



# Cahier des charges fonctionnel

| Version | Date       | Auteur                                             | Validation          | Commentaires                                                          |
|---------|------------|----------------------------------------------------|---------------------|-----------------------------------------------------------------------|
| V1      | 13/11/2025 | Antonin DRUJON                                     | Équipe : à vérifier | RAS                                                                   |
| V2      | 08/12/2025 | Rémi MAHIEUX                                       |                     |                                                                       |
| V3      | 10/12/2025 | Antonin DRUJON,<br>Julien DURIEUX                  | Équipe : à vérifier | Julien : J'ai corrigé et amélioré la cohérence. à voir pour la suite. |
| V4      | 11/12/2025 | Mavrick GOUIX<br>Antonin DRUJON,<br>Julien DURIEUX | Équipe : vérifié    | RAS sur ce qui a été fait. Le document n'est pas terminé.             |
| V5      | 26/02/2026 | Antonin DRUJON                                     | Équipe : à vérifier | Fin du document                                                       |

## 1. Présentation du contexte

### Description du projet

Le projet **Le Vinyle** consiste en la conception d'une application web destinée à enrichir l'expérience de streaming musical collaboratif sur Twitch. Elle permet aux streamers d'organiser des sessions participatives où les viewers peuvent proposer des morceaux en temps réel. L'application centralise la gestion des propositions, la modération, l'interaction communautaire et l'intégration avec les plateformes musicales, notamment Spotify. Elle vise ainsi à offrir un environnement structuré, contrôlé et interactif pour la construction collective d'ambiances sonores en direct.

### Objectifs généraux

Le Vinyle a pour ambition de faciliter la création et l'animation de sessions musicales collaboratives tout en assurant un haut niveau de contrôle au streamer. Le projet vise à :

- améliorer l'interactivité entre le streamer et sa communauté.
- proposer un système de gestion fluide, ergonomique et modérable des propositions musicales.
- automatiser la création et l'alimentation de playlists Spotify liées aux sessions.
- réduire les comportements abusifs et favoriser une expérience utilisateur harmonieuse.
- garantir une intégration naturelle avec l'écosystème Twitch, notamment via l'authentification.

## Parties prenantes

Le projet rassemble plusieurs acteurs ayant des rôles et attentes distincts :

- **Les streamers**, responsables de la création et de la gestion des sessions, qui recherchent un outil sécurisé, personnalisable et efficace pour animer leur communauté.
- **Les modérateurs**, chargés d'assurer la qualité des propositions musicales, de modérer le comportement des participants et de garantir le bon déroulement des sessions.
- **Les viewers**, utilisateurs finaux participant aux sessions en proposant des morceaux et en interagissant avec l'environnement musical défini par le streamer.
- **Les plateformes tierces**, dont Spotify et Twitch, fournissant les services d'authentification, de recherche musicale et de synchronisation des playlists.
- **L'équipe de développement**, responsable de la conception, de la mise en œuvre et de la maintenance de l'application.

## Enjeux et motivations

Le Vinyle répond à une évolution des pratiques de streaming où l'interaction communautaire est devenue un levier majeur d'engagement. Les streamers cherchent à renforcer la participation de leur audience en leur donnant un rôle

actif dans la construction de l'ambiance sonore de leurs émissions. Cependant, les outils existants manquent de flexibilité, de contrôle et d'intégration harmonieuse avec les plateformes musicales.

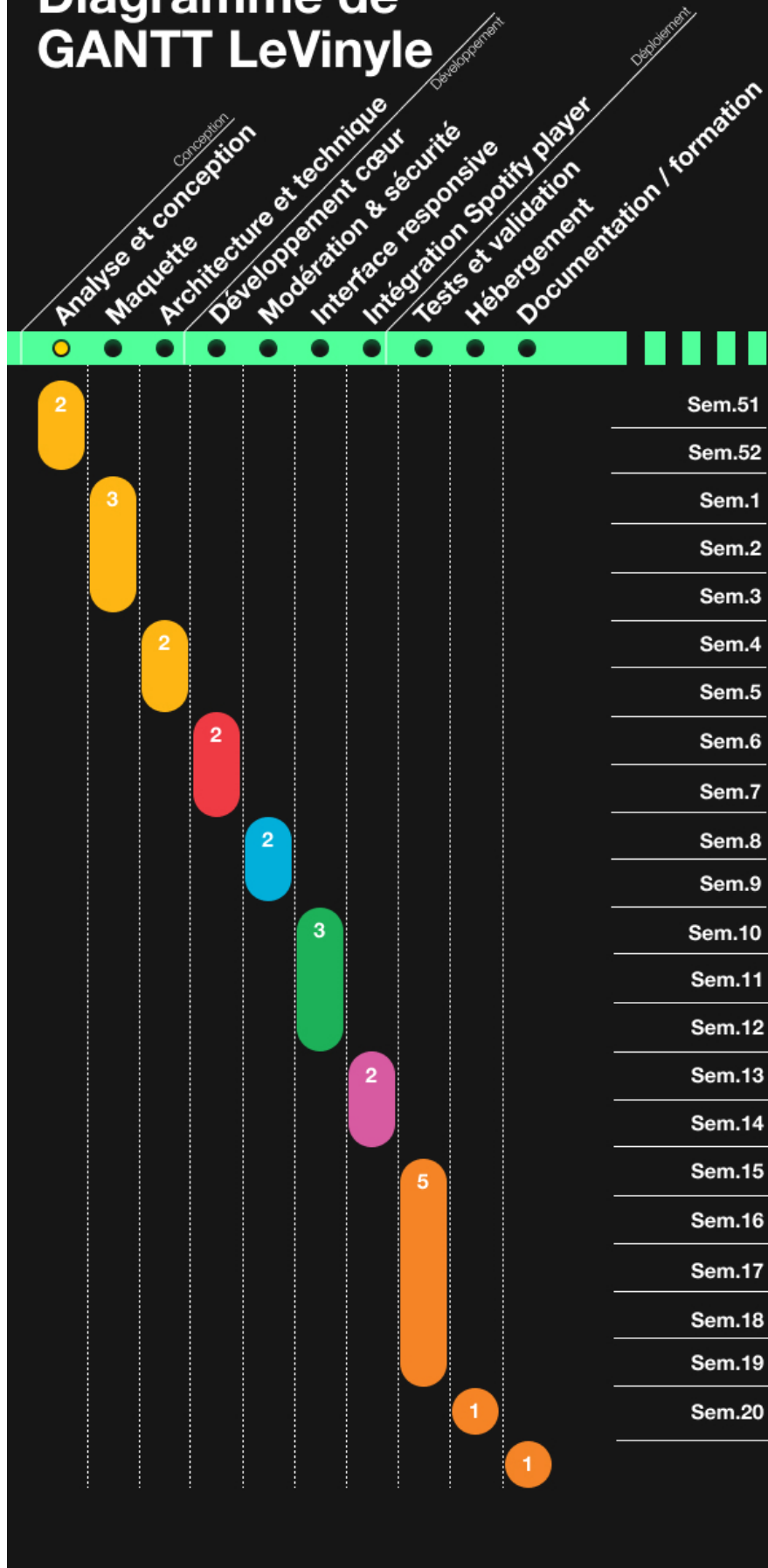
L'application vise donc à combler ce manque en offrant une solution complète et fiable permettant :

- une gestion maîtrisée des propositions en temps réel.
- une modération efficace et transparente.
- une expérience fluide pour les participants.
- une meilleure cohérence éditoriale au sein des sessions.
- une évolution progressive vers des fonctionnalités plus avancées comme les votes, l'historique public ou l'intégration d'autres plateformes musicales.

## **2. Planning prévisionnel (Diagramme de Gantt)**

Ce diagramme de Gantt présente la planification prévisionnelle du projet. Il décrit les différentes phases du projet, leurs durées estimées, ainsi que les dépendances entre les tâches, depuis l'analyse des besoins jusqu'à la livraison finale.

# Diagramme de GANTT LeVinyle



## 3. Définition du besoin

### Problème à résoudre

Les streamers Twitch qui souhaitent organiser des sessions d'écoute musicale avec leur communauté disposent aujourd'hui d'outils fragmentés et inadaptés. Les solutions existantes ne permettent ni une gestion fluide des propositions musicales en temps réel ni un contrôle suffisant sur leur diffusion. L'absence de modération structurée, de prévention du spam et de synchronisation directe avec les playlists Spotify génère une perte de cohérence éditoriale, une surcharge de travail pour le streamer et une expérience frustrante pour les viewers.

Le Vinyle vise à éliminer ces limitations en proposant une solution unifiée, modulable et ergonomique.

### Finalité recherchée

Le projet ambitionne de fournir une plateforme centralisée qui permet aux streamers de créer et d'animer des sessions musicales collaboratives tout en maintenant un haut niveau de contrôle sur les propositions. L'objectif est de garantir une expérience interactive, harmonieuse et engageante pour la communauté, soutenue par une intégration technique fiable avec Twitch et Spotify.

La finalité est double : améliorer la qualité de l'animation musicale en live et renforcer la participation active des viewers.

### Utilisateurs concernés (profils et attentes)

- **Streamers** : recherchent un outil performant pour organiser des sessions musicales, gérer plusieurs playlists, modérer les propositions, éviter les doublons et maintenir un environnement conforme à leur charte. Ils souhaitent un contrôle total sur l'ordre de lecture, les modes de gestion des propositions (FIFO, aléatoire, etc.) et l'intégration automatique avec Spotify.
- **Modérateurs** : ont besoin d'une interface claire, en temps réel, leur permettant d'analyser les propositions, d'écouter un extrait, de motiver un

refus, et de sanctionner les comportements abusifs (mute, ban). Ils attendent une traçabilité des interventions et une bonne visibilité sur l'activité de la session.

- **Viewers** : souhaitent une expérience rapide, intuitive et gratifiante. Ils veulent pouvoir proposer facilement des morceaux via Spotify, suivre l'état de leurs contributions, comprendre les refus, et participer sans friction à une ambiance collective. L'authentification Twitch leur garantit continuité et cohérence de leur identité.

## **Périmètre du projet (inclus / exclu)**

### **Inclus dans le périmètre :**

- Création, gestion et paramétrage de sessions musicales.
- Authentification via Twitch et Spotify pour les streamers, modérateurs et viewers.
- Recherche et proposition de morceaux via l'API Spotify.
- Gestion des propositions : acceptation, refus, justification, prévention des doublons.
- Modération des comportements : mute temporaire, ban, historique des décisions.
- Ajout des morceaux validés dans une playlist Spotify liée à la session.
- Interfaces dédiées par rôle : streamer, modérateur, viewer.
- Affichage en temps réel (mises à jour dynamiques).
- Gestion des sessions privées.

