

Université Sorbonne Paris Nord  
Sup Galilée  
Spécialité Informatique



## Conduite et gestion de projet

Spécialité : INFO2

Sprint 3

---

# Interface de Génération sonore par IA

---

***Étudiants :***

Seyfeddine JOUINI  
Yosra SASSI  
Sirine TLILI  
Hasna ELGARANI  
Kaoutar BRAHIMI

***Client :***  
Vittascience

6 février 2025

## Table des matières

<b>Introduction.....</b>	<b>2</b>
<b>1. Recherche et analyse technologique.....</b>	<b>2</b>
<b>2. Description détaillée des cas prioritaires.....</b>	<b>2</b>
2.1. Cas d'utilisation 1 : Génération de speech à partir d'un prompt.....	2
2.2. Cas d'utilisation 2 : Lipsync à partir d'un audio et d'une image.....	4
2.3. Cas d'utilisation 3 : Speech-to-Speech.....	5
<b>3. Avancement sur les cas prioritaires.....</b>	<b>7</b>
3.1. Cas d'utilisation 1 : Génération de speech à partir d'un prompt.....	7
3.2. Cas d'utilisation 2 : Génération de texte à partir d'une entrée vocale.....	8
3.3. Cas d'utilisation 3 : Lipsync à partir d'un audio et d'une image.....	8
3.4. Cas d'utilisation 4 : Speech-to-text.....	9
<b>4. Prototype.....</b>	<b>9</b>
4.1. Cas d'utilisation 1 : Génération de speech à partir d'un prompt.....	9
4.2. Cas d'utilisation 2 : Lipsync à partir d'un audio et d'une image.....	11
4.3. Cas d'utilisation 3 : Speech-to-Speech.....	12
<b>5. Conception des cas d'utilisation restants.....</b>	<b>13</b>
5.1. Text-to-Speech (version2).....	13
5.2. Génération de Musique.....	13
<b>6. Défis et solutions.....</b>	<b>14</b>
6.1. Text-to-Speech.....	14
6.2. Speech-to-Text.....	14
6.3. Lipsync.....	14
6.4. Speech-to-Speech.....	14
<b>7. Diagramme de Gantt.....</b>	<b>15</b>
<b>Conclusion.....</b>	<b>16</b>

## Introduction

Ce rapport présente les activités réalisées durant le Sprint 3 du projet de développement d'un module de génération sonore par l'intelligence artificielle. Tout d'abord, une recherche approfondie a été menée pour identifier et analyser les technologies adaptées à chaque cas d'utilisation. Ensuite, des descriptions détaillées des cas prioritaires ont été élaborées. Par la suite, les progrès réalisés dans ce sprint ont été présentés, et enfin, un prototype a été conçu pour valider les solutions techniques, accompagné d'une analyse des principaux défis rencontrés et des solutions apportées.

## 1. Recherche et analyse technologique

Dans cette section, nous nous sommes concentrés sur l'identification et l'analyse des technologies possibles pour réaliser le projet de génération sonore par IA. Notre approche a consisté à regrouper les technologies adaptées à des cas d'utilisation, à en analyser les avantages et les inconvénients, et à construire une carte conceptuelle qui illustre clairement les options disponibles.

Lien de la carte conceptuelle : [https://miro.com/app/board/uXjVLGnXeWk=](https://miro.com/app/board/uXjVLGnXeWk=/)

### Critères de choix technologique

Après discussion avec l'équipe et notre client, les critères suivants ont été fixés pour guider le choix des technologies :

- Fonctionner localement sans connexion Internet (offline).
- Être compatible avec JavaScript pour une intégration facile.
- Offrir des performances suffisantes pour des cas d'utilisation interactifs.
- Permettre la manipulation de paramètres sonores (exemple : hauteur, vitesse, etc..).

Cette analyse a permis d'identifier les technologies les plus adaptées pour chaque cas d'utilisation, tout en respectant les contraintes techniques et fonctionnelles définies. Ces choix guideront la conception et le développement des prototypes à venir.

