API Mashup Report

# Spotifind

CAB432 – Assessment 1

Atrey Gajjar – n9767142

17 September 2018

## CONTENTS

## INTRODUCTION

Spotifind is a mashup with the aim of providing fans with an easy way to find concerts to attend and enjoy. It combines Leaflet JS, with APIs provided by Songkick and Spotify, to provide a simple and clean service to search for live events, with additional functionality to alleviate issues that normally arise in finding concerts.

Users can search for events by artist and/or area, with additional date filters available, providing flexibility on the specificity of event search criteria. Following the initial query, a detailed list of area and artist information found via the text search is provided. This allows the user to select exact search parameters to prevent input vagueness (e.g. a search of 'brisbane' in the area could mean Brisbane CA, US or Brisbane QLD, AU). Additionally, to combat the common issue of being limited to artists the searcher is familiar with, Spotifind provides the discover feature. This feature enables users to find events of artists that sound similar to the initially selected artist.

Concerts or music festivals that are found are marked on a map, with a summary of the key information rendered onto the sidebar. Users can interact with the map or the sidebar to uncover event details or physical location of the associated event respectively. Music preview snippets for each artist found are provided where possible, to allow the user to determine whether they enjoy the style of music before investigating further into events of the artist.

### Leaflet JS

Link: https://leafletjs.com/

Leaflet is a JavaScript library for interactive maps. It provides an extensive API reference, with a wide variety of functionality available. A subset of this functionality is the ability to place markers on the map with a popup.

Spotifind utilises this to visually represent the location of a live music event on the map. It allows users to judge relative distances from some reference point, determine areas of high event concentration for an artist, etc. Intuitively, the markers can be selected to view information held within the popup, and show the relevant event summary in the sidebar.

### Songkick API

Link: https://www.songkick.com/developer

Songkick is a music service with an extensive directory of upcoming concerts. It also provides ticket selling services for the aforementioned concerts. It provides an API to search for artists, areas, venues, and events based on a number of filters.

Spotifind uses the service to get unique identifications for artists and areas, based on text searches, with the '/search/artists.json' and '/search/locations.json' endpoints respectively. The IDs are consequently used to find events with the '/events.json' endpoint. The event information provided is filtered down, and rendered onto the client-side map and sidebar.

### Spotify API

Link: https://developer.spotify.com/documentation/web-api/

Spotify is an extensive music streaming service, with millions of tracks available for listening. Information regarding an artist's discography, user profiles, and more is available via the API.

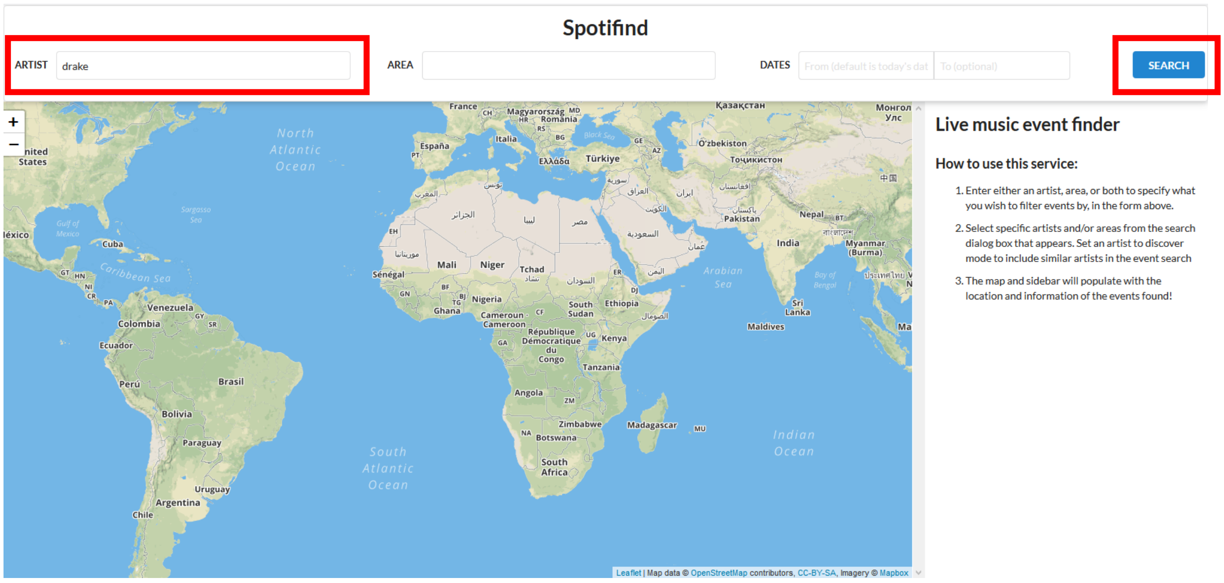
Spotifind uses the '/search' endpoint to retrieve Spotify's information for a searched artist. Genre data, images, and an id is collected to accurately depict the artist in search results. Once events are found, the '/artists/{id}/top-tracks' endpoint is used to get the top track for an artist for previewing.

## USE CASES

### Case 1

*As a user, I want the system to display when and where my favourite artist is performing, so I can attend their shows.*

This user navigates to the site and enters their favourite artist into the 'ARTIST' search field and presses the search button. As shown in figure 1 below.



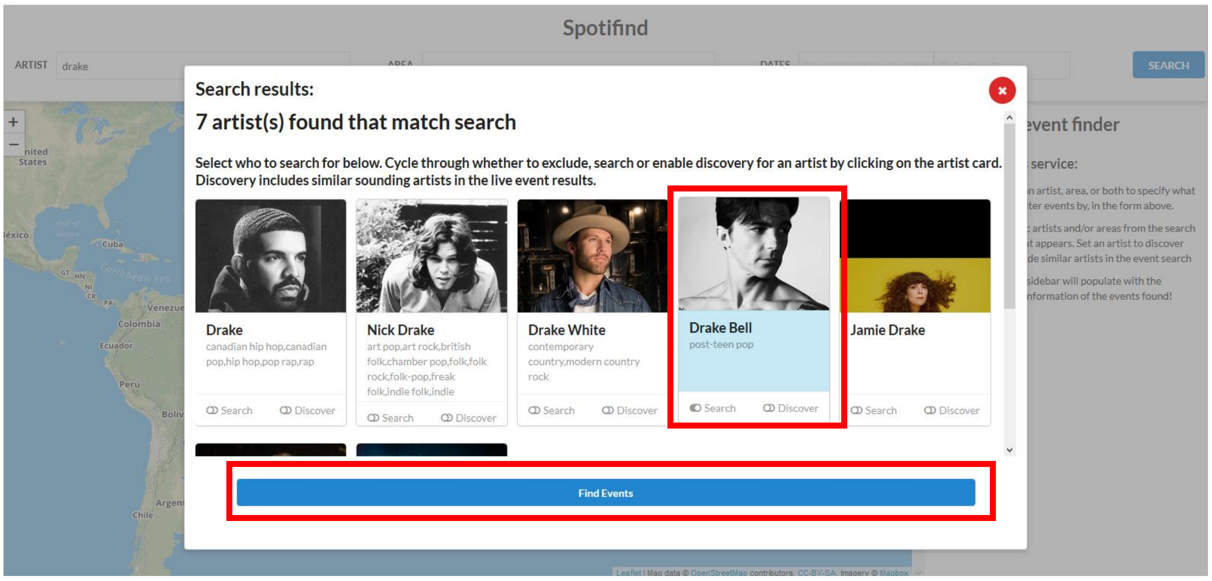
Live music event finder

How to use this service:

1. Enter either an artist, area, or both to specify what you wish to filter events by, in the form above.
2. Select specific artists and/or areas from the search dialog box that appears. Set an artist to discover mode to include similar artists in the event search
3. The map and sidebar will populate with the location and information of the events found!

**Figure 1 – Searching for an artist**

Following this, the user is faced with a search result dialog box. They initially intended to search for 'Drake Bell' and thus click on the artist card once to enable search for that artist. Since they only want results for 'Drake Bell', they decide not to include any others in the event search and to leave the discover feature off (by not clicking twice). Happy with the selection, the user presses the find events button as shown in figure 2.



