|  |
| --- |
| Petar Dakov  S3-CB04    2023/2024 |

Crenit | The Netherlands

architecture choices

Architecture document

2023

Contents

[1. Design/Architecture 1](#_Toc150251167)

[2. Database 2](#_Toc150251168)

[3. Frontend 3](#_Toc150251169)

[4. Backend 4](#_Toc150251170)

[5. C4 Diagram 4](#_Toc150251171)

[5.1 C1 – Context 4](#_Toc150251172)

[5.2 C2 – Context 6](#_Toc150251173)

[5.3 C3 – API 7](#_Toc150251174)

# Design/Architecture

Using the Clean Architecture for my project can offer several benefits. The goal of this software architecture is to minimize the human resources required to build and maintain the required system.

Some of the advantages are:

* Testable
* Maintainable
* Changeable
* Easy to Develop
* Easy to Deploy
* Independent

This architecture makes it easier to maintain and extend my application over time, as changes in one layer have minimal impact on other layers. It also encourages writing unit tests for the core business logic (use cases or user stories) without needing to rely on the external dependencies like databases or user interfaces. This makes the codebase more testable. The clean architecture also aims to isolate the core business logic from the details of external frameworks and technologies. This allows me to upgrade or switch out frameworks, libraries, or databases with minimal disruption to the core functionality. It also makes it easier to adapt the application to changing requirements. Adding new features or modifying existing ones without unnecessary complexity.

# Database

For this assignment I’ll be using MySQL as my database. Entity framework will be also included in the project.

1 of the tables was for the users. I wanted to know what the pros and cons are for having 2 or 3 different tables for them or just 1. The users can be administrators or just normal users in my case, so I decided to go with only 1 table, because the data between the 2 different types is the same. I need to include just 1 more column for “user type”. It will simplify my database scheme and reduce the complexity. This will allow me to control access and permissions through my application. However, if I had multiple types of users, it would have been better to have multiple tables, because the speed of the query execution will be much faster.

In conclusion, the difference between have separate tables for users and administrators or just one depends on the complexity of my application, the differences between those users, and my specific requirements.

A screenshot of a computer

Description automatically generated

# Frontend

The framework that I’ll use for the implementation of the frontend for the project will be React.js. These are some of the reasons why:

At first React has a large and active community of developers. It’s easy to find resources, libraries, and solutions to common problems. It’s also frequently updated and improved by Facebook. React can also be easily integrated with other libraries or frameworks, giving you flexibility in your technology stack. It’s widely used in the industry, so learning it will increase your job opportunities as a frontend developer.

Most used frameworks:

A graph with numbers and a number of people

Description automatically generated with medium confidence (Vailshery, 2023)

A screenshot of a computer

Description automatically generated

# Backend

The framework that I’ll use for the implementation of the backend will be .NET framework. Some of the reasons why are:

* A unified story for building web UI and web API’s
* Architected for testability.
* Open source
* Ability to target any application type running on any platform.

One of the standout features of .NET Core is its cross-platform support. Your application can run on Windows, macOS, and various Linux distributions. This framework is also known for its performance improvements over previous versions. ASP.NET Core encourages and supports unit testing and integration testing of your application components, helping to ensure the reliability and maintainability of your code.

A screenshot of a computer

Description automatically generated

# C4 Diagram

## C1 – Context

A diagram of a puzzle

Description automatically generated

## C2 – Context

A diagram of a system

Description automatically generated

## C3 – API

A diagram of a company

Description automatically generated