 Département de génie logiciel et des TI	PROJET <b>LOG/GTI 795</b>	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION <b>1.10</b>
	TITRE PFE 008	PAGE 1		PAGES 22


**PROPOSITION**  
**Projet de fin d'études**  
**Département de génie logiciel et des TI**

**PFE 008**  
**Reconnaissance visuelle de partitions musicales avec OMR**

**Auteurs**  
**Laflèche Chevrette**  
 CHEL70100001  
**Jérémy Da Costa**  
 DACJ03029800  
**Xavier Jeanson**  
 JEAX04039709  
**Philippe Langevin**  
 LANP13029806  
**Charlie Poncsak**  
 PONC06109907

**Professeurs superviseurs**  
**Camille Coti, Pascal Girard**


**Date**  
**22 mai 2024**

 Département de génie logiciel et des TI	PROJET <b>LOG/GTI 795</b>	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION <b>1.10</b>
	TITRE PFE 008	PAGE 2		PAGES 22

## Suivi des changements

\*A – Ajouté M – Modifié S – Supprimé


NUMÉRO DE VERSION	DATE aaaa/mm/jj	NUMÉRO DE FIGURE, TABLE OU SECTION	A* M S	BRÈVE DESCRIPTION DU CHANGEMENT	NUMÉRO DE DEMANDE CHANGEMENT	Auteur(s)
1.0	2024/05/13	1, 4	A	Ajout des informations préliminaires du plan de projet		La Flèche ...
1.1	2024/05/14	1, 4, 5.1	A,M	Ajout d'information préliminaire, modification de l'information ajouté en 1.0		Philippe La...
1.2	2024/05/17	1, 2, 3, 4	A	J'ai ajouté les infos que le client nous a donné lors de la rencontre.		La Flèche ...
1.3	2024/05/18	1, 4	A,M	Complétion du tableau des échéanciers		Jérémy Da ...
1.4	2024/05/19	4,6	A,M	Ajout de risques		Charlie Pon...
1.5	2024/05/20	1, 2, 7	A, M	Ajout des techniques et outils. Restructuration de 1 et 2.		Philippe La...
1.6	2024/05/22	4, 5	A, M	Ajout des livrable et planification, et modification des rôles		Xavier Jea...
1.7	2024/05/26	1, 2, 7	A, M	Ajout des retombées du projet et complétion de nos choix technologiques.		Jérémy Da ...
1.8	2024/05/27	3, 8, 9.1, Annexe B	A,M	Ajout du paragraphe sur les rencontres dans 3, création des objectifs de réussites, création des contraintes fonctionnelles, Création de l'annexe B		La Flèche ...
1.9	2024/05/27	1-9	M	Correction des fautes d'orthographe		Xavier Jea...

 Département de génie logiciel et des TI	PROJET LOG/GTI 795	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION 1.10
	TITRE PFE 008	PAGE 3		PAGES 22

1.10	2024/05/27	7.3, 9.2	A	Ajout de techno back-end, Ajout de contraintes N-F		Philippe La...
------	------------	-------------	---	---	--	----------------

## TABLE DES MATIÈRES

<b>1. PROBLÉMATIQUE ET CONTEXTE</b>	<b>4</b>
<b>2. OBJECTIFS DU PROJET</b>	<b>5</b>
2.1 Objectifs principaux	5
2.2 Objectifs secondaires (non obligatoires)	6
2.3 Retombées du projet	6
<b>3. MÉTHODOLOGIE</b>	<b>7</b>
3.1 Tableau des échéanciers	9
<b>4. COMPOSITION DE L'ÉQUIPE ET RÔLES</b>	<b>11</b>
<b>5. LIVRABLES ET PLANIFICATION</b>	<b>12</b>
5.1 Description des artefacts	12
5.2 Planification	12
<b>6. RISQUES</b>	<b>13</b>
<b>7. TECHNIQUES ET OUTILS</b>	<b>14</b>
7.1 Outils et techniques de gestion de projet	14
7.2 Outils de communication	15
7.3 Technologies et outils de développement.	15
<b>8. OBJECTIFS DE RÉUSSITE</b>	<b>16</b>
<b>9. CONTRAINTES</b>	<b>16</b>
9.1 Contraintes fonctionnelles	16
9.2 Contraintes non-fonctionnelles	17
<b>10. RÉFÉRENCES</b>	<b>19</b>
<b>ANNEXE A : PLAN DE TRAVAIL</b>	<b>20</b>
<b>ANNEXE B : LISTE DES ÉLÉMENTS D'UNE PARTITION</b>	<b>22</b>

 Département de génie logiciel et des TI	PROJET LOG/GTI 795	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION 1.10
	TITRE PFE 008	PAGE 4		PAGES 22


## 1. PROBLÉMATIQUE ET CONTEXTE

Le club étudiant MusiquÉTS travaille actuellement sur le projet MAESTRO, qui consiste à développer des instruments capables de jouer des partitions musicales de manière autonome. L'un des défis majeurs du projet est de concevoir et d'implémenter un système de reconnaissance optique de musique (OMR) capable de générer automatiquement les fichiers MIDI nécessaires pour MAESTRO.

