–

J. Racaud;A. Simon;J. Harrault;J. Blondeel;S. Daguenet;F. Corradin

Music Sheet Writer

Music Sheet Writer

Bilan Architecture

Objectifs du document

Résumé

Glossaire

MVC

Description du document

|  |  |
| --- | --- |
| Titre | Music Sheet Writer : Bilan Architecture |
| Date de création | 21/12/2015 |
| Date de publication | 10/01/2016 |
| Auteur | J. Racaud;A. Simon;J. Harrault;J. Blondeel;S. Daguenet;F. Corradin |
| Responsable | Jonathan Racaud |
| E-mail | musicsheetwriter\_2017@labeip.epitech.eu |
| Sujet | Bilan Architecture |
| Version du modèle | 2.0 |

Tableau des révisions

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Auteur | Section(s) | Commentaire |
| 21/12/2015 | Jeremy HARRAULT | Toutes | Création du document à partir de la version 1.0 |

Sommaire

[1. Introduction 1](#_Toc438422828)

[1.1. Rappel de l’EIP 1](#_Toc438422829)

[1.2. Contexte et périmètre du projet 1](#_Toc438422830)

[2. Représentation de l’architecture globale 2](#_Toc438422831)

[3. Vue globale du projet 3](#_Toc438422832)

[3.1. Cas d’utilisation principaux 3](#_Toc438422833)

[3.2. Cas d’utilisation détaillés 3](#_Toc438422834)

[4. Vue logique de l’application 4](#_Toc438422835)

[4.1. Le logiciel 4](#_Toc438422836)

[4.2. Le site internet 4](#_Toc438422837)

[4.3. Les applications mobiles 4](#_Toc438422838)

[5. Vue processus 5](#_Toc438422839)

[5.1. Le logiciel 5](#_Toc438422840)

[5.2. L’API 5](#_Toc438422841)

[5.3. Les applications mobiles 5](#_Toc438422842)

[6. Vue déploiement 6](#_Toc438422843)

[6.1. Le logiciel 6](#_Toc438422844)

[6.2. Le site internet & les applications mobiles 6](#_Toc438422845)

[7. Implémentation 7](#_Toc438422846)

[7.1. Le logiciel 7](#_Toc438422847)

[7.2. L’API 7](#_Toc438422848)

[7.3. Les applications mobiles – iPhone 7](#_Toc438422849)

[7.4. Les applications mobiles – Android 7](#_Toc438422850)

[7.5. Les applications mobiles – Windows Phone 7](#_Toc438422851)

[8. Vue données 8](#_Toc438422852)

[8.1. Modélisation 8](#_Toc438422853)

[8.2. Description 8](#_Toc438422854)

[9. Taille et performance 9](#_Toc438422855)

[10. Annexes 10](#_Toc438422856)

[10.1. Le document du CDC 10](#_Toc438422857)

Liste des Tableaux

**No table of figures entries found.**

Liste des Figures

**No table of figures entries found.**

# Introduction

## Rappel de l’EIP

## Contexte et périmètre du projet

# Représentation de l’architecture globale

# Vue globale du projet

## Cas d’utilisation principaux

### Le logiciel

### Le site internet

### Les applications mobiles

## Cas d’utilisation détaillés

### Le logiciel

#### Editer une partition

#### Lire une partition

#### Gérer un projet

### Le site internet & application mobile

#### Accéder à la communauté

#### Gérer son compte utilisateur

#### Accéder à la vitrine (site internet uniquement)

# Vue logique de l’application

Pour chacun des livrables, la vue logique est présentée grâce à un diagramme de composant ainsi qu’une description textuelle mettant en avant les composants essentiels du diagramme.

