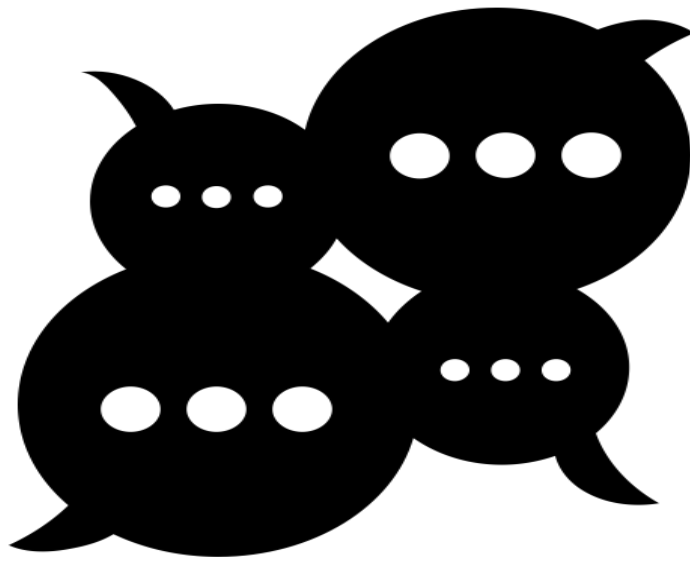


Rapport POO



CHAT'APP

Réalisées par Vincent BAURES, Arnaud BUISSON et Alicia CALMET

Table des matières

1) Administration de l'application	1
a- Code source	1
b- Installation de l'application	1
c- Initialisation du servlet	1
2) Guide d'utilisation	2
a- Première utilisation	2
b- Se connecter	2
c- Choix du pseudo	3
d- Liste des personnes connectées.....	3
e- Menu	4
f- Chatter avec un collègue	4
3) Choix d'implémentation	7
a- Connection avec login et mot de passe.....	7
b- BDD locale ou centralisé.....	7
4) Procédures de tests et de validation	8
a- Authentification et gestion des logins.....	8
b- Utilisateurs déconnectés et connectés	8
c- Transfert de messages et de fichiers.....	8
d- Historique des messages	8
e- Servlet.....	8
f- Déploiement sur différents systèmes d'exploitation	8
5) Notes et futures améliorations	9
a- Interface graphique (GUI).....	9
b- Authentification d'un utilisateur	9
c- Sécurité.....	9
d- Configuration.....	9

1) Administration de l'application

a- Code source

Afin d'accéder au code source vous pouvez vous rendre directement sur le répertoire git du projet à cette adresse :

<https://github.com/VBaures/ChatApp>

Vous pouvez également cloner directement le répertoire sur votre ordinateur grâce à la commande suivante :

git clone <https://github.com/VBaures/ChatApp>

Dans ce répertoire vous trouverez en plus de ce rapport, l'ensemble du code de notre application mais également de notre serveur de présence ainsi que le guide utilisateur et les diagrammes de conception que nous avons réalisés.

b- Installation de l'application

Avant de procéder à l'installation de l'application, veuillez à vérifier plusieurs caractéristiques de votre ordinateur :

- Le projet ayant été compilé avec le Java Développement Kit 11, il est nécessaire de posséder cette version ou une version supérieure sur votre ordinateur.
- Notre application utilise un port spécifique pour accéder au réseau. Il s'agit du port 1040, si ce port est utilisé par une autre application la messagerie ne fonctionnera pas.
- Vous devez également posséder le VPN de l'INSA Toulouse

Une fois ces prérequis en place, vous pouvez télécharger le fichier exécutable en vous rendant à cette adresse :

https://github.com/VBaures/ChatApp/raw/master/ChatApp_Code/out/artifacts/ChatApp_Code_jar/ChatApp_Code.jar

