

Assignment 1 - Vision and Scope Document

Team [25]

September 7, 2025

1. Project Vision

1.5 Vision Statement

For music enthusiasts who seek hassle-free playlist creation, searching and sharing, with precise control over which parts of songs are heard in a playlist. Our playlist manager supports track start/end markers as metadata and allows building playlists that play only the track parts you want. Unlike other media player vendors, it lets you set start/end markers for each individual track and combine them into quick playlists, skipping the undesired parts of your music library altogether.

1.3 Business Objectives

- Users can create, delete and name their playlists, as well as set the visibility of their playlists to others.
- Users can add start and end markers to individual tracks, allowing them to play only the parts they want, skipping unwanted sections.
- The order of tracks in each playlist remains stable after edits or deletion of the actual sound files.
- Playlists can be tagged, searched and filtered, helping users find the content they want.
- Users can share their playlists via visibility setting and save public playlists made by others.
- No one other than the playlist owner can modify their original playlist.
- A user can make a copy of another user's playlist if they want to add their own modifications without manually rebuilding the said playlist.
- The playlist system should be smooth and reliably support all presented use cases.

2.2 Scope of the Initial Release

- Accounts: creation, deletion, profile customisation, the ability to sign in.
- Playlists: creation, deletion, naming, renaming, visibility setting (public/private).
- Default track catalogue containing full metadata (e.g., artist, song title, album, tags).
- Start/end marker metadata, stored per track and applied within a playlist.
- Track search with result filtering and paginated results.
- Stable ordering rules within a playlist.
- Adding and removing playlists from favourites.
- Playlist copying for personal modifications.
- The ability to set or remove a playlist image.
- Ensuring that only playlist owners can delete or modify their playlists.

2. Use Cases

2.1 Use Case 1 – Add tracks to a playlist

Primary Actor: User.

Preconditions: The user must be logged into the system, the user must have a playlist that has already been created.

Success Guarantee: The song that has been selected by the user is added to the playlist.

Main Success Scenario: The user opens a playlist that already exists. He selects the “add song” option. He searches for a song (as described in UC2) and selects it. The system adds the song to the playlist and gives confirmation that the song has successfully been added to the playlist.

Extensions / Alternated Scenarios: If a song has already been added to the playlist, the system could display “this song has already been added to the playlist”.

Miscellaneous / Open issues: The ability to add multiple songs at once.

2.2 Use Case 2 – Search tracks

Primary Actor: User.

Preconditions: None.

Trigger: User selects the “search” option.

Main Success Scenario: The user enters a search query (song name, artist, album, etc). The system displays a list of matching songs. The user can view details of a selected song.

Success Guarantee: The system returns relevant songs that match the search criteria.

Extensions / Alternate Scenarios: If no results are found, the system displays “No songs found.” If the search query is empty, the system prompts the user to enter a valid search term.

2.3 Use Case 3 – Create Playlist

Primary Actor: User

Preconditions: User is signed in to his account.

Main Success Scenario: User creates a new playlist, gives details about the playlist such as, the name and the privacy setting (public or private). The system validates if the required fields have been filled out. The system creates a playlist and displays the new playlist page, so that the user can start adding songs to his new playlist.

Success Guarantee: The new playlist is saved and created, and is ready for adding songs and sharing.

Extensions / Alternate Scenarios:

-We might have missing requirement information such as if the user does not provide a playlist name. The system would then display “Playlist name is required”, this issue must be fix by the user before he could continue.

2.4 Brief Use Cases

Brief Use Case 1 (UC4) – Delete playlist

The user is able to permanently delete a playlist.

Brief Use Case 2 (UC5) – Share playlist

The user is able to generate a link to share his playlist with friends, family and the community.

Brief Use Case 3 (UC6) – Save playlists

The user is able to save any public playlist to their favorites, or remove it from favorites.

Brief Use Case 4 (UC7) - Remove song from a playlist

The user can remove a song from his playlist.

Brief Use Case 5 (UC8) – Sort and Filter Playlist

User can filter his playlist by genre, artist or duration. Or sort by title, date or when the track was added.

Brief Use Case 5 (UC9) – Let the user specify start and end times for tracks in a playlist

User can specify a start and end time for tracks in his playlist.

3. Project Plan

Inception: Use cases and vision scope

Elaboration: UC1, UC2, UC3, UC6

Construction: UC4, UC5, UC7

Transition: UC8, UC9

Sprints	Use Cases	Process phase	Effort estimation	Prioritization
1	UC1 – Add Tracks to Playlist	Elaboration	5	10
1	UC2 – Search Tracks	Elaboration	7	20
1	UC3 - Create Playlist	Elaboration	5	10
2	UC4 – Delete Playlist	Construction	3	10
2	UC5 – Share Playlist	Construction	4	20
1	UC6 – Save a Playlist	Elaboration	5	30
2	UC7 – Remove Song from Playlist	Construction	3	10
3	UC8 – Sort and Filter Playlist	Transition	10	30
3	UC9 – Let user specify a start and end times for tracks in a playlist	Transition	12?	40

4. Project skeleton

https://github.com/jok10/HBV501G_Hopur25

jok90, jok10 -- Jóhann Kjartansson

Loftur5 -- Loftur Páll Eiríksson

masderni -- Andri Már Sigurðsson

MountResonance -- Valentin Grechikhin