IMIE 2022/2023

DOSSIER DE SOUTENANCE

CONCEPTEUR DEVELOPPEUR D'APPLICATIONS

RÉDIGÉ PAR :
RANDY BOURDON

Table des matières

Liste des compétences du référentiel

N° Fiche AT	Activités types	N° Fiche CP	Compétences professionnelles	
1	Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité	1	Maquetter une application	✓
		2	Développer une interface utilisateur de type desktop	Projet pro.
		3	Développer des composants d'accès aux données	✓
		4	Développer la partie front-end d'une interface utilisateur web	✓
		5	Développer la partie back-end d'une interface utilisateur web	✓
2	Concevoir et développer la persistance des données en intégrant les recommandations de sécurité	6	Concevoir une base de données	✓
		7	Mettre en place une base de données	✓
		8	Développer des composants dans le langage d'une base de données	✓
3	Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité	9	Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement	Projet pro.
		10	Concevoir une application	✓
		11	Développer des composants métier	✓
		12	Construire une application organisée en couches	✓
		13	Développer une application mobile	Projet pro.
		14	Préparer et exécuter les plans de tests d'une application	✓
		15	Préparer et exécuter le déploiement d'une application	✓

I. Les remerciements

Tout d'abord, je souhaite remercier l'IMIE de m'avoir permis de poursuivre mes études en alternance, ainsi que les différents formateurs pour leurs enseignements qui m'auront permis d'acquérir des compétences et de consolider mes connaissances.

Je voudrais ensuite remercier, le président d'Interspheres, M. DE ROCHECHOUART et M. AUGE pour m'avoir donné l'occasion d'obtenir une alternance.
Je remercie également M. KAPLAN et Mme MANISY qui m'ont beaucoup appris et qui m'ont partagé leurs connaissances et expériences dans le milieu du développement, tout en m'accordant leurs confiances.

Je remercie enfin toute l'équipe d'Interspheres qui s'est toujours rendu disponible lorsqu'il y avait besoin et qui m'a toujours conseillé au mieux pour m'améliorer.

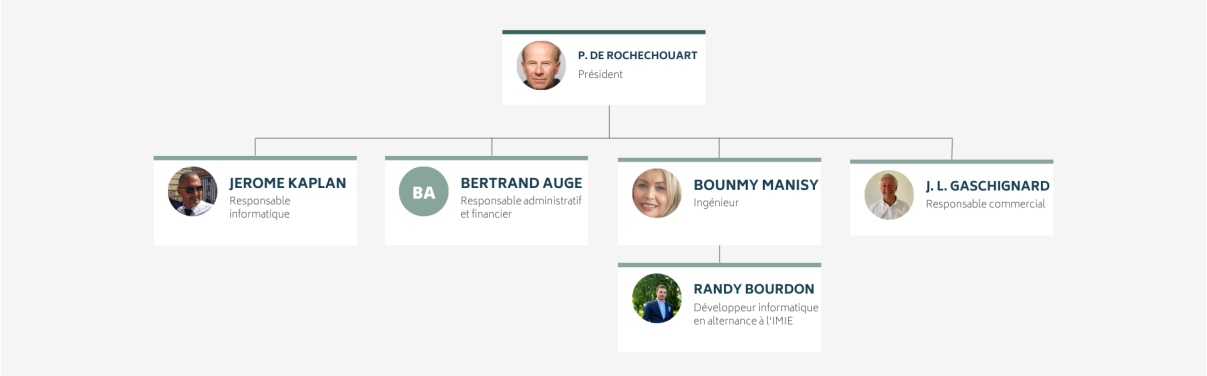
II. Présentation de l'entreprise

Créée en 1993, INTERSPHERES est une ESN (entreprise de services du numérique) spécialisée à l'origine dans l'intégration de solutions numériques pour la gestion des bâtiments. En 2018, elle devient un éditeur de solution informatique, consistant à intégrer des solutions pour la gestion de parcs immobiliers. Ses outils de pilotage lui permettent une optimisation économique associée aux charges d'exploitation du parc immobilier, ainsi qu'une gestion et une amélioration de l'environnement et de la qualité de vie au travail.

L'entreprise exploite, pour cela, les trois solutions suivantes : les plans numériques, afin de numériser et consolider des données du parc immobilier, le Costflow, pour le suivi et l'optimisation des flux énergétiques, les normes réglementaires, ainsi que le pilotage des sites, et l'ImmoData, pour la stratégie de l'environnement au travail tel que le télétravail.

INTERSPHERES est une entreprise de cinq collaborateurs, dont un développeur, et ses clients sont généralement des entreprises membres du CAC 40, souhaitant pouvoir gérer et piloter des sites. Les solutions apportées permettent ainsi aux entreprises d'effectuer des refacturations de surface afin de faire des économies.

Organigramme



III. Project summary

I'm a student in computer science and I'm doing my work-study in the company Interspheres, based in Paris near the Arc de Triomphe.

The company is a small SS2I with less than 10 employees.

During my internship, I had to recreate the company's website so that it was up to date.
For this work, I used HTML, CSS and JavaScript programming language.
For another mission, I had to develop a script in JavaScript to create portraits of people on a building map.

I also developed a script to automate a planning software in Visual basic and python.

Tunely is my project, and it's all about creating a free music streaming website. I want to offer users like you unlimited access to a vast library of music that spans across thousands of songs from various genres. With my intuitive and modern platform, you'll be able to discover new artists, curate personalized playlists, and easily share your favorite music with your friends.

One of the best things about Tunely is that I've worked hard to ensure that you won't be interrupted by annoying advertisements while enjoying your music.
I understand how important it is to have a seamless listening experience, and I want to provide that to you.

My goal is to provide a high-quality music streaming platform that's accessible to all music lovers, without any fees or restrictions. I believe that everyone should have the opportunity to enjoy music without any barriers.

I wish to continue my studies with a master, still half working and half studying.
Later, I would like to become a project manager in computer science.

IV. Présentation du projet

Le projet Tunely se présente comme une plateforme de streaming musical totalement gratuite.
Tunely a pour ambition de fournir aux utilisateurs un accès illimité à une immense collection de chansons provenant de divers genres musicaux. La plateforme, dotée d'une interface intuitive et contemporaine, permettra aux utilisateurs d'explorer de nouveaux artistes, de confectionner des listes de lecture personnalisées et de partager aisément leurs morceaux favoris avec leurs cercles sociaux.

De plus, Tunely garantira une expérience sans publicités, afin de garantir une écoute sans interruptions. L'objectif est de mettre en place une plateforme de diffusion musicale de haute qualité, ouverte à tous les mélomanes, sans frais ni contraintes.

Tunely offrira également aux artistes la possibilité d'ajouter facilement leurs compositions musicales, albums ou projets sur la plateforme, assortie d'options permettant d'en ajouter ou d'en supprimer.

V. Cahier des charges

Pour le cahier des charges, j'ai utilisé la méthode pour structurer et analyser un projet en posant les bonnes questions dans différents domaines clés.
QQOQCP est un acronyme dont chaque lettre correspond à une question : qui, quoi, où, quand, comment et pourquoi, cette méthode incite donc à un questionnement large et constructif.

1. Qui ?

Les utilisateurs ciblés sont des amateurs de musique de tous âges, recherchant un accès facile à une grande variété de styles musicaux et voulant découvrir de nouveaux artistes.

Les artistes quant à eux, seront des musiciens ou des groupes de différents genres musicaux, des artistes émergents aux artistes établis, cherchant à accroître leur visibilité et à toucher un public plus large.

2. Quoi ?

Le site de streaming musical proposera une vaste collection de chansons, d'albums et de playlists de divers artistes.

Les principales fonctionnalités du site incluront la recherche de musique, la possibilité de créer des playlists, de recommander de la musique en fonction des préférences de l'utilisateur, de diffuser des médias et de partager des pistes avec d'autres utilisateurs.

L'objectif principal du site sera de fournir aux auditeurs une plateforme conviviale et accessible pour découvrir, écouter et partager de la musique de manière légale et gratuite.

3. Où ?

Le service de streaming de musique en ligne sera accessible via un site web responsive afin qui s'adapte à tout type de format, offrant une expérience utilisateur cohérente sur différentes plateformes, telles que iOS et Android.

Les bases de données de musique et les fichiers audio seront hébergés sur des serveurs sécurisés, probablement dans un centre de données cloud pour assurer une disponibilité optimale.

La plateforme sera accessible à l'échelle mondiale prochainement, permettant aux utilisateurs de différents pays d'y accéder.

4. Quand ?

Le développement du site de streaming de musique commencera au début de 2023, et la mise en ligne est prévue pour fin 2023.

Les mises à jour et les améliorations du site seront effectuées régulièrement sur 2 ans pour garantir une expérience utilisateur optimale.

5. Comment ?

La création de Tunely sera développée grâce à un framework.
Contrainte technique

- Base de données MySql
- Il devra avoir une gestion de la bande passante pour transmettre les données audio de manière fluide.
- Il est nécessaire d'avoir une infrastructure de stockage suffisamment évolutive pour héberger une vaste bibliothèque musicale.
- De la sécurité afin d'éviter toutes attaques ou fuite de données comme des informations personnelles.

Moyens personnelles:

- Il y a un seul développeur pour ce projet, Randy Bourdon.

6. Pourquoi ?

Tunely est nécessaire pour répondre à la demande croissante d'accès à la musique en ligne.
Elle fournira aux utilisateurs une expérience d'écoute pratique et personnalisée.

Les utilisateurs devraient choisir Tunely pour sa vaste bibliothèque musicale, ses fonctionnalités conviviales et sa capacité à recommander de la musique adaptée à leurs goûts.

Les artistes devraient choisir de publier leur musique sur ce site pour atteindre une audience plus large, promouvoir leur travail et ainsi augmenter leur popularité.

7. Garantir l'accessibilité et l'écoconception

a. L'accessibilité pour les utilisateur en situation d'handicape

Il est important aussi de garantir l'accessibilité des utilisateurs en situation de handicap. Pour cela, Tunely devra respecter le référentiel général d'améliorations de l'accessibilité (RGAA).
Le RGAA est un ensemble de règles et de critères techniques visant à garantir l'accessibilité des sites web sur internet pour les personnes en situation de handicap, développé en France, il est largement reconnu comme un ensemble de normes et de bonnes pratiques pour améliorer l'accessibilité numérique.

b. L'écoconception

L'écoconception fait référence à la conception et au développement de services numériques visant à réduire leur impact environnemental au long de leur cycle de vie.
Elle a pour objectif de minimiser la consommation en ressource ainsi que les émissions de gaz à effet de serre associées.

c. W3C

L'organisme de normalisation W3C (World Wide Web Consortium) a conçu des standards afin de garantir l'interopérabilité, l'accessibilité et l'évolutivité des technologies du web.
Respecter les normes W3C permet un référencement naturel donc avoir un site web mieux référencé ainsi qu'une meilleure visibilité.
W3C a aussi des recommandations pour garantir que les contenus web soient accessibles aux personnes ayant différents types de handicaps.

VI. Choix des solutions

1. Benchmark des frameworks

Pour ce projet, j'ai dû choisir entre plusieurs frameworks à ma disposition. Le choix se fait par rapport à plusieurs critères comme la popularité du framework, plus le framework est populaire et plus il y a de chance que si j'ai un problème durant le développement je puisse trouver une solution, cela rejoint la documentation du framework.
D'autres facteurs rentrent en compte dans le choix du framework, c'est pour ça que j'ai décidé d'effectuer un tableau benchmark (tableau de référence en français) qui permet de comparer différents sujets, dans ce cas, différents framework sur différents critères.

Critères	React	Symfony	Angular	Vue.js
Informations	- Librairie javascript Open-Source développé par Facebook en 2011 - Américain	- Développé par Symfony en 2005 - Écrit en PHP - Français	- Publié en 2016 par Google - Open source, basé sur TypeScript	- Créé en 2013 par Evan You - Framework JavaScript open-source
Popularité	- Très grande popularité - Utilisé par de nombreuses entreprises et développeurs	- Popularité élevée dans la communauté des développeurs - Cela se traduit par une grande quantité de ressources disponibles	- Framework très populaire	- Framework populaire
Courbe d'apprentissage	- React a une courbe d'apprentissage moyenne - Besoin d'un expérience en développement web	- Symfony présente une courbe d'apprentissage moyenne	- Angular a une courbe d'apprentissage rude - Certaines syntaxes exclusives	- Vue.Js a une courbe d'apprentissage douce - Syntaxe intuitive
Performances	- Réputé pour ses performances élevées - Applications réactives et rapides	- Bonnes performances grâce à son architecture optimisée	- Angular à des bonnes performances	- Vue.js offre de bonnes performances avec son système de rendu réactif et sa gestion du DOM
Flexibilité	- Utilisé pour des projets de toutes	- Symfony est un framework	- Angular offre une grande	- Vue offre une flexibilité

Critères	React	Symfony	Angular	Vue.js
	tailles - Compatible avec d'autres bibliothèques ou frameworks	modulaire, qui peuvent être sélectionnés et utilisés indépendamment	flexibilité avec une architecture modulaire	
Documentation	- Documentation complète et de haute qualité - Disponible en Français	- Excellente documentation, de haute qualité - Disponible en Français	- Documentation détaillée -Uniquement en anglais	- Documentation détaillée - Disponible en Français

Tableau comparatif sur GitHub (18/07/2023) :

	React	Symfony	Angular	Vue.js
Étoiles sur GitHub	211K	28K	89.3K	204K

Pour conclure le choix des solutions, j'ai donc pris en compte toutes les différences de ces framework ainsi que mes compétences informatiques et c'est pourquoi j'ai décidé d'utiliser Symfony.

