Introduction

The project is a Pokemon-database made for people who want to keep track of the Pokemon they are growing and their unique attributes. The system works as a database for Pokemon (added by an administrator) and allows users to keep their own list of Pokemon including their level and stats.

The goal is to implement a website which is easy to use and is made of high quality, readable code and with an emphasis on minimalism. My language of choice is Ruby, with use of the Sinatra web-development framework. Client-side code such as Javascript will not be used, to make the site as fluid and gimmick less as possible. The underlying database will be a PosgreSQL one.

User groups

User:

Users can sign up, or just view information about Pokemon that exist in the database.

Regular signed up user:

The regular user will be able to login to add their favourite pokemon to their watchlist.

Administrator

The admin can add Pokemon to the system and edit or delete them. They can also manage their attributes and specific skills.

Use cases

User:

Listing pokemon:

Anybody can list Pokemon in the database and receive an indexing of all the Pokemon as well as some facts about them

Viewing a single Pokemon:

Anybody can view one Pokemon and it's attributes.

Signing up:

Users can sign up to access extra features

Signing in:

Users can sign in to access extra features

Searching:

Anyone can search for a Pokemon using a search string

Regular user:

Adding a pokemon to watchlist:

Users can add a Pokemon to their watchlist

Viewing watchlist:

After signing in, users can view their watchlist for an index of pokemons

Modifying own pokemon

Users can modify the stats of Pokemon in their watchlist

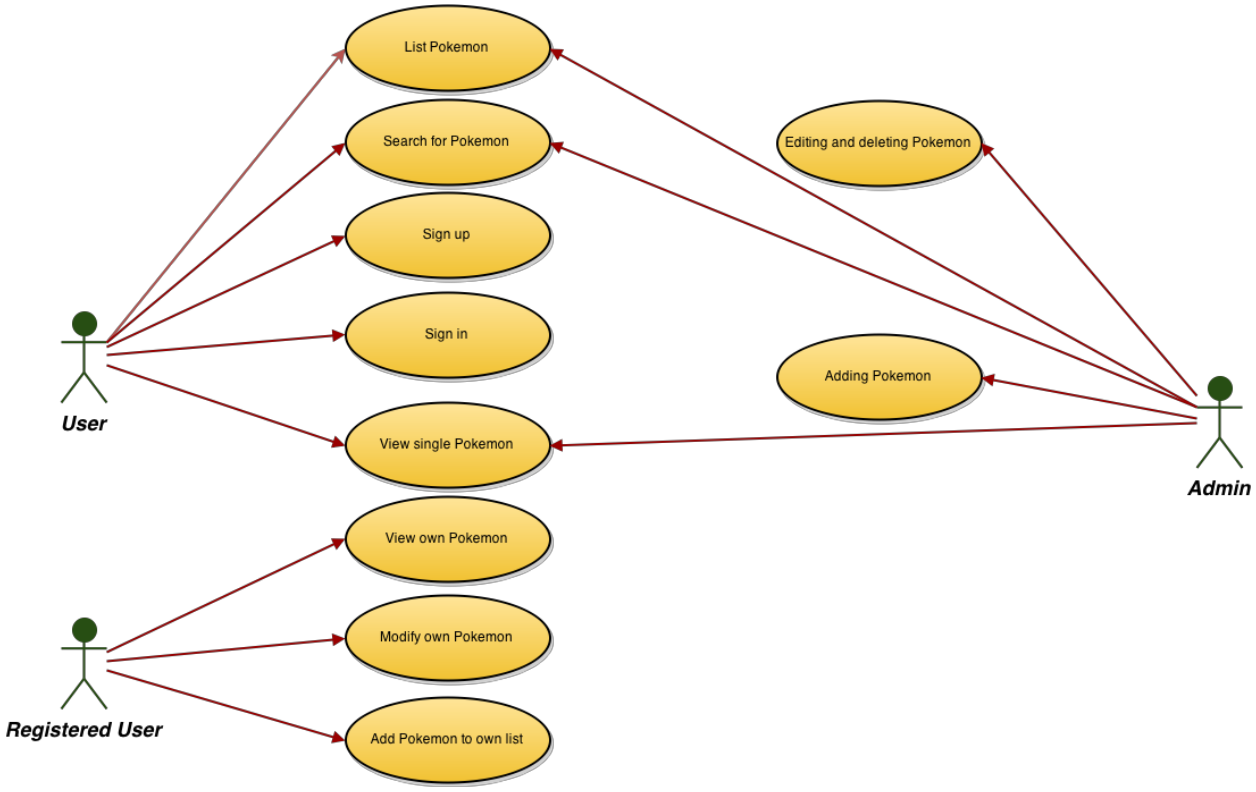
Admin:

Adding Pokemon:

The admin can add pokemon to the system

Editing and deleting Pokemon:

The admin can edit Pokemon in the system, as well as delete them



General structure

The app follows a normal Rails file structure. All functionality is located within the app folder, which in turn includes folders for models, controllers and views separately, as according to the MVC-model. Static assets are located in the assets folder, which includes some of my own CSS and Javascript, as well as those used by Bootstrap.

The db folder stores the database structure. The entire database is viewable in schema.rb, and individual SQL statements used to create the database can be seen in the migrations folder, that is, db/migrate.

In addition, I use the seeds.rb file to seed the first 151 pokemon from Pokeapi to my database. This is a free API from which i get all the pictures, names and descriptions of the Pokemon in the system. This way users and admins don't have to manually add any Pokemon. Later on it would be easy to get the rest of the Pokemon if wanted.

Currently the app is highly dependent on Pokeapi.co. In the future I'd like to get a dump of all the descriptions and Pokemon pictures to my own database to make the app independent.

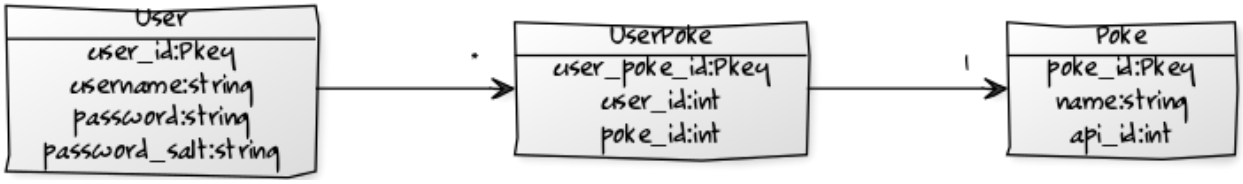
Data structure

Attribute	Datatype	Description
user_id	Integer, 32-bit	Primary key for user
username	String, 50 chars	User's name
password	String, 64 chars	Hashed password
password_salt	String, 64 chars	salt for password