## 2. Description détaillée des cas prioritaires

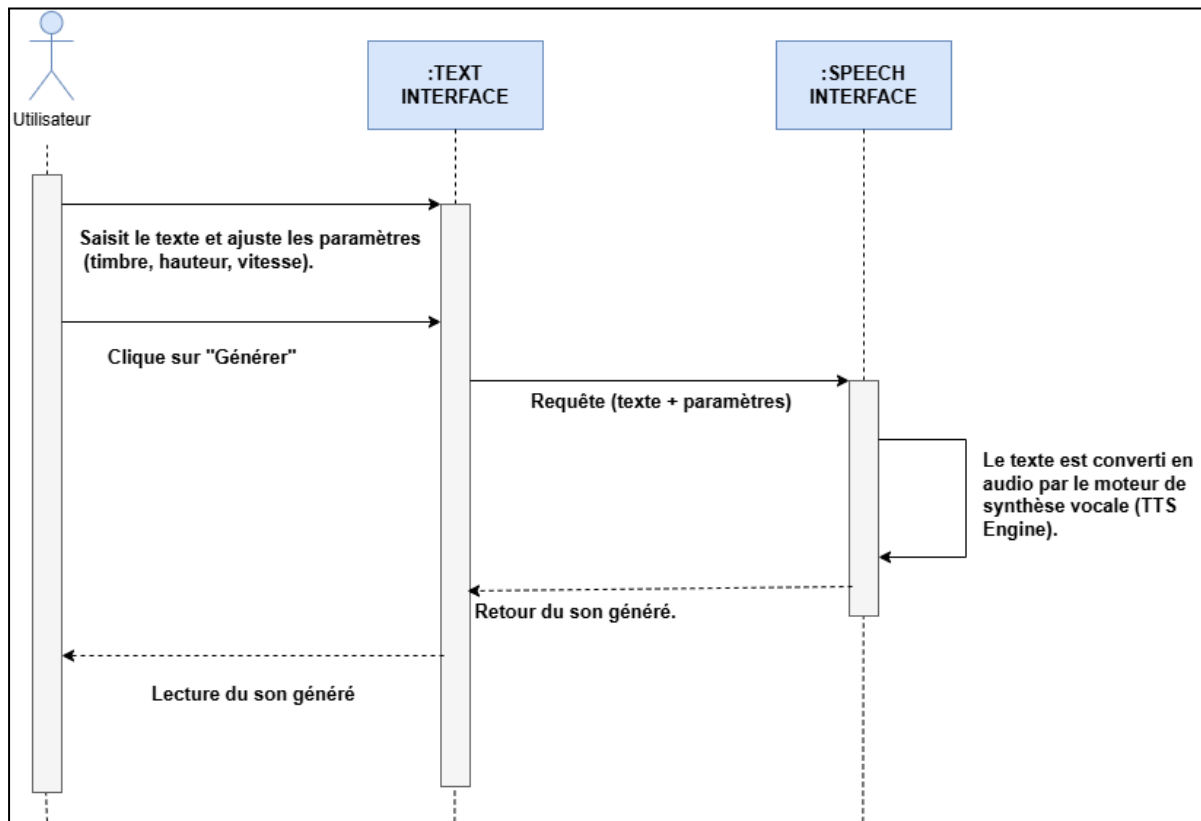
### 2.1. Cas d'utilisation 1 : Génération de speech à partir d'un prompt

#### Objectif :

Ce cas d'utilisation vise à permettre à un utilisateur de générer un son synthétique réaliste à partir d'un texte saisi (Text-To-Speech), avec des paramètres personnalisables tels que le type de voix (homme ou femme), la hauteur et la vitesse. Cette fonctionnalité répond au besoin de transformer du texte en voix de manière fluide et intuitive, adaptée à des usages pédagogiques ou créatifs.

**Scénario utilisateur :**

1. L'utilisateur ouvre l'interface de génération.
2. Il saisit un texte dans le champ prévu à cet effet.
3. L'utilisateur ajuste les paramètres de la voix (par exemple : hauteur, vitesse, etc..).
4. Il clique sur le bouton "Générer".
5. Le système traite la requête et génère un fichier audio correspondant au texte saisi.
6. Le fichier audio est restitué à l'utilisateur, qui peut l'écouter directement.

**Diagramme de séquence:****Technologies :****1. API Web Speech (SpeechSynthesis) :**

- **Description** : L'API Web Speech est une fonctionnalité native des navigateurs modernes qui permet de synthétiser de la voix à partir de texte sans dépendre d'une connexion Internet ou de serveurs externes.
- **Avantages** :
  - Fonctionne localement (offline) sur les navigateurs supportant cette API.
  - Permet de personnaliser les paramètres vocaux (hauteur, vitesse, etc..).
  - Facilité d'intégration dans le projet du notre client..
- **Inconvénients** :
  - Support limité selon le navigateur utilisé (incompatibilité avec certains navigateurs obsolètes).
  - Options de voix dépendantes des voix disponibles sur le système de l'utilisateur.

## 2. JavaScript :

- **Description** : Utilisé pour développer la logique de l'application et interagir avec l'API Web Speech.

## 3. HTML et CSS :

- **Description** : Utilisés pour construire l'interface utilisateur et améliorer l'expérience utilisateur.