Symfony est un framework écrit en PHP (Hypertext Preprocessor), basé sur le **pattern MVC** (Model, View, Controller), créé en 2005 par Symfony SAS en France.
Il est conçu pour tout type de projet, que ce soit un site web complet, ou une application PHP plus simple.

De plus, Symfony utilise Doctrine, une bibliothèque ORM (Object-Relational Mapping), qui facilite la communication entre les objets de l'application et la base de données relationnelle. Au lieu d'écrire des requêtes SQL directement, on utilise des entités PHP pour représenter les tables de la base de données.

Il existe deux types de pattern d'ORM :
- Le Active Record
- Le Data Mapper

Symfony utilise le Data Mapper.

Pour la manipulation des objets, j'utilise l'Entity Manager, qui permet la gestion des entités et de leurs états.

VII. Organisation du projet

La gestion de projet consiste à planifier, organiser et coordonner afin de mener à bien un projet.
Il existe plusieurs méthodes de gestion de projet, avec des avantages, des inconvénients et des domaines d'application spécifiques.
Pour cela, j'ai aussi créé un benchmark afin de comparer les différentes méthodes de gestion de projet.

1. Benchmark des méthodes de gestion de projet

Critères	Agile	Cycle en V	Kanban	Scrum
Avantages	- Flexibilité - Collaboration et implication des parties prenantes - Livraisons itératives	- Tests et validations approfondis - Structurée - Anticipe tous les besoins et les spécifications nécessaires - Documentation complète	- Flexibilité - Visibilité en temps réel - Adaptée aux flux de travail continus	- Flexibilité - Collaboration et implication des parties prenantes
Inconvénients	- Moins de documentation formelle - Peut être difficile à gérer pour les projets de grande envergure ou avec des équipes distribuées	- Moins adaptée aux changements fréquents - Peut entraîner des retards si des problèmes sont détectés tardivement dans le processus	- Peut manquer de structure pour les projets de grande envergure ou complexes - Nécessite une culture d'amélioration continue pour tirer pleinement parti de la méthode - Peut nécessiter une période d'ajustement et de transition pour les équipes habituées à d'autres méthodes de gestion	- Peut nécessiter une gestion stricte des rôles et des cérémonies pour réussir - Peut être difficile à maîtriser pour les équipes novices

Pour ce projet, j'ai décidé d'appliquer la méthode "**Cycle en V**" pour la gestion de mon projet car je trouve qu'il est le plus adapté.
De plus, durant ce projet, je serais seul comme développeur donc c'est une raison de plus pour choisir la méthode cycle en V.

1. La méthode du cycle en V

a. Histoire du cycle en V

Le cycle en V est un mode de gestion de projet qui a été développé dans les années 1980 et appliqué au champ des projets industriels.
Il découle du modèle en cascade théorisé (waterfall model) présenté par Walker Royce dans les années 1970, et qui permet de représenter des processus de développement de manière linéaire et en phases successives.

b. Le cycle en V

Le cycle en V est une séquence de plusieurs phases constituée de plusieurs étapes.

- Une phase descendante qui va de l'analyse des besoins jusqu'à la programmation et le développement de l'application
- Une phase ascendante qui elle enchaîne les étapes de vérifications.

Les deux phases forment un V, ce qui donne son nom à cette méthode.

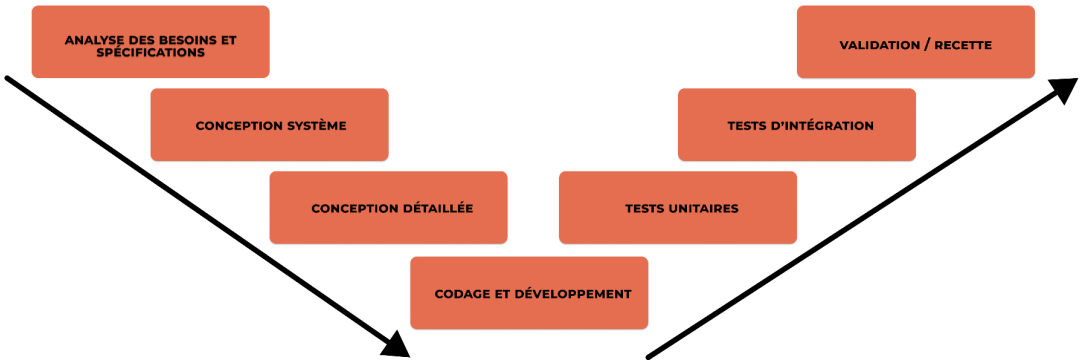


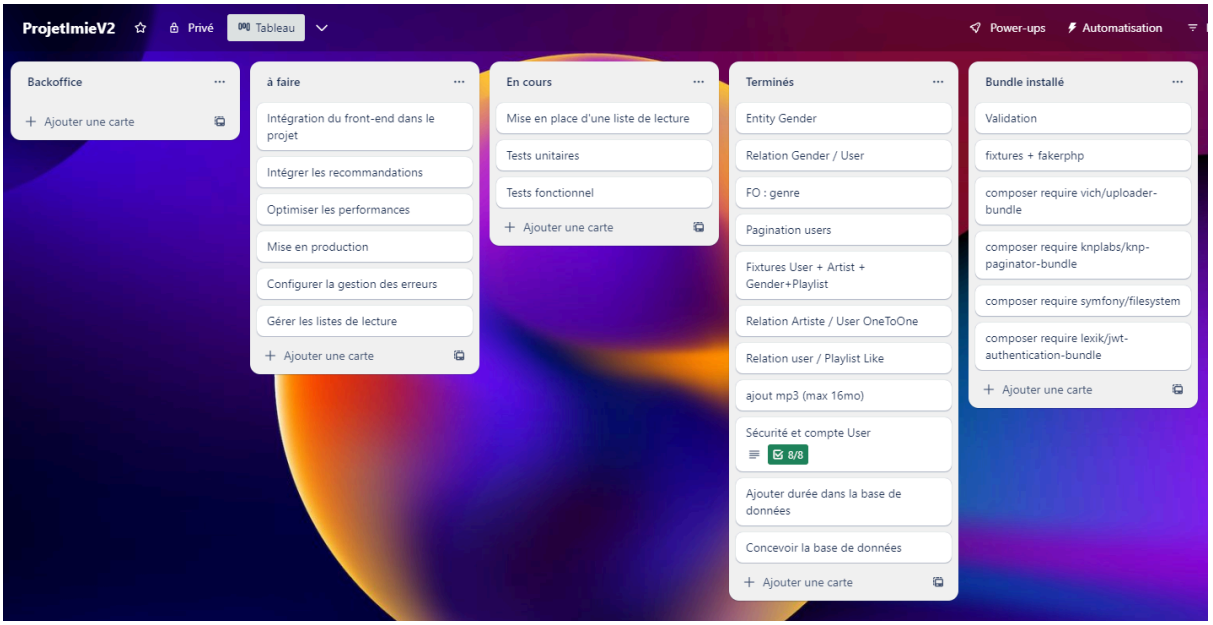
Schéma des phases du Cycle en V

2. Outils d'organisation

J'ai utilisé aussi Trello pour l'organisation du projet.

C'est un outil permettant de gérer n'importe quel type de projet avec une interface intuitive.
Il utilise un système de cartes et de listes qui permet de visualiser facilement l'état d'avancement des tâches et qui permet la collaboration entre membres d'équipe.

J'ai aussi utilisé Notion, développé par Notion Labs, qui m'a permis la prise de note rapide.



Exemple de projet Trello

3. Outil de versionning

Enfin, pour gérer le versioning (versionnage en français), ce qui permet de faire des sauvegardes à un instant donné du projet afin de garder une trace ou au cas où en cas de problème dans le projet.

Il est possible de revenir à une ancienne version du projet au cas où il le faut.

Pour cela j'utilise Git, c'est un logiciel gratuit qui fait de la gestion de versions décentralisé.

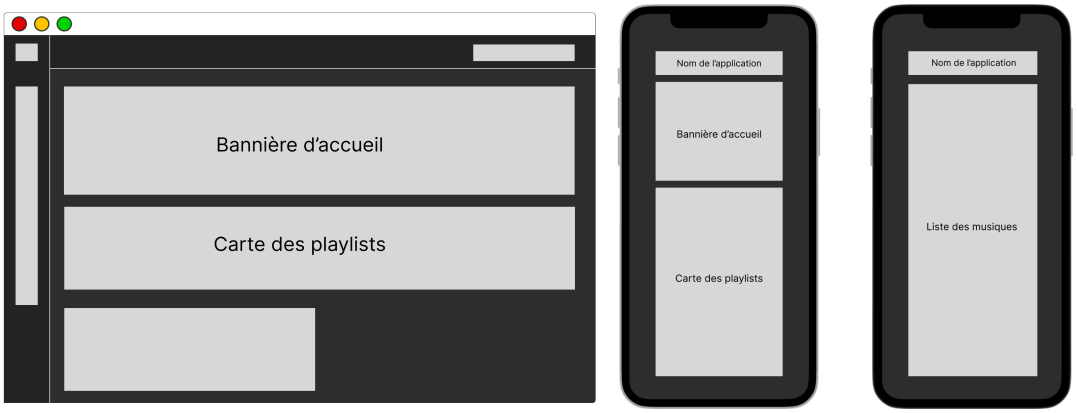
Depuis 2010, il est le logiciel le plus populaire dans le développement logiciel et web,avec plus de dizaines de millions d'utilisateurs.

VIII. Conception graphique

La conception graphique s'est faite avec Figma qui est un outil de design graphique gratuit permettant le travail collaboratif, simple et intuitif.
Il existe d'autres outils comme Adobe XD qui demande un abonnement et qui est un peu plus complexe en termes de prise en main.

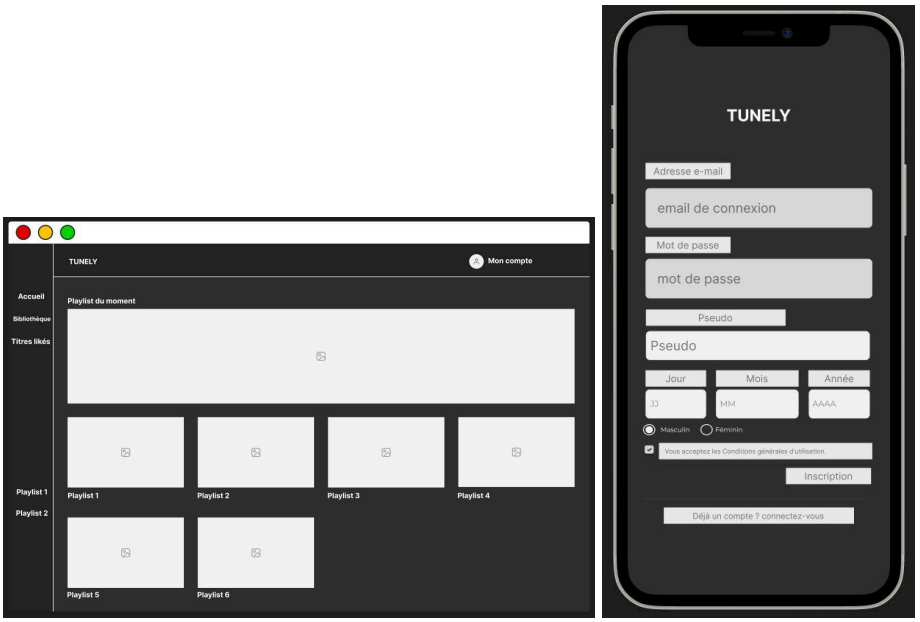
1. Zoning

Le zoning est une technique utilisée pour déterminer la structure des pages d'un site web. Cela consiste en la création de schémas simples qui définissent l'emplacement des principaux éléments de la page web, comme le menu, le logo, les visuels ou la barre latérale.



