CS348 Project – Milestone 3 (Final Report)

# 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. Here is how the user interacts with the main pages:

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. The battle page contains a constant polling request to the server to obtain new updates every 5 seconds (as well as queuing for battles).
2. 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.”
3. 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 Pokémon in the search results will bring the user to a page with more detailed information about the Pokémon as well as an image of the Pokémon.
* There is also a (heart-shaped) button to “Favourite” a Pokémon for easy access in the Profile page

1. 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.

1. Detailed Pokémon Information Page: This page shows detailed information about the Pokémon as well as an image of the Pokémon and is accessed by clicking on the name of a Pokémon from the Pokémon Searcher.
2. Catch Pokémon Page: This page allows the user to add randomly generated Pokémon instances with randomized stats and moves to their collection. It shows each random Pokémon instance’s stats/moves/image and their deviations from the Pokémon species’ base stats to help the user decide if they want to add the Pokémon to their collection. All moves are valid in being learnable for the specific Pokémon (obtained through web-scraping).
3. Organize Pokémon Page: This page allows the user to add/remove Pokémon in their collection to their Party and assign an order for Pokémon in their party. They may also remove Pokémon from their collection. The page also provides a table displaying summary information of the Pokémon instance as part of the graphical user interface to improve usability/user experience. There are also additional checks to prevent any edge-cases such as releasing Pokémon that are in your party.
4. 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 (as well as a redirect to the base information of the Pokémon).
5. Share Parties: This page displays a table of Cards, where each card represents parties of a certain player. Each card shows one of the Pokémon in the party as well as names for all of the Pokémon.

* 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.
* 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.

**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 Pokémon 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 Pokémon 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 Pokémon moves from <https://bulbapedia.bulbagarden.net/wiki/List_of_moves>.

We will also use a custom WebScraper tool (webscraper/ moveLearnScraper.py) we built to scrape a list of Pokémon moves from <http://serebii.net>. We will also use the custom python script (webscraper/csv\_undup.py) to remove duplicates from the learn-set (generated when a Pokémon can learn a move in multiple ways).

A tiny bit of manual data cleaning was necessary (generating a standardized move name by removing white space and capitalization as well as fixing spelling, notably ‘Vise Grip’ to ‘Vice Grip’ from Bulbapedia). We also used custom python scripts webscraper/moveToSQL.py and webscraper/ learnableMoveToSQL.py to create an SQL insert script to import the csv made by the webscrapers into the MySQL database. Note some moves are rejected from insertion into the MySQL database currently as we require moves to be damaging and have an accuracy and pp (all not null). The database automatically filters and rejects moves missing these values (this removes status moves and moves with special effects as they are not implemented in our application).

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.

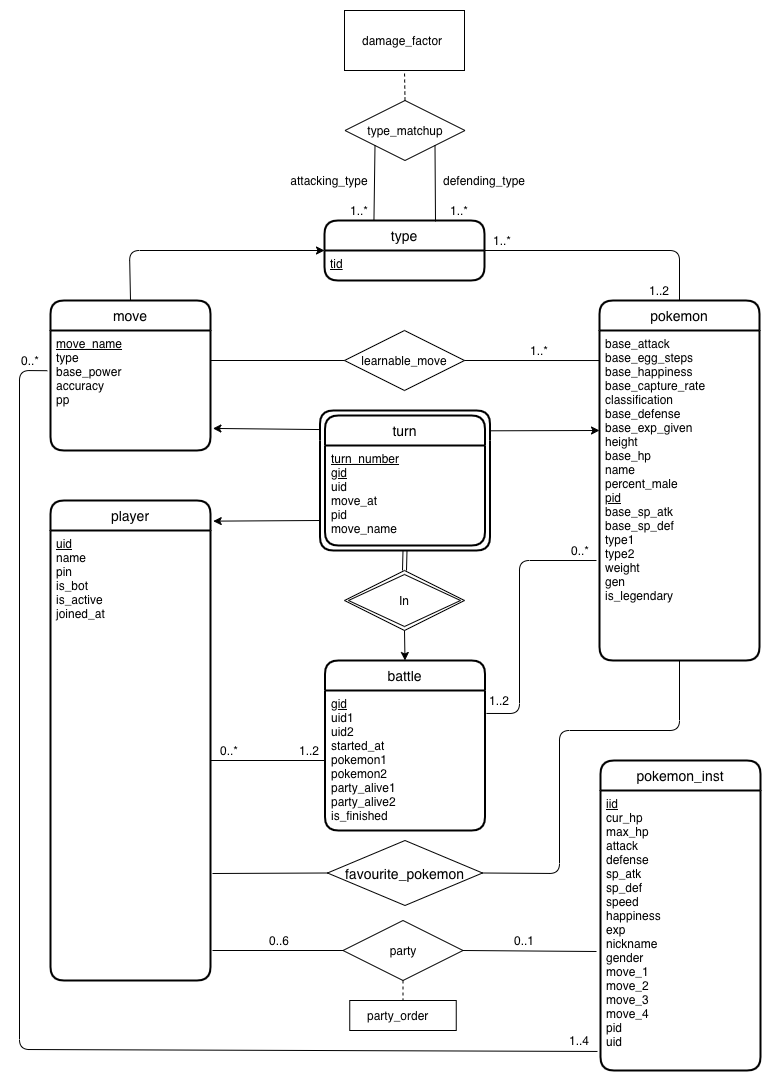
For design, each of the webpages we are we are using Bootstrap as our CSS framework. The reason for choosing Bootstrap is that the whole process of designing the pages is much faster as most of our team members are familiar with using it.

# 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 Pokémon (inclusive)
* Each Pokémon has at least one learnable move
* The party\_alive1 and party\_alive2 strings in the battle table are comma delimited and is a list of references to valid pokemon\_inst iid belonging to the corresponding player represented by uid1 and uid2 respectively (this can be moved to a separate relation in the future)
* The pokemon1 and pokemon2 fields of the battle table reference valid iid from pokemon\_inst and are also a valid pid from pokemon (bug to fix)
* Each pair of Types exists in the type\_matchup table
* The pid of each Pokémon in the pokemon table is its Pokédex number

**E/R Diagram:**

****

**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 |
| battle | gid |
| turn | turn\_number, gid |