### 2.2. Cas d'utilisation 2 : Lipsync à partir d'un audio et d'une image

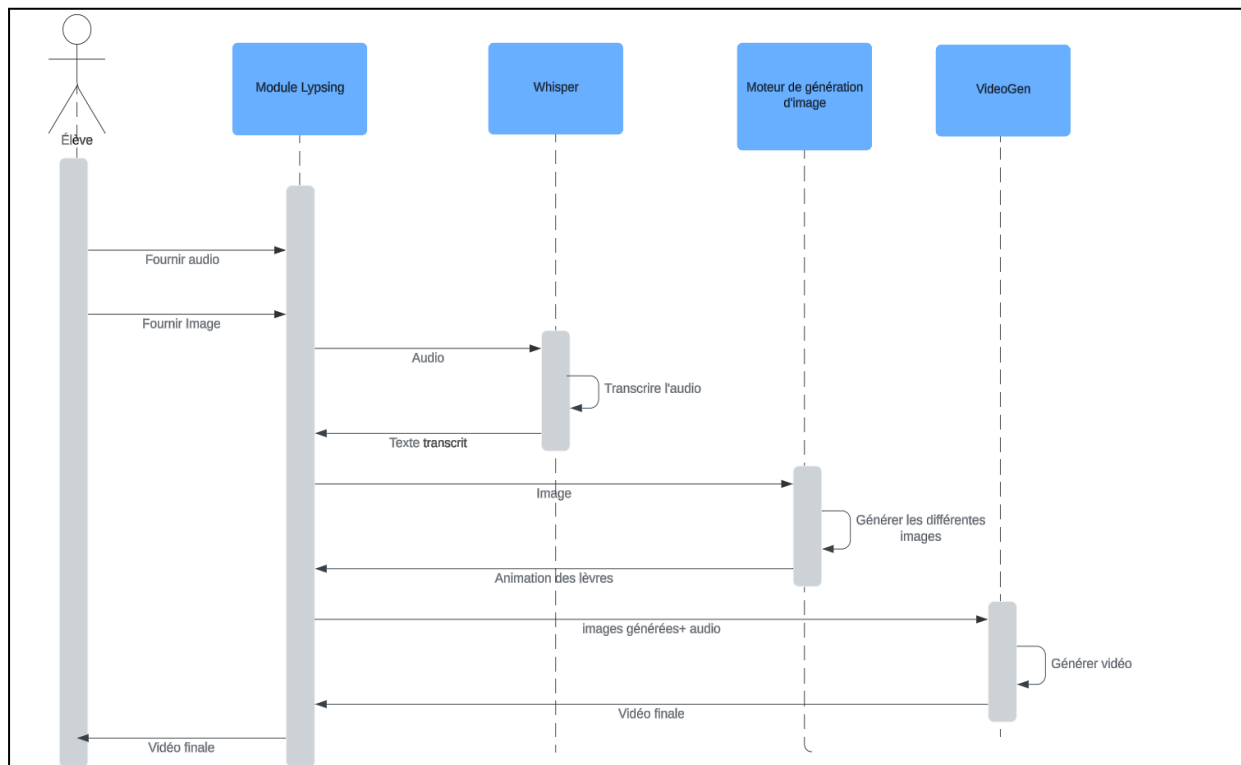
#### Objectif :

Cette fonctionnalité permet à l'utilisateur d'animer une image contenant un visage en synchronisant les mouvements des lèvres avec un dialogue ou une chanson d'un audio fourni, pour créer l'illusion réaliste que le personnage parle ou chante.

#### Scénario utilisateur :

1. Fournir une image contenant un visage et un audio.
2. Le système transcrit l'audio.
3. Le système génère les différentes formes du visage (bouche ouverte/ bouche ronde ...).
4. Le système créer la vidéo en faisant la correspondance entre les lettres du texte transcrit et les images.
5. La vidéo est restituée à l'utilisateur.

#### Diagramme de séquence:



## Technologies :

### 1. Python :

- **Description** : Utilisé pour le développement de l'intégralité de la fonctionnalité (tout fonctionne en off-line)
- **Remarque** : Dans cette partie il y a eu des changements par rapport à l'autre fois afin d'améliorer les performances du système.
- **Changement** : Remplacement du modèle **Vosk** par la bibliothèque **Whisper** .

## Bibliothèques utilisées :

1. **Whisper**: C'est une bibliothèque Python qui permet la transcription d'un audio.
2. **Numpy**: C'est une bibliothèque utilisée pour le traitement d'images qui nous a permis la génération des différentes images.

