

Génie Logiciel : Projet

Acteurs du projet :

Thibaut Riffort est le product owner, il avait la charge du Trello, il a rédigé la liste des exigences ainsi que le backlog. Il a également produit les diagrammes de cas d'usage et de séquence.

Alexandre Hayez est développeur. Il a implémenté le système de journalisation et un design pattern singleton. Il était en charge des tests unitaires. Il a produit le diagramme de classe UML.

Jimmy Moren est développeur, il a veillé au respect des principes SOLID. Il a implémenté le MVC, les fonctionnalités liées au lecteur audio. Il avait également la charge de la javaDoc et du script Maven.

Le projet :

L'équipe a choisi d'ajouter deux fonctionnalités à l'application :

- La possibilité de pouvoir lire aléatoirement un élément
- Un lecteur permettant de lancer, arrêter et recommencer la lecture d'éléments

La liste des exigences fonctionnelles et non-fonctionnelles est disponible au sein du Trello de notre équipe : <https://trello.com/b/BbmndW8b/jmusiclub>

Lien du GitHub : <https://github.com/JamelAuto/GLPOO>

Analyse SOLID :

La responsabilité unique est respectée car chaque classe et méthode n'ont qu'une seule responsabilité. Pour le concept d'ouvert/fermé l'abstraction de la classe AudioElement permet de le valider. La substitution de Liskov est respectée grâce à l'utilisation des classes Song et AudioBook qui sont des enfants de AudioElement. On peut dire que le principe de Ségrégation des interfaces est respecté étant donné qu'aucune interface n'est utilisée. L'Inversion des dépendances n'est respectée, la principale raison est qu'il n'y a aucune interface dans le programme.

Test d'acceptance :

ID	1001
Description	Lors de l'ajout d'un élément qui se fait correctement. Notre méthode nous retourne true.
User Story	1.1
Résultat :	Réussis.
ID	1002

Description	Lors de l'ajout d'un album qui se fait correctement. Notre méthode nous retourne true.
User Story	1.1
Résultat :	Réussis.
ID	1003
Description	Lors de l'ajout d'une playlist qui se fait correctement. Notre méthode nous retourne true.
User Story	1.2
Résultat :	Réussis.
ID	1004
Description	Lors de la suppression d'un album qui se fait correctement. Notre méthode vérifie que la taille est bien descendue de 1.
User Story	1.4
Résultat :	Echec.

myLogger.log rassemble tous les logs obtenus par notre système.

Durant la phase de tests des fonctionnalités nous avons identifié un bug au niveau de la création de playlist, après avoir entré le nom de la playlist il y'a un affichage des noms d'éléments puis le système attend une entrée utilisateur (à cause d'un scanner en attente). Ce bug est ennuyeux car si l'utilisateur ne sait pas qu'il doit entrer une valeur et qu'il appuie uniquement sur entrer alors le programme plante. Il peut compromettre l'utilisation du logiciel donc sa sévérité est **S2** et sa priorité **P1**.

Nous avons également rencontré un bug lors du choix de la longueur d'un album / playlist / livre audio, lorsque l'utilisateur rentre des lettres alors qu'on attend des chiffres. Le programme lève une exception. Pour gérer ce problème nous avons utilisé un try / catch. Sa sévérité est **S2** et sa priorité **P2**.

Diagramme UML des cas d'utilisations (version originale disponible sous /Rendu projet) :

