# Amplify Software Requirements Document

Alexia Filler, Ben Kern, Ian Lizarda, Donovan Moini and Serena Zafiris

Table of Contents

## 5.1 Introduction

### 5.1.1 Amplify will be a community music streaming service that will allow users to create and join rooms that host public music playlists using Spotify and YouTube services, similar to platforms like plug.dj and Outloud. Users will be able to join rooms and add songs of their choice from Spotify or YouTube to the community queue, which will be cycled through in round-robin style between all users' queued songs. Amplify will also have features useful to an individual user; the user will be able to import multiple playlists and rotate through those playlists via round-robin style. The frontend will use React and Typescript. The backend will use JavaScript, Apollo GraphQL, Amazon Cognito, Amazon Relational Database Services, Amazon Gateway, and Amazon Lambda.

### 5.1.2 Diagrams

Figure 1: Frontend Diagram

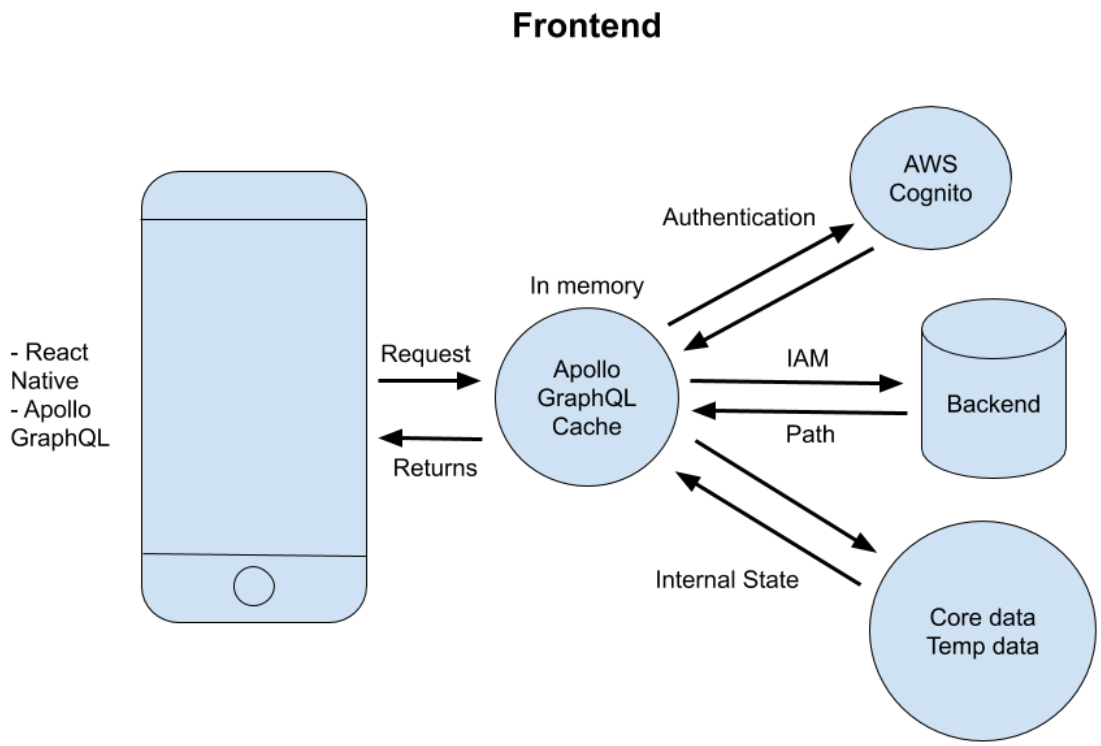


Figure 2: Backend Diagram

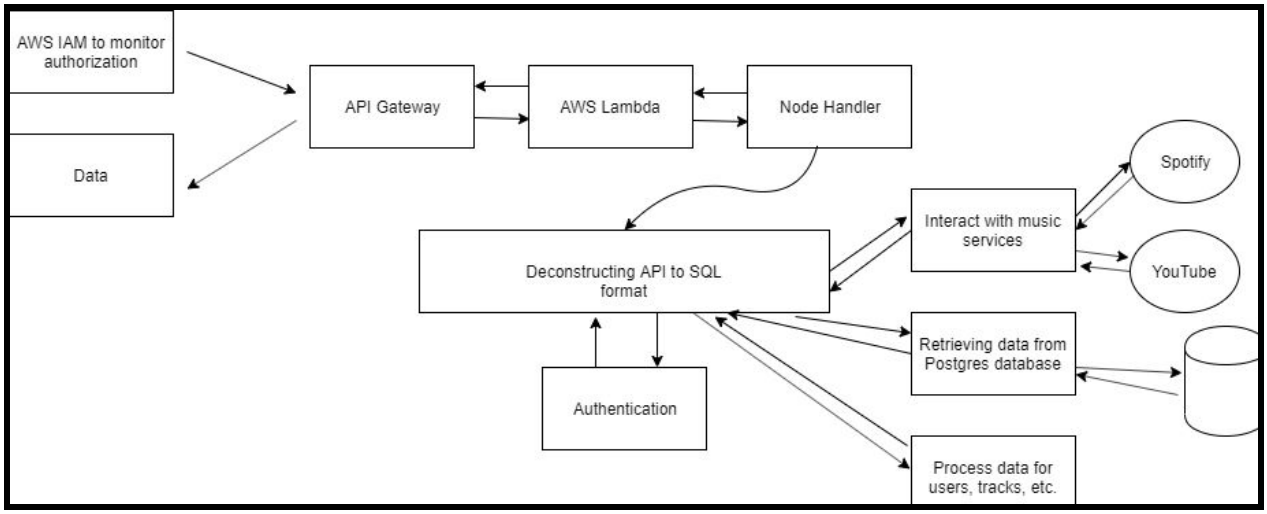
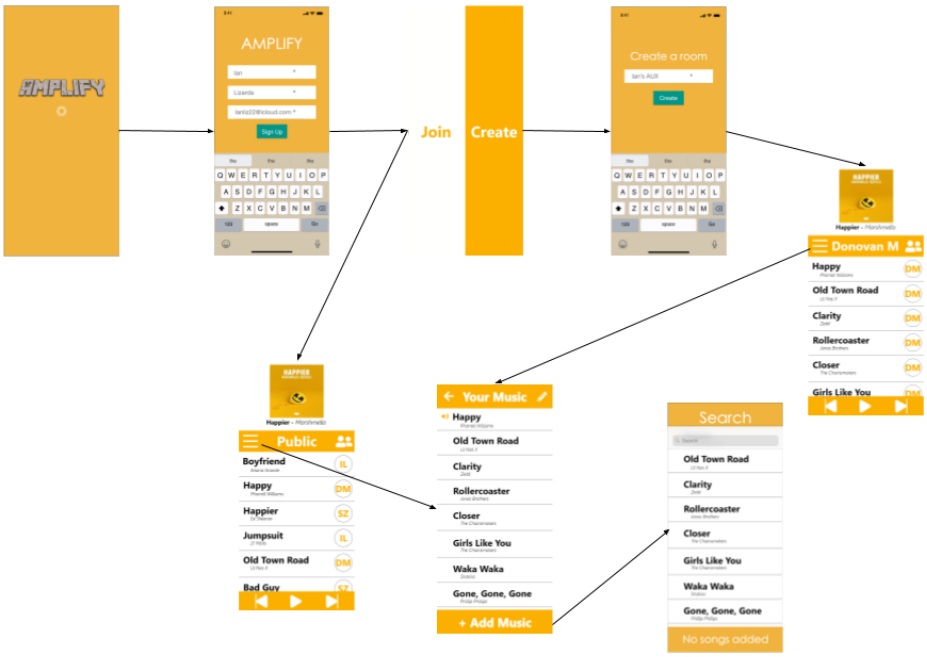



Figure 3: Frontend User Interface Diagram