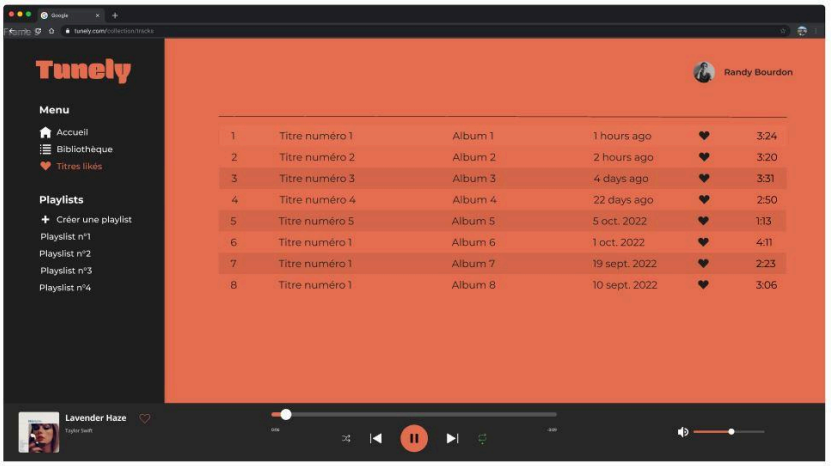
2. Wireframe

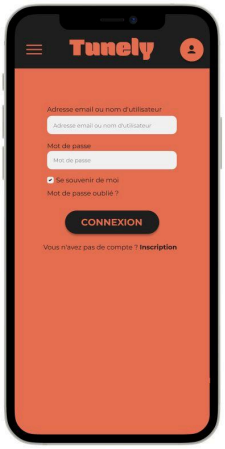
Un wireframe est une représentation visuelle simplifiée et schématique de l'interface. Il permet de définir la structure, l'agencement et les fonctionnalités de manière conceptuelle, sans prendre en compte les aspects esthétiques.

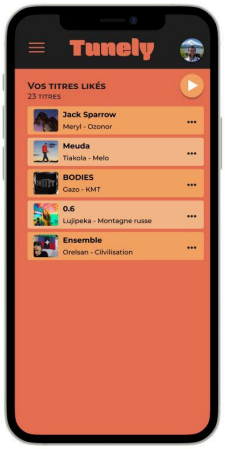


3. Maquette graphique

La maquette graphique est une représentation visuelle statique de l'interface utilisateur finale.
Elle met en évidence les éléments visuels tels que les couleurs, les typographies, les images, les icônes et la disposition des éléments. La maquette graphique permet aux clients de visualiser et d'évaluer l'apparence esthétique de l'interface avant sa mise en production.
Pour cette maquette, je l'ai mise au format mobile car il est important de respecter le responsive.





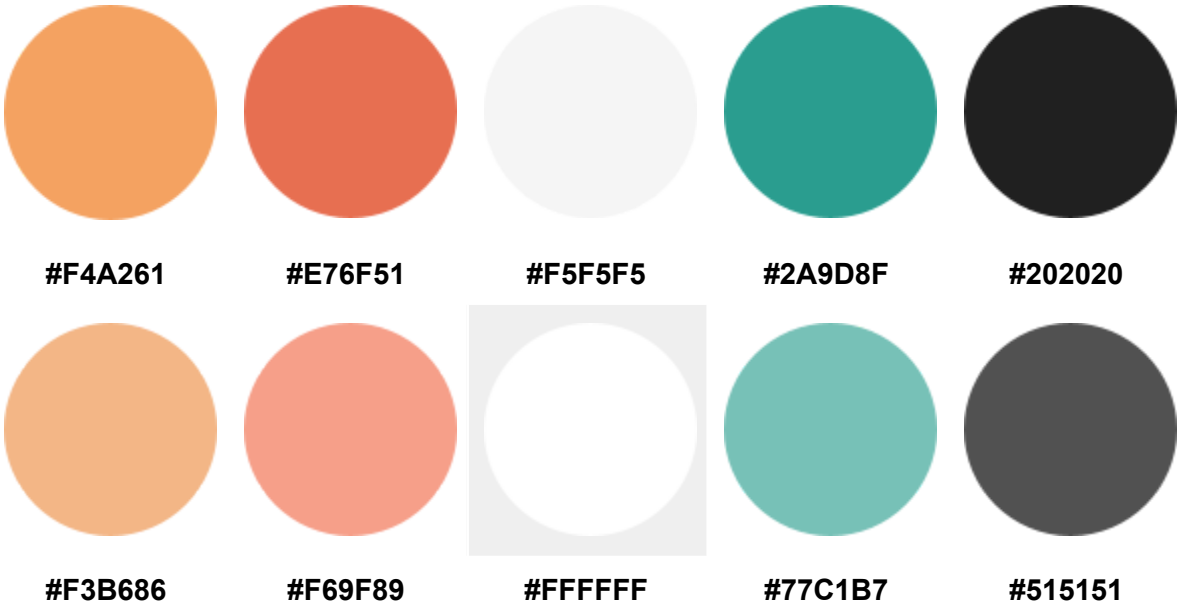


4. Charte graphique

La charte graphique est un ensemble de règles visuelles qui définit l'identité visuelle d'une entreprise ou d'une marque. Elle assure la cohérence des éléments visuels tels que les couleurs, les typographies, les logos, et garantit une image de marque professionnelle et reconnaissable. La charte graphique facilite la création de supports de communication et aide à se différencier des concurrents.

a. Les références couleurs

Pour habiller ma plateforme, j'ai choisi de sélectionner 5 couleurs pour ma palette principale, ainsi que 5 sous-tons, c'est-à-dire 5 couleurs similaires mais un peu plus claires.



b. La police

Pour la police d'écriture, j'ai décidé de choisir la police Montserrat, car c'est une police d'apparence moderne, sans empattement.
Il y a deux épaisseurs de police, une épaisseur moyenne et une épaisseur grasse.

Montserrat Medium
Montserrat Bold

c. Taille de la police

J'ai choisi des tailles de police d'écriture pour les différents types de textes.

Type de texte	Taille de la police	Epaisseur
Titre	24px	Bold
Sous-titre	18px	Medium
Corps de page	14px	Medium

d. Responsive design

Le responsive design est une approche de conception de sites web et d'applications qui vise à offrir une expérience utilisateur optimale sur différents appareils.

Il consiste à créer des interfaces flexibles qui s'adaptent automatiquement à la taille de l'écran utilisé, offrant ainsi une navigation fluide et une lisibilité optimale du contenu.
Le responsive design permet d'améliorer l'accessibilité et la convivialité, tout en évitant la nécessité de développer des versions distinctes pour chaque appareil.

Il existe aussi une autre approche, le "**Mobile first**", qui place le format mobile au centre de la conception, c'est-à dire que je vais me concentrer sur conception puis développement en format téléphone car de plus en plus de personne utilise le téléphone à l'ordinateur.

IX. Conception de la base de données

1. Méthode merise

La méthode Merise est une méthode de conception de base de données, de développement et de réalisation de projets informatiques.
On utilise des diagrammes pour comprendre comment les données circulent et comment les différentes parties du système interagissent. Cela m'aide à planifier et à développer le Tunely étape par étape.
Cela permet de créer le modèle conceptuel de données et modèle logique de données.

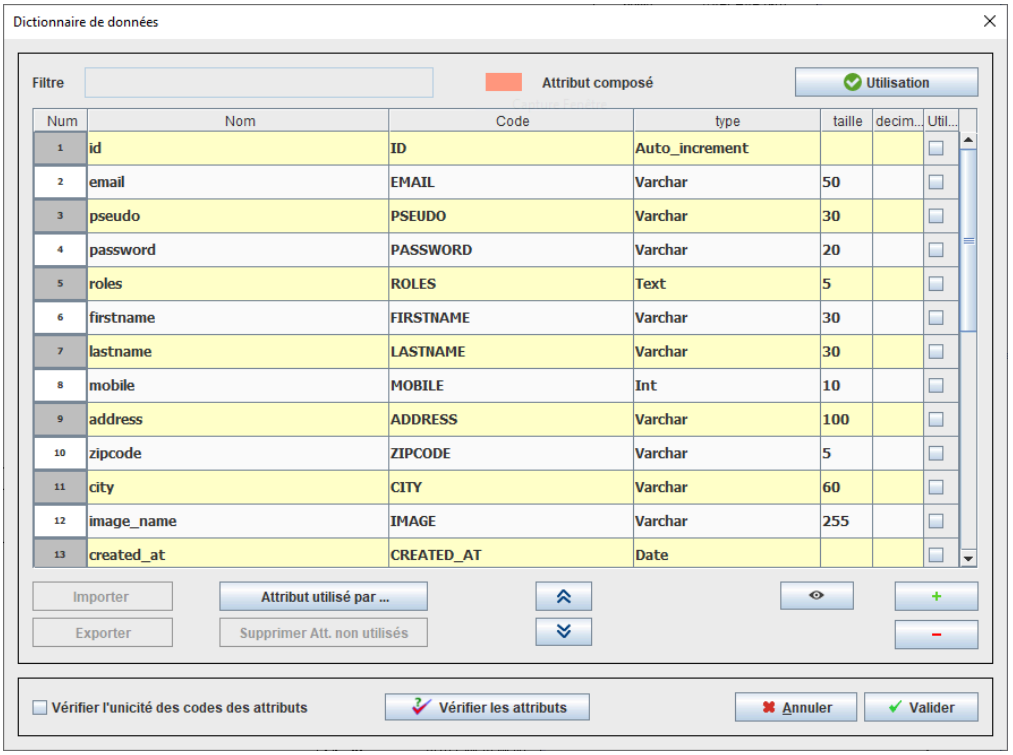
a. Pourquoi choisir la méthode merise ?

La méthode Merise est appréciée car elle aide à organiser la réflexion et à bien comprendre comment tout s'emboîte. En suivant cette méthode, on peut mieux documenter nos décisions et faciliter la maintenance du système à l'avenir.

Pour ce projet, j'ai décidé d'utiliser l'application JMerise afin de concevoir le **Modèle conceptuel de données** (MCD) et le **Modèle logique de données** (MLD)

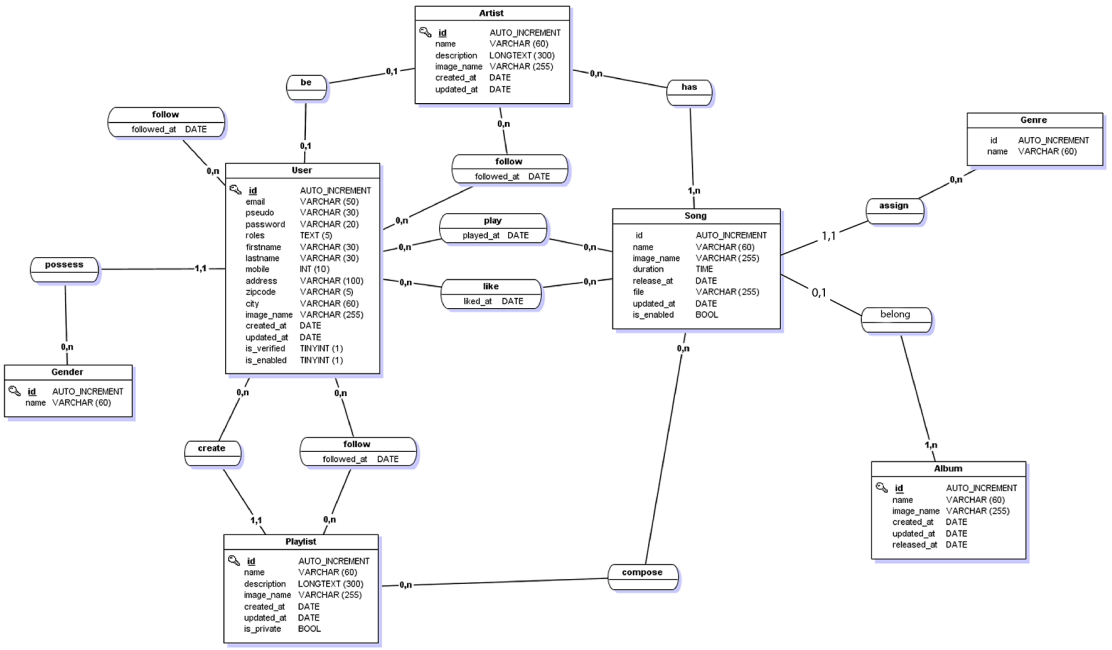
2. Dictionnaire de données

Le dictionnaire de données dans la méthode Merise est un élément clé de la conception, il permet de faciliter et d'accélérer la création des attributs des entités et des associations.



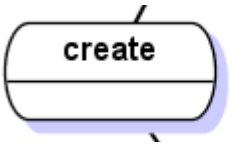
3. Diagramme MCD

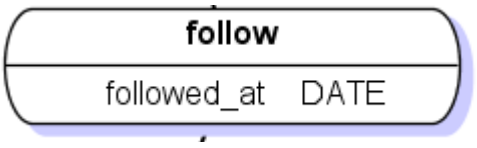
Le MCD est une représentation abstraite des entités, de leurs relations. Il permet de comprendre le système, de savoir comment les différents éléments sont liés entre eux, de détecter les erreurs.
Le MCD est un outil crucial pour concevoir et développer efficacement des systèmes d'information.



a. Les entités
Elles sont constituées d'un nom et d'une liste d'attributs, où chaque attribut est caractérisé par un type et peut servir d'identifiant pour l'entité.

b. Les associations
Les associations permettent de lier les entités entre elles.
En général, le nom des associations est un verbe définissant le lien entre les entités.



