



ÉCOLE POLYTECHNIQUE DE L'UNIVERSITÉ DE TOURS

64, Avenue Jean Portalis

37200 TOURS, FRANCE

Tél. (33)2-47-36-14-14

Fax (33)2-47-36-14-22

www.polytech.univ-tours.fr

Rapport

Projet de Programmation et Génie Logiciel

Analyse et représentation de signaux audio

Auteur(s)

Théo Souchon

[\[theo.souchon@etu.univ-tours.fr\]](mailto:theo.souchon@etu.univ-tours.fr)

Michelle Pepe

[\[michellepepe97@gmail.com\]](mailto:michellepepe97@gmail.com)

Mohamed Mouhameden

[\[mouhameden.mohameden@etu.univ-tours.fr\]](mailto:mouhameden.mohameden@etu.univ-tours.fr)

Encadrant(s)

Pascal Makris

[\[pascal.makris@univ-tours.fr\]](mailto:pascal.makris@univ-tours.fr)

Polytech Tours

Département Informatique

Version du 12 janvier 2023

Table des matières

Remerciement	4
1 Contexte général du contexte	5
1.1 Présentation de l'organisation d'accueil	5
1.1.1 Université de Tours	5
1.1.2 Département informatique	5
1.1.3 Laboratoire	6
1.2 Contexte du projet	6
1.2.1 Motivations du projet	6
1.2.2 Objectifs et engagement du projet	6
1.2.3 Conduite de projet et génie logiciel	6
1.2.4 Répartition des tâches	7
2 Étude de l'existant et proposition d'une solution	9
2.1 Situation actuelle	9
2.2 Moyens mis à disposition	9
2.2.1 Techniques	9
2.2.2 Technologies	9
3 Analyse et Conception	10
3.1 Analyse et Spécifications	10
3.1.1 Identification des acteurs	10
3.1.2 Diagramme de cas d'utilisation	10
3.1.3 Description des scenarios de cas d'utilisation	11
3.2 Conception de la solution	13
3.2.1 Modèle statique (diagramme de classe)	13
4 Mise en œuvre de la solution	14
4.1 Environnement de développement	14
4.1.1 La technologie Java	14
4.1.2 L'architecture MVC	14
4.1.3 Librairies utilisées	14
4.1.4 Les outils de développement	14
4.2 Tests et exemples d'exécution	15
4.2.1 Tests des algorithmes	15
4.2.2 Tests des données	16
5 Conclusion et perspectives	17

Table des figures

1.1	Diagramme PERT effectif	7
1.2	Diagramme MOA/MOE	7
1.3	Diagramme de Gantt	8
3.1	Diagramme de cas d'utilisation	10
3.2	Diagramme de classe	13
4.1	Tests de différents algorithmes pour un même signal	15
4.2	Tests des données selon l'algorithme n°2	16

Remerciement

Nous tenons tout d'abord à remercier M. Makris, notre tuteur de projet au sein de Polytech, qui occupe la fonction de client et qui nous a accueillis comme étudiants débutants. Il nous a beaucoup appris sur les enjeux des projets de génie logiciel et sur les missions dont nous étions responsables. Durant notre mission, nous avons eu la chance de bénéficier de ses conseils et nous voulons le remercier pour le temps qu'il nous a accordé et pour son aide précieuse.

Nous souhaitons également exprimer nos remerciements les plus sincères aux équipes pédagogiques et administratives de Polytech de Tours qui ont facilité nos démarches et ont assuré un suivi tout au long de notre travail.

Chapitre 1

Contexte général du contexte

1.1 Présentation de l'organisation d'accueil

1.1.1 Université de Tours

L'université de Tours a été créée en 1971, suite à la loi Faure qui visait à moderniser et à démocratiser les structures de l'enseignement supérieur en France après les événements de 1968. Elle regroupe des établissements d'enseignement supérieur qui existaient jusque-là, et conformément au code de l'éducation qui fixe l'organisation légale des universités publiques en France, l'université de Tours peut être divisée en plusieurs composantes. On peut y trouver sept unités de formation et de recherche (UFR), une École polytechnique universitaire d'ingénieurs (Polytech Tours) et deux instituts universitaires de technologie (IUT).

Connue internationalement comme premier opérateur de recherche en région Centre-Val de Loire, ces recherches s'étendent également au domaine de l'imagerie médicale, bio médicaments, de la cancérologie, et plus spécifiquement dans l'étude de l'autisme en partenariat avec l'Inserm.

