

SpotiPod Project Presentation

Step 1: Understanding Why - The Problem Domain

- Many music lovers still use legacy devices like iPods for a distraction-free experience.
- Modern streaming services (e.g., Spotify) do not natively support syncing playlists to these devices.
- Users face a gap: no easy way to transfer Spotify playlists to iPods or similar hardware.

Problems Identified

- 1. No simple way to convert Spotify playlists for iPod use**
- 2. Manual conversion is tedious and error-prone**
- 3. Metadata and organization are often lost in the process**

Step 2: From Problems to Features

- **Problem 1 → Feature 1:** Convert Spotify links to downloadable audio files
- **Problem 2 → Feature 2:** Preserve metadata (title, artist, album, art)
- **Problem 3 → Feature 3:** Organize files for easy syncing to devices

Features Overview

- Input Spotify track or playlist URL
- Retrieve and convert tracks to iPod-compatible formats (MP3, AAC)
- Store files locally, organized by playlist or artist
- Show progress and handle errors gracefully

Requirements Breakdown

- **Functional:**
 - Accept Spotify URLs
 - Retrieve metadata via Spotify API
 - Convert and store audio files
 - Handle invalid links robustly
- **Non-Functional:**
 - Efficient conversion
 - Simple, intuitive UI
 - Error logging for debugging

Example User Stories

- As a user, I want to input a Spotify playlist URL so I can transfer my music to my iPod.
- As a user, I want the app to preserve song details and album art.
- As a user, I want to see conversion progress and be notified of any issues.

Implementation Plan

Sprint 1 (Required Features):

1. Fetch track metadata from Spotify for formatting locally
 - R.1. Integration with Spotify API for song, album, artist, playlist endpoints
 - R.2 Format that metadata into usable objects for local
2. Find available music formats based off of track metadata.
 - R.1 yt-dlp must be able to download formats based on ISRC metadata
 - R.2 Apply the metadata from Spotify to the local music file
 - R.3 Formats must be able to be stored locally in an efficient matter

Sprint 2 (More non-trivial features):

3. Automatic integration with Apple Music syncing to iPod

R.1 User must have Apple Music or iTunes installed in order to interface with iPod

R.2 Files must be able to be moved to specific directories for syncing.

4. A clean UI/UX experience through a deployed App

R.1 App must be easy to navigate and have good modularity.

R.2 Efficient error handing for user error or app error.

R.3 Settings must be readily available to the user for format or general app controls.

R.4 App must be packaged via Electron, or a similar technology.

Testing Approach

- Unit tests for API handling, URL validation, and file naming
- Integration tests for end-to-end conversion
- Edge case tests (invalid links, empty playlists)
- Manual testing for UI and device syncing

Summary

- SpotiPod bridges the gap between modern streaming and classic music devices
- Focused on usability, reliability, and preserving the music experience
- Open for feedback and future enhancements

Thank you!

Questions?