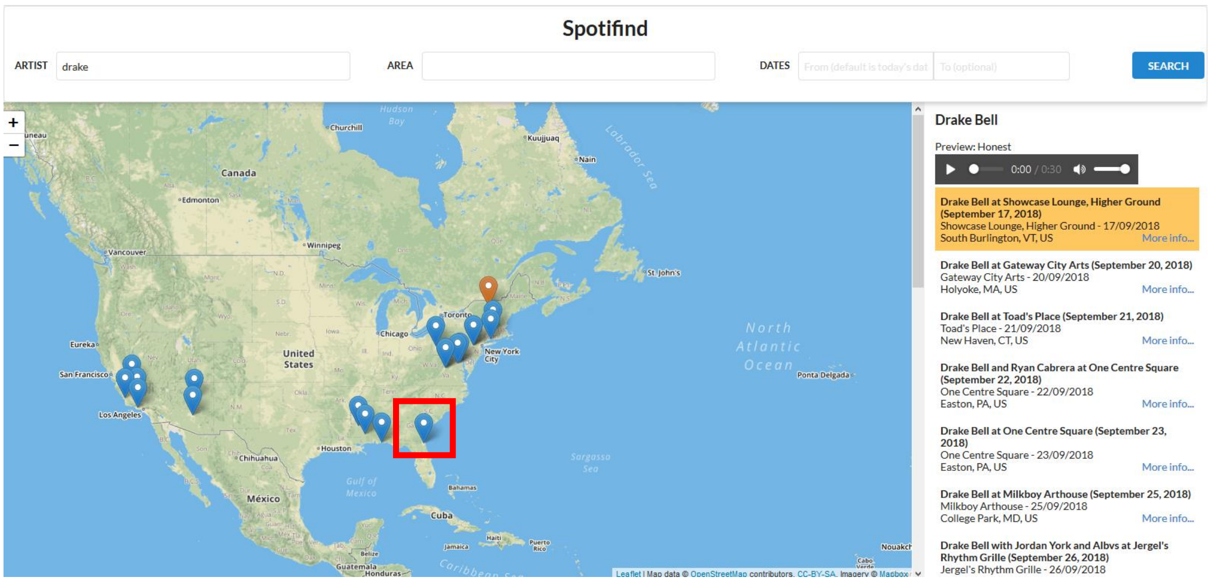
Search results:

7 artist(s) found that match search

Select who to search for below. Cycle through whether to exclude, search or enable discovery for an artist by clicking on the artist card. Discovery includes similar sounding artists in the live event results.

Drake
canadian hip hop,canadian pop,hip hop,pop rap,rap
Nick Drake
art pop,art rock,british folk,chamber pop,folk,folk rock,folk-pop,freak folk,indie folk,indie
Drake White
contemporary country,modern country rock
Drake Bell
post-teen pop
Jamie Drake
Search
Discover
Find Events

**Figure 2 – Search selection of single artist with discover feature not active**



Spotifind

ARTIST: drake
AREA:
DATES: From (default is today's dat | To (optional)
SEARCH

Drake Bell

Preview: Honest

0:00 / 0:30

Drake Bell at Showcase Lounge, Higher Ground (September 17, 2018)
Showcase Lounge, Higher Ground - 17/09/2018
South Burlington, VT, US — More info...

Drake Bell at Gateway City Arts (September 20, 2018)
Gateway City Arts - 20/09/2018
Holyoke, MA, US — More info...

Drake Bell at Toad's Place (September 21, 2018)
Toad's Place - 21/09/2018
New Haven, CT, US — More info...

Drake Bell and Ryan Cabrera at One Centre Square (September 22, 2018)
One Centre Square - 22/09/2018
Easton, PA, US — More info...

Drake Bell at One Centre Square (September 23, 2018)
One Centre Square - 23/09/2018
Easton, PA, US — More info...

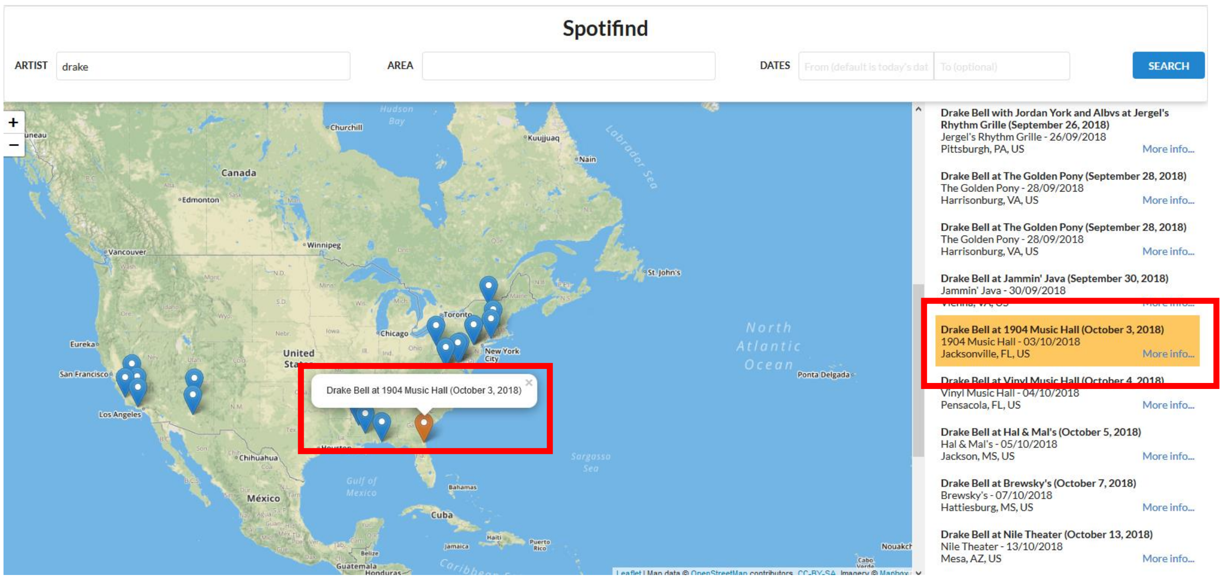
Drake Bell at Milkboy Arthouse (September 25, 2018)
Milkboy Arthouse - 25/09/2018
College Park, MD, US — More info...

Drake Bell with Jordan York and Albvs at Jergel's Rhythm Grille (September 26, 2018)
Jergel's Rhythm Grille - 26/09/2018

Canada
United States
México
North Atlantic Ocean
Gulf of Mexico
Cuba
Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox

**Figure 4 - Search results for an artist search of 'Drake Bell'**

From figure 3 above, the user realises that Drake Bell is currently touring and has a number of events. Noticing a marker at a location they were planning on visiting anyway, the user decides to click the marker, resulting in figure 4. Given the additional information in the sidebar, the user decides that it would be a good idea to visit and attend the concert at the same time. They click on the 'More info...' link to find out extra details and buy tickets.



Spotifind

ARTIST: drake
AREA:
DATES: From (default is today's dat | To (optional)
SEARCH

Drake Bell with Jordan York and Albvs at Jergel's Rhythm Grille (September 26, 2018)
Jergel's Rhythm Grille - 26/09/2018
Pittsburgh, PA, US — More info...

Drake Bell at The Golden Pony (September 28, 2018)
The Golden Pony - 28/09/2018
Harrisonburg, VA, US — More info...

Drake Bell at The Golden Pony (September 28, 2018)
The Golden Pony - 28/09/2018
Harrisonburg, VA, US — More info...

Drake Bell at Jammin' Java (September 30, 2018)
Jammin' Java - 30/09/2018
Vienna, VA, US — More info...

Drake Bell at 1904 Music Hall (October 3, 2018)
1904 Music Hall - 03/10/2018
Jacksonville, FL, US — More info...

Drake Bell at Vinyl Music Hall (October 4, 2018)
Vinyl Music Hall - 04/10/2018
Pensacola, FL, US — More info...

Drake Bell at Hal & Mal's (October 5, 2018)
Hal & Mal's - 05/10/2018
Jackson, MS, US — More info...

Drake Bell at Brewsky's (October 7, 2018)
Brewsky's - 07/10/2018
Hattiesburg, MS, US — More info...

Drake Bell at Nile Theater (October 13, 2018)
Nile Theater - 13/10/2018
Mesa, AZ, US — More info...

