

Université Sorbonne Paris Nord
Sup Galilée
Spécialité Informatique



Conduite et gestion de projet

Spécialité : INFO2

Cahier de charge

Interface de Génération Sonore par IA

Étudiants :

Seyfeddine JOUINI
Yosra SASSI
Sirine TLILI
Hasna ELGARANI
Kaoutar BRAHIMI

Client :
Vittascience

27 novembre 2024

Table des matières

1	Analyse du contexte	2
1.1	Description du projet	2
1.2	Analyse de l'existant	2
2	Analyse des Besoins	2
2.1	Les Acteurs	2
2.2	Besoins Fonctionnels	2
2.3	Besoins Non Fonctionnels	3
3	Architecture Globale	3
4	Méthodologie de Travail	3
5	Choix Techniques	4
5.1	Technologies Utilisées	4
5.2	Environnement Logiciel	4
6	Organisation	4
6.1	Organisation du groupe	4
6.2	Planification	4

Introduction

Ce document présente le cahier des charges pour le projet de développement du module de génération sonore par IA. Notre objectif est de proposer une interface intuitive et interactive permettant aux élèves de découvrir et d'expérimenter avec les technologies de synthèse vocale et de génération musicale de manière pédagogique et immersive. Ce module vise à fournir un outil complet, fonctionnant intégralement en local sans nécessiter de connexion Internet, et offrant une qualité sonore réaliste adaptée à un usage éducatif.

1 Analyse du contexte

1.1 Description du projet

Le projet consiste à développer un module de génération sonore, intégré à la plateforme Vittascience, destiné aux élèves. Ce module permettra une découverte pratique des concepts fondamentaux de l'intelligence artificielle à travers deux fonctionnalités principales : la synthèse vocale (Text-to-Speech) et la génération musicale. L'interface proposera une expérience interactive et engageante, permettant aux utilisateurs d'explorer la manipulation de paramètres sonores pour créer des voix et des musiques.

1.2 Analyse de l'existant

Fondée en 2018, la startup EdTech Vittascience propose des outils interactifs pour l'enseignement de la programmation par blocs visuels et la transition vers des langages tels que Python. Dans le domaine de l'intelligence artificielle, elle a déjà développé des interfaces pour la classification d'images, de sons, de postures, ainsi que pour la génération d'images et de textes. Le module de génération sonore répond à la continuité de cette démarche en introduisant un nouveau champ d'application pratique pour l'IA dans un cadre éducatif.

2 Analyse des Besoins

2.1 Les Acteurs

- **Élèves** : utilisateurs finaux de l'interface, qui découvrent les concepts d'IA à travers la manipulation de voix et de musiques générées par IA.
- **Enseignants** : responsables de la création de classes et de la gestion des élèves sur la plateforme, ils utilisent cet outil pour sensibiliser les élèves aux technologies de l'IA dans un cadre pédagogique.

2.2 Besoins Fonctionnels

- **Synthèse Vocale (Text-to-Speech)** : Permettre aux utilisateurs de saisir un texte et de générer une voix synthétique réaliste avec des réglages pour le timbre, la hauteur et la vitesse de la voix.

- **Génération Musicale** : Générer automatiquement de la musique dans un style sélectionné, en deux étapes : création des paroles, puis composition de la musique avec une voix synthétique.
- **Interface Intuitive** : Utiliser une interface simple et visuelle, avec des blocs glisser-déposer pour ajuster des paramètres comme la fréquence, le volume et la tonalité.
- **Exemples et Tutoriels** : Fournir des exemples prédéfinis et des tutoriels pour guider les utilisateurs dans la découverte et l'utilisation des fonctionnalités.

2.3 Besoins Non Fonctionnels

- **Exécution Locale** : Fonctionnement entièrement hors ligne, sans besoin de connexion Internet.
- **Compatibilité** : Compatible avec les principaux navigateurs web pour être accessible à tous les utilisateurs.
- **Simplicité de Développement et de Maintenance** : Utiliser des technologies simples (JavaScript, HTML/CSS) pour faciliter le développement et la maintenance.

3 Architecture Globale

L'architecture repose sur une exécution locale côté client pour garantir la fluidité et minimiser la dépendance aux serveurs externes :

- **Frontend** : Développé en JavaScript natif pour les interfaces utilisateur, permettant de saisir du texte, de choisir des options de voix, et de lancer la génération musicale.
- **Moteur de Traitement Sonore** : Utilise des bibliothèques de génération sonore pour la synthèse vocale et des technologies pour manipuler les sons et générer de la musique en temps réel.
- **Backend** : Utilisé uniquement si nécessaire pour la gestion de modèles volumineux ou pour le stockage des projets des utilisateurs. Des solutions de backend pourront être choisies en fonction des besoins techniques.

s

4 Méthodologie de Travail

Le projet adopte une méthodologie Agile (SCRUM) pour assurer une grande flexibilité et la capacité à s'adapter aux changements. Le travail est divisé en sprints, chacun se concentrant sur un ensemble de fonctionnalités prioritaires. Des livraisons fréquentes et des réunions régulières avec le client garantissent que les objectifs du projet sont bien compris et que les ajustements nécessaires sont effectués à chaque étape. Cette approche assure une transparence totale et une collaboration étroite entre les équipes techniques et le client.

5 Choix Techniques

5.1 Technologies Utilisées

- **JavaScript natif** : pour le développement de l'interface utilisateur.
- **Python** : pour gérer les requêtes du front-end et les traitements TTS et musicaux.
- **Bibliothèque TTS** : pour transformer le texte en voix synthétique réaliste, avec contrôle du timbre, du ton et de la vitesse.
- **Modèle de génération de texte** : pour générer des paroles en fonction du thème.
- **Logiciel de composition musicale** : pour composer et arranger la musique avec les paroles.