1.1.2 Département informatique

Le département informatique de Polytech Tours est une filière d'enseignement spécialisée dans les nouvelles technologies et le numérique. Selon le site officiel de Polytech Tours, la filière informatique a plusieurs objectifs, tels que :

- Développer des compétences généralistes en informatique pour maîtriser les fondamentaux du domaine, en particulier les sciences de la décision, les techniques de développement de logiciels, l'intelligence artificielle, les systèmes d'information et les systèmes d'exploitation et les réseaux.

1.1.3 Laboratoire

Polytech Tours est une école qui abrite 4 laboratoires de recherche : LIFAT, LAMé, GREMAN et CITERES. Plus précisément, le laboratoire qui s'intègre à la filière informatique est le LIFAT (Laboratoire d'Informatique Fondamentale et Appliquée de Tours, ERL CNRS). En 2020, le LIFAT comptait 48 enseignants-chercheurs, 35 doctorants, 5 postdoctorants et 9 IATOS (titulaires et sous contrat). Il est structuré en trois équipes de recherche : Bases de données et traitement des langues naturelles (BdTLn), Recherche Opérationnelle, Ordonnancement et Transport (ROOT) et Reconnaissance des formes et analyse d'images (RFAI).

1.2 Contexte du projet

1.2.1 Motivations du projet

La motivation du projet est de développer un outil d'aide à la décision dans le domaine médical pour identifier des sons xiphoïdiens selon l'état pathologique du patient. Pour cela, nous procédons à l'extraction des informations à partir des signaux sonores. en utilisant les lois Zipf et Zipf inverse adaptés.

1.2.2 Objectifs et engagement du projet

L'objectif du projet est de comprendre les différentes méthodes scientifiques mises en œuvre (Zipf), de fournir une bibliothèque permettant de créer une image représentative du signal et de lier celle-ci à une interface graphique pour un utilisateur. Nous devons ainsi fournir une image ou un graphique représentatif de ce signal. Cependant, les motivations du projet ont évolué au cours du temps. Nous avons abandonné l'idée de l'application de la loi de Zipf en raison du manque de temps et de moyens. Nos efforts se sont donc concentrés sur la représentation graphique du signal par une image.

1.2.3 Conduite de projet et génie logiciel

Ce projet vise à être repris par d'autres groupes pour améliorer et approfondir nos analyses et notre développement autour du sujet. C'est pourquoi il était important d'adopter des méthodes de travail permettant de faciliter au maximum l'héritage du projet et une meilleure compréhension de celui-ci de l'extérieur. De plus, ce groupe de projet est constitué de 3 personnes de nationalités différentes, ce qui accentue l'importance de la bonne communication et des interactions au sein du groupe. Ainsi, nous avons réparti les rôles et les responsabilités visibles dans les diagrammes de relation (fig1.2) et de Gantt (fig1.3). Nous avons également opté pour une méthode de travail agile, dont les raisons sont expliquées ici (1.2.4). D'autres diagrammes sont par ailleurs présents dans ce rapport pour mieux comprendre et reprendre le projet. Enfin, l'architecture de développement du projet est basée sur l'exemple du modèle-vue-contrôleur (MVC), qui peut également être trouvé dans cette section du projet. (fig3.2)

1.2.4 Répartition des tâches

Nous avons adopté une méthode de travail très proche de la méthode "agile". Les raisons sont les suivantes : une communication constante avec le MOAd (Pascal Makris) était nécessaire et utile pour s'assurer d'une bonne gestion du temps et avancer dans la bonne direction. Tandis qu'une planification plus stricte, comme le cycle en V, aurait pu nous enfermer dans une direction qui aurait été la mauvaise. Finalement, le risque aurait été de livrer un projet qui n'aurait pas été en adéquation avec les besoins du client.

