

Vibe Player Milestone #4 Requirements

This document describes the requirements for the second development milestone (**Milestone #4**) of the **Vibe Player** app.

The goal is to make playlist control more deeply integrated into the listening experience — allowing users to add songs from the Now Playing screen, rename or delete playlists, change their cover art, and browse their contents.

The mockups define the app's appearance and behavior. Feel free to decide how you implement specific things (e.g., how you display a specific loading progress or how you specify error messages).

You can find the Vibe Player **mockups here**:

<https://www.figma.com/design/yge7H3tImtXHQZXCG9hUVr/VibePlayer?node-id=0-1>

Milestone #4 Goal

- Build a **complete playlist management system** that seamlessly connects music playback and library organization.
- Users should be able to quickly add songs to playlists, create new ones, edit or delete existing playlists, and view or start playback directly from any playlist.
- All interactions should feel natural, maintaining smooth navigation and continuous playback throughout the app.

Icons

You may use Material Design icons where appropriate. If a suitable Material icon is not available, use the custom icons provided in the Figma mockups.

In Figma, any icon or image can be **exported** by selecting the element and clicking “Export” in the **right-hand panel**. In this panel, you can also choose the desired format (PNG, SVG, etc.).

Adaptive Layouts

The app must support two breakpoints:

- **Up to 840 dp** → mobile layout.
- **From 840 dp and above** → wide-screen layout.

- 📌 Since there's already plenty to do in this challenge, full wide-screen adaptation is **optional**. Only **portrait orientation** is required, but still, check the 840dp+ mockups to understand how elements should look and align on larger screens.

Technical Requirements

🎧 Now Playing Screen — Updates

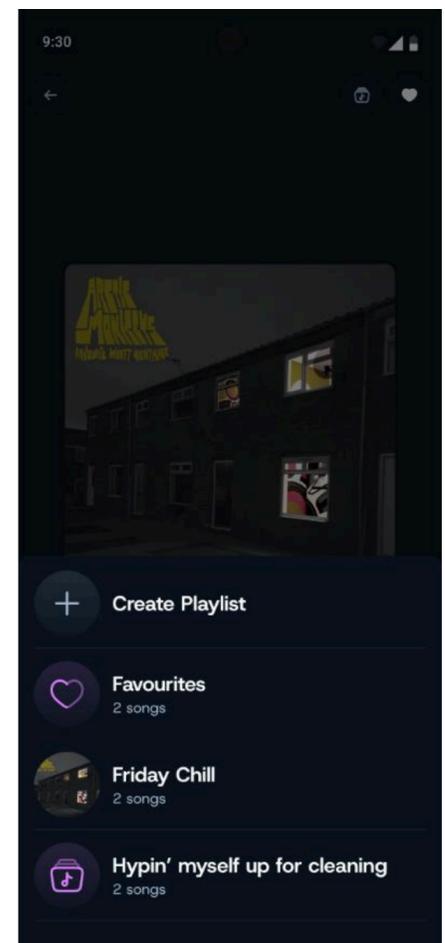
Add to Favourites Icon

- Located in the **top-right corner** of the Now Playing screen.
- Has two visual states:
 - **Inactive** — outlined heart (song not in favourites);
 - **Active** — filled heart (song added to favourites).
- Tapping the icon instantly toggles between states and adds/removes the song in the **Favourites** playlist (stored locally).
- Music playback continues without interruption



Add to Playlist Icon

- Positioned to the left of the Favourites icon in the top bar.
- Opens the **“Add to Playlist” Bottom Sheet**, which includes:
 - **Title:** “Add to Playlist”;
 - **First item:** “+ Create Playlist”
 - Opens the already implemented “Create New Playlist” Bottom Sheet.
 - When **Create** is pressed:
 - The bottom sheet closes;
 - The current song is **added** to the newly created playlist;
 - Playback **continues** without any interruption.
 - **Playlist list:**
 - Displays all **existing user playlists**.
 - Each item includes:
 - **Icon** — either a custom cover image or a default placeholder;
 - **Playlist name**;
 - **Song count** (e.g., “2 songs”).
 - When a playlist is tapped:
 - The current song is added to that playlist.
 - A short **SnackBar message** appears: *“Added to playlist Friday Chill”*.