5.2 Environnement Logiciel

- **Éditeur de Code** : Visual Studio Code.
- **Outils de Versioning** : Git et GitHub pour la gestion du code.
- **Navigateurs Compatibles** : Chrome, Firefox, Edge.

6 Organisation

6.1 Organisation du groupe

L'équipe projet est organisée de manière à couvrir toutes les compétences nécessaires pour le succès du projet.

Étudiant	Rôle	Email
Jouini Seyfeddine	Chef de projet	jouini.seyf.js@gmail.com
Tlili Sirine	Développeur Front-End	cyrinetlili07@gmail.com
Sassi Yosra	Responsable Back-End	sassiyosra66@gmail.com
Elgarani Hasna	Architecte Logiciel	elgaranihasna@gmail.com
Brahimi Kaoutar	Spécialiste IA	kaoutarbrahimi28@gmail.com

TABLE 1 – Attribution des rôles au sein de l'équipe

6.2 Planification

Le projet suit la méthodologie Scrum sur une période de 165 jours, répartis en **16 sprints de 10 jours chacun**. Chaque sprint vise à livrer des fonctionnalités incrémentales, avec un suivi régulier des progrès. Le **Product Owner** (Léo Briand) définit les priorités et exigences du projet, tandis que le **Scrum Master** (Jouini Seyfeddine) facilite le processus et résout les obstacles. Les **Développeurs** (Tlili Sirine, Sassi Yosra, Elgarani Hasna, Brahimi Kaoutar) se concentrent sur le développement des fonctionnalités, notamment le front-end, le back-end et l'intégration de l'IA. Le tableau ci-dessous présente la planification des sprints et les tâches principales associées à chaque phase du projet.

Sprint	Tâches Principales
Sprint 1 : Initialisation du projet et analyse des besoins	Analyse des besoins, établissement du backlog, mise en place de l'environnement de développement, prototype de l'interface utilisateur, définition des user stories, début de la conception de l'architecture.
Sprint 2 : Développement et intégration du module TTS	Développement de l'API TTS, intégration du logiciel de TTS, tests unitaires du module TTS, développement front-end/back-end pour TTS, création de l'interface utilisateur pour TTS, début des tests d'intégration.
Sprint 3 : Développement du module de génération musicale	Développement du module de génération musicale, composition des morceaux avec les paroles générées, tests unitaires du module musical, intégration du logiciel pour la génération de paroles.
Sprint 4 : Finalisation et optimisation des performances	Finalisation des fonctionnalités TTS et génération musicale, optimisation des performances (Flask, front-end), implémentation du cache audio, ajustements de l'interface utilisateur, tests d'acceptation, revue de code.
Sprint 5 : Tests d'acceptation et documentation	Tests d'acceptation utilisateur, collecte des retours, corrections en fonction des retours, préparation de la documentation technique et utilisateur, guides d'utilisation et de déploiement.

TABLE 2 – Planification des Sprints pour le projet de génération sonore (répartition égale des tâches)

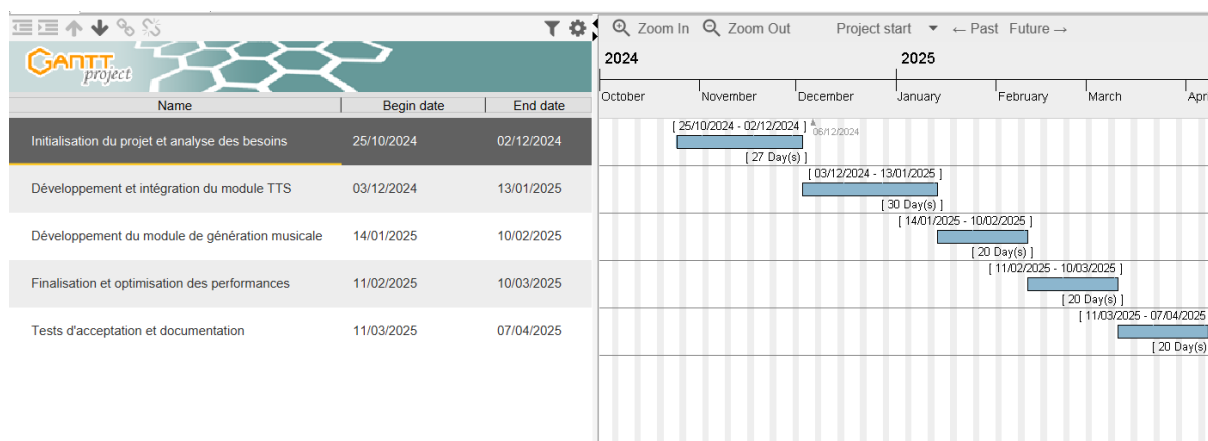


FIGURE 1 – Diagramme de Gantt

Conclusion

Le développement du module de génération sonore par IA constitue une avancée majeure dans la création d'outils pédagogiques interactifs pour l'apprentissage de l'IA. En proposant des fonctionnalités telles que la génération de texte en parole (TTS) et la composition musicale automatisée, ce projet offre une expérience ludique et pratique qui permet aux élèves d'explorer les principes de l'IA tout en développant des compétences techniques. En adoptant une méthodologie Agile et en centrant le processus de déve-

loppement sur les besoins des utilisateurs, nous visons à fournir un outil performant et accessible pour enrichir l'enseignement des technologies de l'IA. Grâce à ce projet, nous espérons non seulement répondre aux attentes des enseignants et des élèves, mais aussi inspirer l'usage créatif de l'IA dans le domaine éducatif.