

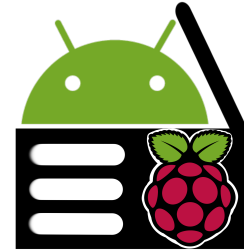


bickoSiTiEff

office@bickositieff.me
<https://bickositieff.me>

Raspio

25. Mai 2021



Übersicht

Controller-Software um Radios mit neuesten Smartphones kabellos zu steuern. Mithilfe eines Raspberry Pi und einer Antenne können so Radios in der Umgebung verknüpft werden. Mit gewohnten Playlist-Funktionen kann spielend leicht zwischen Songs navigiert werden.

Ziele

1. **Einfaches Setup:** Der Endnutzer sollte nur das SD-Karten Image downloaden, es auf eine SD-Karte übertragen und den Raspberry Pi einschalten müssen. Alles andere sollte innerhalb der App eingerichtet werden.
2. **Intuitive Benutzeroberfläche:** Alle Funktionen sollen leicht erreichbar sein und das Design sollte sich an den Material-Design Richtlinien orientieren.
3. **Integration in andere Dienste:** Es sollten möglichst viele Dienste als Quelle für Musik unterstützt werden.

Sprints

1. **Initial Sprint** 25. Mai - 1. Juni

Im Initial Sprint wird das Projekt geplant und wichtige Dokumente erstellt. Weiters werden die für die Projektdurchführung wichtigen Dienste, wie YouTrack und GitHub, konfiguriert und leere Projekte für die Server-Komponente und die App angelegt.

2. **Basic Features** 1. Juni - 8. Juni

Im Basic Features Sprint werden Basisfunktionen, wie das Abspielen von Songs über den Audioausgang am Server, sowie die grobe UI und Funktionalität der Android-App erstellt.

3. **Featureschleuder** 8. Juni - 22. Juni

Bei der Featureschleuder werden die restlichen komplexeren Funktionen, wie der Einrichtungsassistent und das Übertragen des Audios über Radio, implementiert und zusätzliche Features hinzugefügt.

4. **Polishing** 22. Juni - 29. Juni

Im Polishing Sprint wird das Design und die User Experience verbessert sowie Bugfixes durchgeführt. Weiters werden die Tech-Demo und Präsentation des fertigen Produktes vorbereitet.

Arbeitspakete

Initial Sprint

- README erstellen
- Setup Android App
- Setup basic REST server
- Projekt-Tasks planen
- Projektantrag fertigstellen
- Projekt Setup
- Design Mockups
- Projekt-Logo Design
- Write API Specification

Basic Features

- Android basic UI
 - Playlist
 - Server-Management
- Server (am Raspberry Pi)
 - API für MPD Befehle
 - MPD starten, wenn der Server gestartet wird
- Client Funktionalität
 - Lieder abspielen (Start und Stop)
 - Server IP eingeben

Featureschleuder

- Einstellungen
 - Server
 - Frequenz steuern
 - Client
 - UI-Theme
 - FM-Transmitter Start/Stop
 - Automatisch verfügbare Server suchen
- FM-Transmitter (Server)
 - Virtuelles Audio Gerät erstellen (mit Pulse Audio)
 - FM-Transmitter starten mit virtuellem Audio Gerät als Eingang
 - MPD konfigurieren mit virtuellem Audio Gerät als Ausgang
- Song-Download (client-side)
 - (Suchfunktion mit verschiedenen Diensten)
- Songs empfangen (server-side)
- Setup-Assistent

Polishing

- User-Experience
- Animationen
- Bug-Fixes
- Erstellung einer Projekt-Homepage mit Downloads
- Erstellung fertiger SD-Karten-Abbilder für Server-Komponente
- GitHub Release
- Präsentation

Machbarkeitsstudie

Eine technische Machbarkeitsstudie wurde von unserem Entwicklerteam bereits erfolgreich durchgeführt.

Alle Rechte sind bickoSiTiEff vorbehalten.

Wir weisen darauf hin, dass diese elektronische Post einschließlich aller Anlagen, insbesondere auch Dateianlagen, unser geistiges Eigentum ist und ohne unsere schriftliche Zustimmung nicht abgeändert werden darf.

Bei rechtlichen Problemen wenden Sie sich an legal@bickositieff.me.