

Project Backlog

Iguana Team

**Alec Gorge, Scott Opell, Andrew Shildmyer, Kirby
Kohlmorgen, Manik Kalra**

(Team 1)

Problem Statement

We will add user accounts and playlist support to the existing Iguana Suite and update the iOS and web interfaces accordingly. We will add alternate interfaces such as a Chrome extension and possibly a Chromecast application. The Iguana Suite is a pre-existing suite of software developed by Alec Gorge that provides programmatic access to hundreds of artists and hundreds of thousands of songs. The Iguana Suite will be better than the existing Archive.org interface by allowing filtering and sorting of concerts.

Background Information

The Iguana Suite is an existing API server, simple web app and simple iOS app built by Alec Gorge. The app has over 35k installations and is used thousands of times per day and streams over 100k songs by more than 80 artists every month. People using the app (Listen to the Dead, LotusOD) and website (lotusod.com, marcoallday.com) are fans of the band or music buffs who are interested in finding new live music. Music is legally sourced from the Live Music Archive on Archive.org. Archive.org has a very poor interface and Iguana represents a significant improvement in many of those core tenets. However, the functionality is currently very basic and there is significant room for value-added features such as Chromecast integration, user accounts, fluid browsing between multiple artists,

playlists and more.

There are no applications that are targeted specifically at the Live Music Archive. There are other iOS apps that act as a client to Archive.org, but they suffer from many of the same issues due to poor data organization on Archive.org. Iguana address this issue by collecting all the data in a database and then going through a process that cleans up, duplicates and categorizes the data in a way that is very useful for browsing and listening to music.

Requirements

Functional Requirements

- [] As a user, I would like to access all the artists available via the API in one iOS app.
- [] As a user, I would like to sign into my relisten.net account in the iOS app.
- [] As a user, I would like to be able to sign up for a relisten.net account in the iOS app.
- [] As a user, I would like to have a playlist in the iOS app.
- [] As a user, I would like to have the ability to share what I'm listening to in the iOS app.
- [] As a user, I would like to have the option to favorite artists, shows, and songs on the iOS app.
- [] As a user, I would like to have an easy-to-use, user-friendly iOS interface.
- [] As a user, I would like to be able to manage playback of music in a reorderable queue in the iOS app.
- [] As a user, I would like to have gapless playback in the iOS app.
- [] As a user, I would like to be able to play, pause, fast-forward, rewind and seek to control my music playback in the iOS app.
- [] As a user, I would like to be able to manage playback of music in a reorderable queue in the web app.
- [] As a user, I would like to be able to play, pause, fast-forward, rewind and seek to control my music playback in the web app.
- [] As a user, I would like to have gapless playback in the web app.
- [] As a user, I would like to access all the artists in one web app.

- ☐ As a user, I would like to be able to sign into my relisten.net account in the web app.
- ☐ As a user, I would like to be able to sign up for a relisten.net account in the web app.
- ☐ As a user, I would like to have playlist support in the web app.
- ☐ As a user, I would like to have the ability to share in the web app.
- ☐ As a user, I would like to have the option to favorite artists, shows and songs on the web app.
- ☐ As a user, I would like to have an easy-to-use, user-friendly web interface.
- ☐ As a user, I would like to have a chrome extension to control website playback.
- ☐ As a user, I would like to be able to cast my music playback to Google Chromecast.
- ☐ As a user, I would like to access all the artists in one OS X app.
- ☐ As a user, I would like to have an easy-to-use, user-friendly OS X interface.
- ☐ As a user, I would like to be able to manage playback of music in a reorderable queue in the OS X app.
- ☐ As a user, I would like to have gapless playback in the OS X app.
- ☐ As a user, I would like to be able to play, pause, fast-forward, rewind and seek to control my music playback in the OS X app.

Non-functional Requirements

- ☐ As a user, I would like to have fast response times and efficient use of bandwidth.
- ☐ As a sysadmin, I would like the API server to be efficient with the use of server resources.
- ☐ As a sysadmin, I would like for the database and the API server to be able to hosted on separate machines for scaling.
- ☐ As a sysadmin, I would like one average machine be able to serve 30 req/sec.
- ☐ As an administrator, I would like the API to store information about FLAC and MP3 tracks.
- ☐ As a sysadmin and developer, I would like the API server to include

database migrations to easy deployment.

- ☐ As a developer, I would like to have a staging API server and a production API server.
- ☐ As a developer, I would like to deploy to staging or production in one command.
- ☐ As a developer, I would like it to be easy to run a local development server.
- ☐ As a developer, I would like it to be easy to connect the iOS app to a different API server (staging, production, IP address, etc).
- ☐ As a developer, I would like it to be easy to connect the web app to a different API server (staging, production, IP address, etc).
- ☐ As a developer, I would like the API methods to have publically available documentation.
- ☐ As a developer, I would like the show reviews to be queryable and stored in a separate table.