Actuellement, MAESTRO ne dispose d'aucune solution automatisée pour la reconnaissance des partitions. Les tests se limitent à l'utilisation de fichiers MIDI pour essayer de faire jouer les instruments automatiquement, d'où la nécessité de notre solution.

Pour surmonter ces obstacles, deux approches s'offrent à nous: soit la réutilisation et l'amélioration d'un projet open-source existant, tels que cadencV, oemer, OMReader, Mozart, etc., soit développer une solution entièrement nouvelle. Bien que partir de zéro puisse offrir plus de flexibilité, cela risque d'être plus complexe et chronophage. Nous avons donc décidé de nous concentrer initialement sur un type de partition spécifique, en l'occurrence les partitions de piano à deux portées (une en clé de sol et une en clé de fa), afin de maximiser l'efficacité de notre développement et d'assurer une reconnaissance précise et fiable des partitions.

En ce qui concerne les technologies actuelles de reconnaissance optique de musique (OMR) présentent des limitations importantes pour répondre pleinement aux besoins spécifiques du projet MAESTRO. Plusieurs solutions open-source existent, mais aucune ne propose une solution complète et parfaitement adaptée aux exigences de notre projet.

 Département de génie logiciel et des TI	PROJET LOG/GTI 795	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION 1.10
	TITRE PFE 008	PAGE 5		PAGES 22

## 2. OBJECTIFS DU PROJET

Dans ce projet, nous devons concevoir et implémenter une application qui permettra de créer un fichier MIDI à partir d'une partition musicale numérique ou manuscrite. Le fichier MIDI doit respecter au maximum la partition afin d'être le plus fidèle possible à la musique d'origine.

### 2.1 Objectifs principaux

Conception et implémentation d'une application dorsale :

- *Lecture de partition musicale* : Développer une application dorsale capable de lire des partitions musicales sous format PDF, en utilisant des techniques de reconnaissance optique de musique (OMR).
- *Génération de fichier MIDI* : Implémenter un système qui génère un (ou plusieurs) fichier MIDI à partir de la partition lue, garantissant la fidélité à la partition originale en termes de notes, rythmes et dynamiques.
- *Interface API* : Mettre en place une interface (API) permettant d'accéder à l'application dorsale pour envoyer des partitions et recevoir les fichiers MIDI générés. Cette interface API permettra de découpler l'interface utilisateur et l'algorithme de reconnaissance optique de musique afin d'offrir au club MusiquÉTS plus de flexibilité dans l'évolution de l'application.

Développement d'une Application frontale :


- *Interface utilisateur simple* : Concevoir et implémenter une application frontale simple permettant aux utilisateurs de téléverser des partitions (format PDF ou images) et de récupérer les fichiers MIDI générés par l'application dorsale.

Validation et Tests :

- *Séries de tests exhaustives* : Développer des séries de tests exhaustives pour valider la robustesse de l'application, vérifier la précision de la reconnaissance optique de musique et déterminer les limites de la solution proposée.

Documentations et guide d'utilisation :

- *Documentation technique* : Rédiger une documentation technique comprenant différents artefacts utiles pour la prise en main du système, tels que des diagrammes d'architecture, des diagrammes de classes et des diagrammes de séquence.
- *Guide utilisateur* : Produire un guide utilisateur détaillé expliquant comment utiliser l'application, incluant les forces et les limites de l'application, afin de faciliter l'adoption par les utilisateurs finaux.

 Département de génie logiciel et des TI	PROJET <b>LOG/GTI 795</b>	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION <b>1.10</b>
	TITRE PFE 008	PAGE <b>6</b>		PAGES <b>22</b>

## 2.2 Objectifs secondaires (non obligatoires)

Ces objectifs seraient intéressants à ajouter au projet, mais ne sont pas obligatoires dans le contexte limité en temps pour le projet.

Support de Multi-Instruments :

- *Génération de fichiers MIDI pour plusieurs instruments* : Étendre la capacité de l'application à générer des fichiers MIDI capables de jouer plusieurs instruments (au moins deux).
- *Reconnaissance multi-instruments* : Améliorer la lecture de partitions pour reconnaître plusieurs instruments sur une même feuille, en identifiant correctement les différentes portées et leurs attributs.

Amélioration de l'Interface utilisateur :

- *Interface utilisateur avancée* : Apporter des améliorations à l'interface utilisateur pour rendre l'application plus intuitive et agréable à utiliser, facilitant l'interaction avec le système pour les utilisateurs. Ajouter une fonctionnalité pour effectuer la lecture des fichiers MIDI générés.

Conception et Implémentation d'une Base de Données :


- *Base de données pour stockage et gestion* : Concevoir et implémenter une base de données connectée à notre application dorsale pour stocker les partitions téléchargées, les fichiers MIDI générés, et éventuellement d'autres métadonnées pertinentes.

Pour plus d'information sur les objectifs, voir le document d'analyse des besoins.