**Remarque** : Les fichiers audio peuvent être de type : mp3, mp4, mpeg, mpga, m4a, wav et webm.

## 2.3. Cas d'utilisation 3 : Speech-to-Speech

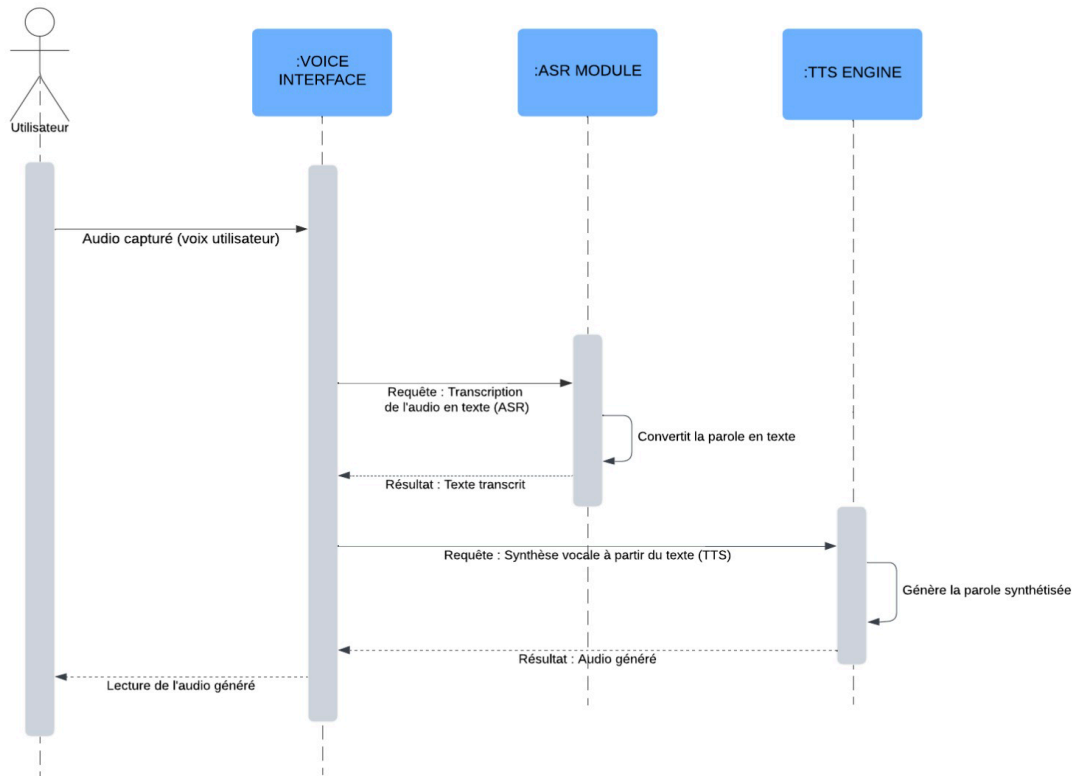
### Objectif :

Ce cas d'utilisation vise à permettre à un utilisateur de générer une voix synthétique réaliste à partir d'un enregistrement audio court, en utilisant les caractéristiques vocales extraites pour créer une synthèse vocale personnalisée. Cette fonctionnalité répond au besoin de créer des voix synthétiques naturelles et adaptables, même pour des langues peu dotées en ressources.

### Scénario utilisateur :

1. L'utilisateur ouvre l'interface de génération vocale.
2. Il enregistre un court échantillon audio de sa voix (environ 5 secondes).
3. Le système extrait automatiquement les caractéristiques vocales de l'échantillon.
4. L'utilisateur clique sur "Transcrire" pour voir le texte qu'il a dit .
5. L'utilisateur clique sur "Synthétiser".
6. Le système génère un fichier audio correspondant au texte saisi avec la voix imitée.
7. L'utilisateur peut écouter le résultat directement dans l'interface.

## Diagramme de séquence:



## Technologies :

1. **Python** : Langage de programmation utilisé pour intégrer les différentes technologies et développer la logique de l'application.

## Bibliothèques utilisées :

### 1. Whisper :

Description : Modèle de reconnaissance vocale développé par OpenAI, utilisé pour la transcription audio.

Avantages :

- Haute précision dans la transcription, même pour des accents variés.
- Supporte de nombreuses langues.

Inconvénients :

- Peut nécessiter des ressources de calcul importantes pour le traitement en temps réel.

### 2. Librosa :

Description : Bibliothèque Python pour l'analyse audio et l'extraction de caractéristiques vocales.

Avantages :

- Offre des fonctions avancées pour l'extraction de caractéristiques comme le pitch.
- Bien documentée et largement utilisée dans la communauté du traitement audio.

Inconvénients :

- Peut être lente pour le traitement en temps réel de grands fichiers audio.

### 3. YourTTS :

Description : Modèle de synthèse vocale zero-shot multi-locuteur basé sur l'architecture VITS, capable de générer des voix naturelles à partir d'un court échantillon audio.

Avantages :

- Permet la synthèse vocale pour de nouveaux locuteurs sans réentraînement.
- Supporte plusieurs langues et le transfert de voix entre langues.
- Nécessite peu de données pour s'adapter à une nouvelle voix (20-60 secondes suffisent).

Inconvénients :

- Peut nécessiter des ressources de calcul importantes pour le traitement en temps réel.

## 3. Avancement sur les cas prioritaires

### 3.1. Cas d'utilisation 1 : Génération de speech à partir d'un prompt

#### Activités réalisées

- Création d'une application pour tester une API Text-to-Speech (TTS) fonctionnant localement sans dépendre d'une connexion Internet.
- Développement d'une interface utilisateur intuitive permettant de saisir un texte, d'ajuster les paramètres (hauteur, vitesse, volume) et de sélectionner la langue et la voix(homme/femme).

