

## PokeDex WebApp- Done by Jason, Ralph, Tyler

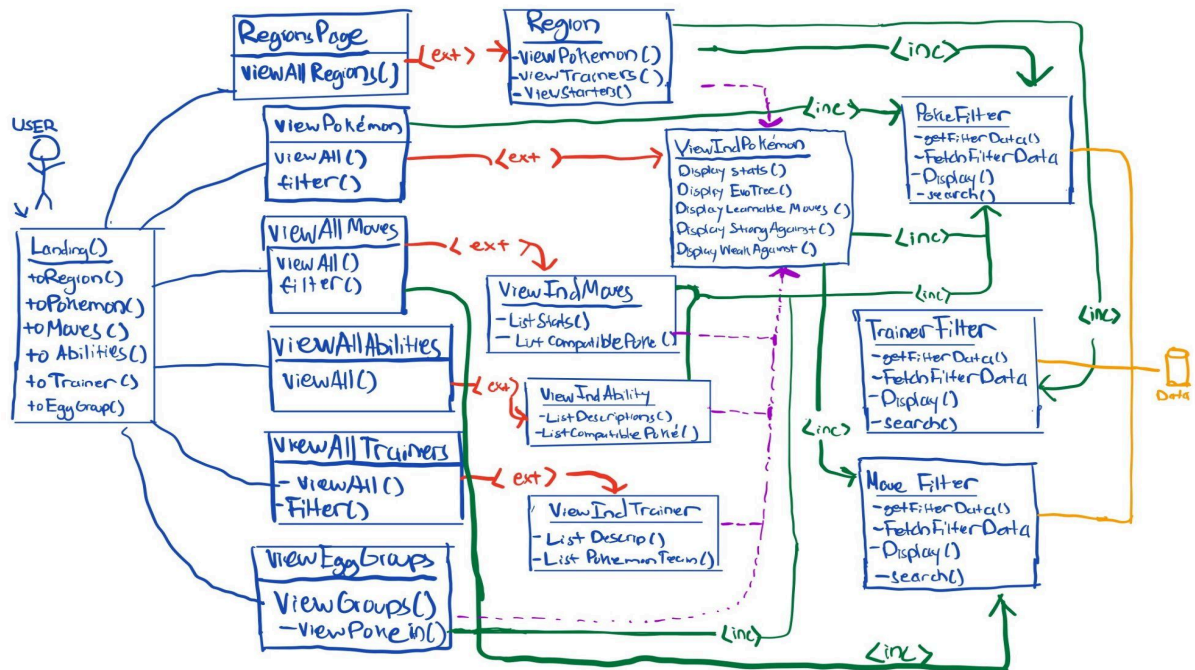


### Project Description:-

Pokedex web application to view general information about the game of Pokemon.

Includes aspects of the Pokemon video game. Allows user to learn about pokemon and inspect interactions of pokemon