### **Exclus du périmètre (phase initiale) :**

- Système avancé de votes pour les propositions.
- Intégration d'autres plateformes musicales (YouTube Music, Apple Music, Deezer...).
- Historique public complet des sessions et statistiques avancées.
- Export automatisé de rapports.
- Intégration native dans OBS ou Streamlabs.

## 4. Fonctions du système

### 4.1. Fonctions principales

L'application **Le Vinyle** doit permettre aux streamers de créer et de gérer des sessions musicales collaboratives. Chaque session peut être configurée selon différents paramètres. Le streamer doit pouvoir visualiser en temps réel l'activité de la session, gérer les propositions reçues et superviser l'interaction avec sa communauté.

Les viewers, authentifiés via Twitch, doivent pouvoir proposer facilement des morceaux depuis Spotify, soit par lien, soit via une recherche intégrée. Ils doivent avoir accès au statut de leur proposition en temps réel, ainsi qu'à une **éventuelle** justification en cas de refus.

Les modérateurs doivent disposer d'un espace dédié leur permettant de consulter l'ensemble des propositions, d'écouter avant décision et d'accepter ou refuser les morceaux. Leur décision peut être accompagnée d'une justification visible par le viewer. Lorsqu'un morceau est validé, il sera joué durant la session de streaming.

La modération doit également couvrir les comportements abusifs. Les modérateurs doivent pouvoir appliquer un mute temporaire ou un bannissement à un viewer, avec possibilité de revenir sur la sanction. Un historique doit conserver la trace des actions de modération pour garantir la transparence et la cohérence du suivi.

Enfin, l'intégration Spotify doit permettre la création automatique / la liaison d'une playlist dédiée à chaque session ou globale et l'ajout potentiel des morceaux acceptés, assurant une synchronisation fiable entre les décisions de la communauté et la musique diffusée.

### 4.2. Fonctions contraintes

L'application doit respecter plusieurs contraintes techniques, notamment l'utilisation obligatoire des API Spotify et Twitch pour la gestion des morceaux et l'authentification. Le système doit offrir une mise à jour en temps réel des propositions et des actions de modération, nécessitant une technologie adaptée à ce type d'échanges (WebSockets ou équivalent).

Le Vinyle doit être compatible avec les principaux navigateurs et utilisable sur plusieurs taille d'écran, avec une interface responsive et ergonomique. La

distinction entre les rôles, streamer, modérateur, viewer doit être claire pour garantir une utilisation intuitive.

La sécurité constitue un aspect essentiel : les communications doivent être chiffrées, les données personnelles protégées et l'accès aux fonctionnalités sensibles strictement contrôlé. Les règles imposées par les plateformes tierces, notamment Spotify et Twitch, doivent être respectées, en particulier concernant l'utilisation des API et les limitations associées.

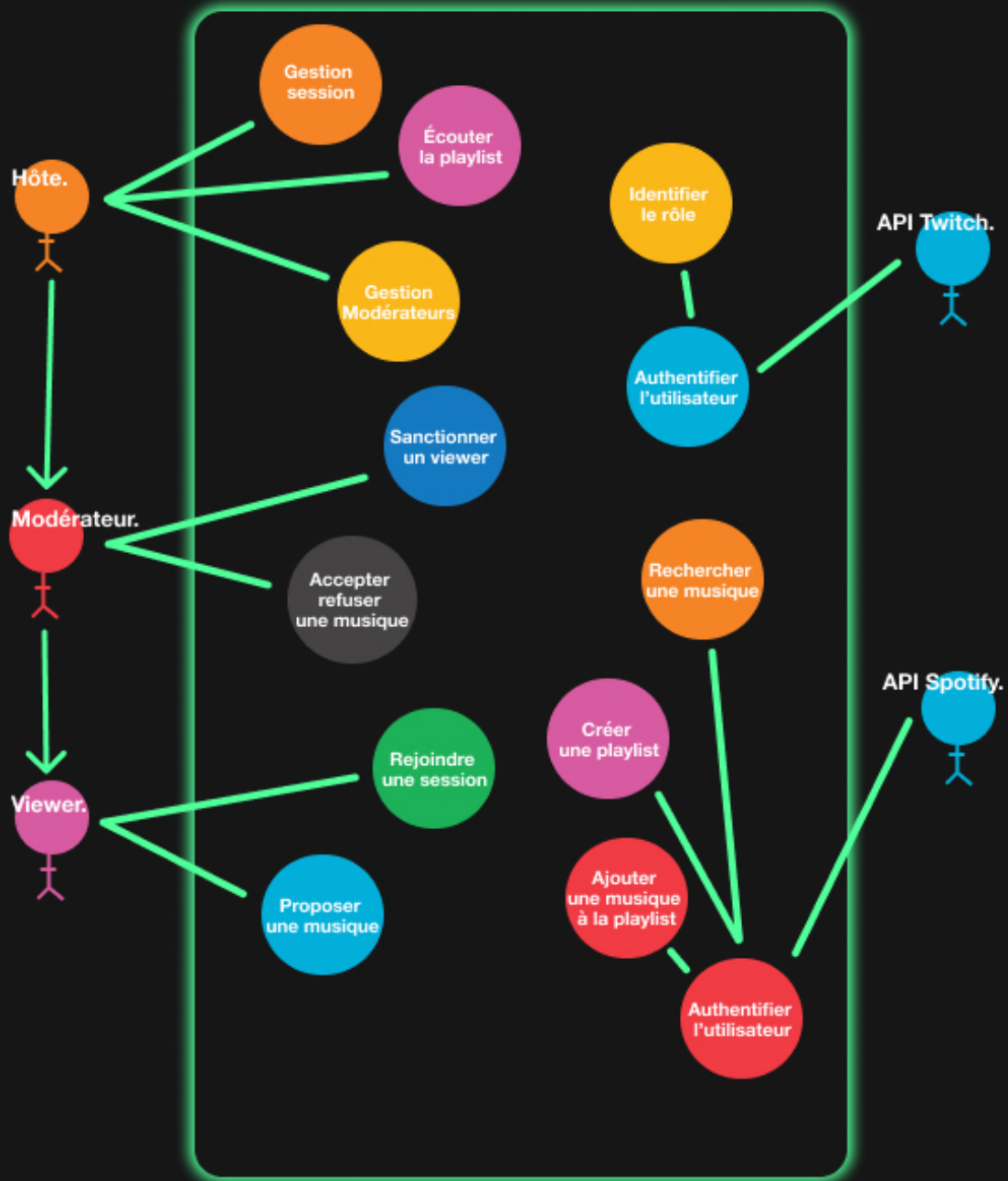
Enfin, le système doit maintenir des performances satisfaisantes, en assurant un temps de réponse court pour la recherche de morceaux, l'envoi des propositions et l'affichage des mises à jour. Il doit également pouvoir gérer une activité importante lors des sessions rassemblant une communauté large et active.

