Architecture Design

This document will show and explain my architecture designs and choices.

Contents

[Front-End 2](#_Toc156569522)

[Framework 2](#_Toc156569523)

[Testing 2](#_Toc156569524)

[Back-End 3](#_Toc156569525)

[Programming Language 3](#_Toc156569526)

[Database 3](#_Toc156569527)

[ORM 3](#_Toc156569528)

[Software Quality 4](#_Toc156569529)

[Version Control 4](#_Toc156569530)

[CI/CD 4](#_Toc156569531)

[Static Code Analysis 4](#_Toc156569532)

[Miscellaneous 5](#_Toc156569533)

[PokeAPI 5](#_Toc156569534)

[Auth0 5](#_Toc156569535)

# Front-End

## Framework

To develop a front-end it is easier to use a framework like React, Angular or Vue. I decided to use Vue.js to develop my front-end in this application. I researched the front-end frameworks and I decided based of the result of that research what framework I would use. This research can be found in either my Github Documentation Repository or in the portfolio at LO1.

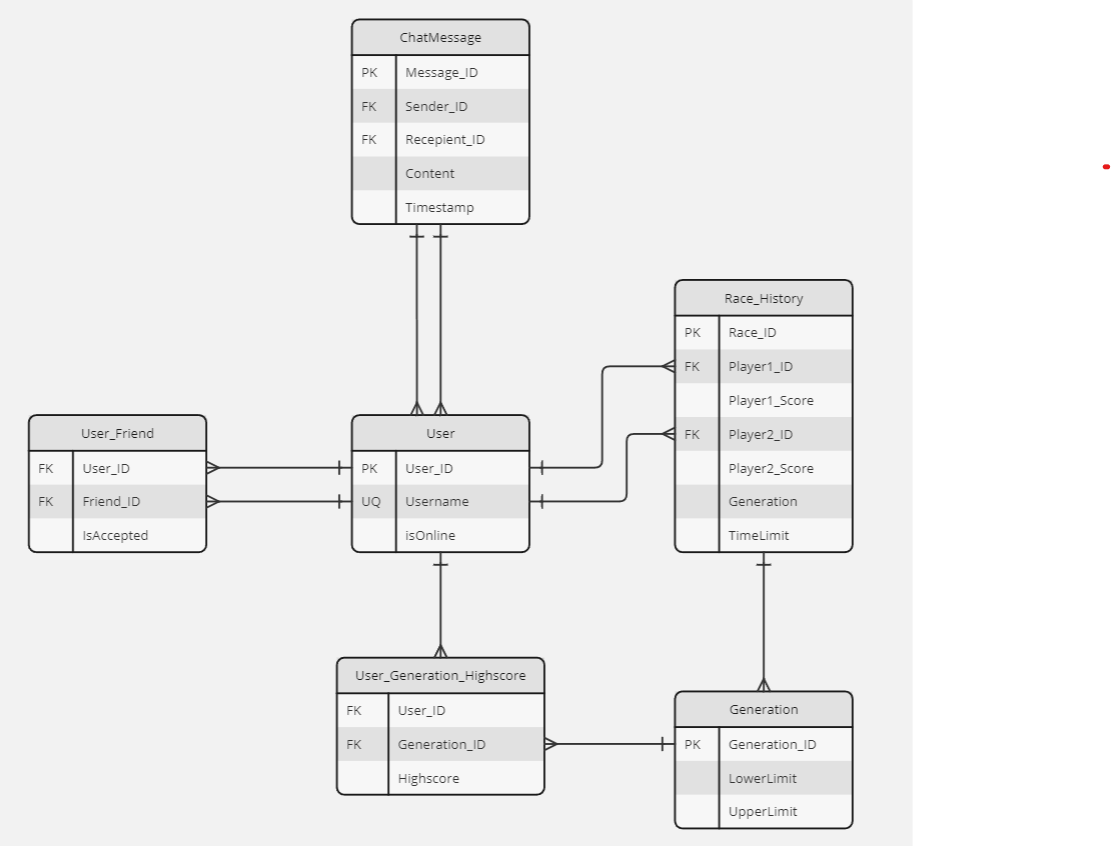
## Testing

For the testing of my front-end I’m using Cypress to create and run End to End tests.

# Back-End

## Programming Language

The programming language I’m going to be using Java with the Springboot framework. The reason I chose Java is because in the first year of my study I learned C# and Dotnet, while being a great language, it became very stale very quick. I want to give myself more of a challenge by learning a new language to code in.



## Database

For my Database I've decided to use MySQL, which we were going to use for our group project anyway so I might as well use it for my own so if I have any questions I could easily ask my groupmates for help because they know more about mySQL than I do.

## ORM

When using a database it’s pretty good to use an ORM. I chose to use JPA and Hibernate. This is because they are basically integrated in the Spring Framework, so they were the first choice.

# Software Quality

## Version Control

For version control I have decided to use Github. The reasoning behind this is because I’m pretty familiar with Github and how to use it. Even though Fontys does offer their own Gitlab servers, I’ve found them to be not as nice to work with as Github.

## CI/CD

For my CI/CD I’ve decided to use Github Actions, since I was using github already, Github Actions was a no-brainer option to choose for managing and running my CI/CD pipeline.

## Static Code Analysis

To check the software quality I am using Sonarqube. Sonarqube checks the quality of your code including bugs, duplications, security risks and codesmells. With the help of JaCoCo (Java Code Coverage), sonarqube is able to show me the codecoverage of my project to see how much of my project is covered by tests. What JaCoCo does is it runs the tests and then generates a code coverage report which Sonarqube uses to show me, because Sonarqube does not run the tests itself.

# Miscellaneous

## PokeAPI

Because my application requires the usage of all Pokémon. I have decided that instead of storing all the pokemon data inside of my own database, that I will use an external API called the PokeAPI. The PokeAPI allows me to gather information about all pokémon. From name to id to where to catch them. This way I know I always have enough data to use in my project.

## Auth0

Instead of making a custom login system I have decided to use Auth0. Auth0 handles the login for me, where I only have to manage the Authentication and Authorization. Using Auth0 is also a safer than making my own login system because the login data (Username, password, email, etc) won’t be saved in my own database, but an external one.