

MUSIC SHEET WRITER CAHIER DES CHARGES

Jonathan Rcaud, Antoine Simon, Jeremy Harrault, Julien Blondeel, Simon Daguene, Florian
Corradin

MUSIC SHEET WRITER

Objectifs du document

Résumé

Music Sheet Writer est un logiciel d'édition de partition pour musiciens néophytes. La principale fonctionnalité du logiciel est de pouvoir générer une partition à partir d'un piano ou d'une guitare branchés sur l'ordinateur. Il est accompagné d'applications mobiles ainsi que d'un site internet.

Ce projet est soumis à différentes contraintes fonctionnelles, comme la portabilité du logiciel sur Windows et Mac, techniques, comme la sécurisation des données utilisateur stockées sur le site internet, économiques et matérielles comme le prix de certaines licences d'API que nous utiliserons ou le besoin de périphériques informatiques.

Music Sheet Writer est donc composé de trois types de livrables : le logiciel à destination des ordinateurs personnels, le site internet et les applications mobiles.

Le logiciel prévoit donc des fonctionnalités d'édition de partition, mais aussi de lecture et de gestion de projet. Nous définissons une partition comme un projet au sein de Music Sheet Writer.

Le site internet, quant à lui, nous servira à faire la promotion du logiciel et des applications mobiles. Il intégrera aussi un accès à une communauté parmi laquelle l'utilisateur pourra consulter et partager ses partitions. Il y aura aussi un espace dédié à l'utilisateur au sein duquel il pourra gérer ses informations et partitions.

Les applications mobiles, disponibles sur Android, iOS et Windows Phone, seront une alternative au site internet et permettront à l'utilisateur d'avoir cet accès à la communauté Music Sheet Writer où qu'il soit.

Nous travaillerons donc sur un environnement Windows pour le développement du logiciel Windows, des applications mobiles Android et Windows Phone et du site internet. Pour le logiciel Mac OS et l'application mobile iOS, l'environnement de travail sera Mac OS. Le logiciel sera développé en C++ et le site internet utilisera les technologies classiques Apache/PHP pour le back-end et JavaScript/CSS3/HTML pour le front-end. Les applications mobiles seront quant à elles développées dans leurs langages natifs.

Afin d'avoir une application de qualité, nous adopterons une méthode de développement dirigée par les tests. Pour cela, nous nous servirons de plusieurs outils spécifiques aux plateformes de développements et aux langages que nous utiliserons.

Enfin, pour la bonne réalisation du projet, différentes règles, outils et méthodologies s'appliqueront aux membres du groupe. Cela afin de faire face aux différentes difficultés que nous pourrions rencontrer comme l'éloignement géographique des membres du groupe. Les outils de planification, de gestion de projet et de communication ont été choisis avec cette difficulté en tête. Le travail à effectuer sera aussi attribué en fonction des compétences de chacun par rapport aux domaines à aborder.

Glossaire

– A –

API (*Application Programming Interface*) : une API est une interface pour langages de programmation, matérialisées par des primitives, permettant à une application d'accéder à des programmes système pour, par exemple, communiquer ou extraire des données.

– B –

BPM (*Battement par minute*) : le BPM est une unité de mesure utilisée pour exprimer le tempo de la musique, quantifiée par le nombre de battements se produisant en une minute.

– C –

Chiffage (solfège) : le chiffage d'une mesure se définit grâce à deux nombres disposés l'un au-dessus de l'autre. Le premier, celui du dessus, indique le nombre de temps d'une mesure. Le second indique l'unité de temps. Si l'unité de temps est de 1, alors la ronde durera 1 temps, si l'unité de temps est de 2, alors la blanche (une demi-ronde) durera 1 temps, si l'unité de temps est de 4, alors la noire (un quart de ronde) durera 1 temps, etc.

Clef (solfège) : une clef (ou clé) est un signe graphique placé au début de la portée qui indique la hauteur des notes associées à chaque ligne.

– I –

IDE (*Integrated Development Environment*) : un IDE est un ensemble d'outils ayant pour but d'augmenter la productivité des programmeurs qui développent des logiciels. Il comporte un éditeur de texte destiné à la programmation, des fonctions qui permettent, par pression sur un bouton, de démarrer le compilateur ou l'éditeur de liens ainsi qu'un débogueur en ligne, qui permet d'exécuter ligne par ligne le programme en cours de construction. Certains environnements sont dédiés à un langage de programmation en particulier.

– M –

Mesure (solfège) : la mesure est une segmentation de la durée du discours musical. En d'autres termes, la mesure est la division d'un morceau de musique en parties d'égales durées. Cette division s'indique au moyen de barres verticales sur la portée, appelées barres de mesure. Le type d'une mesure est indiqué par une clef et un chiffage.

MIDI (*Musical Instrument Digital Interface*) : le MIDI est un protocole de communication et de commande permettant l'échange de données entre instruments de musique électronique. Le format de fichier MIDI a été défini en 1988, pour stocker les commandes MIDI sur disquettes en y ajoutant des informations temporelles.

– N –

Nuance (solfège) : une nuance est un signe noté sur une partition qui indique l'intensité relative d'une note, d'une phrase, ou encore d'un passage entier d'une œuvre musicale.

– O –

Ornement (solfège) : un ornement est symbole désignant des notes secondaire dont la fonction est d'embellir la ligne mélodique principale.

– P –

Portée (solfège) : la portée est un ensemble de cinq lignes horizontales permettant de représenter les hauteurs. Elle est destinée à recevoir les figures de notes et de silences, les clefs et le chiffage des mesures.

– T –

Temps (solfège) : le temps est l'unité de mesure de la durée musicale.

