CS348 Project – Milestone 2

Project Description:

The purpose of this application will be to provide an easy and accessible way for fans of the Pokémon series to browse a list of Pokémon to create their own parties, compare them with that of other users, and take part in battles against other players or bots. Here is how the user interacts with the main pages:

1. Login page: When a user enters the site, they get the option to enter their username and PIN. Then the user clicks "Login" and they are taken to the Pokémon Browser page. As for new users, they can enter a username, and PIN, and then click "Create Account."

2. Pokémon Searcher: After logging in, the user is taken to a page with a list of Pokémon which has information such as their Pokédex number, name, and type. Here are the main features of this page:
   - Above the list there is a search bar where the user can enter the name of the Pokémon they are looking for and hit "Search" which displays all the Pokémon matching the search.
   - From the list they may select certain Pokémon as their favourite by clicking the checkbox column next to their name.
   - They can also click their profile icon which will take them to the Profile Page.
   - Clicking on the name of a Pokemon in the search results will bring the user to a page with more detailed information about the Pokemon as well as an image of the Pokemon.
   - There is also a (heart-shaped) button to "Favourite" a Pokemon for easy access in the Profile page

3. Profile Page: This page has the basic user information. Some of the features of this page are:
   - View profile information such as their name, username, and the date they joined in.
   - A table showing a list of their favourite Pokémon.
   - (To be implemented) From here they can click the table which takes them to the Party Analysis page.

4. Detailed Pokemon Information Page: This page shows detailed information about the Pokemon as well as an image of the Pokemon and is accessed by clicking on the name of a Pokemon from the Pokemon Searcher.

5. Catch Pokemon Page: This page allows the user to add randomly generated Pokemon instances with randomized stats and moves to their collection. It shows each random Pokemon instance's stats/moves/image and their deviations from the Pokemon species' base stats to help the user decide if they want to add the Pokemon to their collection.

6. Organize Pokemon Page: This page allows the user to add/remove Pokemon in their collection to their Party and assign an order for Pokemon in their party. They may also remove Pokemon

from their collection. The page also provides a table displaying summary information of the Pokemon instance as part of the graphical user interface to improve usability/user experience.

7. (To be implemented) Party Analysis: This page allows the user to view their parties in more detail. There is a table for each of the parties they have, and each party has a maximum of 6 Pokémon in them. For each entry in the list, they can see the stats of the Pokémon and the moves that they have learnt. They can also click "Share" which will take them to the Share Parties page.

8. (To be implemented) Share Parties: This page displays a table of Cards, where each card represents parties of a certain player.
   - Clicking the cards allows the user to view the party in more detail, similar to how they are in the user's own Party Analysis page. However, they are not able to see the moves that the Pokémon has.
   - There is also a search bar where the user can enter a username and hit "Search", which looks up a player and shows the parties that player has.

Other planned features:

1. Pokémon Battle: The user will have the option to challenge other active users to Pokémon battles between their parties in a turn-based format similar to how they are in the Pokémon games. There will also be the option to battle bots as well.

The plan for getting the data to Populate the database:

The dataset we use will be based off of https://www.kaggle.com/rounakbanik/pokemon made by Rounak Banik on Kaggle. This dataset was scraped from http://serebii.net (a Pokemon fan-site). This dataset should be sufficient for our application with some restructuring of the data. If additional data is needed, we are prepared to scrape additional data based on Pokemon games from https://www.smogon.com/ , https://bulbapedia.bulbagarden.net/ , and http://serebii.net.

We will also get Type and Type Matchup data from https://rankedboost.com/pokemon-sun-moon/type-chart/.

We also will use a custom WebScraper tool (webscraper/moveScraper.py) we built to scrape a list of Pokemon moves from https://bulbapedia.bulbagarden.net/wiki/List_of_moves.

Examples of what the data looks like once imported into our Database looks like are provided in test-production.out. The corresponding queries which generated the query outputs is given in test-production.sql.

Programming framework chosen and Database support:

We have selected to use Google Cloud Platform (GCP) for our project. We decided to take advantage of the ability to create a PHP application with GCP App Engine that interfaces with MySQL as it seems to provide sufficient tools to create a dynamic and responsive web application (important as the application will be a multi-player game).

Some instances of the queries can be found in test-production.sql. Queries to initialize and clean the database are provided in the CreateTableSQLQueries directory.