## 2.3 Retombées du projet

Nous espérons que suite à la complétion de ce projet, plusieurs bénéfices tangibles et intangibles vont se manifester pour le club étudiant MusiquÉTS :

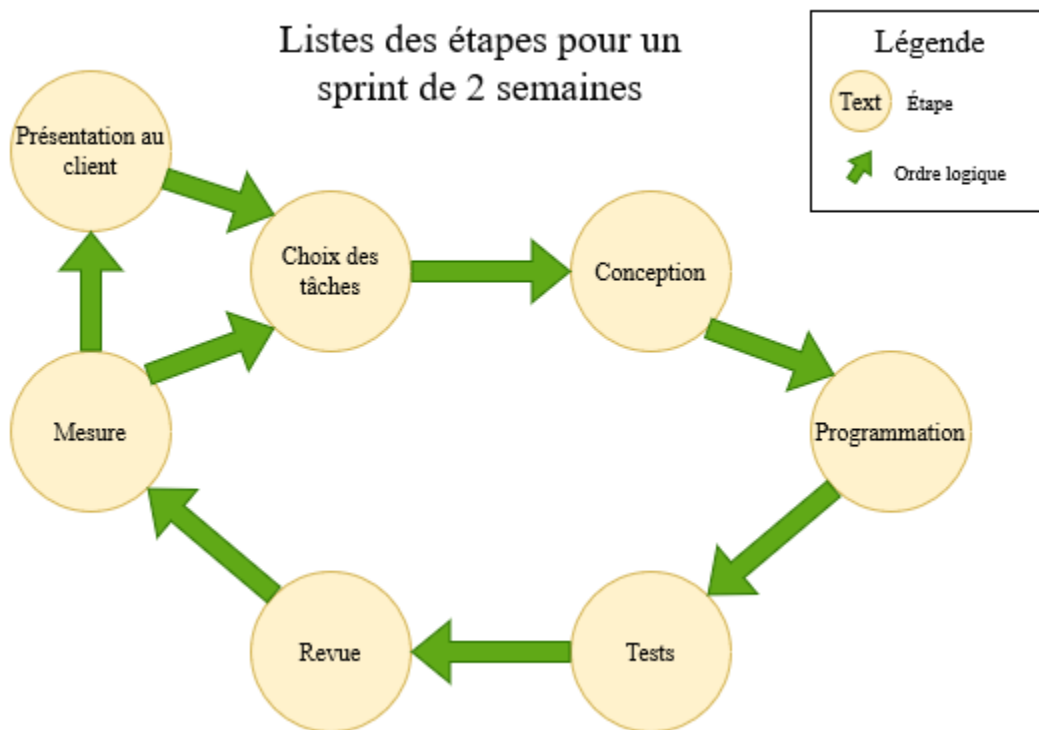
- *Automatisation de la collecte de partitions* : Le projet permettra au club de collecter et numériser des partitions musicales de manière automatisée, réduisant ainsi le besoin de saisie manuelle et de traitements fastidieux.
- *Base solide pour futurs développements* : La réalisation de ce projet fournira une base technologique solide que le club pourra étendre et améliorer à l'avenir. Cela ouvrira des possibilités pour des fonctionnalités supplémentaires et des améliorations continues.
- *Visibilité et attractivité du club* : La réussite de ce projet pourrait renforcer la visibilité du club MusiquÉTS au sein de l'École de technologie supérieure et pourrait attirer de nouveaux membres intéressés par la musique et les technologies de type OMR.

 Département de génie logiciel et des TI	PROJET LOG/GTI 795	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION 1.10
	TITRE PFE 008	PAGE 7		PAGES 22

### 3. MÉTHODOLOGIE


Comme la plupart des projets en informatique, des modifications et des changements seront demandés par le client au cours de ce projet. Nous allons utiliser la méthode Agile afin de capturer les demandes de changement le plus rapidement possible et ainsi nous adapter pour éviter tout retard potentiel.

Nous effectuerons des “sprints” de deux semaines. À l’exception des deux premiers sprints, chaque période de deux semaines respectera le schéma suivant :



**Figure 1** : Schématisation des étapes et tâches d'un sprint de deux semaines.

- Choix des tâches** : Avec ou sans le client, l'équipe choisira les tâches sur lesquelles elle devra se concentrer lors du prochain sprint. Il sera aussi possible de créer de nouvelles tâches à la demande du client lors de cette étape.
- Conception** : L'équipe concevra la solution pour une tâche donnée. La plupart de la documentation technique sera produite lors de cette étape. La documentation sera amenée à être modifiée lors des étapes suivantes. C'est aussi à cette étape que les choix seront effectués.

 Département de génie logiciel et des TI	PROJET <b>LOG/GTI 795</b>	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION <b>1.10</b>
	TITRE PFE 008	PAGE 8		PAGES 22

3. **Programmation** : L'équipe implémentera les solutions trouvées lors de la phase de conception.
4. **Tests** : L'équipe implémentera la suite de tests suivant les directives du plan de tests pour les solutions programmées.
5. **Revu** : Les membres de l'équipe devront effectuer une revue du code des autres membres pour s'assurer qu'ils aient une compréhension complète du projet en plus de détecter les erreurs possibles. Cette phase permettra de s'assurer que le code respecte les plus hauts standards de qualité en termes de programmation.
6. **Mesure** : L'équipe effectuera les mesures pour vérifier et prouver que les solutions et les tests respectent les contraintes et les exigences. Toutes les mesures seront documentées.
7. **Présentation au client** : Au cours du projet, deux prototypes seront officiellement présentés au client, cependant à la demande de celui-ci, l'équipe pourra présenter l'état de l'application à la fin du sprint.