#### Résultat

- Application fonctionnelle, testée avec succès sur les navigateurs modernes prenant en charge l'API Web Speech.
- Les paramètres ajustables fonctionnent correctement, offrant une personnalisation vocale précise.

### 3.2. Cas d'utilisation 2 : Génération de texte à partir d'une entrée vocale

#### Activités réalisées



- Test de plusieurs API pour identifier une solution performante et fonctionnant hors ligne en JavaScript.
- Évaluation des performances de SpeechRecognition et exploration des alternatives comme Vosk.js.

### Résultat

- Une solution partiellement fonctionnelle a été développée et testée.
- Des contraintes techniques persistent pour un usage hors ligne optimal, nécessitant des ajustements et une éventuelle optimisation des modèles utilisés.

### 3.3. Cas d'utilisation 3 : Lipsync à partir d'un audio et d'une image

#### Activités réalisées

- Exploration des outils et technologies capables de synchroniser des mouvements labiaux avec un fichier audio.
- Recherche sur les bibliothèques disponibles pour le traitement audio et l'analyse des phonèmes, notamment pour mapper les phonèmes de l'audio avec des animations faciales réalistes.
- Transcription d'audio réussite en utilisant Whisper.
- Synchronisation des images avec l'audio dans une vidéo.

### Résultat

- Fonctionnalité testée permettant d'obtenir une vidéo animée avec les lèvres synchronisées avec l'audio.
- Les images générées montrent un changement au niveau des lèvres cependant ce n'est pas encore au point.

### 3.4. Cas d'utilisation 4 : Speech-to-Speech

#### Activités réalisées :

- Implémentation d'un système d'enregistrement audio en temps réel utilisant la bibliothèque sounddevice.
- Extraction réussie des caractéristiques vocales clés (pitch, timbre, vitesse de parole) à l'aide de librosa.
- Intégration du modèle Whisper d'OpenAI pour la transcription audio vers texte.
- Mise en place d'un pipeline de synthèse vocale utilisant le modèle YourTTS via l'API TTS.

- Développement d'une fonction pour ajuster la vitesse de la voix synthétisée en fonction du tempo détecté.
- Création d'un flux de travail complet allant de l'enregistrement audio à la synthèse vocale personnalisée.

**Résultat :**

- Système fonctionnel capable d'enregistrer un court échantillon audio, d'extraire ses caractéristiques vocales, de transcrire le contenu, et de générer une voix synthétique basée sur ces caractéristiques.
- La voix synthétisée reflète certaines caractéristiques de la voix originale, notamment en termes de vitesse et de tonalité générale.
- Les fichiers audio générés montrent une amélioration de la naturalité par rapport aux systèmes TTS standard, mais il reste une marge d'amélioration en termes de réalisme et de fidélité à la voix originale.

## 4. Prototype

### 4.1. Cas d'utilisation 1 : Génération de speech à partir d'un prompt

**Fonctionnalités techniques de l'application :**

- **Personnalisation des voix :**
  - L'utilisateur peut choisir parmi différentes voix (masculines et féminines) disponibles sur son système.
  - Les voix sont filtrées selon la langue sélectionnée (Français ou Anglais).
- **Paramètres ajustables :**
  - **Hauteur (pitch) :** Contrôle le timbre de la voix pour des variations émotionnelles.
  - **Vitesse (rate) :** Permet de ralentir ou d'accélérer la lecture.
  - **Volume :** Ajuste l'intensité sonore de la voix générée.
- **Compatibilité linguistique :**
  - Support pour le Français et l'Anglais grâce à un menu de sélection des langues.
  - Résultats obtenus ou problèmes identifiés lors de cette conception.

## Application Text-to-Speech

Transformez vos textes en audio pour l'apprentissage et l'éducation.

Saisissez votre texte :

Tapez votre texte ici...

Langue :

Français

Voix disponible :

Femme - Français

Hauteur (Pitch) :

Vitesse (Speed) :

Volume :

Lancer la lecture

### Avantages de cette approche technologique :

- **Exécution locale (offline)** : Aucune connexion Internet n'est requise, ce qui améliore la confidentialité des données et la rapidité d'exécution.
- **Simplicité d'utilisation** : Une interface intuitive permet à l'utilisateur de générer du son en quelques clics.
- **Flexibilité et personnalisation** : Les paramètres ajustables offrent une expérience utilisateur adaptée à des besoins variés (éducation, narration, etc.).
- **Open source et extensibilité** : Le code basé sur JavaScript et l'API Web Speech peut être facilement étendu pour inclure des fonctionnalités supplémentaires (exemple : nouvelles langues, nouvelles voix).