Bottom Sheet “Playlist Options”

This **Bottom Sheet** appears when the user taps the ... **(three dots)** button next to any playlist on the **Playlist tab** of the main screen. It allows quick management actions for the selected playlist.

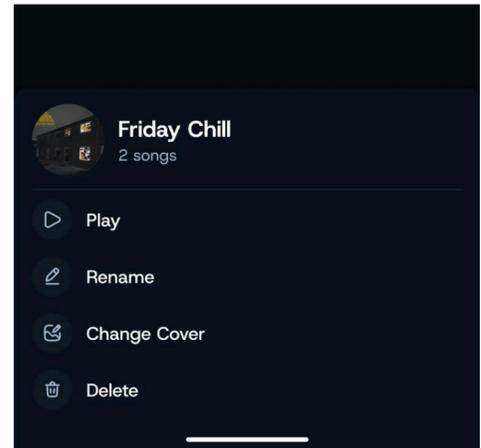


Regular Playlist

The Bottom Sheet for a regular (user-created) playlist includes the following actions:

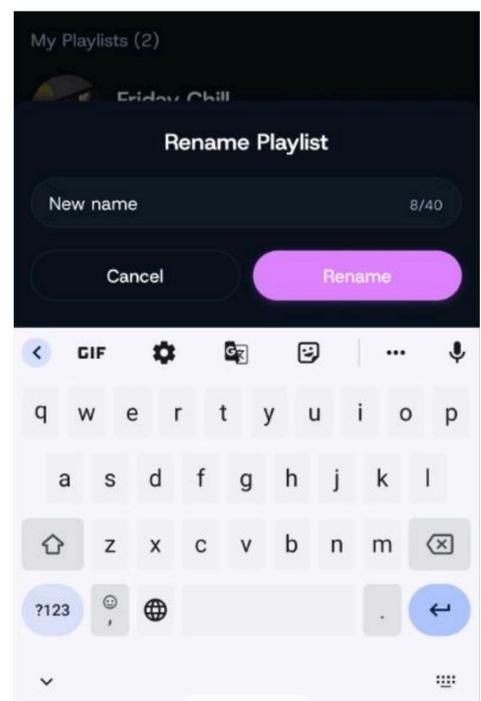
Play

- Opens the **Playlist Playback Screen**, where the user can view and play all songs from the selected playlist.
- Playback starts in the saved order of tracks within that playlist.
- Music continues to play even when navigating to other sections of the app.



Rename Playlist

- Opens the “**Rename Playlist**” Bottom Sheet , which contains:
 - **Title:** “*Rename Playlist*”;
 - **Input field** with the current playlist name pre-filled;
 - **Character counter** in the top-right corner of the field (0/40), indicating a maximum length of 40 characters; once the user reaches the 40-character limit, no additional input is accepted.
 - Buttons **Cancel** and **Rename**.
- Rename button **behavior**:
 - If the name hasn’t been changed — the button is disabled.
 - Once the text differs from the original name — the button becomes active and adopts the accent color.
- When **Rename** is pressed:
 - The playlist name is updated in the local database.
 - The change appears immediately in the playlist list.
 - The Bottom Sheet closes automatically.



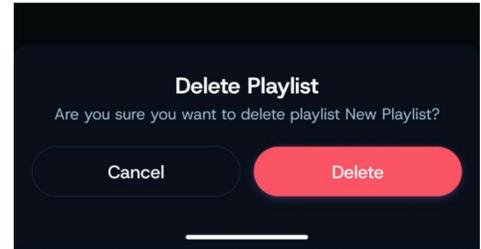
Change Cover

- Opens the **user’s gallery** to select a new cover image for the playlist.
- After selecting an image:
 - The new image is applied instantly as the playlist’s cover.
 - It replaces the placeholder or previous image in the Playlist tab.
- The Bottom Sheet closes right after the selection.

