

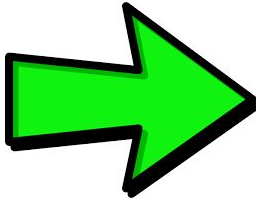
*getBusy*

---

An App For Finding Events In Your City

# getBusy Concept

Our goal was to create an application that enables people to search events by city and present them as pins on a Google map with markers.

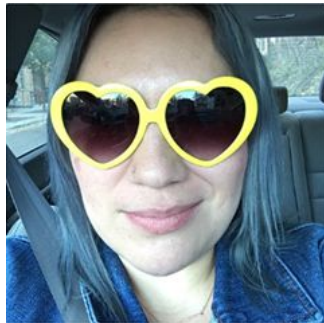


# getBusy Team Members and Responsibilities

Design and UI/UX - Jess Hust

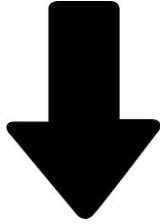
API/Javascript/jQuery- Mike Henry

API/Firebase/jQuery UI - Monica Szabo

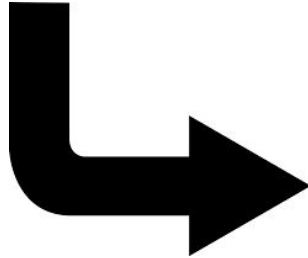


# getBusy's Changing Concepts

Original Goal - Outdoor Activities



Morphed into - Camping specifically



Settled on:

**Random Events in Your City**



# getBusy Design



## Materialize

A modern responsive  
front-end framework based on Material Design

- **Materialize Framework**

- Parallax template
- Responsive, clean design
- Grid system, same as Bootstrap

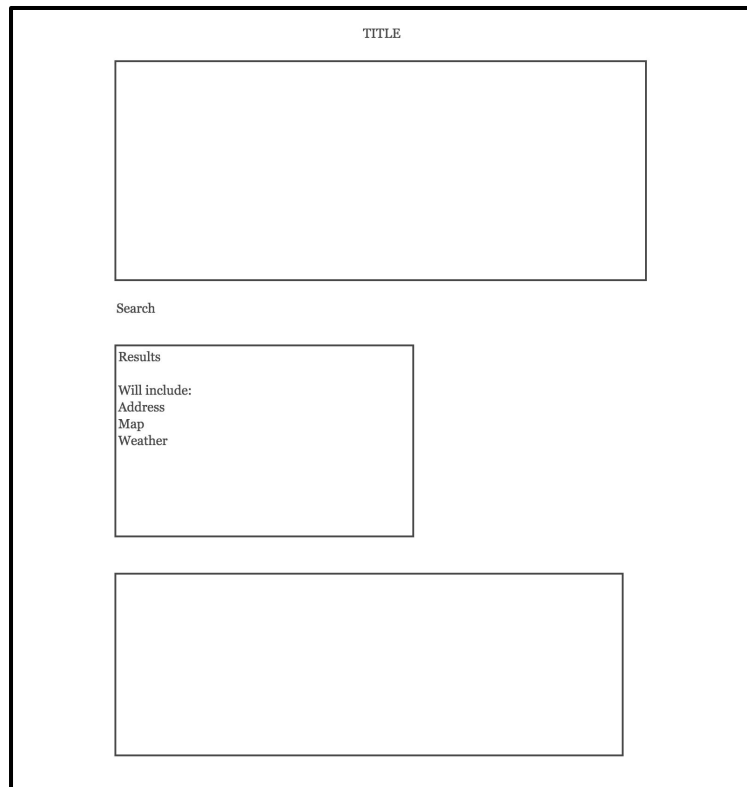
- **User Interface**

- Created one page for simplicity
- Search for a city
- Populate results on Google Map