Drake Bell at 1904 Music Hall (October 3, 2018)

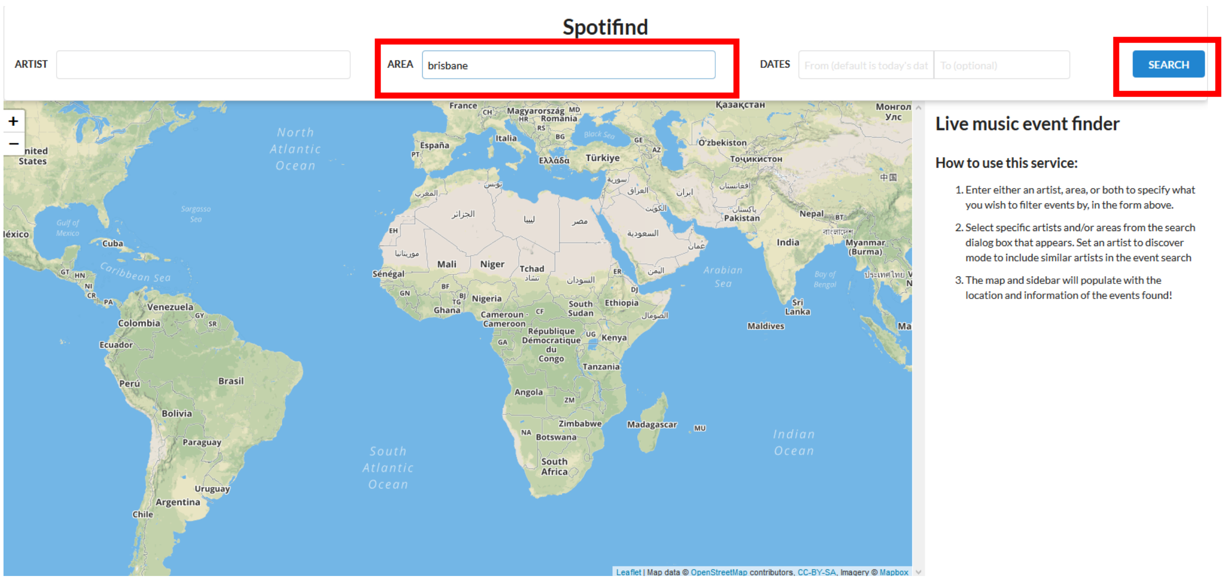
Canada
United States
México
North Atlantic Ocean
Gulf of Mexico
Cuba
Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox

**Figure 3 – Selected event to find more information about**

## Case 2

*As a user, I want the system to show me concerts in my area, so I do not have to travel far to enjoy music.*

This user navigates to the site and enters their local area in the 'AREA' search field and presses the search button, shown below.



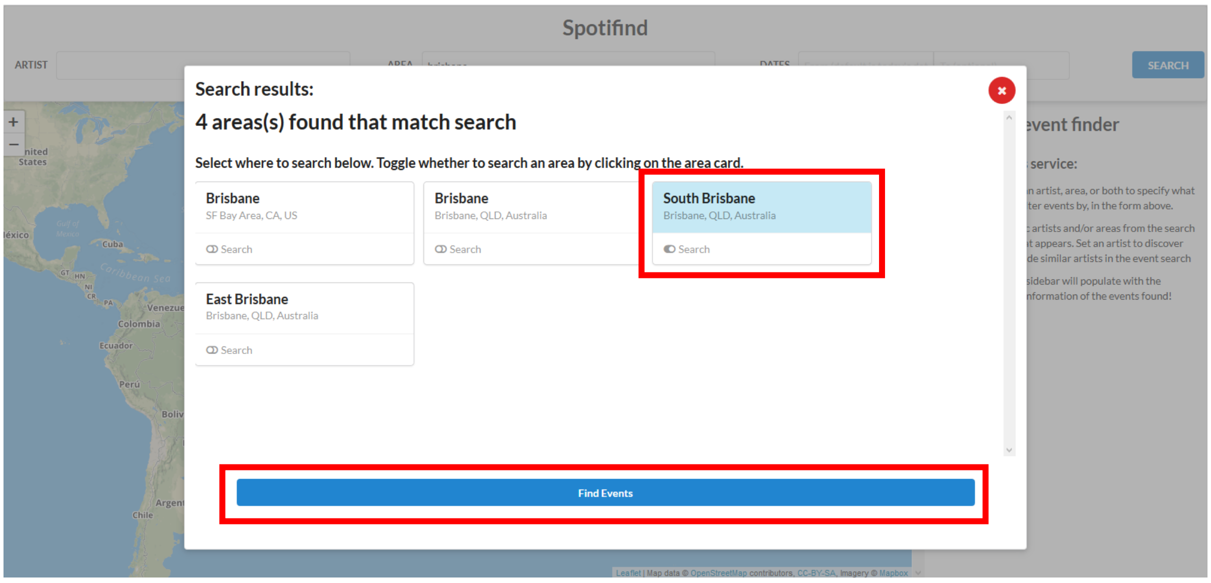
Live music event finder

How to use this service:

1. Enter either an artist, area, or both to specify what you wish to filter events by, in the form above.
2. Select specific artists and/or areas from the search dialog box that appears. Set an artist to discover mode to include similar artists in the event search
3. The map and sidebar will populate with the location and information of the events found!

**Figure 5 – Searching by an area**

Given the search selection dialog box in figure 6, the user selects the most accurate entry to minimise travel and presses find events.



Search results:

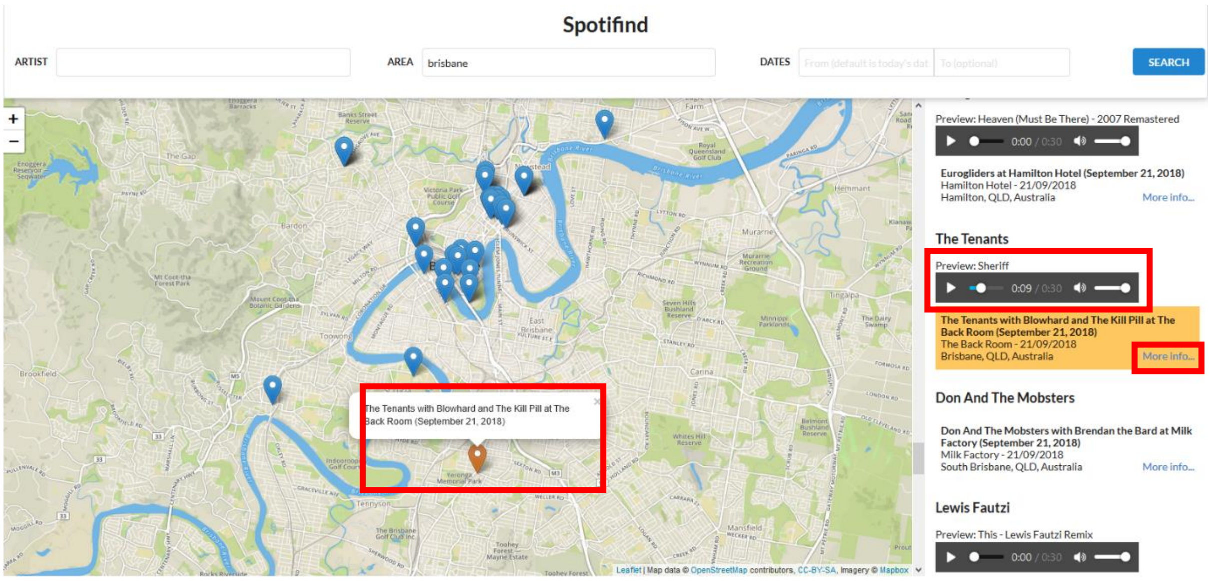
4 areas(s) found that match search

Select where to search below. Toggle whether to search an area by clicking on the area card.

| | | |
|---|---|---|
| Brisbane<br>SF Bay Area, CA, US<br>Search | Brisbane<br>Brisbane, QLD, Australia<br>Search | South Brisbane<br>Brisbane, QLD, Australia<br>Search |
| East Brisbane<br>Brisbane, QLD, Australia<br>Search | | |

**Figure 6 – Search selection of area with one selection**

The user zooms in on the map view to obtain a better view of the results, given the concentrated distribution of events. As they do not wish to venture into the city for an event, they select the marker closest to their house and obtain figure 7. As the user is not familiar with the artists performing, they play the preview song of The Tenants. Luckily, they enjoy it and decide to attend this event.


Spotifind

ARTIST

AREA brisbane

DATES From (default is today's dat To (optional)

SEARCH

The Tenants with Blowhard and The Kill Pill at The Back Room (September 21, 2018)

Preview: Heaven (Must Be There) - 2007 Remastered

0:00 / 0:30

Eurogliders at Hamilton Hotel (September 21, 2018)
Hamilton Hotel - 21/09/2018
Hamilton, QLD, Australia
More info...