## 5. Description fonctionnelle

Le diagramme de cas d'utilisation ci-dessous présente les interactions entre les différents acteurs du système (Streamer / Hôte de session, Modérateur, Viewer) et les fonctionnalités principales de l'application *Le Vinyle*.

# DCU LeVinyle

Il reprend les besoins fonctionnels décrits dans le cahier de charge et identifie les acteurs, cas d'utilisation principaux et relations.



| Fonctionnalités - Hôte de session                                                         | Fonctionnalités - Modérateur de session                                                                      |                            |
|-------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|----------------------------|
| <ul style="list-style-type: none"> <li>• Modifier les paramètres d'une session</li> </ul> | <ul style="list-style-type: none"> <li>• Bannir / Kick / Mute des viewers</li> </ul>                         |                            |
| <ul style="list-style-type: none"> <li>• Ajouter des morceaux à la playlist</li> </ul>    | <ul style="list-style-type: none"> <li>• Accepter / Refuser les morceaux proposés par les viewers</li> </ul> |                            |
| <ul style="list-style-type: none"> <li>• Écouter les morceaux</li> </ul>                  |                                                                                                              |                            |
| <ul style="list-style-type: none"> <li>• Finir une session musicale</li> </ul>            |                                                                                                              |                            |
| <ul style="list-style-type: none"> <li>• Bannir / Kick / Mute des viewers</li> </ul>      |                                                                                                              |                            |
|                                                                                           | <th>Fonctionnalités - Viewer</th>                                                                            | Fonctionnalités - Viewer   |
|                                                                                           | <ul style="list-style-type: none"> <li>• Proposer des morceaux lors d'une session</li> </ul>                 |                            |
|                                                                                           |                                                                                                              |                            |
|                                                                                           | <th>Fonctionnalités - Globales</th>                                                                          | Fonctionnalités - Globales |
|                                                                                           | <ul style="list-style-type: none"> <li>• Créer une session paramétrable</li> </ul>                           |                            |
|                                                                                           | <ul style="list-style-type: none"> <li>• Rejoindre une session</li> </ul>                                    |                            |

## Fonctionnalités Hôte de session

### Modifier les paramètres d'une session

L'hôte peut modifier les paramètres de la session (titre, visibilité, mode de validation, limites, mode de lecture...). Le système applique immédiatement les changements.

**Entrée :** Nouveau paramétrage

**Sortie :** Confirmation de mise à jour

**Règles :** Ne pas interrompre la session en cours. Appliquer uniquement les paramètres autorisés

**Erreur :** Message en cas d'échec.

### MUST HAVE

**Indicateurs :** Temps d'application

**Valeur attendue :** < 2 secondes

### Accepter ou refuser une musique

L'hôte peut valider ou refuser une musique proposée. Le système met la liste à jour.

**Entrée :** Proposition musicale

**Sortie :** Statut + information pour le viewer

**Règles :** Ajout possible à la playlist si accepté.

**Erreur :** Message si ajout Spotify impossible.

### MUST HAVE

**Indicateurs :** Délai de traitement

**Valeur attendue :** < 1 seconde (hors API Spotify)

### Écouter les musiques

L'hôte écoute les musiques présentes dans la playlist via l'intégration Spotify.

**Entrée :** Playlist

**Sortie :** Lecture audio

**Règles :** Respecter les limitations Spotify API.

**Erreur :** Message si lecture impossible.

#### **MUST HAVE**

**Indicateurs :** Temps de démarrage

**Valeur attendue :** < 2 secondes

#### **Finir une session**

L'hôte peut clôturer la session. Le système désactive les interactions et archive les données nécessaires.

**Entrée :** Action "Fin de session"

**Sortie :** Session fermée

**Règles :** Aucune nouvelle action après fermeture.

**Erreur :** Message en cas d'échec.

#### **MUST HAVE**

**Indicateurs :** Temps de fermeture

**Valeur attendue :** < 1 seconde

#### **Bannir / kick / mute des viewers**

L'hôte applique des sanctions aux viewers. Le système met à jour immédiatement les droits du viewer.

