**Cahier des Clauses Techniques Particulières (CCTP)**

1. Introduction
   1. Objet du document

Ce document a pour objectif de définir les spécifications techniques et fonctionnelles du projet de développement d'une application client-serveur avec gestion de salons et une interface graphique côté client.

* 1. Contexte du projet

Le projet vise à créer une application de chat en temps réel permettant aux utilisateurs de se connecter à des salons de discussion et d'interagir avec d'autres participants. L'application sera développée en utilisant le langage de programmation Python et la bibliothèque de sockets pour la communication entre le client et le serveur.

* 1. Objectifs du projet

Les objectifs du projet sont les suivants :

* + Créer une interface graphique conviviale côté client pour la gestion des salons et des conversations.
  + Mettre en place un serveur permettant la création, la gestion et la suppression des salons de discussion.
  + Assurer une communication en temps réel entre les clients connectés et le serveur.
  + Garantir la sécurité des communications et la confidentialité des données.

1. Description générale du projet

2.1 Description de l'application client-serveur

L'application sera basée sur une architecture client-serveur, où le client sera une application avec une interface graphique permettant aux utilisateurs de se connecter aux salons, d'envoyer et de recevoir des messages, et de gérer les paramètres de leur compte. Le serveur sera responsable de la gestion des salons et de la communication entre les clients.

2.2 Fonctionnalités principales

Les fonctionnalités principales de l'application seront les suivantes :

* + Connexion et déconnexion des utilisateurs.
  + Création, suppression et gestion des salons de discussion.
  + Envoi et réception de messages en temps réel.
  + Affichage de la liste des utilisateurs connectés.

2.3 Technologies utilisées

Le développement de l'application se fera en utilisant les technologies suivantes :

* + Langage de programmation : Python.
  + Bibliothèque de sockets pour la communication client-serveur.
  + Bibliothèque graphique pour l'interface utilisateur côté client (Tkinter).

1. Spécifications techniques

3.1 Architecture client-serveur

L'architecture client-serveur sera basée sur le modèle de communication en sockets. Le serveur écoutera les connexions entrantes des clients et gérera la communication entre eux. Chaque client disposera d'une instance de l'application avec une interface graphique.

3.2 Interface graphique côté client

L'interface graphique côté client sera développée en utilisant une bibliothèque graphique adaptée au langage Python (Tkinter). Elle devra permettre aux utilisateurs de se connecter, de choisir un salon, d'envoyer et de recevoir des messages.

3.3 Gestion des salons

Le serveur sera responsable de la création, de la gestion et de la suppression des salons de discussion. Les utilisateurs pourront rejoindre les salons existants, en créer de nouveaux et inviter d'autres utilisateurs à les rejoindre.

3.4 Communication entre le client et le serveur

La communication entre le client et le serveur se fera via des sockets. Les messages échangés seront encodés dans un format spécifique, incluant les informations nécessaires pour l'identification des utilisateurs, la gestion des salons et les actions à effectuer.

1. Exigences fonctionnelles

4.1 Connexion et déconnexion du client

Les utilisateurs devront pouvoir se connecter en fournissant leurs identifiants et mots de passe. Une fois connectés, ils pourront se déconnecter de l'application.

4.2 Création et gestion des salons

Les utilisateurs pourront créer de nouveaux salons de discussion et gérer ceux auxquels ils participent. Ils pourront également inviter d'autres utilisateurs à rejoindre les salons.

4.3 Chat en temps réel

Les utilisateurs pourront envoyer et recevoir des messages en temps réel dans les salons de discussion. Les messages seront affichés en temps réel à l'écran, et une notification sonore pourra être émise lorsqu'un nouveau message est reçu.

1. Exigences non fonctionnelles

5.1 Performances

L'application devra être capable de gérer un grand nombre d'utilisateurs connectés simultanément, tout en maintenant des performances optimales.

5.2 Scalabilité

L'architecture de l'application devra être conçue de manière à pouvoir être facilement étendue pour supporter une augmentation du nombre d'utilisateurs et de salons.

5.3 Interface utilisateur conviviale

L'interface graphique côté client devra être intuitive, ergonomique et conviviale pour faciliter la prise en main de l'application par les utilisateurs.

1. Livrables

6.1 Application client

* + Code source de l'application client avec une interface graphique fonctionnelle.
  + Documentation d'installation et de configuration de l'application.

6.2 Application serveur

* + Code source du serveur avec les fonctionnalités de gestion des salons, de communication entre les clients et de sauvegarde des conversations.
  + Documentation d'installation et de configuration du serveur.

6.3 Guide d'utilisation

* + Une capture vidéo est disponible montrant comment lancer les différents scripts Python et comment les clients communiquent entre eux.

1. Planning et ressources

7.1 Phases du projet

* + Phase 1 : Analyse des besoins et conception de l'architecture.
  + Phase 2 : Développement de l'application client.
  + Phase 3 : Développement du serveur et des fonctionnalités de gestion des salons.
  + Phase 4 : Intégration du client et du serveur, tests et débogage.
  + Phase 5 : Documentation, finalisation et livraison.

7.2 Répartition des tâches

* + Développeur client : Responsable du développement de l'interface graphique côté client.
  + Développeur serveur : Responsable du développement du serveur et des fonctionnalités côté serveur.
  + Chef de projet : Responsable de la coordination globale du projet, de la planification et de l'organisation des ressources.

7.3 Ressources requises

* + Environnement de développement Python avec les bibliothèques nécessaires.
  + Serveur d'hébergement pour le déploiement de l'application.

1. Modalités de suivi et de validation

8.1 Réunions de suivi

Des réunions de suivi régulières seront organisées pour discuter de l'avancement du projet, résoudre les problèmes éventuels et valider les fonctionnalités développées.

8.2 Tests et validation

Des tests manuels ont été réalisé pour s’assurer que le produit réponde bien aux exigences du client.

1. Conditions contractuelles

9.1 Durée du projet

La durée totale du projet est estimée à 2 jours.

9.2 Budget et paiements

Le budget total du projet est estimé à 0€. Les modalités de paiement seront convenues entre les parties avant le démarrage du projet.

9.3 Responsabilités des parties

* + Le client fournira les spécifications détaillées du projet, les éléments graphiques nécessaires et les ressources d'hébergement.
  + Le prestataire sera responsable du développement de l'application conformément aux spécifications fournies et s'engagera à respecter les délais et la qualité attendue.