

SAE3.02 : Réflexion sur le projet

1) Introduction

Du mardi 14 novembre au 31 décembre 2023, nous avons développé de manière individuelle un service de tchat composé d'un serveur, de plusieurs clients et d'une base de données. Ce projet aura été prévu sur 41,5 heures dont 28 en autonomie. En plus de ce document et de nos programmes, nous devons rendre une documentation client, une documentation serveur et une documentation développeur directement dans notre code sous forme de DocString.

Ce document servira à détailler le processus de développement de cette application en y ajoutant nos réflexions en termes de difficultés rencontrées, de sécurité et de viabilité de cette application dans le temps. Nous vous souhaitons une agréable lecture.

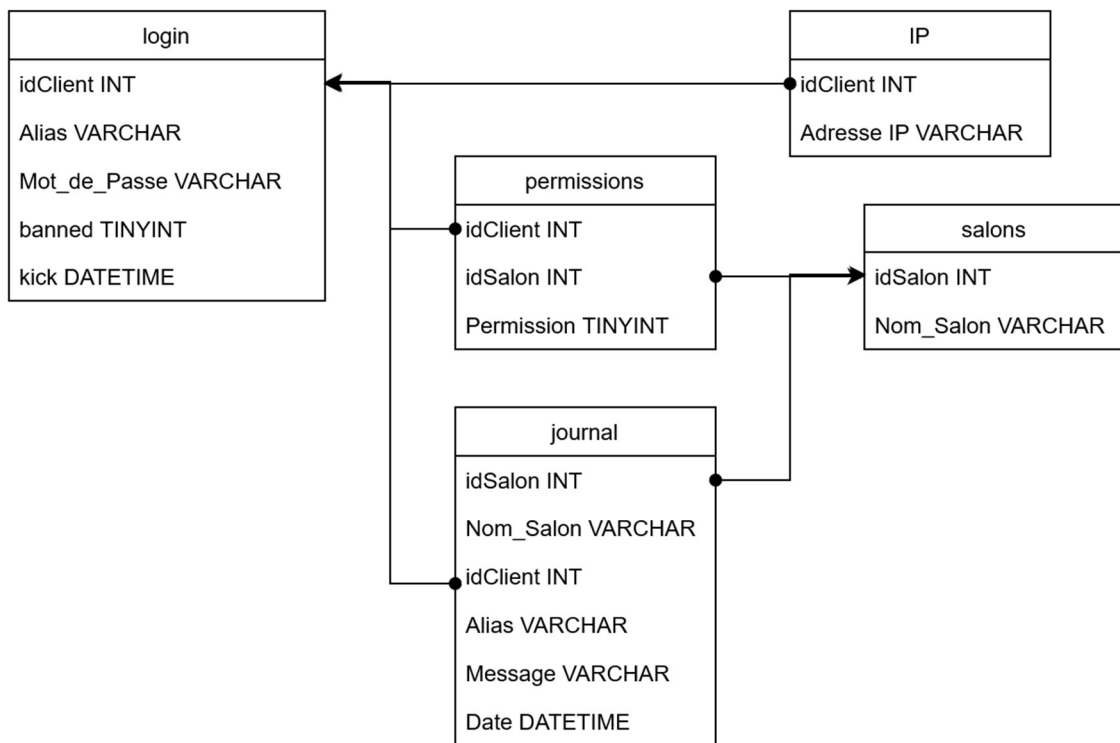
Table des matières

SAE3.02 : Réflexion sur le projet	1
1) Introduction.....	1

2) Base de données

Avant même de commencer à programmer nous avons travaillé sur le modèle relationnel de la base de données que nous avons utilisé. L'inconvénient principal d'une base de données et d'un modèle relationnel est leur trop faible possibilité de modification. Une fois que les tables sont créées et les premières valeurs entrées, il est très difficile de modifier les relations ou les colonnes dans cette base de données. Il faut donc être sûr de ce que l'on veut dès le début du projet et penser au futur de l'application. Découvrant les outils à notre disposition nous n'étions pas assez équipés en termes de connaissances pour produire un modèle relationnel complet et fonctionnel pour la programmation du serveur de tchat. Il nous aura donc fallut recommencer plusieurs fois la base de données afin qu'elle corresponde au mieux au serveur au fur et à mesure de son développement. De même si nous voulons ajouter de nouvelles fonctionnalités dans le futur, il faudra sûrement reconstruire de A à Z la base de données, ce qui implique une sauvegarde et un reformatage de toutes les informations qu'elle contient.

Nous avons tout de même abouti à un schéma relationnel complet et fonctionnel correspondant à la version actuelle de notre serveur de tchat :



3) Serveur

La programmation du serveur aura présenté un challenge surtout au niveau des threads permettant d'effectuer plusieurs actions simultanément tout en les faisant communiquer. Plusieurs fonctionnalités nous ont été demandées et un tri en fonction des priorités a été effectué. En effet, pour le bon déroulement d'un projet il est nécessaire d'identifier les fonctionnalités obligatoires, importantes et optionnelles. Parmi les fonctionnalités obligatoires, le serveur asynchrone était une évidence. Notre serveur est en capacité de connecter plusieurs clients en même temps et de communiquer avec chacun d'entre eux. Chaque message reçu est redistribué à tous les clients et enregistré dans la table « journal » de la base de données afin de conserver une trace des conversations.

