

Spotify Companion

Dominik Scheffler
Philipp Steib

November 22, 2020

Contents

1	Overview	3
1.1	Project Intention	3
1.2	Current State	3
1.3	Desired State	3
1.4	Similar Apps	3
2	Requirements	4
2.1	Must Have	4
2.2	Should Have	4
2.3	Could Have	4
2.4	Won't Have	4
3	Non-functional requirements	5
4	Architecture	6
5	GUI Prototyping	7
5.1	Collector Mode	8
5.2	Refinement Mode	10

1 Overview

1.1 Project Intention

The Project revolves around the automation of Playlist Modification via the Spotify-AppRemoteAPI. The general idea is the automatic removal of commonly skipped songs and addition of those the User appears to favor.

The main objective is making the Process of creating playlists less tedious and possibly improve some other aspects, mainly quality of life additions.

1.2 Current State

No possibility to create and add to existing playlists automatically nor to remove frequently avoided tracks. The official app often has loading times when selecting song-options such as queuing and adding to a list.

1.3 Desired State

The user barely ever has to adjust the lists manually after creation, since the companion manages removal and addition of titles accordingly.

1.4 Similar Apps

"Companion For Spotify" exists but is more focused on presenting changes to playlists and newly released titles. It does allow for concatenation of playlists, this is a feature we could add for convenience but did not include in the original mockup.

2 Requirements

2.1 Must Have

- (M3) comprehensible UI
- (M1) removal of repeatedly skipped songs
- (M2) adding not skipped songs to chosen playlist

2.2 Should Have

- (S4) intuitive and clean UI conforming to common usability standards found in most prevailing apps.

2.3 Could Have

- (C5) custom app logo
- (C6) better car-mode-player
 - (C7) interface with track-playtime-slider
 - (C8) improve suggested lists (list of favorites)
 - (C9) add buttons for specific songs and lists
 - (C10) show next title

2.4 Won't Have

- (W11) standalone player, the app is meant to accompany and control the official application

3 Non-functional requirements

This section lists the non-functional requirements. These describe non-technical requirements as well as physically intangible requirements.

- Usability:
The application must ensure that the controls can be used quickly and intuitively.
- Error avoidance
The application must ensure that accidental error made by the user can be made undone and, if possible, can be avoided.
- Clarity
Each view within the application must not be overloaded with too much information.
- App design guidelines
The app must comply with the design guidelines of the marketplaces "Play Store (Google)" and "App Store (Apple)". Furthermore it must comply with the branding guidelines of Spotify (Found here: [Branding guidelines](#))
- Compatibility
The app must be compatible to the API, rules and guidelines of the Spotify music platform.
- Authenticity
The app must ensure that all actions are bound to the authenticated user profile.

4 Architecture

For the implementation we will be using the "Spotify Android SDK" and the "Spotify App Remote Library".

The Companion will run alongside the official application, all actions regarding connection, audio playback, resource fetching and database updates will still be handled as they have been previously.

We only manage the requests for Spotify to perform certain actions.

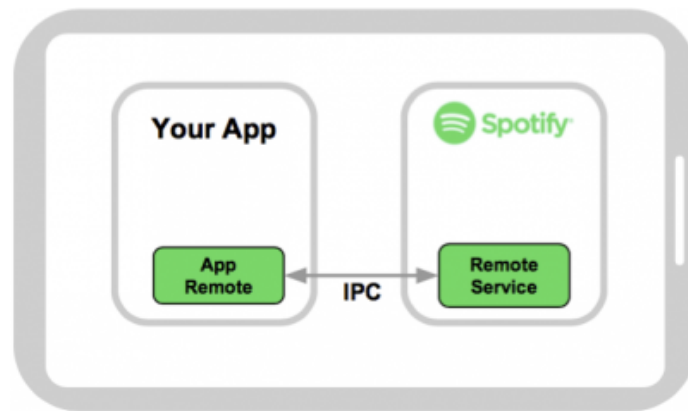


Figure 1: Spotify App Remote Library [1]

The image above displays how the "App Remote Library" is intended to be used. Our architecture will be based upon that.

5 GUI Prototyping

A clickable prototype has been developed to combine a general mockup with a more specific design- and color guide in mind. This accelerates development cycles with less reiterations needed to match the developed design guide.

