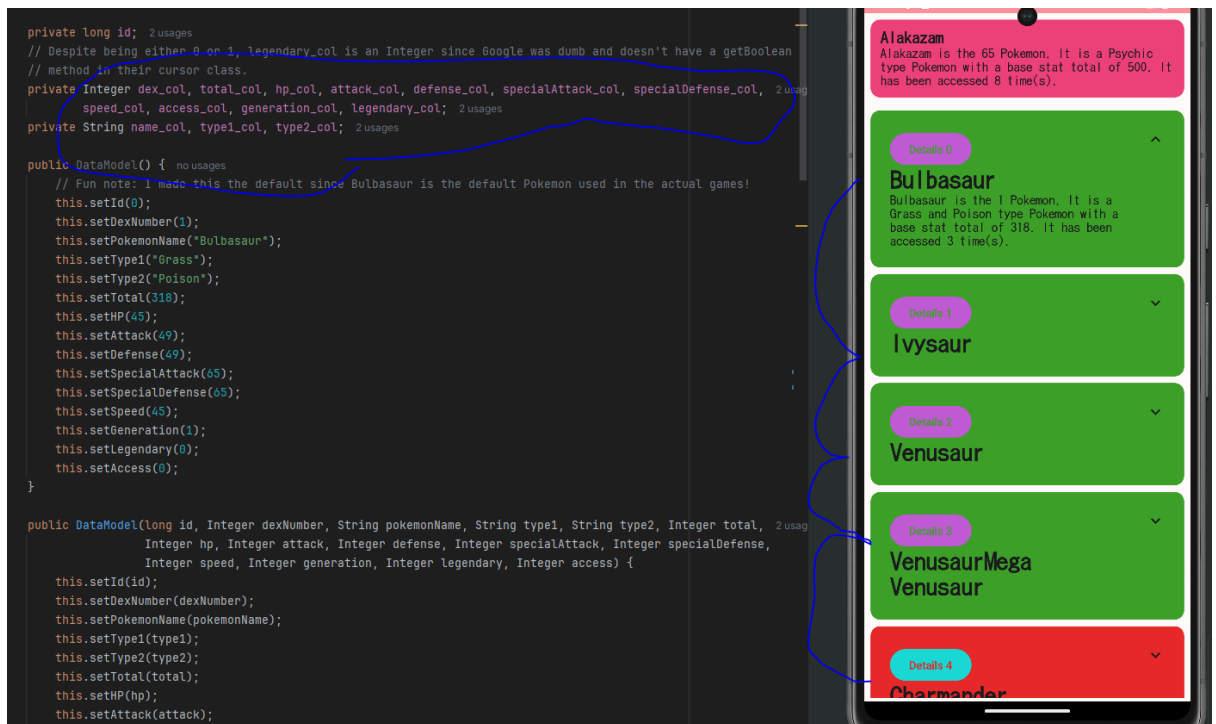


Github: (<https://github.com/JermNet/MOBI>)

- **[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.

```

private void addDefaultRows() { // usage
    bufferedReader = new BufferedReader(new InputStreamReader(inputStream));
    int loop = 0;
    try {
        InputStream inputStream = context.getAssets().open(fileName: "pokemon.csv");
        bufferedReader = new BufferedReader(new InputStreamReader(inputStream));
        String line;
        // This is to skip the first line, which contains the names for the cols in the CSV
        bufferedReader.readLine();

        while ((line = bufferedReader.readLine()) != null) {
            String[] pokeData = line.split(regex: ",");
            int id = loop;
            loop++;

            int dexNumber = Integer.parseInt(pokeData[0]);
            String name = pokeData[1];
            String type1 = pokeData[2];

            // Instead of having an empty value for a second type if the Pokemon doesn't have it, "None" is used.
