Test Plan (3/26)

Team TreeHouse
Project: Whos-That-Pokemon
Abhinav Selvaraj, Breanna Chi, Prabhleen Bagri, Wen Yao (Ken) Ho

---

Testing Plans (SinglePlayer)

*Please refer to expert_user documentation to set up the project locally. This is a required step for testing.

Backend:
Required hardware: A windows PC or Mac
Required Software: Postman
Recommended Software: Visual Studio Code
Password/Key(s) required: N/A

We will be using Postman to verify that our backend is set up correctly so that our api may make calls to our database and request information based on specific parameters. The steps below describe how to test our backend.

1. Download Postman: https://www.postman.com/downloads/
2. Set up the project by referring to our expert user documentation (we recommend using VScode).
3. Navigate to this directory in the project directory: **\whos-that-pokemon\dev\Who-s-That-Pokemon\src\main\java\com\treehouse\Who\s\That\Pokemon>**
    a. This is our backend. Here, you should find 5 .java files. Open Application.java and run it.
4. Once in Postman, create a new collection and name it pokemon-api.
5. Next, create a new get request for the api call we want to test, and name the get request appropriately.
6. We will then type the request we want to get. Since the backend is hosted on local host port 8080, the URLS should look like : **localhost:8080/api/v1/pokemon.** This will return all the Pokemon in our database.
7. Our requests may vary from getting a random pokemon to getting a pokemon's type only, etc. Here is a list of all possible requests that will be tested with localhost:8080/api/v1/pokemon as the root url :
    a. /random
    b. /{id}
    c. /region/{id}
    d. /types/{id}

These requests retrieve a random Pokemon, retrieves a Pokemon's pokedex entry by their id, retrieves a Pokemon's region of origin by their id, and retrieves a Pokemon's types by their id's, respectively.

The expected time to complete the tests can be seen in the breakdown below

- Download Postman : Reliant on Individual Internet Connection
- Download any IDE: Reliant on Individual Internet Connection
- Setting up Project following expert documentation: ~5 min
- Launching Backend: < 1 min
- Creating Postman collection: < 1 min
- Each request takes around 569 milliseconds to complete.

Total Backend Testing Time: < 10 minutes

Frontend:
Required hardware: A windows PC or Mac
Required Software: Jest, Babel, Selenium
Password/Key(s) required: N/A

Jest will be used to test the rendering of our React components. We will also use it to test the snapshots of our react components, making sure they're rendered correctly.

1. Installing Jest: https://jestjs.io/docs/getting-started
2. Installing Selenium: https://www.browserstack.com/guide/selenium-react-tutorial
3. Set up the project by referring to our expert user documentation (we recommend using VScode).
4. Navigate to this directory in the project directory:
   \whos-that-pokemon\dev\PokemonClient\game-v1
   a. This is our frontend. If it's your first time cloning this project, you will have to install the necessary packages via: `npm install, npm install use-sound, npm install howler`
   b. In the same directory, do `npm start` to start the front end. Make sure that the backend is running too since we will be generating the pokemon from the backend.
5. We will be testing mainly three things:
   a. Snapshots/useEffect states of our component - when a component is rendered via useEffect hook, does it render correctly in the virtual DOM.
   b. aSync calls. Since we have asynchronous functions in our project, we want to ensure certain operations are called before another. We will be testing the promises and callback function.
   c. Order of execution. In the single player mode, the match ends when the user reaches the nth question where n is the total number of questions. We will use Jest to verify the order of function calls. We will be using both Selenium and Jest for this part of the testing. Selenium provides a more robust testing environment where it can automate some of the tests for us, which speeds up the test plan.

The expected time to complete the tests can be seen in the breakdown below

- Download Jest : Reliant on Individual Internet Connection
- Download Selenium: Reliant on Individual Internet Connection
- Download any IDE (preferably VS Code): Reliant on Individual Internet Connection

- Setting up Project following expert documentation: ~5min
- Launching Backend: < 1 min
- Launching Frontend: < 1 min
- Running Tests: < 10 mins

Total Frontend Testing Time: < 20 minutes