The Tenants

Preview: Sheriff

0:09 / 0:30

The Tenants with Blowhard and The Kill Pill at The Back Room (September 21, 2018)
The Back Room - 21/09/2018
Brisbane, QLD, Australia
More info...

Don And The Mobsters

Don And The Mobsters with Brendan the Bard at Milk Factory (September 21, 2018)
Milk Factory - 21/09/2018
South Brisbane, QLD, Australia
More info...

Lewis Fautzi

Preview: This - Lewis Fautzi Remix

0:00 / 0:30

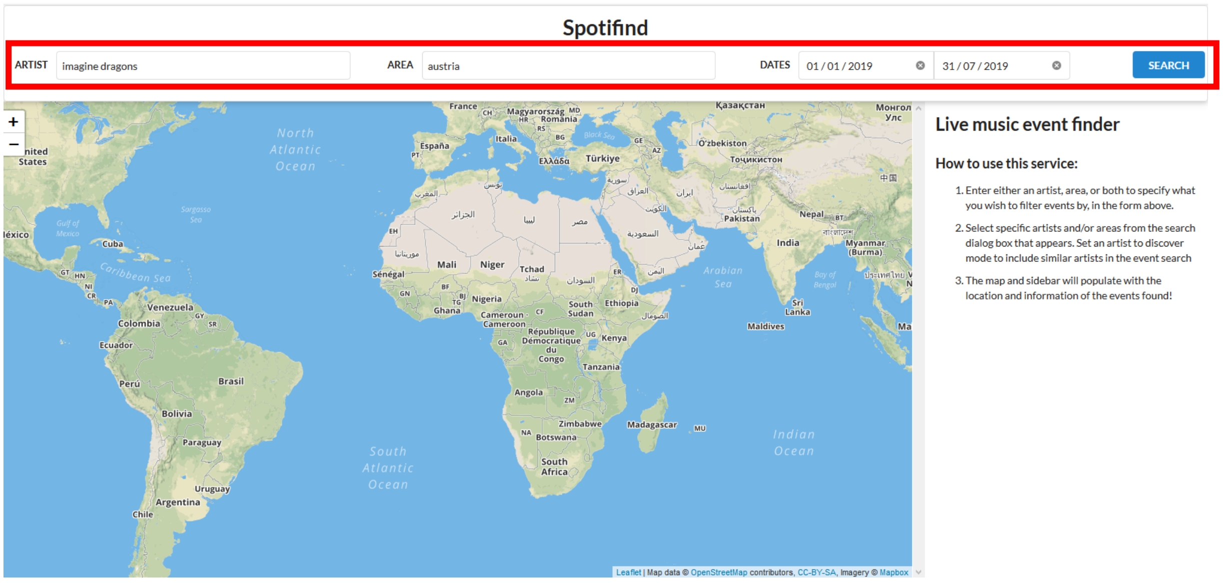
Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox

**Figure 7 – Zooming into Brisbane, finding event closest to a location, and previewing an artist's top track**

## Case 3

*As a user, I want the system to show events of an artist I like, and other suggestions, during my holiday next year overseas.*

This user will enter all the search fields, to provide specific results to their scenario, and press the search button.


Spotifind

ARTIST imagine dragons

AREA austria

DATES 01 / 01 / 2019 31 / 07 / 2019

SEARCH

Live music event finder

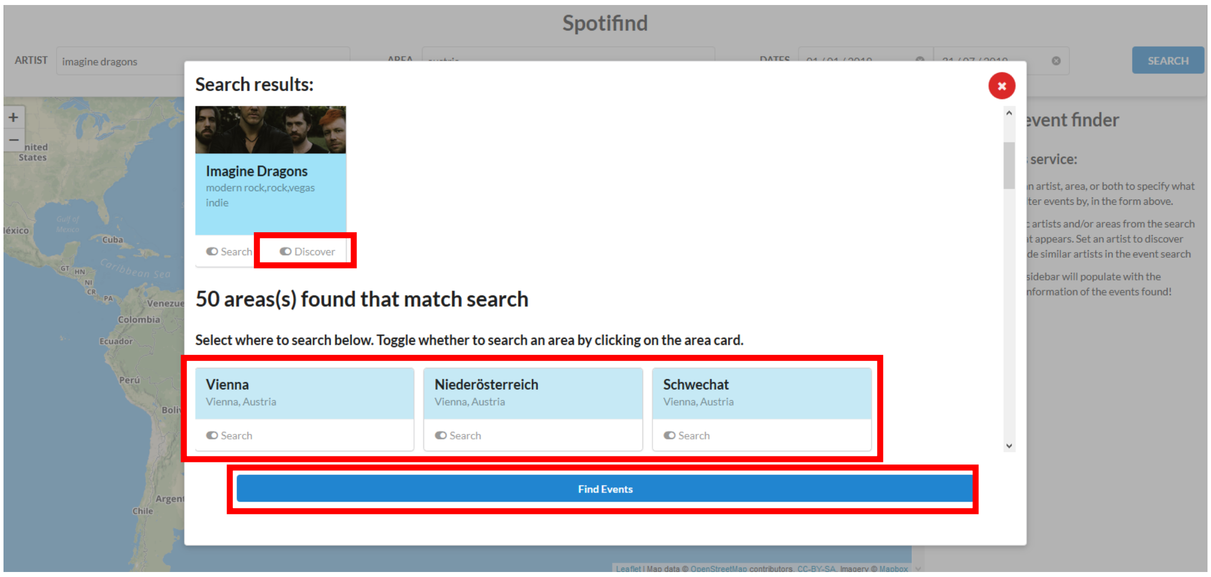
How to use this service:

1. Enter either an artist, area, or both to specify what you wish to filter events by, in the form above.
2. Select specific artists and/or areas from the search dialog box that appears. Set an artist to discover mode to include similar artists in the event search
3. The map and sidebar will populate with the location and information of the events found!

Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox

**Figure 8 – Complete scenario with all filters search entry**

As both area and artist are selected, the user will have to select options from both sections in the search result dialog box. As the user wants to get suggestions, they click twice on the Imagine Dragons artist card to enable discovery mode for that artist. Additionally, they know they will be spending time in Vienna, Niederosterreich, and Schwechat so they select these areas. The find events button is pressed once the appropriate selections are made.


Search results:

Imagine Dragons
modern rock,rock,vegas indie

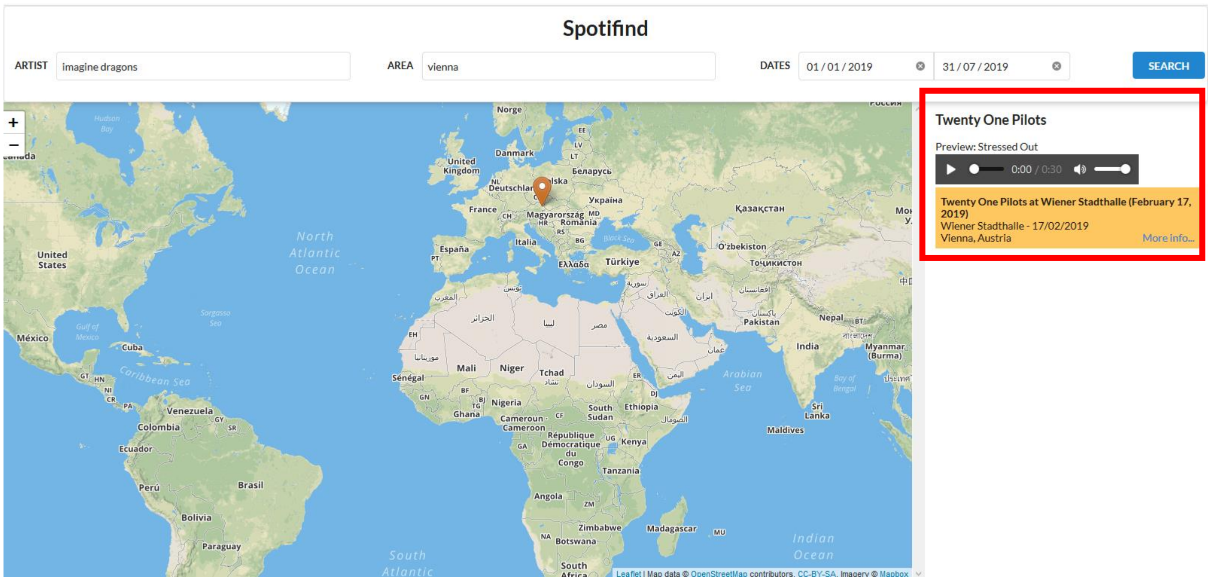
50 areas(s) found that match search

Select where to search below. Toggle whether to search an area by clicking on the area card.

