|  |
| --- |
| Brett Mulder |
| Portfolio Semester 3 |
| In dit document zit alles in betrekking tot mijn leerdoelen.  How to Create a Portfolio |

# Inhoud

[Inhoud 1](#_Toc116631639)

[Leerdoelen uitgelegd 3](#_Toc116631640)

[1. Full-stack Web Application 3](#_Toc116631641)

[2. Software Quality 3](#_Toc116631642)

[3. Agile methods 3](#_Toc116631643)

[4. CI/CD 3](#_Toc116631644)

[6. Requirements en Design 3](#_Toc116631645)

[7. Business Processes 3](#_Toc116631646)

[8. Professional 3](#_Toc116631647)

[Full-stack Web Application 4](#_Toc116631648)

[Design 4](#_Toc116631649)

[4](#_Toc116631650)

[API 5](#_Toc116631651)

[Software Quality 6](#_Toc116631652)

[Agile Methods 7](#_Toc116631653)

[CI/CD 8](#_Toc116631654)

[Cultural differences and ethics 9](#_Toc116631655)

[Requirements en Design 10](#_Toc116631656)

[Business Processes 11](#_Toc116631657)

[Professional 12](#_Toc116631658)

# Leerdoelen uitgelegd

## Full-stack Web Application

Je designd en bouwt een **user-friendly, full-stack** web applicatie.

## Software Quality

Je maakt gebruik van **tooling en methodology** dat continu aantoont dat de code verbeterd wordt tijdens het development proces.

## Agile methods

Je kiest en implementeert de meest toepasselijke **Agile methods** voor je software project.

## CI/CD

Je designd en implementeert een (semi)automatische software release proces dat overeen komt met je project.

1. Cultural differences and ethics

Je herkent en houdt rekening met culturele verschillen tussen project stakeholders en ethische aspecten bij softwareontwikkeling.

## Requirements en Design

Je analyseert (geen functionele) requirements, designs en deze doet testen doormiddel van verschillende manieren van tests.

## Business Processes

Je analyseert en beschrijft een simpel business proces dat overeen komt met je project.

## Professional

Je gedraagt je op een professionele manier tijdens het maken van je programma en het leren.

# Full-stack Web Application

Ik heb gekozen voor VueJS als front-end framework, deze keuzes heb ik beschreven in [dit document](https://github.com/Fontys-Brett-Mulder/CardGames/blob/main/Documentation/CardGames.docx). Ik ben gebruik gaan maken van microservices ([voor uitleg lees dit document](https://github.com/Fontys-Brett-Mulder/CardGames/blob/main/Documentation/Microservice.docx)), hierbij heb ik verschillende API’s. Deze API’s zijn gemaakt doormiddel van ASP.net Core Web API.

## Design

Ik heb de keuze gemaakt om alleen een Design te maken voor mobiele versie. Dit heb ik gedaan omdat mijn doelgroep ligt bij degene die kaartspellen vergeten zijn en toch een spel willen spelen. Je hebt altijd een telefoon op zak dus vandaar alleen een mobiele versie.

## 

## API

Voor de GamesService API heb ik de volgende structuur aan gehouden voor het ophalen van Games. De API endpoint <https://localhost:7099/api/Games/getAllGames> geeft een lijst terug van games.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Een Game heeft bepaalde informatie, deze informatie is hieronder te zien.

