

# Speakify – Cahier des Charges

## 1. Introduction

### 1.1 Présentation du projet

Speakify est une plateforme interactive d'apprentissage des langues qui permet aux utilisateurs d'interagir avec du contenu audio multilingue via des playlists structurées. La plateforme offre la possibilité de créer et de gérer des expériences audio personnalisées basées sur la traduction, rendant l'acquisition de langue plus immersive et plus efficace. Speakify est conçu pour être accessible sur mobile, ordinateur et écran TV.

### 1.2 Objectifs & Problématique

Speakify répond à un problème courant chez les apprenants en langues : le manque de temps, de pratique structurée et d'exposition contextuelle. Alors que de nombreuses applications proposent des traductions ou des jeux de vocabulaire, peu offrent des expériences audio en situation réelle, adaptées au rythme et aux objectifs de chaque utilisateur.

La plateforme est pensée pour soutenir l'apprentissage en mobilité — pour les personnes en déplacement, les professionnels occupés, ou encore les immigrants — en s'appuyant sur des boucles audio immersives et des playlists intelligentes contextuelles.

L'objectif est de rendre l'apprentissage des langues efficace, passif et naturel, avec des playlists adaptatives selon la répétition, la difficulté, ou les situations spécifiques (ex : aéroport, entretien d'embauche, rendez-vous médical).

#### Ce projet relève plusieurs contraintes :

- **Contrainte de temps** : Les utilisateurs n'ont souvent que 5 à 15 minutes par session.
- **Pertinence contextuelle** : Les cartes mémoire classiques manquent d'ancrage dans la réalité.
- **Accessibilité** : Le système doit fonctionner en mode mobile, sur ordinateur, et hors ligne.
- **Immersion** : La lecture audio doit simuler des conversations réelles.

Parfait, je continue.

---

### 1.3 Public cible

Speakify est conçu pour une large gamme d'utilisateurs, chacun ayant des besoins spécifiques en apprentissage linguistique :

- **Professionnels pressés** : personnes disposant de peu de temps pour étudier, qui bénéficient de l'apprentissage passif pendant les trajets quotidiens, les séances de sport ou les pauses au travail.
- **Immigrés et nouveaux arrivants** : individus arrivant dans un nouveau pays et ayant besoin d'acquérir rapidement un vocabulaire pratique et une aisance conversationnelle pour s'adapter socialement et professionnellement.
- **Enseignants et formateurs linguistiques** : professeurs à la recherche d'outils audio structurés pour concevoir des exercices, évaluer la prononciation ou proposer des activités d'écoute variées en classe.
- **Étudiants préparant des examens** : apprenants ayant besoin d'exercices d'écoute répétitifs pour renforcer la grammaire, le vocabulaire ou la compréhension orale.
- **Travailleurs multilingues et voyageurs** : professionnels ou voyageurs fréquents nécessitant un vocabulaire spécialisé dans des secteurs comme la santé, la technologie, l'hôtellerie ou l'aéronautique.
- **Personnes neurodivergentes ou apprenants auditifs** : individus apprenant mieux grâce à la répétition audio, l'immersion contextuelle ou la pratique espacée, incluant ceux atteints de dyslexie ou de TDAH.

Cette large accessibilité garantit que Speakify soutient un apprentissage inclusif, flexible et orienté vers les objectifs.

---

## 1.4 Sécurité & Accessibilité

Speakify intègre des considérations essentielles de sécurité et d'accessibilité dès les premières étapes du développement.

### Mesures de sécurité :

- Toutes les interactions utilisateur sont traitées via une API backend sécurisée pour éviter toute exposition de données sensibles.
- Des jetons (tokens) sont utilisés pour gérer l'authentification de session (aucun mot de passe stocké côté client).
- Prise en charge planifiée du chiffrement HTTPS pour le déploiement.

- L'accès aux données est compartimenté et sécurisé contre les injections ou utilisations abusives.
- Les playlists personnelles et le suivi des progrès seront protégés par un accès basé sur les autorisations (pour les utilisateurs connectés).

#### **Engagements en matière d'accessibilité :**

- Speakify suit les lignes directrices WCAG 2.1 en matière de conception inclusive.
- Les contenus audio sont accompagnés d'équivalents textuels pour les utilisateurs malentendants.
- L'interface propose un mode contraste élevé et des tailles de police adaptables pour le confort visuel.
- Les zones tactiles, la navigation clavier et les mises en page simplifiées sont optimisées pour les personnes ayant des difficultés motrices ou cognitives.
- Compatibilité prévue avec les lecteurs d'écran et les balises ARIA pour tous les éléments interactifs.

En intégrant ces normes dans la conception de base, Speakify garantit une expérience sûre et inclusive pour chaque utilisateur.