Description du document

Titre	Music Sheet Writer : Cahier des charges
Date	18/02/2015
Auteur	Jeremy Harrault
Responsable	Jonathan Racaud
E-mail	musicsheetwriter_2017@labeip.epitech.eu
Sujet	Cahier des charges fonctionnel
Version du modèle	1.0

Tableau des révisions

Date	Auteur	Section(s)	Commentaire
18/02/15	Jeremy Harrault	Toutes	Création du document
25/02/15	Florian Corradin	Contraintes	Ajout partie
25/02/15	Jeremy Harrault	Description des différentes parties du programme à réaliser	Ajout partie
25/02/15	Antoine Simon	Environnement de réalisation	Ajout partie
27/02/15	Antoine Simon	Environnement de réalisation	Finition de la partie Site Internet
28/02/15	Antoine Simon	Environnement de réalisation	Ajout SeeScore
28/02/15	Jeremy Harrault	Description des différentes parties du programme à réaliser	Ajout des diagrammes des cas d'utilisation d'UML + modification des fonctionnalités de la partie « Application Mobile »
02/03/15	Jonathan Racaud	Description des différentes parties du programme à réaliser	Mise à jour des diagrammes de use case et ajout des diagrammes de use case pour la partie mobile.
02/03/15	Jonathan Racaud	Environnement de réalisation	Ajout des logos de sublime text, php, javascript + Modification du titre numéro 4 'Musique SDK' → SeeScore
02/03/15	Julien Blondeel	Organisation projet	Ajout partie
03/03/15	Jeremy Harrault	Contraintes	Suppression des « contraintes ergonomiques »
03/03/15	Jeremy Harrault	Description des différentes parties du programme à réaliser + Glossaire	Mise en page : 1 partie/sous-partie par fonctionnalité + Ajout du vocabulaire sur le solfège
04/03/15	Julien Blondeel	Organisation projet	Ajout des parties ressources et méthodologie.
05/03/15	Simon Daguenet	Description des tests	Ajout partie
05/03/15	Tous	Toutes	Correction
06/03/15	Jonathan Racaud	Description des différentes parties du projet	Modification de l'uml
06/03/15	Jonathan Racaud	Organisation projet - Environnement de réalisation	Modification de la partie sur la méthodologie. Modification de la partie sur Windows Phone
07/03/15	Jonathan Racaud	Résumé - Toutes	Ajout résumé – Corrections
08/03/15	Jeremy Harrault	Environnement de réalisation - Toutes	Fusion PortAudio/PortMidi + Ajout description d'Android Studio + Correction

Sommaire

1. Rappel de l'EIP	1
1.1. Objectif de l'EIP et Epitech	1
1.2. Principe de base du système futur.....	1
2. Contraintes	2
2.1. Contraintes fonctionnelles	2
2.2. Contraintes techniques	2
2.3. Contraintes économiques et matérielles	2
3. Description des différentes parties du projet à réaliser	3
3.1. Le logiciel	3
3.1.1. Description générale	3
3.1.2. Le gestionnaire de projets	3
3.1.2.1. Créer un projet	4
3.1.2.2. Ouvrir un projet	4
3.1.2.3. Sélectionner un projet	4
3.1.2.4. Sauvegarder un projet	4
3.1.2.5. Exporter un projet	4
3.1.2.6. Fermer un projet	4
3.1.3. L'éditeur de partition	5
3.1.3.1. Modifier le type de mesure	5
3.1.3.2. Placer une note ou un silence	5
3.1.3.3. Supprimer une note ou un silence	5
3.1.3.4. Placer une nuance ou un ornement	5
3.1.3.5. Supprimer une nuance ou un ornement	5
3.1.3.6. Modifier le tempo.....	5
3.1.3.7. Générer une partition depuis un instrument branché à l'ordinateur	6
3.1.3.8. Placer les curseurs de début et de fin d'édition	6
3.1.3.9. Utiliser le métronome	6
3.1.4. Le lecteur de partition	6
3.1.4.1. Lire une partition	6
3.1.4.2. Placer le curseur de lecture	6
3.1.4.3. Mettre en pause la lecture	7
3.1.4.4. Arrêt de la lecture.....	7
3.2. Le site internet et les applications mobiles	7
3.2.1. Description générale	7
3.2.2. Espace vitrine et contact (site internet uniquement)	7
3.2.2.1. Présentation et achat de Music Sheet Writer	7
3.2.2.2. Contacter l'équipe	7
3.2.3. Accès à la communauté	8
3.2.3.1. Rechercher un membre	8
3.2.3.2. Consulter le profil d'un membre	8
3.2.3.3. Rechercher une partition.....	8
3.2.3.4. Consulter une partition	8
3.2.4. La gestion de compte utilisateur	9
3.2.4.1. Créer un compte.....	9
3.2.4.2. Se connecter à un compte.....	9

3.2.4.3.	Se déconnecter d'un compte	9
3.2.4.4.	Gérer les informations de compte	9
3.2.4.5.	Gestion des partitions partagées.....	10
3.2.4.6.	Gestion des abonnements.....	10
3.2.4.7.	Gestion des partitions favorites	10
4.	Présentation de l'environnement de réalisation.....	11
4.1.	Le logiciel PC.....	11
4.1.1.	Langage de programmation utilisé	11
4.1.2.	Qt	11
4.1.3.	SeeScore.....	11
4.1.4.	PortAudio et PortMidi.....	11
4.2.	Le site internet	11
4.2.1.	Langages de programmation utilisés	11
4.2.1.1.	PHP 5	11
4.2.1.2.	JavaScript.....	11
4.2.2.	Sublime Text	12
4.2.3.	Apache	12
4.2.4.	MySQL	12
4.3.	Mobile	12
4.3.1.	iOS.....	12
4.3.1.1.	Présentation	12
4.3.1.2.	Xcode	12
4.3.2.	Android	12
4.3.2.1.	Présentation	12
4.3.2.2.	Android Studio.....	12
4.3.3.	Windows Phone	13
4.3.3.1.	Présentation	13
4.3.3.2.	Visual Studio	13
4.3.3.3.	Blend.....	13
5.	Description des tests	14
5.1.	Test.....	14
5.1.1.	Définition	14
5.1.2.	Test unitaire	14
5.2.	Outils de développement.....	14
5.2.1.	Outil pour le logiciel.....	14
5.2.2.	Les outils des applications mobiles.....	14
5.2.3.	Outil pour le site internet	14
6.	Organisation projet.....	15
6.1.	Les ressources	15
6.1.1.	Les ressources humaines	15
6.1.2.	Les ressources matérielles	15
6.2.	Le planning.....	15
6.3.	La méthodologie.....	15
6.3.1.	Les outils	15
6.3.2.	La méthode	15
6.3.3.	Les réunions	16

1. Rappel de l'EIP

1.1. Objectif de l'EIP et Epitech

EPITECH est l'école de l'expertise informatique, transformant une passion en véritable expertise. L'apprentissage à EPITECH est fondé sur une pédagogie par projets, individuels ou en groupe, validant un certain nombre de connaissances et de notions à assimiler. Tout au long de leur cursus, les étudiants se familiarisent avec le milieu professionnel, notamment grâce aux stages en première, troisième et cinquième année d'une période de quatre à six mois. L'école forme les étudiants à s'adapter à des situations inhabituelles avec la mise en place de rush (projets à réaliser sur un week-end, sur des sujets et notions dont les élèves n'ont aucune connaissance) ou le départ à l'international pendant leur quatrième année ; année durant laquelle l'étudiant va devoir faire preuve d'autonomie et de capacité d'adaptation.

