

Generations

Made by Ben

Fontys S3 2023

Contents

[Software / Frameworks 3](#_Toc145665706)

[PokeAPI 3](#_Toc145665707)

[Angular 3](#_Toc145665708)

[React 3](#_Toc145665709)

[Flask 3](#_Toc145665710)

[Ruby on rails 3](#_Toc145665711)

[Requirements 3](#_Toc145665712)

[Front end 4](#_Toc145665713)

[Back end 5](#_Toc145665714)

[Use-Cases 6](#_Toc145665715)

[Use-Case 6](#_Toc145665716)

[Use-Case Description 6](#_Toc145665717)

[Testplan 7](#_Toc145665718)

[Testmatrix 7](#_Toc145665719)

[Conceptual model 8](#_Toc145665720)

## Software / Frameworks

### PokeAPI

Free and easy to use API for getting the data I need; I will be using this to inflate my tables with data and to show all the Pokémon’s which you can select from.

### Angular

Angular is a fast TypeScript based framework, the thing I like about angular is that its fast and it works with modules. I find it easy to structure my projects with module-based frameworks, this is because you can split every single functionality, which also translates to single responsibility.

### React

A fast JSX based framework which is widely used in modern days, big platforms like Netflix use this framework because it’s easy to learn has a great community and is overall greatly documented.

### Flask

A lightweight python web framework which big companies like Netflix and Samsung use, flask is nice because it’s a microframework and it’s very easy to get started with. I will be challenging myself this semester by learning this framework because it seems interesting.

### Ruby on rails

One of the fastest web backend frameworks out there, also a new framework for me to learn.

## Requirements

QR-GEN01: The application should log / return errors whenever something goes wrong.

QR-GEN02: Project should apply the 3 layered structure, UI/UX, Business logic, Data Access.

### Front end

**MUST**

FR03: User needs to be able to create competitive Pokémon teams.

03.1 PVP Teams should be categorized by format.

03.2 PVE Teams should be categorized by generation.

FR04: User will be able to export Pokémon teams.

04.1 The export format should be in Pokémon showdown format so it can be imported on those websites as well.

04.2

**SHOULD**

FR05: User will be able to import Pokémon teams.

05.1 The user will be able to import a team but needs to assign it to category him/herself.

**COULD**

FR01: User needs to be able to log in to the application.

01.1 User can only login with their email and password.

01.2 Login should contain a password reset function.

FR02: User should be able to sign up.

02.1 The details needed for signing up should be Username, Email, Password.

02.2 User must input password and email twice for verification.

**WOULDN’T**

### Back end

**MUST**

FR01: Backend gets all the Pokémon’s from API called PokeAPI.

01.1

QR-01.1 Each Pokémon fetched from API should be turned into an object.

FR02: Backend will update database when user sends a post request.

02.1

FR03:

03.1

**SHOULD**

**COULD**

**WOULDN’T**

## Use-Cases

### Use-Case

### Use-Case Description

|  |  |
| --- | --- |
| **Name** UC01: | |
| Summary |  |
| Actors |  |
| Assumptions |  |
| Scenario |  |
| Exceptions |  |
| Result |  |

## Testplan

## Testmatrix

## Conceptual model

A diagram of a diagram

Description automatically generated with medium confidence