**Entrée :** Viewer + type de sanction

**Sortie :** Confirmation

**Règles :** Mute temporaire, kick immédiat, ban durable.

**Erreur :** Message si sanction impossible.

#### **SHOULD HAVE**

**Indicateurs :** Temps d'application

**Valeur attendue :** < 1 seconde

## Fonctionnalités Modérateur

#### **Bannir / kick / mute des viewers**

Le modérateur peut appliquer des sanctions aux viewers selon les permissions de l'hôte.

**Entrée :** Viewer + type de sanction

**Sortie :** Confirmation

**Règles :** Mute temporaire, kick immédiat, ban durable, Impossible de sanctionner l'hôte.  
**Erreur :** Message en cas d'échec.

**SHOULD HAVE**

**Indicateurs :** Temps d'application  
**Valeur attendue :** < 1 seconde

**Accepter / refuser les musiques**

Le modérateur valide ou refuse les musiques proposées. Le viewer est informé du résultat et d'une justification éventuelle.

**Entrée :** Proposition musicale  
**Sortie :** Statut + justification  
**Règles :** Justification obligatoire en cas de refus.  
**Erreur :** Message si traitement impossible.

**SHOULD HAVE**

**Indicateurs :** Temps de traitement  
**Valeur attendue :** < 1 seconde

## Fonctionnalités Participants

**Proposer une musique**

Le participant propose un morceau via recherche ou lien Spotify. Le système valide la requête puis l'ajoute à la file d'attente.

**Entrée :** Musique proposée  
**Sortie :** Confirmation  
**Règles :** Pas de doublon. Interdit si participant mute.  
**Erreur :** Message si proposition invalide.

**MUST HAVE**

**Indicateurs :** Temps de validation  
**Valeur attendue :** < 1 seconde

## Fonctionnalités globales

**Créer une session**

L'utilisateur crée une session avec des paramètres personnalisés. Une playlist Spotify est créée automatiquement.

**Entrée :** Paramètres  
**Sortie :** Session + playlist

**Règles :** Authentification Twitch + Spotify obligatoire.

**Erreur :** Message si création impossible.

**MUST HAVE**

**Indicateurs :** Temps de création

**Valeur attendue :** < 3 secondes

**Rejoindre une session**

L'utilisateur rejoint une session existante.

**Entrée :** ID session / lien

**Sortie :** Connexion

**Règles :** Interdire l'accès aux utilisateurs bannis.

**Erreur :** Message d'accès refusé.

**MUST HAVE**

**Indicateurs :** Temps d'entrée

**Valeur attendue :** < 2 secondes

**Disponibilité :** 100%

## 6. Les interfaces utilisateurs

 La maquette

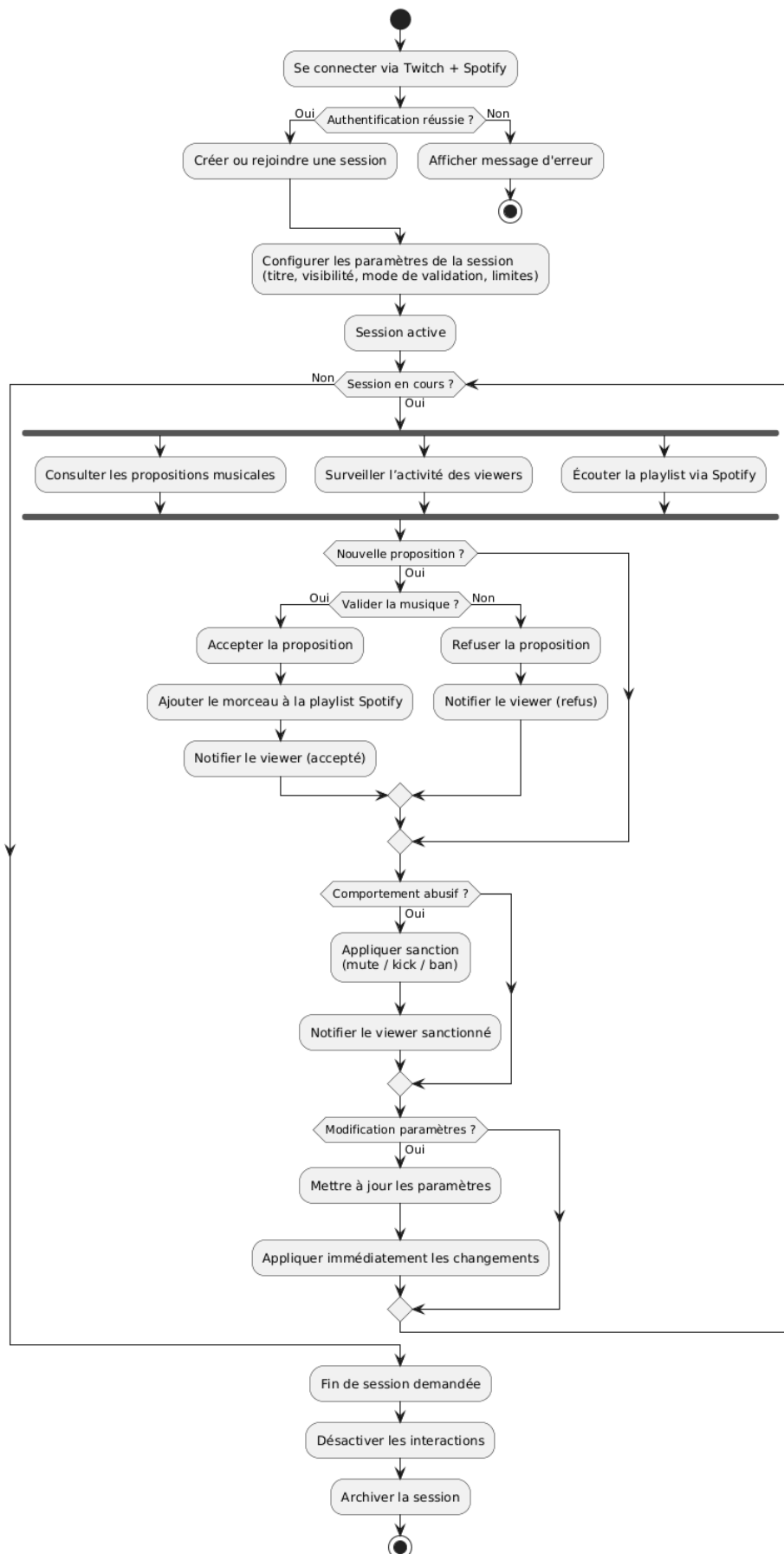
## 7. Scénarios d'usage / cas d'utilisation