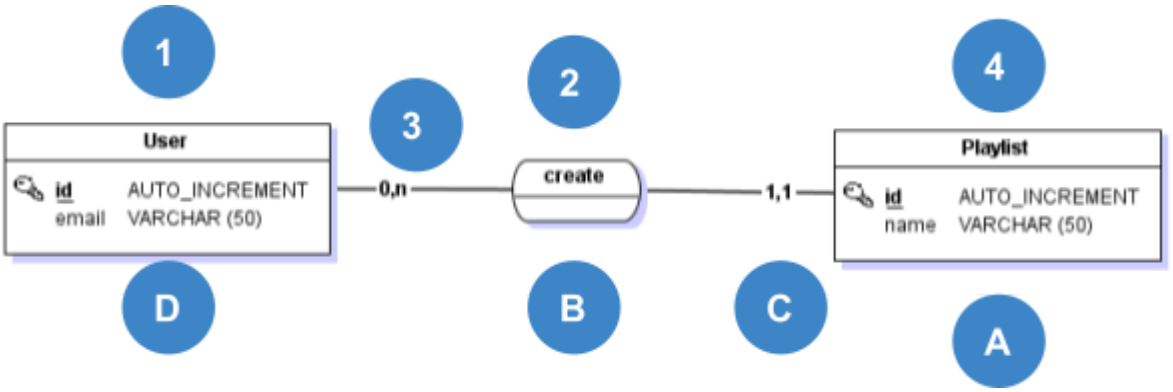


c. Liens et cardinalités

La connexion entre les entités et les relations se fait grâce à des liens porteurs de cardinalité.

Les cardinalités fournissent des informations précieuses et jouent un rôle essentiel dans la construction ultérieure de la base de données avec la création de tables intermédiaires ou de clés étrangères.

La lecture des cardinalités se fait comme ça :



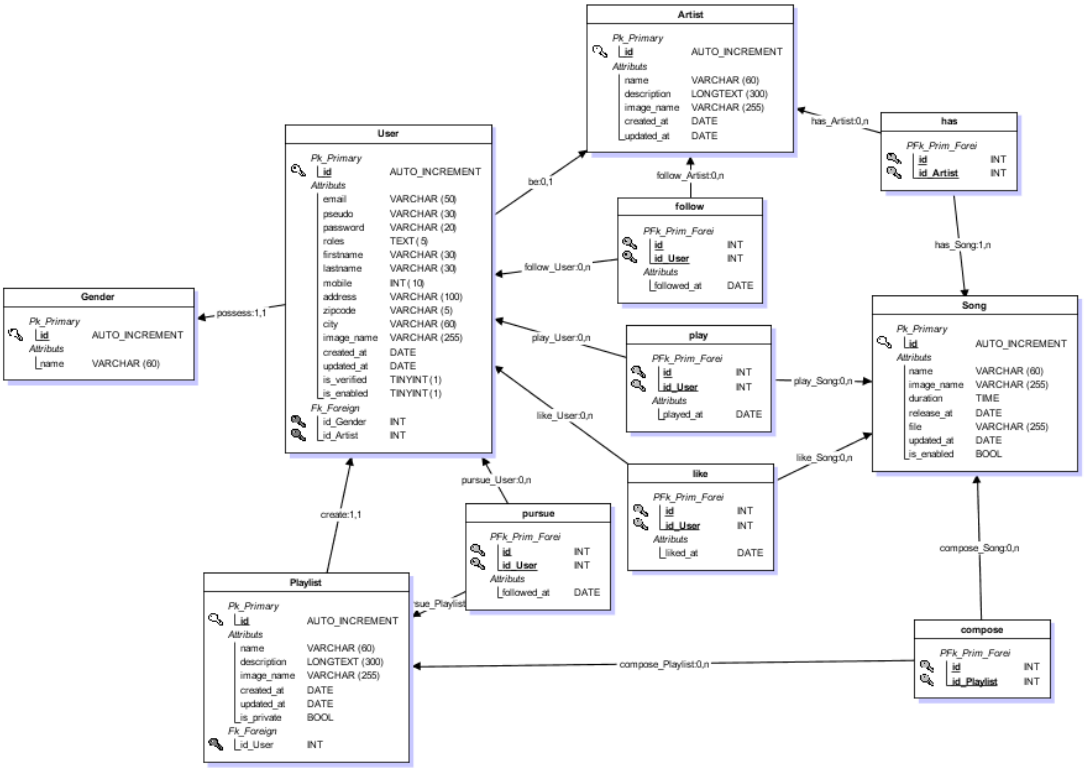
En utilisant 1, 2, 3, 4 : Tout User peut créer aucune ou plusieurs Playlist
En utilisant A, B, C, D : Toute Playlist est créer par un et un seul User

4. Diagramme MLD

Le modèle logique de données (MLD) permet d'implémenter la base de données en transformant le MCD en instructions SQL, il représente de manière détaillée la structure des données, les relations et les contraintes d'une base de données.
Les relations du MLD sont issues des entités du MCD et de certaines associations.

Au passage du MCD au MLD, plusieurs changements interviennent :

- Une entité du MCD devient une relation, c'est-à-dire une table.
- Son identifiant devient la clé primaire de la relation.
- Les autres propriétés deviennent les attributs de la relation.
- Une association de type 1 : N se traduit par la création d'une clé étrangère dans la relation correspondante à l'entité côté "1".
- Une association de type N : N se traduit par la création d'une table dont la clé primaire est composée des clés étrangères référençant les relations correspondant aux entités liées par l'association.



X. Conception de l'application

Pour concevoir ce projet j'ai utilisé l'UML de l'anglais Unified Modeling Language, c'est un langage de modélisation graphique pour les domaines du développement logiciel et en conception orientée objet.

Mais pourquoi modéliser ?

La modélisation est une étape importante avant la réalisation d'un projet, cela permet de mieux comprendre le fonctionnement du système, il est aussi important d'utiliser un langage commun afin de faciliter si équipe il y a, la communication et la compréhension d'un projet.

1. Les user stories

Pour ce projet, j'ai décidé d'utiliser les user stories malgré qu'elles soient issues de la méthode Agile, j'ai repris ce format afin que cela soit plus clair avec les besoins du projet.

Les user stories sont introduites pour la première fois dans les années 90 grâce à Kent Beck et Ward Cunningham, en collaboration avec Ron Jeffries et d'autres contributeurs du mouvement Agile.

La rédaction des user stories est importante pour exposer les besoins des utilisateurs et des fonctionnalités du projet.
Les User stories doivent être compréhensibles, concises et décrivent la valeur qu'elles apportent à l'utilisateur.

Afin de bien réaliser des User stories, il faut :

- Utiliser un format standard comme " En tant que [utilisateur], je veux [fonctionnalité], afin de [bénéfice ou objectif] "
- Être précis
- Se concentrer sur les utilisateurs et leurs besoins
- Eviter les User stories trop grandes

User stories

EN TANT QUE	JE VEUX	AFIN DE
Visiteur	Consulter un artiste/album	Découvrir un artiste/album
Visiteur	M'inscrire	Devenir un utilisateur
Visiteur	M'authentifier	Pouvoir écouter de la musique
Visiteur	Réinitialiser mon mot de passe	De se connecter
Utilisateur	Me déconnecter	Ne pas laisser mon session ouverte
Utilisateur	Gérer mon profil	Modifier mon pseudo
Utilisateur	Gérer mes playlists	Modifier des playlists
Utilisateur	Ecouter des musiques	Passer un bon moment
Utilisateur	Ajouter à mes musiques sauvegardés	Retrouver mes titres likés
Utilisateur	Ajouter des musiques dans une liste d'écoute	Pouvoir enchainer des musiques pendant mon écoute
Utilisateur	Rechercher un titre selon son titre/artiste/style	Trouver une chanson
Utilisateur	S'abonner à des artistes/playlists/utilisateurs	Ne pas rater des sorties de musique
Artiste	Gérer mon profil artiste	Pouvoir modifier mon profil artiste
Artiste	Gérer mes musiques	Pouvoir ajouter mes musiques
Artiste	Gérer mes albums	Pouvoir modifier mon album
Administrateur	Gérer les musiques	Pouvoir modifier les musiques

Les user stories du projet

2. Quel logiciel pour les diagrammes en UML ?

Pour la conception des diagrammes UML, j'ai décidé d'utiliser le logiciel StarUML.
Cet outil est dédié à la modélisation UML, créé par la société coréenne MKLabs, en 2006.

Il offre des fonctionnalités de modélisation orientée objet et permet de créer des diagrammes comme le diagramme de classe, le diagramme de cas d'utilisation ou encore le diagramme de séquence.

3. Le diagramme de cas d'utilisation (Use case)

Le diagramme de cas d'utilisation représente la structure des principales fonctions requises par les utilisateurs du système. C'est le premier diagramme du modèle UML, assurant la relation entre l'utilisateur et les objets mis en œuvre par le système.

L'objectif du diagramme de cas d'utilisation est de fournir une vue globale des interactions entre les acteurs et le système afin de comprendre les fonctionnalités du système et les rôles des différents acteurs.

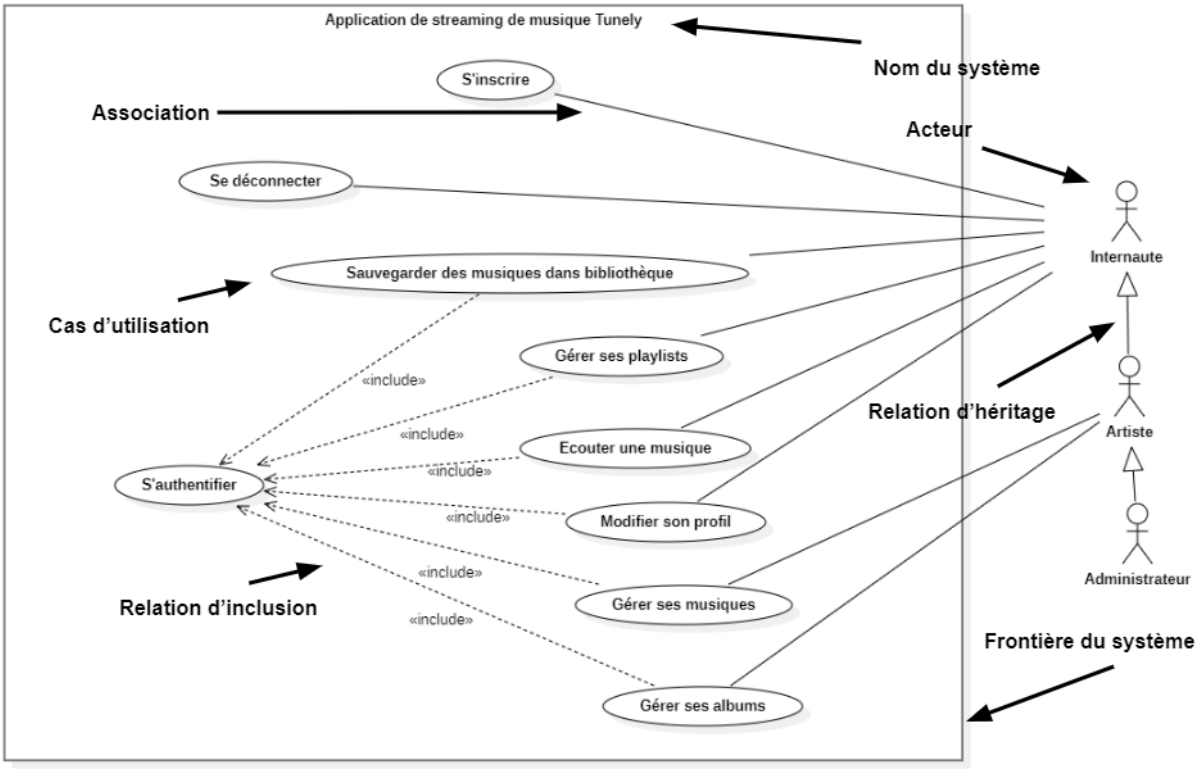


Diagramme de cas d'utilisation de Tunely du front-office

Il y a plusieurs éléments clés dans un use case comme :

- Le système : Qui représente le système à produire et regroupe les différents cas d'utilisation

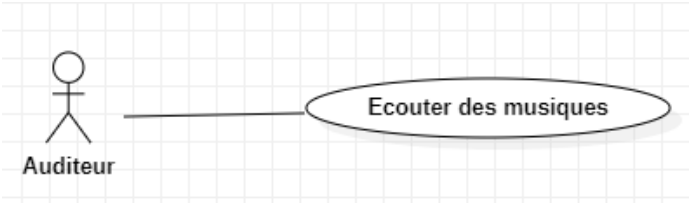
- Les acteurs : Représente un rôle externe qui interagit avec le système. Un acteur peut être un utilisateur humain, un autre système logiciel, un appareil matériel, ou toute entité externe qui communique avec le système étudié. Ils peuvent être liés entre eux par un héritage.

