GNG 2501

Manuel d’utilisation de produit pour le projet de conception

**IA de traduction universelle**

Soumis par:

Dawid Lapinski, 300449088

Momar Ndour, 300309764

Joseph Tshimanga, 300292316

Benjamin Rousseu, 300407337

Jaures Carin Wanda Kwimi 300453467

Raoui Omar 300409570

04 avril, 2025

Université d’Ottawa

Table des matières

[Table des matières i](#_Toc66485180)

[Liste de figures iv](#_Toc66485181)

[Liste de tableaux v](#_Toc66485182)

[Liste d’acronymes et glossaire vi](#_Toc66485183)

[1 Introduction 1](#_Toc66485184)

[2 Aperçu 2](#_Toc66485185)

[2.1 Conventions 4](#_Toc66485186)

[2.2 Mises en garde & avertissements 4](#_Toc66485187)

[3 Pour commencer **Error! Bookmark not defined.**](#_Toc66485188)

[3.1 Considérations pour la configuration 6](#_Toc66485189)

[3.2 Considérations pour l’accès des utilisateurs 7](#_Toc66485190)

[3.3 Accéder au système 7](#_Toc66485191)

[3.4 Organisation du système & navigation 7](#_Toc66485192)

[3.5 Quitter le système 8](#_Toc66485193)

[4 Utiliser le système 9](#_Toc66485194)

[4.1 <Fonction/Caractéristique donnée> **Error! Bookmark not defined.**](#_Toc66485195)

[4.1.1 <Sous-fonction/Sous-caractéristique donnée> **Error! Bookmark not defined.**](#_Toc66485196)

[5 Dépannage & assistance 12](#_Toc66485197)

[5.1 Messages ou comportements d’erreur 12](#_Toc66485198)

[5.2 Considérations spéciales 12](#_Toc66485199)

[5.3 Entretien 12](#_Toc66485200)

[5.4 Assistance 13](#_Toc66485201)

[6 Documentation du produit 14](#_Toc66485202)

[6.1 <Sous-système 1 du prototype> **Error! Bookmark not defined.**](#_Toc66485203)

[6.1.1 LDM (Liste des Matériaux) 18](#_Toc66485204)

[6.1.2 Liste d’équipements 18](#_Toc66485205)

[6.1.3 Instructions 18](#_Toc66485206)

[6.2 Essais & validation 18](#_Toc66485207)

[7 Conclusions et recommandations pour les travaux futurs 19](#_Toc66485208)

[8 Bibliographie 20](#_Toc66485209)

[APPENDICES 21](#_Toc66485210)

[9 APPENDICE I: Fichiers de conception 21](#_Toc66485211)

[10 APPENDICE II: Autres Appendices **Error! Bookmark not defined.**](#_Toc66485212)

Liste de figures

Figure 1 : interface d’affichage de traduction

Figure 2 : interface des paramètres

Figure 3 : représentation graphique du système

Figure 4 : Interface utilisateur au complet

Figure 5 : interface d’affichage de traduction

Figure 6 : Interface des paramètres

Figure 7 : interface d’affichage de traduction

Figure 8 : code écrit en python avec les bibliothèques

Figure 9 : code écrit en python avec les bibliothèques

Figure 10 : code écrit en python avec les bibliothèques

Figure 11 : code écrit en python avec les bibliothèques

Liste de tableaux

[Table 1. Acronymes vi](#_Toc63684231)

[Table 2. Glossaire vi](#_Toc63684232)

[Table 3. Documents référencés 21](#_Toc63684233)

.

Liste d’acronymes et glossaire

Table 1. Acronymes

|  |  |
| --- | --- |
| **Acronyme** | **Définition** |
| API | Interface de programmation d'application |
| CPX | Conception pour X |
| ACV | Analyse du cycle de vie |
| NDM | Nomenclature des matériaux |
| SDK | Software development kit |
|  |  |

Table 2. Glossaire

|  |  |  |
| --- | --- | --- |
| **Terme** | **Acronyme** | **Définition** |
| Intelligence artificielle | IA | Une intelligence est un programme capable de résoudre des taches de façons indépendantes tout en suivant un raisonnement qu’un humain aurait suivi |

# Introduction

Ce manuel d’utilisation et de produit (MUP) fournit les informations nécessaires à <types d'utilisateurs> pour utiliser efficacement le <nom du système (acronyme)> et pour la documentation du prototype.

