PhinCon
Phintraco Consulting

Project Assignment

Sofware Engineer – Mobile or Web Platform

Requirements:

1. Build a **Mobile or Web** app using Technology Stack (see notes below) you Preferred, where it should have 3 pages, **Pokemon List**, **Pokemon Detail**, and **My Pokemon List**.
2. You can use this API http://pokeapi.co as your data source.
3. **Pokemon List page;** should show a list of Pokemons' names and picture. When a Pokemon is clicked, it should go to that Pokemon Detail page.
4. **Pokemon Detail page;** should show a picture of the Pokemon with its moves and types (this information is from the API, feel free to add more information of the Pokemon if you want to). The mandatory thing is that there should be a button to catch the Pokemon **(success probability is 50%, *see poin 6*)**, if success then user can **give the Pokemon a nickname** and add that Pokemon to 'My Pokemon List'
5. **My Pokemon List page;** should show a list (like Pokemon List page, but with each of their nicknames as well) of all Pokemons you have caught. It should also be possible to **release and rename (*see poin 6*)** a Pokemon from the list on this page.
6. There should be 3 REST API delivered as a backend service:
    - REST API to return probability is 50% when catching Pokemon.
    - REST API to release pokemon. This API should return a prime number, if the number returned by the API is not a prime number, then release will fail and vice versa.
    - REST API to rename pokemon. This function should return a combination of first name assigned combined with Fibonacci sequence, e.g.:
        - First name assigned is "Mighty Pikachu", first time renamed should be: "Mighty Pikachu-0"
        - Second time renamed should be: "Mighty Pikachu-1"
        - Third time renamed should be: "Mighty Pikachu-1"
        - Fourth time renamed should be: "Mighty Pikachu-2", and so on.
7. Any additional implementation microservice with containerization is an added value

Notes for Web:

1. *Your web app UI/UX should be Mobile-first or Single Page Application*
2. *Don't worry too much about getting the perfect design (UI/UX wise), as long as its usable and not too bad, it's okay*
3. *Consider using a global state management library*

Technology Stack for Web:

1. Frontend: (a) React JS with Redux or Hooks (b) Angular (c) any Frontend technology stack you preferred
2. Backend: (a) Node.js as JavaScript runtime with hapijs or Express.js as backend (b) Java Spring Bot

Notes for Mobile:

1. *If Mobile app is developed with concept of **Modular Framework** will be an added Value*
2. *Don't worry too much about getting the perfect design (UI/UX wise), as long as its usable and not too bad, it's okay*
3. *Consider using a global state management library*

Technology Stack for Mobile Native:

1. Frontend: Java/Kotlin Android or Swift iOS
2. Backend: Any tech stack is OK as long as it can perform actions/functions required by the Mobile App