Vienna, Vienna, Austria
Niederösterreich, Vienna, Austria
Schwechat, Vienna, Austria

Figure 9 – Search window with both key parameters completed. Artist is set to discovery.

Figure 10 shows that unfortunately Imagine Dragons (primary artist searched for) does not have any events in the selected areas during the specified time period. However, an event of a similar artist is available. The user previews the song sample, and decides to attend the Twenty One Pilots concert.


ARTIST: imagine dragons
AREA: vienna
DATES: 01 / 01 / 2019 – 31 / 07 / 2019

Twenty One Pilots
Preview: Stressed Out
Twenty One Pilots at Wiener Stadthalle (February 17, 2019)
Wiener Stadthalle - 17/02/2019
Vienna, Austria

Figure 10 – Events found for the specific search criteria with area, artist (on discover), and date filters.

## Service API Calls

All use cases utilise the same services. The external services supporting particular functionality are:

| | |
|---|---|
| Artist search | https://api.spotify.com/v1/search; https://api.songkick.com/api/3.0/search/artists.json |
| Similar artists | https://api.spotify.com/v1/artists/{id}/related-artists |
| Area search | https://api.songkick.com/api/3.0/search/locations.json |
| Event search | https://api.songkick.com/api/3.0/events.json |
| Music preview | https://api.spotify.com/v1/artists/{id}/top-tracks |
| Map markers | L.marker([lat, lng]).bindPopup(displayName) |

## TECHNICAL DESCRIPTION

Emphasis was placed on client and server distinction in creating the general architecture of Spotifind. The server was to handle all requests of the client, appropriately retrieving data from APIs when required as depicted in figure 11 below.

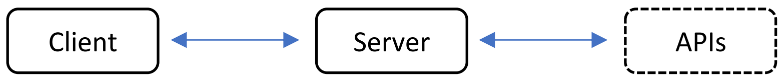


**Figure 11 – General overview of communication between components**

Express-generator (ExpressJS, 2018) was used to create the directory structure and boilerplate for this application. As such, the '/public/' acted as the client directory, with locally required files stored within. This included stylesheets, images, and required client-side js.

Processing was aimed to be minimised on the client side for performance and security reasons. As such, the core logic within the application was held at the '/' index route of the server; the only route. The client does not navigate from the index page to maintain a smooth experience. However, this route handles two request endpoints, '/search' and '/events', for the two stages of searching in this application. Both endpoints manage the client input, make the required API calls, and return the processed results back to the client.

### Client

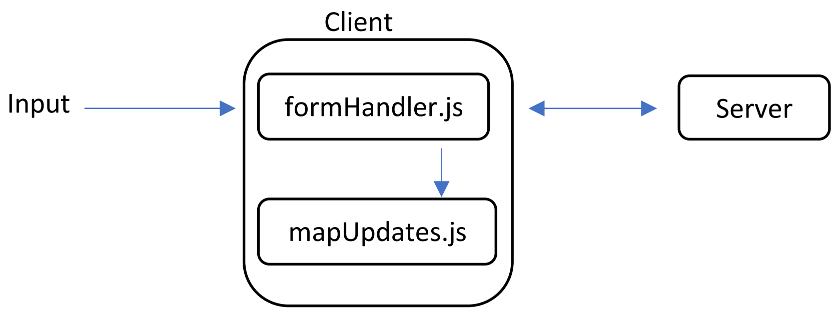


**Figure 12 – Detailed client interactions**

#### Input to Server interaction

The input form was processed via JavaScript before being sent to the server, to provide better feedback to the user if the input was invalid (i.e. neither artist nor area field filled out). More importantly however, it allowed an asynchronous 'fetch' to be sent. The fetch API sent resource requests to be sent along a network; akin to AJAX (Mozilla, 2018). This allowed information to be retrieved from the server, with only the associated DOM elements being updated. Reloading the entire page on server response, proved detrimental to the user experience and processing, as components like the map were needlessly re-rendered.

The server response involved queried data and a HTML string representation of the relevant user-facing data. The element (sidebar or search window) was updated with the HTML to display the response to the user. The data was used with event listeners to create events with specific data. If event data is received by the client, it accesses a map handler object to add events to the leaflet map.

#### Leaflet Map

The map was initially setup once the page loaded, and bounded to one 'earth map'. A map handler object was also initialised for external access, with methods to alter the map. Events were placed by using one such method, which created markers from event data and added them as layers to the map.

### Styling

Basic design was provided to the application via Semantic-UI. This framework was chosen as it could provide a clean, minimal look with ease (Semantic-UI, 2018). Semantic would also provide a degree of responsiveness, for a better experience on a range of client devices. It had extensive documentation which reduced the complexity of use and time to get started.

## Server

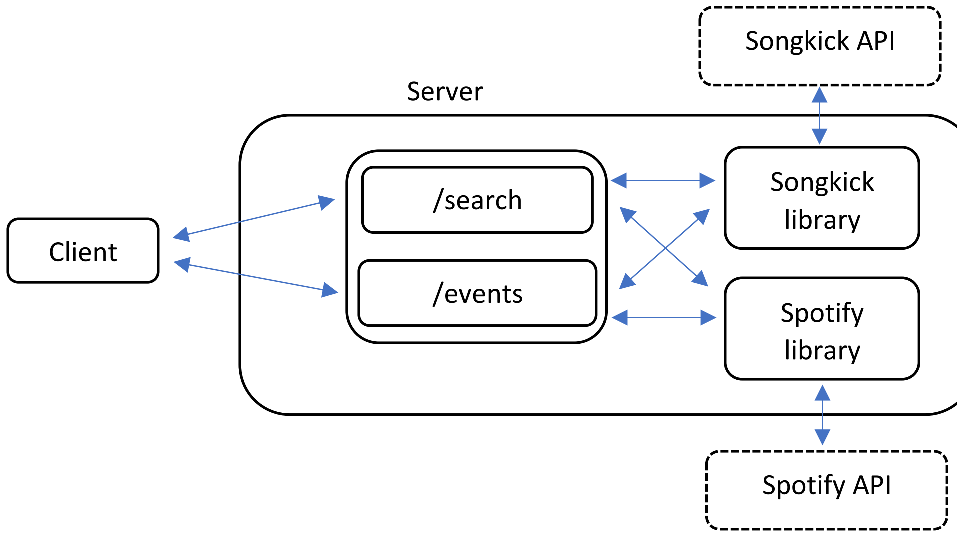


**Figure 13 – Detailed server interaction**

### Endpoints

The client accessed the /search and /events endpoints in the index route to retrieve data from the server, based on which stage the search was in. /search received the text inputs of the form and responded with specific search results back to the client. /events received the specific search selections in a similar format, and responded with event details. Each endpoint utilises the custom API libraries to get data, which is compiled together if necessary, and sent back to the client.

### API Libraries

A pseudo-library was created for each API used in this application. They consisted of a set of functions which requested information from the APIs as seen in figure 13. The request-promise module was used to make these requests, as it built on the standard request library but provided promise functionality to chain asynchronous commands. Due to the large response objects normally received from Spotify and Songkick, these functions also consisted of code to filter the responses to key information, in the format required by the application server.

Additionally, errors received from the API that could be fixed without client input, were handled in the library. For example, a 401 Authentication error from Spotify, could be systematically fixed by reauthenticating and making the request call again without client knowledge. The modularity provided by these as separate files and functions allows modifications to be made in how data is retrieved from external services without large impact on the application, further elaborated in the limitations section.

### Promises

Promises were used extensively within the server as multiple asynchronous requests were to made, with many depending on the outcome of previous requests. Multiple promise requests could be created simultaneously to improve performance, with the flexibility of using functions like

Promise.all() to wait for the result of all of them. The syntax proved easier to exercise over the call back alternative, with a familiar chaining sequence over repeated nesting. Catch statements were also supported in this sequential format for more logical error bubbling and handling.

### Templating

Pug was used as a templating engine for this project, as some elements were to be rendered with information received from the server (Pug, 2018). The index page was the only completely rendered page served to the client, as it was the only page used in the application. However, the search window and sidebar HTML were to be updated with search results and event data respectively. Pug templates were compiled and used to generate HTML strings for these elements based off response data on the server, which were passed to the client to set as the inner HTML of the relevant DOM elements. This approach was used instead of constructing a HTML representation on the client side, to reduce processing load, allow flexibility if the display format is to be updated, and provide a more intuitive method of designing a structure (HTML template over JS string manipulation).