## 4.2. Cas d'utilisation 2 : Lipsync à partir d'un audio et d'une image

### Fonctionnalités techniques de l'application :

Dans le sprint 2, on mettait en oeuvre la fonctionnalité de Lipsync comme suit :

- **Transcription de l'audio :**

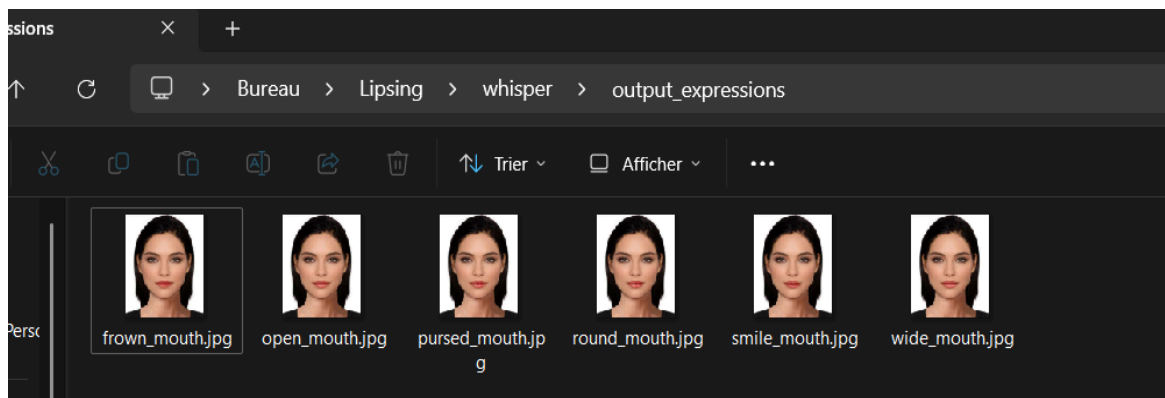
La transcription de l'audio se fait selon le modèle Whisper comme le montre la figure suivante :

```
PS C:\Users\joun\Desktop\Lipsing\whisper> python .\transcript.py
C:\Users\joun\AppData\Local\Programs\Python\Python312\Lib\site-packages\whisper\_init_.py:150: FutureWarning: You are using 'torch.load' with 'weights_only=False', which uses the default pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See https://docs.python.org/3/library/pickle.html#security-considerations for more details). In a future release, the default value for 'weights_only' will be flipped to 'True'. This limits the objects that can be loaded, and doesn't allow loading arbitrary objects. It is recommended you start setting 'weights_only=True' for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any feedback.
  checkpoint = torch.load(fp, map_location=device)
Transcription:
Vous écoutez la voie mondiale. Bienvenue pour fêter notre 6e anniversaire. Vous écoutez de Global Voice. Bienvenue pour fêter notre 6e anniversaire.
PS C:\Users\joun\Desktop\Lipsing\whisper>
```

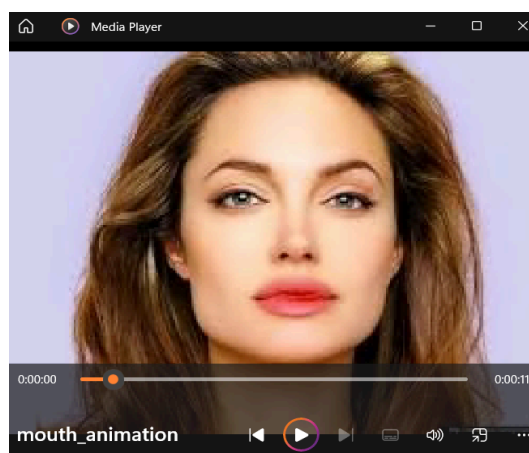
La figure ci-dessous illustre la transcription d'un audio.

- **Génération des expressions du visage :**

L'animation des lèvres se fait par le processus de génération d'images avec les différentes formes des lèvres de la bouche.



### Le résultat d'une bouche souriante :



Comme on le voit dans la figure, le résultat n'est pas optimal. De ce fait, on a essayé de trouver une alternative qui permet de mieux animer les lèvres.

**Alternative :** Utilisation d'un modèle existant : **Wav2Lip**

Ce modèle permet de faire la synchronisation de la voie avec l'animation des lèvres.

Le lien github de ce modèle : <https://github.com/Rudrabha/Wav2Lip>

### 4.3. Cas d'utilisation 3 : speech-to-speech

**Fonctionnalités techniques de l'application :**

1. Enregistrement audio personnalisé :
  - L'utilisateur peut enregistrer un court échantillon de sa voix (environ 5 secondes).
  - Le système capture l'audio en temps réel .
2. Extraction des caractéristiques vocales :
  - Analyse du pitch (F0) utilisant l'algorithme pYIN de librosa.
  - Calcul des coefficients MFCC pour capturer le timbre de la voix.
  - Estimation de la vitesse de parole via la détection du tempo.
3. Transcription audio :
  - Utilisation du modèle Whisper d'OpenAI pour transcrire l'audio en texte.
  - Support multilingue pour la transcription.
4. Synthèse vocale personnalisée :
  - Utilisation du modèle YourTTS pour générer une voix synthétique.
  - Adaptation de la synthèse aux caractéristiques vocales extraites.