Puisque nous utilisons une méthode agile, les étapes précédentes pourront se faire en parallèle (ex : programmation et tests).


De plus, comme spécifié plus haut, les deux premiers sprints ne respecteront pas le schéma. Étant au début du projet, des étapes préliminaires devront être faites proprement débiter le projet. Les étapes préliminaires sont :

- Rédaction du plan de projet
- Rédaction du document d'analyse des besoins
- Rédaction de l'analyse de risque
- Rédaction du plan de test
- Découpage des tâches

Il est à noter que la plupart des livrables seront modifiés et améliorés au cours du projet.


En temps normal, dans les principaux cadres agiles, il y a beaucoup de rencontres entre l'équipe (Stand-Up, Planification de sprint, Sprint review, post-mortem, etc). Puisqu'aucun des membres du projet ne travaille sur celui-ci à temps plein et qu'en plus il ne dure qu'environ 3 mois, nous avons décidé que durant les sprints, il y aurait deux rencontres officielles (planification de sprint et rencontre avec les professeurs), et des rencontres non officielles (rencontre entre les membres).



 Département de génie logiciel et des TI	PROJET <b>LOG/GTI 795</b>	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION <b>1.10</b>
	TITRE PFE 008	PAGE 9		PAGES 22


### 3.1 Tableau des échéanciers

Nom de la tâche ou du livrable	Début	Fin	Durée
Rédaction du plan de projet	15 mai 2024	22 mai 2024	7 jours
Rédaction du plan d'analyse des besoins	15 mai 2024	29 mai 2024	14 jours
Rédaction de l'analyse de risque	15 mai 2024	29 mai 2024	14 jours
Rédaction du plan de test	15 mai 2024	29 mai 2024	14 jours
Découpage des tâches	15 mai 2024	29 mai 2024	14 jours
Analyses et tests des solutions déjà existantes	22 mai 2024	29 mai 2024	7 jours
Rédaction compte rendu des analyses et tests des solutions existantes	22 mai 2024	29 mai 2024	7 jours
Planification du sprint #1	29 mai 2024	29 mai 2024	1 jour
Exécution du sprint #1	29 mai 2024	12 juin 2024	14 jours
Rétroaction (sprint #1)	12 juin 2024	12 juin 2024	1 jour
Planification du sprint #2	12 juin 2024	12 juin 2024	1 jour
Exécution du sprint #2	12 juin 2024	26 juin 2024	14 jours
Rétroaction (sprint #2)*	26 juin 2024	26 juin 2024	1 jour

 Département de génie logiciel et des TI	PROJET LOG/GTI 795	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION 1.10
	TITRE PFE 008	PAGE 10		PAGES 22


<b>Démonstration du prototype #1*</b>	<b>26 juin 2024</b>	<b>26 juin 2024</b>	<b>1 jour</b>
Planification du sprint #3*	26 juin 2024	26 juin 2024	1 jour
Exécution du sprint #3	26 juin 2024	10 juillet 2024	14 jours
Rétroaction (sprint #3)	10 juillet 2024	10 juillet 2024	1 jour
Planification du sprint #4	10 juillet 2024	10 juillet 2024	1 jour
Exécution du sprint #4	10 juillet 2024	24 juillet 2024	14 jours
Rétroaction (sprint #4)	24 juillet 2024	24 juillet 2024	1 jour
<b>Démonstration du prototype #2</b>	<b>24 juillet 2024</b>	<b>24 juillet 2024</b>	<b>1 jour</b>
Planification du sprint #5	24 juillet 2024	24 juillet 2024	1 jour
Exécution du sprint #5	24 juillet 2024	7 août 2024	14 jours
Rétroaction (sprint #5)	7 août 2024	7 août 2024	1 jour
<b>Démonstration finale</b>	<b>7 août 2024</b>	<b>7 août 2024</b>	<b>1 jour</b>
<b>Fin du projet, remise du projet</b>	<b>8 août 2024</b>	<b>8 août 2024</b>	<b>1 jour</b>

\* La date de la démonstration #1, de la réintroduction du sprint #2 et de la planification du sprint #3, déterminés pour le 26 juin seront à réévaluer pour être à l'intérieur des 7 jours suivants considérant une permutation d'horaire de l'école qui ne permet pas la rencontre.