- Les cas d'utilisations : Représente une fonctionnalité ou une action spécifique offerte par le système. Chaque cas d'utilisation est lié à un ou plusieurs acteurs et décrit une interaction spécifique entre eux et le système.

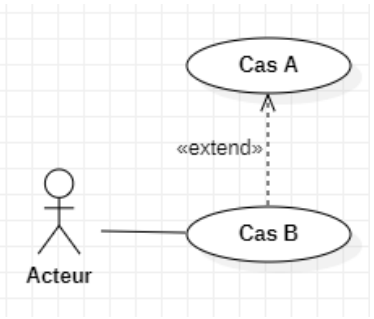
Ils sont représentés par un verbe à l'infinitif et complément comme “Écouter une musique”

- Les relations :

i. Les associations : Représente une interaction bidirectionnelle qui peut relier plusieurs acteurs simultanément, elle ne donne pas d'information temporelle et ne précise pas d'éventuelle correspondance.

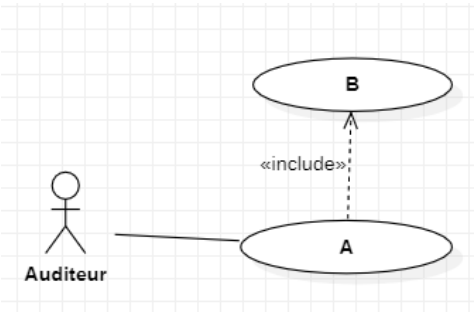


ii. Les extensions : Représente la relation par des pointillés entre deux cas d'utilisation, le cas d'utilisation A étend un cas d'utilisation B.
Cette dépendance est représentée par un “<<extend>>”



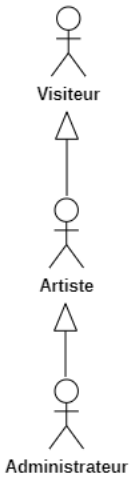
iii. L'inclusion : Cette relation représente une dépendance entre deux cas d'utilisation.

Lorsque A est sollicité, B l'est obligatoirement, elle est représentée par un <<include>>
Cela permet de factoriser un e partie d'un cas d'utilisation qui serait commune à d'autres cas d'utilisation



iv. L'Héritage / Généralisation : Elle est utilisée lorsque deux cas d'utilisation partagent un ensemble de fonctionnalités communes, mais l'un est plus spécifique que l'autre.

Ici, l'artiste hérite des fonctionnalités du visiteur, et l'administrateur hérite des fonctionnalités de l'artiste, il peut donc ajouter ou modifier des musiques.



4. Le diagramme de classes

Le diagramme de classe représente la structure interne du projet.
Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation.
Il montre également les classes du système, leurs attributs, leurs méthodes et les relations entre les classes.

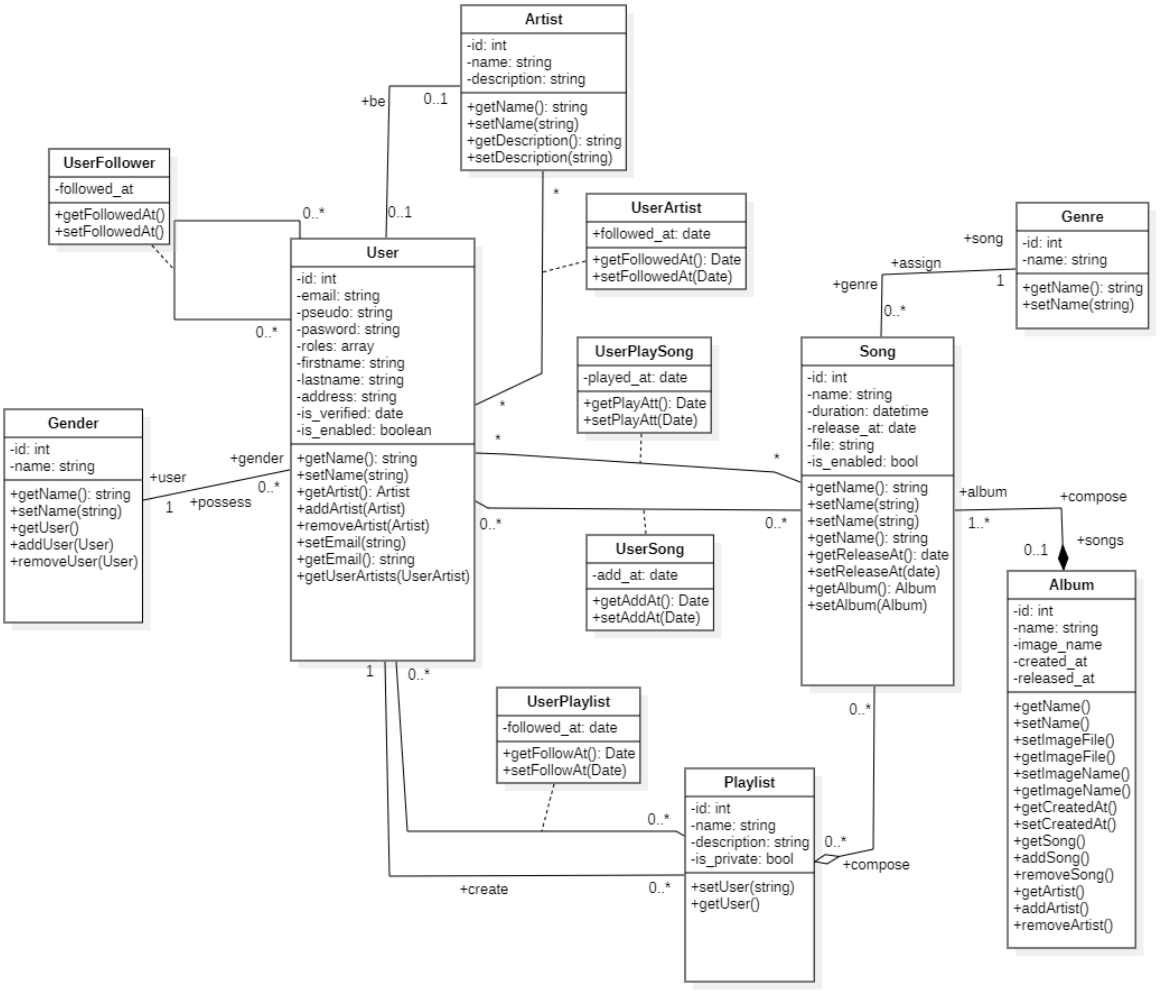
Comme pour le diagramme de cas d'utilisation, il existe plusieurs éléments clés :

- La **classe** :
  Représentée par un rectangle divisé en plusieurs parties. La première section contient le nom de la classe, la deuxième contient les attributs de la classe (variables), et la troisième contient les opérations (méthodes) de la classe.

- Les **attributs** : Ce sont des variables définies au sein de la classe et représentation des caractéristiques ou des données que les objets de cette classe possèdent.
  Ils sont généralement affichés sous la forme de paires nom:type, où "nom" est le nom de l'attribut et "type" est le type de données de l'attribut.

    - Visibilité des attributs :
        - Le + indique que la propriété est public
        - Le - indique que la propriété est privée
        - Le # indique que la propriété est protégée
        - Le ~ indique que la propriété est issue d'un package
        - Le / indique que la propriété est dérivée (c'est à dire calculé à partir d'une autre propriété)
        - Le ^ indique que la propriété est hérité

- Les **opérations** / méthodes : Les opérations sont les fonctions ou les actions que la classe peut effectuer.
  Quand le nom d'une opération apparaît plusieurs fois avec des paramètres différents, on dit que l'opération est surchargée. En revanche, il est impossible que deux opérations ne se distinguent que par leur valeur retournée.

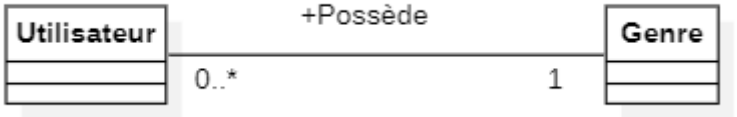


Mon diagramme de classe

Les **relations** : Pour relier les différentes classes d'un diagramme de classe, il est possible d'utiliser plusieurs types de relations :

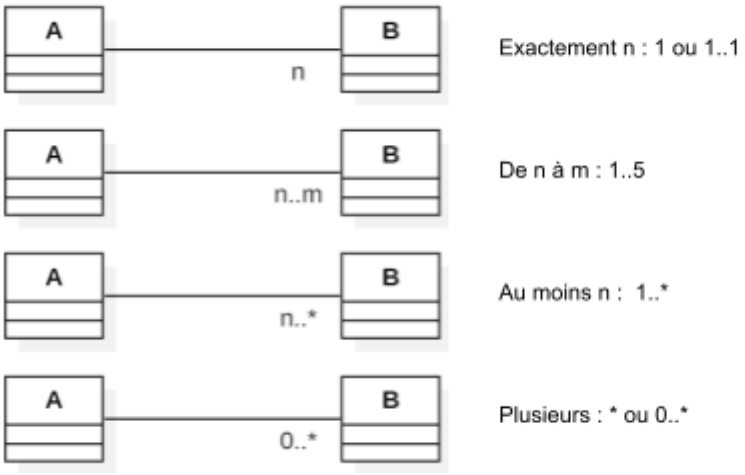
- **L'association** : Elle représente les relations entre les classes, indiquant comment les objets de ces classes sont liés les uns aux autres

  Chaque association possède un nom, à chaque extrémité de l'association se trouve un "**Rôle**" qui nomme l'extrémité de l'association et une **multiplicité**, qui spécifie combien d'objets d'une classe sont liés à un objet de l'autre classe.



Exemple d'association

Détail des multiplicité
Nombre d'objets de la classe B associés à un objet de la classe A, voici quelques exemples :



Pour définir les cardinalités, on peut se poser les question par exemple :
- Combien A à au minimum de B ?
- Combien A à au maximum de B ?

- **Agrégation** :
  Cela représente la relation entre deux classes dissymétriques (une classe prédominante sur l'autre).

  Il existe deux types d'agrégation :

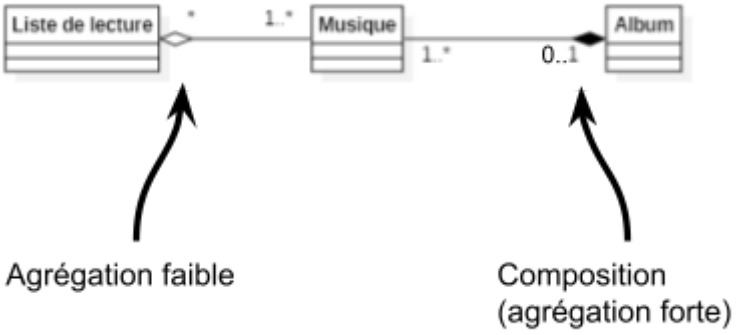
- **Agrégation faible / Agrégation par référence** : Représenté par un losange creux du côté du plus grand, elle est utilisée quand on souhaite modéliser une relation tout/partie, une classe constitue un élément plus grand (tout) composé d'éléments plus petits (partie)

  Dans l'exemple ci dessous, une liste de lecture est composée d'un ensemble de musique, une musique peut appartenir à plusieurs listes de lectures et supprimer la liste ne supprime pas les musiques contenues dedans

- **Composition** : Appelée aussi agrégation forte, c'est une composition qui est représentée par un losange plein.

  La création ou destruction du composite entraîne la création ou la destruction de ses composants.
  Un objet ne fait partie que d'un composite à la fois

  Dans l'exemple ci dessous, une musique n'appartient qu'à un seul album et la suppression de l'album entraîne la suppression de toutes les musiques de l'album.



Exemple d'agrégation

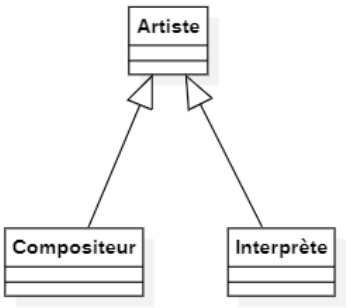
- **Héritage** : l'héritage représente une relation d'héritage entre une classe parent (ou classe de base) et une ou plusieurs classes enfants (ou classes dérivées).

  Il existe quelques propriétés principales de l'héritage :

  - La classe enfant hérite de toutes les caractéristiques (attributs et méthodes) de ses classes parentes, mais elle ne peut pas accéder directement aux caractéristiques privées (attributs et méthodes privées) de ces classes parentes.

  - En héritant d'une classe parente, toutes les associations de cette dernière s'appliquent également aux classes dérivées.

  Elle est représentée par une ligne avec une flèche dirigée de la classe enfant vers la classe parent.