Si vous êtes sur Linux vous pouvez aussi directement utiliser cette commande :

wget https://github.com/VBaures/ChatApp/raw/master/ChatApp_Code/out/artifacts/ChatApp_Code_jar/ChatApp_Code.jar

c- Initialisation du servlet

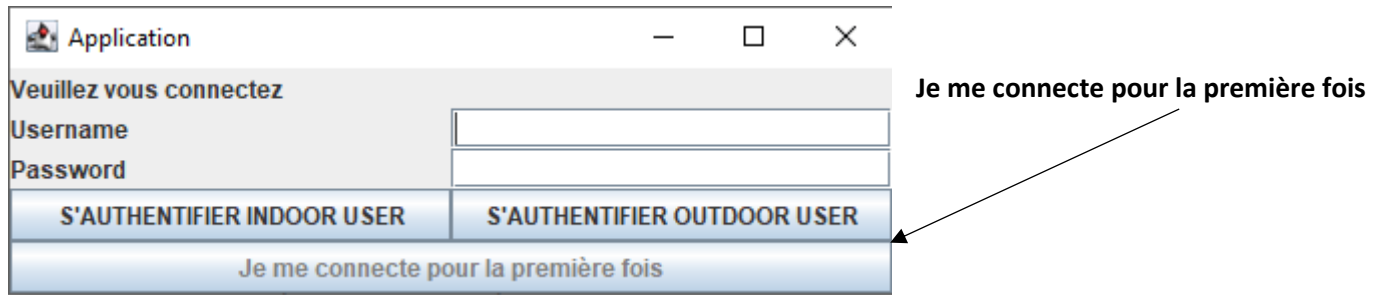
Même si cela n'est pas obligatoire nous recommandons de réinitialiser le servlet. Pour ce faire connecter vous au VPN de l'INSA si vous n'êtes pas sur place et recharger le servlet Chat'App_Server via ce site :

<https://srv-gei-tomcat.insa-toulouse.fr/manager/html>

2) Guide d'utilisation

a- Première utilisation

- Pour utiliser Chat 'APP la première fois, lancer l'application
- La page d'authentification s'affiche
- **Cliquer** sur le bouton « **Je me connecte pour la première fois** »

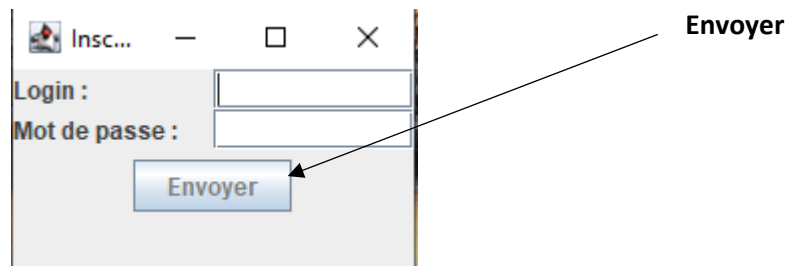


- Vous devez alors choisir un login et un mot de passe

À noter, le username et le mot de passe sont demandés à chaque connexion. Il est important de ne pas les oublier.

- Lorsque vous venez de finaliser la création de votre compte, cliquer sur le bouton « Envoyer »

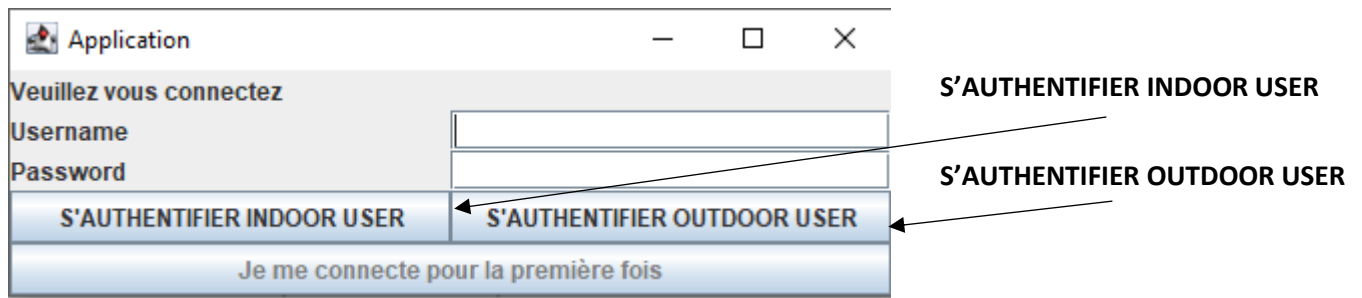
- Un message de validité de l'opération s'affiche. Cliquer sur « OK »
- Vous passez désormais à la procédure de la partie 2 *Se Connecter*



b- Se connecter

- Pour vous connecter, lancer l'application Chat 'APP
- Sur cette page, votre username et votre mot de passe vont vous être demandés

À noter, il n'y a qu'un seul identifiant et donc un seul mot de passe par utilisateur