Les Epitech Innovative Projects sont des projets à réaliser sur le cycle master du cursus Epitech. Ils sont conçus à la manière d'un véritable projet entrepreneurial, dans toutes ses composantes : business, techno, design & communication. Un EIP est appelé à devenir une start-up viable. Le but de l'EIP est donc de faire découvrir aux étudiants le monde de l'entrepreneuriat en leur demandant de mettre un place un projet et de le réaliser en faisant face à des difficultés qu'ils n'avaient jusqu'alors pas rencontrées. Le principal obstacle est la gestion de groupe composé de membres dispersés dans des pays différents, faisant face alors aux problèmes de gestion du temps et des zones horaires pour leur quatrième année. Les problématiques de communication et de vente du produit sont aussi abordées.

1.2. Principe de base du système futur

Music Sheet Writer est un logiciel d'édition de partition destiné aux musiciens néophytes qui n'ont pas forcément les connaissances théoriques du solfège pour écrire leurs compositions. Il se présente donc comme tout logiciel d'édition de partition existant, mais apporte une fonctionnalité majeure : la génération d'une partition depuis un piano ou une guitare branchés à l'aide d'un câble JACK ou d'une interface audio USB.

Le mot d'ordre de Music Sheet Writer est d'être simple d'utilisation. En effet, en ajoutant cette fonctionnalité, nous simplifions la phase d'écriture lors de la composition d'une musique. Laisant l'utilisateur se concentrer sur la musique avant son écriture.

Bien entendu, les musiciens aguerris ne seront pas en reste puisque Music Sheet Writer incorporera les outils qui leur permettront d'écrire leurs musiques de manière très précise.

Music Sheet Writer s'accompagne aussi d'applications mobiles disponibles sur Android, iOS et Windows Phone, ainsi que d'un site internet.

2. Contraintes

Dans cette partie, nous listons toutes les contraintes liées au développement des différents livrables du projet Music Sheet Writer.

2.1. Contraintes fonctionnelles

- Le logiciel doit être porté sur Windows et Mac.
- Les applications mobiles doivent être développées en natif sur Android, iOS, Windows Phone.
- Les partitions générées doivent être conformes aux règles d'écriture musicale.
- Les partitions générées doivent être compatibles avec le standard MusicXML.

2.2. Contraintes techniques

- Protection des données utilisateurs stockées sur le serveur.
- Sécurisation des transmissions de données.

2.3. Contraintes économiques et matérielles

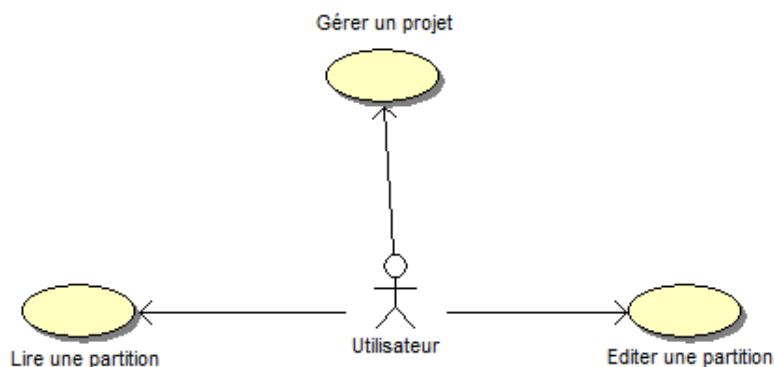
- Carte son avec entrée MIDI
- Certificat Apple
- Certificat SSL
- Licences des différents SDK et librairies
- Serveur web
- Clavier MIDI

3. Description des différentes parties du projet à réaliser

Le développement de Music Sheet Writer se décompose en différents livrables : un logiciel à destination de Windows et Mac OS, des applications mobiles disponibles sur Android, Windows Phone et iOS ainsi qu'un site internet.

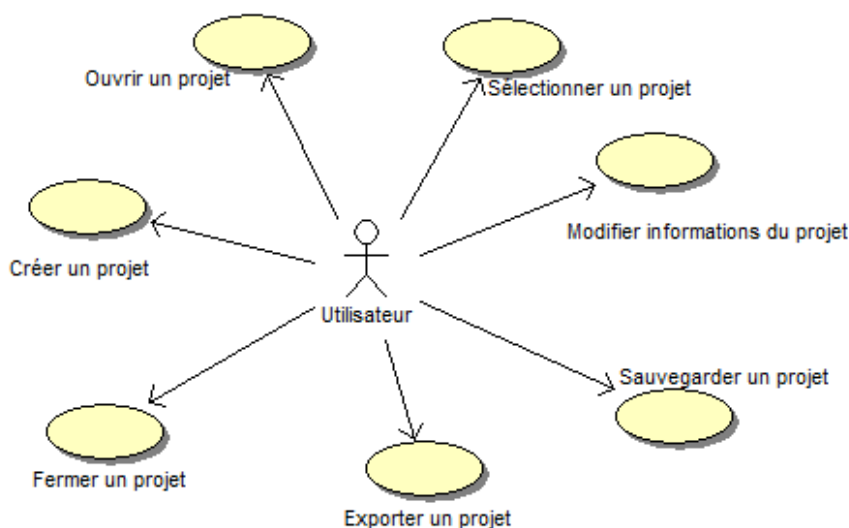
3.1. Le logiciel

3.1.1. Description générale



L'utilisateur pourra gérer ses projets, un projet sous Music Sheet Writer étant une partition de musique. En effet, il pourra créer une partition vierge, en ouvrir une déjà existante, la modifier et la sauvegarder. De plus, l'utilisateur pourra éditer sa partition à l'aide du clavier et de la souris mais également en jouant depuis une guitare ou un piano branchés à son ordinateur. Enfin, il pourra la lire grâce à un lecteur audio intégré au logiciel.

