

**Cross Platform Playlist Sharing - PRD**

| **Technical Lead**  [Brett Wright](mailto:brettwrightsemail@gmail.com)  **Product Manager**  [Dan Worwood](mailto:danworwood@gmail.com) | **Project Kickoff Date**  Dec 29, 2024 |
| --- | --- |

## 

Table of Contents

[**Business Case Summary - Business Requestor 2**](#_7ss6chq21s55)

[**Discovery Questions/Decisions - Product Manager 4**](#_cgfsokieegq4)

[**Discovery - Assess & Design - Product Manager 6**](#_1brfmjyj5y7u)

[**Implementation Strategy - Engineering Lead 6**](#_t9w2012g41jh)

[**Delivery Strategy - Engineering Lead 8**](#_yvkswow320dj)

[**TRD Roles & Responsibilities (RACI) 10**](#_wkv3rgb6ifs5)

[**Appendix 11**](#_c8d2ud2tvfkx)

# 

# **Business Case Summary**

| **Problem /Opportunity Description:**  *(What is the problem we are solving with this project? Who does it negatively impact today? )* |
| --- |
| ***Sharing music playlists across platforms like YouTube Music and Spotify is a frustrating and time-consuming process for users.***  Users who subscribe to one platform often find it difficult to engage with playlists shared by friends or others using a different service. The lack of seamless interoperability forces users to either:   1. Listen on an unfamiliar platform they don’t subscribe to, often with limited access or ads, or 2. Manually recreate the playlist on their preferred service, which is tedious and detracts from the joy of sharing and discovering music.   This fragmentation reduces the social and collaborative experience of enjoying music, creating a significant pain point for users across ecosystems. |
| **Solution Description:**  *(What would solve the problem/issue above?)* |
| **Reduce the number of systems to be manually updated:**:  A cross-platform playlist conversion and sharing service that automates playlist transfers between streaming platforms like YouTube Music and Spotify |

| **Internal Personas Impacted** | **Key Attributes** | **Use Case** |
| --- | --- | --- |
| **The Social Music Sharer**: A young, socially active individual who loves discovering and sharing music with friends across different platforms. They value seamless sharing and want their friends to enjoy playlists without barriers. | Tech-savvy, active on social media, values convenience, uses streaming platforms for personal and group enjoyment. | Alex curates a "Road Trip Playlist" on Spotify and wants to share it with friends, some of whom use YouTube Music. Alex uses the service to quickly convert and share the playlist, ensuring everyone can access it on their preferred platform without hassles. |
| **The Collaborative Creator**: A music enthusiast or hobbyist who enjoys co-creating and refining playlists with friends or collaborators. This persona seeks tools for easy collaboration and updates across platforms. | Passionate about music curation, focused on collaboration, values precision in song selection and order. | Taylor and their bandmates use separate platforms but want to build a shared playlist for inspiration. Taylor initiates the playlist on YouTube Music, and with bidirectional syncing, updates made by their bandmates on Spotify appear seamlessly. |
| **The Casual Listener with a Music Network**: A less tech-savvy user who doesn’t want to be left out when friends or family share playlists on different platforms. They seek simplicity and convenience without technical hurdles. | Prefers ease of use over customization, not deeply familiar with advanced tech but regularly uses streaming platforms. | Jordan’s friend sends a curated holiday playlist from Apple Music, but Jordan uses Spotify. Instead of giving up or trying to recreate the playlist manually, Jordan uses the platform to convert the playlist in a few clicks, enjoying it without needing technical knowledge. |

| **Goals/Objectives for proposed solution**  *(What must be true for this to be considered a success?)* |
| --- |
| * **Automate Playlist Conversion**: Utilize APIs from popular streaming platforms (e.g., YouTube Music, Spotify) to programmatically map and convert playlists between services. * **Centralized Sharing Interface**: Develop a lightweight, user-friendly web or mobile application where users can input playlist IDs and receive a converted playlist that is compatible with their preferred service. * **Preserve Metadata and Order**: Ensure song metadata (e.g., title, artist, album) and playlist order are maintained during the conversion to enhance the user experience. * **Enable Bidirectional Syncing**: Implement functionality that allows playlists to remain synchronized across platforms, enabling collaborative updates from any platform to reflect on the other seamlessly. * **Handle Edge Cases Gracefully**: Provide clear feedback for unavailable or mismatched songs, with options to suggest alternatives or leave placeholders. * **Privacy and Security**: Protect user data through secure API integrations and clear permissions for accessing user playlists. |

