<u>**Team**</u>**: William Lu**
      Perl Module Theodore "Big T" Peters
      Ivan "R." Zhang
      Adil "Pickle" Gondal
      Imad "The Flip-Flop" Belkebir
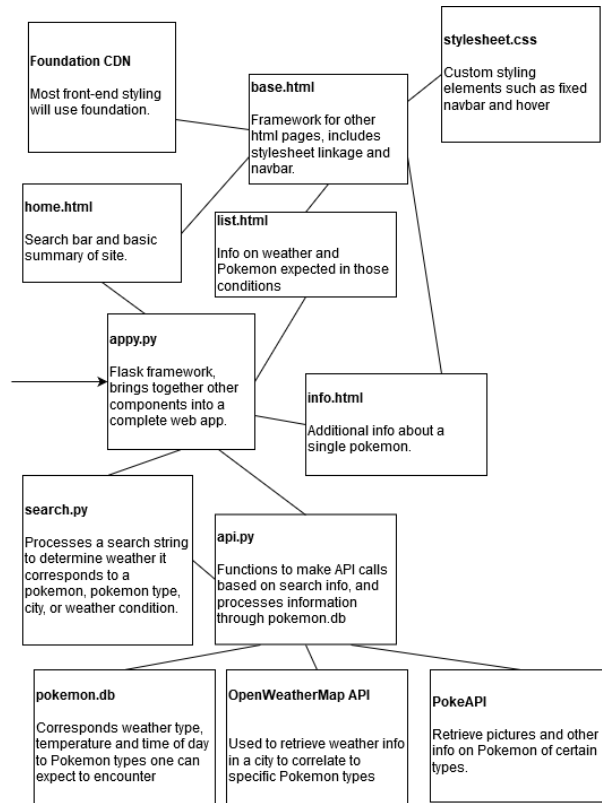   Honorary Member William "Squidward" Lu

<u>**Site**</u>**: Weather You'll See Pokemon** Combine the OpenWeatherMap API (for weather) and a PokeAPI (for Pokemon data) to determine which pokemon one can expect to see in their city based on weather

Core Components
- Search a location given by the user in the Open Weather API, retrieve forecast
- Correspond weather forecast (time of day, temperature, wind, sunny/raining/overcast) to different pokemon types
- Display pokemon types in readable, clean manner
- Use Foundation to style front-end
- Nice looking view of weather & pokemon user can expect to find (cards w/ images)
- Extra info about pokemon on user request

Extra
- Ability to filter pokemon
- Autocomplete on searching city/pokemon
- Display a map of user's location
- Fancy CSS hover menus for info on Pokemon
- Store last entered city with cookies so user can quickly look up their city across multiple uses
- Search where to find a specific Pokemon/type nearby
- Random city
- Predict when you might see pokemon with week ahead forecast
- If you type William Lu you will be told you about the ultra rare William Lu pokemon (fighting type)

Foundation CDN

Most front-end styling will use foundation.

base.html

Framework for other html pages, includes stylesheet linkage and navbar.

stylesheet.css

Custom styling elements such as fixed navbar and hover

home.html

Search bar and basic summary of site.

list.html

Info on weather and Pokemon expected in those conditions

appy.py

Flask framework, brings together other components into a complete web app.

info.html

Additional info about a single pokemon.

search.py

Processes a search string to determine weather it corresponds to a pokemon, pokemon type, city, or weather condition.

api.py

Functions to make API calls based on search info, and processes information through pokemon.db

pokemon.db

Corresponds weather type, temperature and time of day to Pokemon types one can expect to encounter

OpenWeatherMap API

Used to retrieve weather info in a city to correlate to specific Pokemon types

PokeAPI

Retrieve pictures and other info on Pokemon of certain types.

Database Schema

We plan on having a simple database that stores the type of weather and the types of Pokemon that correlate with that weather type. For example, if it is raining, we want a Pokemon that is a water type. If it is snowing we would want a Pokemon that is either ice or water type, so the database would return both of those values.
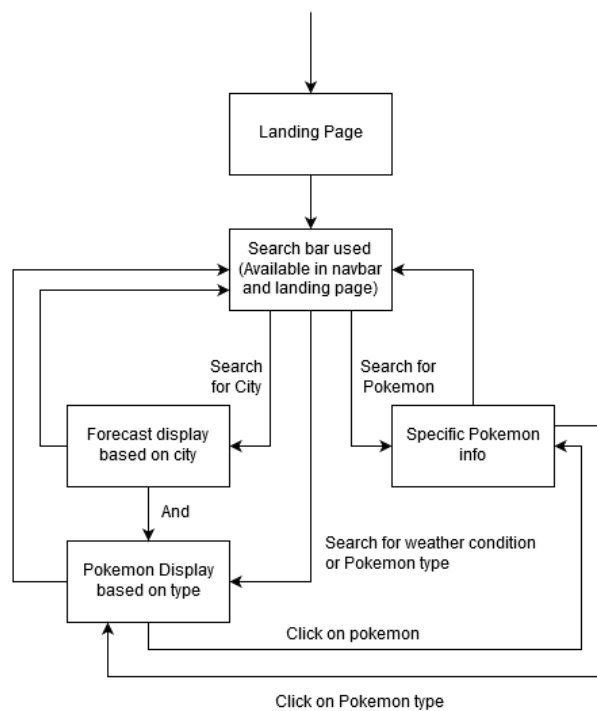
As opposed to the last project where the database was constantly being updated, the current database is going to be constant throughout. We are going to hard code the types of weather, time of day, temperature, and wind as per the Open Weather API, and all the possible types of Pokemons that correlate with them.

Type of Weather	Type of Pokemon
Rain	Water, ...
Thunderstorm	Electric, ...
Snow	Ice, Water, ...

etc.	etc.

Although we are only going to have one database stored with our website, we will be using multiple databases through our API use. When we are working with weather data, we will be implementing the OpenWeatherMap API and the use of its database. The PokeApi will be used to search pokemon by either type or name and get stats and pictures of said pokemon.

Sitemap

First Tasks

Page styling (Styling with foundation and custom styling for hover menus & navbar etc.) - Ivan "R." Zhang

API handling (Library of functions to make API calls, procuring API keys, processing API data to more usable dictionaries)- Adil "Pickle" Gondal

Pokemon-weather database dealings (Database to correspond weather to pokemon types, and methods to access this database with weather info to return a list of pokemon types that might be seen, ordered based on likelihood.) - Imad "The Flip-Flop" Belkebir

Flask Apparatus & Search functionality (processing user input to correspond to calls to different APIs) - Theodore "Big T" Peters

General inspiration & Being Cool - William "Squidward" Lu