Décrire comment l'utilisateur doit interagir avec la solution.

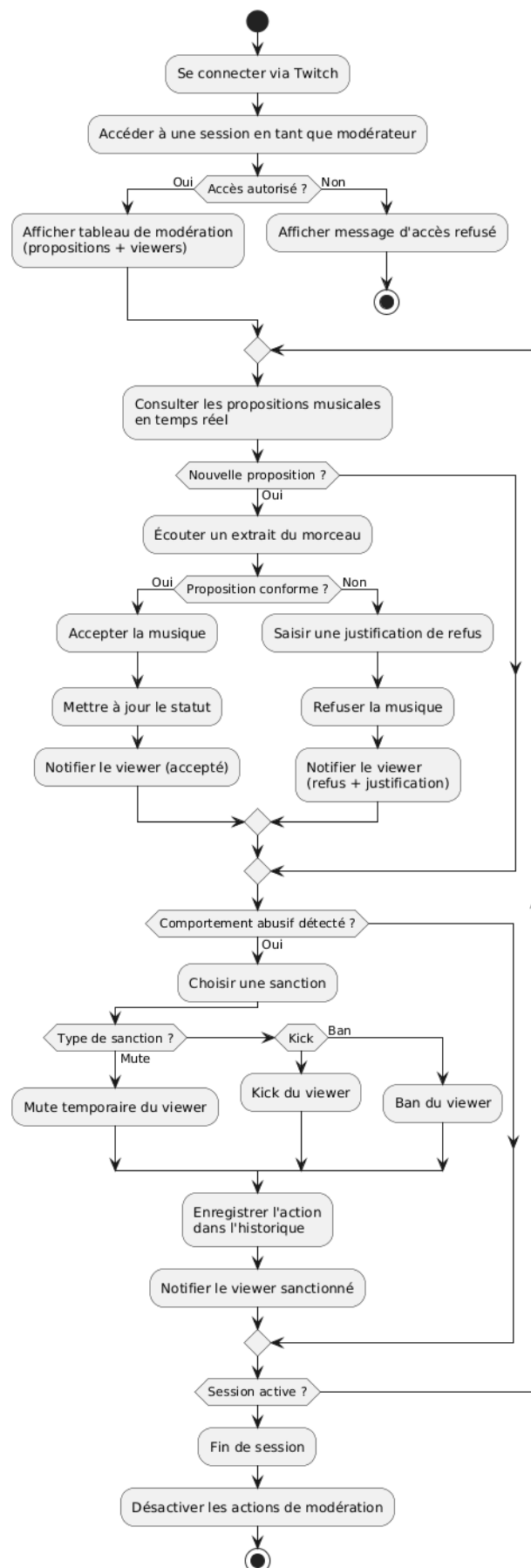
Exemples :

- Parcours utilisateur
- Scénarios typiques
- Diagrammes d'usage (optionnel)

