

Cahier de charges

Projet M1 ANDROIDE

*Développement d'une interface d'évaluation
automatique de textes*

Auteurs

HIBAOUI Imane
NAJEM Mohamed
VU Hoang Nguyen

Encadrant

M. NEBEL Léo

Année universitaire

2024-2025

Table des matières

1	Contexte du Projet	3
2	Objectifs du Projet	3
3	Tâches à Réaliser	4
3.1	Développement de l'Interface Utilisateur	4
3.2	Collecte et Gestion des Données	4
3.3	Intégration de l'Algorithme d'Analyse	4
3.4	Pistes supplémentaires de développement	5
3.4.1	Module de Feedback Interactif	5
3.4.2	Dashboard pour les Enseignants	5
3.4.3	Gamification	5
4	Besoins Fonctionnels	6
5	Conclusion	6

1 Contexte du Projet

Le projet consiste à développer une application web d'évaluation automatique de textes, destinée aux élèves de seconde. Cette interface permet aux élèves de répondre à divers exercices de français, tels que des réponses développées, des textes à trous, des QCM, etc., en saisissant un texte dans un champ dédié. L'application collecte également des données relatives aux frappes clavier (telles que les touches utilisées, les timestamps et les positions du curseur) pendant la saisie du texte. Ces données sont ensuite envoyées à un algorithme d'analyse existant pour fournir des retours détaillés sur la qualité du texte produit et sur le processus de révision de l'élève. L'objectif est d'améliorer la rédaction des élèves tout en collectant des données précieuses pour comprendre et évaluer leur processus d'écriture.

2 Objectifs du Projet

Les objectifs principaux de ce projet sont les suivants :

- **Créer une interface pédagogique interactive** permettant aux élèves de répondre à des exercices de français de manière engageante.
- **Collecter et analyser les données de saisie** des élèves, y compris le texte saisi et les frappes clavier, afin de fournir des retours détaillés.
- **Fournir des feedbacks automatiques** sur les erreurs orthographiques et syntaxiques, ainsi que sur le vocabulaire utilisé et le processus de révision de l'élève.
- **Développer une interface modifiable** et évolutive pour permettre l'intégration de nouvelles évaluations et de fonctionnalités personnalisées pour les enseignants.

3 Tâches à Réaliser

Les tâches nécessaires pour accomplir ce projet sont détaillées ci-dessous, avec les méthodes de réalisation proposées pour chaque tâche.

3.1 Développement de l'Interface Utilisateur

- **Création d'un design responsive et ergonomique** : L'interface utilisateur sera développée en utilisant des technologies web modernes telles que HTML5, CSS3 et JavaScript. Le design sera conçu pour être accessible et intuitif sur tous types d'écrans (ordinateurs, tablettes, smartphones), garantissant une expérience fluide et agréable pour l'élève.
- **Intégration d'un module de personnalisation pour les enseignants** : Les enseignants pourront configurer les types d'exercices proposés (QCM, texte à trous, réponse développée, etc.), en personnalisant les consignes et les critères d'évaluation. Cela sera réalisé via un tableau de bord accessible à l'enseignant, utilisant un framework comme React pour une gestion dynamique de l'interface.

3.2 Collecte et Gestion des Données

- **Optimisation du stockage et de la synchronisation des données de frappe** : Nous utiliserons une base de données relationnelle comme PostgreSQL pour stocker les données de saisie des élèves, en particulier les frappes clavier. Les données seront organisées sous forme de trois listes synchronisées : les touches, les timestamps et les positions du curseur. Cette organisation garantit la bonne synchronisation des données, essentielle pour l'analyse du processus de révision.
- **Développement d'outils d'analyse préliminaire pour vérifier l'intégrité des données collectées** : Avant d'envoyer les données à l'algorithme d'analyse, des outils d'analyse seront développés en Python pour vérifier la validité des données collectées. Ces outils effectueront des contrôles sur la structure des données et sur leur cohérence temporelle.

3.3 Intégration de l'Algorithme d'Analyse

- **Mise en place d'un système d'API pour faciliter la transmission des données** : Nous créerons une API RESTful utilisant Django Rest Framework pour transmettre les données collectées vers l'algorithme d'analyse des textes. Cette API sera conçue pour garantir la sécurité et l'efficacité du transfert de données entre l'interface et l'algorithme.
- **Test et validation des retours générés par l'algorithme** : Une fois les données envoyées à l'algorithme, nous testerons les retours générés pour vérifier leur pertinence et leur précision. Cela comprendra des tests automatisés et manuels, en recueillant les retours des utilisateurs pour ajuster les feedbacks.

3.4 Pistes supplémentaires de développement

3.4.1 Module de Feedback Interactif

- **Proposition de stratégies de révision et d'amélioration personnalisées :** En fonction des erreurs et des corrections effectuées par l'élève, l'application proposera des stratégies de révision adaptées, comme des exercices supplémentaires similaires présents sur le site afin de cibler les points faibles de l'élève.

3.4.2 Dashboard pour les Enseignants

- **Visualisation en temps réel des performances des élèves :** Un tableau de bord permettra aux enseignants de suivre en temps réel les performances des élèves, avec des visualisations des erreurs récurrentes et des progrès réalisés.
- **Statistiques et graphiques d'évolution des compétences :** Des graphiques seront générés pour afficher l'évolution des compétences des élèves, basés sur l'analyse des erreurs et des révisions effectuées.

3.4.3 Gamification

- **Mise en place d'un système de récompenses et de niveaux :** Un système de points et de niveaux sera introduit pour encourager la progression des élèves. Chaque exercice terminé avec succès permettra à l'élève de gagner des points ou de débloquent des niveaux.
- **Intégration d'éléments ludiques :** Des badges et des défis seront utilisés pour stimuler l'engagement des élèves. Par exemple, un élève pourrait obtenir un badge pour avoir corrigé un certain nombre d'erreurs dans ses exercices.

4 Besoins Fonctionnels

Les besoins fonctionnels du projet sont les suivants :

- Un champ de saisie permettant aux élèves de rédiger leur réponse.
- La sauvegarde des textes saisis dans une base de données, réalisée grâce au module Django existant.
- L’enregistrement des frappes clavier, comprenant l’enregistrement des touches utilisées, des timestamps et des positions du curseur.
- Un système d’identification permettant de relier chaque élève à un exercice spécifique.
- La possibilité de proposer divers types d’exercices, tels que des textes à trous, des QCM, des réponses développées, etc.
- Une interface qui transmet en arrière-plan les données collectées à l’algorithme d’analyse pour générer des retours sur les erreurs et sur le processus d’écriture.
- Un design modulable et adaptable permettant d’ajouter de nouveaux types d’exercices ou de modifier l’interface selon les besoins.
- Un système d’authentification pour sécuriser l’accès à l’application.

5 Conclusion

Ce cahier des charges présente un cadre structuré pour le développement d’une interface d’évaluation automatique de textes. En collectant les données de saisie des élèves et en fournissant des feedbacks détaillés sur la qualité des textes et le processus de révision, ce projet vise à améliorer les compétences rédactionnelles des élèves. Les fonctionnalités avancées comme la gamification, l’intégration multimédia et la personnalisation des retours renforceront l’interactivité et l’engagement des élèves. La gestion soignée des données et l’évolution possible de l’application garantiront sa durabilité et son efficacité à long terme.