            // No big reason for it, I just prefer it this way
            String type2 = !pokeData[3].isEmpty() ? pokeData[3] : "None";
            int total = Integer.parseInt(pokeData[4]);
            int hp = Integer.parseInt(pokeData[5]);
            int attack = Integer.parseInt(pokeData[6]);
            int defense = Integer.parseInt(pokeData[7]);
            int specialAttack = Integer.parseInt(pokeData[8]);
            int specialDefense = Integer.parseInt(pokeData[9]);
            int speed = Integer.parseInt(pokeData[10]);
            int generation = Integer.parseInt(pokeData[11]);

            // Legendary check because I can't use boolean because Google is dumb (no cursor.getBoolean method)
            int legendary;
            if (pokeData[12].equals("True")) {
                legendary = 1;
            } else {
                legendary = 0;
            }
            int access = 0;

            DataModel dataModel = new DataModel(id, dexNumber, name, type1, type2, total, hp, attack,
                defense, specialAttack, specialDefense, speed, generation, legendary, access);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

-
- **[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.



I did the drawing with my right hand... (I'm left handed)

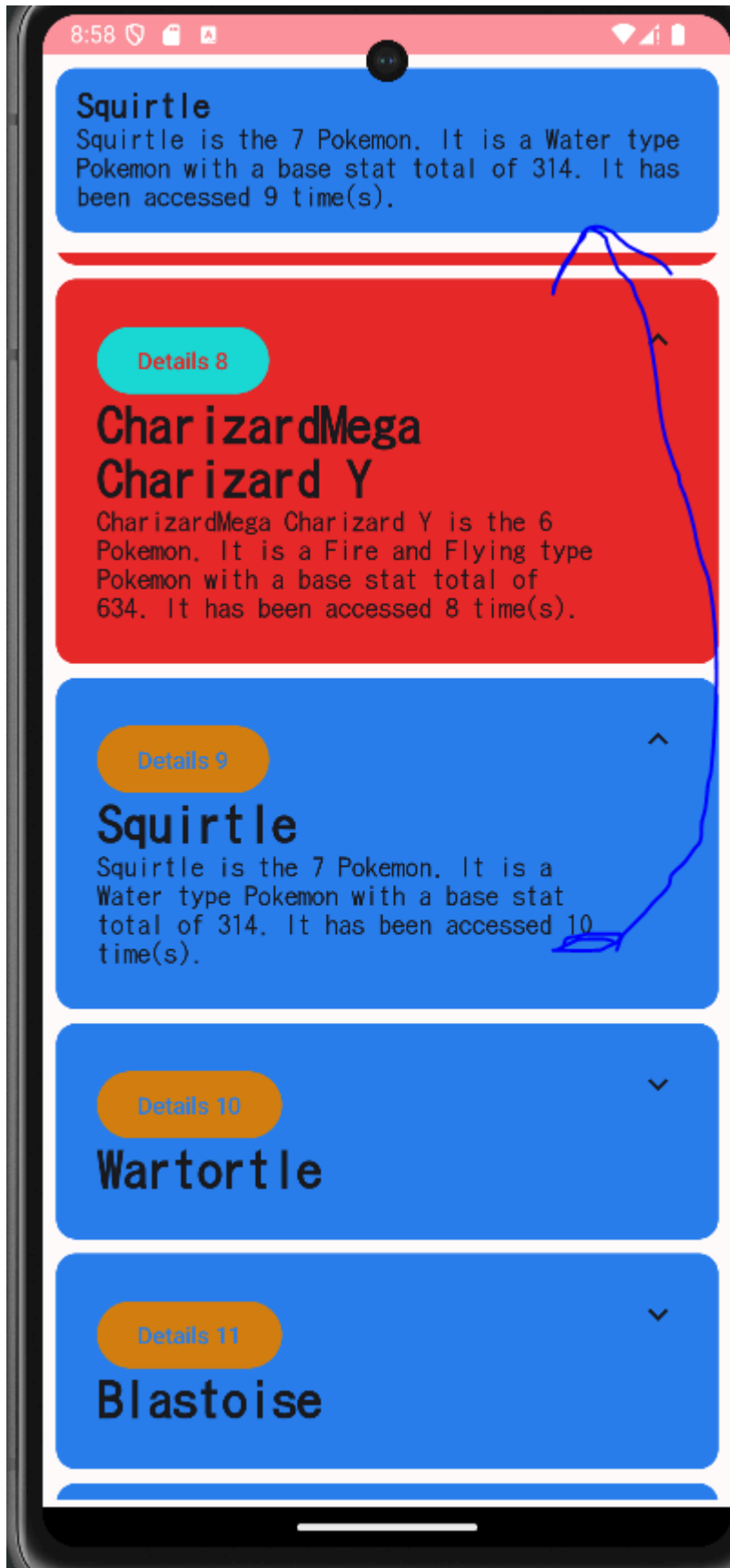
- **[10%]** Create at least 6 items in the list, but there are ways to load the DB in your PC, then upload that DB to the App to easily have all Pokémon....no need for many insert statements.