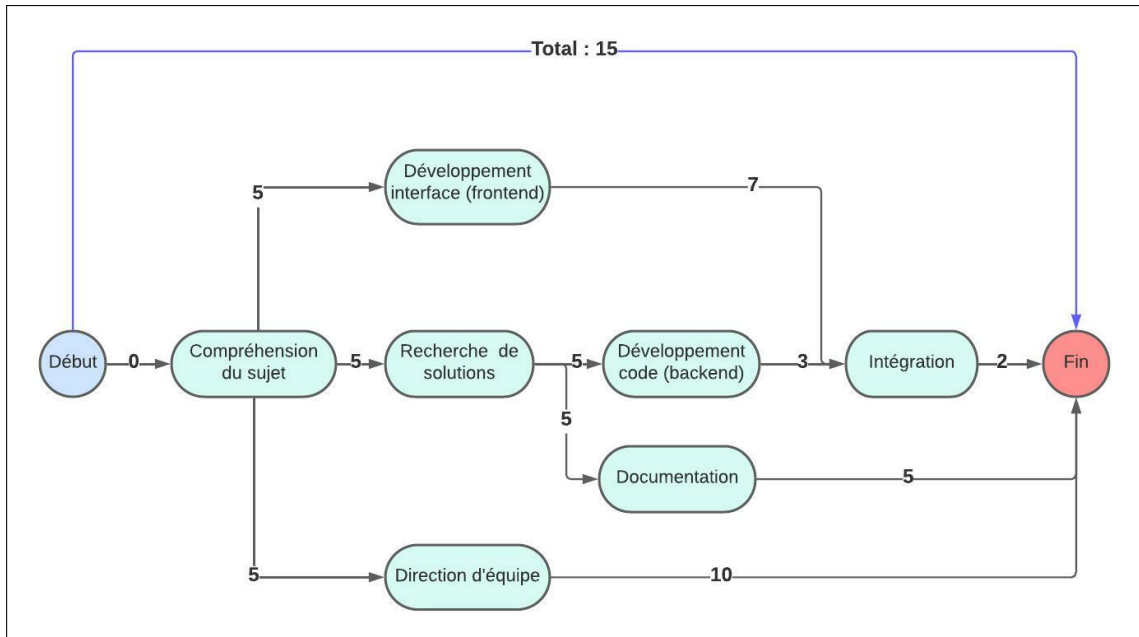


FIGURE 1.1 – Diagramme PERT effectif

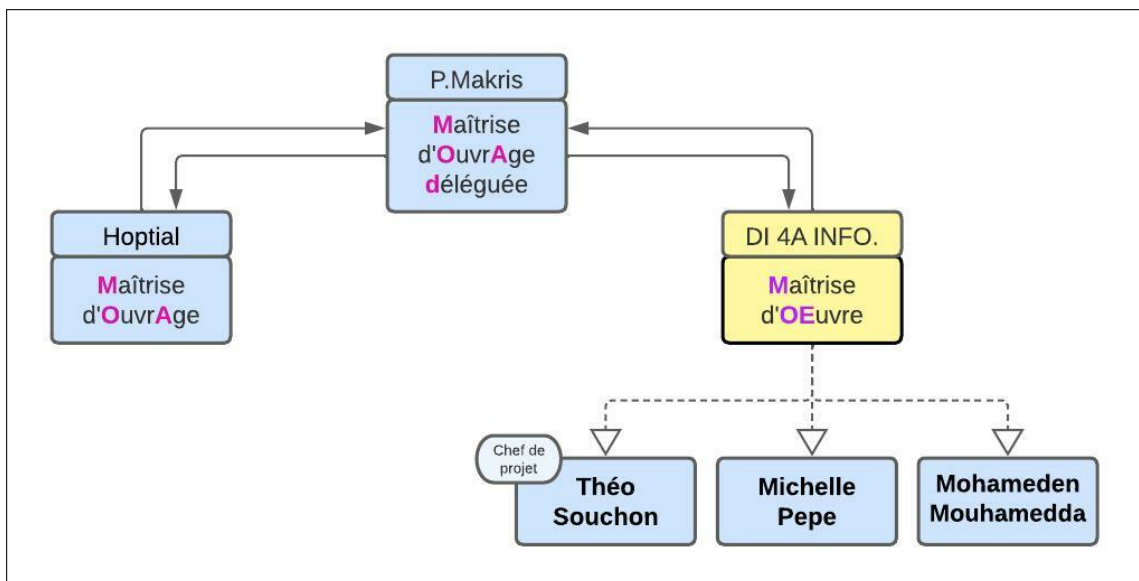


FIGURE 1.2 – Diagramme MOA/MOE

Ressources

Nom

Théo Souchon
Michelle Pepe
Mohameden Mouhamedda

Rôle par défaut

Chef de projet
Développeur
Chef de projet

Tâches

Nom

Date de début

Date de fin

Nom	Date de début	Date de fin
Compréhension du sujet	19/09/2022	23/10/2022
Développement Interface (frontend)	24/10/2022	11/12/2022
Recherche de solutions	24/10/2022	27/11/2022
Développement code (backend)	28/11/2022	18/12/2022
Intégration	19/12/2022	01/01/2023
Documentation	28/11/2022	01/01/2023
Direction d'équipe	24/10/2022	01/01/2023