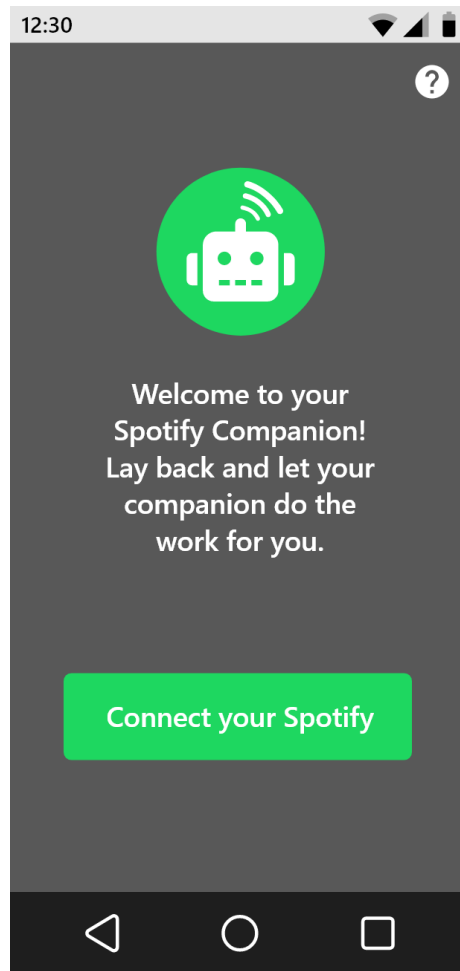


Figure 2: First Welcome Screen

The image above displays how user is greeted and has the option to launch the companion and connect it to an active Spotify-App.

5.1 Collector Mode

The Collector Mode is used when the user browses foreign playlist or mixes. If the user listens a track for a set time, the track will be added to a set playlist.

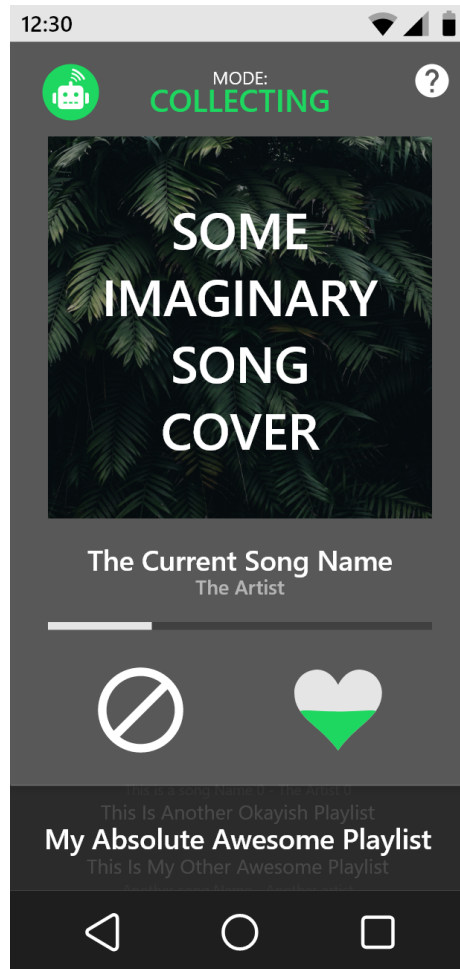


Figure 3: Collector Mode - short listening time

The demonstrated scenario shows the app with an active track in Collector-Mode, when the user did not reach the threshold for the companion to add said track to the selected playlist. The user can manually toggle the "add to playlist"-queue with a press on the heart button. If the user skips the track, it will not be added to the playlist.

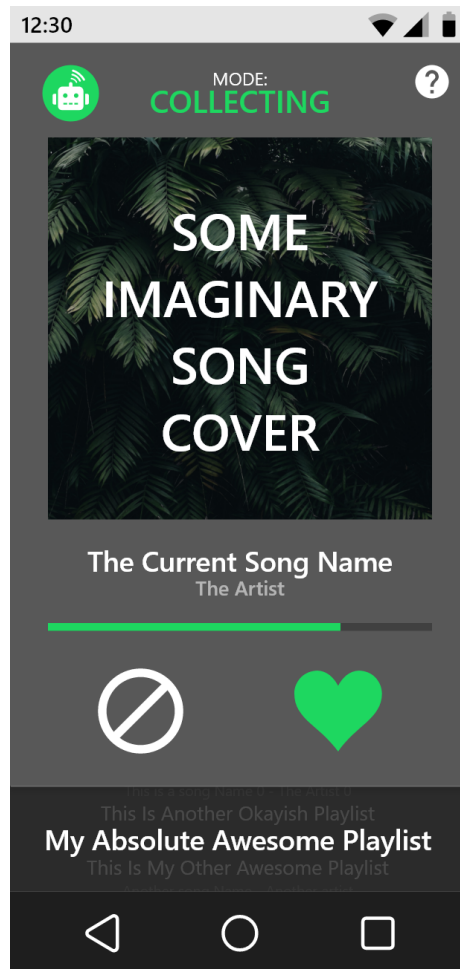


Figure 4: Collector Mode - long listening time

The demonstrated scenario shows the app with an active track in Collector-Mode, when the user did reach the threshold for the companion to add said track to the selected playlist. When the track ends it will be added to the active playlist. The user can prevent that behaviour by pressing the restraint button on the left side and the track will be ignored.