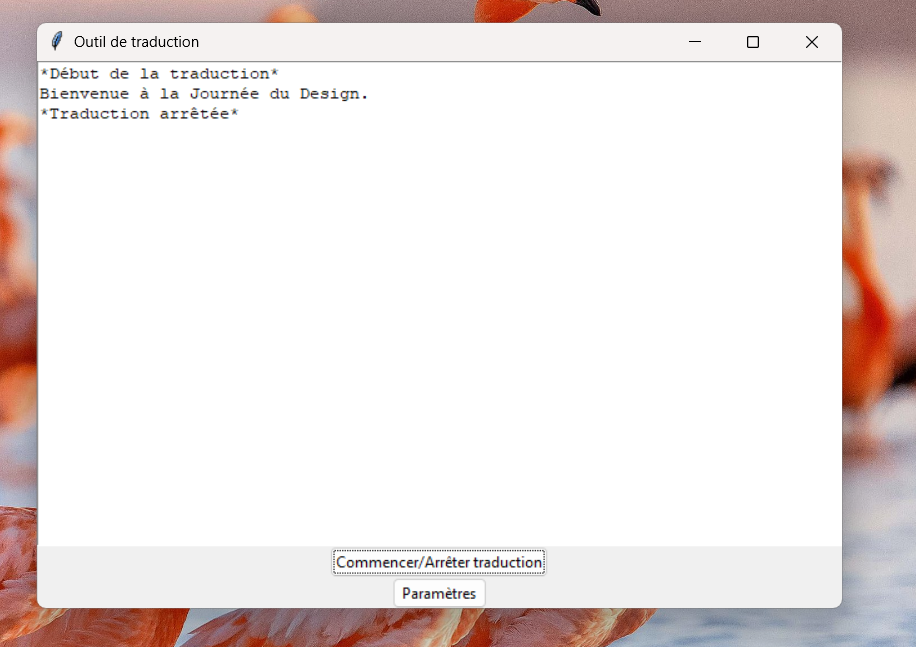
Dans le cadre d’un nouveau programme offert par l’université, il subsiste un problème de communication car des étudiants complètement francophones ont du mal à suivre en classe. Dans le thème de l’accessibilité, notre travail a consisté à évaluer les problème auxquels font face ces étudiants et qui souhaitent s’inscrire aux cours académiques uniquement offertes en anglais, de penser à des solutions potentielles qui aideront les étudiants concernés à comprendre le contenu du cours, et d’essayer d’implémenter une de nos solutions. Pour s’y rendre, nous allons passer par et considérer le développement durable (et CPX) de notre projet, nous allons analyser son cycle de vie, définir le problème, développer les concepts et planifier le projet, ainsi que faire sa conception détaillée.

# Aperçu

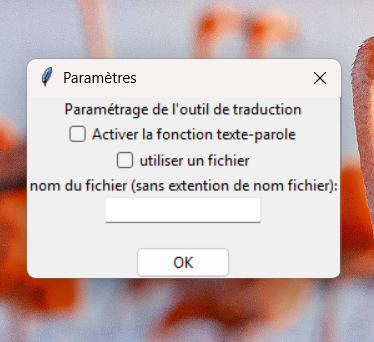
Le problème que nous abordons dans ce projet est celle de plusieurs étudiants francophones, soit la barrière linguistique anglaise dans certains cours offerts uniquement en anglais. Ce problème pose un obstacle significatif pour les étudiants qui étudient dans un programme spécifique ; cela les empêche de performer de leur mieux puisqu’ils galèrent à même comprendre les mots articulés en classe pour les matières qui n’ont pas d’équivalent français.