```

22 public class DBClass extends SQLiteOpenHelper implements DB_Interface {
152 private void addDefaultRows() {
153     try {
154         InputStream inputStream = context.getAssets().open("pokemon.csv");
155         BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));
156         String line;
157         // This is to skip the first line, which contains the names for the cols in the CSV
158         bufferedReader.readLine();
159
160         while ((line = bufferedReader.readLine()) != null) {
161             String[] pokeData = line.split(",");
162             int id = loop;
163             loop++;
164
165             int dexNumber = Integer.parseInt(pokeData[0]);
166             String name = pokeData[1];
167             String type1 = pokeData[2];
168
169             // Instead of having an empty value for a second type if the Pokemon doesn't have it, "None"
170             // No big reason for it, I just prefer it this way
171             String type2 = !pokeData[3].isEmpty() ? pokeData[3] : "None";
172             int total = Integer.parseInt(pokeData[4]);
173             int hp = Integer.parseInt(pokeData[5]);
174             int attack = Integer.parseInt(pokeData[6]);
175             int defense = Integer.parseInt(pokeData[7]);
176             int specialAttack = Integer.parseInt(pokeData[8]);
177             int specialDefense = Integer.parseInt(pokeData[9]);
178             int speed = Integer.parseInt(pokeData[10]);
179             int generation = Integer.parseInt(pokeData[11]);
180
181             // Legendary check because I can't use boolean because Google is dumb (no cursor.getBoolean)
182             int legendary;
183             if (pokeData[12].equals("True")) {
184                 legendary = 1;
185             } else {
186                 legendary = 0;
187             }
188
189             int access = 0;
190
191             DataModel dataModel = new DataModel(id, dexNumber, name, type1, type2, total, hp, attack,

```

- **[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)



Squirtle

Squirtle is the 7 Pokemon. It is a Water type Pokemon with a base stat total of 314. It has been accessed 9 time(s).

Details 8

CharizardMega Charizard Y

CharizardMega Charizard Y is the 6 Pokemon. It is a Fire and Flying type Pokemon with a base stat total of 634. It has been accessed 8 time(s).

Details 9

Squirtle

Squirtle is the 7 Pokemon. It is a Water type Pokemon with a base stat total of 314. It has been accessed 10 time(s).

Details 10

Wartortle

Details 11

Blastoise

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

9:01

Squirtle

Squirtle is the 7 Pokemon. It is a Water type Pokemon with a base stat total of 314. It has been accessed 9 time(s).