3.1.2. Le gestionnaire de projets



3.1.2.1. Créer un projet

L'utilisateur pourra créer un nouveau projet. Les projets créés seront automatiquement ouverts et affichés dans l'éditeur. Les mesures composant la partition seront vides.

Lors de la création d'un projet, l'utilisateur devra renseigner :

- Le nom du projet
- Le titre de la partition
- Le nom du compositeur
- Le nom de l'auteur
- Le type de mesure
- Le tempo de la musique

L'utilisateur pourra modifier à tout moment ces informations.

3.1.2.2. Ouvrir un projet

L'utilisateur pourra ouvrir un projet, s'ajoutant alors à la liste de ceux déjà ouverts. Cette liste sera apparente dans la vue du gestionnaire de projets. Un projet ne pourra pas être ouvert plusieurs fois.

Le logiciel pourra ouvrir les fichiers dans l'un des formats suivants :

- MusicXML
- MIDI

3.1.2.3. Sélectionner un projet

L'utilisateur pourra sélectionner un projet depuis la vue du gestionnaire de projets afin de l'afficher dans l'éditeur.

3.1.2.4. Sauvegarder un projet

L'utilisateur pourra sauvegarder son projet au format MusicXML. Lors de la première sauvegarde, il devra choisir l'emplacement du projet au sein de son ordinateur. Les prochaines sauvegardes écraseront automatiquement l'ancienne version.

3.1.2.5. Exporter un projet

L'utilisateur pourra exporter son projet dans différents formats. Il devra pour cela choisir un emplacement au sein de son ordinateur ainsi que le format d'export.

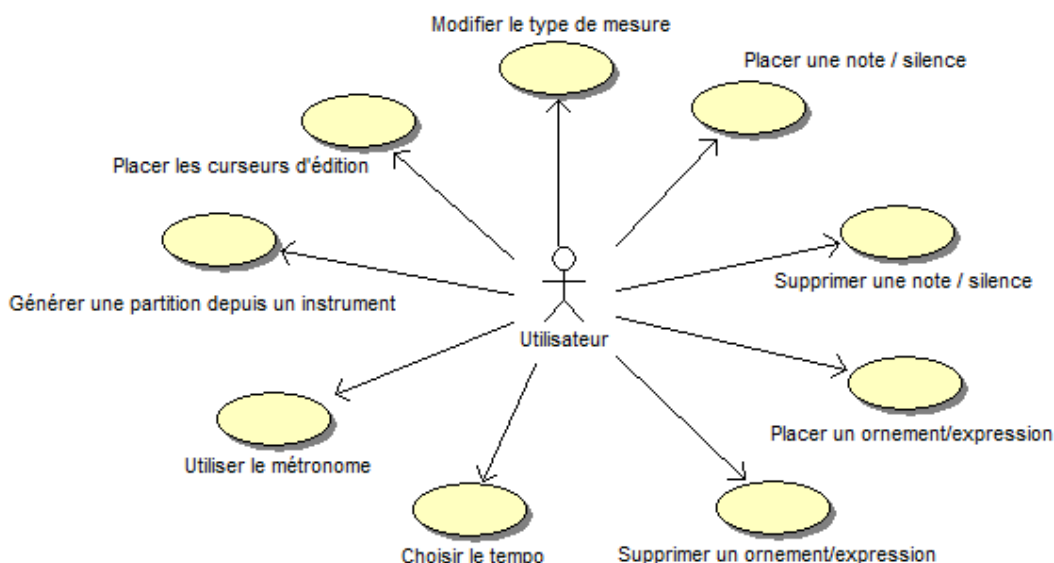
Les formats gérés sont :

- MusicXML
- PDF
- MIDI
- MP3

3.1.2.6. Fermer un projet

L'utilisateur pourra fermer un projet qu'il aura préalablement ouvert. Une fois fermé, le projet ne sera plus visible dans la vue du gestionnaire de projets.

3.1.3. L'éditeur de partition



3.1.3.1. Modifier le type de mesure

L'utilisateur pourra modifier le type de mesure. Il pourra attribuer une clef et un chiffage.

Pour la clef, l'utilisateur aura le choix entre :

- La clef de Sol
- La clef de Fa
- La clef d'Ut

Pour le chiffage, il devra sélectionner l'unité de temps et le nombre de temps d'une mesure.

3.1.3.2. Placer une note ou un silence

L'utilisateur pourra placer une note ou un silence sur la partition. Pour cela, il n'aura qu'à sélectionner ceux qu'il souhaite ajouter et cliquer sur la partition à l'emplacement voulu. L'élément placé sera alors visible sur la partition.

Un message d'erreur sera affiché si l'utilisateur tente d'ajouter une note ou un silence plus long que le nombre de temps disponibles dans la mesure.

3.1.3.3. Supprimer une note ou un silence

L'utilisateur pourra supprimer une note ou un silence de la partition. Ces derniers ne seront alors plus visibles sur la partition.

3.1.3.4. Placer une nuance ou un ornement

L'utilisateur pourra placer une nuance ou un ornement sur la partition. Pour cela, il n'aura qu'à sélectionner ceux qu'il souhaite ajouter et cliquer à l'emplacement voulu. Pour ceux affectant un ensemble de notes (comme les crescendo), il sera possible de modifier leurs longueurs et leurs intensités. L'élément ajouté sera alors visible sur la partition.

3.1.3.5. Supprimer une nuance ou un ornement

L'utilisateur pourra supprimer une nuance ou un ornement de la partition. Ils ne seront alors plus visibles sur la partition.

3.1.3.6. Modifier le tempo

L'utilisateur pourra modifier le tempo de la partition. Il devra choisir une valeur exprimée en BPM. L'utilisateur pourra avoir un aperçu du tempo sélectionné avant de le valider.

3.1.3.7. Générer une partition depuis un instrument branché à l'ordinateur

L'utilisateur pourra générer sa partition directement à partir d'un piano ou d'une guitare branchés à une entrée MIDI, microphone ou USB de l'ordinateur.