L’utilisateur de notre produit a principalement besoin de comprendre ce qui est articulé en classe, soit par le professeur pendant les sessions de lecture, soit par ses camarades de classe pendant les périodes de discussion.

Notre produit est de meilleure qualité que la concurrence car il est optimisé pour notre mise en situation, soit celle des étudiants francophones, tandis que les autres compagnies bien connues, tel que ChatGPT, Google Translate, DeepL... ils omettent la personnalisation, ce qui rend l’utilisation de leurs produits, pour le cas spécifique susmentionné, frustrant. Nous offrons la possibilité d’enregistrer des paroles pendant une durée prolongée des discussions/conférences complètes. En addition, nous offrons une option de télécharger toute la transcription. Finalement, nous avons aussi implémenté une option “texte-parole” pour ceux qui souhaitent écouter les paroles traduites en direct.



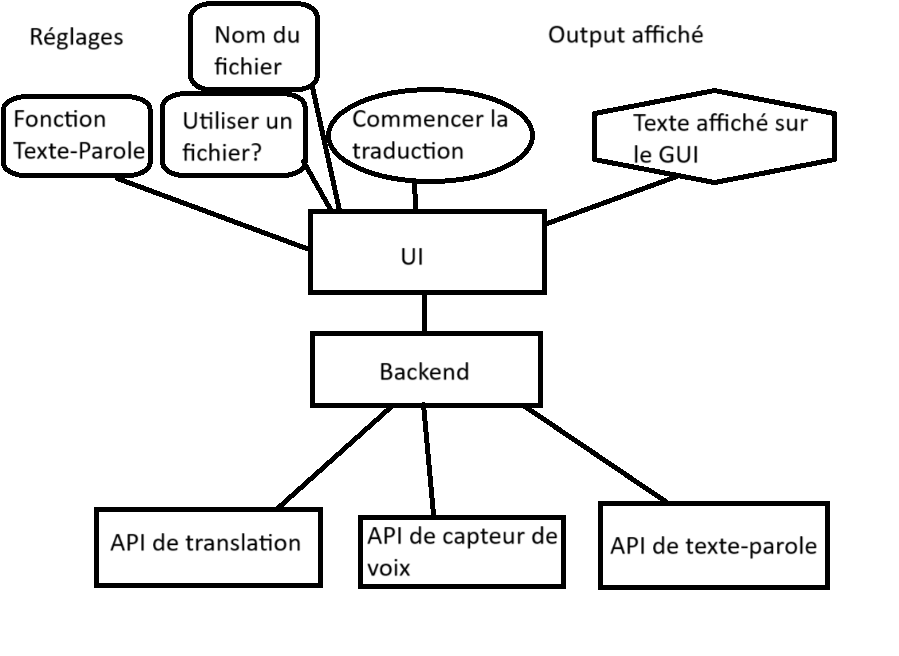
*Figure 1. Interface d’affichage de traduction*



*Figure 2.* *Interface des paramètres*

Le système est composé en deux sous-parties, le backend, et le GUI (interface utilisateur).

Notre système est présentement sous forme de fichier python exécutable (extension. Pyw) et il peut être manipulé par les utilisateurs par moyens de boutons, cases à cocher et entrées textes présents sur l’interface utilisateur (veuillez faire référence aux images ci-dessus).



*Figure 3. Représentation graphique du système*

## Conventions

Pour le cas échéant, les actions requises ayant rapport à l’utilisation du logiciel, serait principalement lié à la compilation des codes et l’utilisation des bibliothèques informatique, ce qui entrainerait l’affichage du message suivant : **ACTION : <ERREUR DE COMPILATION> ou ACTION : <VEUILLEZ ENREGISTRER LES CODES DANS UN MEME DOCUMENT>**.

## Mises en garde & avertissements

Pour utiliser le module Microsoft Speech SDK, suivre les instructions sur le site de Microsoft. Certains coûts peuvent être associés avec l’utilisation. Lien : <https://learn.microsoft.com/en-us/azure/ai-services/speech-service/get-started-speech-translation?tabs=windows%2Cterminal&pivots=programming-language-python>

Notre produit a été implémenté en utilisant Python 3 avec les librairies de Microsoft azure speech SDK et Tkinter pour l’implémentation de notre interface utilisateur.Pour le produit qu’on propose actuellement, l’exécution se fait à partir d’un fichier Python.