## Issues Encountered

1. Setting up the architecture

A key issue initially in creating this project was determining how to structure the project. This was due to inexperience in developing web applications. A few directory structures were created, however neither provided clear division between server and client. As it was determined that ExpressJS would be used on top of node for the server implementation, research was undertaken on the established way to create express applications. The express-generator module was then uncovered and utilised to create the base structure. This was examined to get a deeper understanding on how the client-server interactions were to work, and it was built upon with the additional requirements of Spotifind.

2. Searching for input

It was quickly realised that text search input of artist and areas introduced a level of vagueness in what was being searched for. Certain queries resulted in multiple matching results for both artist and areas (e.g. Brisbane, US – Brisbane, AU). As the user may have meant either of them, a design decision was made to provide these results back to the user for them to select exactly which entries they wished to search for.

In terms of artist results, as data was to be compiled from Songkick (events) and Spotify (genres, image, etc), artists had to possess a link between the services. As there was no common unique ID between the two platforms, this proved difficult; artist names may not have been an exact match (e.g. JAY Z and Jay-Z). As such it was decided to normalise artist names by making them all lower case without any special characters and then compare them as a unique identifier.

3. Spotify Authentication

Spotify authentication was challenging due to the number of steps involved. Due to the various usages of the API, there were three authentication flows available. In addition to this, even after a client_id and client_secret was obtained for the client credentials flow, these were to be encoded in base64 and sent to Spotify to receive an access token. Finally, this access token could make API requests. An example request using the request library provided by Spotify assisted in getting this to work.

However, the received token too expired after a time limit. To combat this issue, a call back had to be created which re-authenticated when required, and resent the API call.

## DOCKER CONTAINERISATION

Docker is a virtualisation platform that lies above a host operating system (Docker, 2018). It is used to create an executable container image to house everything required to run an application. A docker image of Spotifind was created to ensure it maintained a standard configuration environment wherever it was run. It also reduces launch time on different machines as requirements are pre-built into the image. A few configuration decisions were made in creating the image, which will be elaborated on. Refer to Appendix B for the complete Dockerfile.

1.

```
#Base image of node.js v8
FROM node:8
```

The image inherited from the official docker image for node, specifically the current major supported update of v8 (NodeJS, 2018). This was opted for as v8 of node was used in development. The base node image was used instead of an ubuntu image, as a robust node.js installation is already present, reducing the complexity required of the Spotifind image. A slim version of the node image was not deemed necessary, as the machine had sufficient storage, and to ensure all modules required by Spotifind had the complete node installation for support.

2.

```
#Install app dependencies
COPY package*.json ./
RUN npm install --production
COPY . .
```

A npm install was run with the –production flag to exclude development only packages (e.g. nodemon) from being installed in the image, reducing its size. The app directory was copied over to the image, with exclusions provided by a .dockerignore file, which primarily included the node_modules folder, as these were unnecessary given the clean install of packages.

3.

```
#Run app in production environment
ENV NODE_ENV=production
CMD npm start
```

An environment variable was set to run Spotifind in production mode. This alters certain undesirable behaviour, such as not rendering a full error stack when the client receives a routing error.

## TESTING AND LIMITATIONS

Testing was a critical part of the development process, along with strategies to mitigate errors from the onset. Primarily, alterations were made incrementally to the existing code base, with more functionality only being added after the previous code was tested. Also, code was initially written for effectiveness over elegance, to keep simplicity and reduce risk of error. It was later refactored to maintain effectiveness, but improve modularity and reduce redundancy.

A number of error types were identified including syntactic errors, runtime errors, server errors, and logic error. Syntactic errors were easily prevented by the use of a code editor with a linter. Runtime errors were generally found through unexpected behaviour in usage, followed with an error log in the console. Server errors were similar with instead an error status code being logged. These were fixed by identifying the error from the log, and reassessing the problem code. If it was identified that error could be expected in certain cases, catch statements were used to handle it as a 'checked exception'.

Logic errors proved the most difficult to amend as the code was functioning, however not as expected with clear indication on why. To diagnose the functionality of the application at this level, a testing plan was devised and updated regularly. From below, it can be seen that all functionality tested is working as intended.

### Testing Plan

| Task | Expected Outcome | Result |
|---|---|---|
| **Search form** | | |
| Search with no fields entered | Error alert to user and form doesn't get sent to server. | Pass |
| Search with artist name | Search result dialog box appears with matched artist cards. | Pass |
| Search with area name | Search result dialog box appears with matched area cards. | Pass |
| Search with artist and area name | Search result dialog box appears with matched area and artist cards. | Pass |
| Select a date to search events from | Date selector appears allowing a date from the current date onwards. | Pass |
| Select a date to search events to | Date selector appears allowing a date to be selected after the 'from' date (if selected) or the Pass current date. | Pass |
| **Search results window** | | |
| (On finding artists or areas from search form) | Set an overlay (to disable background interactions) and display the results found in a div. | Pass |
| (On not finding artists or areas from search form – tested by using random text as input) | Provide an alert that results could not be found for the search criteria. | Pass |
| Select artist for search (click once) | Artist card changes to a light blue background colour, and search icon is toggled on. | Pass |
| Select artist for discovery mode (click twice) | Artist card changes to a darker blue background colour, and discover icon is toggled on. | Pass |
| Unselect artist from searching (click thrice) | Artist card changes back to default white background colour, and search/discover icons are both toggled off. | Pass |
| Select area for search (click once) | Area card changes to a light blue background colour, and search icon is toggled on. | Pass |
| Unselect area for search (click twice) | Area card changes back to default white background, and search icon is toggled of. | Pass |

| | | |
|---|---|---|
| Pressing find events without selecting any search items | Alert that at least on item must be selected. | Pass |
| Pressing find events after selecting search items | Events are found, and displayed on the map and sidebar. Overlay and search window are removed. | Pass |
| Pressing close button on top right after results are shown to change query | Overlay and search window are removed. | Pass |
| (on no events being found) | An alert is displayed to indicate that the user should try altering their search query. | Pass |
| **Map** | | |
| Pan and zoom the map within the single world image | Map pans and zooms as expected. | Pass |
| Pan and zoom the map trying to leave the single world image | Zoom is disabled at a maximum level, and map 'bounces' back at extreme pans. | Pass |
| (On events being found) | Markers are placed at latitude and longitude coordinates of events. | Pass |
| Clicking on an event marker | Popup with the display name of the event appears. Marker colour changes to indicate the focused event, and the sidebar scrolls to the relevant event summary information and highlights it. | Pass |
| (On subsequent event searches) | The previous result markers are cleared from the map, to be replaced with new ones. | Pass |
| **Sidebar** | | |
| (On initial load) | Display instructions on how to use the application. | Pass |
| (on events being found) | Display the list of events, as grouped by artist, with a summary of key information. Includes a preview clip of an artist's top song if possible. | Pass |
| Clicking play on the preview snippet | Plays a 30 second clip of the artist's top song. | Pass |
| Click on an event on the sidebar | Displays focus on the event by changing the background colour to light orange. Focuses the corresponding marker, showing the popup and changing the marker colour. If the marker is not viewable in the map frame, pan to include it. | Pass |

| | | |
|---|---|---|
| (artist groupings) | Events are listed under a header of the main artist performing. | Pass |

Note: Extensive testing was undertaken to ensure that the events matched the criteria set in the search (artist, area, dates, and if similar artists were included) but are not included in the test plan due to the number of combinations possible and tested for.

### Limitations

A limitation in terms of testing is the lack of control in what the user can input into the search fields. Thus, it was not possible to conduct exhaustive testing of all possible inputs and there may be certain cases which when sent to the server and consequently to the API, break the program. This does, however, have a low chance of occurring given the testing undertaken.

Another limitation is regarding the dependence on external service data to provide this information. If a service fails the application would break as well. Additionally, if the service alters the format of the response provided or even alters how the API is accessed, the codebase would need to be adapted. The impact of this limitation is mitigated by having modular libraries to access the services.

Functionality compromises had to be made due to time constraints. Later in the development process, new features had to be limited to work within the current program flow and be completed in time. For example, the aim was to add multiple preview songs, however when it was implemented there was not enough time to add the ability to switch between songs. Hence, only one song is currently available in preview. The following section elaborates on possible extensions that could be possible given additional time.