# **Discovery Questions/Decisions**

| **Discovery Questions** (*What would need to be decided or answered so IT can assign the right work to the right stakeholders for discovery)* | | | |
| --- | --- | --- | --- |
| **Impacted Workstream** | **Outstanding Decision/Question** | **Decision Owner** | **Decision Status** |
|  |  |  | Not started |
|  |  |  | Not started |
|  |  |  | Not started |
|  |  |  |  |

# **Discovery - Assess & Design**

*(What are the requirements and solutions that would be required for delivery?)*

| **Technical requirements** (*What would need to be true technically to deliver the above needs?)* | | |  |
| --- | --- | --- | --- |
| **Impacted Area** | **Requirement** | **Technical Solution Summary** | **Requirement Status** |
| **Automate Playlist Conversion** | As a user, I want to input a playlist link or playlist ID from one streaming service and receive a converted playlist on my preferred service. | Integrate APIs from platforms (e.g., Spotify, YouTube Music) to map and convert playlist data programmatically. | Ready For Technical Design |
| As a user, I want only the songs available on the target platform to be included in the converted playlist to avoid errors. | Develop filtering logic to exclude unavailable tracks before delivering the converted playlist. | Ready For Technical Design |
| As a user, I want to see a report of songs that could not be converted due to platform limitations. | Generate a report during conversion to log unavailable tracks with explanations for omissions. | Ready For Technical Design |
| **Centralized Sharing Interface** | As a user, I want a simple interface to paste a playlist link or upload a file to start the conversion process. | Build a lightweight web/mobile app with a playlist input form. | Ready For Technical Design |
| **Preserve Metadata and Order** | As a user, I want my playlists to retain song metadata (title, artist, album) during conversion to ensure accuracy. | Map metadata fields from the source platform to the target platform using API schemas. | Ready For Technical Design |
| As a user, I want the playlist order to remain the same after conversion. | Implement playlist sorting logic during data processing to preserve order. | Ready For Technical Design |
| **Enable Bidirectional Syncing** | As a user, I want my playlists to sync automatically across platforms whenever I make updates. | Use webhooks or polling mechanisms to track updates and synchronize playlists across APIs in real time. | Ready For Technical Design |
| **Handle Edge Cases Gracefully** | As a user, if the system confidence of a match is low, it would require user confirmation before mapping a specific song | Use AI or matching algorithms to recommend similar tracks from the target platform's library. | Ready For Technical Design |

| **High Level Dependencies** |
| --- |
|  |

# 

## **Sprint Planning Delivery Strategy**

| **Epic breakdown**  *(Epics are defined as a single whole, but each epic undergoes incremental implementation.)* | **Proposed Deliverables for Epic** | **Estimated Effort** | **Timeline/ Deadline** | **Deliverable Status** |
| --- | --- | --- | --- | --- |
|  |  |  |  | Discovery |
|  |  |  |  | Discovery |
|  |  |  |  | Discovery |
|  |  |  |  | Discovery |

# 

| **Sequencing and Dependencies**  *(Describe any constraints for sequencing the epic and identify any potential dependencies with other epics or solutions)* |
| --- |
|  |

**Change/Decisions Log** *(The change log tracks all modifications made to the project over time. It ensures transparency, accountability, and provides a historical record of decisions and changes.)*

| **Date** | **Description** | **Reason** | **Cost of Change** | **Time Required** | **Trade-offs** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |