A5 - Compose DB Pokemon

Assignment Submission

By: Connor Goodwin

W#: W0488245

Date: 2025-10-09

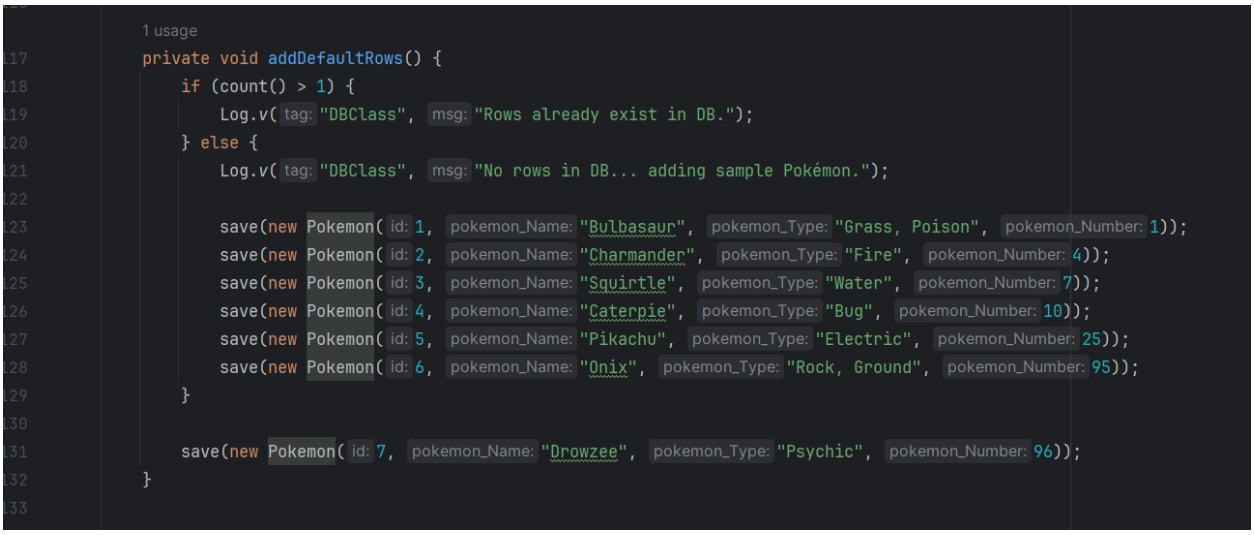
- **[20%] Change the DB to list Pokémon characters**. Determine what fields you wish to save for each character (at least these fields, **text** or otherwise, **name**, **number**, **power level**, **description**, **access_count** ...).
- Your App need not allow entry of a new record. Default loading of the DB with insert statements, or a pre-loaded DB is sufficient.

```
public class Pokemon {
    3 usages
    private String Pokemon_Name;
    3 usages
    private String Pokemon_Type;
    3 usages
    private Integer Pokemon_Number;
    // ...

    9 usages
    public Pokemon(long id, String pokemon_Name, String pokemon_Type, Inte
        this.id = id;
        Pokemon_Name = pokemon_Name;
        Pokemon_Type = pokemon_Type;
        Pokemon_Number = pokemon_Number;

    }

    @Override
    public String toString() {
        return  "ID: " + getId() + "\n"
                + "Pokemon Name: "  + getPokemon_Name() + "\n"
                + "Pokemon Type: " + getPokemon_Type() + "\n"
                + "Pokemon Number: " + getPokemon_Number() + "\n"
                ;

    }
```

- **[20%] Your compose main screen shows Favorite Pokémon and then the list of Pokémon**. When one Pokémon of the list is selected, it displays details of that Pokémon, and then increments the access count of that Pokémon.

  -Did not do

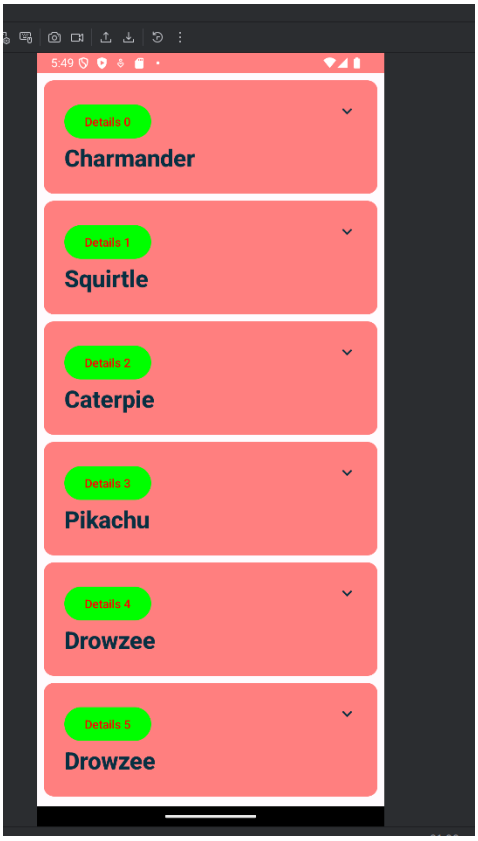
- **[10%] Create at least 6 items in the list**, but there are ways to load the DB in your PC, then upload that DB to the App to easily have all Pokémon….no need for many insert statements.



```
1 usage
private void addDefaultRows() {
    if (count() > 1) {
        Log.v( tag: "DBClass",  msg: "Rows already exist in DB.");
    } else {
        Log.v( tag: "DBClass",  msg: "No rows in DB... adding sample Pokémon.");

        save(new Pokemon( id: 1,  pokemon_Name: "Bulbasaur",  pokemon_Type: "Grass, Poison",  pokemon_Number: 1));
        save(new Pokemon( id: 2,  pokemon_Name: "Charmander",  pokemon_Type: "Fire",  pokemon_Number: 4));
        save(new Pokemon( id: 3,  pokemon_Name: "Squirtle",  pokemon_Type: "Water",  pokemon_Number: 7));
        save(new Pokemon( id: 4,  pokemon_Name: "Caterpie",  pokemon_Type: "Bug",  pokemon_Number: 10));
        save(new Pokemon( id: 5,  pokemon_Name: "Pikachu",  pokemon_Type: "Electric",  pokemon_Number: 25));
        save(new Pokemon( id: 6,  pokemon_Name: "Onix",  pokemon_Type: "Rock, Ground",  pokemon_Number: 95));
    }

    save(new Pokemon( id: 7,  pokemon_Name: "Drowzee",  pokemon_Type: "Psychic",  pokemon_Number: 96));
}
```

- **[10%] Each access of a record increments an "access_count" field of that Pokémon** .This is how we determine the favorite Pokémon. So going back to the main screen shows the favorite. Also, restarting the App shows the favorite. (note that the fav is the max(access_count) of all the Pokémon)
    - **Did not do**

- **[10%]** Show running of your application on 3 layouts (say; TV, Tablet, Smartphone Portrait, and Smartphone landscape)



- **[30%]** … max 30%, 5% each change, max 6 changes marked] Make any visible change to the compose screen (color, font, …)

  A. Bold to thin

```
254                fontWeight = FontWeight.Thin
```

  B. HeadlineMedium to HeadlineLarge

```
253            style = MaterialTheme.typography.headlineLarge.copy(
```

  C. Container Color Green to Blue

```
236                containerColor = Color.Blue,
```

D. ContentColor Red to Black

```
237                contentColor = Color.Black
```

E. Padding 12 to 20

```
232                .padding( all = 20.dp)
```

F. Stiffness Low to High

```
225                stiffness = Spring.StiffnessHigh
```