L'authentification et l'inscription d'utilisateurs font partie des fonctionnalités importantes que nous avons identifiées. Au démarrage de la connexion entre le client et le serveur s'opère un échange d'identifiants de connexion qui sont vérifiés par le serveur en recherchant dans la base de données. En fonction du bouton cliqué côté client le serveur va chercher à authentifier ou inscrire le client dans la base de données. Si les identifiants ne sont pas reconnus ou si le mot de passe est erroné, par exemple, le serveur envoie une notification au client qui pourra modifier sa saisie et effectuer les actions correctes.

La présence de salons de discussion est aussi une fonctionnalité importante du serveur. Tous les salons sont connus du client, mais tous les utilisateurs n'ont pas le droit d'écriture dans ces salons. Au niveau du serveur, seules les permissions sont vérifiées et un message de confirmation est envoyé au client s'il possède le droit d'écrire dans ce salon. Cette vérification est effectuée à chaque message qui est envoyé dans un autre salon que « Général » qui est un salon ouvert à tous. Si le client n'est pas autorisé à écrire dans un salon, l'administrateur reçoit une notification en ligne de commande l'informant de la volonté de l'utilisateur de pouvoir écrire dans ce salon. Afin d'accéder à la demande de l'utilisateur, l'administrateur doit modifier directement la table « permissions » de la base de données. (Une commande a été imaginée afin d'éviter à l'administrateur de devoir modifier directement dans la base de données mais cette commande n'a pas été implémentée).

Un historique de messages par salon devait être envoyé à chaque utilisateur lorsqu'il changeait de salon, mais cette fonctionnalité a été abandonnée en raison du caractère trop aléatoire de la réception de cet historique qui pouvait amener le client à planter.

Un système de tentatives maximum d'authentifications a été mis en place au début de la programmation du serveur mais il a été abandonné car trop contraignant pour son caractère optionnel. Ce système est néanmoins resté dans le code final sous forme de commentaires.

L'administrateur a la possibilité d'entrer les commandes « ban », « kick », « unban » et « kill » en ligne de commande en même temps que le script principal fonctionne. Ces commandes interagissent avec la base de données dont les valeurs sont vérifiées à chaque tentative d'authentification d'un client. Cependant, ces actions ne sont liées qu'à l'alias de l'utilisateur. L'utilisation de l'adresse IP ayant été jugée optionnelle, elle figure dans la base de données sous la forme de table « ip » mais n'a pas été implémentée dans le code.

Comme indiqué pour la base de données, le principal problème pouvant être rencontré est la modification ou l'ajout de nouveaux champs dans le modèle relationnel. Il faudrait alors modifier tout le code du serveur pour le préparer à recevoir les bons messages et les bonnes instructions. De plus, les messages envoyés par socket ne sont pas fiables et peuvent être source d'erreurs du côté du client et du serveur.

4) Client

Le programme client propose une interface graphique du tchat en communication avec le serveur. Au lancement du script une fenêtre de connexion s'affiche et permet d'entrer les informations de connexion au serveur puis les identifiant d'authentification ou d'inscription. Une fois cette étape confirmée et terminée par le serveur, cette fenêtre laisse place à la fenêtre de tchat principale permettant de changer de salon, d'envoyer des messages et de les recevoir.

En plus de la difficulté liée aux threads comme pour le serveur, s'est ajoutée la découverte des interfaces graphiques. En programmant cette interface il faut faire attention en permanence à ne pas créer de boucle qui empêcherait le programme principal de fonctionner. En revanche, la programmation de cette partie client aura été la plus simple et la plus rapide car plus indépendante. Le client doit recevoir des messages bien précis du serveur et en envoyer selon un certain format, mais en dehors de ça, le client ne dépend ni de la base de données ni du serveur, ce qui présente un avantage pour le développement de cette application dans le futur.

La première réflexion que nous avons eu concernant le client était qu'il devait effectuer le moins d'actions de vérification (authentification, permissions...). Ces actions doivent être gérées par le serveur qui regarde dans la base de données. Si ces actions étaient effectuées depuis le client, cela présenterait une énorme faille de sécurité qui a par conséquent été évitée.

5) Conclusion

En dehors de la programmation de cette application plutôt complète, nous avons trouvé ce projet intéressant du point de vue gestion de projet. En effet, nous avons trouvé que ce projet nous simulait une situation réelle dans laquelle nous devons produire une application en fonction des demandes d'un client. Nous avons aussi dû nous appliquer à rendre des livrables simples et compréhensibles par le plus grand nombre en nous projetant à la place des utilisateurs.

D'un point de vue personnel, je n'ai pas aimé le côté développement de ce projet malgré l'importance que cela représentait. Il fallait se projeter sur des fonctionnalités dont nous ne savions rien et utiliser des bibliothèques que nous découvrons. En revanche, ce projet aura fortement sollicité ma prise de décision et ma gestion de projet afin de rendre un projet fonctionnel et dans les attentes du client dans le délai imposé. J'espère que ce projet sera à la hauteur de vos attentes bien qu'il ne soit plus développé après son rendu. Nous vous remercions de votre attention envers notre travail.