, VEUILLEZ VOUS LOGGER
[ SERVER> MOI ] : CONNECTION REUSSITE, VOUS POUVEZ COMMENCER A CHATTER
[ SERVER> MOI ] : Votre groupe a été créé : test-1
[ chris> karim ] : salut c'est chris
```

3.4.1 Envoyer un message à un utilisateur connecté

Pour envoyer un message à un autre utilisateur, sélectionner “Membres” au niveau de la liste déroulante à gauche de la fenêtre puis cliquer sur le membre désiré et envoyer votre message.

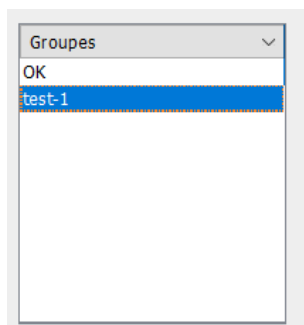


3.4.2 Envoyer un message à tous les utilisateurs connectés

Sélectionner “TOUT LE MONDE” dans la liste des membres puis envoyer votre message.

3.4.3 Envoyer un message à un groupe

Pour envoyer un message à un groupe, sélectionner “Groupes” au niveau de la liste déroulante à gauche de la fenêtre, puis cliquer sur le groupe désiré et envoyer votre message.

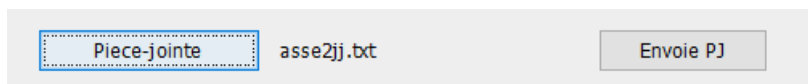


3.5. Partage de fichier

3.5.1 Envoyer un fichier

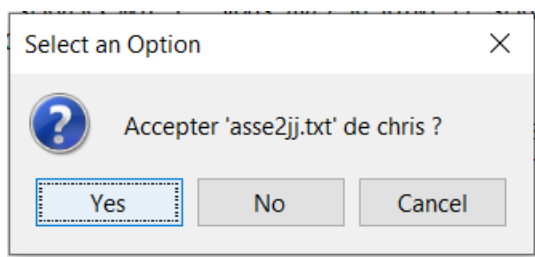
Vous avez la possibilité de partager des fichiers seulement entre membres :

- Cliquer sur le membre destinataire, puis sur le bouton “pièce jointe” en bas de la fenêtre
- Sélectionner un fichier, un bouton “envoie PJ” apparaîtra, cliquer dessus pour transférer le fichier.



3.5.2 Télécharger un fichier

Lorsque vous recevez un fichier d'un autre membre, une pop-up apparaît sur votre fenêtre, qui vous demande d'accepter ou non le Fichier.



Si vous cliquez sur “Yes”, l’explorateur de fichier s’ouvrira :

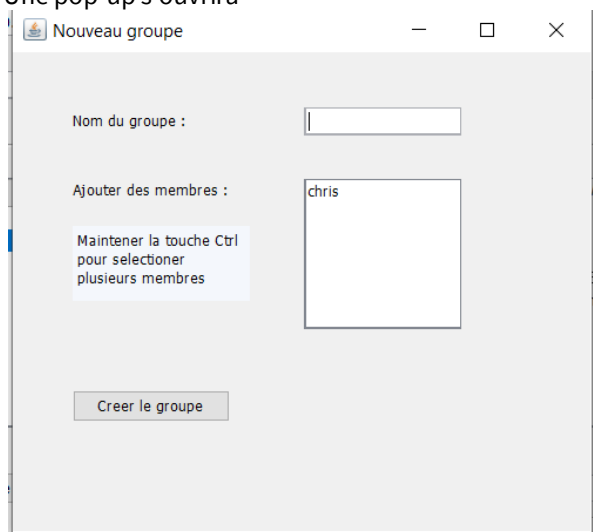
- sélectionner le dossier dans lequel vous souhaitez le sauvegarder
- cliquer sur “SAVE” pour le télécharger.

3.6. Groupe

Un groupe est un ensemble d’utilisateurs, qui peuvent communiquer seulement entre eux.

3.6.1 Créer groupe

- Cliquer sur le bouton “Nouveau groupe” en bas à gauche de la fenêtre
- Une pop-up s’ouvrira



- Entrer le nom du groupe puis sélectionner les membres
- Cliquer sur “Créer le groupe” pour finaliser la création

Votre groupe est créé vous pouvez commencer à échanger avec ce groupe.

4. BASE DE DONNEE

La Base de donnée de notre application est un fichier XML, contenant les identifiants des utilisateurs et les groupes créés.

Elle est nécessaire au bon fonctionnement de notre application.

5. DESIGN PATTERN

Whatsupp bénéficie de l'implémentation de Design pattern, afin de répondre à des problématiques techniques.

5.1. Singleton

Nous avons utilisé le Design Pattern SINGLETON, pour l'implémentation de classe Database.java afin de n'avoir qu'une seule instance de notre Database. Cela évite des accès concurrentes au fichier Database.xml et donc minimiser les conflits pouvant être liés à cette concurrence.

```
/**
 * Gestion de la base de donnée XML
 * Singleton
 * @author Christophe Kede
 */
public class Database {
    ...
    private static Database INSTANCE = null;
    ...
    public static Database getInstance(String filePath)
    {
        if (INSTANCE == null)
        {
            INSTANCE = new Database(filePath);
        }
        return INSTANCE;
    }
}
```

6. LES PRINCIPES SOLID

La responsabilité unique est respectée dans notre class Logique Wserver.java, Database.java, ..

Le principe de ségregration est respectée.

7. TESTS

Nous avons introduit une partie sur l'application basée sur Junit.