#### **1.5 Structure des données & Modèle Conceptuel de Données (MCD)**

Pour garantir une base de données évolutive et bien structurée, Speakify repose sur un modèle de données relationnel. Le Modèle Conceptuel de Données (MCD) définit les entités clés et les relations utilisées dans toute l'application :

- **Utilisateur** : identifié par un e-mail ou un jeton de session, lié aux playlists personnelles et aux préférences sauvegardées.
- **Playlist** : une session d'apprentissage personnalisée composée d'un ou plusieurs blocs de traduction.
- **Bloc de traduction (TB)** : une phrase dans une langue source, liée à une ou plusieurs traductions.
- **Traduction** : une phrase dans une autre langue, éventuellement liée à un fichier audio.
- **Schéma** : une configuration de lecture appliquée à une ou plusieurs playlists.

- **Langue** : répertorie les langues prises en charge et leurs codes.
- **Session** : suit l'activité d'un utilisateur anonyme ou connecté.

Chaque entité est clairement normalisée, avec des clés étrangères assurant l'intégrité référentielle. La conception prend en charge l'expansion multilingue, la synchronisation hors ligne et un contrôle fin des schémas de lecture.

Un diagramme MCD visuel sera présenté lors de la soutenance, affichant toutes les cardinalités ( $1 \leftrightarrow N$ ,  $N \leftrightarrow N$ , etc.) et les relations entre les tables regroupées par zones (utilisateur, contenu, métadonnées).

---

## 1.6 Budget & Calendrier

### Budget estimé

Speakify est actuellement un projet étudiant développé sans financement externe. Toutefois, une mise en œuvre professionnelle impliquerait les coûts approximatifs suivants :

Catégorie	Estimation (€)	Remarques
Design UI/UX	800 – 1 200	Conception d'interfaces responsives
Développement frontend	1 500 – 2 000	HTML/CSS/JS, vues responsives
Développement backend	1 500 – 2 000	API, intégration base de données
Services audio & TTS	500 – 1 000	Génération audio initiale
Hébergement & infra	300 – 600	12 mois d'hébergement, domaine, backups

Audit accessibilité






300 – 500

Audit tiers optionnel

### Total estimé : 4 900 – 7 300 €

Ces estimations sont données à titre indicatif, basées sur des tarifs freelances standards.

### Calendrier de développement

Phase	Dates	Livrables
 Planification & recherche	25 – 29 mars	Analyse besoins, fonctionnalités, schéma BDD
 API & Backend	30 mars – 5 avril	API fonctionnelle, système de sessions sécurisé
 Frontend (MVP)	6 – 12 avril	UI pour lecture, dashboard, éditeur de playlist
 Tests & QA	13 – 19 avril	Tests manuels, retours utilisateurs, corrections
 Finalisation & soutenance	20 – 23 avril	Démo finale, documentation, présentation

Les dates respectent le calendrier officiel du projet et garantissent que tous les livrables sont prêts pour la soutenance du 23 avril.

---

### 1.7 Outils & Méthodologie de travail

Pour garantir une collaboration efficace, le suivi des tâches et le contrôle de version, les outils suivants ont été sélectionnés et intégrés au flux de travail du projet :

- **Git & GitHub** : utilisés pour le versioning du code frontend et backend. Commits descriptifs, branches pour dissocier dev et prod.
- **Trello** : tableau Kanban pour suivre les tâches de la planification au déploiement. Colonnes : À faire / En cours / En revue / Terminé.
- **Discord** : communication quotidienne, messages de statut, réunions vocales. Salon dédié pour captures, liens, décisions.
- **Notion** : hub de documentation pour les comptes-rendus, brainstorming, recherches, personas utilisateurs, specs.
- **Figma** : création et validation des wireframes et composants UI. Prototypes partagés pour alignement rapide.
- **Visual Studio Code** : IDE principal avec extensions PHP, HTML/CSS, JS, Git, linters intégrés.

Tous les outils ont été choisis pour leur accessibilité, simplicité et adéquation avec les workflows agiles modernes. Cette stack permet un travail individuel structuré et scalable, mais aussi facilement extensible à une équipe.

## 2. Fonctionnalités & Résolution de problèmes

### 2.1 Fonctionnalités de l'application

- **Playlists audio multilingues** : Les utilisateurs peuvent créer et personnaliser des playlists composées de boucles de lecture avec des fichiers audio.
- **Boucles de lecture (PLs)** : Unité structurée contenant une phrase dans la langue source et ses traductions dans plusieurs langues, chacune avec un fichier audio.
- **Schéma de lecture personnalisable** : L'utilisateur définit la séquence, le nombre de répétitions et les temps de pause.
- **Compatibilité multiplateforme** : Fonctionne sur mobile, bureau et télévision pour un apprentissage flexible.
- **Apprentissage interactif** : Options de lecture en boucle, vitesse ajustable, entraînement à la prononciation.
- **Mode intensif** : Sessions répétitives et ciblées avec accélération et difficulté adaptative.

- **Mode hors ligne (prévu)** : Stockage local des traductions et audios.
- **Gestion des playlists utilisateurs** : Création, modification, organisation selon les besoins.
- **Classement par difficulté et contexte** : Par exemple, vocabulaire professionnel spécifique à un secteur.
- **Apprentissage en déplacement** : Conçu pour les trajets, séances sportives, pauses boulot.
- **Soutien aux immigrés** : Facilite l'apprentissage du langage courant dans un nouvel environnement.
- **Extension future avec l'IA** : Intégration de services de traduction IA.
- **Contrôle de pertinence des traductions** : L'utilisateur peut augmenter ou réduire la fréquence d'affichage d'une traduction.
- **Schéma basé sur la pertinence** : Appliquer différents schémas selon le niveau de maîtrise.