## POSSIBLE EXTENSIONS

This application could be extended by improving the development stack and process, refining the current functionality, or introducing new APIs to this mashup for additional features. Suggestions to each of these areas of improvement are provided below:

- Development stack and process
  - A testing framework, such as Jest, could be introduced to the development process. This would allow automated testing to speed reduce time spent retesting functionality on addition of new features. It would also help drive a TDD approach.
  - A frontend library like ReactJS, could be used to construct a more robust user interface. As the UI holds elements that have state, ReactJS components could be leveraged to provide smoother DOM updates on state change.
- Current functionality
  - Provide more preview songs for each artist, as one song does not provide a complete representation of the sort of music they play.
  - Provide more control in navigating the events found by including an artist/location filter. Results could also be sorted by a number of other metrics (artist popularity, venue review, etc.) and paginated to allow more results to be displayed without making the interface unusable.
- Additional API integration features
  - OpenWeatherMap API to provide weather data at the location and time of the event to see if it will be pleasant and to decide on a mode of transport.
  - TripAdvisor API to find accommodation near an event location to arrange stay details if traveling far from the user's residence.

## REFERENCES

Docker. (2018). What is a container. Retrieved from https://www.docker.com/resources/what-container

ExpressJS. (2018). Express application generator. Retrieved from https://expressjs.com/en/starter/generator.html

Mozilla. (2010). Fetch API. Retrieved from https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API

NodeJS. (2018). Official docker image for Node.js. Retrieved from https://github.com/nodejs/docker-node

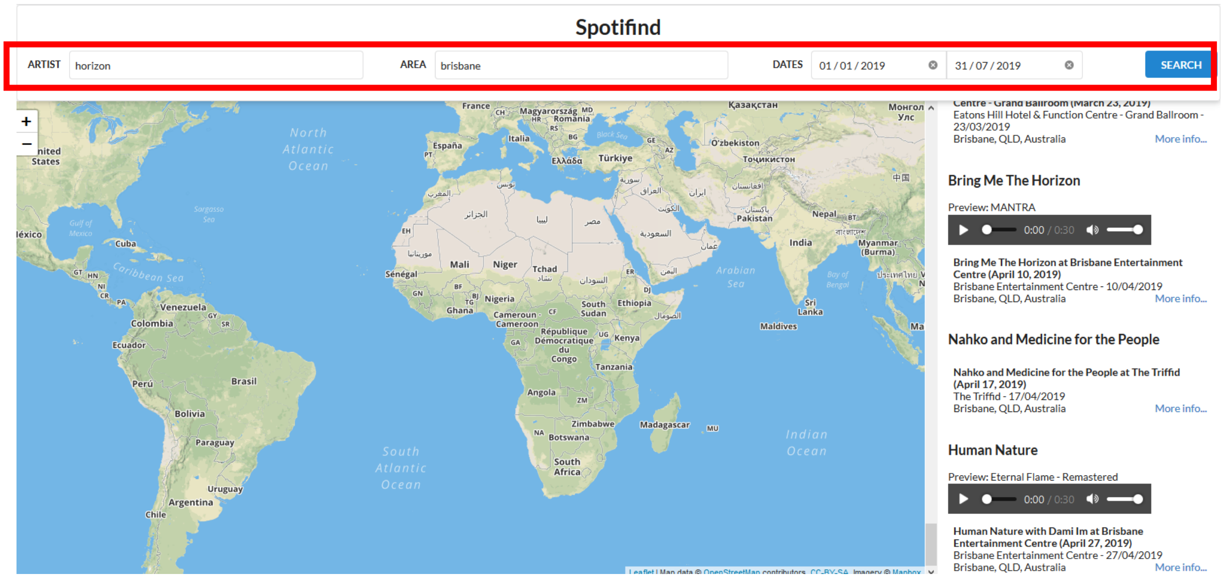
Pug. (2018). Getting started. Retrieved from https://pugjs.org/api/getting-started.html

Semantic-UI. (2018). Getting started. Retrieved from https://semantic-ui.com/introduction/getting-started.html

# APPENDIX

## Appendix A – User Guide

1. Enter either an artist, area, or both to specify what you wish to search events for in the form. Fill out the date fields if you want to filter events between specific dates. Press 'SEARCH'.



Spotifind

ARTIST horizon AREA brisbane DATES 01 / 01 / 2019 31 / 07 / 2019 SEARCH

Centre - Grand Ballroom (March 23, 2019)
Eatons Hill Hotel & Function Centre - Grand Ballroom - 23/03/2019
Brisbane, QLD, Australia More info...

Bring Me The Horizon

Preview: MANTRA

0:00 / 0:30

Bring Me The Horizon at Brisbane Entertainment Centre (April 10, 2019)
Brisbane Entertainment Centre - 10/04/2019
Brisbane, QLD, Australia More info...

Nahko and Medicine for the People

Nahko and Medicine for the People at The Triffid (April 17, 2019)
The Triffid - 17/04/2019
Brisbane, QLD, Australia More info...

Human Nature

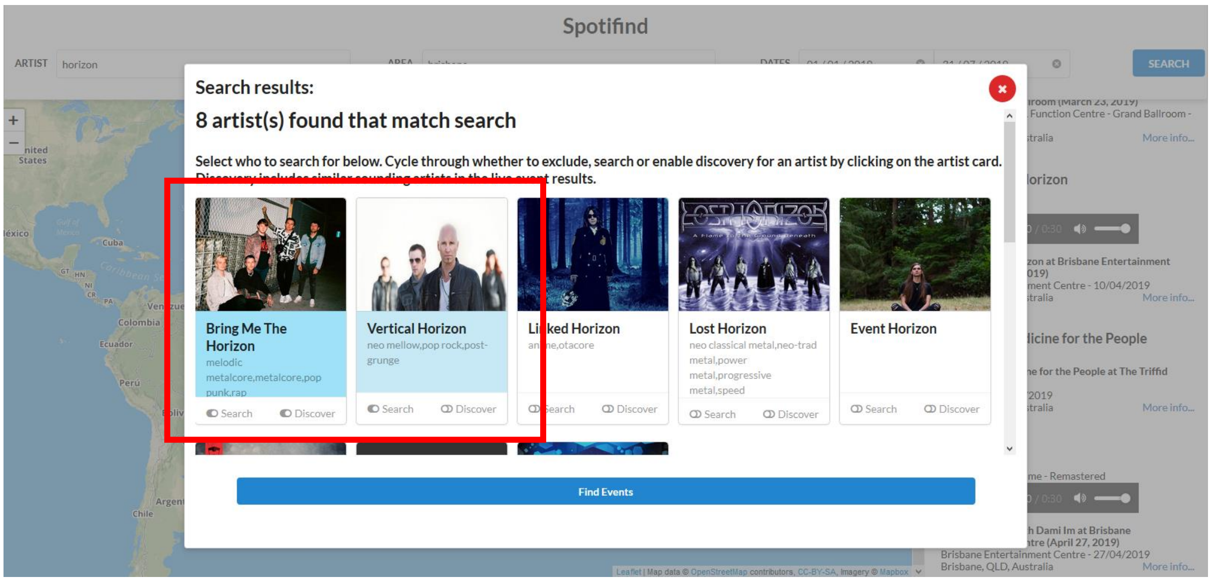
Preview: Eternal Flame - Remastered

0:00 / 0:30

Human Nature with Dami Im at Brisbane Entertainment Centre (April 27, 2019)
Brisbane Entertainment Centre - 27/04/2019
Brisbane, QLD, Australia More info...

Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox

2. Select the search specifics, and then press 'Find Events'.
   a. If input was put into the artist field, select artist cards to search for by clicking on them. Click twice to enable discovery mode. The following image shows Vertical Horizon and Bring Me The Horizon selected for search. However, Bring Me The Horizon has discover enabled meaning, artists which sound similar to Bring Me The Horizon will also be searched for.