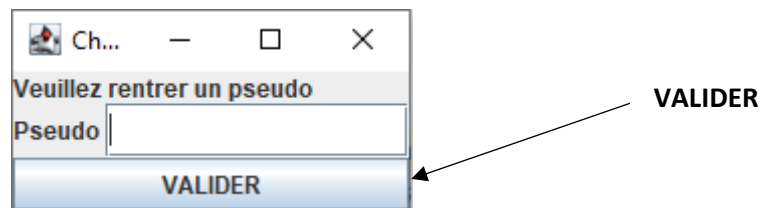
- Si vous êtes sur le réseau de l'entreprise, cliquer sur « S'AUTHENTIFIER INDOOR USER » sinon cliquer sur « S'AUTHENTIFIER OUTDOOR USER »

- Si vous avez rentré un mauvais username ou un mauvais mot de passe un message d'erreur s'affiche

c- Choix du pseudo

- Après s'être correctement authentifier, vous entrez sur la page pseudonyme
- Sur cette page, un pseudonyme va vous être demandé

À noter, le pseudonyme est unique. Il peut être différent à chaque connexion et peut-être changer pendant l'utilisation de l'application. A noter également de ne pas utiliser le pseudo « notdefine »



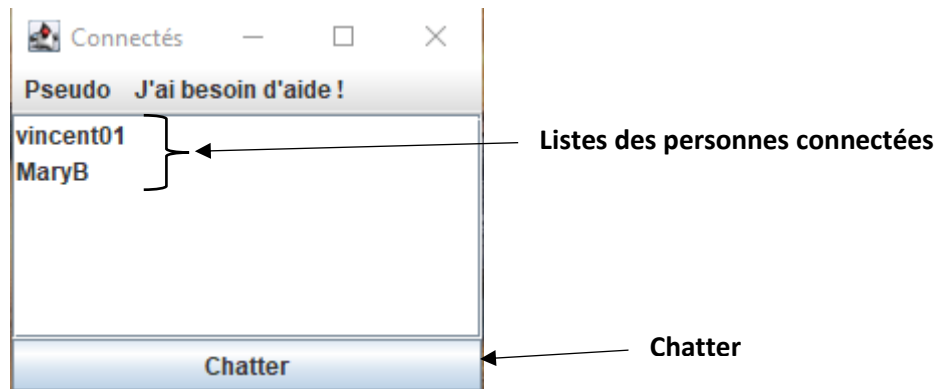
- Cliquer sur « Valider »
- Si le pseudonyme est déjà utilisé par un autre utilisateur alors un message d'erreur s'affiche. Vous devez alors refaire la procédure de choix de pseudonyme

d- Liste des personnes connectées

Maintenant que vous êtes connectés, vous entrez dans l'application de chat, Chat 'APP. Nous allons vous décrire toutes les fonctionnalités de l'application les unes après les autres.

Dès votre connexion, vous trouverez la liste de vos collègues connectés. Vous trouverez aussi un menu que nous évoquerons par la suite.

Pour chatter avec un de vos collègues, cliquez sur son pseudonyme dans la liste puis cliquez sur « Chatter ».



e- Menu

Sur la gauche de la fenêtre des personnes connectées, se trouve deux onglets : « **Options** » et « **J'ai besoin d'aide** ».

a) Options

1- Modification du pseudonyme

- Cliquer sur « **Changer Pseudo** »
- La même procédure qu'à la partie 3 *Changer son pseudonyme* est à refaire

À noter, une fois le changement de pseudonyme effectué, votre nouveau pseudonyme sera visible immédiatement par les autres utilisateurs de l'application.

2- Déconnexion

- Cliquer sur « **Déconnexion** »

À noter, si vous fermez la fenêtre cela agit comme une déconnexion.

b) Aide

- Cliquer sur « **J'ai besoin d'aide** » puis sur « **Aide** »
- Le manuel d'utilisation s'ouvre sous format PDF.

f- Chatter avec un collègue

Lors d'une session de clavardage, le nom et le prénom de votre collègue s'affiche. De plus, si vous avez déjà eu une conversation par le passé, l'historique de votre conversation est automatiquement chargé et visible.