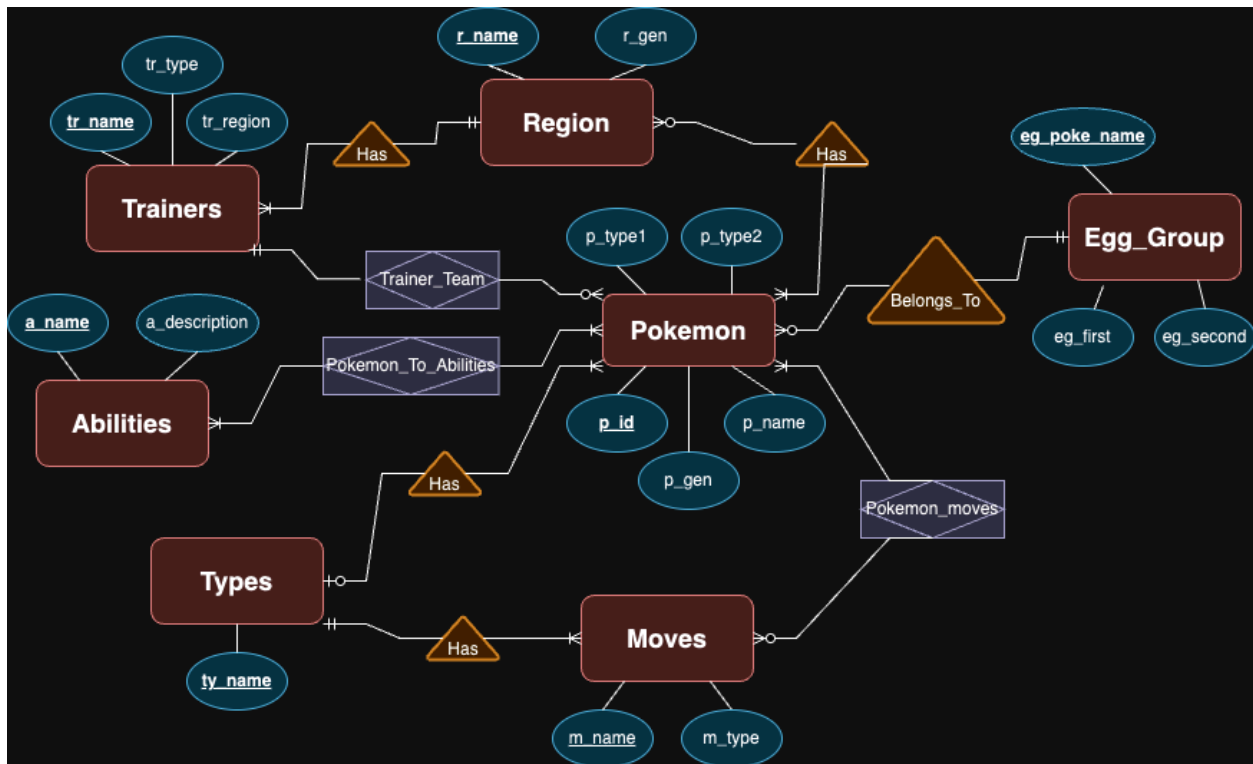
## UML Diagram:-



- The main use case is giving the user detailed information about any Pokemon or move.
- From the landing page the user is able to explore information based off a number filters including regions, pokemon, moves, abilities, trainers, and egg groups.
- The user selects a main category to explore and then applies additional filters to the given results
- The user is then able to select and individual item (eg. A specific pokemon, move, trainer) and receive detailed information about the given item.

- The system displays attributes such as stats, evolutions, element interactions and other items.

## ER Diagram:-



## Relational Schema:-

```
CREATE TABLE trainers (
    t_id    INT NOT NULL,
    t_region TEXT NOT NULL,
    t_gen   INT NOT NULL,
    t_name  TEXT NOT NULL,
    t_type  TEXT NOT NULL,
    t_role  TEXT NOT NULL
);
```

```
CREATE TABLE region (
    r_region_name TEXT NOT NULL,
    r_gen         INT NOT NULL,
    r_fire_starter TEXT NOT NULL,
    r_water_starter TEXT NOT NULL,
    r_grass_starter TEXT NOT NULL,
    r_box_1       TEXT,
    r_box_2       TEXT,
    r_box_3       TEXT
);
```

```
CREATE TABLE trainer_to_pokemon (
    tp_trainer TEXT NOT NULL,
    tp_pokemon TEXT NOT NULL
);
```

```
CREATE TABLE pokemon_to_abilities (
    pa_pokemon TEXT NOT NULL,
    pa_ability1 TEXT NOT NULL,
    pa_ability2 TEXT,
    pa_hidden_ability TEXT
);
```

```
CREATE TABLE abilities (
    a_name TEXT NOT NULL,
    a_description TEXT NOT NULL,
    a_generation INT NOT NULL
);
```

```
CREATE TABLE egg_groups (
    e_pokemon TEXT NOT NULL,
    e_egg_group_1 TEXT NOT NULL,
    e_egg_group_2 TEXT
);
```

```
CREATE TABLE typeChart (
    tc_type TEXT NOT NULL,
    tc_type_against TEXT NOT NULL,
    tc_effectiveness DOUBLE NOT NULL
);
```

```
CREATE TABLE pokemon (
    p_against_bug DOUBLE NOT NULL,
    p_against_dark DOUBLE NOT NULL,
    p_against_dragon DOUBLE NOT NULL,
    p_against_electric DOUBLE NOT NULL,
    p_against_fairy DOUBLE NOT NULL,
    p_against_fight DOUBLE NOT NULL,
    p_against_fire DOUBLE NOT NULL,
    p_against_flying DOUBLE NOT NULL,
    p_against_ghost DOUBLE NOT NULL,
    p_against_grass DOUBLE NOT NULL,
    p_against_ground DOUBLE NOT NULL,
    p_against_ice DOUBLE NOT NULL,
    p_against_normal DOUBLE NOT NULL,
    p_against_poison DOUBLE NOT NULL,
    p_against_psychic DOUBLE NOT NULL,
    p_against_rock DOUBLE NOT NULL,
    p_against_steel DOUBLE NOT NULL,
    p_against_water DOUBLE NOT NULL,
    p_attack INT NOT NULL,
    p_base_egg_steps INT NOT NULL,
    p_base_happiness INT NOT NULL,
    p_base_total INT NOT NULL,
    p_capture_rate INT NOT NULL,
    p_classification TEXT NOT NULL,
    p_defense INT NOT NULL,
    p_experience_growth INT NOT NULL,
    p_height_m DOUBLE NOT NULL,
    p_hp INT NOT NULL,
    p_japanese_name TEXT NOT NULL,
    p_name TEXT NOT NULL,
    p_percentage_male DOUBLE NOT NULL,
    p_id INTEGER PRIMARY KEY,
    p_sp_attack INT NOT NULL,
    p_sp_defense INT NOT NULL,
    p_speed INT NOT NULL,
    p_type1 TEXT NOT NULL,
    p_type2 TEXT,
    p_weight_kg DOUBLE NOT NULL,
    p_gen INT NOT NULL,
    p_gender TEXT,
    p_rarity TEXT
);
```

## Implementation Details:-

- Flask Application
- Connected DB through sqlite3 commands.
- Executed queries from flask route.
- Send query data to frontend HTML and parse with Jinja2.
- Each html page can link to other flask routes to access other pages.

