



SAE 302

Document réponse

DE AZEVEDO Kevin

SOMMAIRE :

I – Introduction :	3
II – Réalisations effectives du projet :	4
A. Description générale du projet :	4
B. Fonctionnalités implémentées :	5
C. Défis rencontrés et solutions apportées :	6
III – Limites de l'application :	7
IV – Améliorations à prévoir :	8
V - Conclusion :	9

I – Introduction :

Dans un paysage numérique où la confidentialité des données devient de plus en plus précaire, ce projet émerge comme une tentative à la fois d'apprentissage des moyens de mise en place d'un système d'échange multifactoriel et de réponse aux défis de la communication moderne.

L'objectif principal était de créer une application qui facilite la communication en temps réel dans un environnement intuitif. L'application se veut simple d'utilisation tout en manipulant une multitude de leviers techniques invisibles, accessible à tous les utilisateurs, indépendamment de leur niveau de compétence en matière de technologie.

Le projet vise à briser le stéréotype selon lequel les applications informatiques sont complexes et peu conviviales, essayant ainsi de démontrer qu'il est possible d'allier ergonomie, fonctionnalité et sécurité.

Le défi principal résidait dans la capacité à élaborer un système qui à la fois répond à certaines normes actuelles de sécurité et est également conçu pour évoluer et s'adapter à des changements futurs.

II – Réalisations effectives du projet :

A. Description générale du projet :

L'application conçue se compose de deux parties principales : le serveur de messagerie et l'interface client.

Le serveur, développé en Python, utilise des sockets TCP/IP pour gérer les connexions réseau et s'interface avec une base de données MySQL pour le stockage des données utilisateur et des messages.

Le serveur est responsable de la gestion des connexions client, de l'authentification, de la transmission des messages et de la maintenance de l'état global de l'application.

L'interface client, construite avec PyQt6, offre une expérience utilisateur graphique pour la connexion au serveur, l'authentification, l'inscription, et la participation à des discussions dans des salons publics.

L'interface est conçue pour être intuitive et facile à naviguer, tout en fournissant un accès direct aux fonctionnalités offertes par le serveur.

B. Fonctionnalités implémentées :

→ Authentification et inscription : Les utilisateurs peuvent créer un compte et se connecter à l'application. Le serveur gère l'authentification et stocke les informations relatives aux utilisateurs dans la base de données MySQL.

→ Contrôle d'accès : Les utilisateurs peuvent se voir octroyer ou révoquer des accès à différents salons ou fonctionnalités du serveur, en fonction de leur adresse email et de leurs adresses IP.

→ Communication en temps réel : L'application utilise des sockets TCP/IP pour permettre une communication fluide en temps réel dans les différents salons.

→ Salons de discussion : Les utilisateurs peuvent rejoindre différents salons de discussion et y communiquer librement.

→ Système de modération : Les administrateurs peuvent bannir ou expulser (kick) des utilisateurs pour modérer les discussions.

→ Stockage des messages : Les messages envoyés par les différents utilisateurs sont stockés dans la base de données MySQL, permettant de maintenir un historique des conversations des salons publics.

→ Résilience du serveur : Le serveur de messagerie est conçu pour permettre une tolérance aux pannes élevée, s'assurant de pouvoir offrir une exécution en continu face aux perturbations des communications.

→ Modularité : Les codes du système de messagerie serveur-client sont décomposés en une multitude de rôles différents, permettant une haute évolutivité et adaptativité en réponse à des besoins futurs.

→ Expérience utilisateur : Le client de messagerie a été pensé pour être simple d'utilisation et relativement esthétique, pour un meilleur confort à l'usage.

C. Défis rencontrés et solutions apportées :

→ Gestion de la concurrence : Avec l'approche multi-thread, le serveur peut gérer efficacement plusieurs connexions clients simultanément. Les verrous garantissent que les interactions avec la base de données sont sécurisées et cohérentes.

→ Communication Serveur-Client : L'utilisation de sockets TCP/IP offre une méthode fiable et efficace pour la communication réseau. Cette approche garantit une transmission rapide des données et une synchronisation en temps réel entre le serveur et les clients.

→ Interface utilisateur intuitive : L'utilisation de PyQt6 pour l'interface client, bien que complexe, a permis de créer une interface esthétique et conviviale. Le design et la disposition des composants sont pensés pour rendre l'expérience utilisateur aussi agréable et facile que possible.