#### Listes intelligentes (Smart Lists) :

- **Contextes ciblés par mot-clé** : Phrases contenant un mot spécifique, avec usage réel.
- **Listes de conjugaison** : Phrases illustrant la conjugaison de verbes variés.
- **Listes contextuelles** : Situations précises comme l'aéroport, l'hôpital, etc.
- **Phrases aléatoires** : Pour exposer à des structures variées.
- **Dialogues** : Conversations audio pour favoriser la compréhension orale.
- **Histoires** : Playlists narratives pour un apprentissage naturel par immersion.
- **Mot du jour** : Un mot ou verbe quotidien accompagné de phrases contextualisées.

---

### 3. Navigation & Maquettes

#### 3.1 Schéma de navigation utilisateur

## ► Lancement de l'application



### Tableau de bord

- Démarrer une nouvelle session → 🎧 Lecteur
- Playlist récente → 🎧 Lecteur
- Mot du jour → 🧠 Listes intelligentes > Mot du jour
- Playlists → 📖 Bibliothèque de playlists



### Bibliothèque de playlists

- Voir la playlist → 🎧 Lecteur
- + Créer une playlist → 🖋 Éditeur de playlist
  - Assigner un schéma → 🔧 Sélecteur de schéma ou ➕ Créer un nouveau schéma
- Modifier le schéma → 🔧 Éditeur de schéma



### Éditeur de schéma

- Définir l'ordre de lecture (ex. : EN → FR → Pause → Répéter)
- Définir le nombre de répétitions par segment
- Ajuster les délais / la vitesse
- Sauvegarder le schéma



### Listes intelligentes

- Mot du jour → Phrases contextuelles
- Conjugaisons → Phrases avec variations de verbes
- Contextes thématiques → ex. : Aéroport, Restaurant
- Dialogues / Histoires → Conversations audio contextualisées



### Lecteur

- Contrôles : Lecture, Pause, Boucle, Suivant, Précédent, Vitesse



### Paramètres

- Vitesse audio
- Préférences linguistiques
- Options du mode hors ligne

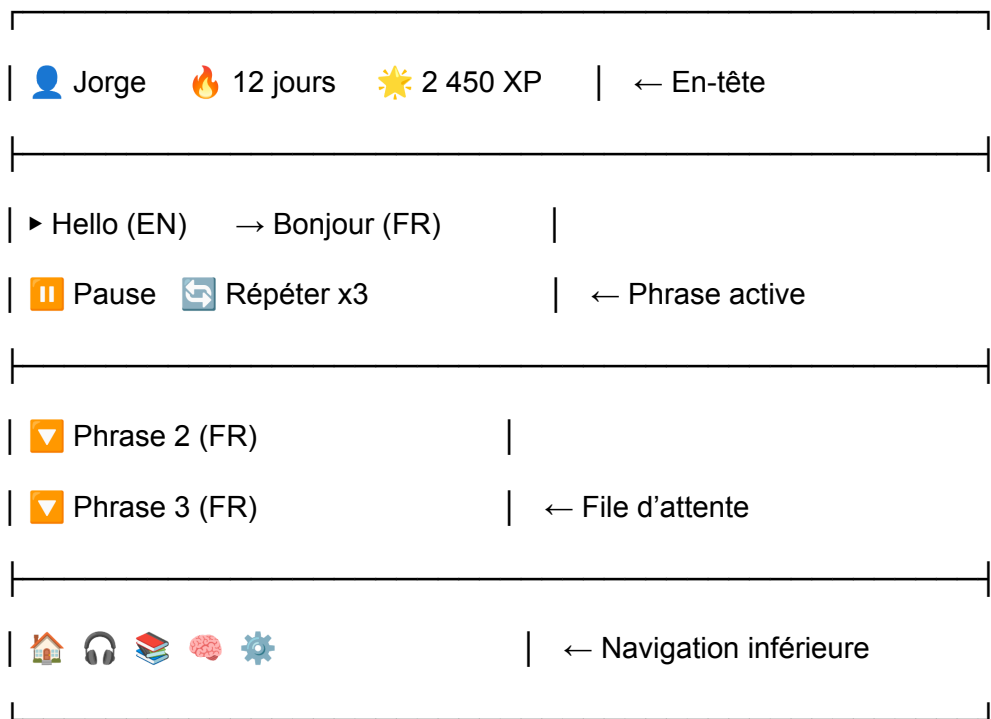


### Connexion (optionnelle)




- Synchronisation des données, sauvegarde du profil, fonctionnalités cloud



3.2 Maquette (basse fidélité) – Vue lecteur 🎧



3.3 Écrans de l'interface (Feuille de route MVP)

