Design Document

Zhaklin Yanakieva

Student number: 3811468

Date: 27/10/2020

Diagram

Description automatically generated

Explanation:

On the diagram above, there are three different roles for users of the website. One for the guest-user, who can only see the home page, the other – aggregated user, who is the actually logged-in user who has most of the functions, and the admin, who can do the functions for the website support, such as uploading videos, deleting users, etc.

I chose to use MySql for storing the data because of its high availability and quick-start capability, which means that there are features self-management capabilities like auto restart, space expansion and automatic configuration changes for ease of management. It also comes with a comprehensive set of migration tools and a fully loaded graphical management suite. Furthermore, H2 as a mock database, which will be useful for the creation of the unit tests.

Diagram

Description automatically generated

Explanation:

The diagram above displays the connection between the back and front end through a restful API.

A picture containing diagram

Description automatically generated

Explanation:

Diagram

Description automatically generatedUML diagram ( C4)

Diagram

Description automatically generated

Justification for the front-end framework of choice:

According to the research I made, React.js is easier to be learned for beginner developers. One of the main concerns developers have is choosing a framework (or library) that is not confusing and can be implemented in a way the learner can understand it. React is easy to grasp for developers who are familiar with Javascript. However, even if I am not that good in Javascript, React can be the right place to start my learning process. Unlike Angular, React holds a smooth learning curve.

In React, your application comprises of components. Ideally, it is started by building small components like buttons, checkboxes, dropdowns, menus, etc. and create wrapper components around these smaller components. And as going on writing the higher level wrapper components, a single root component and several hierarchical components are created. Now, here’s a no brainer: each component in React has its own logic, so the component may be re-used.

I tried to make test projects using the three frameworks: React.js; Vue.js and Angular.js, and after experiencing the work with them, I realized that it would be more convenient for my project to use React.js. Not only is it more understandable for a beginner with this frameworks, but the error that occur while implementing a code, happen to be found more often on the internet. Also, I chose React.js because the versions are updated automatically while the ones of Angular are done manually which will waste more time.

Justification for the back-end framework of choice:

* Autoconfiguration:

1. Developers can automatically configure their Spring application and also the framework gives the chance of changing the configuration based on the dependencies the user lists instead of them. For example, when there is “MySQL” listed as a dependency, it will configure your Spring application with the “MySQL connector” included. Yet, if the user wants to add a custom configuration, the user can create a class that overrides the default configuration for your “MySQL connector”.

* Standalone:

1. There’s no need to deploy your application to a web server. You simply enter the run command to start the application.

* Opinionated:

1. On the official page, we find that Spring Boot decides for you which defaults to use for the configuration. Also, it decides which packages to install for the dependencies a user requires and this setup helps developers to get started quickly on their projects.

* Better documentation:

1. The how-to pages of the spring boot framework are better explained.
2. There is more information about the errors that may occur during the process of learning spring boot.

A set of metrics include the following methods:

* algorithms analysis;
* number of code lines;
* the complexity of a software;
* functional points analysis;
* number of bugs per 1000 code lines;
* level of testing;
* number of classes and interfaces;

Bibliography:

1. Nitin Pandit, https://www.c-sharpcorner.com/article/what-and-why-reactjs/
2. Michiel Mulders, https://stackify.com/what-is-spring-boot/