PokeDex by Team **Laser Sharks**

Alice Ni, Connor Oh, Leia Park, Hilary Zen

OBJECTIVE

The PokeDex is an ultra, super-deluxe, mega, revamped version of the classic handheld encyclopedia device of pocket monsters.

FEATURES

- Displays description and details of every known pokemon
- Plays Pokemon cries (https://pokemoncries.com/)
- Who’s that pokemon? Game
- “Scans” a pokemon and identifies it for you
- Can sort the pokemons based on: type, generation, evolution, alphabetical, region, etc.; Can also show random order
- Spider diagrams for the pokemon’s stats (possibly)
- Search bar for pokemon by name & number
- Login/register system. Users can save their favorite pokemon to their personal pokedex. Uses a database.

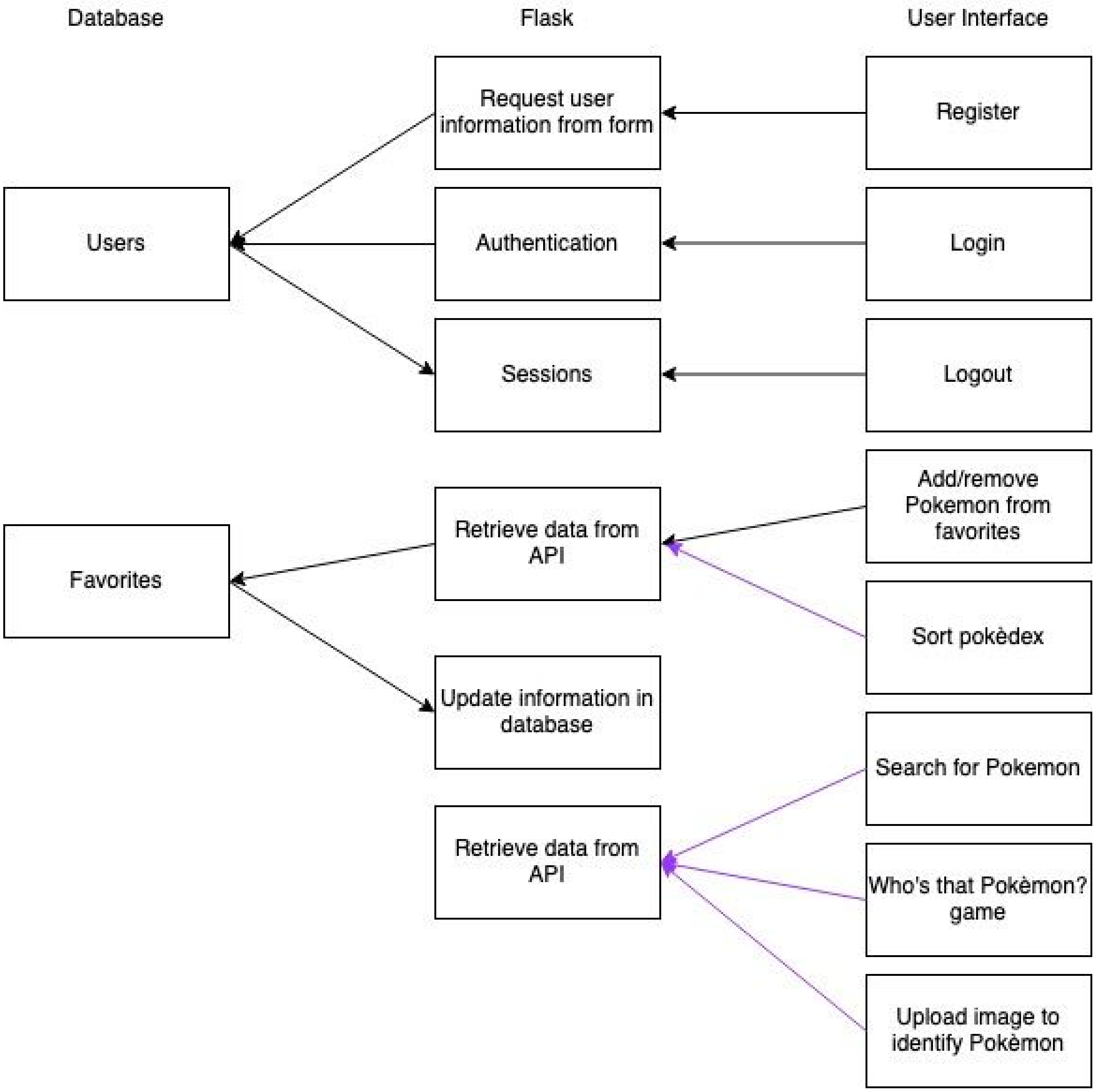
ROLES

- Leia Park: Project manager
- Alice Ni: Back end API/data
- Connor Oh: Front end bootstrap
- Hilary Zen: Python code