Une fois, la page complètement chargée, deux actions principales sont accessibles : l'envoi de messages et l'envoi de fichiers.

a) Envoie de messages

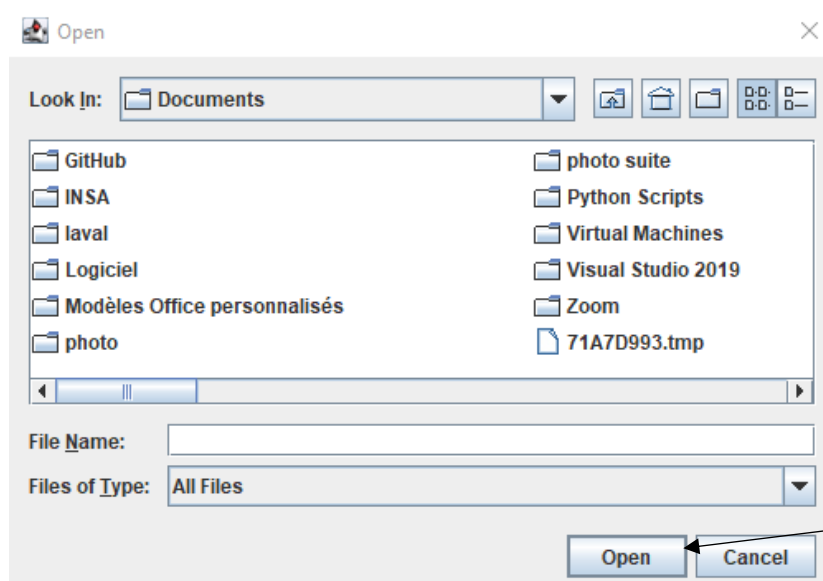
- Lorsque vous voulez discuter avec votre collègue, il vous suffit d'écrire le message dans la zone réservée à cet effet

- Cliquer sur « Envoyer »
- Le message envoyé s'affiche avec votre pseudonyme en bleu et la date d'envoi en italique suivis du message.



b) Envoie de fichiers

- Pour envoyer un fichier, vous devez cliquer sur « Envoyer Fichier » à droite du bouton « Envoyer »
- Une fenêtre affichant les dossiers et les fichiers contenus dans vos Documents s'affiche. Il vous suffit de naviguer et de sélectionner le fichier à envoyer puis de cliquer sur « Open » pour envoyer le fichier.