Delete

- Opens the “Delete Playlist” Bottom Sheet for confirmation, which contains:

- **Title:** “Delete Playlist”;
- **Description:** “Are you sure you want to delete playlist [Playlist Name]?” — where [Playlist Name] dynamically displays the actual name of the playlist being deleted.
- **Buttons:**
 - **Cancel** — closes the Bottom Sheet without changes;
 - **Delete** — removes the playlist from the local database and updates the UI.
- **After deletion:**
 - A short **SnackBar message** appears: “Playlist deleted.”
 - The total playlist count updates automatically.

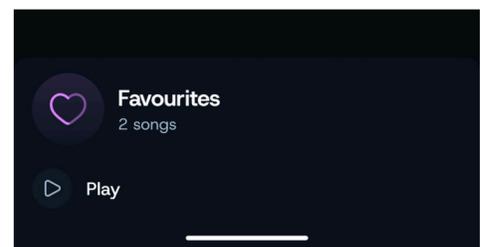


System Playlist “Favourites”

The Bottom Sheet for the system playlist Favourites contains only one action:

Play

- Starts playback of the Favourites playlist.
- Opens the **Now Playing** screen showing the songs in their respective order.



💡 The “Favourites” playlist is a system playlist, so it cannot be renamed, have its cover changed, or be deleted.

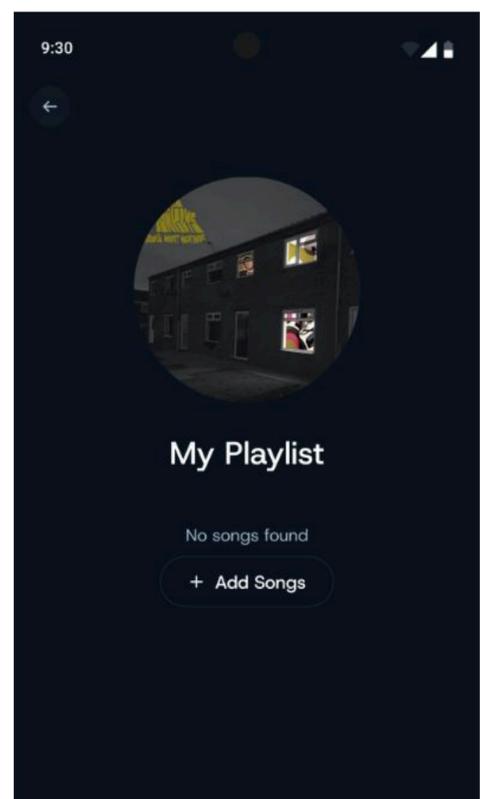
🎵 Playlist Playback Screen

The **Playlist Playback Screen** opens when the user taps **Play** from the *Playlist Options Bottom Sheet* or selects a playlist from the **Playlist tab**. It displays the selected playlist’s cover, title, list of songs, and playback controls.

Empty Playlist State

If the playlist has no songs, a simplified view is shown:

- **Top Bar:**
 - ← **Back** button — returns the user to the **Playlist Tab**.
- **Cover Art:**
 - Displayed at the top in a circular container.
 - If a custom cover is set — it appears here; otherwise, a default **placeholder** (music note icon) is used.
- **Playlist Name:**
 - Shown below the cover, centered on the screen.
- **Text:** “No Songs Found” — displayed under the playlist name.
- **Button “+ Add Songs”** which positioned below the message.