## Vue globale

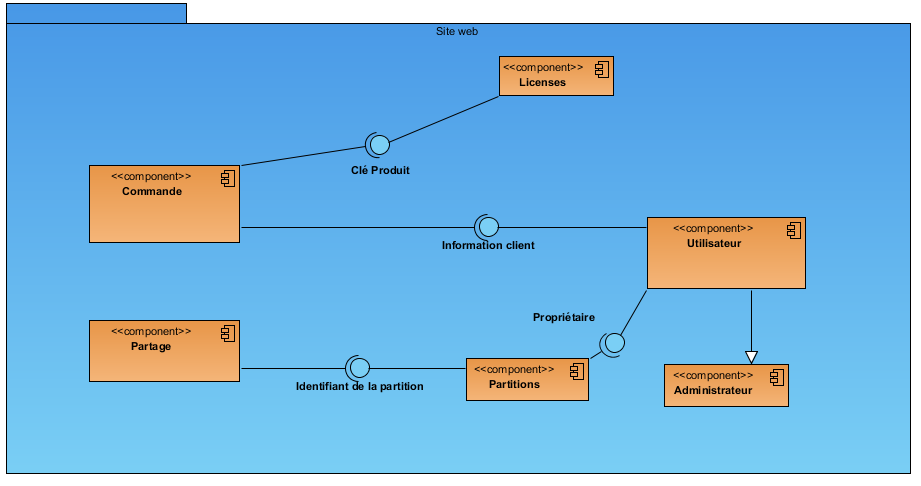
### Le site internet

‑ Diagramme de déploiement du site web

Le site web est hébergé sous un serveur linux et fonctionne avec le serveur web Apache 2.

La structure logicielle Symfony est utilisée pour regrouper toutes les interactions entre le système de mail, la base de données et l’application.

Des données fixes sont aussi utilisées pour charger les comptes administrateurs en base lors du déploiement de l’application.

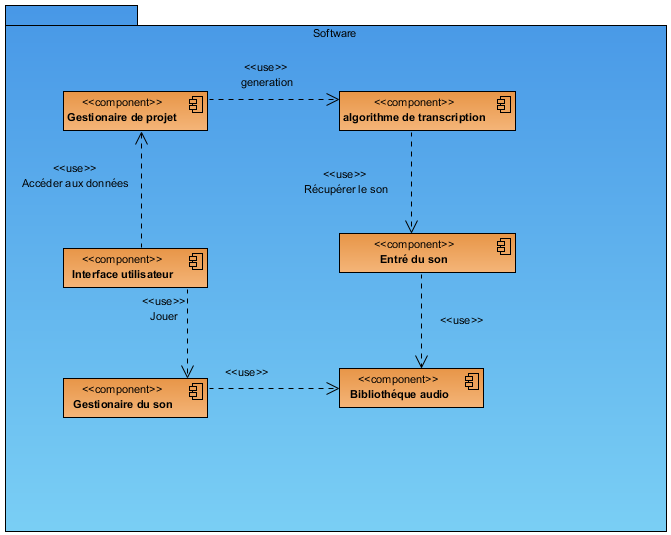


‑ Diagramme de composants du site web

Les clients du site internet sont représentés par le composant utilisateur qui est utilisé par héritage pour les administrateurs qui sont des utilisateurs avec des privilèges.

Le partage de partition (composant partage) permet l’accès aux partitions pour un client connecté. Le composant partition regroupe l’ensemble des actions qui peuvent être effectués par l’utilisateur sur la gestion de ces partitions.

### Le logiciel



‑ Diagramme de composants du logiciel

### Les applications mobiles

## Composants principaux

### Le site internet

#### Base de données

La base de données regroupe toutes les données utilisateur, comprend les partitions, informations des utilisateurs et données de l’espace communautaire. La base de données peut être du type MySQL, MariaDB ou PostgreSQL.

#### Serveur d’application

Le serveur d’application héberge le site internet et l’application web. Un serveur apache 2 est utilisé pour permettre le fonctionnement du Framework Symfony.

#### Serveur mail

Le serveur mail est utilisé pour envoyer des mails depuis l’application vers des utilisateurs ou pour contacter l’équipe de développement du projet.

### Le logiciel

#### Le projet

Le projet regroupe toutes les données nécessaires pour l’édition de partition, l’export de partition, la lecture, l’affichage et la suppression de la partition.

#### L’interface Utilisateur

L’interface utilisateur permet à l’utilisateur du logiciel d’avoir un aperçu de la partition sur laquelle il travail et de gérer l’ensemble de ces projets.

#### Le son

Ce composant va gérer toute les sorties sonore du logiciel notamment lors de la lecture d’une partition.

#### L’algorithme de transcription

C’est ce composant qui va permettre de transcrire les données numérique reçu depuis l’instrument dans un format de notation musicale.

#### La récupération du son

Permet de récupérer les signaux sonore ou donnée MIDI depuis un port de la machine.

