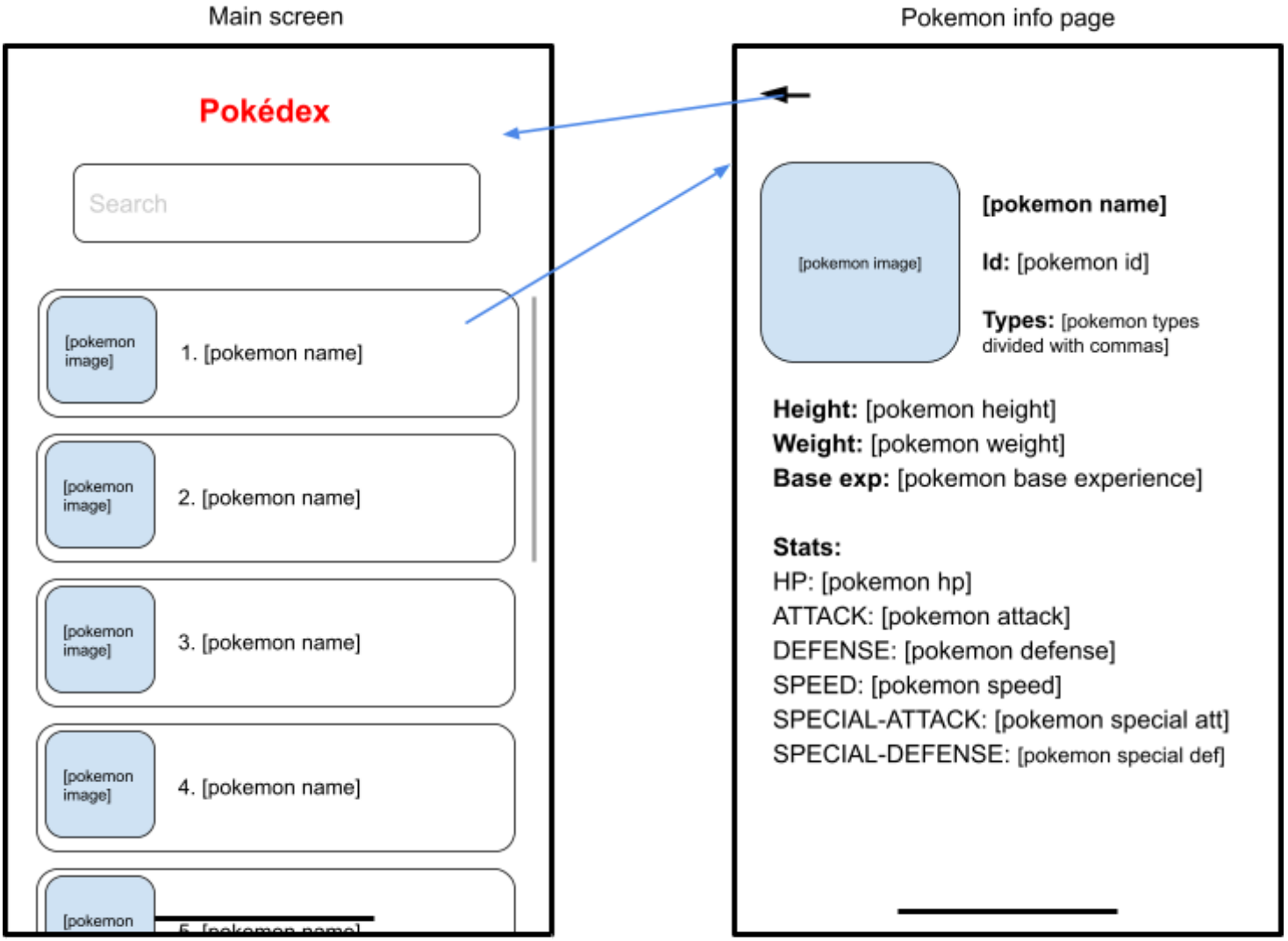
Redvike React Native task

Using the provided template, create a simple React Native application, which will display data from [Poke API v2](). The general layout of the app should look as follows:



Description and requirements:

- The blue arrows indicate the direction of navigation between screens.
- All **[ ]** should be filled with appropriate data and blue squares should be actual images of the Pokemons.
- All Pokemons in the list should be pressable and should take the user to the chosen Pokemon info page.
- For simplicity, **the list should include only Pokemons from 1 to 151** (first generation pokemons ie. from *Bulbasaur* to *Mew*). The images for those are shipped with the template repository you've received (in `src/assets` folder), as there is no endpoint to fetch the list of image urls separately.
- The *Search* input should allow the user to search the pokemons in the list by their name.
- Using TypeScript is mandatory, do not forget about correct typing.
- Navigation should be handled by React Navigation library, which has been pre-configured in the provided project. The structure of the navigation should be implemented and typings should be configured.
- Requests should be made through Redux, and pokemon data should be stored there
- Request errors should be handled
- You can use any library you want (that includes the component libraries), but bear in mind that the choice of the technologies used will also influence the final evaluation.

Link to PokeAPI just in case: https://pokeapi.co/docs/v2

Evaluation Expectations:

This task can be completed in 1–2 hours, but we encourage you to approach it as if you were building the foundation of a scalable application.
Think beyond the minimum requirements: show us how you'd structure a production-ready codebase, even for a simple app.

Specifically, we're looking for:
– Well-structured architecture (e.g., separation of concerns, modular code, reusable components).
– Thoughtful file organization.
– Good development practices (e.g., error boundaries, state normalization, etc.).
– Code readability and maintainability.
– Scalability in mind — how easy would it be to extend this app with more generations, features, or functionality?

Make it clean. Make it scalable. Make it yours.