**Résultats :**

- Système fonctionnel capable d'enregistrer, analyser, et synthétiser des voix personnalisées.
- Bonne qualité de transcription audio grâce à Whisper.
- Synthèse vocale adaptative reflétant certaines caractéristiques de la voix originale.

**Problèmes identifiés :**

- Latence dans le traitement, en particulier lors de l'utilisation de Whisper et YourTTS.
- Qualité de la synthèse vocale variable selon les caractéristiques de la voix d'origine.
- Besoin d'optimisation pour le traitement en temps réel sur des systèmes moins puissants.

## 5. Conception des cas d'utilisation restants

### 5.1. Text-To-Speech ( Version 2 )

Après la validation de la **Version 1**, qui permettait de générer un audio à partir d'un texte de manière fluide et personnalisable, nous devons maintenant ajouter la fonctionnalité de **sauvegarde de l'audio généré**.

#### Objectif

- Permettre aux utilisateurs de télécharger l'audio généré sous un format adapté (MP3, WAV, OGG) tout en garantissant une qualité sonore optimale et une conversion rapide.

#### Prochaines étapes

- Étudier les formats audio les plus adaptés en termes de qualité et de compatibilité.
- Explorer les bibliothèques et API permettant l'enregistrement et l'exportation de fichiers audio.
- Concevoir une interface permettant de télécharger facilement l'audio après la génération.

#### Résultat attendu

- Mise en place d'une solution permettant de sauvegarder et télécharger l'audio généré sans latence excessive.
- Optimisation du processus pour garantir une expérience utilisateur fluide.

### 5.2. Génération de Musique

Ce cas propose une extension créative du système, permettant la génération automatique de compositions musicales personnalisées en fonction des paramètres définis par l'utilisateur, tels que le style, la durée et le tempo.

#### Activités réalisées

- Recherche approfondie sur des outils IA comme Magenta (Google), Jukebox (OpenAI), et VQ-VAE-2 pour la composition musicale.

#### Conception

- Définition des paramètres utilisateur personnalisables (style musical, tonalité, durée).
- Développement d'un modèle IA permettant de générer une composition musicale cohérente à partir des préférences de l'utilisateur et, si nécessaire, d'un texte fourni.
- Validation des résultats via des tests d'écoute et ajustement des paramètres en fonction des retours.

#### Résultat attendu

- Génération d'un fichier musical simple, conforme aux paramètres choisis par l'utilisateur, et adapté à des styles préconfigurés.

## 6. Défis et solutions

### 6.1. Text-to-Speech

Un des principaux défis rencontrés lors de ce sprint concerne la compatibilité limitée de l'API Web Speech avec certains navigateurs. Cette restriction peut entraîner des difficultés d'accès pour certains utilisateurs. Pour y remédier, l'application a été optimisée et testée sur les navigateurs les plus largement utilisés, en veillant à garantir une expérience fluide et cohérente tout en restant entièrement basée sur JavaScript.

### 6.2. Speech-To-Text

Contrairement à la partie TTS, la reconnaissance vocale en JavaScript hors ligne est plus complexe, car l'API Web Speech, bien que disponible dans certains navigateurs, nécessite une connexion Internet, limitant ainsi son utilisation en mode déconnecté. Une implémentation totalement hors ligne requiert des bibliothèques comme Vosk.js, qui utilise WebAssembly pour exécuter des modèles de reconnaissance vocale en local. Toutefois, ces modèles impliquent des contraintes techniques importantes, notamment le chargement de fichiers volumineux (modèles acoustiques et linguistiques) et une consommation élevée de ressources (mémoire et CPU), ce qui peut restreindre leur usage sur des appareils moins puissants ou dans des environnements de navigateur limités.

### 6.3. Lipsync

L'exécution du modèle se fait sur le GPU CUDA qui est limité à 4Go alors que une image et un audio simple et de petite taille demandent plus d'espace pour s'exécuter.

De ce fait, on a diminué la résolution de l'image ce qui a diminué l'espace requis pour le traitement. Lors de notre dernière réunion avec le client, on a vérifié les licences et les autorisations de chaque modèle et outils utilisés. On s'est rendu compte que la licence ne nous permet pas de l'utiliser dans un besoin commercial ce qui ne nous convient pas.

Dans ce cas, on doit trouver une autre alternative.

### 6.4. Speech-to-Speech

Suite aux nouvelles exigences du client, notre projet de synthèse vocale fait face à plusieurs défis majeurs nécessitant une refonte partielle de l'architecture technique. Ces défis sont les suivants :

#### 1. Remplacement du moteur de synthèse vocale

La licence de YourTTS n'étant pas compatible avec la commercialisation du projet, nous devons trouver une alternative répondant aux critères suivants :

- Licence permettant une utilisation commerciale.
- Qualité vocale naturelle, supérieure à celle obtenue avec Mozilla TTS.
- Compatibilité avec l'environnement Python existant.
- Capacités de personnalisation et d'adaptation aux caractéristiques vocales extraites.