Exemple d'héritage où Compositeur et Interprète hérite de la classe Artiste

XI. Dépendances : C'est un lien qui permet d'indiquer d'une autre classe est utilisée dans les méthodes de sa classe (flèche en pointillé)

XII. Multicouches

1. L'architecture 3-tiers c'est quoi ?

L'architecture 3-tiers est basée sur 3 couches logiques, Présentation, Application et Données.
Cette architecture est très populaire, en particulier pour les applications client-serveur traditionnelles. Au fil du temps, cette architecture permet aux développeurs de créer des applications plus flexibles et évolutives.

a. Couche de présentation

La couche présentation représente l'interface de communication entre l'application et l'utilisateur final. L'objectif de cette couche est d'afficher les données/informations à l'utilisateur et recueillir les données/saisies auprès de lui.

b. Couche logique

Cette couche est le cœur de mon application, elle est aussi appelée niveau logique. Le traitement des informations recueillies dans la couche supérieure est fait à ce niveau avant d'être envoyé au niveau de données.
La logique est décrite/définie par les développeurs de l'application dans le but de transformer les données de la couche application avant de les transmettre en forme de requêtes à la base de données.

c. Couche de données

Le niveau de données, appelé aussi niveau de la base de données.On va trouver toutes les informations traitées par la couche supérieure stockées et gérées. La construction de votre couche base de donnés peut se baser sur une base de données relationnelle (SQL) ou Non relationnelle (NoSQL).

2. Le modèle MVC

Le modèle MVC (Modèle-Vue-Contrôleur) est un modèle de conception inventé fin des années 70 par Trygve Reenskaug.

a. Le fonctionnement :

Le modèle MVC établit une distinction claire entre les données, les opérations de traitement et la présentation. C'est pourquoi l'application est divisée en trois composants essentiels : le modèle, la vue et le contrôleur. Chacun de ces composants joue un rôle spécifique et défini.

- **Le Modèle** : Gère ce qu'on appelle la **logique métier**, contient la gestion des données qui sont stockées ainsi que de la logique en rapport avec les données.

- **La Vue** : La vue contient des éléments visuels ainsi que la logique nécessaire pour **afficher** les données provenant du modèle passant par le contrôleur.

- **Le Contrôleur** : Le contrôleur joue un rôle central dans la **gestion** des actions et des réactions d'une application web. Il est responsable de la réception des requêtes, de l'activation de la logique métier basée sur les objets, et du choix de la vue qui représente la réponse à la requête.

b. Les avantages du MVC

Le modèle MVC comporte plusieurs avantages :
- Séparation des préoccupations
- Du code réutilisable
- Une maintenance facilité
- Testabilité facile

En résumé, le modèle MVC fournit une structure organisée pour le développement d'applications qui favorise la séparation des préoccupations, la réutilisation du code, la maintenabilité, la testabilité, l'évolutivité et une collaboration efficace. Ces avantages permettent de créer des applications plus puissantes, modulaires et gérables

c. Les inconvénients du MVC

Mais le modèle MVC comporte quelques inconvénients :
- Complexité initial
- Surcharge de fichiers
- Gestion de multiples fichiers
- Coût des mise à jour fréquentes

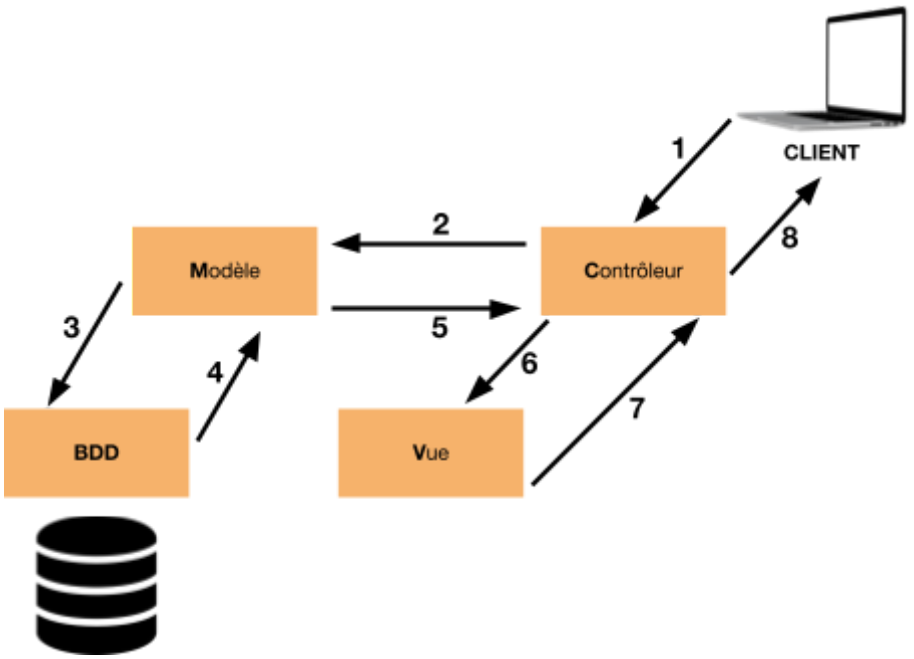


Schéma du MVC

1. Le client souhaite accéder à une url, le Contrôleur réceptionne cette demande
2. Le Contrôleur contrôle la demande et la traite en demandant les données au Modèle
3. Le Modèle fait une requête en SQL à la base de données afin de récupérer les informations nécessaire
4. La base de données renvoie une réponse au Modèle
5. Le Modèle renvoie les données au Contrôleur
6. Le Contrôleur envoie les données à la Vue qui va permettre d'afficher les données ou la réponse demandé
7. La vue retourne la page avec les données
8. Le client reçoit la page demandé

XIII. Sécurité

La sécurité informatique en général est primordiale, car elle permet d'anticiper les vulnérabilités, et de prévenir les futures attaques, comme les fuites de données ou les intrusions dans les systèmes par exemple.

Dans mon cas, la fuite de données est le risque principal et doit impérativement être évité. En effet, les utilisateurs entrent des informations comme des mails ou des données bancaires, qui constituent des données à caractère personnel.

La gestion et la sécurisation de ce type de données est notamment soumise au règlement général sur la protection des données (RGPD).

Pour cela, la mise en place de processus de sécurité est primordiale.

Il faut notamment que je me tienne informé des nouvelles vulnérabilités afin de me préparer, il existe plusieurs sites qui recensent les failles de sécurité, un des plus connus étant :

https://attack.mitre.org/

Ou l'OWASP (Open Web Application Security Project) :

https://owasp.org/

Ils contiennent un catalogue de vulnérabilités, ainsi que d'outils et de ressources pour prévoir et compléter ma stratégie de sécurité.

Je me suis aussi appuyé sur les recommandations et les normes dans le domaine de la sécurité comme l'Agence nationale de la sécurité des systèmes d'information (ANSSI) joue un rôle essentiel dans la sécurité.

Créée en 2009, l'agence dépend du Secrétariat général de la Défense et de la Sécurité nationale (SGDSN) qui fait partie du gouvernement français.

Elle émet des directives et des conseils en matière de sécurité informatique pour aider les organisations à protéger leurs systèmes, leurs données et leurs infrastructures contre les menaces et les risques cybernétiques.

1. Attaque CSRF

a. Une attaque CRSF, c'est quoi ?

Une attaque CSRF (Cross-Site Request Forgery) est une technique d'attaque qui exploite la confiance d'un site Web dans la session authentifiée d'un utilisateur. L'objectif principal des attaques CSRF est d'effectuer des actions inattendues aux utilisateurs authentifiés sur les sites Web auxquels ils font confiance.

b. Comment se protéger de cette attaque ?

Il existe plusieurs méthodes afin de se protéger de l'attaque CRSF :

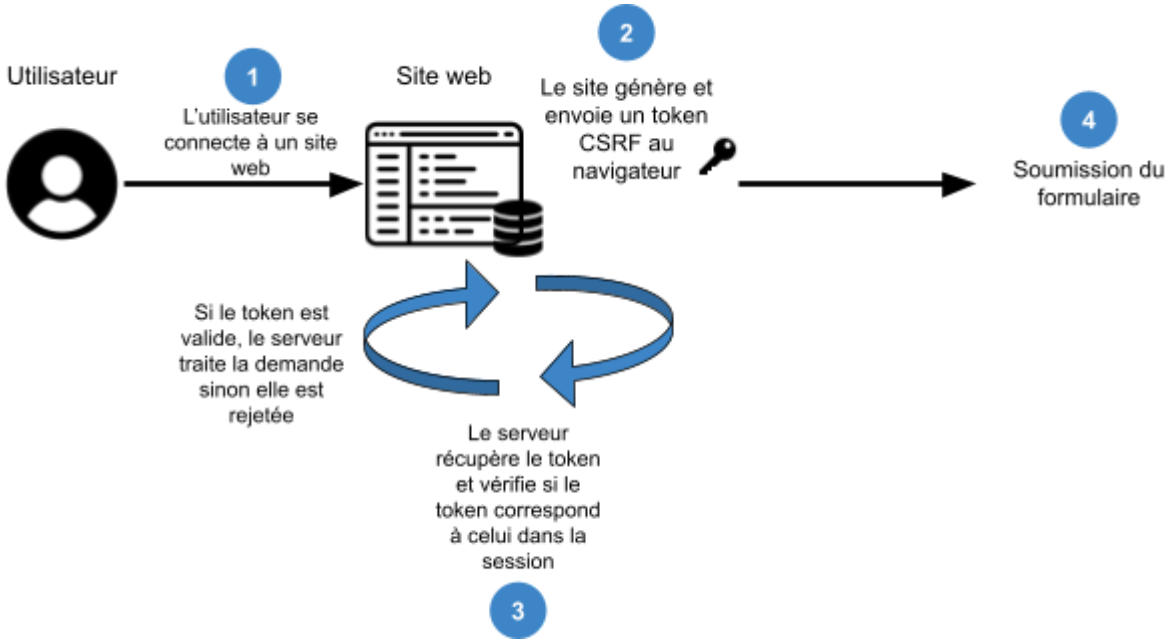
- Cookies SameSite
- Implémentation des tokens anti-CSRF

c. Cookies SameSite

SameSite est un attribut qu'on peut insérer sur les cookies, quand l'attribut est placé sur les cookies de sessions ceux-ci ne seront pas envoyés au serveur si la requête ne provient pas du domaine de l'application.

d. Tokens anti-CSRF

Les jetons CSRF sont des valeurs uniques générées aléatoirement côté serveur et envoyées au client. Ces valeurs étant uniques pour chaque requête, elles permettent de renforcer la sécurité d'une application en rendant difficile (voire impossible) pour un attaquant de les deviner et donc d'exploiter une vulnérabilité CSRF.



2. Injection SQL

a. Une injection SQL, c'est quoi ?

Une injection SQL (Structured Query Language) est une technique d'attaque courante visant à exploiter une vulnérabilité de sécurité présente dans les applications web qui ne filtrent pas correctement les entrées utilisateur avant de les inclure dans des requêtes SQL.

b. Les conséquences ?

La personne malveillante peut, grâce à cette technique, dérober des informations secrètes comme des mots de passe ou informations personnelles.

c. Comment se protéger de cette attaque ?

Il existe plusieurs méthodes afin de se protéger d'une injection SQL par exemple :

i. Les requêtes préparées :

L'utilisation de requêtes préparées permet de séparer les données des instructions SQL, ce qui rend les attaques par injection SQL plus difficiles à exécuter.
Le Query Builder qui est un mécanisme objet proposé par l'ORM Doctrine. C'est le mécanisme à privilégier lors des requêtes dynamiques car la sécurité est assurée par l'ORM.

```php
public function getUsersByUsername(UserRepository $userRepository, $username)
    {
        $users = $userRepository->getUsersByUsername($username);

        return $this->render('user/list.html.twig', ['users' => $users]);
    }
```

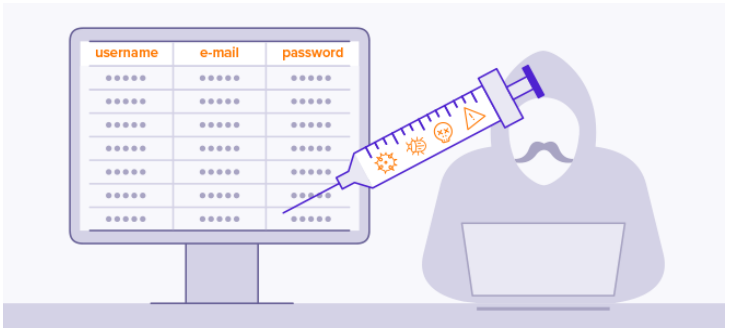
Exemple de méthode getUsersByUsername() dans le contrôleur avec symfony

```php
public function getUsersByUsername($username)
{
    $qb = $this->createQueryBuilder('u')
        ->where('u.username = :username')
        ->setParameter('username', $username);

    return $qb->getQuery()->getResult();
}
```

Implémentation de la méthode pour récupérer les utilisateurs par leur nom d'utilisateur en utilisant une requête préparée.

ii. L'échappement de caractère spéciaux :

Cette méthode permet de prévenir les attaques par injection SQL, elle a pour principe de transformer les caractères spéciaux potentiellement dangereux en une représentation qui est interprétée comme des données littérales plutôt que du code SQL



3. **Faille XSS**
   a. Une faille XSS, c'est quoi ?