## 5.2 CSCI Component Breakdown

### 5.2.1 Frontend Components

#### 5.2.1.1 PlayerView Component - Creates the GUI for the room

#### 5.2.1.2 CreateJoinView Component - Allows a user to create or join a room.

5.2.1.3 CreateUserView Component - Allows a user to create a profile.
5.2.1.4 LoadingView Component - Loading Screen for transitions between components
5.2.1.5 SearchView Component - Search screen for songs
5.2.1.6 ListView Component - List structure for songs

5.2.2 Backend Components

5.2.2.1 searchTracks Endpoint - Returns tracks based on search query.
5.2.2.2 getUser Endpoint - Returns user based on user ID.
5.2.2.3 getRoomUsers Endpoint - Returns list of user IDs in a room.
5.2.2.4 getCurrentRoom Endpoint - Returns current room of given user.
5.2.2.5 getCurrentRoomTracks Endpoint - Returns current master queue of current room of given user.
5.2.2.6 addTrack Endpoint - Adds track to queue.
5.2.2.7 createUser Endpoint - Creates a user.
5.2.2.8 createRoom Endpoint - Creates a room.
5.2.2.9 joinRoom Endpoint - Adds a user to a room.
5.2.2.10 leaveRoom Endpoint - Removes a user from a room.
5.2.2.11 createUserRoomInfo Endpoint - Creates user room information for room to store as property.
5.2.2.12 Database - Stores data for rooms, users, and tracks for users to access and update.