Diagramme de Gantt

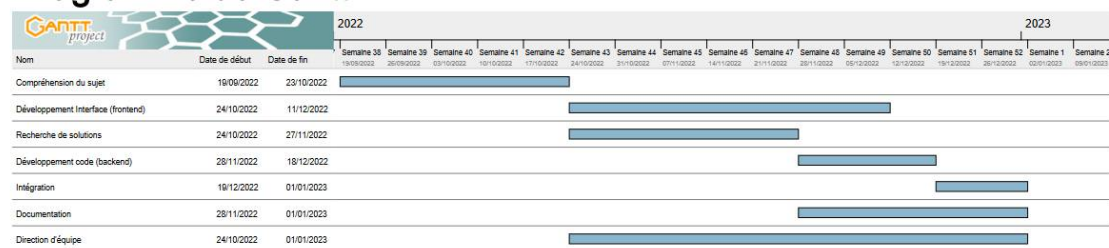


Diagramme des Ressources

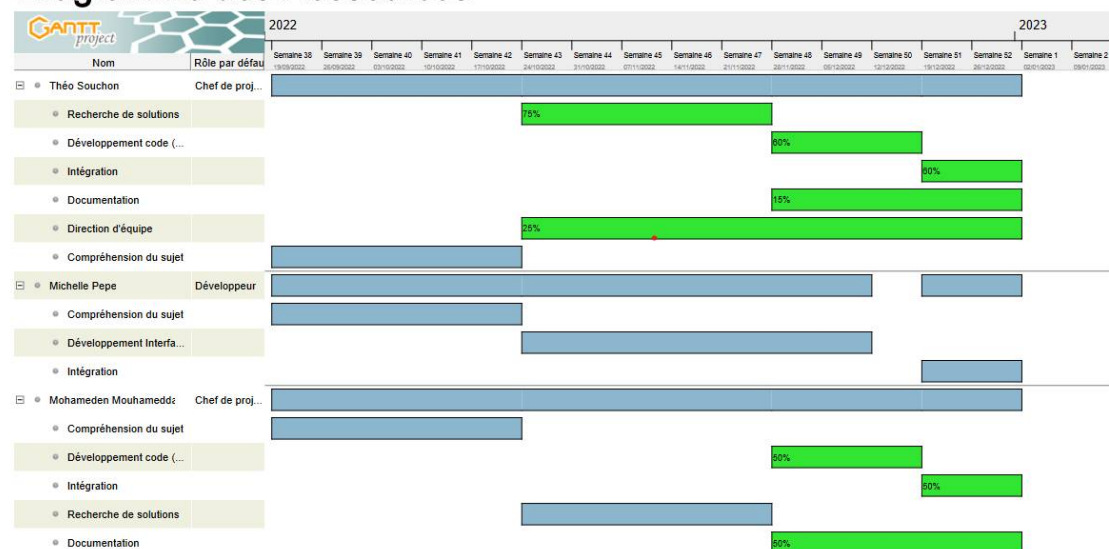


FIGURE 1.3 – Diagramme de Gantt

Chapitre 2

Étude de l'existant et proposition d'une solution

2.1 Situation actuelle

Il est important de noter que le projet sur lequel nous avons travaillé ne correspond pas au projet attribué initialement. En effet, le projet a évolué au cours du temps, d'abord pour des raisons administratives puis pour des raisons de moyens. Nous avons pu nous appuyer sur des travaux antérieurs qui ont été réalisés dans un sujet de thèse très similaire dont Monsieur Makris faisait partie. Cela nous a servi au début du projet pour obtenir une direction et des connaissances techniques pour commencer. Ces travaux étaient constitués de rapports de thèse et d'une application informatique. Malheureusement, nous n'avons pas pu obtenir de véritables données sonores de l'hôpital en raison du manque de réponse.

2.2 Moyens mis à disposition

2.2.1 Techniques

Différentes techniques ont été nécessaires dans le projet. Le projet repose sur les principes scientifiques des transformations de Fourier, c'est-à-dire la transformation en fréquence de signaux sonores. La technique du spectrogramme est dérivée de ce même principe, car elle consiste en une succession de transformations de Fourier dans le temps. De plus, les techniques de génie logiciel et de gestion de projet ont également été mises en œuvre.

2.2.2 Technologies

En termes de technologie, nous utilisons 3 principaux thèmes d'application. Le front-end est géré par des vues XML et du style CSS. Le back-end est géré par du Java et du Python. Les données sont stockées dans une base de données. À cela s'ajoutent des technologies utilitaires telles que Maven afin d'importer automatiquement les bibliothèques nécessaires à l'importation du projet, ce qui facilite l'héritage du projet. Il est également important de noter que nous avons travaillé ensemble en utilisant GitHub.

Chapitre 3

Analyse et Conception

3.1 Analyse et Spécifications

3.1.1 Identification des acteurs

Utilisateur connecté : Utilisateur connecté à un compte existant. Il pourra convertir un signal, consulter les informations du centre, consulter les informations de contact du centre.