# Vue processus

## Le logiciel

## L’API

### Création de compte

#### Description

#### Vue dynamique

### Connexion

#### Description

#### Vue dynamique

### Changement d’adresse email

#### Description

#### Vue dynamique

### Réinitialisation du mot de passe

## Les applications mobiles

# Vue déploiement

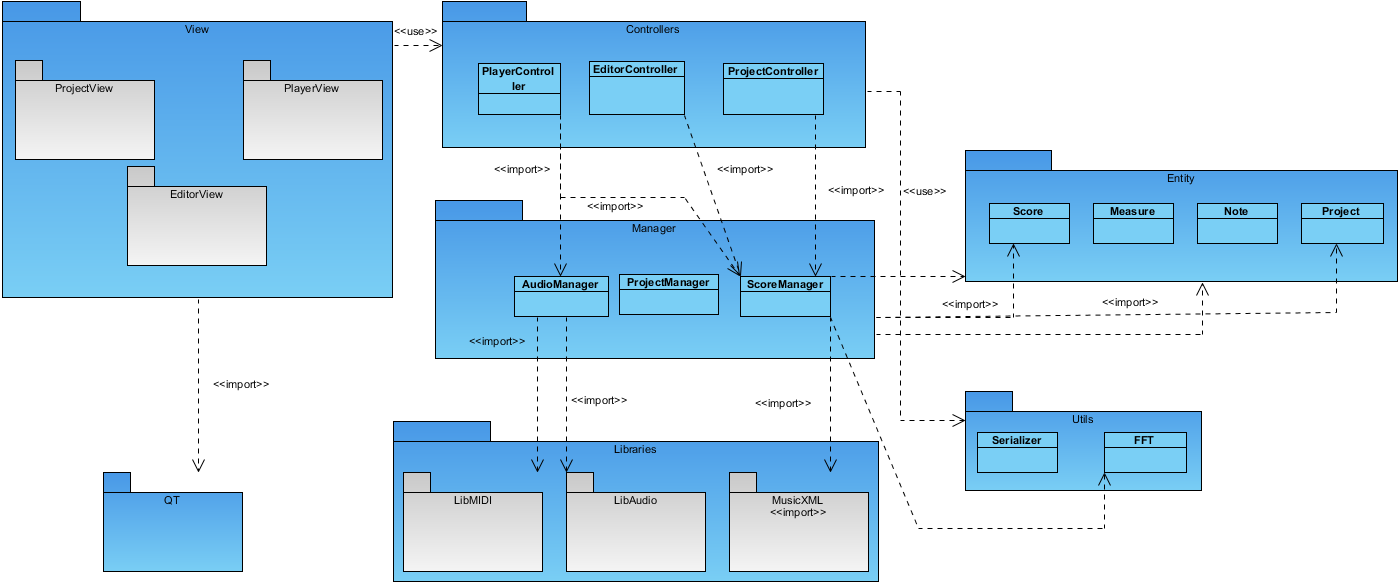
## Le logiciel

## Le site internet & les applications mobiles

# Implémentation

## Le logiciel

### Vue globale



‑ Vue globale du logiciel

Le logiciel suit une architecture MVC ou la vue est représentée par les composants **ProjectView**, **PlayerView** et **EditorView**. Les contrôleurs vont permettre de contrôleur l’ensemble des actions sur les vue correspondantes. Ainsi chaque vue à son contrôleur associé.