Avant le début de l'écoute des entrées des périphériques, le logiciel jouera une mesure pour que l'utilisateur ait un rappel du tempo.

Une fois que l'utilisateur aura fini de jouer son morceau, le logiciel analysera les données reçues afin de reconnaître les notes jouées. Elles seront par la suite ajoutées dans la partition et affichées dans l'éditeur.

3.1.3.8. Placer les curseurs de début et de fin d'édition

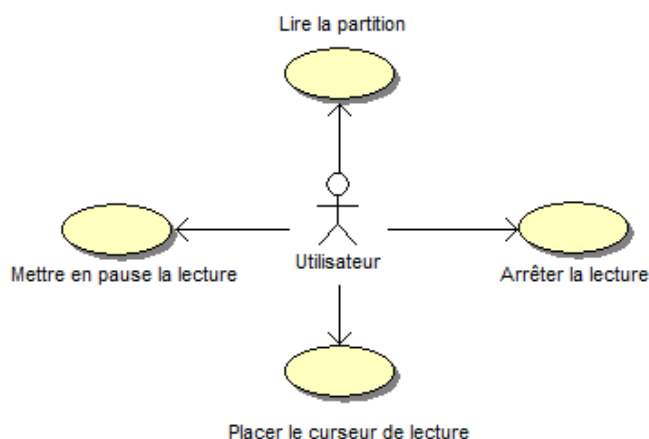
L'utilisateur pourra placer deux curseurs sur la partition qui indiqueront le début et la fin de l'édition. Ces curseurs seront utilisés pour délimiter une zone d'édition pour la modification de la partition à l'aide d'un instrument.

Ils seront aussi dédiés à la délimitation de la zone de lecture des partitions.

3.1.3.9. Utiliser le métronome

L'utilisateur pourra utiliser un métronome pendant qu'il joue son morceau. Le métronome donnera visuellement et/ou auditivement l'allure de la musique, celle-ci étant dépendante du tempo.

3.1.4. Le lecteur de partition



3.1.4.1. Lire une partition

L'utilisateur pourra lire la partition ouverte dans l'éditeur. Un curseur de lecture se déplacera au fur et à mesure que la partition est lue. La vitesse de lecture correspondra au tempo sélectionné par l'utilisateur. Les sons utilisés seront au format MIDI.

Lors de la lecture de la partition, l'utilisateur ne pourra pas modifier la partition.

3.1.4.2. Placer le curseur de lecture

L'utilisateur pourra placer le curseur de lecture sur la partition. Celui-ci marquera l'endroit à partir d'où la lecture commencera.

Si l'utilisateur ne désigne pas d'endroit spécifique sur la partition, la lecture commence alors à l'emplacement du curseur de début d'édition.

3.1.4.3. Mettre en pause la lecture

L'utilisateur pourra mettre en pause la lecture de la partition. Le curseur de lecture s'immobilisera sur la partition à l'endroit où la lecture s'est arrêtée.

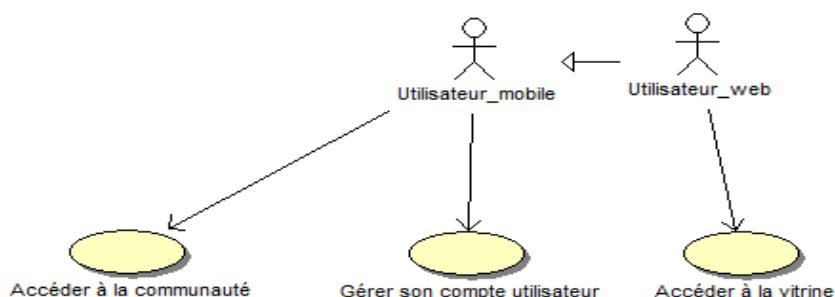
L'utilisateur pourra alors éditer la partition.

3.1.4.4. Arrêt de la lecture

L'utilisateur pourra arrêter la lecture de la partition. Le curseur de lecture retournera automatique au début de la partition ou, là où le lecteur l'avait placé avant de démarrer la lecture.

L'utilisateur pourra alors éditer la partition.

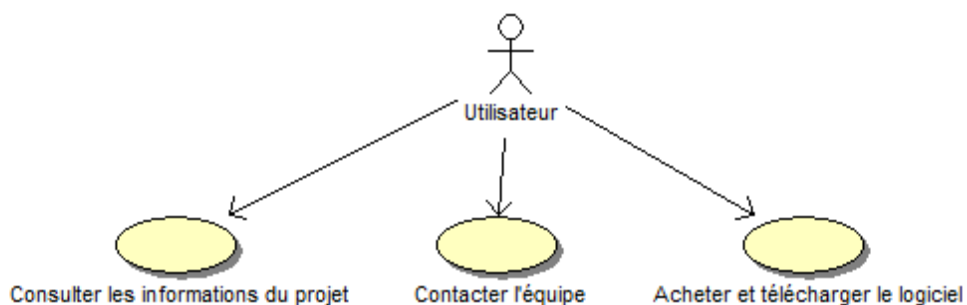
3.2. Le site internet et les applications mobiles



3.2.1. Description générale

L'utilisateur pourra se renseigner sur Music Sheet Writer et l'équipe de développement. En effet, une présentation des produits proposés ainsi qu'une plateforme de contact seront fournies. De plus, l'utilisateur pourra découvrir la communauté en recherchant des utilisateurs et en écoutant leurs compositions. Enfin, il pourra lui-même créer un compte afin de partager ses créations et d'être informé sur les activités des autres utilisateurs.