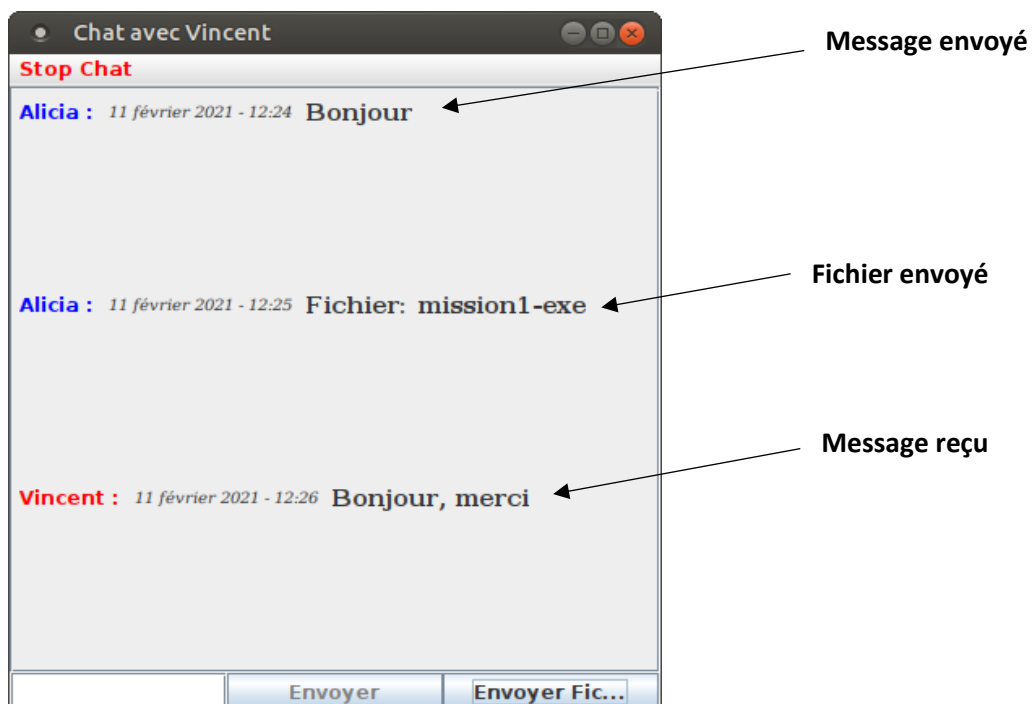
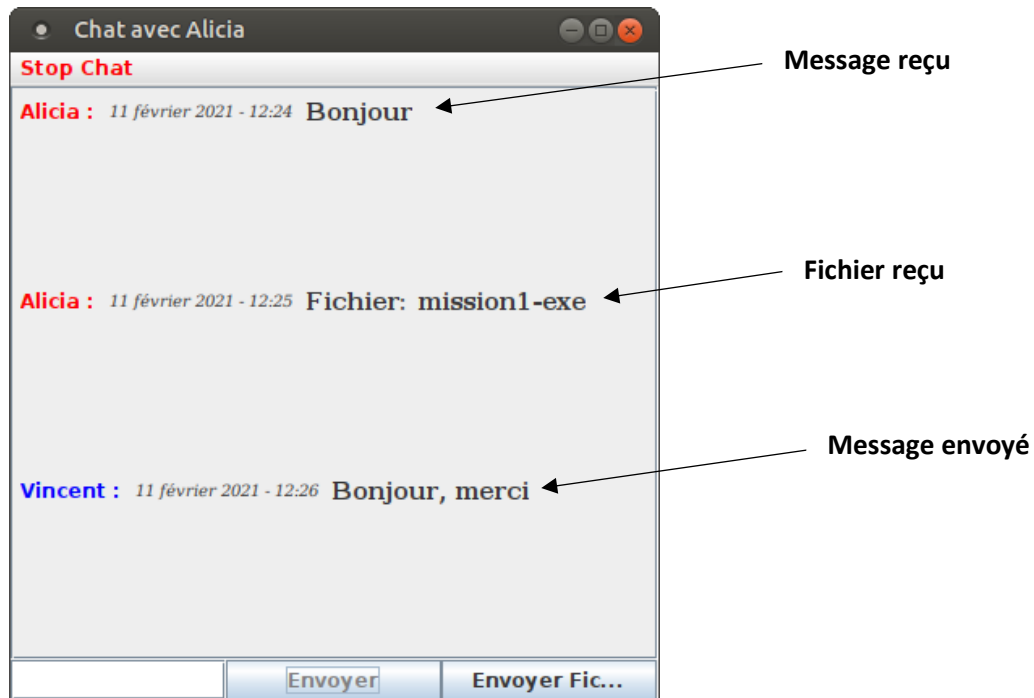


À noter, tous les types de fichiers sont acceptés.

c) Réception de messages et de fichiers

- Les messages que vous recevez sont affichés avec le pseudonyme de votre collègue en rouge et la date suivie du message
- Les fichiers reçus, sont affichés sous la forme *Fichier :* suivi du nom de fichier. Un clic gauche sur le fichier permet de le télécharger sur votre ordinateur

Ci-dessous un exemple de communication entre deux utilisateurs, Alicia et Vincent



3) Choix d'implémentation

a- Connection avec login et mot de passe

Bien que cela n'était pas spécifié dans le cahier des charges, nous avons décidé de donner la possibilité aux utilisateurs de notre application de s'inscrire avec un login et un mot de passe. Ainsi, il est possible pour chaque personne d'accéder à ces messages depuis n'importe quel ordinateur.

b- BDD locale ou centralisé

Ce choix d'implémentation découle du premier, en effet si nous avions utiliser une base de données par ordinateur, ceci aurait impliquer de stocker l'ensemble des informations de compte et l'ensemble des messages de tous les utilisateurs sur chaque machine et donc une perte de place importante. Avec le choix d'une base de données centralisée cela facilite le stockage et l'accès au données. Le seul inconvénient de notre choix est la nécessité de se connecter à un VPN pour y accéder car la base de données se trouve sur les serveurs de l'INSA.

