

## Projet S2

### Génie Logiciel et Programmation Orientée objet

#### I. Organisation du projet

Ce projet a été réalisé par quatre étudiants : CHAUVREAU-MANAT Steve, LEJEUNE Gaël, MORCRETTE Antonin et PROUX Angélique.

CHAUVREAU-MANAT Steve était en charge du serveur, ce qui consistait à coder le fonctionnement serveur-client de l'application afin que les données soient stockées sur le serveur et que le client n'ait qu'à s'y connecter pour y avoir accès. Il avait également les responsabilités suivantes : **liste des responsabilités**.

Il a effectué les tâches suivantes comme l'atteste le GitHub : **liste des tâches**.

LEJEUNE Gaël était en charge de la fusion du code, ce qui consistait à fusionner certaines classes créées pour nos deux projets POO (celui de Steve et de Gaël et celui de Manelle et d'Angélique) en choisissant celles qui répondent plus aux attentes du projet. Il avait également les responsabilités suivantes : **liste des responsabilités**.

Il a effectué les tâches suivantes comme l'atteste le GitHub : **liste des tâches**.

MORCRETTE Antonin était en charge du serveur, ce qui consistait à coder le fonctionnement serveur-client de l'application afin que les données soient stockées sur le serveur et que le client n'ait qu'à s'y connecter pour y avoir accès. Il avait également les responsabilités suivantes : **liste des responsabilités**.

Il a effectué les tâches suivantes comme l'atteste le GitHub : **liste des tâches**.

PROUX Angélique était en charge de la lecture de l'audio, ce qui consistait à créer des classes et des méthodes pour écouter sur le client l'audio transmis par le serveur. Elle avait également les responsabilités suivantes : **liste des responsabilités**.

Elle a effectué les tâches suivantes comme l'atteste le GitHub : **liste des tâches**.

#### II. Spécification

##### II.1) Cahier des charges

##### **II.1.A) Exigences fonctionnelles et non-fonctionnelles**

ID	Nom	Type
F REQ_1	Fonctionnalité de base	Fonctionnelle
F REQ_2	Réglages musicaux	Fonctionnelle
F REQ_3	Playlist	Fonctionnelle

<b>NF REQ_1</b>	Portabilité sur PC	Non fonctionnelle
<b>F REQ_4</b>	Simplicité	Fonctionnelle
<b>F REQ_5</b>	Recherche par tri	Fonctionnelle
<b>NF REQ_2</b>	Langages utilisés	Non fonctionnelle
<b>NF REQ_3</b>	Interface	Non fonctionnelle
<b>NF REQ_4</b>	Standards du projet	Non fonctionnelle
<b>F REQ_6</b>	Commandes de base	Fonctionnelle
<b>F REQ_7</b>	Journalisation des erreurs	Fonctionnelle

<b>ID</b>	<b>F REQ_1</b>
Description	L'application devra permettre d'écouter des chansons et des livres audio disponibles en ligne.
Priorité	Must
Version	1.0

<b>ID</b>	<b>F REQ_2</b>
Description	Des réglages seront disponibles : volume, morceau écouté, lecture, pause, ...
Priorité	Must
Version	1.0

<b>ID</b>	<b>F REQ_3</b>
Description	L'application devra permettre de créer des playlists contenant les chansons, les livres audio et les albums présents dans l'application.
Priorité	Must
Version	1.0

<b>ID</b>	<b>F REQ_4</b>
Description	L'application devra être simple d'utilisation.
Priorité	Must/Should
Version	1.0

<b>ID</b>	<b>F REQ_5</b>
Description	L'application devra proposer le tri des musiques, livres audio, albums et playlists par genre, catégorie, langue, artiste et auteur.
Priorité	Must
Version	1.0

<b>ID</b>	<b>F REQ_6</b>
Description	L'application doit répondre correctement à des commandes données (c, l, a, p, +, -, s, h).
Priorité	Must
Version	1.0

ID	F REQ_7
Description	Les erreurs (erreurs, avertissements ou informations) survenues lors de l'exécution doivent être écrites dans un fichier log quand elles ont lieu. La date, l'heure et le type d'exception doit être renseigné.
Priorité	Must
Version	1.0

ID	NF REQ_1
Description	L'application devra être proposée sur Linux et Windows. L'application ne doit donc pas être disponible ni sur le web ou ni sur mobile.
Priorité	Must
Version	1.0

ID	NF REQ_2
Description	L'application devra être codée en JAVA et utiliser des XML pour stocker les éléments.
Priorité	Must
Version	1.0

ID	NF REQ_3
Description	L'interface de l'application doit être sur console.
Priorité	Must
Version	1.0

ID	NF REQ_4
Description	L'application devra respecter les standards SOLID, suivre 2 modèles de conception, être disponible sur Github, utilisable avec JUnit et comporter un script Maven.
Priorité	Must
Version	1.0

ID	NF REQ_5
Description	L'interface de l'application doit être sur console et non web ou mobile.
Priorité	Must
Version	1.0

### II.1.B) Diagramme UML des cas d'utilisation et de séquence

*voir ce que j'ai commencé*

### II.1.C) Backlog des User Stories et des Constraint Stories

Ci-dessous vous trouverez un exemple d'User Stories (US) avec ses critères d'acceptance (AC).

US_1
<i>En tant que client, je veux accéder au contenu audio des morceaux afin d'écouter de la musique en ligne.</i>

US\_1 – AC

- Depuis le menu principal, je peux choisir d’écouter des albums / playlist / chansons / livres audio.
- Depuis le menu principal, je peux choisir d’arrêter l’audio en cours d’écoute.
- Depuis le menu principal, je peux passer à l’audio suivant si j’écoute un album ou une playlist.
- Depuis un album ou une playlist, je peux démarrer en aléatoire l’écoute.

Le reste des User Stories et les Constraint Stories sont visibles dans le Trello.

#### **II.1.D) Lien vers le Trello**

<https://trello.com/b/8mDdc95w/glpoo>

### III. Conception

#### III.1) Diagrammes UML

*À insérer ci-dessous avec les bons symboles*

##### **III.1.A) Diagramme package**

##### **III.1.B) Diagramme composants**

##### **III.1.C) Diagramme déploiement**

##### **III.1.D) Diagramme classes**

#### III. 2) Patrons de conception utilisés

*2 doivent avoir été codés dans le code*

#### III.3) Analyse de la conception respectant les principes SOLID

*Pourquoi chaque principe est respecté ou non*

### IV. Lien vers le GitHub

Lien :

Voici les identifiants des quatre étudiants sur GitHub :

CHAUVREAU-MANAT Steve : **Ilryss-Lurgorg**  
LEJEUNE Gaël : **Gael-Lejeune**  
PROUX Angélique : **angelique2000**  
MORCRETTE Antonin : **Antonin-Morcrette**

## V. Rapport de test

*liste des tests d'acceptance exécutés + liste des bugs trouvés*

*1 test par User / Constraint Story → réussi/non réussi → tableau avec estimation sévérité/priorité de chaque bug*

FIN DU RAPPORT