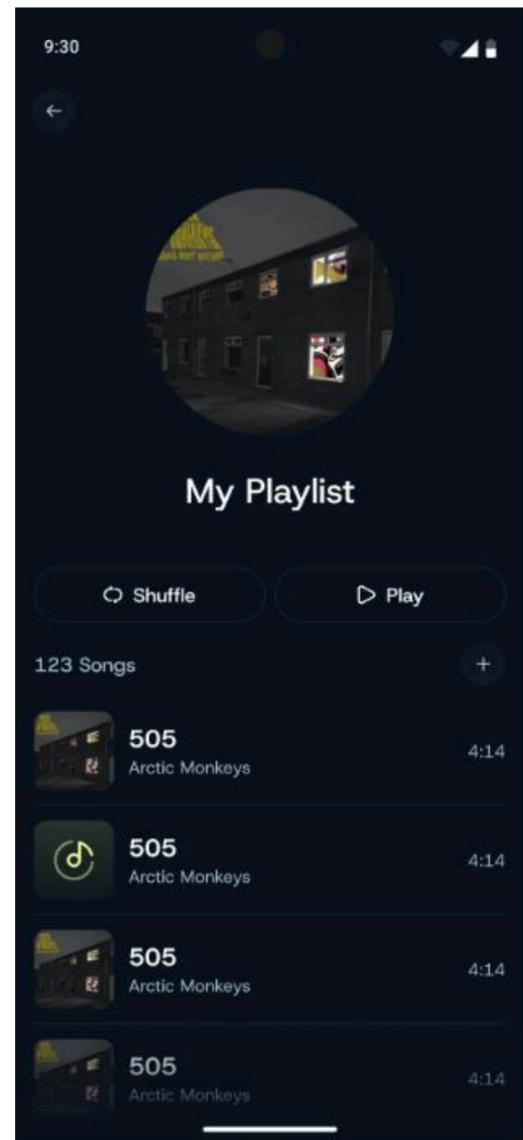


- Tapping it opens the **Add Songs** screen (described in Milestone #3) for adding tracks to the playlist.

Playlist with Songs State

When the playlist contains songs, the following structure is displayed:

- **Top Bar:**
 - ← **Back** button — returns the user to the Playlist Tab.
- **Cover Art:**
 - Displayed at the top center in a circular container.
 - If a cover image exists — it is shown; otherwise, a placeholder icon is used.
- **Playlist Title:**
 - Displayed below the cover art in large font.
- **Playback Buttons:**
 - Two horizontally aligned buttons:
 - **Shuffle** — plays all tracks in random order.
 - **Play** — plays tracks in their saved order.
 - Both buttons open the **Now Playing** screen and start playback of the playlist tracks.
- **Song Counter and Add Button:**
 - Below the playback buttons, the total number of songs is displayed (e.g., “123 Songs”).
 - On the right, a “+” **Add Songs** button opens the Add Songs screen to add more tracks.
- **Track List:**
 - Shows all songs in the playlist, each item including:
 - **Album Art** — from track metadata, or a default placeholder if missing;
 - **Song Title** — displayed prominently;
 - **Artist Name** — secondary text below the title;
 - **Track Duration** — right-aligned in **mm:ss** format.
 - The list is **scrollable** if it exceeds the visible area.
 - The top section (cover, title, Shuffle/Play buttons, and song counter) remains **sticky** while scrolling.



Behavior

- When the user adds new songs via the **Add Songs** screen:
 - The playlist updates automatically after returning to this screen.
 - The song count and list refresh instantly.
 - If the playlist was previously empty, the view switches to the **Playlist with Songs** state.
- All updates are synchronized with the **local database** to ensure data persistence.

Useful Links for This Challenge

- [Create a Splash Screen](#)
- [UX With Material3](#)
- [Full Guide to Material3 Theming](#)
- [Request runtime permissions](#)
- [Introduction to Jetpack Media3](#)
- [Create a basic media player app using Media3 ExoPlayer](#)
- [Media3 ExoPlayer](#)
- [MediaMetadataRetriever](#)
- [Access media files from shared storage](#)
- [Save data in a local database using Room](#)
- [Animations in Compose](#)
- [Photo picker](#)
- [The Full Jetpack Compose Responsive UI Crash Course](#)
- [How to Save & Restore the Scroll Position of a LazyColumn Persistently](#)
- [Stateful vs. Stateless Composables](#)
- [State Hoisting in Compose](#)
- [Managing State in Jetpack Compose \(Codelab\)](#)