Vue	Statut	Description
Vue lecteur	 Terminé	Lecture audio avec séquence de traduction
Vue éditeur de schéma	 Prochaine étape	Création et personnalisation de la logique de lecture
Bibliothèque de playlists	 À faire	Recherche, navigation, gestion de playlists

Éditeur de playlist	<input type="checkbox"/> À faire	Création / assignation de schémas
Vue listes intelligentes	<input type="checkbox"/> À faire	Listes générées automatiquement (verbes, contextes...)
Vue paramètres	<input type="checkbox"/> À faire	Vitesse, langue, options hors ligne

---

### Fonctionnalités supplémentaires (planifiées)

Fonctionnalité	Statut
Vue mot du jour	<input type="checkbox"/>
Vue connexion & profil	<input type="checkbox"/>
Vue statistiques & succès	<input type="checkbox"/>
Vue mode hors ligne	<input type="checkbox"/>

## 4. Identité Graphique (UI/UX Design)

Speakify vise à offrir une expérience visuelle claire, immersive et cohérente sur tous les appareils — mobiles, ordinateurs et téléviseurs. Le système de design privilégie la lisibilité, la concentration et la convivialité pour favoriser un apprentissage linguistique engageant sur le long terme.

---

### 4.1 Palette de couleurs & Identité visuelle

Usage	Nom de la couleur	Code Hex	Remarques
Couleur principale	Bleu profond	#0057B7	Confiance, calme, concentration
Accent	Vert électrique	#00E676	Actions importantes (ex. : Lecture)
Arrière-plan	Gris clair	#F5F7FA	Réduction de la fatigue visuelle
Surface	Blanc doux	#FFFFFF	Fonds de cartes et de conteneurs
Texte principal	Noir charbon	#2E2E2E	Contraste élevé, lisibilité optimale
Alerte / Erreur	Rouge corail	#FF5252	Pour erreurs, suppressions, alertes critiques

L'interface reste globalement sobre et apaisante, en n'utilisant les couleurs vives que pour les actions essentielles ou les retours utilisateurs.

---

## 4.2 Typographie

Speakify utilise des polices modernes, lisibles et adaptées aux contenus multilingues et à l'accessibilité.

Usage	Police principale	Polices de secours
Titres	Segoe UI Bold	system-ui, sans-serif

Texte principal   Segoe UI Regular   Helvetica, Arial

Éléments UI   Antipasto Pro (Light)   Segoe UI

Les polices ont été sélectionnées pour leur disponibilité multiplateforme et leur harmonie visuelle. Tous les textes sont responsives et optimisés pour le mobile.




---

#### 4.3 Composants UI & Mise en page

- **Boutons** : coins arrondis (12px+), contraste fort, changement de couleur au survol
  - **Cartes / Conteneurs** : ombres douces, espacements réguliers, fond neutre
  - **Navigation** : onglets en bas sur mobile ; barre latérale ou en haut sur desktop
  - **Icônes**  : bibliothèques modernes (Lucide, Tabler, Material Symbols)
  - **Animations** : transitions légères (ex. : expansion de bloc actif)
  - **Mode sombre** : prévu comme option pour usage nocturne
- 

#### 4.4 Cohérence multi-appareils

Speakify est développé comme une **application web progressive (PWA)** totalement responsive, avec une interface cohérente et un comportement stable quel que soit l'appareil.

-  **Téléphones** : boutons adaptés au pouce, navigation inférieure
-  **Ordinateurs portables / fixes** : zones de lecture élargies, navigation latérale ou supérieure
-  **TVs et écrans intelligents** : textes larges, contrastes élevés, commandes simplifiées

Grâce à une grille fluide, des composants réutilisables et un style partagé, l'interface assure une expérience unifiée et professionnelle.

## 5. Éco-Responsabilité

Speakify s'engage à adopter des pratiques numériques durables. En tant que plateforme moderne d'apprentissage des langues basée sur le web, elle sera conçue pour optimiser la consommation d'énergie, réduire les transferts de données et utiliser efficacement les ressources sur tous les appareils.

---

### 5.1 Efficacité des serveurs

Pour réduire l'empreinte carbone de la plateforme, Speakify privilégie :

- **Des services backend légers** : APIs optimisées ne retournant que les données essentielles
  - **Des requêtes SQL performantes** : utilisation de caches, pagination, et indexation efficace
  - **Des hébergeurs verts** : préférence pour les fournisseurs utilisant des énergies renouvelables (ex. : Infomaniak, Scaleway Green)
  - **Une infrastructure auto-scalable** : allocation dynamique des ressources selon la charge réelle, évitant le surdimensionnement
- 

### 5.2 Optimisation pour les faibles débits

L'interface est conçue pour rester fluide même en conditions réseau dégradées (zones rurales, 3G, connexions internationales).

- **Chargement paresseux (lazy loading)** des images, fichiers audio et contenus lourds
  - **Streaming audio adaptatif** : qualité ajustée selon la bande passante
  - **Formats légers et compressés** : WebP pour les images, Brotli ou Gzip pour les ressources
  - **Réduction des scripts tiers** : uniquement les outils indispensables à l'expérience utilisateur
-

### 5.3 Mode hors-ligne

Le mode hors-ligne sert à la fois l'accessibilité et la sobriété numérique, notamment dans les régions avec couverture limitée ou coûts élevés de données.

