

Software Requirements Specification (SRS)

Sastha Spotify

Prepared by: Aadarsh T Rajeev

1. Introduction

1.1 Purpose

The Sastha Spotify app is designed for music streaming, providing users with an intuitive interface to explore, play, and manage their favorite tracks. It aims to deliver a seamless audio experience, with robust performance and easy navigation.

1.2 Scope

The app enables users to:

- Browse and search for music.
- Stream tracks in real time.
- Create and manage playlists.
- Enjoy offline music by downloading songs.

The app is developed for Android platforms using Java, XML, and Firebase for real-time updates and storage.

1.3 Definitions, Acronyms, and Abbreviations

- SRS: Software Requirements Specification
- GUI: Graphical User Interface
- API: Application Programming Interface

2. Overall Description

2.1 Product Perspective

Sastha Spotify is a standalone mobile app that interfaces with a backend (Firebase) for storing user preferences, playlists, and track metadata.

2.2 Product Features

- Music Search: A search bar for finding tracks.
- Playlists: Options to create, edit, and manage playlists.
- Real-time Streaming: Access a vast music library.
- Offline Mode: Download tracks for offline listening.
- User Profiles: Personalized settings and preferences.

2.3 Operating Environment

- Platform: Android
- OS: Android 6.0 or later
- Internet Access: Required for streaming; optional for offline mode.

3. System Features

3.1 Music Streaming

Description: Stream songs in real time.

Inputs: Song selection by users.

Outputs: Continuous audio playback.

3.2 Playlist Management

Description: Create and manage playlists.

3.3 Offline Mode

Description: Enable downloading of songs for offline playback.

4. External Interface Requirements

4.1 User Interfaces

- A clean, intuitive UI for music browsing and playback.
- A search bar and personalized recommendation tiles.

4.2 Software Interfaces

- Backend: Firebase for real-time data synchronization.

5. System Requirements

5.1 Functional Requirements

- Play, pause, and skip tracks.
- Save and retrieve playlists from Firebase.

5.2 Non-Functional Requirements

- Usability: User-friendly UI.
- Performance: Songs should load with minimal delay.