**cahier des charges**

**MAKON Maniym ma**

**NAJMI Mehdi**

**MVE MEYE BEKOUROU Sankara**

**IOINASCU Félicia**

# **Génie logiciel et Projet GLPOO**

* MAKON MANYIM MA
* MVE MEYE BEKOUROU Sankara
* NAJMI Mehdi
* Professeur: Ioinascu Félicia

**Table des matières**

[I. CADRE DU PROJET 2](#_Toc68161689)

[1. Présentation de l’équipe 2](#_Toc68161690)

[2. Enjeux et objectifs 2](#_Toc68161691)

[3. Livrables 2](#_Toc68161692)

[4. Planning prévisionnel 2](#_Toc68161693)

[II. SPECIFICATIONS 3](#_Toc68161694)

[1. Exigences fonctionnelles et non-fonctionnelles (faire un tableau de deux colonnes) 3](#_Toc68161695)

[2. Diagramme de cas d’utilisation 3](#_Toc68161696)

[3. Diagramme de séquence 4](#_Toc68161697)

[4. Backlog 6](#_Toc68161698)

[III. CONCEPTION 6](#_Toc68161699)

[1. Diagramme UML de package, de composant, de déploiement et de classe 6](#_Toc68161700)

[2. Patrons de conception utilisés 6](#_Toc68161701)

[3. Analyse de la conception selon les principes SOLID avec argumentaire 6](#_Toc68161702)

[IV. RAPPORTS DE TESTS 6](#_Toc68161703)

[1. Tests d’acceptance exécutés 6](#_Toc68161704)

[2. Listes des bugs trouvés avec leur priorité et leur sévérité 6](#_Toc68161705)

# CADRE DU PROJET

## Presentation de l’équipe

Notre équipe est composée de trois étudiants de classe 31 qui sont

MAKON MANYIM MA

Sankara MVE MEYE BEKOUROU

Mehdi NAJMI.

## Enjeux et objectifs

L’objectif du projet génie logiciel est de nous faire travailler de manière collaborative sur un sujet donné, en mettant à l’œuvre les notions du cours, en améliorant nos compétences en POO et Java, et en produisant tous les artefacts d’un processus de développement de projet.

## Livrables

Comme livrable, il est attendu une application fonctionnelle. Celle-ci sera basée sur l’application fournie par le professeur et notre but est de l’améliorer.

## Planning prévisionnel

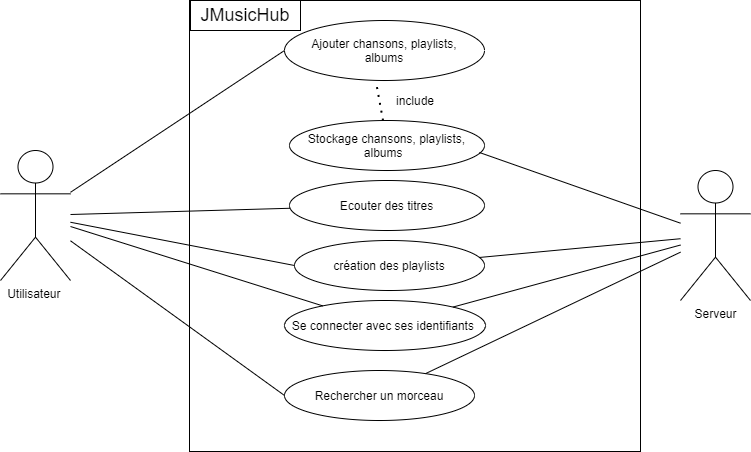
Toutes les modifications logicielles que nous devrons appliquées au logiciel seront opératinnelles au 15 avril 2021.

# SPECIFICATIONS

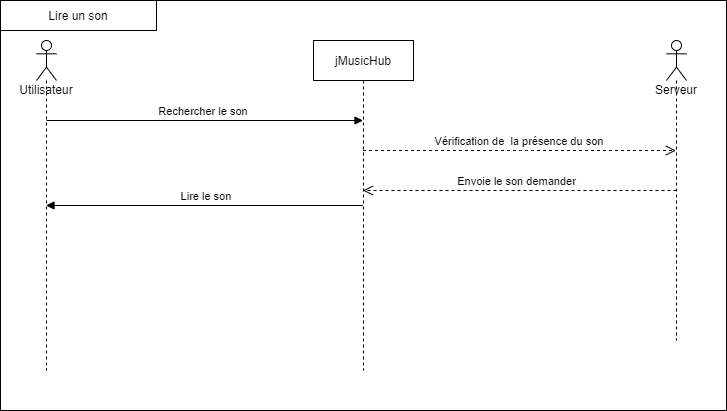
## Exigences fonctionnelles et non-fonctionnelles