3.2.2. Espace vitrine et contact (site internet uniquement)



3.2.2.1. Présentation et achat de Music Sheet Writer

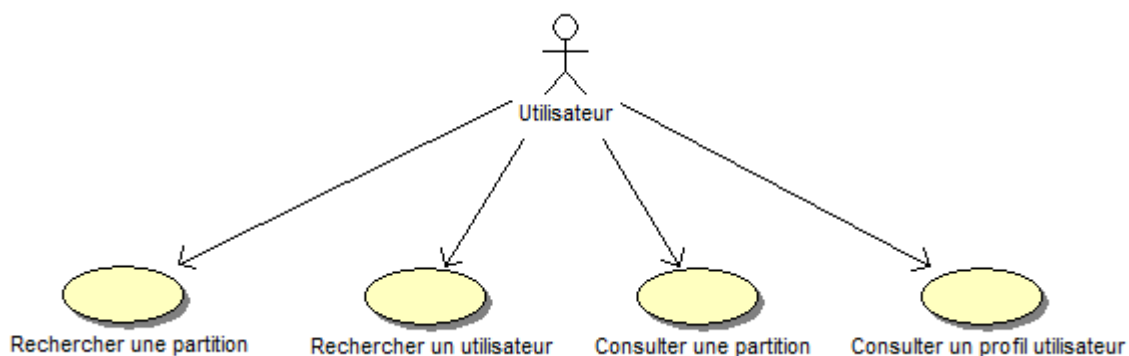
L'utilisateur pourra avoir des informations sur les produits et services proposés par Music Sheet Writer. Une présentation de l'équipe de développement sera également disponible.

Il pourra directement accéder à la plateforme d'achat et de téléchargement du logiciel.

3.2.2.2. Contacter l'équipe

L'utilisateur pourra contacter l'équipe Music Sheet Writer directement sur le site internet. Il n'aura qu'à écrire l'objet et le corps du message à envoyer. L'équipe le recevra alors par e-mail.

3.2.3. Accès à la communauté



3.2.3.1. Rechercher un membre

L'utilisateur pourra rechercher un membre de la communauté Music Sheet Writer par son nom d'utilisateur. Si ce nom n'est attribué à aucun membre, une liste des utilisateurs avec un nom similaire sera affichée.

3.2.3.2. Consulter le profil d'un membre

L'utilisateur pourra consulter la page de profil d'un membre. Cette page contiendra :

- Son nom d'utilisateur
- Sa photo de profil
- Une courte description de l'utilisateur
- La liste de ses partitions partagées
- La liste de ses abonnements.
- La liste de ses abonnés.

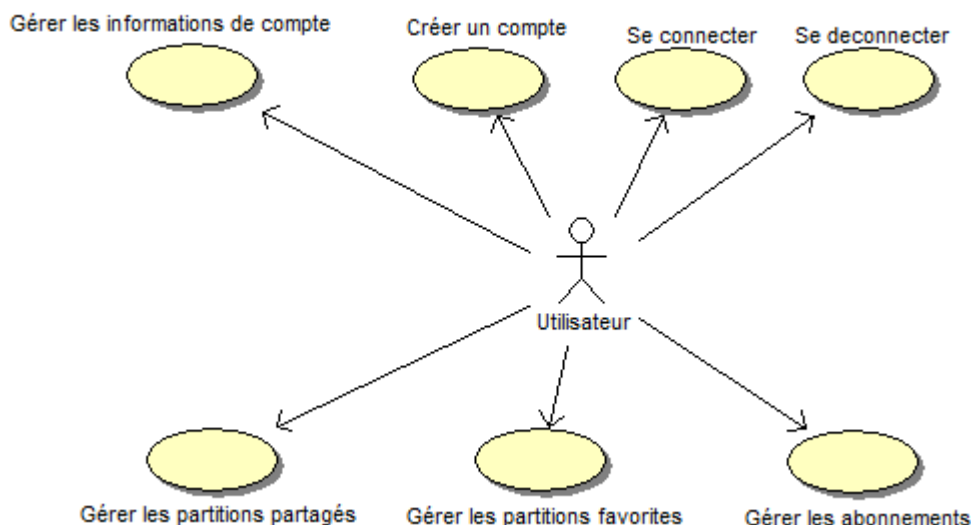
3.2.3.3. Rechercher une partition

L'utilisateur pourra rechercher une partition par son titre ou directement depuis la liste des partitions d'un utilisateur. Si le titre recherché ne présente aucun résultat, une liste de partition avec un titre similaire sera affichée. Si, au contraire, le titre recherché est attribué à plusieurs partitions, elles devront être listées et le nom de l'auteur devra être visible pour pouvoir les différencier.

3.2.3.4. Consulter une partition

L'utilisateur pourra visualiser une partition et la lire comme un fichier MIDI. Il pourra également la télécharger au format MusicXML.

3.2.4. La gestion de compte utilisateur



3.2.4.1. Créer un compte

L'utilisateur pourra créer un compte gratuitement. Il devra renseigner :

- Un nom d'utilisateur
- Un mot de passe
- Une adresse e-mail

Un message d'erreur sera affiché si des champs ne sont pas valides ou si le nom d'utilisateur ou l'adresse mail entrés sont déjà utilisés.

Un e-mail de confirmation d'inscription sera envoyé à l'adresse renseignée par l'utilisateur. Cet e-mail contiendra un lien nécessaire à l'activation du compte afin de vérifier l'adresse mail.

3.2.4.2. Se connecter à un compte

L'utilisateur pourra se connecter à un compte en utilisant :

- Son nom d'utilisateur/adresse e-mail
- Son mot de passe

L'utilisateur aura la possibilité de retrouver son mot de passe perdu. Il devra alors saisir l'adresse e-mail associé à son compte. Un message contenant un lien à usage unique et limité dans le temps, lui permettant de changer de mot de passe, lui sera donc envoyé.

3.2.4.3. Se déconnecter d'un compte

L'utilisateur pourra se déconnecter d'un compte. La déconnexion n'est possible que si l'utilisateur est connecté à un compte.

3.2.4.4. Gérer les informations de compte

L'utilisateur pourra gérer certaines informations de son compte :

- L'adresse e-mail
- Une photo de profil
- Une description utilisateur
- Le mot de passe

Une fois l'adresse e-mail changée, l'utilisateur recevra un e-mail de vérification sur son ancienne adresse et de confirmation sur la nouvelle.

Lors d'une modification du mot de passe, l'utilisateur recevra un e-mail de vérification.

3.2.4.5. Gestion des partitions partagées

L'utilisateur pourra partager ses partitions en ligne. Les partitions partagées seront alors visibles, lisibles et téléchargeables par tout le monde. La liste des partitions partagées sera affichée sur la page de profil de l'utilisateur.

L'utilisateur pourra supprimer à tout moment une partition qu'il a partagée.