A l’ouverture vous verrez notre interface utilisateur :

A screenshot of a computer

AI-generated content may be incorrect.

*Figure 4. Interface utilisateur au complet*

A partir de ce menu vous sélectionner une/les options ci-dessous :

* Activer la fonction texte-parole : Elle permettra de pouvoir avoir l’option d’entendre en français ce que l’interlocuteur en anglais dit.
* Utiliser un fichier : son rôle est de garder la transcription de la traduction dans un fichier qui sera lisible.

Pour démarrer le programme, appuyez sur **Commencer/Arrêter la traduction,** vous verrez la fenêtre ci-dessous :

A screenshot of a computer

AI-generated content may be incorrect.

*Figure 5. Interface d’affichage de traduction*

Dans cette image, l’utilisateur a dit « Welcome to the Design day » et la traduction est affiché à la deuxième ligne.

Après utilisation appuyer **Commencer/Arrêter la traduction** et la fenêtre indiquera que la traduction est finie.

## Considérations pour la configuration

L’utilisation de notre logiciel requiert d’avoir un ordinateur portable à sa disposition préférablement, mais un ordinateur fixe fonctionnerait aussi parfaitement. A ce stade, notre produit final n’est encore qu’un prototype nécessitant plusieurs améliorations, mais étant très utilisable, pour ce faire, l’utilisateur devra impérativement avoir sur son ordinateur une bonne connexion internet, les codes et les bibliothèques informatiques capable d’exécuter le logiciel de traduction, ainsi que d’avoir une sortie audio qui fonctionne correctement ou préférablement d’avoir à sa disposition un microphone en bonne erat. Après cela, il suffirait juste de suivre les instructions d’utilisation.

## Considérations pour l’accès des utilisateurs

Initialement, notre produit, est capable de pouvoir traduire des paroles en anglais en texte écrit en français exclusivement, puis de pouvoir lire le texte a haute voix. Les principaux utilisateurs seraient des étudiants éprouvants des difficultés à comprendre la langue anglaise ou à pouvoir suivre des cours dispensés en anglais. Pour chaque traduction, il faudrait attendre un maximum de 30 secondes avant de voir le texte s’afficher sur l’interface utilisateur, ce qui constitue la plus grande restriction pour les utilisateurs.

## Accéder/installation du système

Pour utiliser le logiciel, il faut télécharger Python 3 (<https://www.python.org/downloads/>), ainsi que l’API Microsoft speech et ses dépendances. Il faut créer une ressource sur Microsoft pour obtenir une clé d’utilisation et créé des variables dans le system d’exploitation (instructions : <https://learn.microsoft.com/en-us/azure/ai-services/speech-service/get-started-speech-translation?tabs=windows%2Cterminal&pivots=programming-language-python>).

Il faudrait aussi télécharger notre fichier python et l’exécuter après avoir installé les fichiers mentionner ci-dessus.

## Organisation du système & navigation

Pour un prototype physique, Les termes généraux pour décrire des éléments physiques pour l'organisation du système peuvent inclure:

* La composante principale : Unité centrale ou ordinateur avec programme IA intégré.
* Les accessoires ou pièces jointes : Écran tactile pour affichage des traductions, microphones pour reconnaissance vocale, haut-parleurs pour lecture audio, caméras pour reconnaissance de texte via OCR.
* Les connexions aux fonctionnalités principales : Le microphone capture l’entrée audio puis le module de reconnaissance vocale convertit en texte en suite le module de traduction traite la phrase et enfin le résultat affiché ou lu à voix haute.  
   Une possibilité d’avoir une interface utilisateur avec des boutons physiques pour choisir la langue cible. Chaque fonctionnalité du système peut être décrite sous des sous-sections comme : “Reconnaissance vocale”, “Affichage des résultats”, “Commande vocale”.
* Pour un prototype logiciel, Les termes généraux pour décrire l’organisation logiciels pour l'organisation du système peuvent inclure:
* La composante principale : l’interface utilisateur (application mobile, site web, programme embarqué).
* Le menu système ou page d’accueil : la sélection de la langue source et cible, paramètres de traduction (voix, texte, OCR).
* Le chemin de navigation : le chargement du texte/audio puis le traitement via API d’IA et le résultat affiché ou lu.
* Accès aux paramètres pour ajuster la précision ou le style de traduction.

