|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | December 16th  Individual Track |
| Design Document | | | |
| By Janne van Seggelen  S3-CB03 | | | |

Contents

[Introduction 3](#_Toc120177797)

[C4 diagram 4](#_Toc120177798)

[How am I going to guarantee SOLID? 5](#_Toc120177799)

[What design choices did I make? 5](#_Toc120177800)

[CI Diagram 5](#_Toc120177801)

[API 6](#_Toc120177802)

[Description of the entities 6](#_Toc120177803)

# Introduction

In this document I am going to explain my project and how I am going to guarantee SOLID, what design choices I made, and the working and build of my application.

# C4 diagram

Diagram

Description automatically generated

Diagram

Description automatically generated

# How am I going to guarantee SOLID?

I will use Dependency Injection for the Database layer. You can view the C$ Diagram for further explanation.

# What design choices did I make?

I have chosen Bootstrap for the frontend design, because it makes developing a responsive website easier. No further design requirements where given. Next that, I have a lot of experience with Bootstrap already. Bootstrap is a powerful, extensible, and feature-packed frontend toolkit. You can build and customize with Sass, utilize prebuilt grid system and components, and bring projects to life with powerful JavaScript plugins.

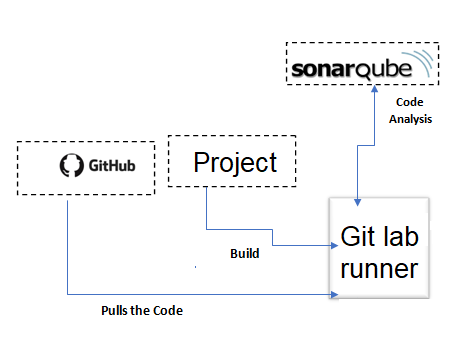
* <https://getbootstrap.com/>

I have chosen React because it was recommended at school.

*“React is a*[*free and open-source*](https://en.wikipedia.org/wiki/Free_and_open-source_software)[*front-end*](https://en.wikipedia.org/wiki/Front_end_and_back_end)[*JavaScript library*](https://en.wikipedia.org/wiki/JavaScript_library)*for building*[*user interfaces*](https://en.wikipedia.org/wiki/User_interfaces)*based on UI*[*components*](https://en.wikipedia.org/wiki/Component-based_software_engineering)*. It is maintained by*[*Meta*](https://en.wikipedia.org/wiki/Meta_Platforms)*(formerly Facebook) and a community of individual developers and companies. React can be used as a base in the development of*[*single-page*](https://en.wikipedia.org/wiki/Single-page_application)*, mobile, or server-rendered applications with frameworks like*[*Next.js*](https://en.wikipedia.org/wiki/Next.js)*. However, React is only concerned with state management and rendering that state to the*[*DOM*](https://en.wikipedia.org/wiki/Document_Object_Model)*, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.”*

* <https://en.wikipedia.org/wiki/React_(JavaScript_library)>

# CI Diagram



# API

|  |  |  |  |
| --- | --- | --- | --- |
| URL | Resource | Operation |  |
| /bundles | Bundles | GET | Gets a list of bundles |
| /bundles/12/packages | Bundles | GET | Gets a list of packages from a specific bundles |
| /bundles/12/reviews | Reviews | GET | Gets a list of review from a specific bundles |
| /bundles?price=5.00 | Bundles and Packages | GET | Gets a list of bundles with the price from 5.00 or higher |
| /bundles/2 | Bundles | DELETE | Deletes the bundle with the id 2 |
| /bundles/12/packages/2 | Packages | DELETE | Delete the package with id 2 |
| /bundles | Bundles | POST | Adds a new bundle |
| /bundles/12/packages | Packages | POST | Adds a new bundle |
| /bundle/12 | Bundle | PUT | Updates the bundle with the id 12 |
| /bundle/12/packages/2 | Packages | PUT | Update the package with the id 2 |
| /accounts | Accounts | GET | Gets a list of accounts |
| /accounts?role=teacher | Accounts | GET | Gets a list of accounts that are teachers |
| /teachers | Teachers | GET | Gets a list of teachers |
| /teachers?name=Josh | Teachers | GET | Gets a list of teachers with the name Josh |
| /teacher/12 | Teacher | GET | Gets a specific teacher |

## Description of the entities

1. Teachers  
   The teachers have the following instances: Name, Age, Gender, and Education.
2. Bundles  
   The bundles have the following instances: Name, Description, Price, and a list of packages.
3. Packages  
   The packages have the following instances: Name, Description, and Price.
4. Accounts  
   The accounts have the following instances: Username, Password, Email, Phone number, and a Role.  
   The role are Teacher, Admin, and Customer.