Spotifind

Search results:

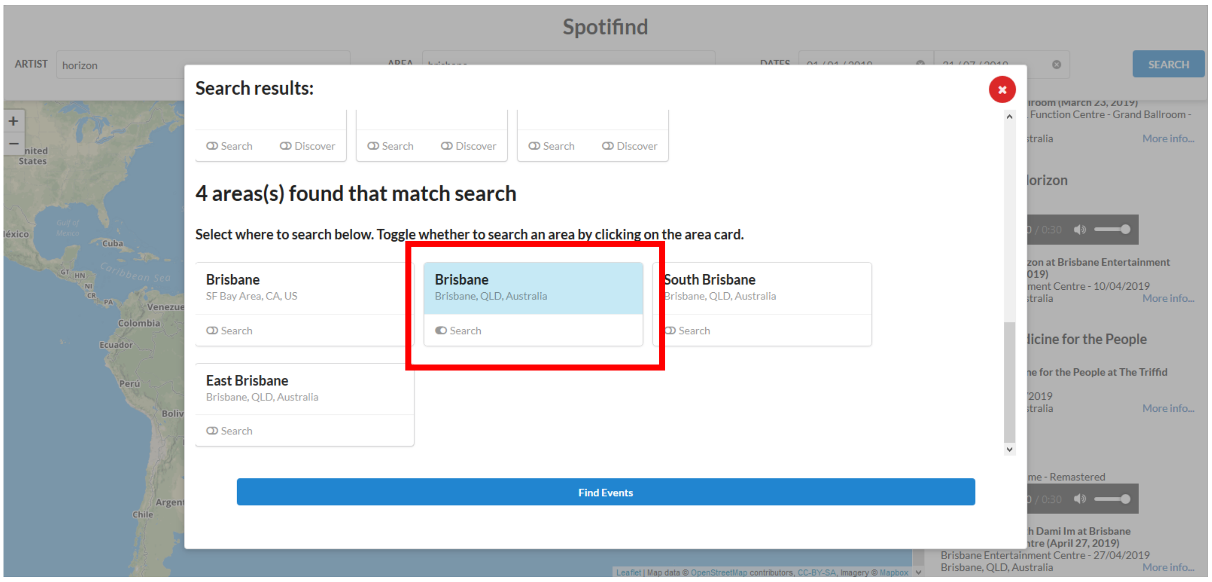
8 artist(s) found that match search

Select who to search for below. Cycle through whether to exclude, search or enable discovery for an artist by clicking on the artist card. Discovery includes similar sounding artists in the live event results.

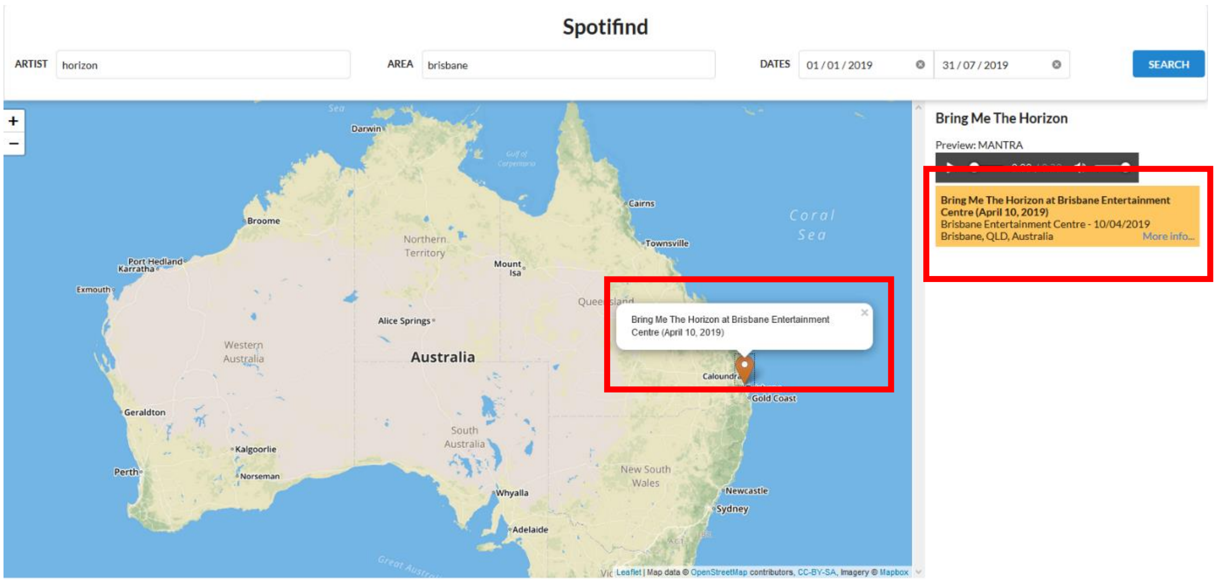
| Bring Me The Horizon | Vertical Horizon | Li ked Horizon | Lost Horizon | Event Horizon |
|---|---|---|---|---|
| melodic metalcore,metalcore,pop punk,rap | neo mellow,pop rock,post-grunge | an ne,otacore | neo classical metal,neo-trad metal,power metal,progressive metal,speed | |
| Search Discover | Search Discover | Search Discover | Search Discover | Search Discover |

Find Events

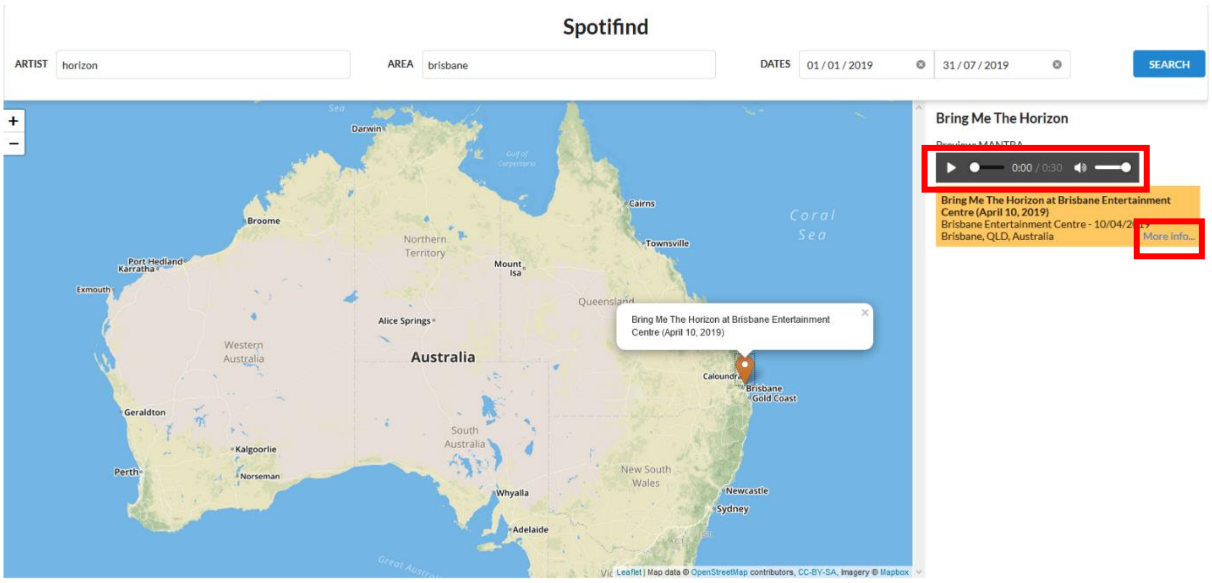
   b. If input was put in the area field, select area cards to search for by clicking on them.



3. View the events on the sidebar and the map. Click on the sidebar event or map marker to focus an event (click on a marker and the user will also focus the sider bar summary and vice-versa).


Spotifind
ARTIST horizon
AREA brisbane
DATES 01 / 01 / 2019 31 / 07 / 2019
SEARCH
Bring Me The Horizon
Preview: MANTRA
Bring Me The Horizon at Brisbane Entertainment Centre (April 10, 2019)
Brisbane Entertainment Centre - 10/04/2019
Brisbane, QLD, Australia
More info...
Bring Me The Horizon at Brisbane Entertainment Centre (April 10, 2019)

4. Preview the artist's music and click on more info if the user wants to attend the event.


Spotifind
ARTIST horizon
AREA brisbane
DATES 01 / 01 / 2019 31 / 07 / 2019
SEARCH
Bring Me The Horizon
0:00 / 0:30
Bring Me The Horizon at Brisbane Entertainment Centre (April 10, 2019)
Brisbane Entertainment Centre - 10/04/2019
Brisbane, QLD, Australia
More info...
Bring Me The Horizon at Brisbane Entertainment Centre (April 10, 2019)

## Appendix B – Dockerfile

```
#Base image of node.js v8
FROM node:8

#Creating app directory
WORKDIR /usr/src/app

#Install app dependencies
COPY package*.json ./
RUN npm install --production
COPY . .

#Access port
EXPOSE 3000

#Run app in production environment
ENV NODE_ENV=production
CMD npm start
```