## Quitter le système

Pour quitter le prototype physique  
 - Éteindre l’appareil via un bouton d’arrêt sécurisé.  
 - Débrancher les périphériques (micro, écran, etc.) si nécessaire.  
 - Vérifier que les données sensibles sont effacées ou stockées correctement.  
Pour quitter le prototype logiciel  
 - Fermer l’application en sauvegardant les préférences de l’utilisateur.  
 - Déconnexion des services en ligne si nécessaire.  
 - Arrêter les processus d’IA pour libérer les ressources du système.

## Pour correctement éteindre le système, l’utilisateur devra attendre que la traduction se termine puis cliquer sur arrêter le programme dans le terminal et enfin il devra juste cliquer sur la croix rouge en haut tout à droite.

# Utiliser le système

1.EXECUTER LE PROGRAMME DU LOGICIEL

-S’assurer que tous les documents sont téléchargés

-Ouvrir le fichier prototype12.py

-Une interface utilisateur s’affiche devant vous

2.UTILISATION DE L’INTERFACE UTILISATEUR

2.1 UTLISATION DE L’INTERFACE UTLISATEUR DES PARAMETRES

-Cliquer sur la boite à gauche du texte ‘activer la fonction texte-parole’ pour activer cette fonction.

- Cliquer sur la boite à gauche du texte ‘utiliser un fichier’ pour activer cette fonction.

-Dans la boite inferieure, veuillez taper le nom du fichier qui gardera les paroles traduites.

2.2 UTLISATION DE L’INTERFACE UTLISATEUR POUR LA TRANSCRIPTION

-Cliquer sur le bouton ’Commencer/arrêter traduction’ pour commencer l’enregistrement de la traduction en continu.

- Cliquer sur le bouton ’Commencer/arrêter traduction’ pour arrêter l’enregistrement -

de la traduction en continu.

Les sous-sections suivantes fournissent des instructions détaillées, étape par étape, sur la façon d'utiliser les diverses fonctions ou caractéristiques de notre application de traduction.

## Fichier du programme

Le fichier Python de notre programme portera le nom de prototype12. Par défaut il se trouvera dans le dossier des téléchargements (Voir l’image en dessous).



En exécutant ce fichier, on tombera sur notre interface utilisateur.

### Menu d’accueil

Il est constitué de la fenêtre de paramétrage ainsi que de la fenêtre d’affichage du texte.

La fenêtre d’affichage

\A screenshot of a computer

AI-generated content may be incorrect.

*Figure 6. Interface des paramètres*

### Cette fenêtre vous pernet de choisir les modes dont vous aurez besoins pour exécuter le programme. Pour son utilisation, rendez-vous au début de la section 3.

A screen shot of a computer

AI-generated content may be incorrect.

*Figure 7. Interface d’affichage de traduction*

Cette fenêtre sert d’interface de communication entre le code et l’utilisateur, les instructions d’utilisation sont au début de la section 3.