3.2.4.6. Gestion des abonnements

L'utilisateur pourra s'abonner à d'autres membres. Il pourra choisir de recevoir un e-mail lorsque de nouvelles partitions sont partagées par les utilisateurs auxquels il est abonné. La liste des abonnements sera affichée sur la page de profil de l'utilisateur.

L'utilisateur pourra se désabonner à tout moment d'un membre auquel il est abonné.

3.2.4.7. Gestion des partitions favorites

L'utilisateur pourra ajouter une partition partagée par un autre membre dans sa liste de favoris. La liste de favoris sera affichée sur la page de profil de l'utilisateur.

L'utilisateur pourra retirer une partition de ses favoris. Toutes les partitions qui ne seront plus partagées par leurs propriétaires seront retirées de la liste des favoris de tous les utilisateurs.

4. Présentation de l'environnement de réalisation

Dans cette partie, nous décrivons les environnements de développement Windows et Mac OS que nous utiliserons pour la réalisation du projet.

4.1. Le logiciel PC

Le développement du logiciel à destination des ordinateurs personnels est conditionné par l'utilisation de différentes librairies et SDK pour la manipulation du format MusicXML et de la récupération des entrées utilisateurs.

4.1.1. Langage de programmation utilisé

Nous avons choisi d'utiliser le C++ pour sa compatibilité Windows et Mac OS, mais aussi pour bénéficier des différentes librairies audio, MIDI et graphiques qui sont majoritairement développées dans ce langage.

4.1.2. Qt

Qt est un Framework C++ multiplateforme fondamentalement conçu pour la création de fenêtres. De plus, de nombreux modules sont disponibles pour la gestion de certains langages tel que le XML, JavaScript ou la mise en place d'architectures réseaux, la création de jeux vidéo, etc.

Nous avons opté pour Qt afin de répondre à la contrainte d'un logiciel multiplateforme. Nous aurons un seul code pour tous les systèmes d'exploitation ainsi qu'une interface utilisateur unifiée.

La communauté autour de ce Framework est un facteur important dans ce choix. Les ressources non officielles disponibles sont alors légions. De plus, la documentation officielle est de très bonne qualité.

4.1.3. SeeScore

SeeScore est une API multiplateforme gérant le format MusicXML. Elle nous permettra d'afficher et de lire les partitions.

Cette API est soumise à des licences donnant accès à différentes fonctionnalités. Il existe une licence gratuite ainsi qu'une licence professionnelle payante.

4.1.4. PortAudio et PortMidi

PortAudio et PortMidi sont des libraires audio multiplateformes open-sources qui permettent de gérer les entrées et sorties audio. PortAudio permet la récupération des sons sur les entrées Jack et USB tandis que PortMidi s'occupe de l'entrée MIDI. Nous les avons choisis pour leur compatibilité avec les systèmes Windows et Mac OS.

4.2. Le site internet

4.2.1. Langages de programmation utilisés

4.2.1.1. PHP 5

Nous utiliserons PHP pour le back-end du site internet. Sa facilité d'utilisation, la disponibilité de multiples Framework ainsi que notre affinité avec le langage sont les points décisifs de ce choix.

4.2.1.2. JavaScript

Nous utiliserons JavaScript pour le front-end du site internet. Nous avons choisi JavaScript pour sa capacité à rendre dynamique le contenu des sites internet. De plus, JavaScript est la technologie la plus utilisée dans ce domaine. Disposant de nombreux Framework et d'une communauté très active.

4.2.2. Sublime Text

Sublime Text est un éditeur léger et simple d'utilisation bénéficiant d'une forte communauté agréant l'éditeur de plugin pour faciliter le développement dans certains langages.

Nous avons choisi de l'utiliser en raison de l'affinité que nous avons avec ce logiciel, de sa légèreté, mais aussi pour l'ergonomie que ce dernier nous offre en matière d'édition de code.

4.2.3. Apache

Ayant choisi PHP comme langage pour la réalisation du back-end du site internet, nous utiliserons un serveur Apache pour l'hébergement de ce dernier. De plus, il s'agit d'un serveur que nous connaissons.

4.2.4. MySQL

MySQL est un système de gestion de base de données simple et rapide d'utilisation. L'ensemble des Framework pour le développement web gère cette base de données. De plus, MySQL est gratuite. Ce sont donc ces aspects qui nous ont fait choisir cet outil.

4.3. Mobile

4.3.1. iOS

4.3.1.1. Présentation

iOS est un système d'exploitation mobile développé par la société Apple. Ce système n'est distribué que sur les produits de cette même marque, c'est-à-dire iPhone et iPad. iOS est le deuxième système d'exploitation mobile le plus utilisé sur le marché actuellement. Les applications disponibles se trouvent sur l'App Store, qui contient aujourd'hui pas moins de 1,20 million d'applications.

Développer une application iOS est donc essentiel pour une meilleure visibilité.

4.3.1.2. Xcode

Xcode est un IDE développé par la société Apple. Il est uniquement disponible sur Mac. On peut développer en différents langages tels que le C, C++, Objective-C.

Il est possible de choisir la plateforme de déploiement (Safari, Mac OS X ou iPhone/iPad). Chaque application doit être signée par un certificat avant la mise à disposition sur l'App Store. Xcode facilite ce processus en l'automatisant. De plus, Il intègre toutes les librairies nécessaires ainsi qu'un émulateur permettant de tester les applications.

4.3.2. Android

4.3.2.1. Présentation

Android est un système d'exploitation mobile développé par la société Google. Il peut être retrouvé aujourd'hui sur une large gamme d'appareils mobiles, autant dans le domaine de la téléphonie que celui des tablettes. La majorité des parts de marché mobile est tenue par Android. Le Play Store, plateforme de téléchargement des applications Android, en contient 1,43 million différentes.

4.3.2.2. Android Studio

Android studio est un IDE développé par Google, spécialement conçu pour le développement natif d'application Android en JAVA. Bien qu'Eclipse, couplé à ADT (Android Development Tools) est apparu comme la première solution du développement d'application Android, Android Studio possède de plus en plus de qualité. Il est d'ailleurs possible que Google décide d'abandonner un jour ADT pour seulement proposer gratuitement Android Studio.