- **Cache local** : les blocs de traduction et fichiers audio sont stockés par playlist
  - **Téléchargement à la demande** : l'utilisateur décide ce qu'il souhaite rendre disponible hors-ligne
  - **Préchargement contrôlé** : possibilité de choisir des packs thématiques à synchroniser à l'avance
  - **Logique sensible à la batterie** (prévu) : limitation des tâches lourdes lorsque la batterie est faible
- 

### 5.4 Améliorations futures

- **Mode sombre éco-énergétique** : réduction de la consommation sur écrans OLED
- **Analyses d'usage** : détection des surcharges serveur pour optimiser la performance
- **IA légère côté serveur** : intégration d'intelligences artificielles économes, exécutées en périphérie ou sans serveur (serverless)

En intégrant ces pratiques dès la conception, Speakify allie haute qualité pédagogique et responsabilité environnementale.

Voici la traduction complète en français, tout en respectant le format et le style du texte original :

---

### 5.4 Améliorations futures

- **Mode sombre économe en énergie** : Réduction de la consommation d'énergie des écrans OLED.
- **Analytique basée sur l'usage** : Suivi de la charge du système pour identifier des opportunités d'optimisation.

- **IA légère et responsable** : Si des fonctionnalités IA sont intégrées, elles utiliseront des solutions serverless ou edge computing avec un impact énergétique minimal.

En appliquant ces stratégies, Speakify vise à offrir une expérience d'apprentissage de haute qualité tout en adoptant une empreinte numérique responsable et durable.

---

## 6. SEO & Présence en ligne

Pour que Speakify atteigne un public mondial d'apprenants, d'enseignants et de professionnels multilingues, une stratégie SEO réfléchie sera mise en œuvre. L'objectif est de rendre la plateforme très visible, facilement trouvable et accessible dans plusieurs langues et régions.

### 6.1 Stratégie SEO

L'approche SEO de Speakify se concentrera à la fois sur le SEO technique (structure du site, performance) et sur le SEO de contenu (pertinence, clarté).

- **Optimisation de l'application PWA** : S'assurer que l'application est bien structurée et indexable, avec rendu côté serveur ou dynamique si nécessaire.
- **Structure d'URL propre** : Les URLs seront sémantiques et localisées (ex. : /fr/playlists, /en/word-of-the-day).
- **Balises meta & Open Graph** : Chaque page contiendra un titre, une description et des métadonnées sociales significatives.
- **Sitemaps & robots.txt** : Génération automatique de sitemaps XML et configuration correcte des fichiers robots.txt pour un indexage optimal.
- **Temps de chargement rapides** : Livraison d'actifs optimisée et conformité aux Core Web Vitals pour améliorer le classement dans les moteurs de recherche.

### 6.2 Optimisation par mots-clés

Des mots-clés pertinents seront recherchés et intégrés au contenu, en ciblant différents contextes d'apprentissage et intentions utilisateur.

#### Cibles de mots-clés principaux :

- application d'apprentissage des langues
- pratique immersive des langues
- apprentissage audio basé sur la traduction

- écoute passive de langues
- apprendre le français en déplacement
- créateur de playlists linguistiques
- entraîneur de vocabulaire intelligent

#### **Stratégies de mots-clés contextuels :**

- Mots-clés de longue traîne pour des domaines spécifiques (ex. : "apprendre des phrases en espagnol médical")
- Termes basés sur la localisation ou culturellement adaptés (ex. : "anglais quotidien pour les immigrants en France")
- Optimisation pour la recherche vocale (ex. : "comment dire bonjour en japonais ?")

### **6.3 Indexation multilingue**

Puisque Speakify vise des utilisateurs multilingues à l'échelle mondiale, le contenu doit être accessible et indexable dans plusieurs langues.

- **Balises hreflang** : Pour indiquer la langue du contenu et la région ciblée.
- **Métadonnées spécifiques à la langue** : Titres, descriptions et slugs localisés.
- **URLs localisées** : Chemins distincts par langue (ex. : /fr, /en, /es, etc.).
- **Correspondance internationale de mots-clés** : Identification des équivalents dans les principales langues cibles (EN, FR, ES, DE, etc.).

Toutes les pages multilingues seront optimisées indépendamment pour le SEO afin d'améliorer leur visibilité dans les résultats de recherche en langue maternelle.

### **6.4 Présence en ligne & Visibilité de la marque**

Au-delà de l'indexation, Speakify développera sa notoriété via :

- **Intégration des métadonnées sociales** (balises OG, Twitter Cards)
- **Balises Schema.org** pour les résultats enrichis (ex. : "cours de langue", "leçon audio")



- **Optimisation dans les App Stores** (si la PWA est proposée via un wrapper mobile)
- **Compatibilité Google Discover** : Articles structurés et contenu vedette tel que le “Mot du jour”
- **Blog ou centre de ressources à venir** pour générer du trafic organique grâce à du contenu éducatif

Le SEO ne sera pas qu’un outil de découverte pour Speakify — ce sera un moteur de croissance aligné avec l’accessibilité mondiale et la pertinence des contenus.

