George Puliafico | Steven Gurney | Vincent DiBlasio | Christopher Bartoli

LinkList Software Requirements Document

1. Introduction

LinkList is an Android application intended to provide a platform for Streamlined Collaborative Playlists (SCP). Users will join a lobby started from a host device where they will select songs to create a queue to be played through the audio channel of the host.

2. User Requirements

2.1 Software Interfaces

- Android Development Kit (ADK) (Android Studio 2.2.1.0 IDE)
- Spotify Android SDK (ASDK)
- C++ XMLRPC - C library
- Spotify Web Wrapper API
- Android XML-RPC library

2.2 User Interfaces

ADK - Ease in the development of both user interface and client side server connection. The development kit includes access to several android-specific Java APIs and standard libraries. The user interface will streamline all interaction between the user and the client side application as well as the network connection running in the background of the app.

Spotify SDK - Interactive access to Spotify’s database delivered through the use of the Spotify SDK, otherwise data from Spotify’s databases would be protected from external use. Users will only need to interact with this portion of the application to authenticate themselves and use the media player provided (and required) by spotify.

Spotify Web Wrapper API - Allows access to song database and associated metadata. This will allow users to search for what songs they want to play, since the SDK doesn’t support that.

C++ XML-RPC Library - Used to communicate with the app,

2.3 User Characteristics

The product is intended to be no more complex than a standard music playing/streaming app, with a very small learning curve, anyone should be able to host or join a session using interactive buttons from the ADK. The user interface is meant to simplify the overall experience by using graphical input rather than text based inputs. Users will interact via the these Android features and more complex actions will be taken in the background of the app. Tutorial based help may be included in final build to aid in first time user experiences.

2.4 Assumptions and dependencies

- Availability and dependency of an internet connection.
- Reliability and restrictions imposed by Spotify.
- The software is intended to run on the Android OS, and functionality of the application will not extend to other operating systems at this time.
- Spotify SDK requires the application to be compatible with all versions of android since ICS (API level 14, Android version 4.0)
- Spotify Premium account is required to use the Host functionality (and possibly the search as well).

3. System Requirements

These subsections contain all the software requirements at a level of detail sufficient enough to enable designers to design a system to satisfy those requirements, and tests to test that the system satisfies those requirements.

3.1 Functional Requirements

There must be an app and a server.

- These systems must be able to communicate.

The user(in app) must be able to start a room(on server) that other users can join.

- This room must have a list of current users and a list of songs.
- The song list must be able to be changed by the users
- The host app must display song information
- The client apps must display queue information

The server must be able to serve many groups at once.

The app must be able to call for information from the group they are in.

The host app must use the song list to determine which song to play.

3.2 Non-Functional Requirements

The user must have a functional speaker.

We are limited by a need for internet connection for hosts and users.

Because of server resource constraints we must have lobbies that communicate with a limited number of devices.

The users must be able to get invite keys/codes to each other, this will not be done in app

3.2.1 Software Quality Attributes

There must be a consistent connection between the host device and the server in the case of streaming music from the server. If the device will stream directly from another service, then we will expect a consistent internet connection to allow seamless audio streaming.

For the most part we will not be sending private or confidential information, so security concerns will be focused on keeping uninvited users out of private lobbies. This will most likely involve a password or entrance by invitation only, both of which will be communicated with encryption.

List the quality attributes that are important for your system (e.g., reliability, security) and explain why they are important for your system and how they will be measured.

Reliability is important so that users will not have the servers crash while they are trying to use our service. This means that we will have to have a very high percentage of uptime. Also, the client and server interfacing must be seamless so that requests aren’t lost while being sent.