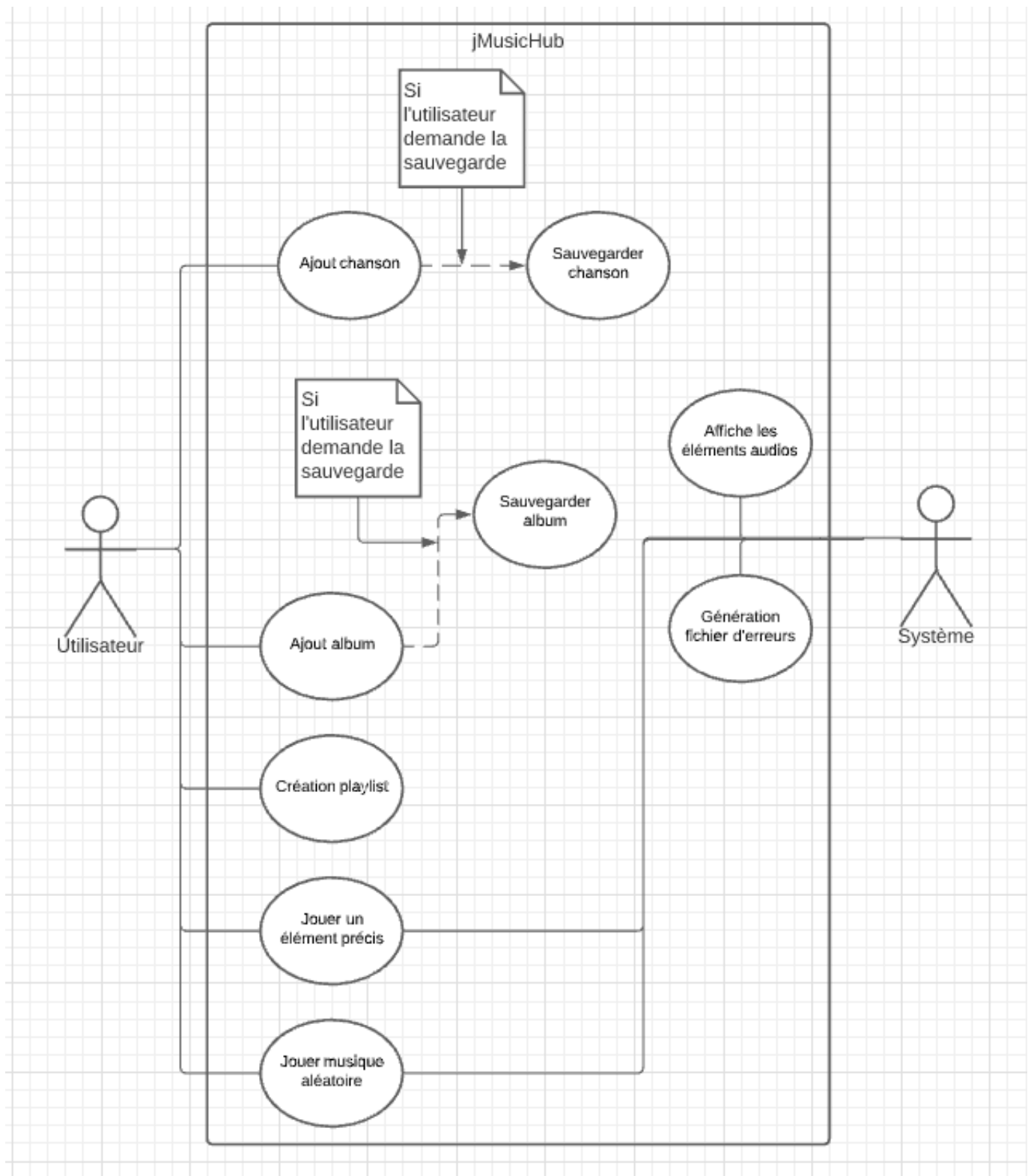


Diagramme UML de séquence (version originale disponible sous /Rendu projet) :

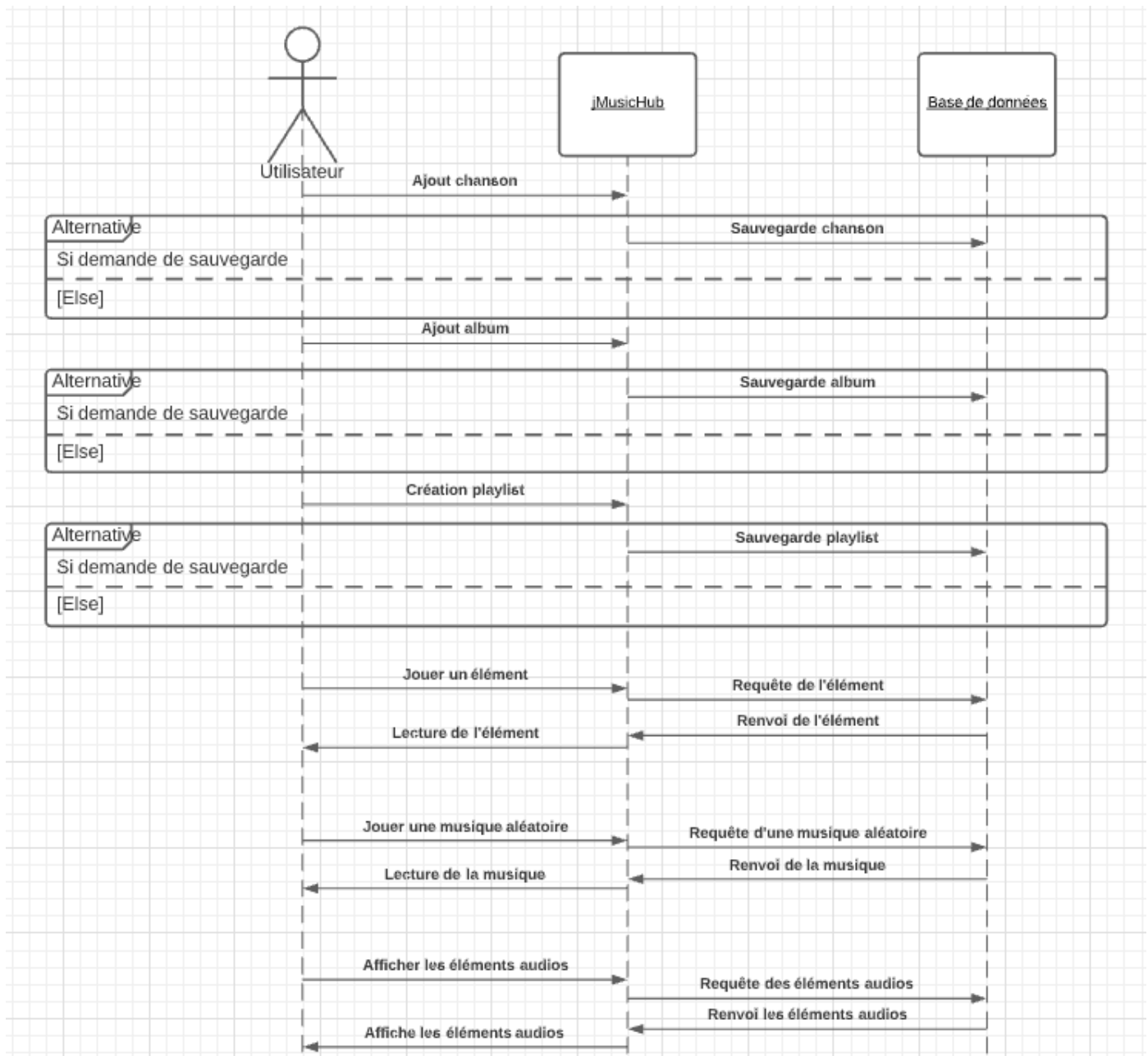


Diagramme UML de classe (version originale disponible sous /Rendu projet) :