Utilisateur non connecté : Utilisateur qui n'est pas connecté à un compte (ou première connexion), il ne peut que se connecter ou se créer un compte.

3.1.2 Diagramme de cas d'utilisation

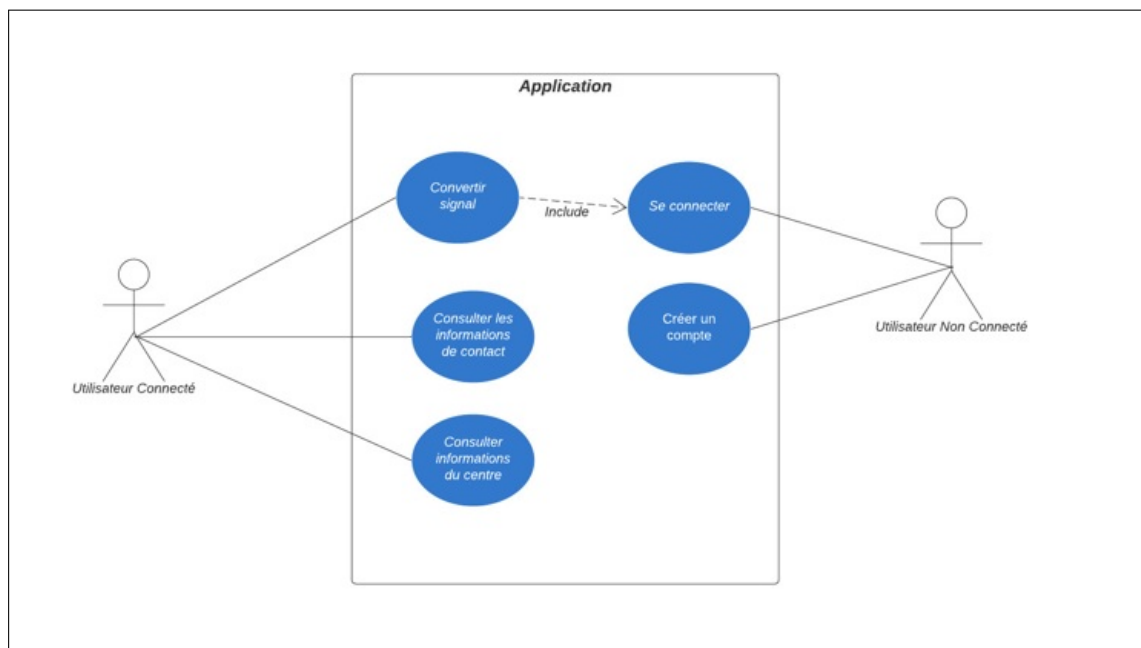


FIGURE 3.1 – Diagramme de cas d'utilisation

Lors de la première connexion d'un utilisateur, deux choix s'offrent à lui : se connecter à un compte existant ou créer un compte auquel se connecter. Une fois connecté, l'utilisateur a accès à 3 actions : consulter les informations personnelles de son compte, consulter le blog lié à l'hôpital ou toute autre source d'information, et convertir un signal afin d'obtenir un spectrogramme.

3.1.3 Description des scenarios de cas d'utilisation

Cas n°1 : Se connecter

Précondition : Utilisateur non connecté

Scénario principal :

1. Remplir les champs "Username", "Password"
2. Les champs sont valides et correspondent à un compte existant
3. L'utilisateur est connecté et passe à une autre page

Scénario particulier :

1. Remplir les champs "Username", "Password"
 2. Les champs sont invalides
 3. Un message d'erreur apparait disant que le username et/ou password sont incorrectes
-

Cas n°2 : S'inscrire

Précondition : Utilisateur non connecté

Scénario principal :

1. Remplir les champs "Name", "Email", "Password"
2. Les champs sont valides et l'username disponible
3. Le compte est créé

Scénario particulier :

1. Remplir les champs "Name", "Email", "Password"
 2. Cliquer sur le bouton inscription
 3. Le champ "username" est déjà utilisé par un autre compte, apparition d'un popup d'erreur
-

Cas n°3 : Convertir un signal

Précondition : Utilisateur connecté

Scénario principal :

1. L'utilisateur clique sur le bouton "Conversion"
2. Un explorateur de fichier s'ouvre
3. L'utilisateur sélectionne le fichier sonore sur sa machine
4. Le spectrogramme du son s'affiche

Scénario particulier : /

Cas n°4 : Consulter les informations de contact

Précondition : Utilisateur connecté

Scénario principal :