Voici la traduction complète en français des sections 7 et 8, avec formatage et structure respectés :

---

## 7. Équipe & Méthodologie de travail

Speakify sera développé par une équipe agile, légère et très concentrée, collaborant efficacement pour livrer un produit abouti dans un délai court. Cette organisation favorise la responsabilité, l’adaptabilité et la livraison continue.

---

### 7.1 Équipe de développement

Rôle	Responsabilités
<b>Chef de projet (Project Owner)</b>	Définit la vision du produit, priorise les tâches, valide les fonctionnalités
<b>Product Manager</b>	Gère le calendrier, les jalons et la communication entre les parties prenantes
<b>Développeur principal (Lead Dev)</b>	Supervise l’architecture, la qualité du code et l’implémentation cœur backend/frontend

<b>Développeur Frontend</b>	Construit les composants de l'interface utilisateur, assure le responsive et l'accessibilité
<b>Développeur Backend</b>	Développe l'API, la base de données et la logique des schémas (ex. : boucles de lecture, progression)
<b>Designer UX/UI</b>	Crée les maquettes, prototypes, le style visuel et garantit une UX cohérente sur tous les supports
<b>Testeur QA</b>	Effectue des tests manuels/automatisés, valide les séquences de lecture et la stabilité générale
<b>Rédacteur/Curateur de contenu</b>	Prépare et gère le contenu multilingue, les traductions et les listes intelligentes
<b>DevOps (optionnel)</b>	Gère la mise en production, la synchronisation cloud et la configuration d'hébergement (à temps partiel)

**Note :** Certains rôles peuvent être assurés par une même personne durant les premières phases du projet (ex. : développeur full stack).

---

## 7.2 Répartition des rôles par phase du projet

Phase	Dates	Tâches	Rôles responsables
<b>Phase 1</b>	25–29 mars 2025	Finalisation du schéma BD, création du MCD	Dev Backend, Dev Principal

<b>Phase 2</b>	30 mars – 5 avril	Construction et tests des API principales	Dev Backend, Dev Principal
<b>Phase 3</b>	6–12 avril	Développement UI (Lecture, Listes intelligentes, Schémas)	Dev Frontend, Designer UX/UI
<b>Phase 4</b>	13–19 avril	Tests, corrections de bugs, optimisation générale	Testeur QA, Dev Principal
<b>Phase 5</b>	20–23 avril	Déploiement, documentation	DevOps, Product Manager

---

## 7.3 Méthodologie de travail

Speakify adopte une approche Agile légère, basée sur :

- **Sprints d'1 semaine**
- **Stand-up quotidien** (asynchrone autorisé)
- **Planification & revue hebdomadaires**
- **Tableau Kanban** : *À faire* → *En cours* → *En révision* → *Terminé*

**Outils utilisés :**

- **Git + GitHub** : Contrôle de version et collaboration
- **Trello / Notion** : Suivi des tâches et sprints
- **Figma** : Conception UI/UX
- **Slack / Discord** : Communication d'équipe

- **VS Code** : Environnement de développement

**Philosophie** : Livrer vite → valider → améliorer.

---

## 8. Spécifications techniques intégrées (Exigence de certification)

Conformément aux attentes de certification, les spécifications techniques suivantes sont clairement intégrées à ce document.

---

### 8.1 Architecture du système

- **Frontend** : HTML, CSS, JS (ES6+), Bootstrap, AJAX
- **Backend** : PHP 8.x avec API REST
- **Base de données** : MySQL

**Structure fonctionnelle :**

- Le frontend interroge et affiche les playlists, schémas et blocs de traduction
  - Le backend fournit dynamiquement le contenu via les endpoints API
  - La base de données stocke tout le contenu structuré (langues, phrases, paires de traduction, sources)
- 

### 8.2 Aperçu du schéma de base de données

Tables relationnelles prévues :

- **languages** : Langues disponibles avec ID, code et nom
- **sentences** : Données de base des phrases avec liaison à la langue
- **translation\_pairs** : Lie deux phrases comme une paire de traduction

- **sources** : Décrit la provenance de la traduction (manuel, API, etc.)
- **translation\_pair\_sources** : Lie les paires de traduction à leurs sources respectives

---

### 8.3 Pile technologique

Couche	Technologie	Notes
Frontend	HTML/CSS/JS	Structure PWA, layout avec Bootstrap
Backend	PHP 8.x	API REST
Base de données	MySQL	Schéma normalisé, tables indexées
Outils	XAMPP	Dev local, Apache + MySQL
API prévue	OpenAI	Services de traduction & TTS (prévu pour une future version)

