

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

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 |
| 28/12/2015 | SIMON Antoine | 4 & 7 (Iphone) | Rédaction |
| 2/12/2015 | SIMON Antoine | 4 & 7 (Iphone) | Nouvelle rédaction pour la partie 7 et correction d’erreur partie 4. |
| 09/01/2015 | Simon Antoine | 7 (Iphone) | Ajout d’information |

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)

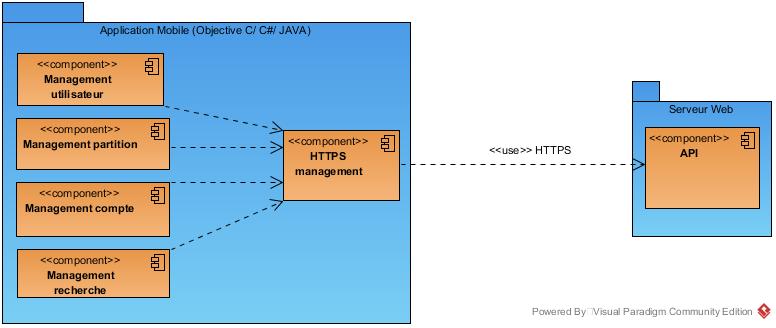
# Vue logique de l’application

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.

## Le logiciel

## Le site internet

## Les applications mobiles



L’architecture et la relation entre les différents composants des applications mobiles sont les mêmes sur les plateformes Android, iOS, Windows Phone.

Chacune de ces applications sera développées dans son langage natif, Java pour Android, Objective C pour IOS et C# pour Windows Phone.

Une application mobile s'organisera autour de quatre parties principales, le management utilisateur, management de partition, management compte et management recherche. Toutes ces parties sont reliées au management du HTTPS pour envoyer et récupérer des informations sur le Serveur Web grâce à l'API mise en place sur ce dernier.

Le management utilisateur sera pour la gestion de tous les utilisateurs sur l'application. Il y a donc l'utilisateur même de l'application et tous ceux créant la communauté. Cette partie sera là pour la récupération, la modification et l'affichage des informations personnelles.

Le management partition lui fera la même chose que celui utilisateur mais pour la gestion des partitions, sans la possibilité de modification.

Le management compte prend en compte la connexion, la création de compte, la gestion du mot de passe perdu. Ce composant est l'un des premiers que l'utilisateur verra sur l'application lors de l'utilisation.

Le management recherche touche les partitions et les utilisateurs. Il servira à la recherche et donnera la possibilité de pouvoir filtrer la liste des utilisateurs et des partitions.

Il y aura aussi le management de la partie HTTPS qui servira ensuite à pouvoir communiquer avec le serveur web mis en place et son API.

# 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

### Couches applicatives

## L’API

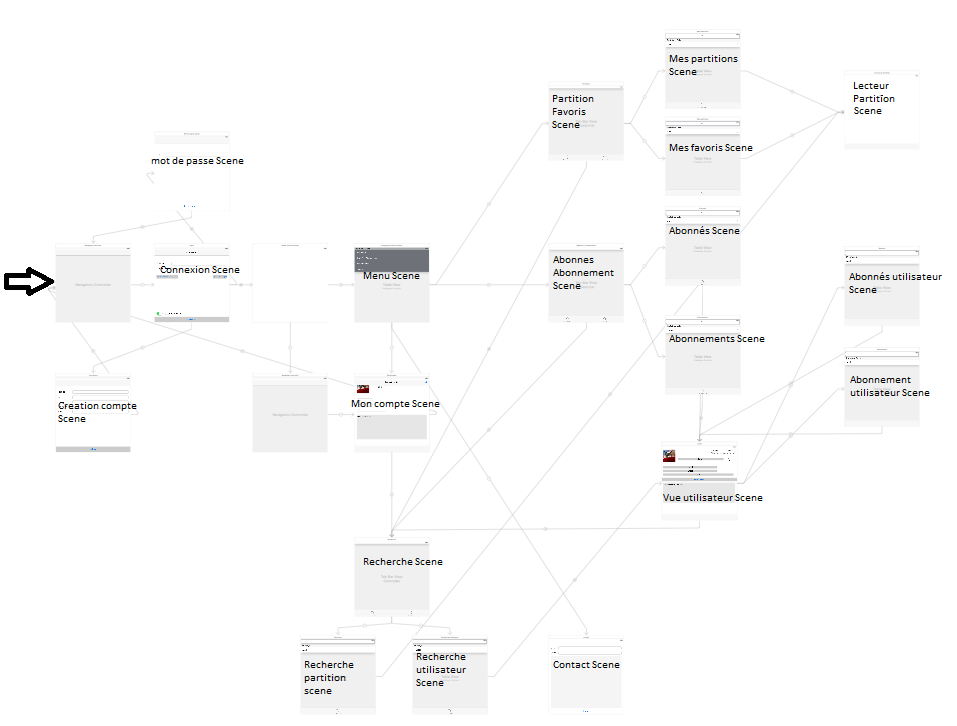
### Vue globale

### Couches applicatives

## Les applications mobiles – iPhone

### Vue globale

L’application IOS n’offrant pas assez de possibilité et de visualisation grâce au diagramme, il a été décidé de montrer les différentes vues avec leurs noms respectifs. En effet, avec Xcode il faut créer une classe par vue ce qui rend l’explication par diagramme confus. Une explication d’une vue sera donné en 7.3.2, seule une vue sera expliqué car elles sont toutes prototypées de la même manière.

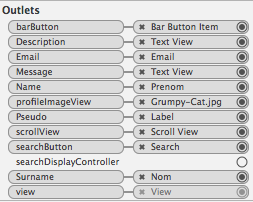


### Couches applicatives

Pour la couche applicative, l’exemple de la vue « Mon compte » sera expliqué.

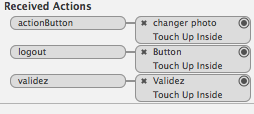
La vue « mon compte » permet l’affichage des informations du compte de l’utilisateur connecté et donc la modification de ces derniers. Cette vue appelle donc la classe AccountViewController pour le chargement et l’affichage des informations.

Chaque élément de la vue est lié au code grâce à Xcode.



C’est ainsi que les informations peuvent communiquer entre le code et la vue et c’est aussi comme cela que les liens sont fait entre les éléments de la vues et les variables du codes.

Certains éléments sont capables d’appeler des méthodes après pression.



Les liens sont une nouvelle fois créer de la même manière.

Le fichier.m quant à lui est prototypé de façon suivante, tout d’abord il y a la méthode « ViewDidLoad » qui est la première méthode appelée. Elle va générer la barre supérieure pour la gestion des raccourcis puis va appeler une méthode « refresh » qui permet d’afficher les bonnes informations de l’utilisateur. Cette méthode « refresh » appelle une méthode de la classe « APIMethod », qui va envoyer ou récupérer des informations selon l’action souhaité.

## 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