### Diagramme d'activité - Hôte de session (Le Vinyle)

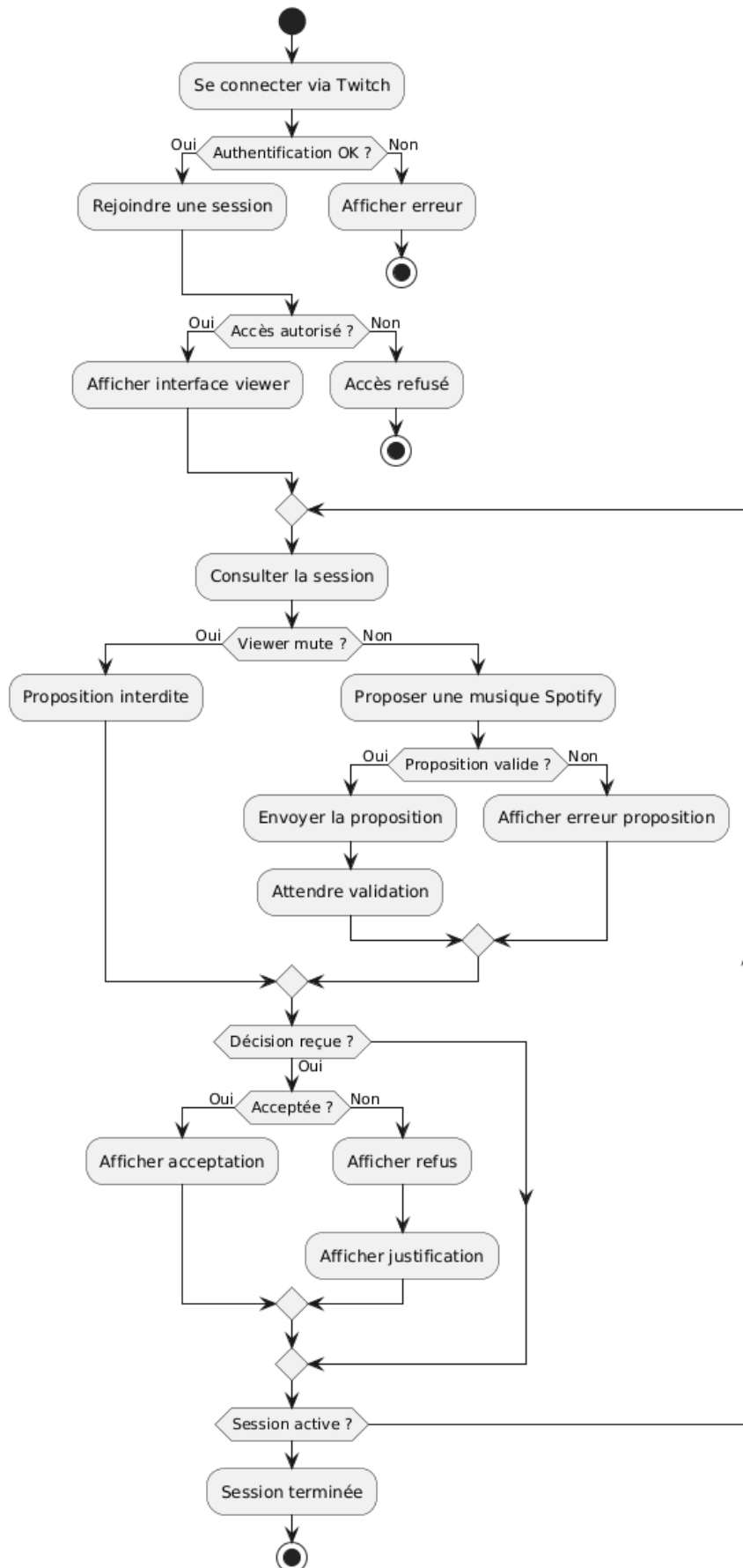


### Diagramme d'activité - Modérateur (Le Vinyle)





### Diagramme d'activité - Viewer (Le Vinyle)



## 8. Contraintes techniques, réglementaires et organisationnelles

### 8.1 Contraintes techniques

#### Architecture et technologies

Le projet *Le Vinyle* repose sur une architecture web moderne, compatible avec les standards actuels du développement d'applications en ligne.

L'application devra :

- Être accessible via les principaux navigateurs (Chrome, Firefox, Edge, Safari).
- Proposer une interface responsive adaptée aux ordinateurs, tablettes et smartphones.
- Utiliser un système de communication temps réel (WebSockets ou technologie équivalente) afin d'assurer la mise à jour instantanée des propositions musicales et des actions de modération.
- Garantir une architecture sécurisée (HTTPS obligatoire, gestion sécurisée des tokens OAuth, protection contre les attaques courantes : XSS, CSRF, injections).

Aucune technologie spécifique n'est imposée à ce stade, mais les choix techniques devront garantir :

- la maintenabilité du code,
- la scalabilité en cas d'augmentation du nombre d'utilisateurs simultanés,
- la facilité de déploiement et de mise à jour.

#### Intégration aux plateformes tierces

L'application dépend directement des API officielles de :

- **Twitch** pour l'authentification et l'identification des utilisateurs.
- **Spotify** pour la recherche de morceaux, la gestion des playlists et l'ajout de titres.

À ce titre :

- Les limitations imposées par ces API (quotas, restrictions de lecture, droits d'accès) devront être respectées.
- Les évolutions ou modifications des API tierces pourront impacter le fonctionnement de l'application.
- L'indisponibilité temporaire d'un service tiers pourra affecter certaines fonctionnalités (recherche, ajout à la playlist, lecture).

## Performance et disponibilité

Le système devra assurer :

- Un temps de réponse inférieur à 1 à 3 secondes selon la fonctionnalité (hors dépendances API externes).
- Une gestion fluide des sessions regroupant un nombre important de participants simultanés.
- Une haute disponibilité pendant les périodes de streaming.

Une surveillance minimale des performances devra être envisagée afin d'anticiper les dégradations de service.

## 8.2 Contraintes réglementaires

### Protection des données personnelles

Le projet devra respecter les réglementations en vigueur relatives à la protection des données personnelles, notamment le **RGPD (Règlement Général sur la Protection des Données)** pour les utilisateurs situés dans l'Union Européenne.

À ce titre :

- Les utilisateurs devront être informés de la nature des données collectées.
- Une politique de confidentialité devra être rédigée et accessible.
- Les utilisateurs devront pouvoir demander la suppression de leurs données.
- Les données ne devront pas être conservées au-delà de la durée nécessaire.