4.3.3. Windows Phone

4.3.3.1. Présentation

Windows Phone est un système d'exploitation mobile développé par la société Microsoft. Il est moins répandu que ses concurrents, mais gagne en part de marché chaque année. Il est disponible le plus souvent sur les téléphones Microsoft, mais aussi de façon plus restreinte chez d'autres constructeurs.

4.3.3.2. Visual Studio

Visual Studio est un IDE développé par Microsoft. Il est prévu pour faire du développement en .Net et est donc principalement à destination de Windows. Il est le seul outil permettant le développement d'application Windows Phone. Il dispose d'outils de design d'interface utilisateur, même si nous lui préférons Blend, d'un émulateur d'appareil Windows Phone et d'un puissant éditeur de code.

4.3.3.3. Blend

Blend est un utilitaire développé par Microsoft spécialisé dans le design d'interface utilisateur pour les plateformes mobiles et les applications Modern UI. Il peut être couplé avec Visual Studio et les changements faits sur l'interface utilisateur sont directement mis à jour dans Visual Studio. De cette manière, nous serons à même de nous concentrer sur l'expérience utilisateur avec Blend et sur le fonctionnement de l'application avec Visual Studio.

5. Description des tests

5.1. Test

5.1.1. Définition

C'est une procédure permettant de vérifier le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme. On écrit un test pour confronter une réalisation à sa spécification. Le test définit un critère d'arrêt (état ou sorties à l'issue de l'exécution) et permet de statuer sur le succès ou sur l'échec d'une vérification. Il permet de vérifier que la relation d'entrée/sortie donnée par la spécification est bel et bien réalisée.

5.1.2. Test unitaire

Nous utiliserons des tests unitaires qui nous permettront de tester individuellement les composants de l'application. On pourra ainsi valider la qualité du code et les performances d'un module.

5.2. Outils de développement

5.2.1. Outil pour le logiciel

Pour effectuer ces différents tests, nous utiliserons un module de Qt : « QTest ». Celui-ci nous permettra d'effectuer l'intégralité des tests nécessaires au projet. QTest semble facile d'utilisation et assez pratique à mettre en place. La documentation de Qt fournit des exemples pour la compréhension de ce module.

5.2.2. Les outils des applications mobiles

Visual Studio permet également d'éditer des tests unitaires via la création d'un nouveau projet. Ainsi, nous serons capables d'effectuer des tests unitaires sur le même environnement de développement. De la même manière, Android Studio est aussi capable de fournir des tests en utilisant « Android Unit Tests ».

5.2.3. Outil pour le site internet

Pour faire des tests unitaires sur la partie client de notre site internet, nous utiliserons « QUnit », qui est un Framework en JavaScript. Pour les tests sur le back-end, nous pourrions utiliser « PHPUnit ».

Afin de tester la robustesse de notre site internet nous ferons des tests de montée de charge. Un test de montée de charge étant la simulation de nombreuses connections d'utilisateurs en simultanée sur le site internet. Pour cela, nous utiliserons « JMeter », un outil de stress test développé par la fondation Apache.

6. Organisation projet

Cette section décrit l'organisation projet en termes de ressources, planning et méthodologie pour la réalisation du projet Music Sheet Writer.

6.1. Les ressources

6.1.1. Les ressources humaines

L'équipe de développement de Music Sheet Writer compte six personnes. Bien qu'issue de la même formation, chacun des membres a ses propres compétences et affinité à certains domaines. Se retrouvent donc des profils de développeurs mobile, web et software. Nous disposons aussi de musiciens avec des connaissances en solfège, point non négligeable pour la réalisation d'un tel projet.

6.1.2. Les ressources matérielles

- Ordinateurs avec environnement Windows et Mac OS
- Une guitare
- Un Zoom G3 (interface audio USB/MIDI pour guitare)
- Deux smartphones Android
- Un iPhone
- Un Windows Phone

6.2. Le planning

Entre Avril 2015 et Septembre 2015 :

- Rendu des documents demandés par le LAB EIP

Entre Septembre 2015 et Octobre 2016 :

- Développement des différents livrables du projet Music Sheet Writer

Entre Novembre 2016 et Janvier 2017 :

- Débogage et finition du projet.
- Communication autour du projet
- Amélioration de l'image de marque
- Prise de contact avec des artistes et majors

6.3. La méthodologie

6.3.1. Les outils

Nous utiliserons une messagerie classique contenant l'ensemble des discussions en rapport avec le projet. Google Hangout sera l'outil que nous utiliserons pour toutes les discussions demandant une réponse rapide.

Nous avons à notre disposition Subversion, couramment appelé SVN, comme outils de gestion de version. Il sera utilisé pour le développement des différents logiciels du projet. Nous nous en servirons aussi comme espace de stockage commun. Y sera stocké tous les documents ayant un rapport avec le projet (documentation, comptes rendu de réunion, etc.).

Afin de mesurer et de voir l'avancement du projet nous utiliserons l'utilitaire Trello. Ce dernier viendra en complément du Gantt. Il sera plus facile pour tout le monde de voir et de spécifier les modifications faites sur le projet en passant par Trello qu'en passant par le Gantt. Ce dernier sera cependant mis à jour à chaque réunion.

6.3.2. La méthode

Nous suivrons une norme de codage spécifique à chaque langage. Cela facilitera la compréhension et la reprise du code d'un membre par un autre si jamais l'occasion se présente.

Lorsqu'une modification est faite sur une partie du projet, il faudra s'assurer du bon fonctionnement des modifications apportées avant de faire un rendu sur le SVN. De plus, il faudra toujours s'assurer d'avoir la dernière version du dépôt avant de faire le rendu. Une fois le rendu effectué, il sera obligatoire d'envoyer un mail afin de prévenir les autres membres du groupe qu'une modification a été apportée. Chaque rendu devra contenir un message explicite expliquant les modifications apportées.

6.3.3. Les réunions

Les réunions s'effectueront en visioconférence via Google Hangout. Au cours de ces réunions, chacun expliquera le travail effectué et les problèmes rencontrés qu'ils soient d'ordre technique, organisationnel ou relationnel. À la fin de la réunion, la date de la prochaine sera décidée. Un compte rendu de la réunion sera alors envoyé au groupe par e-mail. Il sera aussi récupérable sur le SVN.