5.2 Refinement Mode

The Refinement Mode is used when the user plays back owned playlists. If the user skips a track before a set time, the track will be removed from the playlist.

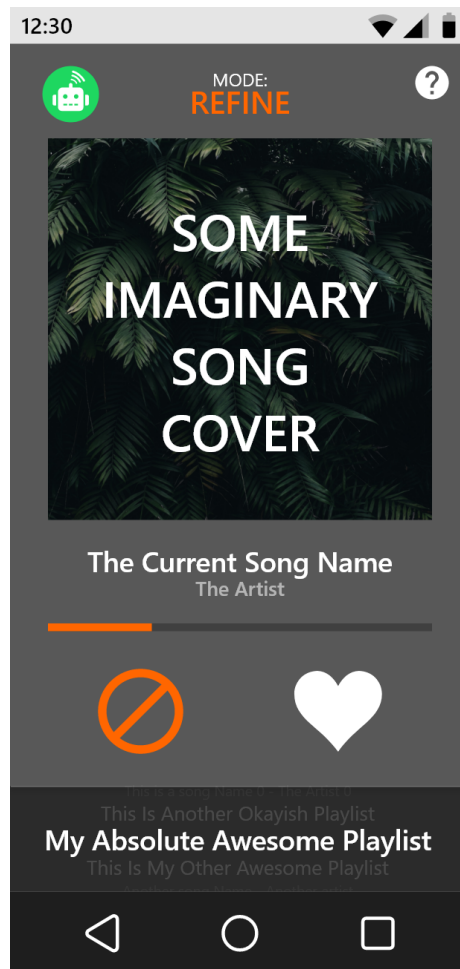


Figure 5: Refinement Mode - short listening time

The demonstrated scenario shows the app with an active track (played from a personal playlist) in Refinement-Mode, when the user did not reach the threshold for the companion to keep said track in the selected playlist. When the track is skipped it will be removed from the played playlist. The user can prevent that behaviour by pressing the heart button on the right side and the track will be ignored.

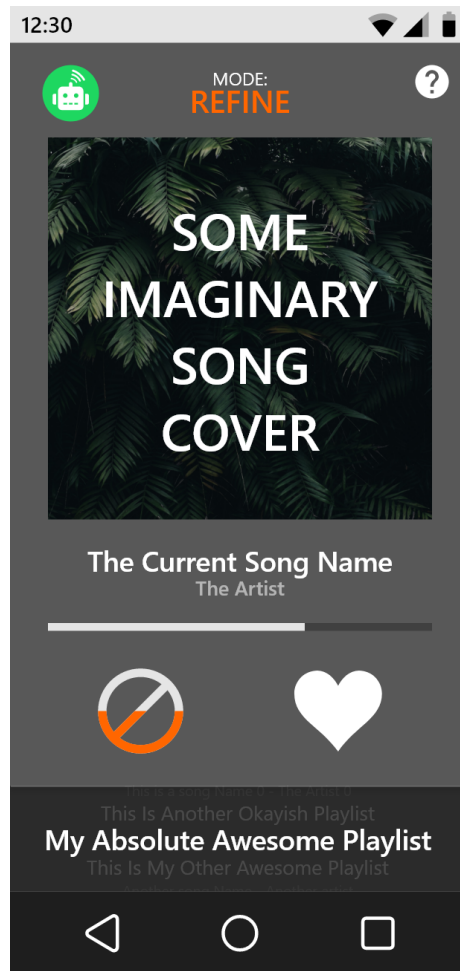


Figure 6: Refinement Mode - long listening time

The demonstrated scenario shows the app with an active track (played from a personal playlist) in Refinement-Mode, when the user did reach the threshold for the companion to keep said track in the selected playlist. When the track is skipped or ends it will stay in the played playlist. The user can prevent that behaviour by pressing the restraint button on the right side. The track will be removed from the list.

References

- [1] Android api. taken from "<https://developer.spotify.com/documentation/android/>" on 18.11.2020.