

Final Report for DBMS Project

By: Oscar Barrera, Alex Barton, Eric Buckner, Marc Casper, Austen Dyer, Chris Ebner, Hadyn McIntyre

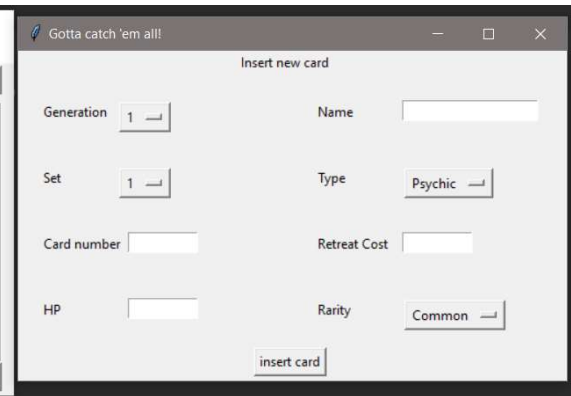
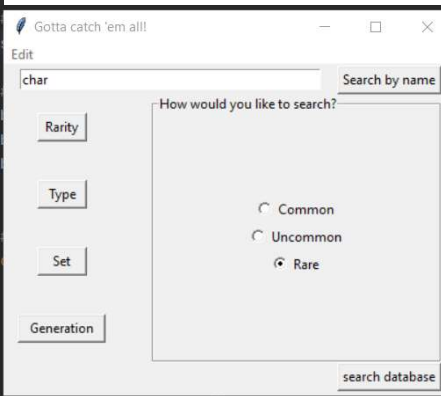
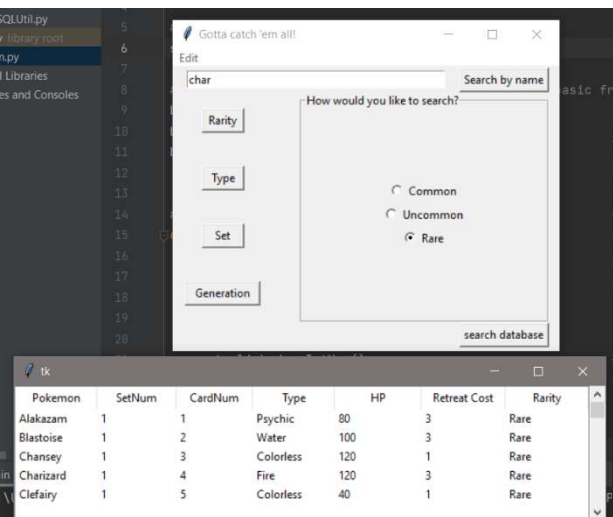
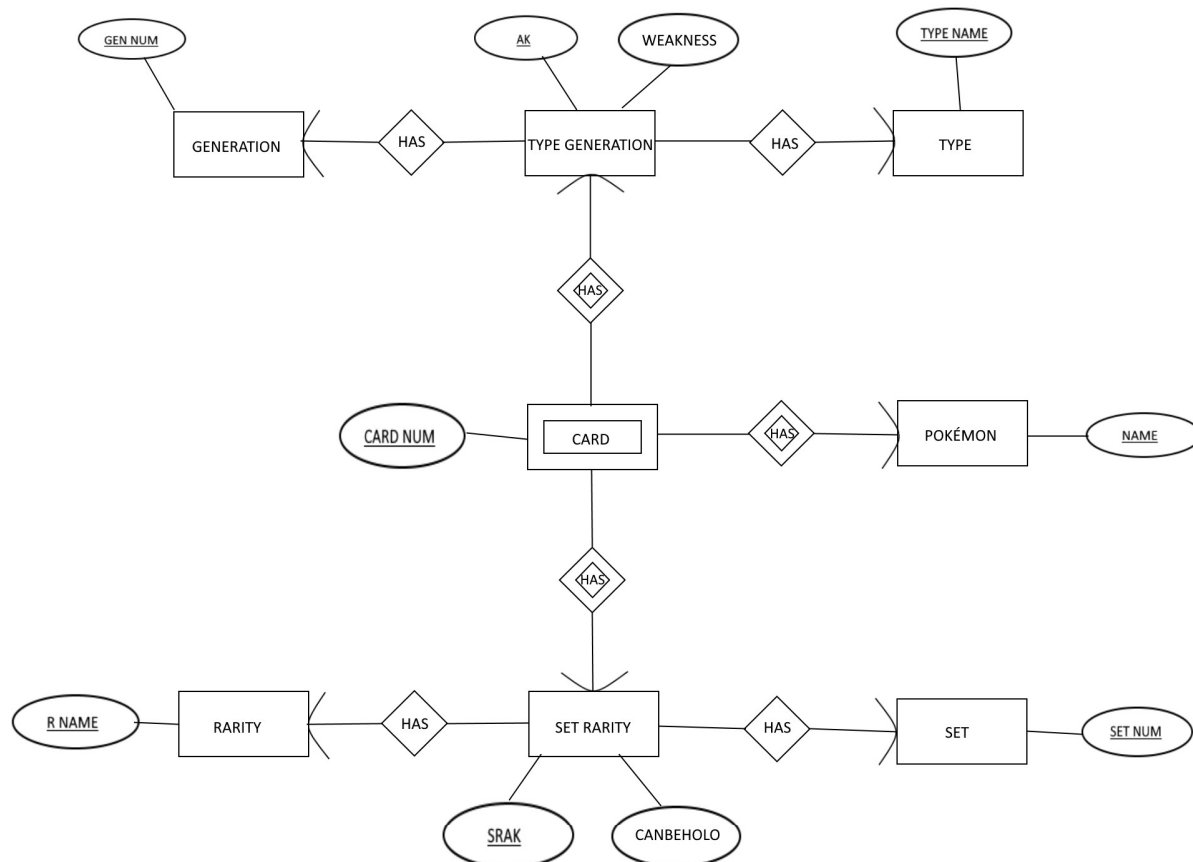
The Super-Efficient and Wonderfully Inventive Pokémon Database with Lovely GUI for Happy Fun Time is a database built to catalog Pokémon trading cards with a corresponding Graphical User Interface to allow for searches and alterations to the database. The database allows for the cataloging of the specifics of each Pokémon card, as well as making it easier to search by different parameters that a collector may wish to search by. These parameters include rarity, type, name, and evolutionary status from specific generations. To safeguard the integrity of the data there are user requirements implemented planned for use in the database. For the base user, who will be the standard everyday user, there are no requirements to view or search the database for cards meeting their selected criteria. However, if one wishes to add or delete cards, they will need to be an admin and log in to their account to do so.