### Respect des conditions d'utilisation des plateformes tierces

Le Vinyle devra respecter :

- Les conditions d'utilisation de l'API Twitch.
- Les conditions d'utilisation de l'API Spotify.
- Les règles liées aux droits d'auteur et à la diffusion musicale.

L'application ne stocke ni ne diffuse directement les fichiers audio, mais agit comme intermédiaire via Spotify. La responsabilité liée à la diffusion du contenu musical reste encadrée par les conditions des plateformes utilisées.

## **8.3 Contraintes organisationnelles**

### **Ressources humaines**

Le projet est développé dans un cadre bénévole par une équipe restreinte.

Cela implique :

- Une disponibilité variable des membres selon leurs engagements personnels ou professionnels.
- Une planification flexible mais structurée.
- Une priorisation stricte des fonctionnalités essentielles (MUST HAVE).

### **Contraintes budgétaires**

Le projet ne dispose pas de financement dédié à ce stade.

Les choix techniques devront donc privilégier :

- Des technologies open-source ou gratuites.
- Des solutions d'hébergement à coût réduit.
- Une optimisation des ressources afin de limiter les frais d'infrastructure.

### **Maintenance et évolutivité**

La maintenance corrective et évolutive sera assurée bénévolement par l'équipe de développement.

L'architecture devra donc :

- Être documentée.
- Être modulaire afin de faciliter les évolutions futures (ex : ajout d'autres plateformes musicales).

- Permettre une correction rapide des anomalies critiques.

## 9. Livrables attendus

Le projet *Le Vinyle* donnera lieu à la production de livrables à la fois documentaires et techniques, permettant de garantir la conformité de la solution avec les besoins exprimés dans le présent cahier des charges.

En premier lieu, le projet devra aboutir à une version fonctionnelle de l'application web. Cette version devra intégrer l'ensemble des fonctionnalités définies comme prioritaires (MUST HAVE), notamment :

- L'authentification via Twitch et Spotify.
- La création et la gestion de sessions musicales.
- La proposition et la modération des morceaux
- L'ajout des titres validés à une playlist Spotify,
- La mise à jour en temps réel des interactions.

L'application devra être stable, utilisable en situation réelle de streaming et conforme aux exigences de performance définies précédemment.

Des versions intermédiaires pourront être produites au fil des différentes phases du projet, conformément au planning prévisionnel. Ces versions permettront de valider progressivement les fonctionnalités développées et d'identifier d'éventuelles anomalies avant la livraison finale.

En complément du produit logiciel, plusieurs éléments documentaires devront être fournis. Le cahier des charges fonctionnel constituera le document de référence du projet. Une documentation technique devra également être rédigée afin de faciliter la compréhension de l'architecture, des choix techniques effectués et des modalités de déploiement. Cette documentation devra permettre d'assurer la maintenance et l'évolution future de l'application.

Un guide d'utilisation synthétique devra être mis à disposition des utilisateurs (hôtes, modérateurs, participants). Il décrira les principales fonctionnalités et les étapes nécessaires à la création et à la gestion d'une session.

Des tests devront accompagner la livraison de l'application. Ils comprendront des tests fonctionnels permettant de vérifier le bon déroulement des principaux parcours utilisateurs, ainsi que des tests d'intégration liés aux API Twitch et Spotify. Une validation interne devra confirmer que les fonctionnalités

respectent les exigences définies et qu'aucune anomalie bloquante ne subsiste.

Enfin, la livraison finale sera considérée comme validée lorsque l'application sera opérationnelle, documentée, conforme au périmètre défini et exploitable sans erreur majeure dans un contexte réel d'utilisation.

## 10. Modalités de validation

La validation du projet *Le Vinyle* vise à s'assurer que l'application développée répond pleinement aux besoins exprimés dans le présent cahier des charges. Elle repose sur une vérification fonctionnelle, technique et opérationnelle de la solution livrée.

Le processus d'acceptation repose d'abord sur la conformité au périmètre défini. L'ensemble des fonctionnalités classées **MUST HAVE** devra être opérationnel et exploitable dans des conditions réelles d'utilisation. Aucune anomalie bloquante ne devra empêcher la création d'une session, la proposition de musiques, la modération des contenus ou l'ajout des morceaux à une playlist Spotify.

Des tests fonctionnels seront réalisés afin de vérifier les principaux parcours utilisateurs : création et paramétrage d'une session par l'hôte, proposition de morceaux par les participants, validation ou refus par les modérateurs, gestion des sanctions et clôture de session. Ces tests permettront de s'assurer que les règles métier sont respectées (absence de doublons, interdiction de proposer en cas de mute, justification obligatoire en cas de refus, etc.).

Une attention particulière sera portée à l'intégration avec les services tiers (Twitch et Spotify). Les mécanismes d'authentification, la recherche de morceaux et l'ajout à une playlist devront fonctionner conformément aux contraintes imposées par leurs API respectives. En cas d'erreur externe, l'application devra réagir de manière claire et contrôlée (message explicite, absence de blocage global du système).

Les performances constitueront également un critère de validation. Les temps de réponse définis dans les sections précédentes devront être respectés dans des conditions normales d'utilisation. L'affichage des mises à jour en temps réel devra être fluide et cohérent pour l'ensemble des participants d'une session.

Le besoin sera considéré comme satisfait lorsque l'application sera stable, conforme au périmètre fonctionnel défini, documentée, et utilisable en situation

réelle de streaming sans dysfonctionnement majeur. La validation finale sera prononcée à l'issue d'une phase de test interne concluante.