## 5.3 Functional Requirements

### 5.3.1 Frontend Functional Requirements:

5.3.1.1 PlayerView

5.3.1.1.1 The PlayerView shall provide a button that plays the music.
5.3.1.1.2 The PlayerView shall provide a button that pauses the music.
5.3.1.1.3 The PlayerView shall provide a button that skips to the next song.
5.3.1.1.4 The PlayerView shall provide a button that restarts the song.
5.3.1.1.5 The PlayerView shall provide an image of the songs album cover.
5.3.1.1.6 The PlayerView shall provide a list of the current songs in the track.

5.3.1.2 CreateJoinView

5.3.1.2.1 The CreateJoinView shall provide a button to create a room.

5.3.1.2.2 The CreateJoinView shall provide a button to join a room.

5.3.1.3 CreateUserView

5.3.1.3.1 The CreateUserView shall provide a text input for the user's first name.

5.3.1.3.2 The CreateUserView shall provide a text input for the user's last name.

5.3.1.3.3 The CreateUserView shall provide a button to create the user.

5.3.1.4 LoadingView

5.3.1.4.1 The LoadingView shall display a screen with the Amplify logo on it during loading sequences.

5.3.1.4.2 The LoadingView shall query if the current user exists.

5.3.1.4.3 The LoadingView shall navigate to the CreateUserView if the current user does not exist.

5.3.1.4.4 The LoadingView shall navigate to the CreateJoin room if the user exists.

5.3.1.5 SearchView

5.3.1.5.1 The SearchView shall have a search bar for the users to look for songs.

5.3.1.5.2 The SearchView shall have a list of the prompted search.

5.3.1.6 ListView

5.3.1.6.1 The ListView shall create the template for any list of songs.

5.3.2 Backend Functional Requirements:

5.3.2.1 GraphQL Backend (GB)

5.3.2.1.1 The GB shall provide an endpoint to view a user's properties.

5.3.2.1.2 The GB shall provide an endpoint to view all the active users's user IDs.

5.3.2.1.3 The GB shall provide an endpoint to view all the active rooms' roomIDs.

5.3.2.1.4 The GB shall provide an endpoint to create user information for the room to store.

5.3.2.1.5 The GB shall provide users with an endpoint to search for songs from Spotify.

5.3.2.1.6 The GB shall provide users with an endpoint to search for songs from YouTube.
5.3.2.1.7 The GB shall provide users with an endpoint to create a user.
5.3.2.1.8 The GB shall provide users with an endpoint to create a room.
5.3.2.1.9 The GB shall provide users with an endpoint to join a room.
5.3.2.1.10 The GB shall provide users with an endpoint to view the current room's queue.
5.3.2.1.11 The GB shall provide users with an endpoint to view the current users in the current room.
5.3.2.1.12 The GB shall provide users with an endpoint to view current room's users' personal queues.
5.3.2.1.13 The GB shall provide users with an endpoint to add tracks to the user's personal queue.
5.3.2.1.14 The GB shall provide users with an endpoint to add tracks to the user's current room's queue.
5.3.2.1.15 The GB shall provide users with an endpoint to leave a room.

## 5.4 Performance Requirements

### 5.4.1 Frontend Performance Requirements

5.4.1.1 No functionality in the MA shall take more than 100 milliseconds to receive feedback from the application.
5.4.1.2 The application shall not crash on completion of any basic, expected feature in the application.

### 5.4.2 Backend Performance Requirements

5.4.2.1 No endpoint in the GB shall take more than 5000 milliseconds to reach full completion.
5.4.2.2 The application shall not crash, fail, or return an error after receiving valid inputs for any given function.
5.4.2.3 The application shall return an error after receiving invalid inputs for any given function.

## 5.5 Project Environment Requirements

### 5.5.1 Development Environment Requirements

5.5.1.1 Amplify development will require a computer that is able to support an iPhone or Android emulator, or an iPhone or Android device to test on.

5.5.1.2 Amplify development will require a computer that is able to run a Node.js server, and support React Native development, preferably through an IDE.

5.5.1.3 Amplify development will require a computer that is able to run TypeScript.

5.5.1.4 Amplify development will require a computer that is able to run a Postgres DB instance for a local database for development purposes.

5.5.2 Execution Environment Requirements

5.5.2.1 Amplify will be downloadable from the Apple App Store and Google Play Store, and will run on any device that can download from these stores.