

# MIWatch

## Technical Documentation

### Technologies Used:

#### **Ruby on Rails:**

Rails is the web framework used to build this application, utilizing the Ruby programming language. Like many contemporary web frameworks, Rails uses the Model-View-Controller (MVC) architecture pattern to organize application programming. Version 2.3.4 (or above) of Rails and 1.8.7 (or above) of Ruby is required to run this application.

#### **Rails Gems:**

Gems are “plugins” utilized by the Rails framework. The following gems are required for this application. They must be installed by the party who is hosting the application.

- geokit (1.5.0 or greater)

#### **Rails Plugins:**

The following plugins are required for this application and already come bundled with the application.

**Note: The geokit tool is required as both a plugin/gem**

- geokit (1.5.0 or greater)
- ym4r\_gm (0.6.1 or greater)

#### **Database Software:**

MySQL is used as the database management system. Database configuration options can be set in: /config/database.yml

#### **Mapping Software:**

We used the Google Maps API to both geocode the locations and then render the map with markers. An API key is required to use the Google Maps API. Each domain requires a separate key. To set yours, go to /config/gmaps\_api\_key.yml and /config/initializers/geokit\_config.rb

## File Structure:

### **/vender**

This folder contains the plugins and gems directories utilized by the application. Each plugin and gem has a separate directory.

### **/config**

This folder contains the configuration files for the application. Database settings and Google Maps API keys are set here.

### **/public/stylesheets**

This folder contains map.css with style settings specific to the map, which are separate from the main site.

### **/app/controllers**

Files in this folder control the flow of application logic.

### **/app/models**

Files in this folder provide connections to database tables and make available the core methods utilized by the controller(s). Each model can be thought of as a class.

### **/app/views**

Files in this folder control what is displayed when the user uses the application. This is where the index.html file and other html files are stored.

## Architecture and data Flow:

### **/import/db**

This is where a MySQL database dump can be found. The database contains location data from New York and New Jersey. The database is stored as one very large table, but see the suggested database structuring document for our suggestions on a proper database format for the future.

### **/import/data**

This is where the data received from the various clients is stored. Each data file is in .csv format. In addition, a ruby file to properly parse and integrate the data into the database is found here.

### **/public/stylesheets/map.css**

This is the main css file used by the application to handle how the map is displayed.

### **/public/stylesheets/map.css**

This is the main css file used by the application to handle how the map is displayed.

### **/app/controllers/map\_controller.rb**

This is the main controller utilized by the application. It is one of the first files called when a user interacts with the application. It then calls methods from the map.rb model.

### **/app/models/map.rb**

This is the main model used by the application. The core method found here is `get_locations`. It handles the database calls (finding the locations) and sends back to the controller an array/hash of the map locations to be displayed on the map. It also generates the html to be displayed on the side panel of the map.

### **/app/models/location.rb**

This is another model used in the application. It has no methods and is required because every database table requires a unique model. Because we make use of various Google Map API functions through the geokit gem/plugin, we have to make the location table “mappable” by calling “`acts_as_mappable`” on the table.

### **/app/views/map/index.html.erb**

This is in the initial page a user sees when he uses the application. The initial search for locations begins on this page.

### **/app/views/map/results.html.erb**

This is the main html file used by the application and where the map is displayed with the locations found. Any changes to how the map, search form, and category options are displayed should be modified here. Once a user does an initial search, all subsequent searches and filtering are done from this page.

### **/app/views/map/filter.rjs**

This is an ajax file used by the application. Any time the user changes which categories are displayed on the map, this file is called. It handles the repopulating of the map by taking what is stored in the array/hash returned by `get_locations` in map.rb. By making use of ajax, the entire page does not have to be reloaded, as only the map has to be changed.

### **/app/views/map/find\_ajax.rjs**

This is an ajax file used by the application. Any time the user searches for locations by a specific zip code, this file is called. It handles the repopulating of the map by taking what is stored in the array/hash returned by `get_locations` in `map.rb`. By making use of ajax, the entire page does not have to be reloaded, as only the map has to be changed.