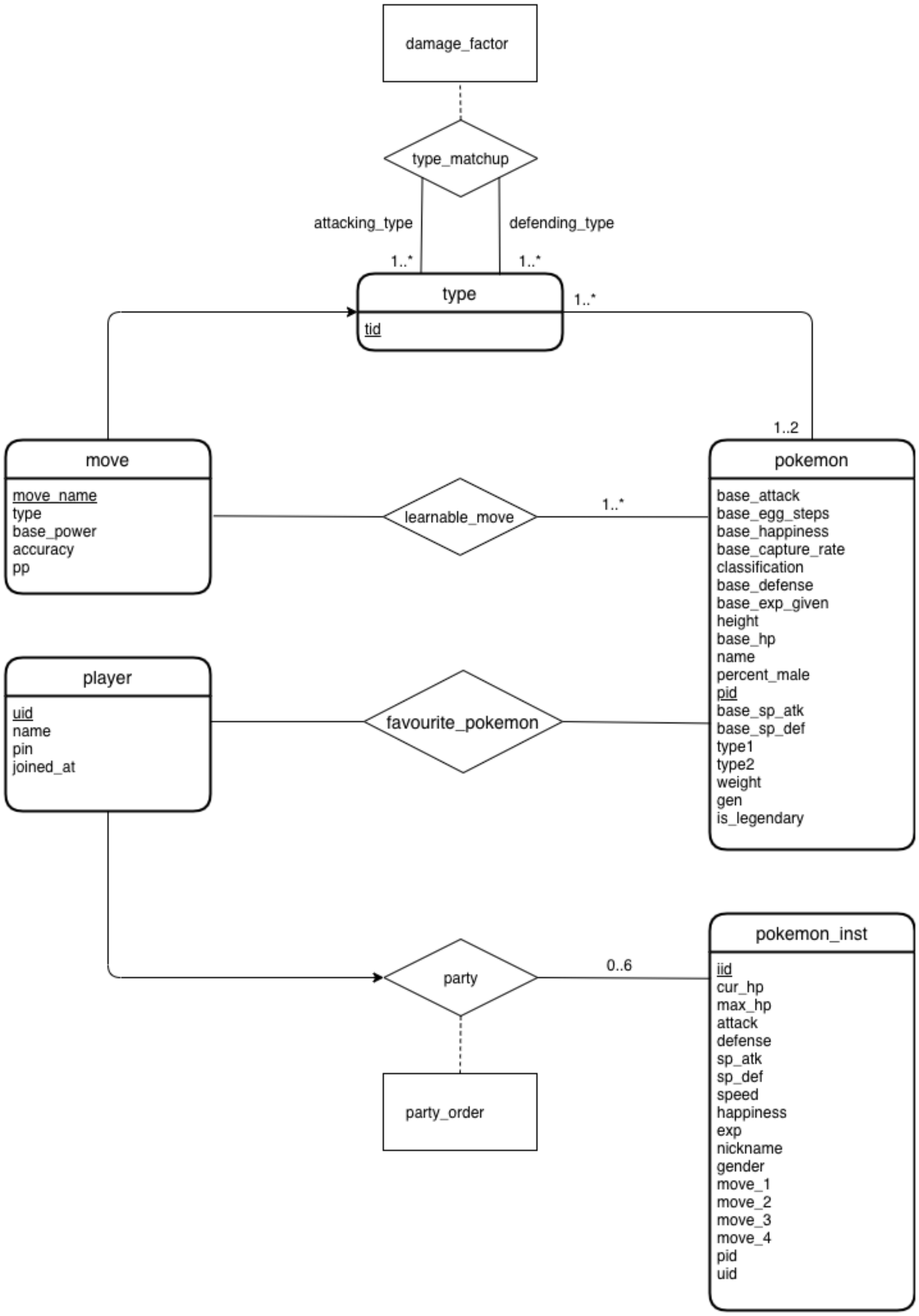
For design, each of the webpages we are we are using Bootstrap as our CSS framework. The reason for choosing Bootstrap is that the whole process of designing the pages is much faster as most of our team members are familiar with using it.

Database Schema:

List of some assumptions about the data being modelled:

- Each player has a unique User ID.
- A player may have multiple parties, and each party is owned by a certain player
- A Pokémon may have multiple instances and many instances of the same Pokémon may be in the same party
- A single user may have multiple favourite Pokémon.
- A party for a user has between 0 and 6 Pokemon (inclusive)
- Each Pokemon has at least one learnable move
- Each pair of Types exists in the Type_Matchup table

E/R Diagram:


damage_factor
type_matchup
attacking_type
defending_type
1..*
1..*
type
tid
1..*
1..2
move
move_name
type
base_power
accuracy
pp
learnable_move
1..*
pokemon
base_attack
base_egg_steps
base_happiness
base_capture_rate
classification
base_defense
base_exp_given
height
base_hp
name
percent_male
pid
base_sp_atk
base_sp_def
type1
type2
weight
gen
is_legendary
player
uid
name
pin
joined_at
favourite_pokemon
party
0..6
party_order
pokemon_inst
iid
cur_hp
max_hp
attack
defense
sp_atk
sp_def
speed
happiness
exp
nickname
gender
move_1
move_2
move_3
move_4
pid
uid

Relational Model

Schema	**Keys**
type_matchup	attacking_type, defending_type
type	tid
move	move_name
learnable_move	move_name, pid
pokemon	pid
favourite_pokemon	uid, pid
player	uid
party	uid, iid
pokemon_inst	iid