1. L'utilisateur clique sur le bouton Contacts
2. Les informations pour contacter l'hôpital s'affichent

Scénario particulier : /

Cas n°5 : Consulter le profil de l'hôpital

Précondition : Utilisateur connecté

Scénario principal :

1. L'utilisateur clique sur le bouton "Profile"
2. Les informations sur l'hôpital s'affichent

Scénario particulier : /

3.2 Conception de la solution

3.2.1 Modèle statique (diagramme de classe)

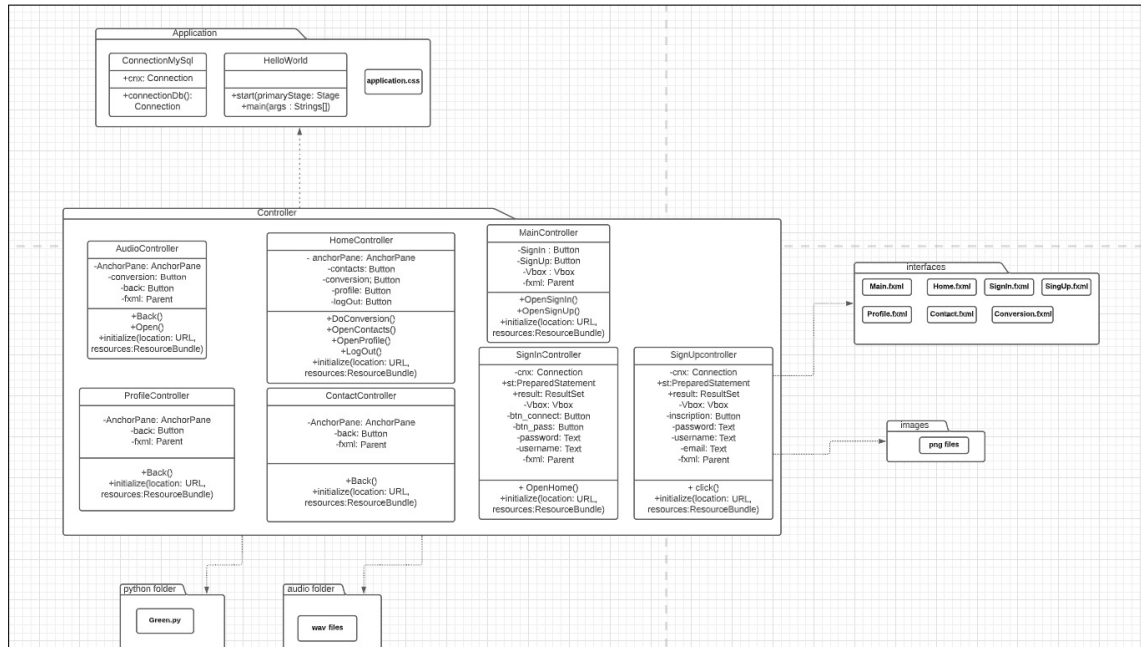


FIGURE 3.2 – Diagramme de classe

Le code source de l'application suit le modèle MVC comme mentionné précédemment. La partie modèle est représentée par le paquet "Application". Il est celui qui lance l'application et charge les vues et leurs contrôleurs. Le paquet "Controller" est également présent, il communique et interagit avec le paquet "Interfaces", qui contient les vues au format fxml. D'autres ressources externes sont également présentes, telles que le script Python et les sources d'images et de fichiers audio qui sont chargés dans l'application.

Chapitre 4

Mise en œuvre de la solution

4.1 Environnement de développement

4.1.1 La technologie Java

On a décidé d'utiliser la technologie JAVA pour le backend de l'application, à l'exception du script de spectrogramme. Les raisons pour lesquelles nous avons choisi ce langage sont diverses, telles que sa notoriété, notre affinité commune pour ce langage et les conseils de notre tuteur Pascal Makris.

4.1.2 L'architecture MVC

Nous implémentons l'application sous la forme d'un modèle MVC (Modèle-Vue-Contrôleur). Cela se reflète dans les dossiers du projet : "Application" (Modèle), "Controller" (Contrôleur) et "Interfaces" (Vue). Cette architecture permet à différentes personnes de mieux comprendre le fonctionnement de l'application. Cela facilite également la reprise et la modification/amélioration du projet par d'autres développeurs.

4.1.3 Librairies utilisées

Nous utilisons également la bibliothèque javax de Java pour créer des interfaces. Ces interfaces sont des éléments visuels destinés à être utilisés par l'utilisateur pour interagir avec l'application. Ces bibliothèques sont automatiquement importées grâce à l'utilisation de Maven dans le projet. Du côté de Python, nous utilisons NumPy pour transformer les données et Pyplot pour afficher ces résultats.