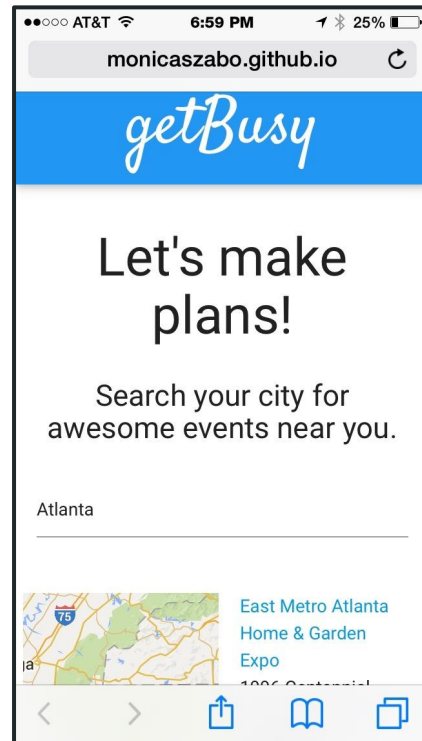
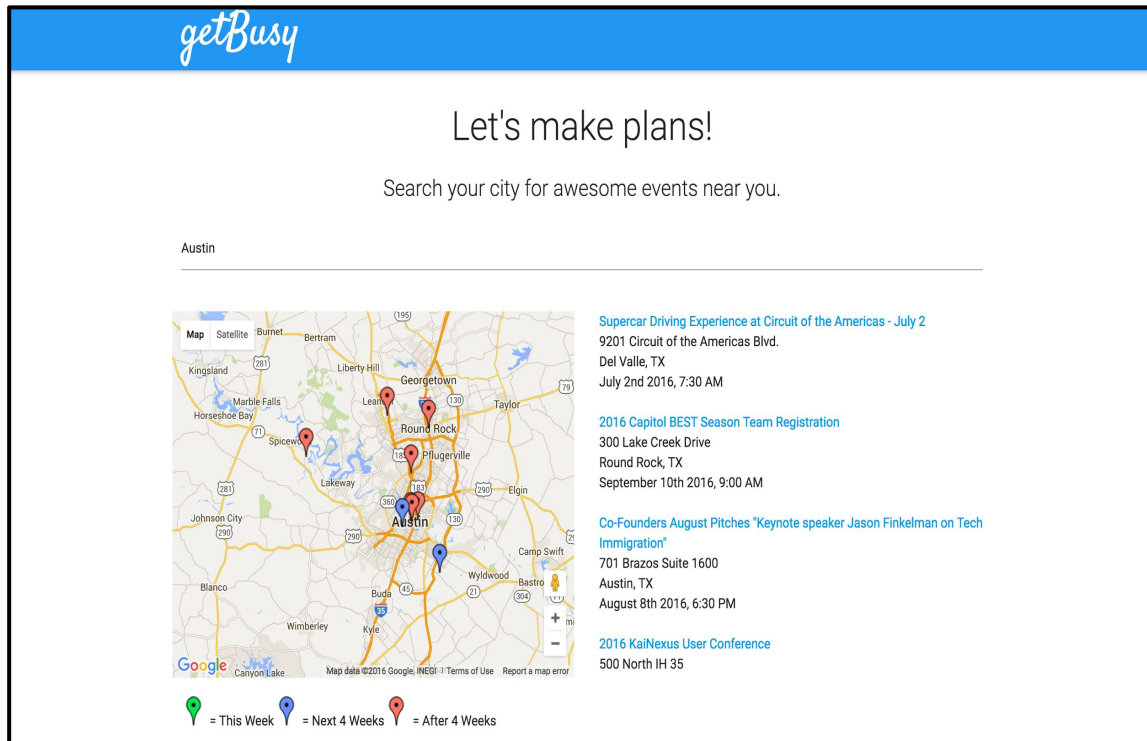
# getBusy Design - Initial Design/Wireframe Mockup

**Initial Design accounted for results and images only.**

- While planning, we wanted to include a Google map and interactive results, images, and directions.



# getBusy Final Design



# getBusy Eventful API

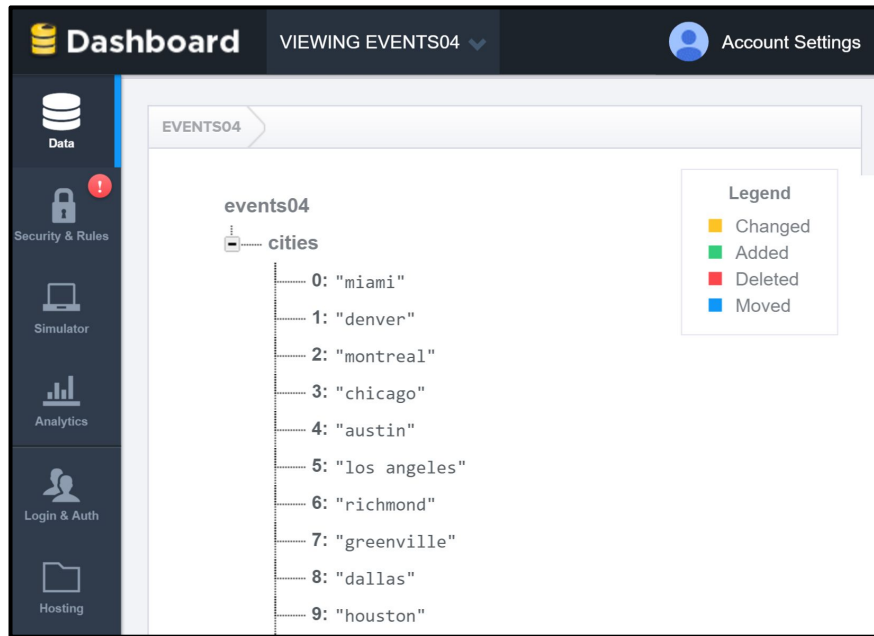
- Used user input to capture which city they want to look for
- Utilized AJAX to obtain the Eventful API JSON for the queried city
- User input is validated
  - Not all cities have events
  - Prevents saving user errors to database
  - Shows **Autocomplete** of successful searches to help user input
- Would save the data to be used in the map
- Employed Moment.js and Eventful to determine timeframe for event