Details 8

CharizardMega
Charizard Y

Details 9

Squirtle

Details 10

Wartortle

Details 11

Blastoise

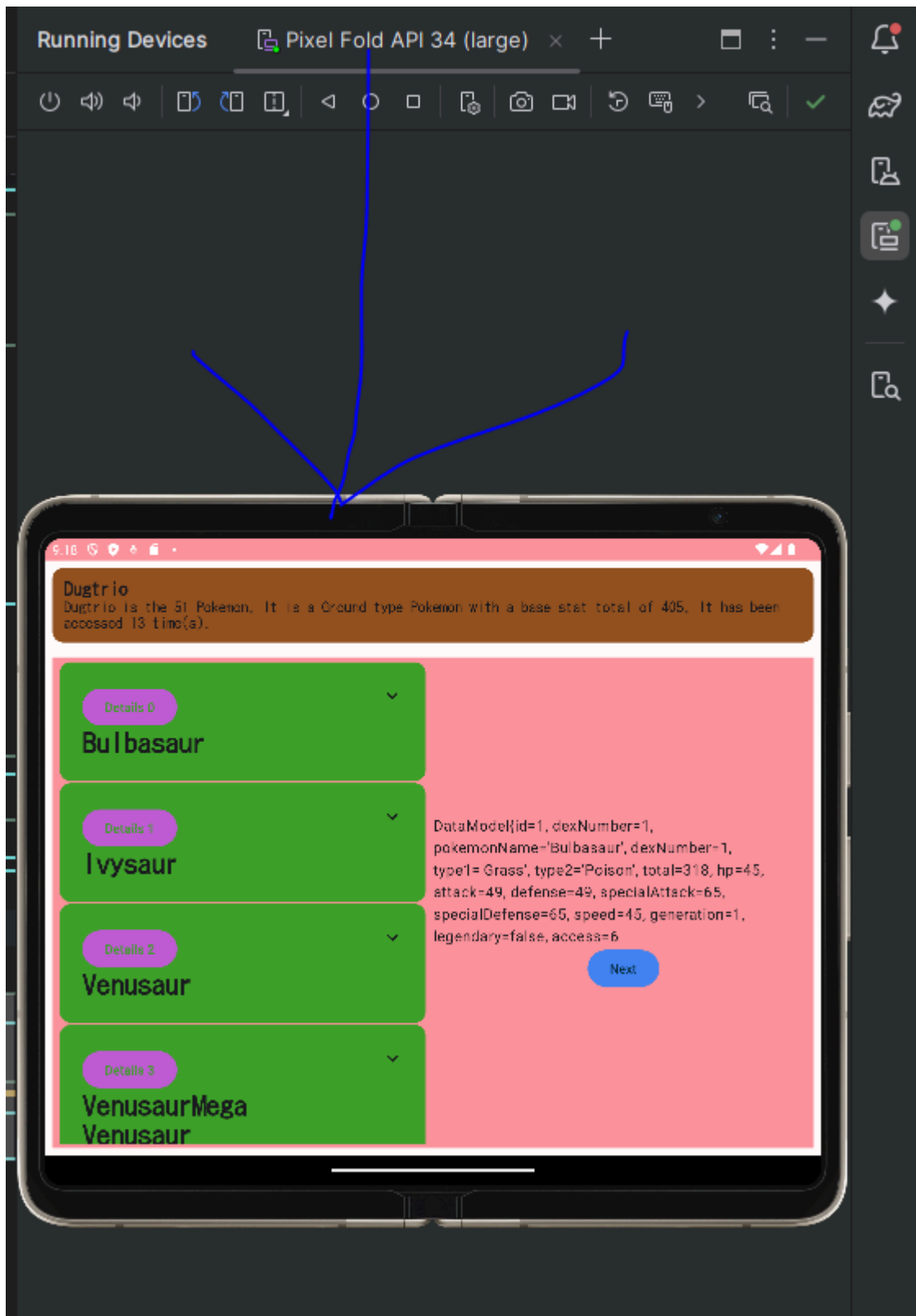
Details 12

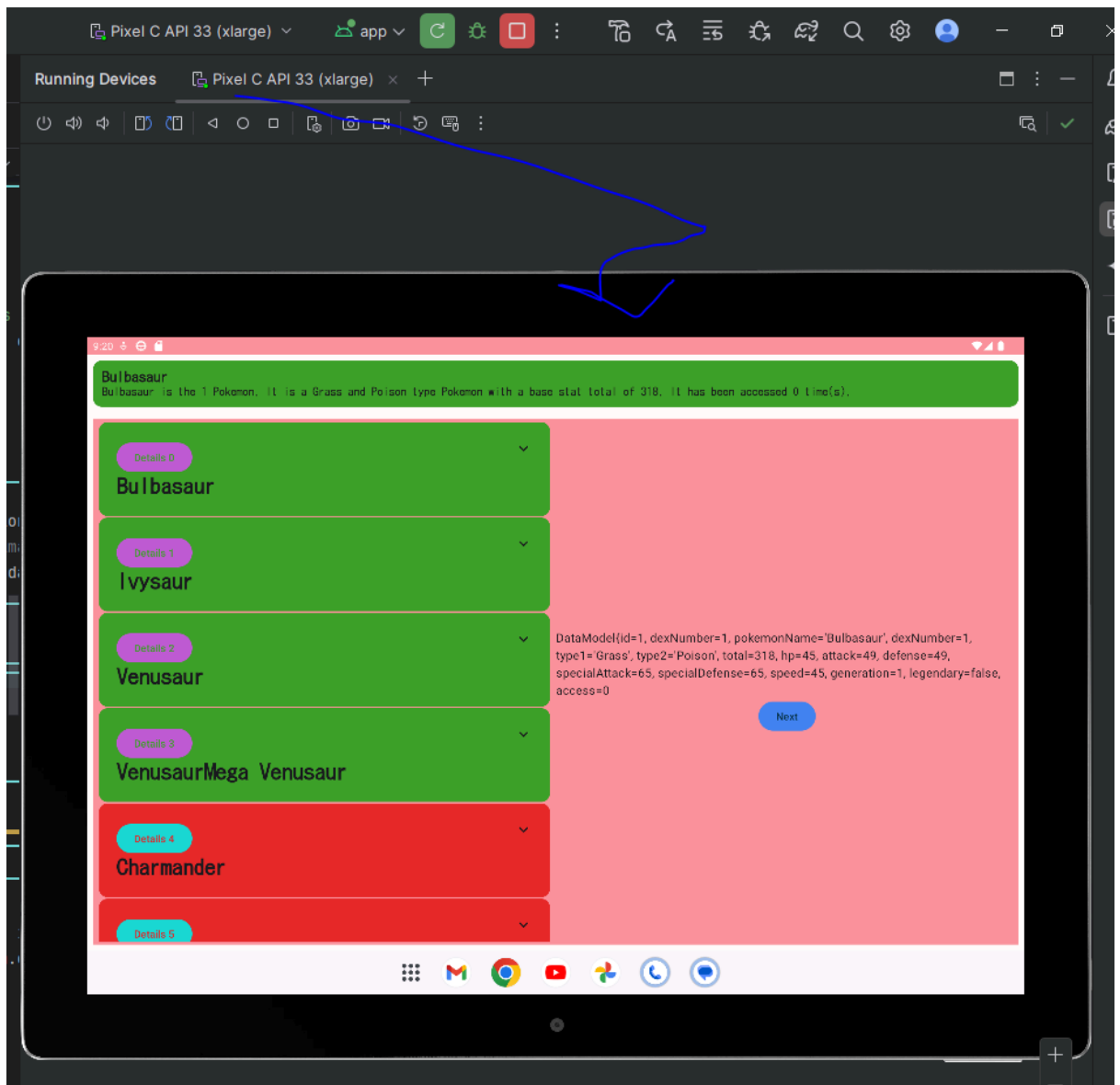
BlastoiseMega
Blastoise



1:1







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



-
- 1: Button/Pokemon Card colour
- 2: Button says "Pokemon" instead of "Details"

- 3: Favourite Pokemon (I had to work my brain to have it stay on the widescreen layout, but disappear on the small details page, so it counts!)
- 4: Background/Master,Next,Previous button colour
- 5. Padding
- 6. Font