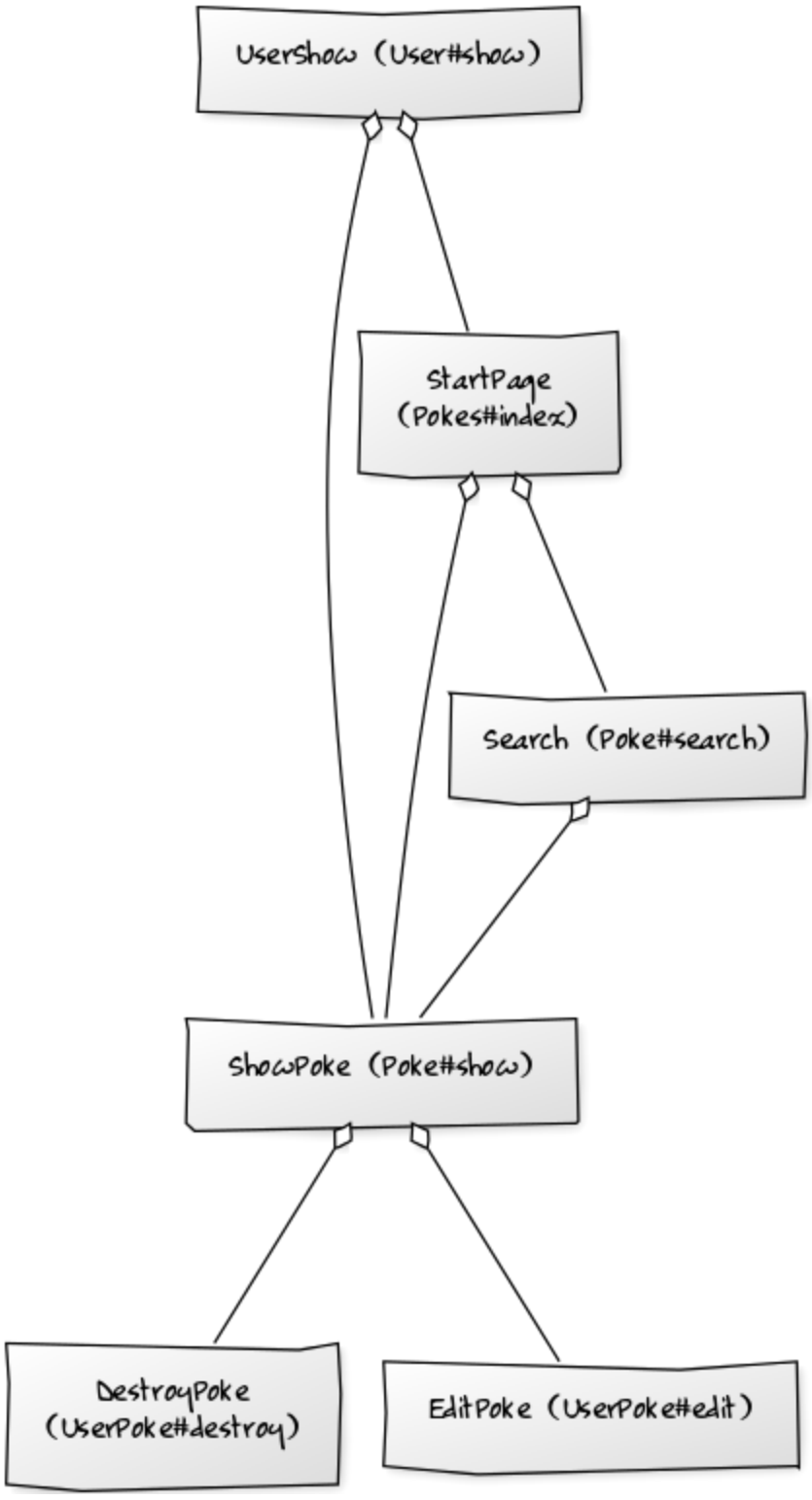
Attribute	Datatype	Description
poke_id	Integer, 32-bit	Primary key for Poke
name	String, 50 chars	Pokemon name
api_id	Integer, 32-bit	Pokemon key in PokeApi

Attribute	Datatype	Description
user_poke_id	Integer, 32-bit	Primary key for UserPoke
user_id	Integer, 32-bit	Foreign key for User
poke_id	Integer, 32-bit	Foreign key for Poke
Level	Integer, 32-bit	UserPoke level
EV	Integer, 32-vit	UserPoke effort value

Relational database structure



Page structure


UserShow (User#show)
StartPage
(Pokes#index)
Search (Poke#search)
ShowPoke (Poke#show)
DestroyPoke
(UserPoke#destroy)
EditPoke (UserPoke#edit)

How to setup

The application is easy to setup. You will need a Ruby on Rails environment with Rails 4.2.0 and RVM 2.2. You'll also need a PostgreSQL database. Just set the database address in config/database.yml. Before starting the server make sure to run rake db:migrate. the command "rails s" with start the server.