      Une faille XSS (Cross-Site Scripting) est une technique d'attaque qui consiste à injecter du code malveillant via les paramètres d'entrée côté client.

   b. Comment se protéger de cette attaque ?

      Il existe plusieurs mesures de sécurité afin de se protéger d'une faille XSS :

      - L'encodage (ou échappement) des données en entrée ou en sortie :

        Afin d'empêcher l'exécution de code malveillant, on peut remplacer les caractères spéciaux par des valeurs encodés pour que toute donnée saisie par l'utilisateur soit interprétée comme du texte et non du code.

        Encodage des données en sortie :

        Avec Twig, le filtre "e" appliqué à cette variable va automatiquement encoder les caractères spéciaux pour les rendre sans danger.

      - Le filtrage des données reçues côté client :

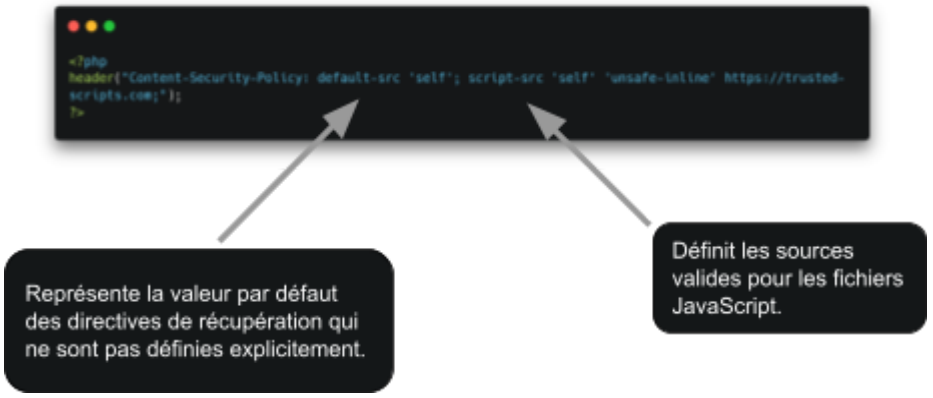
        L'ensemble des données doit passer par un filtre qui élimine les caractères dangereux comme la balise <script>, les gestionnaires d'événements HTML comme onActivate() ou encore le onClick().

```
<script>
        function sanitizeInput(input) {
            // Supprimer les balises <script> et les gestionnaires d'événements HTML
            const sanitizedInput = input.replace(
              /<script\b[^<]*(?:(?!<\/script>)<[^<]*)*<\/script>/gi, ''
            ).replace(/\bon\w+\s*=\s*"[^"]*"/gi, '');

            return sanitizedInput;
        }

        function handleSubmit() {
            const userInput = document.getElementById('userInput').value;
            const sanitizedInput = sanitizeInput(userInput);

            // Le input mis à jour avec la valeur filtrée
            document.getElementById('userInput').value = sanitizedInput;

            // soumission du formulaire
            document.getElementById('myForm').submit();
        }
    </script>
```

Exemple de filtrage en javascript

- Le standard CSP :

  Le standard CSP (Content Security Policy) est un standard de sécurité qui permet de restreindre les sources externes de récupération de données.



Implémentation de CSP

4. Politique de mot de passe

La politique de mot de passe est un ensemble de règles et de directives mises en place pour garantir la sécurité des comptes utilisateurs et des projets informatiques

a. Quelques règles couramment utilisé dans la politique de mot de passe

I. Complexité du mot de passe (Utilisation de lettre, chiffre, caractères spéciaux, majuscules)
II. Longueur minimale (minimum 12 caractères)
III. Non-utilisation de mots courants
IV. Renouvellement régulier
V. Ne pas stocker les mots de passe dans des fichiers texte ou notes

b. Le hachage du mot de passe

Cela consiste a transformé le mot de passe entrée en une séquence de caractères alphanumériques.

Ce processus est conçu pour être irréversible, il n'est donc pas possible de retrouver le mot de passe.

C'est une des pratiques de sécurités basiques qui doit être effectuée car sinon le mot de passe est en clair dans la base de données ce qui représente une faille de sécurité.

```php
public function registration(UserPasswordHasherInterface $passwordHasher): Response
    {

        $user = new User(...);
        $plaintextPassword = ...;

        $hashedPassword = $passwordHasher->hashPassword(
            $user,
            $plaintextPassword
        );
        $user->setPassword($hashedPassword);

        // ...
    }
```

Exemple de fonction qui inscrit un utilisateur et hash le mot de passe

Si l'utilisateur souhaite se connecter, il rentre son mot de passe ainsi que son pseudo, son mot de passe sera hasher et ensuite comparé au hachage présent en base de données. Si les deux hachages sont identiques alors le mot de passe est correct.

Il est possible de “Crypter ” le mot de passe d'un utilisateur, cela consiste à le chiffrer grâce à un algorithme de chiffrement.
Le chiffrement est réversible en utilisant une clé secrète contrairement au hachage ce qui rend cette méthode moins sécurisée.

5. Les attaques bruteforce

Il est important d'utiliser des mots de passe forts, longs et complexes afin de se protéger du “bruteforce” car c'est un type d'attaque où l'attaquant tente de découvrir le mot de passe en essayant toutes les combinaisons possibles jusqu'à ce qu'il trouve la bonne.

6. L'authentification à deux facteurs

Il existe l'authentification à deux facteurs (2FA) qui est une méthode de sécurité qui ajoute une couche supplémentaire de protection lors de l'accès à un compte en ligne ou à un système.
Au lieu de se fier uniquement à un mot de passe, le 2FA exige une deuxième méthode d'authentification, généralement quelque chose que l'utilisateur possède physiquement comme un téléphone portable, où l'utilisateur reçoit un code de vérification et doit le saisir afin de compléter son authentification

L'authentification multi-facteurs

7. Recommandation CNIL et RGPD

La CNIL (Commission nationale de l'informatique et des libertés) est une autorité administrative indépendante française chargée de veiller à la protection des données personnelles et de garantir le respect de la vie privée dans le contexte de l'utilisation des technologies de l'information.

Les principales missions du CNIL sont :

- La protection des données personnelles.
- Le contrôle et la régulation.
- Sanctions administratives.
- Conseil et guidance.
- Sensibilisation du public.
- Réglementation et normes.
- Coopération internationale.
- Recherche et innovation.

La réglementation du CNIL : https://www.cnil.fr/fr/reglement-europeen-protection-donnees

Le RGPD (Règlement général sur la protection des données) est une loi de l'Union européenne qui vise à renforcer la protection des données personnelles des citoyens européens adopté par le Parlement européen en 2016.

Il impose aux entreprises qui collectent des données personnelles comme des noms ou adresses , un consentement des utilisateurs avant de collecter leurs données avec par exemple une pop-up demander la validation d'utilisation des données.

Informer les personnes et assurer la transparences :
https://www.cnil.fr/fr/conformite-rgpd-information-des-personnes-et-transparence

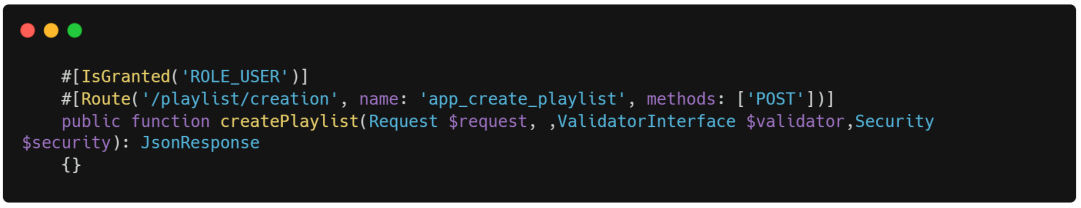
8. Gestion des routes et accès aux données

La gestion des routes permet de mettre en place des règles de contrôle d'accès basées sur les rôles et les permissions des utilisateurs. Je peux restreindre l'accès à certaines pages ou actions en fonction du niveau d'autorisation de l'utilisateur, ce qui réduit les risques d'accès non autorisés.

Dans le fichier config/packages/security.yaml on peut configurer et protéger certaines routes. Dans la partie "access_control", on va par exemple protéger la route /admin pour que seulles utilisateurs avec un rôle admin puisse y accéder

```yaml
access_control:
    - { path: ^/admin, roles: ROLE_ADMIN }
```

Il existe d'autres façon de protéger des routes comme au niveau du contrôleur :



```php
#[IsGranted('ROLE_USER')]
#[Route('/playlist/creation', name: 'app_create_playlist', methods: ['POST'])]
public function createPlaylist(Request $request, ,ValidatorInterface $validator,Security
$security): JsonResponse
{}
```

On peut aussi protéger une route à l'intérieur :

```php
#[Route('/playlist/creation', name: 'app_create_playlist', methods: ['POST'])]
public function createPlaylist(Request $request, ,ValidatorInterface $validator,Security
$security): JsonResponse
{
    $this->denyAccessUnlessGranted('ROLE_USER');
}
```

XIV. Politique de tests

Pour la politique de tests mise en place pour ce projet, j'ai adopté des fixtures pour créer un jeu d'essai complet.
Les data fixtures permettent de remplir automatiquement les tables avec des données générées.

Afin d'installer les fixtures, il faut installer le bundle dans le projet via composer avec la commande :

- composer require --dev orm-fixtures

De plus, j'ai aussi installé Faker, qui est une librairie qui permet de générer tout type de données.
L'installation se fait avec :

- composer require --dev fakerphp/faker

L'installation avec orm-fixtrues crée un fichier AppFixtures.php qui contiendra les fixtures

```php
class AppFixtures extends Fixture
{
    //injection de la dépendance UserPasswordHasherInterface
    private UserPasswordHasherInterface $hasher;
    private Generator $faker;

    public function __construct(UserPasswordHasherInterface $hasher){
        $this->hasher = $hasher;
        $this->faker = Factory::create('fr_FR');
    }

    public function load(ObjectManager $manager)
    {
        /* Autre partie du code */

        //Les users
        for ($i = 0; $i < 10; $i++) {
            $user = new User();
            $user->setPseudo('user'. $i);
            $user->setEmail('user'. $i. '@tunely.com');
            $user->setRoles(['ROLE_ADMIN']);
            $user->setGender( $genders[mt_rand(0, (count($genders)) - 1)]);
            $user->setIsEnabled(true);
            $hasPassword = $this->hasher->hashPassword($user, 'password');
            $user->setPassword($hasPassword);
            $manager->persist($user);
            $users[] = $user;

        }

      /* Autre partie du code */
    }
}
```

Exemple de fixtures

Après l'écriture des fixtures, il faut les chargés dans la base de données avec la commande:

- php bin/console doctrine:fixtures:load

	id	gender_id	artist_id	email	roles	password
Éditer Copier Supprimer	370	115	NULL	user0@example.com	["ROLE_ADMIN"]	$2y$13$v26NlLo5V/oi2iX9/zHt3.dByQutlSrrbhd9dlOOd/Q...
Éditer Copier Supprimer	371	116	NULL	user1@example.com	["ROLE_USER"]	$2y$13$Hl/fGnbLIB2Nur7udGFKP.75HWozEXkRm0XA33IOICf...
Éditer Copier Supprimer	372	116	NULL	user2@example.com	["ROLE_USER"]	$2y$13$flYQoTL7yRN3nJdWCwC5oOTJ1hiiqVqqt.nqDYPA0gA...
Éditer Copier Supprimer	373	117	NULL	user3@example.com	["ROLE_ADMIN"]	$2y$13$pMc2AubnbbUN1jaAkU3YJuWcFPc5ADM.Pv7OyFP1Mgm...
Éditer Copier Supprimer	374	116	NULL	user4@example.com	["ROLE_USER"]	$2y$13$rTYE9heAl3wgQEzLMV20t.JFkBqlSwxNmqXeiM3k9o3...
Éditer Copier Supprimer	375	116	NULL	user5@example.com	["ROLE_ADMIN"]	$2y$13$uU.uq.VE8Ectl2ClJmUKlOSeMe7tWC8li32kJS8FWB4...
Éditer Copier Supprimer	376	116	408	user6@example.com	["ROLE_USER"]	$2y$13$XO69vaiQYJzS4rxP2c/lNeMENUy4RthoIrGcs5l07PF...
Éditer Copier Supprimer	377	116	409	user7@example.com	["ROLE_USER"]	$2y$13$cKWyxPoQB/be6EEPNPwK0uWl/coM6/qjEnRh00umit4...
Éditer Copier Supprimer	378	115	410	user8@example.com	["ROLE_USER"]	$2y$13$vwq.u3fjmPSbUWwYckXMAua/Mz9iVHMGxcYNhKDZaqd...
Éditer Copier Supprimer	379	116	411	user9@example.com	["ROLE_USER"]	$2y$13$TaLCOqTQdfHfxJKnuTepO.JnXfNKDpe2.xOeZU.ZcX2...