 Département de génie logiciel et des TI	PROJET <b>LOG/GTI 795</b>	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION <b>1.10</b>
	TITRE PFE 008	PAGE 11		PAGES 22

#### 4. COMPOSITION DE L'ÉQUIPE ET RÔLES

Prénom	Rôle(s) (voir annexe A)	Responsabilités
1. Philippe Langevin	Chef d'équipe/Développeur	Rédige les documents administratifs, organise les réunions, prise de notes pendant les réunions Rédige la documentation en lien avec le rapport Écris le code
2. Charlie Poncsak	Développeur	Rédige la documentation en lien avec le rapport Écris le code
3. Laflèche Chevette	Animateur/Développeur	Anime les rencontres Rédige la documentation en lien avec le rapport Écris le code
4. Jérémy Da Costa	Développeur	Rédige la documentation en lien avec le rapport Écris le code
5. Xavier Jeanson	Développeur	Rédige la documentation en lien avec le rapport Écris le code

 Département de génie logiciel et des TI	PROJET LOG/GTI 795	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION 1.10
	TITRE PFE 008	PAGE 12		PAGES 22

## 5. LIVRABLES ET PLANIFICATION


### 5.1 Description des artéfacts

*Ajustez selon la nature du projet.*

Nom de l'artefact	Description
Plan de projet	Document décrivant le plan et la direction du projet dans son ensemble.
Document d'analyse des besoins	Document de vision qui englobe les contraintes et les exigences du client, fonctionnelles, non fonctionnelles et autres.
Analyse de risque	Document permettant d'analyser les risques possibles du projet ainsi que les solutions à chacun des risques
Plan de tests	Document planifiant les tests utilisés lors du développement.
Prototype 1	Le premier prototype qui sera présenté au client
Prototype 2	Le second prototype qui sera présenté au client
Prototype final	L'application finale
Rapport final	Le rapport final ne sert qu'à l'équipe dans le cadre du PFE

### 5.2 Planification


*Voir Annexe A*

 Département de génie logiciel et des TI	PROJET <b>LOG/GTI 795</b>	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION <b>1.10</b>
	TITRE PFE 008	PAGE 13		PAGES 22

## 6. RISQUES

L'échelle d'impact et de probabilité est décomposée sur trois tiers: Faible - Moyen - Fort

Risque	Impact	Probabilité	Mitigation / atténuation
Technologie nécessaire trop récente ou jeune	Moyen	Moyen	Effectuer une analyse préliminaire des solutions existantes.
Technologies existantes ne permettant pas d'atteindre le niveau de robustesse voulu	<b>Fort</b>	Moyen	Chercher des papiers de recherche sur l'OMR, plus précisément sur la robustesse des modèles.  Tester en profondeur les prototypes.
Difficulté à trouver des données d'entraînement ou de test	<b>Fort</b>	Faible	Prendre note des données utilisées pour les solutions existantes.
Difficulté à tester les prototypes	Moyen	Faible	Prendre note des tests effectués pour les solutions existantes.
Manque de ressources (base de données, serveurs, etc.)	Faible	<b>Fort</b>	Mettre en place les attentes et exigences du client quant au livrable final à remettre.
Projets existants difficiles à modifier	<b>Fort</b>	Faible	Observer l'architecture et la conception des solutions existantes lors de leur analyse.
Équipe démotivée ou pas à son affaire	Moyen	Faible	Bien définir les rôles des membres.  Bien définir la séparation des tâches et en faire un suivi régulier et ouvert aux changements.

 Département de génie logiciel et des TI	PROJET LOG/GTI 795	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION 1.10
	TITRE PFE 008	PAGE 14		PAGES 22

## 7. TECHNIQUES ET OUTILS

Pour le développement du projet, nos choix d'outils, de technologies et de techniques ont été guidés par plusieurs facteurs clés : la durée limitée du projet (4 mois), les compétences et points faibles des membres de l'équipe, ainsi que les besoins et critères spécifiques du client. Nous avons sélectionné des outils et technologies qui nous permettent de travailler efficacement, de manière collaborative et qui répondent aux exigences techniques du projet.


En plus de cela, nous avons effectué plusieurs tests de technologies déjà existantes pour évaluer leurs capacités et leur adéquation avec les exigences du projet. Nous avons construit une matrice décisionnelle pour comparer ces technologies sur divers critères importants pour le client et le projet. Ce processus nous a conduits à choisir Audiveris comme base technologique pour notre solution de reconnaissance optique de musique.

Il est à noter que ces choix technologiques peuvent évoluer au cours du projet ou après la présentation de notre premier prototype. Certaines technologies, comme les bibliothèques de développement spécifiques, ne sont pas mentionnées ici, car leur impact est minime en comparaison des outils principaux.

Pour plus d'infos, voir le document intitulé matrice évaluatrice des solutions existantes.

### 7.1 Outils et techniques de gestion de projet

- *Gestion de projet agile* : Méthodologie de gestion de projet qui privilégie l'adaptation aux changements, les livrables fréquents et la collaboration étroite avec le client. Nous l'utiliserons pour mieux nous adapter aux exigences changeantes et livrer des résultats de qualité régulièrement.
- *Github* : Une plateforme de développement collaboratif, permettant de gérer le code source, suivre les modifications, et collaborer efficacement avec l'équipe. Il sera utilisé pour héberger le code du projet et gérer les contributions.
- *Github projects* : Un outil de gestion de projet intégré à GitHub, facilitant la planification, le suivi des tâches et la gestion des flux de travail. Il nous aidera à organiser et prioriser les tâches de développement de manière structurée.
- *Google Drive* : Une solution de stockage en ligne et de partage de fichiers qui nous permettra de centraliser et partager facilement les documents, rapports et autres ressources du projet avec l'équipe.