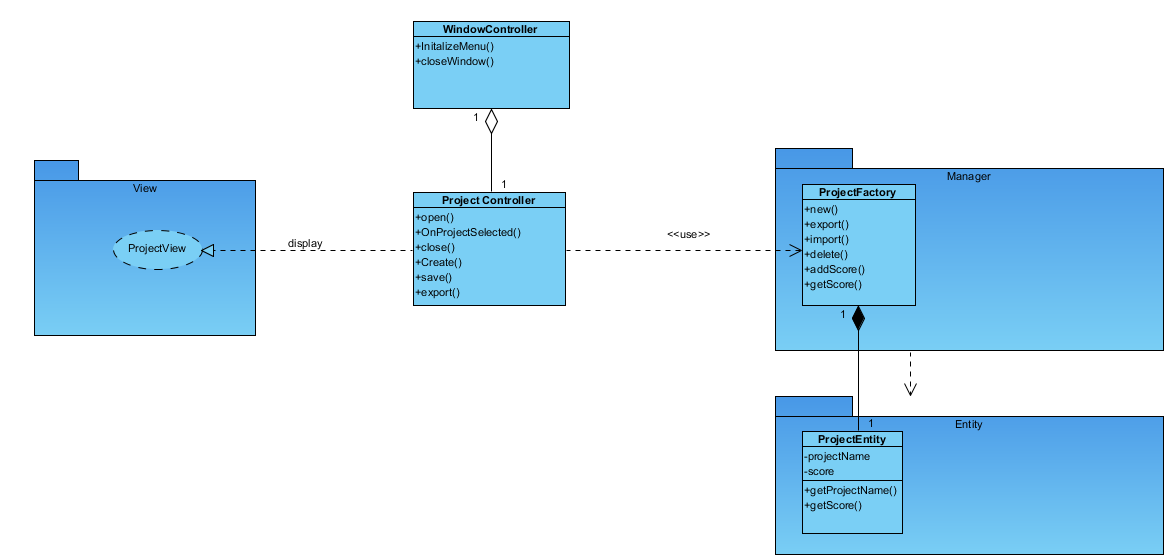
Les entités (Score, Measure, …) représentent couche d’accès aux données qui vont être utilisé.

La partie Manager représente la couche métier de l’application.

### Couches applicatives

#### Couche présentation

##### Affichage d’un projet

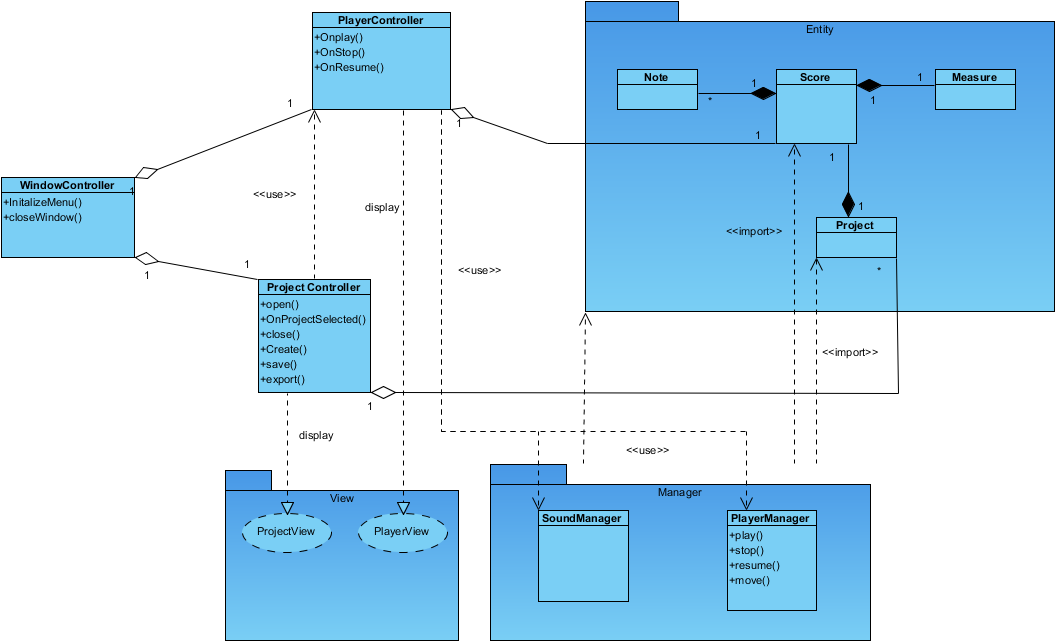


La classe **WindowController** contrôle l’ensemble des actions et des évènements qui peuvent être déclenchés dans la fenêtre principale de l’application.

**ProjectController** contrôle juste la vue concernant la gestion de projet, cette classe est donc contenue dans le **WindowController** puisqu’elle fait partie de la fenêtre principale.

Les actions du contrôleur sont appliquées à la vue du projet (**ProjectView**).

##### Lecture d’une partition



La Classe **PlayerControlleur** contrôle la vue **playerView** et est contenu dans la classe **WindowController**.