4.1.4 Les outils de développement

Différents outils de développement ont été utilisés en fonction des langages. Pycharm a été utilisé pour les scripts Python et Eclipse pour le Java. SceneBuilder a également été utilisé pour créer rapidement des vues au format fxml (javax). MySQL, Apache et PhpMyAdmin ont également été nécessaires pour implémenter et utiliser une base de données dans l'application. L'ensemble des travaux sont mis en commun grâce à GitHub.

4.2 Tests et exemples d'exécution

Les données utilisées dans les tests suivants sont des notes de musique fournies par M. Makris, enregistrées à l'aide d'un véritable instrument de musique.

4.2.1 Tests des algorithmes

Nous avons réussi à obtenir quatre algorithmes différents qui produisent un spectrogramme. Nous avons alors utilisé le même signal de départ, qui représente une gamme couvrant plusieurs notes de musique. Voici les résultats obtenus :

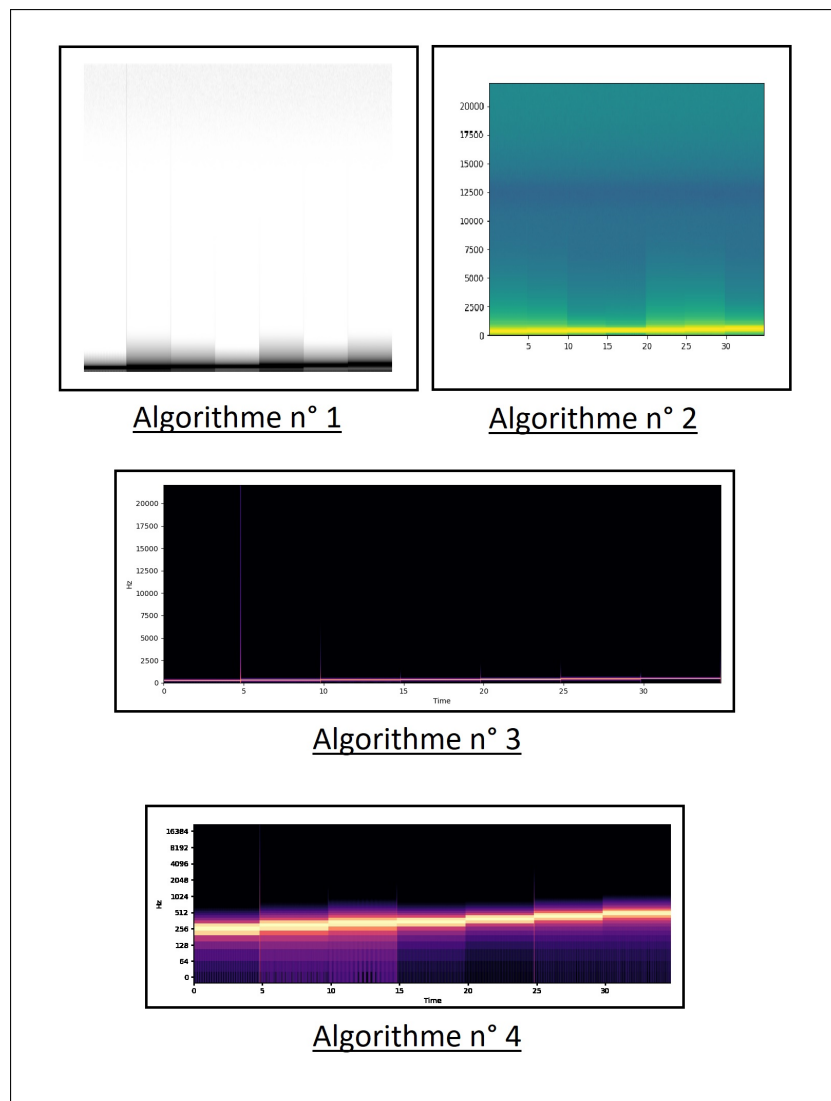


FIGURE 4.1 – Tests de différents algorithmes pour un même signal

Avec ces informations, nous avons écarté l'algorithme n°1, car la nuance de blanc et de noir pourrait poser un problème dans la distinction de l'intensité du signal. L'algorithme n°3 a été exclu de nos choix, car la représentation semblait trop différente des autres et donc potentiellement incorrecte. Enfin, nous avons préféré opter pour l'algorithme n°2 car les couleurs étaient plus agréables à l'œil nu et donc la distinction et la lecture des fréquences plus simples.

Cependant, dans l'exemple donné dans la figure 4.1, il reste difficile de déterminer une distinction entre les différentes notes de musique. Nous allons donc ajuster l'échelle de l'image par la suite en fixant une fréquence maximale de 2000 Hz.