# Dépannage & assistance

Si votre application de traduction ne démarre pas, plusieurs causes sont possibles. Si vous obtenez une erreur indiquant que le fichier est introuvable, vérifiez que vous vous trouvez bien dans le bon dossier et que le nom du fichier est correct. Un message signalant un module manquant signifie que certaines bibliothèques nécessaires ne sont pas installées sur votre ordinateur. Les erreurs de syntaxe, quant à elles, apparaissent clairement dans le terminal et indiquent la ligne problématique il suffit généralement de corriger les caractères manquants comme les deux points ou les parenthèses. Si le programme plante sans message d'erreur, essayez de télécharger à nouveau les fichiers originaux. Sur certains systèmes, vous devrez peut-être vérifier les permissions d'exécution du fichier ou lancer le terminal avec des droits administrateur. Enfin, assurez-vous que votre version de Python est compatible avec le programme. Si le problème persiste après ces vérifications, contactez l'assistance technique en leur communiquant : le message d'erreur complet (le cas échéant), votre système d'exploitation (Windows, Mac, Linux) et la version de Python installée sur votre machine.

## Messages ou comportements d’erreur

Erreur de compilation, erreur d’index, erreur de clé, erreur de type, erreur de variable, erreur de syntaxe.

## Considérations spéciales

En cas d’échéance l’utilisateur devra juste relancer le code si ça ne fonctionne toujours pas alors dans ce cas il faudra debugger le code. Pour le faire, l’utilisateur devra lancer le programme mais cette fois ci en cochant la case de débogage, quand le programme relancera vous recevrez un message d’erreur avec les erreurs possible dans le code et enfin vous n’aurez qu’à suivre les instructions et relancer le programme.

## Entretien

Pour garantir le bon fonctionnement de l’algorithme il faudra régulièrement vérifier que tous les modules python requis son a jour et que vous utilisez une version qui fonctionne toujours si ce n’est pas le cas je vous prie de supprimer la version installer et de réinstaller une nouvelle version afin d'être dans les meilleures conditions possibles. Pour s’assurer que tout se passe pour le mieux penser à vider le cache de votre appareil. Ces simples précautions vous permettront de tenir à jour l’algorithme et d’en profiter dans les meilleures situations possibles.

## Assistance

Lire la documentation fournie dans la bibliographie pour de l’assistance avec le code.

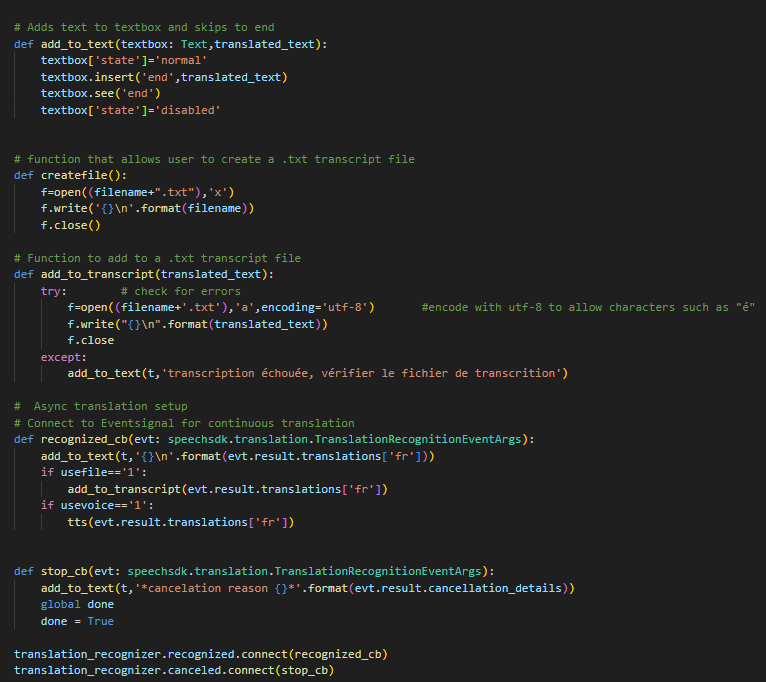
# Documentation du produit

Le code a été écris avec python dans VS code. Les bibliothèques utilisées sont Tkinter et Microsoft speech SDK. Les tests effectués ont été une combinaison de dicter des passages au logiciel pour observer le fonctionnement du system, ainsi que du débogage.

A screen shot of a computer program

AI-generated content may be incorrect.

*Figure 8. Code écrit en python avec les bibliothèques*



*Figure 9. Code écrit en python avec les bibliothèques*

A screen shot of a computer program

AI-generated content may be incorrect.

