## Cahier des charges pour une application de programmation réseau Python pour la gestion de fichiers

### 1. Introduction

Ce cahier des charges définit les exigences d'une application de programmation réseau Python pour la gestion de fichiers. L'application permettra aux utilisateurs de créer un serveur, d'envoyer des fichiers vers ce serveur et de les récupérer en s'y connectant.

### 2. Fonctionnalités

L'application devra proposer les fonctionnalités suivantes :

Création d'un serveur : Les utilisateurs doivent pouvoir créer un serveur sur leur machine locale. Le serveur doit être accessible via une adresse IP et un port spécifiques.

\* \*\*Envoi de fichiers:\*\* Les utilisateurs doivent pouvoir envoyer des fichiers vers le serveur. L'application doit prendre en charge différents formats de fichiers.

\* \*\*Récupération de fichiers:\*\* Les utilisateurs doivent pouvoir récupérer des fichiers stockés sur le serveur. L'application doit afficher la liste des fichiers disponibles sur le serveur et permettre aux utilisateurs de télécharger les fichiers souhaités.

\* \*\*Interface utilisateur:\*\* L'application doit disposer d'une interface utilisateur simple et intuitive. L'interface peut être réalisée en utilisant une bibliothèque graphique Python comme Tkinter ou PyQt.

\* \*\*Gestion des erreurs:\*\* L'application doit gérer les erreurs de manière appropriée et fournir des messages d'erreur clairs aux utilisateurs.

### 3. Limites de temps

La réalisation de cette application devrait prendre environ \*\*5 jours ouvrables\*\*. Ce délai peut varier en fonction de la complexité de l'interface utilisateur et des fonctionnalités supplémentaires souhaitées.

### 4. Tâches et jalons

Le tableau suivant présente les tâches et les jalons pour la réalisation de l'application :

| Tâche | Jalon | Délai |

|---|---|---|

| Conception de l'architecture de l'application | Schéma de l'architecture | 2 jours |

| Développement du module de création de serveur | Serveur fonctionnel | 7 jours |

| Développement du module d'envoi de fichiers | Envoi de fichiers fonctionnel | 5 jours |

| Développement du module de récupération de fichiers | Récupération de fichiers fonctionnelle | 5 jours |

| Développement de l'interface utilisateur | Interface utilisateur complète | 4 jours |

| Tests et validation | Application testée et validée | 1 jour |

### 5. Ressources

Pour réaliser cette application, les ressources suivantes seront nécessaires :

\* Un ordinateur avec Python 3.x installé

\* Une bibliothèque graphique Python comme Tkinter ou PyQt

\* Un logiciel de gestion de version comme Git

### 6. Critères d'acceptation

L'application sera considérée comme terminée lorsque les critères d'acceptation suivants seront remplis :

\* Toutes les fonctionnalités décrites dans ce cahier des charges sont implémentées.

\* L'application fonctionne correctement sans erreurs majeures.

\* L'application dispose d'une interface utilisateur simple et intuitive.

\* L'application est documentée avec des commentaires clairs et concis.

### 7. Révisions

Ce cahier des charges peut être révisé en fonction des besoins et des changements de contexte. Toute modification sera communiquée aux parties prenantes concernées.