4.2.2 Tests des données

Pour la seconde partie de test, nous avons dû tester nos données (notes de musiques). Nous sommes partis de l'algorithme retenu (algorithme n°2) pour créer le spectrogramme de nos données. Ces tests permettent de valider notre algorithme et nos

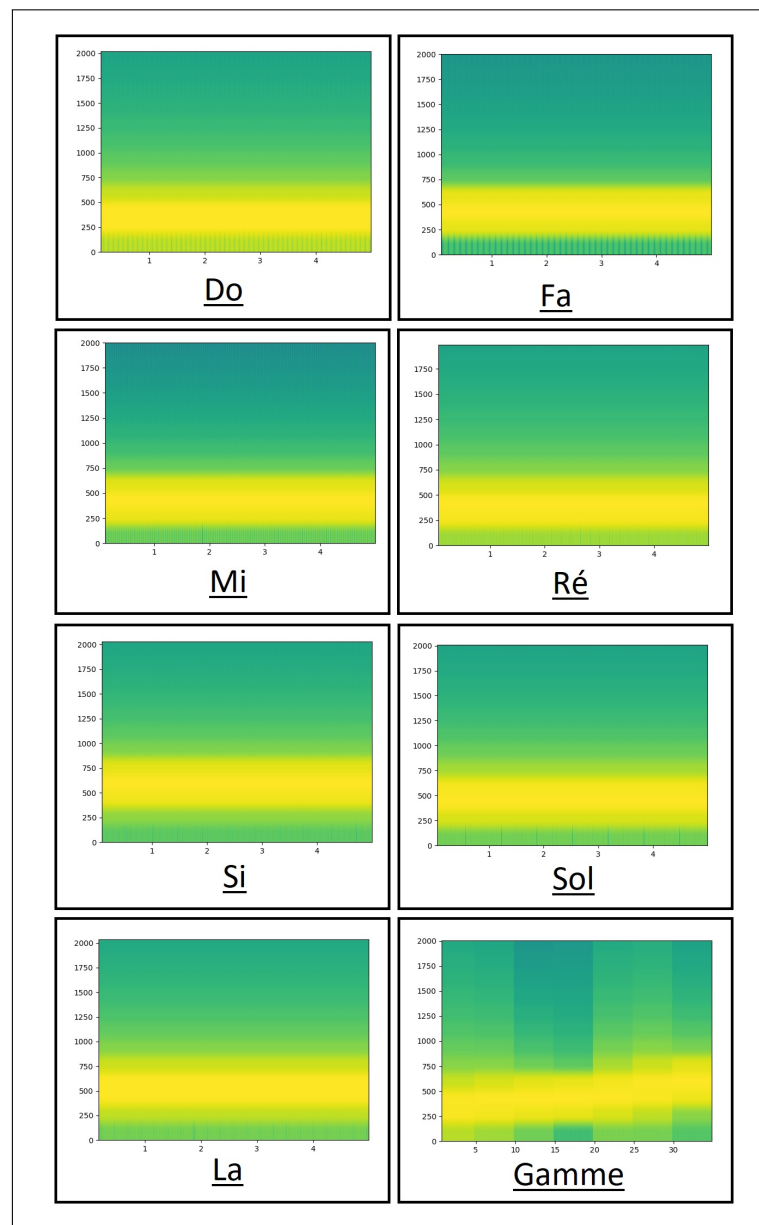


FIGURE 4.2 – Tests des données selon l'algorithme n°2

données. En effet, on remarque la différence de fréquence entre nos notes de musique. De plus, dans le cas de la "gamme", nous pouvons également identifier les différents intervalles de temps de chaque note de musique.

Chapitre 5

Conclusion et perspectives

La problématique de départ du projet était l'ajout de fonctionnalités dans une application d'analyse de vocalises, mais cette problématique a évolué. Nous avons finalement travaillé sur un nouveau sujet, qui était la représentation graphique d'un bruit xiphoïdien sous la forme d'un spectrogramme. Le groupe de projet était composé de 3 personnes de nationalités différentes, ce qui a renforcé l'importance des techniques de gestion de projet et de développement logiciel utilisées. Cela était d'autant plus important que le projet a pour objectif futur d'être repris par d'autres groupes de travail. Finalement, nous avons livré une application Java et son interface fxml qui communiquent avec un script Python afin de répondre au mieux à la problématique. Un rapport a également été livré, contenant les différentes analyses des besoins et du projet.

Une amélioration possible au projet serait de pouvoir comparer 2 signaux via l'application selon la fréquence et le temps.