*Figure 10. Code écrit en python avec les bibliothèques*

A screen shot of a computer program

AI-generated content may be incorrect.

*Figure 11. Code écrit en python avec les bibliothèques*

## Systèmes du logiciel

### NDM (Nomenclature des Matériaux)

* Python 3 : gratuit
* Tkinter : gratuit, partie de la bibliothèque de base de python
* Microsoft speech SDK : gratuit pour une utilisation limitée
  + Instructions pour téléchargement : https://docs.azure.cn/en-us/ai-services/speech-service/get-started-speech-translation?tabs=windows%2Cterminal&pivots=programming-language-python
  + Prix sur le site de Microsoft : https://azure.microsoft.com/en-us/pricing/details/cognitive-services/speech-services/

### Liste d’équipements

* L’application VS code est utilisé pour programmer le logiciel

### Instructions utiliser le code publié sur Maker Repo : <https://makerepo.com/Josephkay/2522>

## Essais & validation

Pour tester le code, des paroles ont été réciter pour observer le fonctionnement du système. Du débogage a été effectuer pour trouver et régler des problèmes.

# Conclusions et recommandations pour les travaux futurs

Nous avons appris, sur le front technique, comment exécuter le code pour l’interface utilisateur et du backend de manière asynchrone. En plus, les bibliothèques à utiliser qui semblent être le plus optimisés pour notre cas d’utilisation est Microsoft Azure pour la fonctionnalité de traduction, et Tkinter, pour le GUI. D’un point de vue générale, nous avons tous appris de l’importance de la communication dans un cadre d’équipe en plus de la planification.

Pour les travaux futurs sur ce projet, je recommanderais mettre vos efforts sur la vitesse de traduction et d’affichage du texte, puisqu’il s’agit d’un de nos CPX qui peuvent voir une amélioration significative. Si le temps nous le permettait, nous aurions changé le fichier à un fichier exécutable (.exe). Autrement, au sujet des pistes à prendre, vous devriez rendre disponible cette application sur plusieurs appareils, le plus que possible, ce qui implique de recréer notre application en utilisant d’autres langues de programmation.

# Bibliographie

Information pour téléchargements

* <https://www.python.org/downloads/>
* <https://azure.microsoft.com/en-us/pricing/details/cognitive-services/speech-services/>
* https://docs.azure.cn/en-us/ai-services/speech-service/get-started-speech-translation?tabs=windows%2Cterminal&pivots=programming-language-python

Microsoft Learn articles by Eric Urban and Nitin Mehrotra

* + https://learn.microsoft.com/en-us/azure/ai-services/speech-service/how-to-translate-speech?tabs=terminal&pivots=programming-language-python
  + https://learn.microsoft.com/en-us/azure/ai-services/speech-service/how-to-recognize-speech?pivots=programming-language-python
  + https://learn.microsoft.com/en-us/azure/ai-services/speech-service/how-to-speech-synthesis?tabs=browserjs%2Cterminal&pivots=programming-language-python
  + https://learn.microsoft.com/en-us/azure/ai-services/speech-service/get-started-text-to-speech?tabs=windows%2Cterminal&pivots=programming-language-python

Microsoft Speech documentation

* + - https://learn.microsoft.com/en-us/python/api/azure-cognitiveservices-speech/azure.cognitiveservices.speech?view=azure-python

Speech recognition samples for the Microsoft Cognitive Services Speech SDK

* + - https://github.com/Azure-Samples/cognitive-services-speech-sdk/blob/master/samples/python/console/speech\_sample.py

Tkinker tutorial

* + - https://docs.python.org/3/library/tkinter.html
    - https://tkdocs.com/tutorial/onepage.html

APPENDICES

# APPENDICE I: Fichiers de conception

Table 3. Documents référencés

|  |  |
| --- | --- |
| **Nom du document** | **Emplacement du document et/ou URL** |
| Page Makerepo | https://makerepo.com/Josephkay/2522 |
|  |  |
|  |  |
|  |  |
|  |  |