 Département de génie logiciel et des TI	PROJET LOG/GTI 795	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION 1.10
	TITRE PFE 008	PAGE 15		PAGES 22

## 7.2 Outils de communication

- *Discord* : Une plateforme de communication vocale et textuelle. Nous l'utiliserons pour les discussions informelles, les sessions de travail collaboratif et les échanges rapides entre les membres de l'équipe, les professeurs et le client.
- *Microsoft Teams* : Outil de collaboration qui intègre des fonctionnalités de messagerie, de visioconférence et de partage de fichiers. Il sera utilisé pour les réunions formelles, les présentations de projet et la communication avec les professeurs, le client et toutes autres parties prenantes extérieures.
- *Outlook* : Application de messagerie électronique qui nous permettra de gérer les courriels et les calendriers. Elle sera utilisée pour la communication officielle et la gestion des rendez-vous et échéances.

## 7.3 Technologies et outils de développement.

- *Java* : Langage de programmation orienté objet robuste et portable, choisi pour le développement de l'algorithme de reconnaissance optique de musique.
- *Spring boot* : Cadriciel choisi pour faire la gestion d'appel API pour l'application dorsale (*back-end*). Le cadriciel en langage Java fera aussi la gestion d'appels avec le programme Audiveris qui sera utilisé.
- *Audiveris* : Logiciel open-source de reconnaissance optique de musique qui sera modifié, adapté et amélioré pour répondre aux besoins spécifiques du projet. L'engin OMR d'Audiveris fournira la base technologique pour la conversion des partitions en fichiers MIDI.
- *Git* : Système de contrôle de version décentralisé qui permettra de suivre les modifications du code source, de gérer les branches de développement et de faciliter la collaboration entre les membres de notre équipe. Git assurera la gestion efficace du code tout au long du projet.
- *IDE (Environnement de Développement Intégré)* : Logiciel qui fournit des outils complets pour l'édition, la compilation, le débogage et le test du code. Nous utiliserons des IDEs pour le développement de l'application dorsale et frontale. L'IDE final n'est pas choisi.
- *React* : Une bibliothèque JavaScript pour la construction d'interfaces utilisateur, utilisée pour développer l'application frontale. React sera notre technologie frontale principale. Elle permettra de créer une interface utilisateur réactive et dynamique pour téléverser les partitions et récupérer les fichiers MIDI générés.

 Département de génie logiciel et des TI	PROJET LOG/GTI 795	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION 1.10
	TITRE PFE 008	PAGE 16		PAGES 22

## 8. OBJECTIFS DE RÉUSSITE

Voici une liste d'objectifs que nous devons atteindre dans le projet afin de considérer celui-ci comme réalisé :


- Les limites de l'application sont claires, précises, testées et documentées.
- Chaque fonctionnalité est accompagnée de documentation (architecture, use case, etc.)
- L'application dorsale fonctionne indépendamment. Elle doit être capable de traiter les entrées et de produire les sorties sans dépendance externes.
- Le logiciel détecte et reproduit correctement :
  - Clés (Sol et Fa)
  - Notes (Do, ré, mi, fa, sol, la, si)
  - Accidents (dièse, bémol, bécarré)
  - Silences (pause, demi-pause, soupir, demi-soupir, quart de soupir)
  - Figures de note (ronde, blanche, noire, croche, double croche)
  - Armure
  - Indication de mesure
  - Articulations (Staccato, accents)
  - Barres de mesure
- Le logiciel s'exécute dans un temps raisonnable, c'est-à-dire en moins d'une heure pour les plus grosses partitions.
- Chaque fonctionnalité est testée par des tests unitaires et des tests d'intégrations.

## 9. CONTRAINTES

### 9.1 Contraintes fonctionnelles

- **Reconnaissance des notes** : Le logiciel doit détecter et identifier les notes d'une partition musicale pour la clé de sol et de fa.
- **Détection des accidents** : Le logiciel doit détecter et identifier les accidents dans une mesure.
- **Reconnaissance des silences** : Le logiciel doit détecter et identifier les silences.
- **Identification des figures de notes** : Le logiciel doit détecter et identifier les figures de notes.
- **Détection des clés** : Le logiciel doit détecter la clé de sol et la clé de Fa en plus d'adapter les notes en conséquence de celles-ci.
- **Lecture de l'armure** : Le logiciel doit être capable de lire l'armure et d'appliquer les accidents provenant de celle-ci.
- **Reconnaissance des indications de mesure** : Le logiciel doit détecter et comprendre les indications de mesure.
- **Identification des articulations** : Le logiciel doit détecter, identifier jouer certaines articulations (staccato et accents).
- **Détection des barres de mesure** : Le logiciel doit être capable de détecter et comprendre les barres de mesure, notamment lorsqu'il s'agit de la barre de reprise.
- **Utilisation du GUI** : L'interface graphique doit pouvoir servir de guide simplifié de l'utilisation de l'API du backend. Si une route existe dans l'application dorsale, elle doit être accessible sur l'application frontale.



 Département de génie logiciel et des TI	PROJET <b>LOG/GTI 795</b>	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION <b>1.10</b>
	TITRE PFE 008	PAGE 17		PAGES 22

- **Lecture des fichiers PDF** : Le logiciel doit être capable de lire les partitions sous le format PDF. Les partitions peuvent être une image ou une partition numérique.
- **Création d'un fichier MIDI** : Le logiciel doit produire un fichier MIDI à partir de la partition reçue en entrée. Cela ne l'empêche pas de pouvoir produire des sorties intermédiaires (ex : musicxml).