---

### 8.4 Conception de l'API (prévisionnelle)

Base URL : <http://localhost/speakify/backend/api.php>

Méthode	Endpoint	Description
---------	----------	-------------

GET	<code>?action=playlists</code>	Récupère les playlists
GET	<code>?action=schemas</code>	Récupère les schémas de lecture
GET	<code>?action=tb&amp;id=xx</code>	Récupère un bloc de traduction spécifique
POST	<i>À définir</i>	Sauvegarde de playlists ou de schémas

Voici la traduction complète en français des sections 8.5 à 12, tout en respectant le format et le style :

---

## 8.5 Notes sur la sécurité

- **MVP** : Pas de connexion utilisateur pour l'instant
  - **Futur** : Authentification (JWT ou sessions), contrôle d'accès
  - **Hygiène API prévue** : Protection contre les injections SQL / XSS
- 

## 8.6 Tests

Type	Description
<b>Manuel</b>	Lecture audio, comportement des boucles, interactivité UI

<b>Unitaire (prévu)</b>	Tests des API backend (PHPUnit)
<b>Frontend</b>	Logique de lecture (Jest/Cypress prévu)
<b>Tests multi-appareils</b>	Téléphones, ordinateurs, téléviseurs

---

## 9. Conclusion

Speakify est bien plus qu'un projet étudiant — c'est une solution concrète à un problème réel : apprendre une nouvelle langue avec peu de temps, dans un contexte quotidien, sur plusieurs appareils.

Il repose sur l'audio immersif comme mécanique centrale, tout en offrant à l'apprenant structure et liberté.

Tout au long de ce Cahier des Charges, nous avons défini non seulement les fonctionnalités, mais aussi les raisons qui les justifient :

- Pourquoi le modèle de données est conçu pour être évolutif
- Pourquoi l'accessibilité n'est pas optionnelle
- Pourquoi les profils utilisateurs, les schémas audio et les listes intelligentes existent
- Pourquoi Speakify convient à la fois aux utilisateurs mobiles et aux environnements éducatifs

Ce document n'est plus une simple liste de technologies ou de fonctions.

Il reflète une architecture réfléchie, évolutive, qui allie précision technique et empathie utilisateur.

### Prochaines étapes :

- Finition de l'interface utilisateur
- Présentation interactive du prototype

- Tests sur plusieurs types d'appareils
- Présentation visuelle du MCD (Modèle Conceptuel de Données) lors de la soutenance pour illustrer l'architecture

**Speakify est prêt** — à la fois comme outil d'apprentissage et comme histoire produit.

---

**Version du document** : 1.2.0

**Date** : Mars 2025

**Auteur** : Jorge

---

## **Speakify - Cahier des Spécifications Techniques**

**Nom du projet** : Speakify

**Version** : 1.0.0

**Date** : Mars 2025

**Auteur(s)** : Jorge

Speakify est une PWA multiplateforme conçue pour faciliter l'apprentissage immersif des langues via des playlists audio multilingues structurées.

Elle est compatible avec les usages mobile, desktop et TV, et s'appuie sur des Blocs de Traduction (TBs) et des schémas de lecture personnalisables pour offrir une expérience flexible.

---

## **10. Architecture du système**

### **10.1 Architecture générale**

- **Frontend (HTML/CSS/JS)** : Interface dynamique PWA
- **Backend (PHP)** : Fournit les APIs de récupération des données
- **Fichiers JSON** : Simulent les données d'API
- **Base de données (prévue)** : Stockage structuré des traductions et des schémas

### **10.2 Détail des composants**

**Frontend** :



- HTML, CSS, Bootstrap, JS
- Chargement AJAX des playlists et schémas
- Interfaces pour la lecture, l'édition de playlists, les listes intelligentes

#### Backend :

- Gestionnaire d'API PHP ([api.php](#))
- Gère la récupération des playlists, TBs, schémas

#### Base de données :

- Schéma prévu avec tables pour traductions, langues, utilisateurs, etc.
- Prise en charge des traductions manquantes et du contrôle de version

#### Services externes / APIs :

- **Prévu** : API OpenAI pour la traduction et la synthèse vocale (TTS)

---

## 11. Pile technologique

Couche	Technologie	Version	Notes
Frontend	HTML/CSS/JS	ES6+	PWA, interface responsive
Backend	PHP	8.x	Endpoints API simples
Base de données	MySQL (prévu)	À définir	Modèle de données structuré

<b>APIs</b>	OpenAI (prévu)	N/A	Services de traduction & synthèse vocale
<b>DevOps/CI</b>	XAMPP / Manuel	N/A	Environnement local de développement
<b>Autres</b>	Fichiers JSON	N/A	Données simulées pour la phase de dev

---

## 12. Modèle de données & Structures

### 12.1 Schéma de base de données (prévu)

Tables relationnelles prévues :

- `languages`
- `sentences`
- `translation_pairs`
- `sources`
- `translation_pair_sources`

### 12.2 Exemple de schéma JSON

```
{
  "playlist_id": "123",
  "name": "Français de base",
  "blocks": [
    {
      "tb_id": "456",
```

```
"text_source": "Hello",  
"text_translation": "Bonjour",  
"audio_source": "hello_en.mp3",  
"audio_translation": "bonjour_fr.mp3"  
}  
]  
}
```