La table user dans la base de données après chargement des fixtures

1. Pourquoi effectuer des tests ?

La fiabilité du logiciel est une part cruciale du développement car elle garantit que l'application fonctionne de manière prévisible et sans erreurs, offrant ainsi une expérience utilisateur positive.

Les tests sont importants pour plusieurs raisons :

- Détecter les erreurs : Trouver et corriger les bugs et défauts.
- Valider les fonctionnalités : Vérifier si toutes les caractéristiques du logiciel fonctionnent correctement.
- Améliorer la qualité : Identifier et éliminer les problèmes pour assurer une meilleure qualité globale du produit.
- Réduire les coûts : Corriger les erreurs tôt dans le processus de développement est plus économique.
- Gagner la confiance des utilisateurs : Offrir une expérience utilisateur stable et sans faille pour gagner leur confiance.
- Se conformer aux normes : Respecter les normes et réglementations applicables dans certains domaines.
- Valider les mises à jour : S'assurer que les changements n'entraînent pas de nouveaux problèmes.

Il existe plusieurs types de tests, que l'on peut répartir dans deux grandes catégories :

- Les test manuels
- Les tests automatiques

  Après avoir terminé un développement, le premier réflexe consiste souvent à effectuer un test manuel en lançant l'application, cliquant sur les composants de l'application afin de vérifier son bon fonctionnement.

Ces tests sont indispensables, mais ils présentent l'inconvénient d'être à la fois coûteux et fastidieux.

Les tests automatiques sont là pour faire gagner du temps et de l'argent en vérifiant automatiquement et rapidement les fonctionnalités de l'application.

2. Les différents niveaux de tests

Il existe un comité de test, le **Comité français du test logiciel** (CFTL).
C'est un organisme français qui représente le métier du test logiciel et travaille sur les normes de ce métier

Le CFTL identifie quatre niveaux de tests :

- Les tests unitaires permettent de tester un composant, ou un bout de code, isolé de ses dépendances.

- Les tests d'intégration permettent de vérifier que tous les bouts de code isolés fonctionnent bien ensemble.

- Les tests système permettent de tester l'application toute entière, dans les conditions proches du matériel client.
  Aussi appelé Validation

- Et enfin, les tests d'acceptation permettent de s'assurer que l'application répond bien au besoin fonctionnel.

Il existe aussi le **Comité International de Qualification du Test Logiciel** (ISTQB), qui est une organisation internationale à but non lucratif qui se consacre à la promotion et à la normalisation des meilleures pratiques dans le domaine du test logiciel.
Le comité à pour but d'établir un ensemble commun de compétences et de connaissances en test logiciel.

3. Les tests unitaires

Les tests unitaires sont des tests simples à mettre en place et doivent être rapides à l'exécution afin de récupérer le résultat du test rapidement.
Le but de ce test est de vérifier le bon fonctionnement d'une unité de code (une fonction, une méthode) de manière isolée de ses dépendances.

Les caractéristiques des tests unitaires sont :
- L'isolation, les tests doivent être indépendants les uns des autres et des autres parties du code.
- L'automatisation, qui permet aux tests de s'exécuter facilement à chaque fois que des modifications sont apportées au code, cela fait gagner du temps.

- Rapidité, ce qui facilite la détection rapide de problèmes potentiels.
- Couverture, ce qui permet de couvrir autant de cas que possible, y compris les cas limites et les scénarios d'erreur.

4. Les tests d'intégration

A la différence des tests unitaires, les tests d'intégration sont là afin de vérifier que tout fonctionne bien ensemble.
Ils permettent d'identifier les problèmes qui pourraient survenir lorsque les différentes parties du logiciel sont assemblées et s'assurer que toutes les interfaces et les dépendances fonctionnent correctement ensemble.

L'utilisation de bouchons dans les tests d'intégration est une pratique permettant de remplacer une dépendance que l'on ne maîtrise pas par un faux comportement sur lequel on a la main, on appelle ça aussi "Bouchonner"

5. Tests d'acceptation appelés tests fonctionnels

Les tests fonctionnels, également appelés tests d'acceptation, se concentrent sur la validation des fonctionnalités globales de l'application. Ils vérifient que l'application répond correctement aux spécifications et aux besoins des utilisateurs.

Contrairement aux tests unitaires et d'intégration qui se concentrent sur les aspects internes du code, les tests fonctionnels évaluent l'application du point de vue des utilisateurs finaux.

Le test fonctionnel requête une URL et récupère la réponse.

6. Comment tester avec Symfony ?

a. Les outils

Afin de mener à bien les tests, j'utilise PHPunit, qui est pré-intégré dans Symfony.
Créé en 2004 par Sebastian Bergmann, c'est une bibliothèque open-source, inspirée du framework de test unitaire JUnit, utilisé en Java et qui permet l'implémentation des tests en vérifiant que les exécutions correspondent aux assertions prédéfinies.

Les assertions permettent de valider si un résultat est conforme aux attentes. Quelques exemples d'assertions courantes sont assertEquals(), assertTrue(), assertFalse(), assertNull()

b. Écriture de tests unitaires avec PHPUnit et Symfony

Pour écrire des tests unitaires avec PHPUnit dans un projet Symfony, je dois créer des classes de test qui étendent la classe PHPUnit\Framework\TestCase.
Chaque méthode dans la classe de test sera un test unitaire.
Les méthodes de test doivent être nommées en préfixant "test" au nom du cas de test. Par exemple : testEntityIsValid().

i. Exemple de test unitaire

Dans cet exemple, je teste l'entité Song de mon application.
J'ai donc créé une classe "SongTest" afin d'avoir toutes mes fonctions de test.
La classe étend du KernelTestCase, grâce à lui, il permet de récupérer le validateur qui permettra de valider l'entité.

Dans l'exemple ci dessous je teste uniquement si l'entité est valide et si le nom est vide alors qu'il devrait être au minimum de 1 caractère et que le titre de la musique est obligatoire.

```php
class SongTest extends KernelTestCase
{

    public function getEntity(): Song
    {
        $date_obj = new \DateTime();
        //Je sélectionne un artiste et un genre musical
        $artist = static::getContainer()->get('doctrine.orm.entity_manager')->find(Artist::class, 1);
        $genre = static::getContainer()->get('doctrine.orm.entity_manager')->find(Genre::class, 1);

        return (new Song())
            ->setName('Song1')
            ->setIsEnable(true)
            ->setGenre($genre)
            ->addArtist($artist)
            ->setReleaseAt($date_obj)
            ->setDuration($date_obj)
            ->setFile('file1');
    }

    public function assertHasError(Song $song, int $number)
    {

        self::bootKernel();
        $container = static::getContainer();

        //Je valide mon entité et récupère une liste d'erreurs
        $errors = $container->get('validator')->validate($song);

        //Je récupère les messages des erreurs dans un tableau
        $messages = [];
        /** @var ConstraintViolation $error */
        foreach ($errors as $error) {
            $messages[] = $error->getPropertyPath() . ' => ' . $error->getMessage();
        }
        //Je m'attends à aucune erreur sur cet objet
        $this->assertCount($number, $errors, implode(', ', $messages));
    }

    public function testEntityIsValid(): void
    {
        //J'attends 0 erreur dans la création de cette entité
        $this->assertHasError($this->getEntity(), 0);
    }

    public function testInvalidName()
    {
        //Je test si en mettant un nom vide cela provoque une erreur
        $song = $this->getEntity();
        $song->setName('');
        //Je mattends à 2 erreurs
        $this->assertHasError($song, 2);
    }
}
```

c. Écriture des tests fonctionnels

Pour les tests fonctionnels je dois utiliser "Symfony\Bundle\FrameworkBundle\Test\WebTestCase" à la place de "PHPUnit\Framework\TestCase" afin d'utiliser toutes les fonctionnalités de PHPUnit pour les tests fonctionnels.

WebTestCase fournit des méthodes pour effectuer des requêtes HTTP, simulant ainsi les interactions avec l'application comme si un utilisateur naviguait réellement sur le site. Cela permet de tester les contrôleurs, les

routes, les vues et de vérifier les réponses HTTP, les redirections, les codes de statut, etc.

d. Exemple de test fonctionnel

Pour cet exemple de test fonctionnel, je vais tester le formulaire d'inscription
Dans mon fichier SecurityControllerTest.php, je crée une fonction testRegistration()
Cette fonction testera comme dit précédemment le formulaire de d'inscription avec des bonnes données.

```php
public function testRegistration(){

    // Création du client
    $client = static::createClient();

    // Requête pour accéder à la page d'inscription
    $crawler = $client->request('GET', '/inscription');

    // Sélectionner le bouton de soumission du formulaire d'inscription
    $submitButton = $crawler->selectButton('S\'inscrire');
    $form = $submitButton->form();

    // Remplissage des champs du formulaire
    $form["register_form[email]"] = "user0000@example.com";
    $form["register_form[pseudo]"] = "Randy";
    $form["register_form[plainPassword][first]"] = "password";
    $form["register_form[plainPassword][second]"] = "password";
    $form["register_form[gender]"] = 1;
    $form["register_form[roles]"] = "ROLE_USER";

    // Soumission du formulaire
    $client->submit($form);

    // Vérification du statut HTTP attendu
    $this->assertResponseStatusCodeSame(Response::HTTP_FOUND);

    // Vérification de l'envoi du courriel
    $this->assertEmailCount(1);

    // Suivre la redirection après soumission
    $client->followRedirect();

}
```

Fonction testRegistration()

Si les données sont correctes, après la commande php bin/phpunit, ça retourne

Time: 00:01.570, Memory: 44.00 MB
OK (4 tests, 5 assertions)

7. Plan de déploiement

a. C'est quoi un déploiement ?

Le déploiement en informatique englobe toutes les étapes essentielles visant à intégrer une application logicielle ou un système dans son environnement opérationnel prévu.
Ce processus comprend l'installation, la configuration, la réalisation de tests, ainsi que d'éventuelles adaptations nécessaires.

b. Quelles sont les étapes ?

Après les tests (unitaire, d'intégration et fonctionnels) je vais devoir déployer l'application d'abord à un groupe restreint d'utilisateurs afin de tester l'application et d'avoir un maximum de retours, le site sera en version beta.

Après les potentiel correctif apporté au site, je pourrais planifier la date de lancement officiel du site afin de le déployer complètement.

Je mettrais en place des outils afin de surveiller les performances du site et un support afin de résoudre les problèmes signalés par les utilisateurs.

Pour mettre en production l'application Tunely, il faut choisir un hébergement pour l'application qui soit performant afin de d'offrir une expérience agréable à l'utilisateur.
Il existe par exemple des hébergeurs très connus comme Azure de Microsoft ou OVH mais j'ai choisi d'utiliser AWS (Amazon Web Services).

Je pourrais mettre en place une stratégie de communication pour informer les futurs utilisateurs et les artistes.

Le site sera régulièrement mis à jour.

XV. Veille technologique

La veille technologique est un processus de surveillance sur les avancées et les tendances technologiques dans un domaine spécifique.

L'objectif de la veille technologique est de rester informé des changements et des progrès technologiques qui pourraient avoir un impact sur une entreprise, une industrie ou même la société dans son ensemble. Elle permet de détecter les opportunités et les menaces, d'anticiper les changements, d'innover et de rester compétitif sur le marché.

Pour ma part, j'utilise différents site web afin d'effectuer la veille technologique comme :

- Twitter en suivant des comptes sur le développement informatique

- dev.to est un site web se basant sur les principes d'un réseau social, qui permet de poster des articles, d'échanger via des commentaires et de noter les articles. C'est un site très actif avec des contenus de qualité.

Daily.dev est un site permettant de se tenir au courant des dernières nouvelles en matière de programmation.

XVI. Difficultés rencontrés

Pendant ce projet, j'ai rencontré différentes difficultés comme pour la conception du projet où il fallait faire un choix de méthode de gestion de projet, prioriser les fonctionnalités en fonction de leur importance et de leur complexité.
Des difficultés sur la manipulation de flux audio avec Howler js, d'ajouter des musiques en file d'attente.
Et des difficultés pour mettre en œuvre des tests unitaires et fonctionnels pour garantir la stabilité de l'application.

XVII. Conclusion

Pour conclure, c'est fut un projet très intéressant, il m'a permis d'apprendre davantage sur la conception d'application, la gestion de base de données, la sécurité sur internet, sur Symfony.

Cela m'a donné envie de **continuer dans les études au sein de l'IMIE** et dans **la gestion de projet**.

Durant le projet, j'ai pu être confronté à des difficultés qui ont été une occasion pour en apprendre davantage et enrichir mes connaissances.
Ce projet m'a également permis de mieux comprendre l'importance de la planification, de la gestion du temps, de l'organisation, et de l'importance de bien documenter pour avoir un code facile à maintenir et à évoluer.

Je tiens à exprimer ma gratitude envers mes enseignants pour leur soutien et leurs conseils tout au long de ce projet et à l'école IMIE.