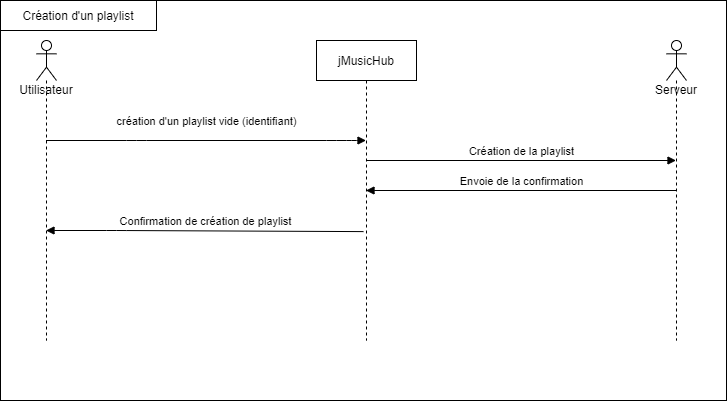
|  |  |
| --- | --- |
| **Exigences Fonctionnelles** | **Exigences Non-Fonctionnelles** |
| Ajouter des morceaux à un album. | Travail uniquement en Java. |
| Lire aléatoirement des morceaux. | Utilisation de l'application sur les OS Windows, Linux, et IOS. |
| Sélection des chansons d'un album. | L'application ne fonctionne que dans la console. |
| Sélection des chansons d'un album. |  |
| Affichage des chansons rangés par date de sortie, par genre, par playlist, par auteur (livre). |  |
| Rajout d'une nouvelle chanson, album. |  |
| Rajout d'une chanson existante à un album. |  |
| Rajout d'un livre audio. |  |
| Création d'une nouvelle Playlist à partir de chansons et livres audio existants. |  |
| Sauvegarde des Playlist, des albums, des chansons et des livres audios dans les fichiers XML respectifs. |  |

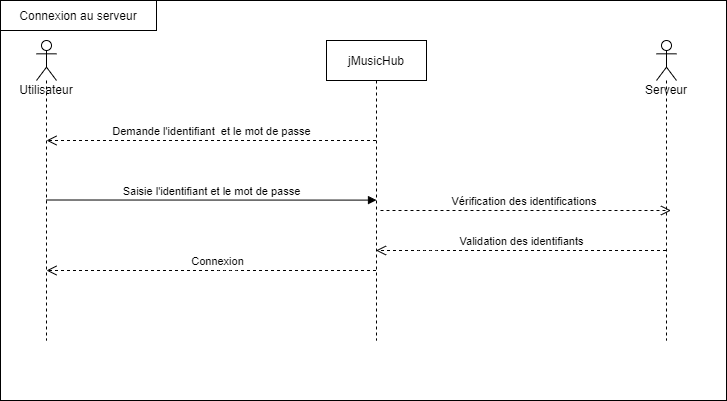
## Diagramme de cas d’utilisation

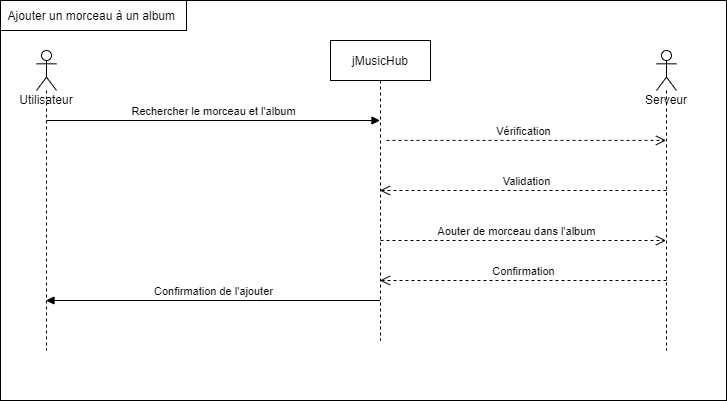


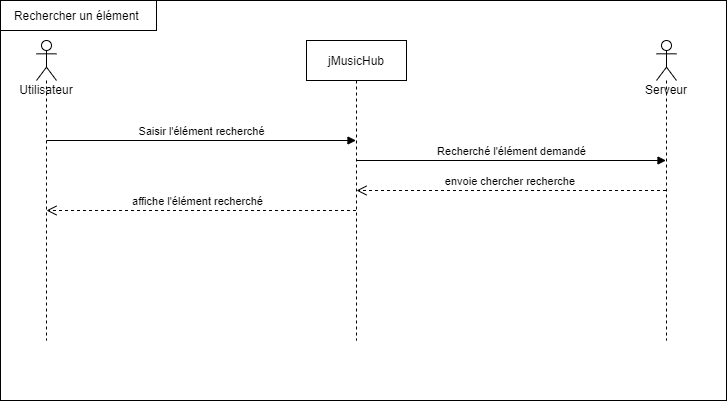
## Diagrammes de sequence











## Backlog

# CONCEPTION

## Diagramme UML de package, de composant, de déploiement et de classe

## Patrons de conception utilisés

## Analyse de la conception selon les principes SOLID avec argumentaire

# RAPPORTS DE TESTS

## Tests d’acceptance exécutés

## Listes des bugs trouvés avec leur priorité et leur sévérité