Voici la **traduction complète en français** des sections 13 à 20, avec style et structure préservés pour intégration dans ton Cahier des Charges :

---

### 13. Points de terminaison de l'API

URL de base : <http://localhost/speakify/backend/api.php>

Méthode	Endpoint	Description	Auth	Paramètres
GET	<a href="#">?action=playlists</a>	Récupère toutes les playlists	Non	—
GET	<a href="#">?action=schemas</a>	Récupère tous les schémas	Non	—
GET	<a href="#">?action=tb&amp;id=xxx</a>	Récupère un bloc de traduction	Non	id

POST	À définir	Crée / met à jour une playlist	Oui	playlist, schema
------	-----------	--------------------------------	-----	---------------------

---

## 14. Logique métier & Flux de travail

### Logique des boucles de lecture (PL) :

- Suit un schéma défini par l'utilisateur : ordre, répétitions, pauses
  - Itère à travers les blocs de traduction d'une playlist
  - Chaque bloc contient une source + une ou plusieurs traductions avec audio
  - Les listes intelligentes génèrent automatiquement des playlists basées sur un contexte ou des critères
  - Une seule boucle est développée à la fois
  - Les boucles ouvertes affichent : barre de progression, répétitions, texte original et traduit
  - Les boucles fermées sont résumées sur une seule ligne
  - Le bouton global *lecture/pause* contrôle toute la session et est flottant en bas à droite de l'écran
  - Les données audio et de traduction sont chargées dynamiquement depuis `data/translations.json`
- 

## 15. Sécurité

- **Aucune authentication** dans le MVP actuel
- **Futur :**
  - Connexion utilisateur via **JWT** ou **cookies de session**

- Contrôle d'accès aux playlists personnelles
- Assainissement des entrées API pour prévenir **injections SQL/XSS**

---

## 16. Structure de l'interface (UI)

Page / Vue	Description	Fichiers associés
<b>Dashboard</b>	Point d'entrée avec raccourcis	<code>dashboard.html</code>
<b>Playback</b>	Interface principale de lecture	<code>playback.html</code>
<b>Bibliothèque</b>	Liste des playlists créées	<code>playlist-library.html</code>
<b>Éditeur de playlist</b>	Création / modification	<code>playlist-editor.html</code>
<b>Listes intelligentes</b>	Playlists générées par contexte	<code>smart-lists.html</code>
<b>Éditeur de schéma</b>	Définition des règles de lecture	<code>schema-editor.html</code>
<b>Paramètres</b>	Préférences utilisateur	<code>settings.html</code>
<b>Succès</b>	Suivi de progression	<code>achievements.html</code>

<b>Connexion / Profil</b>	Authentification (optionnelle)	<code>login-profile.html</code>
---------------------------	--------------------------------	---------------------------------

<b>Mode hors-ligne</b>	Gestion locale des fichiers	<code>offline-mode.html</code>
------------------------	-----------------------------	--------------------------------

Toutes les pages HTML doivent :

- Utiliser un `<head>` cohérent avec les balises meta + liens vers `style.css` et `script.js`
- Définir les sections : `.header`, `.content`, `.footer-nav`
- Inclure **3 icônes interactives** dans `.header`
- Avoir une navigation en bas **fixe** et **responsive**

---

## 17. Environnements & Déploiement

Environnement	URL	Remarques
Local	<code>http://localhost/speakify/</code>	Développement local via XAMPP
Pré-production	<i>À définir</i>	Environnement de test QA
Production	<i>À définir</i>	Déploiement final en ligne

---

## 18. Plan de tests

### Tests manuels pour :

- Fonctionnalité de lecture
- Progression dans les boucles
- Édition des playlists et association avec schémas

### Prévu :

- Tests unitaires des APIs backend (PHPUnit)
  - Validation de la logique de lecture frontend (Jest ou Cypress)
  - Tests multi-appareils (mobile, desktop, TV)
- 

## 19. Performances & Contraintes

- **Design responsive** pour une expérience mobile-first
  - Pied de page **fixe** et contrôles **flottants** pour l'ergonomie
  - Le contenu de lecture est chargé via **AJAX**, non intégré en dur
  - Implémentation du **lazy-loading** audio + effets de transition
  - À la fin d'une boucle, la file est décalée et la suivante chargée
  - **Une seule boucle ouverte à la fois** pour plus de clarté
  - Design léger et unifié pour une expérience utilisateur cohérente
- 

## 20. Annexe

### Exemples JSON :

- `playlists.json`

- `schemas.json`
- `translations.json`

#### Glossaire :

- **TB** = Bloc de traduction (*Translation Block*)
- **PL** = Boucle de lecture (*Playback Loop*)
- **Schéma** = Logique de séquençement de lecture

#### Services externes :

- **API OpenAI** (*prévu*) pour les services de traduction et de TTS

#### Organisation des fichiers :

- Tous les fichiers dans `speakify/model/`
- **Script principal** : `script.js` (chargement différé)
- **Feuille de style principale** : `style.css` avec transitions et design global