## 9.2 Contraintes non-fonctionnelles

### 9.2.1 Aptitude fonctionnelle

#### 9.2.1.1 Exhaustivité fonctionnelle

- Le système doit être capable de lire des partitions musicales sous format PDF, détecter les notes, les silences, les clés, les armures, et autres symboles musicaux, et générer un fichier MIDI conforme à la partition d'origine.

#### 9.2.1.2 Exactitude fonctionnelle

- La reconnaissance des éléments musicaux doit être précise, avec un taux d'exactitude respectant le taux de réussite établi.

#### 9.2.1.3 Pertinence fonctionnelle

- Le système doit produire des fichiers MIDI qui respectent fidèlement la partition musicale originale en termes de notes, rythmes et dynamiques.

### 9.2.2 Efficacité de performance

#### 9.2.2.1 Utilisation des ressources

- L'application doit utiliser les ressources système de manière efficace, sans surcharger la mémoire ou le processeur pour éviter des pannes potentielles du système.

#### 9.2.2.2 Capacité

- Le système doit pouvoir traiter des partitions musicales de différentes tailles et complexités.

### 9.2.3 Compatibilité

#### 9.2.3.1 Coexistence

- L'application doit pouvoir fonctionner en parallèle avec d'autres applications sur le même système sans conflit.


### 9.2.4 Utilisabilité

#### 9.2.4.1 Reconnaissance de l'adéquation

- L'interface utilisateur doit être intuitive, permettant aux utilisateurs de comprendre facilement comment utiliser l'application.

#### 9.2.4.2 Apprentissage

- Le système doit être facile à apprendre, avec une courbe d'apprentissage minimale pour les nouveaux utilisateurs du club MusiquÉTS.

 Département de génie logiciel et des TI	PROJET <b>LOG/GTI 795</b>	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION <b>1.10</b>
	TITRE PFE 008	PAGE 18		PAGES 22

#### 9.2.4.3 Exploitabilité

- L'interface utilisateur doit permettre une interaction facile et efficace avec le système.

### **9.2.5 Fiabilité**

#### 9.2.5.1 Maturité

- Le système doit être stable et fonctionner correctement dans des conditions normales d'utilisation.

#### 9.2.5.2 Tolérance aux pannes

- Le système doit être capable de gérer les erreurs et les défaillances de manière gracieuse, sans perte de données.

### **9.2.6 Maintenabilité**

#### 9.2.6.1 Modularité

- Le système doit être conçu de manière modulaire pour faciliter les mises à jour et les améliorations futures voulant être apportée par le club une fois le projet remis.

#### 9.2.6.2 Analysabilité


- Le code source doit être bien documenté pour permettre une analyse facile et rapide.

#### 9.2.6.3 Modifiabilité

- Le système doit être facilement modifiable pour permettre des ajustements et des améliorations.


#### 9.2.6.4 Testabilité

- Le système doit être conçu de manière à faciliter les tests unitaires et d'intégration.

 Département de génie logiciel et des TI	PROJET <b>LOG/GTI 795</b>	DOCUMENT NO. <b>0001</b>	DATE <b>2024-05-22</b>	VERSION <b>1.10</b>
	TITRE <b>PFE 008</b>	PAGE <b>19</b>		PAGES <b>22</b>

## 10. RÉFÉRENCES

*Vos références initiales. Par exemple: documents de projet existants, compilation de patrons, documentation sur la technologie, etc.*


 Département de génie logiciel et des TI	PROJET LOG/GTI 795	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION 1.10
	TITRE PFE 008	PAGE 20		PAGES 22

## ANNEXE A : PLAN DE TRAVAIL


Le tableau suivant présente la planification pour la réalisation des tâches ou artefacts décrits précédemment.

*Note : Les dates de début et de fin des itérations doivent être visibles dans ce plan. Compléter la liste des tâches et des jalons. Cependant, pour les tâches concrètes de développement de logiciel, ne préciser que les tâches pour la première itération. S'assurer que ces tâches sont exprimées en termes d'activités concrètes d'ingénierie, par exemple « réaliser le scénario X du cas d'utilisation Y » ou « concevoir et implémenter la portion serveur de la fonctionnalité Z » et non en termes d'activités abstraites comme « Analyser les besoins » ou en termes de production d'artefacts comme « Rédiger le document SRS ». Les artefacts affectés par une activité d'ingénierie sont indiqués dans la colonne Livrable(s)/Artefacts. Si le professeur superviseur assiste aux démonstrations de fin d'itération, ça peut être un moyen de le rencontrer.*