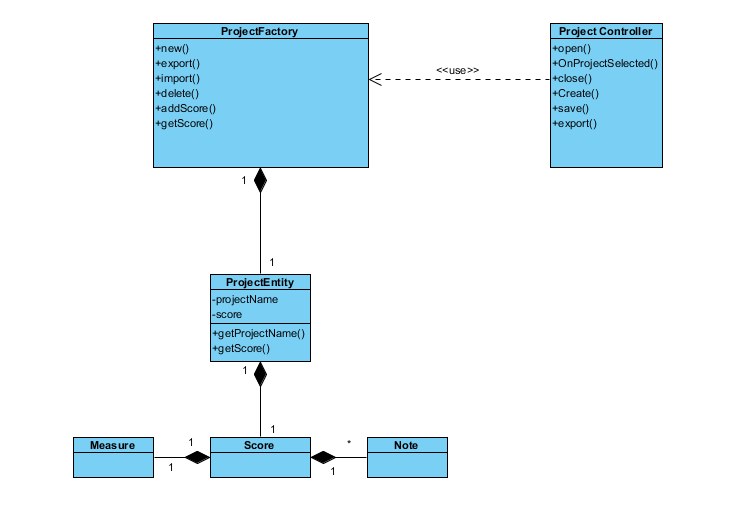
La logique du code implémentant les actions de la lecture de partitions sont dans la partie Manager. La lecture de partitions a besoin du **playerManager** qui va exécuter les taches nécessaires à la réalisation de l’action. Le **PlayerController** requière aussi l’utilisation du **soundManager** qui va permettre de jouer le son.

##### Edition de partition

#### Couche métier

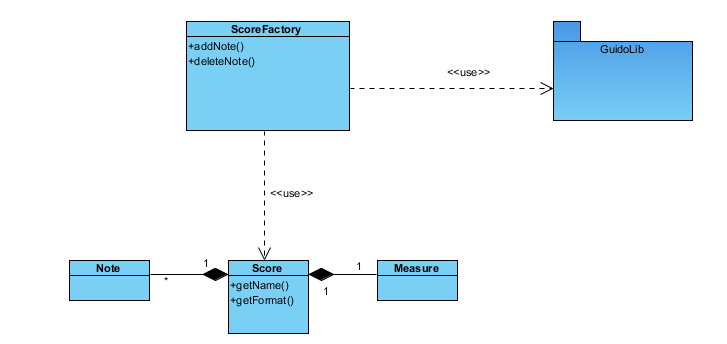
La couche métier représente la partie du code qui va exploiter les données.

##### Gestion d’un projet



Toute la gestion du projet est gérée par la classe **ProjectFactory** qui contient toutes les méthodes nécessaires pour la création, suppression modification d’un projet.

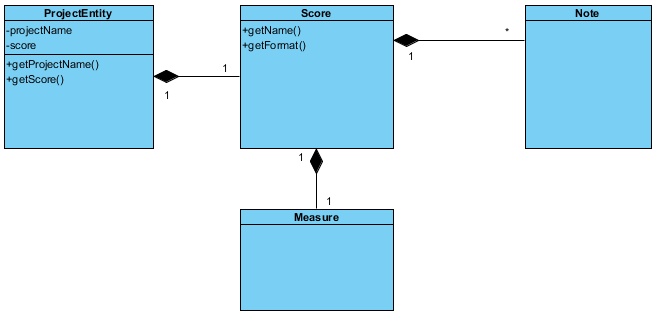
##### Gestion de partitions



#### Couche accès aux données

Cette couche représente l’état dans lequel vont être représenté toutes les données nécessaires pour le fonctionnement du logiciel.

##### Les Entités



Les données relatives aux partitions seront stockées dans l’entité **Score**. Une partition étant composée de mesure et de notes la classe partition contient donc des instances de ces classes.

Un projet et considéré comme l’écriture d’une partition la classe partition fait partie du projet et est stockée dans la classe **Project**.

## L’API

### Vue globale

### Couches applicatives

## Les applications mobiles – iPhone

### Vue globale

### Couches applicatives

## Les applications mobiles – Android

### Vue globale

### Couches applicatives

## Les applications mobiles – Windows Phone

### Vue globale

### Couches applicatives

# Vue données

## Modélisation

## Description

# Taille et performance

# Annexes

CTRL clic permet de visualiser ces documents

## Le document du CDC