The logo for Eventful, featuring the word "eventful" in a lowercase, rounded font. The letters "event" are blue, and the letters "ful" are green.

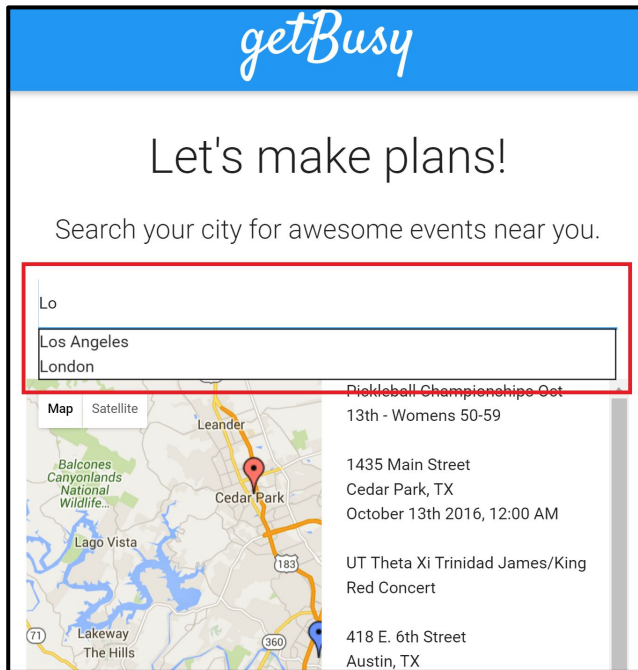


# getBusy Firebase

- Utilized Firebase to store successfully queried cities
- Stored the data into an array of cities
  - If the city is already in the database, does not duplicate
  - If the query does not return results, it is not stored in the database
- This data is then used to complete the jQuery UI Autocomplete



# getBusy jQuery UI

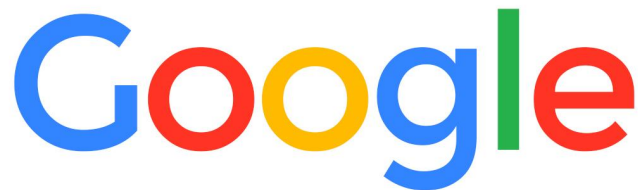


- Employed jQuery UI to create a unique **Autocomplete** of cities
- By working with Firebase, our cities persist if the city creates actual data
  - Will not store junk data or typos
  - If more than 4 cities appear, will be scrollable



# getBusy Google Map

- Render a map in a window
  - Google map API returns visual content instead of a data object
- Use the event API object passed to the map feature to:
  - Place markers on the map and make them clickable
  - Populate hover text and info window
  - Recenter and zoom map to include all markers
- Color code markers to show timeframe of the event
  - Green = this week
  - Blue = next 4 weeks
  - Red = after 4 weeks
- Program the ability to remove the markers when a new search is executed



# getBusy Google Map

- Google maps API documentation is VERY poor but if you just use Google search the information necessary is easily found



- One issue that caused a real slowdown in development was the Google Chrome “feature” that does not allow the use of the geolocation services from a window opened from a “file:///C:/Users” type url

# *getBusy* Demo

---

<http://monicaszabo.github.io/Week8GroupProject/>

# getBusy Future Developments

- Allow the list of events to link to the map markers
- Add directions to the event from your current location
- Add weather API to show expected weather forecast
  - Have a “what to wear” section based on if the event is outdoors/indoors and what the weather is like on the day of the event



WEATHER UNDERGROUND