#	Commence	Termine	Tâches/Jalon	Livrable(s)/Artefacts	Responsable(s)
1	24 avril 2024	24 avril 2024	entrer en contact avec les superviseur		Chef d'équipe
2	8 mai 2024	9 mai 2024	organiser l'horaire des rencontre		Chef d'équipe
2.1	8 mai 2024	8 mai 2024	Rencontre – professeur superviseur		Animateur
2.2	15 mai 2024	15 mai 2024	Rencontre – client		Animateur
3	15 mai 2024	29 mai 2024	Remise du plan de projet	Plan de projet	Chef d'équipe
3.1	15 mai 2024	29 mai 2024	Remise du Document d'analyse de besoins	Document d'analyse de besoins	Chef d'équipe
3.2	15 mai 2024	29 mai 2024	Remise de l'analyse de risque	analyse de risque	Chef d'équipe
3.3	15 mai 2024	29 mai 2024	Remise du plan de test	plan de test	Chef d'équipe
4	29 mai 2024	26 juin 2024	Réalisation des sprint 1 et 2		toutes l'équipe
5	26 juin 2024	26 juin 2024	Démo / Rencontre – professeur superviseur		Animateur
6	26 juin 2024	10 juillet 2024	Réalisation du sprint 3		toutes l'équipe
6.1	10 juillet 2024	10 juillet 2024	Remise du rapport d'étape ( <i>facultatif</i> )	Rapport d'étape	
6.2	10 juillet 2024	24 juillet 2024	Réalisation du sprint 4		toutes l'équipe
7	24 juillet 2024	24 juillet 2024	Rencontre – professeur superviseur		Animateur
8	24 juillet 2024	7 août 2024	Réalisation du sprint 5		toutes l'équipe
9.1	7 août 2024	7 août 2024	Présentation	Présentation	Animateur
9.2	8 août 2024	8 août 2024	Remise du projet	Projet	Chef d'équipe

 Département de génie logiciel et des TI	PROJET LOG/GTI 795	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION 1.10
	TITRE PFE 008	PAGE 21		PAGES 22

#	Commence	Termine	Tâches/Jalon	Livrable(s)/Artéfacts	Responsable(s)
1	24 avril 2024	24 avril 2024	entrer en contact avec les superviseur		Chef d'équipe
2	8 mai 2024	9 mai 2024	organiser l'horaire des rencontre		Chef d'équipe
2.1	8 mai 2024	8 mai 2024	Rencontre – professeur superviseur		Animateur
2.2	15 mai 2024	15 mai 2024	Rencontre – client		Animateur
3	15 mai 2024	29 mai 2024	Remise du plan de projet	Plan de projet	Chef d'équipe
3.1	15 mai 2024	29 mai 2024	Remise du Document d'analyse de besoins	Document d'analyse de besoins	Chef d'équipe
3.2	15 mai 2024	29 mai 2024	Remise de l'analyse de risque	analyse de risque	Chef d'équipe
3.3	15 mai 2024	29 mai 2024	Remise du plan de test	plan de test	Chef d'équipe
4	29 mai 2024	26 juin 2024	Réalisation des sprint 1 et 2		toutes l'équipe
5	26 juin 2024	26 juin 2024	Démo / Rencontre – professeur superviseur		Animateur
6	26 juin 2024	10 juillet 2024	Réalisation du sprint 3		toutes l'équipe
6.1	10 juillet 2024	10 juillet 2024	Remise du rapport d'étape ( <i>facultatif</i> )	Rapport d'étape	
6.2	10 juillet 2024	24 juillet 2024	Réalisation du sprint 4		toutes l'équipe
7	24 juillet 2024	24 juillet 2024	Rencontre – professeur superviseur		Animateur
8	24 juillet 2024	7 août 2024	Réalisation du sprint 5		toutes l'équipe
			Remise du travail	Rapport	

 Département de génie logiciel et des TI	PROJET LOG/GTI 795	DOCUMENT NO. 0001	DATE 2024-05-22	VERSION 1.10
	TITRE PFE 008	PAGE 22		PAGES 22

## ANNEXE B : LISTE DES ÉLÉMENTS D'UNE PARTITION

**Notes** : Do, Ré, Mi, Fa, Sol, La, Si

**Accidents** : Dièse (#), Double dièse (x), Bémol (b), Double bémol (bb), Bécasse (b)

**Silences** : Pause, Demi-pause, Soupir, Demi-soupir, Quart de soupir, Huitième de soupir, Seizième de soupir

**Figures de Note** : Ronde, Blanche, Noire, Croche, Double croche, Triple croche, Quadruple croche

**Clés** : Clé de sol (C), Clé de fa (F), Clé d'ut (C)

**Armures** : Armure de dièses (jusqu'à 7 dièses), Armure de bémols (jusqu'à 7 bémols)

**Indications de Mesure**

**Indications de Tempo** : Marquage du tempo (Andante, Allegro, ...), BPM

**Ligatures** : Liaison, Liaison de phrasé

**Articulations** : Staccato (point), Accent (>), Tenuto (—), Marcato (^)

**Ornements** : Trille, Mordant, Gruppetto

**Dynamique** : Pianissimo (pp), Piano (p), Mezzo-piano (mp), Mezzo-forte (mf), Forte (f), Fortissimo (ff), Crescendo (crescendo), Decrescendo (decrescendo)

**Barres de Mesure** : Barre simple, Double barre, Barre de reprise (ouverte et fermée)

**Autres** : Clefs de lecture, Indications de répétition, Signes de coda et de segno, ghost note