→ Gestion des données : La base de données MySQL sert de pivot central pour le stockage et la gestion des données. Les requêtes SQL intégrées au serveur permettent une manipulation efficace des données.

→ Scalabilité et maintenance : Pour assurer la scalabilité de l'application, une architecture modulaire a été adoptée. Cette approche a permis de mettre en place les fondations nécessaires à une maintenance ultérieure facilitée de l'application.

III – Limites de l'application :

- Transfert de données en clair : Une limitation critique de l'application est le transfert de données sensibles, notamment les mots de passe, en clair. Cette pratique expose les utilisateurs à des risques de sécurité, notamment les attaques de type "man-in-the-middle".
- Stockage non chiffré : Le projet ne comporte pas de chiffrement des données stockées, ce qui est une préoccupation majeure pour la confidentialité et la sécurité des informations utilisateur.
- Sessions simples : Le système actuel ne dispose pas de mécanismes avancés pour la gestion des sessions utilisateurs, tels que les jetons d'authentification ou les cookies de session, essentiels pour renforcer la sécurité globale des utilisateurs.
- Maintenance : À court terme, il serait nécessaire de mettre en place des correctifs de bugs et des mises à jour de sécurité mineures.
- Interface serveur : Bien que fonctionnelle, elle pourrait bénéficier d'une révision du traitement des interactions administrateur.
- Règlementations : L'application ne dispose pas de politique de confidentialité claire ni de procédure de gestion des données pour assurer la conformité avec les différentes réglementations de protection des données.

IV – Améliorations à prévoir :

→ Chiffrement des données : Implémenter le chiffrement SSL/TLS pour sécuriser les communications entre le client et le serveur. Pour les mots de passe, utiliser un algorithme de hachage comme bcrypt ou Argon2.

→ Authentification plus sécurisée : Mettre en place une authentification à deux facteurs (2FA) voir à multi-facteurs (MFA) pour renforcer la sécurité des comptes utilisateurs.

→ Sessions avancées : Intégrer un système de gestion des sessions avec des jetons (tokens) uniques et des cookies de session pour suivre et valider les sessions utilisateur de manière sécurisée.

→ Amélioration des interfaces : Rendre les interfaces clients et serveur plus intuitives et modernes, en améliorant la disposition des éléments, en intégrant des notifications en temps réel et en peaufinant l'esthétique globale.

→ Sécurité de la base de données : Renforcer la sécurité de la base de données en mettant en œuvre des contrôles d'accès plus stricts et en chiffrant les données sensibles stockées.

→ Gestion d'erreurs améliorée : Développer un système de gestion d'erreurs plus robuste qui fournira des retours clairs et utiles aux utilisateurs en cas de problèmes.

→ Journalisation et surveillance : Implémenter un système de journalisation pour enregistrer les activités importantes et potentiellement suspectes, facilitant ainsi la surveillance et l'analyse des problèmes de sécurité.

V - Conclusion :

Le projet de serveur de messagerie a été une aventure haletante qui a permis de mettre en lumière l'importance des communications et les efforts nécessaires pour parvenir à les gérer efficacement.

Bien que le projet ait réussi à fournir une plateforme fonctionnelle pour la messagerie en temps réel, il a également révélé des domaines clés nécessitant des améliorations, en particulier en matière de sécurité des données stockées et transmises.

Le processus de développement a été long et parfois claudiquant, soulignant l'importance du prototypage et de la gestion de projet. Les défis rencontrés ont fourni des opportunités précieuses pour comprendre en profondeur les aspects techniques et l'abstraction nécessaires en matière de développement d'applications complexes.

En conclusion, ce projet étudiant, bien que présentant des limites évidentes, offre une base solide pour une application de messagerie fonctionnelle et conviviale. Avec les améliorations recommandées, il a le potentiel de se transformer en une solution robuste et fiable.

La route vers une application de messagerie optimale est encore longue, mais les efforts fournis ici ont permis d'acquérir des fondations solides permettant d'avoir le recul nécessaire à une meilleure appréhension de futurs projets de développement impliquant des méthodes et protocoles de communication réseau sécurisés.

Merci pour cette expérience, qui s'est avérée nécessaire et enrichissante.

Sincèrement.