#### 2. Optimisation des performances

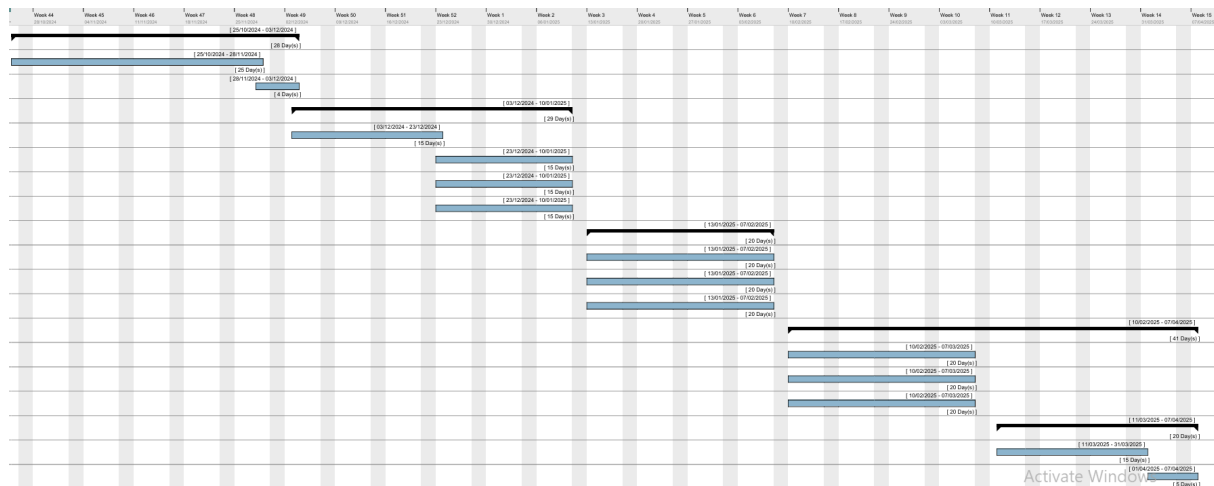
- Amélioration du traitement en temps réel, en particulier pour l'utilisation de Whisper et du nouveau moteur de synthèse vocale.
- Optimisation de l'extraction des caractéristiques vocales pour réduire la latence.

7. Diagramme de Gantt

Le diagramme de Gantt ci-dessous illustre la planification de notre projet. Il détaille les différentes tâches, leur répartition dans le temps ainsi que les dépendances entre elles. Ce planning permet d’assurer une gestion efficace des ressources et de suivre l’avancement du projet de manière structurée.

Name	Begin date	End date	C...	Canl...	New c...
Sprint 1	25/10/2024	03/12/2024			
Initialisation du projet et analyse des besoins / cahier d charge	25/10/2024	28/11/2024			
Développement d'une Démo pour le module TTS	28/11/2024	03/12/2024			
Sprint 2	03/12/2024	10/01/2025			
Recherche et analyse technologique MindMap	03/12/2024	23/12/2024			
Développement et intégration du prototype du module TTS	23/12/2024	10/01/2025			
Développement et intégration du prototype du module Lypsing	23/12/2024	10/01/2025			
Recherche et Développement d'une Démo pour le module STS	23/12/2024	10/01/2025			
Sprint 3	13/01/2025	07/02/2025			
Développement et intégration du prototype du module STS	13/01/2025	07/02/2025			
Finalisation du module Lypsync	13/01/2025	07/02/2025			
Finalisation du module TTS	13/01/2025	07/02/2025			
Sprint 4	10/02/2025	07/04/2025			
Finalisation et optimisation des performances	10/02/2025	07/03/2025			
Finalisation du module STS	10/02/2025	07/03/2025			
Développement du module de génération musicale	10/02/2025	07/03/2025			
Sprint 5	11/03/2025	07/04/2025			
Finalisation du module de génération musicale	11/03/2025	31/03/2025			
Tests d'acceptation et documentation	01/04/2025	07/04/2025			





## Conclusion

Ce rapport a présenté les travaux réalisés durant le Sprint 3, mettant en lumière les avancées majeures sur les cas prioritaires, notamment la création d'une application fonctionnelle pour le Text-to-Speech, Speech-to-speech, la recherche approfondie sur les technologies adaptées et le développement de prototypes pour valider certaines approches. Ces accomplissements constituent une base technique robuste, essentielle pour les phases ultérieures du projet. Ils nous permettent d'aborder les prochains sprints avec confiance, en nous concentrant sur le développement des fonctionnalités restantes et l'optimisation des performances.