# Documentation pour les développeurs

*Dans cette partie, je vais vous guider à comprendre comment le code est architecturé, les fichiers, les classes*

## 1.Structure du Code

### 1.1Architecture Générale :

• Serveur (server.py) : C'est le cœur de notre application. Il gère toutes les connexions, les salons, les messages et les interactions avec la base de données MySQL.

• Client (client.py) : C'est l'interface utilisateur qui permet aux utilisateurs de se connecter au serveur, de participer à des salons, d'envoyer des messages, etc.

### 1.2Modules et Classes Importantes :

• Server Class : Gère les connexions des clients, les salons, la diffusion des messages, etc.

• ClientThread : Un thread côté serveur pour gérer la communication avec chaque client connecté.

• ChatClient : La classe principale côté client pour l'interface utilisateur et la gestion des interactions avec le serveur.

## 2.Base de Données

Pour bien comprendre comment l'application fonctionne, c'est crucial de connaître la structure de la base de données. Voici les tables que j'ai utilisées et leur rôle :

Utilisateurs :

• Contient des infos sur les utilisateurs, comme leur alias, nom, prénom et mot de passe.

• alias est la clé primaire ici, donc chaque utilisateur doit avoir un alias unique.

Salons :

• Liste des salons disponibles pour le chat.

• J'ai déjà inclus quelques salons de base comme 'Général', 'Blabla', 'Comptabilité', 'Informatique' et 'Marketing'

Messages :

• Stocke tous les messages échangés dans les salons.

• Chaque message est lié à un utilisateur (via user\_alias) et un salon (via salon\_name).

PrivateMessages :

• Pour les messages privés entre utilisateurs. (Permet de stocker tous les messages privés)

• Chaque message a un expéditeur (sender\_alias) et un destinataire (recipient\_alias).

BannedUsers :

• Utilisateurs bannis de l'application avec la raison et la date du bannissement.

Avertissements :

• Enregistrements des avertissements donnés aux utilisateurs, avec la raison et la date.

• Sécurité des Mots de Passe : Les mots de passe stockés dans la table Utilisateurs devraient toujours être hashés. C'est une bonne pratique de sécurité pour protéger les informations des utilisateurs.