Collabify

Follow along with our demo by downloading Collabify www.
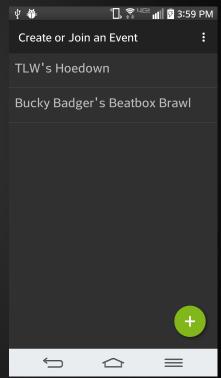collabify.space

**David Gardner, David Hartman, Brandon Hill, Sydney Johnson,
Ricardo Lopez, Andrew McOlash, Erik Olson**

Specification and Design

- Manage a collaborative spotify playlist
- Login with spotify
- Create/join events
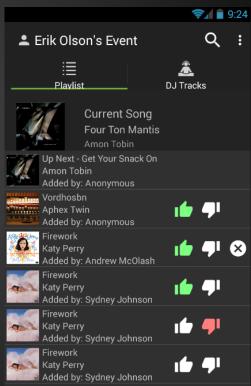- Node js server with MongoDB

Specification and Design

Collabifier (joined an event)

- Add song from spotify search or DJ's public playlists

- View songs in the playlist
  - Upvote/downvote on any song
  - Delete songs they've added

Specification and Design
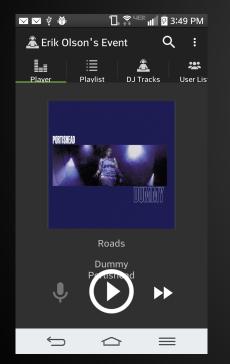
DJ (created the event)

- Play/pause/skip songs

- Mic functionality

- Control over song order in playlist

- Delete any song in playlist

- Add song from search or

  own public playlists

- View users at their event

  - Promote/blacklist users

  - Change event name
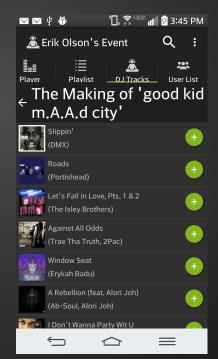 - Turn votes on/off

Accomplishments

- Met all aforementioned specifications
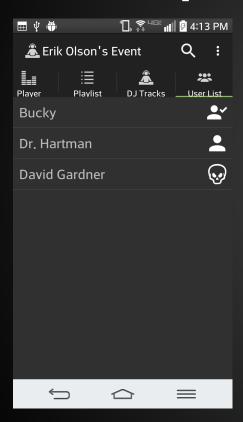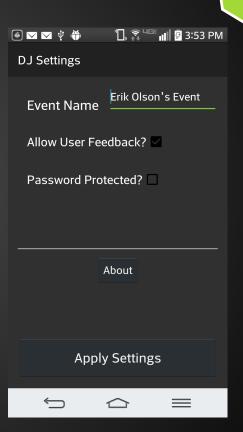
Accomplishments

Shortcomings

- No TLS between backend and application
- Saving the Collabify playlist as a spotify playlist once the event is over
- Song lyrics button
- Automatic updates using WebSockets

Lessons Learned

- Android Studio development
    - Fragments & tabbed layout
- Web technologies
    - Node.js & Nginx
- MongoDB
- Proper documentation
- Revision management

How to: Retrofit

"Retrofit turns your REST API into a Java interface".

This Java interface allows for simple communication between our front-end app and backend server while maintaining the structure of our database models.

```java
public class Song {
    private String  title;
    private String  artist;
    private String  album;
    private Integer year;
    private String  songId;
    private String  artworkUrl;
    private String  userId;
    private String  username;
    private Integer voteCount;
    private Vote    vote;
    ...
}
```

```java
/**
 * Add a song to an event's playlist
 *
 * @param currentUserId
 * @param eventId
 * @param song
 * @return Playlist
 */
@POST(PLAYLIST_SONGS_URL)

public Playlist addSong(@Header(USER_HEADER) String currentUserId,
    @Path(EVENT_PATHVAL) String eventId, @Body SongRequestDO song);
```

How to: Retrofit - CollabifyApi

```java
/**
 * Add a song to an event's playlist
 *
 * @param eventId
 * @param song
 * @return
 * @throws CollabifyApiException
 */
public Playlist addSong(String eventId, SongRequestDO song) throws CollabifyApiException
{
    if (mCurrentUserId == null || "".equals(mCurrentUserId)) {
        throw new CollabifyApiException();
    }
    return mCollabifyService.addSong(mCurrentUserId, eventId, song);
}
```

How to: Retrofit - appManager

```java
public void addSong(final Song song, final CollabifyCallback<space.collabify.android.collabify.models.domain.Playlist> callback) {
    if (song == null) { return; }
    SongRequestDO songDO = new SongRequestDO();
    songDO.setArtist(song.getArtist());
    songDO.setTitle(song.getTitle());
    songDO.setYear(song.getYear());
    ...
    try {
        mCollabifyApi.addSong(mEvent.getEventId(), songDO, new
          Callback<space.collabify.android.collabify.models.domain.Playlist>() {
            @Override
            public void success(space.collabify.android.collabify.models.domain.Playlist playlist, Response response) {
                callback.success(playlist, response);
                callSongAddListeners(song);
            }
            @Override
            public void failure(RetrofitError retrofitError) {
                callback.failure(retrofitError);
            }
        });
    } catch (CollabifyApiException e) {
        ...
    }
}
```

How to: Retrofit - Response

```
05-04 21:44:57.053    1403-1930/space.collabify.collabify D/Retrofit : ---> HTTP POST
  http://collabify.space:1337/events/122056943/playlist/songs/
05-04 21:44:57.053    1403-1930/space.collabify.collabify D/Retrofit : userid: 125929022
05-04 21:44:57.053    1403-1930/space.collabify.collabify D/Retrofit : Content-Type: application/json; charset=UTF-
8
05-04 21:44:57.053    1403-1930/space.collabify.collabify D/Retrofit : Content-Length: 211
05-04 21:44:57.053    1403-1930/space.collabify.collabify D/Retrofit :
{
  "album": "Maiden England \u002788",
  "artist": "Iron Maiden",
  "artworkUrl": "https://i.scdn.co/image/0fdb7e455945e7372eb4369bbca0bbb395b2aa68",
  "songId": "0e7MKWjKXnaXbBfZskfSG6",
  "title": "Run to the Hills",
  "year": 9999
}
05-04 21:44:57.053    1403-1930/space.collabify.collabify D/Retrofit : ---> END HTTP (211-byte body)


05-04 21:44:57.285    1403-1930/space.collabify.collabify D/Retrofit : <--- HTTP 200
  http://collabify.space:1337/events/122056943/playlist/songs/ (231ms)
05-04 21:44:57.285    1403-1930/space.collabify.collabify D/Retrofit : Server: nginx/1.6.2
05-04 21:44:57.285    1403-1930/space.collabify.collabify D/Retrofit : Date: Tue, 05 May 2015 02:48:16 GMT
...
05-04 21:44:57.285    1403-1930/space.collabify.collabify D/Retrofit : <--- END HTTP (3214-byte body)
```

Questions?

Thank you for listening!