Afbeelding met tekst

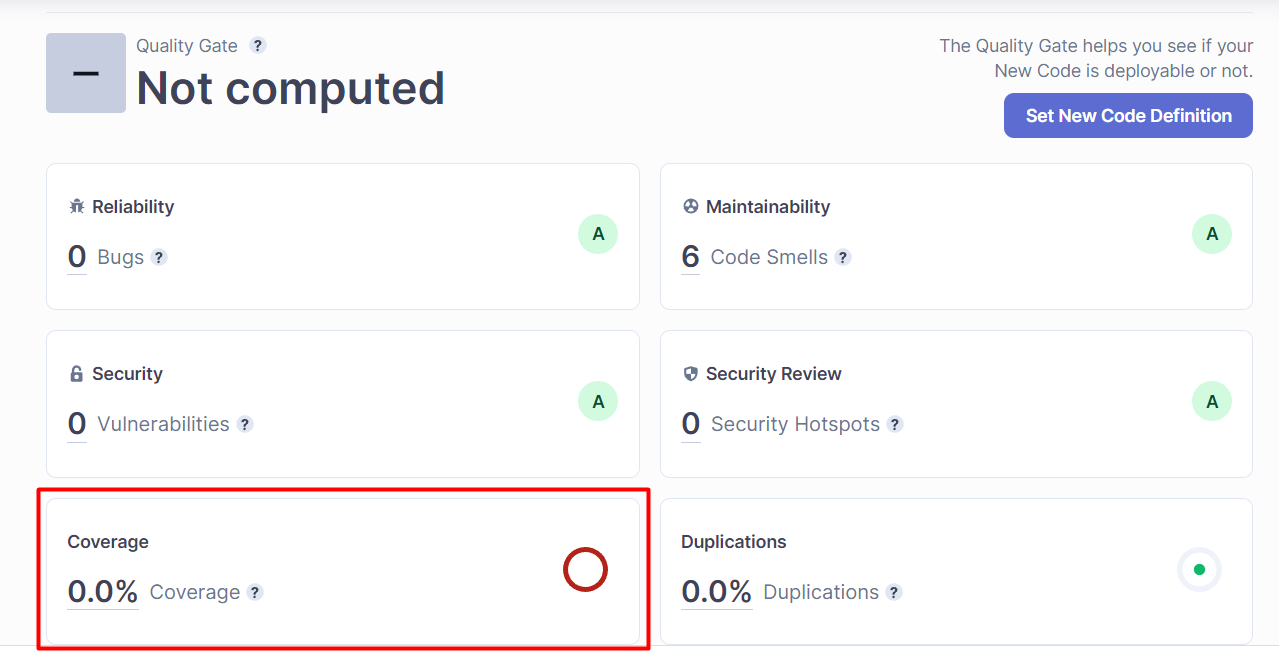
Automatisch gegenereerde beschrijving

# Software Quality

Voor het aantonen dat de code die geschreven wordt er goed uit ziet maak ik gebruik van [SonarCloud](https://www.sonarsource.com/products/sonarcloud/?gads_campaign=Europe-4-SonarCloud&gads_ad_group=SonarCloud&gads_keyword=sonarcloud). SonarCloud doet een scan over mijn code en kijkt naar punten waar het beter kan. Bijvoorbeeld als een class readonly mag zijn, of consistent gebruik van dezelfde hoofdletters bij functies, variables etc.

Bij elke push naar de “main” branch wordt er een check uitgevoerd. Deze uitslagen kan ik dan terugvinden in het [Dashboard van SonarCloud.](https://sonarcloud.io/summary/overall?id=Fontys-Brett-Mulder_GamesServiceAPI)

Hieronder zijn een paar uitslagen te zien, 1 van de 6 checks is rood gemarkeerd. Dit is omdat in dit geval er 0% van mijn code wordt getest. Wanneer er wel meer code wordt getest zal in verloop van tijd het percentage omhoog gaan.



Bij [maintainability](https://sonarcloud.io/project/issues?resolved=false&types=CODE_SMELL&id=Fontys-Brett-Mulder_GamesServiceAPI) wordt gekeken naar stukjes code die beter kunnen. Bijvoorbeeld: dit object moet readable zijn, of een object moet non-nullable zijn etc.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

# Agile Methods

## Scrum

**Projectmanagement** maak ik gebruik van **scrum**. Ik heb voor mezelf **user stories** gemaakt. Uit deze **user stories** kan ik **tickets** halen die op mijn **backlog** komen te staan. Per **sprint** zal ik kijken welke **user story** ik kan oppakken. Hierbij ge ik dan de **tickets** realiseren die bij de **user story** horen.

Bij de **sprint planning** worden alle **tickets** geperiodiseerd. Wanneer een **ticket** af is wordt deze naar done gesleept. Dat betekend dat een gedeelte van de **user story** klaar is. Wanneer alle **tickets** af zijn van een **user story** dan zal als het goed is de **acceptatiecriteria** van het **ticket** ook klaar zijn.

Na elke **sprint** heb je een **retrospective**. Dit betekend dat je gaat kijken, wat ging goed en wat ging minder goed, welke aandachtspunten zijn er? Etc. Normaal gesproken doe je dit met een groep en overleg je wat er beter kan etc. Ik heb daar de **feedpulse** voor. Hierin staat voor mij precies in wat ik gedaan heb en wat er beter kan etc.

# CI/CD

Ik heb voor mijn [GamesService API](https://github.com/Fontys-Brett-Mulder/GamesServiceAPI) een CI script gemaakt geschreven. Wanneer nieuwe code de “main” branch op wilt komen, doormiddel van een direct Push of doormiddel van een Pull Request, wordt de CI uitgevoerd. De code hiervan is te vinden in [dit bestand](https://github.com/Fontys-Brett-Mulder/GamesServiceAPI/blob/main/.github/workflows/dotnet.yml). In dat bestand doe ik eigenlijk een paar dingen die ik hieronder ga uitleggen.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

On → push → branches: “\*\*”. Wanneer ik dus naar welke branch dan ook doe pushen, wordt dit bestand aangeroepen. Voor pull requestst is het alleen op de “main” branch.

# Cultural differences and ethics

# Requirements en Design

## Requirements

Voor de requirements heb ik gebruik gemaakt van [deze Youtube video](https://www.youtube.com/watch?v=3fgfUHKITts&ab_channel=TECHIN5MINUTES). In deze video wordt precies uitgelegd hoe de functionele en niet-functionele requirements van een applicatie er uit moeten komen te zien. Deze heb ik toegepast in het “CardGames” document.

## Design

Voor het design van de applicatie heb ik een C4 model gemaakt. Dit C4 model staat beschreven in [dit document](https://github.com/Fontys-Brett-Mulder/CardGames/blob/main/Documentation/C4Model.md).

# Business Processes

# Professional

Tijdens de proftaak opdracht met de groep Mediaan-DMA (proftaak groep 3 klas S3DB04) toon ik dit vooral aan. Hierin ga ik professioneel op met mijn groepsgenoten. Hiervoor hebben we afspraken opgesteld waar we allemaal over hebben ingestemd. Deze staan [hier](https://github.com/S3-Proftaakgroep-3/Mediaan-DMA/wiki/Samenwerkingsovereenkomst-Mediaan-DMA) beschreven.

Voor het project ben ik de eerste sprint scrum master geweest. Dit betekend dat ik het bord van GitHub goed bij moest houden en de groepsgenoten hierin moest begeleiden zodat ze dit ook gaan snappen.

Qua communicatie met de stakeholders gaat het goed. We hebben nu 1 oplevering gehad. Deze oplevering verliep soepen en kregen complimenten over hoe goed de presentatie was.