The use of the entity relationship diagram that we built allowed for the mapping of proper relations and keys. In so doing, this process made it possible to write the appropriate relational schema. With the entity relationship diagram and relational schema completed we were able to utilize the information therein to write the code for and to build our database. To implement the code for the database we utilized the MySQL database management system. For the front-end programming we selected Python for the ease of use and the members familiarity with language. The graphical user interface, or GUI, was written using the Tkinter framework in Python. The framework allows for the code to be written in a concise and easy to understand manner. The code then outputs a clean GUI, thus increasing the ease of use and accessibility of our database application for the end user. To connect our database to Python we made use of `mysql.connector`.

Moving forward with our project, we have compiled a list of features to be added in future iterations. The first of concern we had was that of security. At the time of this writing, we have yet to implement a proper security protocol. As it stands, security encryption and logins are not needed to insert or delete cards. By not having these security features in place it allows anyone to make changes to the database. As there is not currently a way to control who makes these changes, we are not able to assure the integrity of the database, thus losing the faith of our customer/users. The devastating nature of this potential integrity loss due to the lack of a security protocol is the reason for having this item as our highest priority. To rectify this, we intend to implement an admin-login for those who have the proper credentials to insert to, or delete from the database. This security protocol would also include encrypting the password information using SHA-256. Once this item has been remediated, we plan to move to “quality-of-life” items. Specifically making it possible to add new cards to more than just the existing sets. We would like to be able to add to all our relations such as new sets, new generations, etc. This would allow our database to grow in a manner more befitting the evolution of the Pokémon trading card game rather than locking the user into a strict and possibly outdated model. The final item we currently must address for future versions would be to add more granularity in the way a user can search the database. In its current iteration we only allow for radio buttons to make one selection from a list of sets, rarities, etc. To improve upon this, we would like to implement a checkbox system so that a user can search by multiple selections, rather than being only able to use one. This opens the functionality of our database allowing for more advanced searches, more accurately representing the

complexity of the Pokémon trading card game and thus making it a more attractive option for a user to select.

While creating our database we learned a great many skills that will benefit us in the future. We learned how to create a relational schema in 3NF and implement it into MySQL in a cogent and professional manner, as well as how to connect MySQL to Python. We also learned how to use the Tkinter framework to make a professional-grade user-interface, allowing for users to search our database for their desired card. In addition, we learned several soft skills that will greatly benefit us in our academic, as well as professional, careers. Such skills include the ability to effectively work as a team and communicate our ideas to one another to put together a well-functioning project in a timely manner. By the very nature of a group project with a functioning team, we learned how to build upon one another's strengths and cover one another's weaknesses. This ability is invaluable in every asset of life that requires you to work together with others. The skills gained from working on this project have bolstered the skills of us all, ultimately making us more proficient computer scientists.



Gotta catch 'em all!

Edit

char Search by name

How would you like to search?

Rarity

Type

Set

Generation

☐ Common

☐ Uncommon

☒ Rare

search database

Gotta catch 'em all!

Insert new card

Generation

Name

Set

Type

Card number

Retreat Cost

HP

Rarity

insert card

```

5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

```

Gotta catch 'em all!

Edit

Search by name

How would you like to search?

Rarity

Type

Set

Generation

☐ Normal type

☐ Dark type

☐ Fighting type

☐ Fire type

☒ Grass type

☐ Electric type

☐ Psychic type

☐ Steel type

☐ Water type

search database

tk

Pokemon	SetNum	CardNum	Type	HP	Retreat Cost	Rarity
Nidoking	1	11	Grass	90	3	Rare
Venusaur	1	15	Grass	100	2	Rare
Beedrill	1	17	Grass	80	0	Rare
Ivysaur	1	30	Grass	60	1	Uncommon
Kakuna	1	33	Grass	80	2	Uncommon
Nidorino	1	37	Grass	60	1	Uncommon
Bulbasaur	1	44	Grass	40	1	Common