4) Procédures de tests et de validation

a- Authentification et gestion des logins

Lors du lancement de l'application, la page d'authentification s'affiche. Après avoir créé le couple login/mot de passe, nous avons essayé de nous connecter avec sur différents ordinateurs. Nous arrivons toujours à nous authentifier. Cet objectif est validé.

b- Utilisateurs déconnectés et connectés

Afin de visualiser si une personne est connectée, on lance une instance de l'application sur un ordinateur. Ensuite, on lance une autre instance sur un autre ordinateur (ce dernier peut faire partie ou non du même réseau que le premier ordinateur). Sur la page des personnes connectées, on peut lire le pseudonyme de l'utilisateur qui est connecté. Lorsque ce collègue vient à se déconnecter, son pseudonyme est supprimé de votre liste des utilisateurs connectés. Cet objectif est ainsi rempli.

c- Transfert de messages et de fichiers

L'envoi et la réception de messages est l'un des critères fondamentaux de notre application. C'est une des dernières fonctions que nous avons développées avant la mise en place du servlet. Pour vérifier son bon fonctionnement, nous envoyons un message ou un fichier à un autre utilisateur local de l'application. Il suffit à ce dernier de le lire ou de l'ouvrir. Même si la taille maximale des fichiers (et des messages) est de 4 Go nous considérons cet objectif rempli lui aussi.

Malheureusement, à la suite d'un problème de VPN nous n'avons pas pu tester l'échange de message depuis et vers un utilisateur se trouvant hors du réseau de l'INSA.

d- Historique des messages

En ce qui concerne l'historique des messages, cette fonction n'a été testée qu'après la vérification de la possibilité de communiquer entre deux utilisateurs. Afin de tester cette fonctionnalité, nous avons tout d'abord essayé d'échanger des messages de texte entre deux utilisateurs. Nous nous sommes ensuite déconnectés puis reconnectés pour voir si à notre retour les messages étaient toujours présents. La même procédure a été mise en œuvre pour les échanges de fichier et à chaque fois nous récupérons l'intégralité des messages de l'historique précédemment envoyés. Ainsi, même si cette fonctionnalité n'a pu être testée que sur des utilisateurs locaux nous considérons qu'il n'y a aucune raison que cela ne marche pas pour des utilisateurs connectés à distance. L'objectif est donc validé.

e- Servlet

Le servlet fut la dernière chose mise en place lors du codage de notre application. Afin, de tester ses capacités nous avons d'abord regardé sa capacité à recevoir les informations que l'application lui envoie. Pour se faire nous nous connectons à un utilisateur puis nous nous dirigeons sur le serveur via un navigateur web pour voir son contenu.

Une fois cela validé, nous avons vérifié que l'application pouvait récupérer les informations stockées sur le serveur en nous connectant avec deux utilisateurs. Ce test fut une réussite. En effet un utilisateur distant peut récupérer toutes les caractéristiques des autres utilisateurs de l'application via le serveur.

f- Déploiement sur différents systèmes d'exploitation

Tout au long du développement, nous avons principalement codé et testé sous l'environnement Windows. Nous avons testé Chat'App sous Linux et l'application fonctionne correctement.

Cependant, n'ayant pas d'ordinateur supportant le système d'exploitation MAC, nous n'avons pas pu vérifier son efficacité.

5) Notes et futures améliorations

a- Interface graphique (GUI)

L'interface graphique a été développée sous JavaSwing, elle n'est donc pas très esthétique. Pour améliorer cet aspect visuel nous aurions pu utiliser JavaFX. Chat'App aurait pu ressembler à des applications de chat conventionnelles telles que Messenger.

b- Authentification d'un utilisateur

Pour se connecter à la base de données, nous avons choisi d'utiliser le système d'authentification courant : login/mot de passe. Cependant, nous n'avons pas envisagé dans notre implémentation que l'utilisateur oublie son login ou son mot de passe. Une amélioration serait de permettre à l'utilisateur de récupérer son login ou mot de passe perdu.

c- Sécurité

En termes de sécurité, plusieurs points sont à soulever notamment le non-chiffrement des messages lorsqu'ils sont envoyés sur les réseaux ou stockés dans la base de données. Cependant, le point le plus important en termes de sécurité concerne les mots de passe. En effet, lors de la connexion à l'application le mot de passe est visible de tous et n'est pas non plus crypté dans la base de données.

d- Configuration

Un problème majeur de notre application est son manque de flexibilité. En effet, presque rien n'est configurable sur notre application. Pour changer le numéro de port du serveur, ou modifier les adresses du serveur de présence et de la base de données, il faut accéder directement au code et les modifier chaque fois qu'ils apparaissent. Il serait judicieux de créer dans le futur des variables globales modifiables dans un fichier de configuration sans avoir à toucher au code.