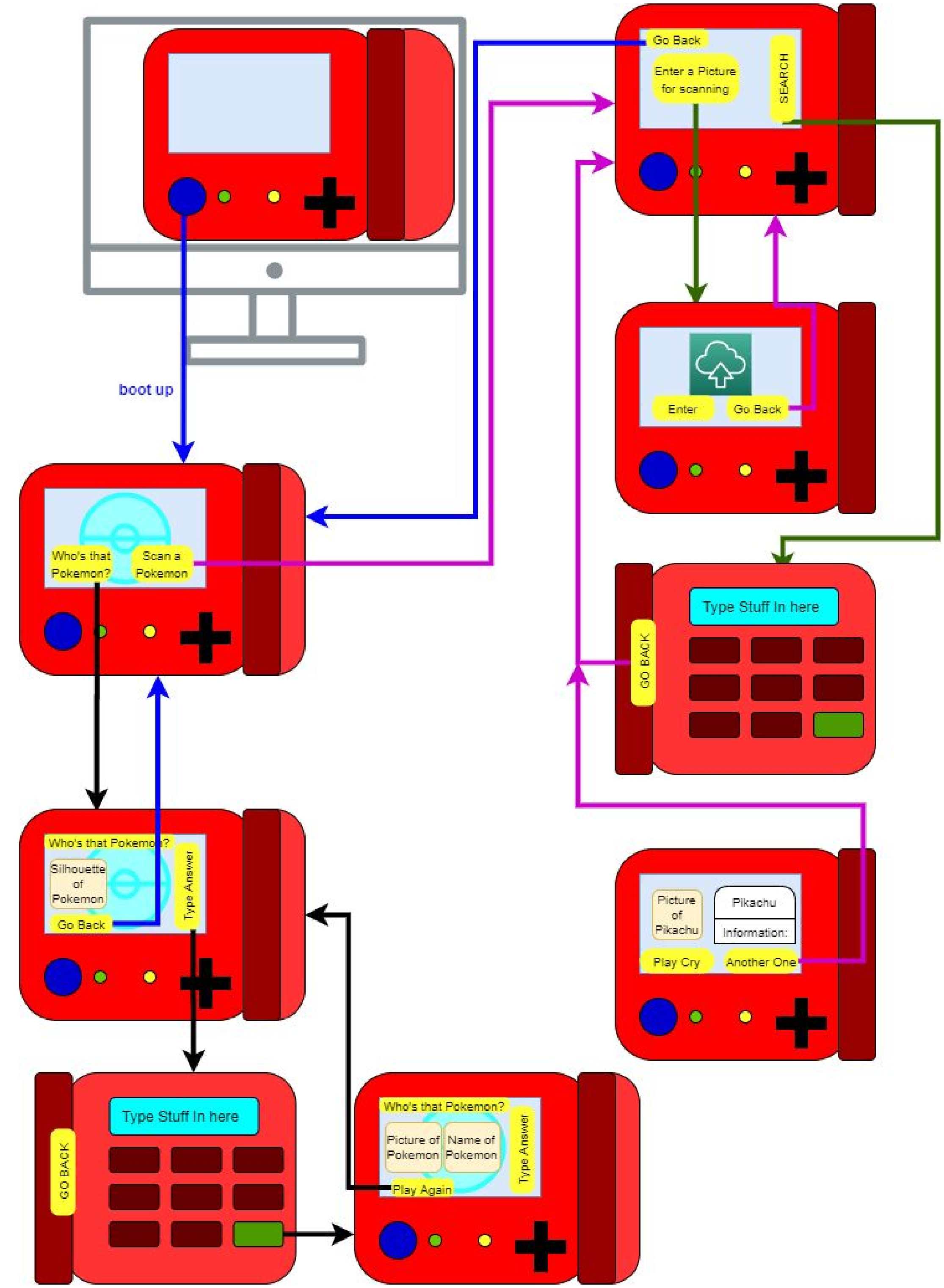
APIs

- PokeAPI
- Pokemon Go API (must make card for this)

COMPONENT MAP (it kinda sucks but will make amends along the way)



SITE MAP


boot up
Who's that Pokemon?
Scan a Pokemon
Go Back
Enter a Picture for scanning
SEARCH
Enter
Go Back
Type Stuff In here
GO BACK
Picture of Pikachu
Pikachu
Information:
Play Cry
Another One
Who's that Pokemon?
Silhouette of Pokemon
Type Answer
Go Back
Type Stuff In here
GO BACK
Who's that Pokemon?
Picture of Pokemon
Name of Pokemon
Type Answer
Play Again

DATABASES

Users

user_id	username	password
INTEGER	TEXT	TEXT
0	bob	123

Favorites

user_id	favorites
INTEGER	TEXT
0	32 89 19 208

*whenever a user adds/removes a pokemon from their favorites list, the corresponding pokemon number will be added/removed from the string of numbers in the favorites column

DATA

The data will be retrieved from the API according to the requests the user makes.

For example, when the user makes a search for a Pokemon, the form will send the user's search (assuming it is either a name or a pokemon ID) to an API call like so:

```
    Search: bulbasaur
    API call: https://pokeapi.co/api/v2/pokemon/bulbasaur
```

This call will return all of bulbasaur's statistics, which a python function will sort out and display to the user in a comprehensible way.

If a user would like to re-sort the default pokedex (which is in numerical order by ID), the selection of a new sort will also trigger a new API call. For example:

```
     Selected sort: by region
     Relevant API calls:
          https://pokeapi.co/api/v2/pokedex/2/  (kanto region)
          https://pokeapi.co/api/v2/pokedex/3/  (johto region)
          .. etc.
```

With each call, the pokemon from the regions will be gathered into a readable json file which we can then parse through using python functions to sort the pokemons differently.