Icon

Description automatically generatedShape

Description automatically generated with low confidence

INFIRFS

Emmanuel van Oudennaarden

Versie: 1.0

Opdrachtgever: tbd

|  |  |
| --- | --- |
| s1234567 | Floris Admiraal |
|  | Tweede auteur |

(Technisch)  
Ontwerp

Versie Beheer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versie | Versienaam | Datum | Auteur | Wijzigingen |
| 0.1 | Initiele versie | 31-08-2023 | F. Admiraal | Opzet document |
| 0.2 | Ontwikkelversie | 19-12-2024 | Emmanuel | Aanmaken hoofdstuk 1 en 2 |
| 0.3 | Ontwikkelversie | 31-08-2023 | C. Prins -Ipema | Aanmaken hoofdstuk 3 |
| 0.4 | Concept versie | 31-08-2023 | V. Spijkers | Wijzigingen doorvoeren in hoofdstuk 5 |
| 0.5 | Concept versie | 31-08-2023 | V. Spijkers | Wijzigingen doorvoeren in hoofdstuk 6 |
| 0.6 | Pre Final | 31-08-2023 | E. Verduin | Inconsistenties verbeterd |
| 1.0 | Final | 02-09-2023 | E. Verduin | Gecontroleerd op spelfouten |

# Management Samenvatting

Dit technisch ontwerp beschrijft de structuur en werking van een webshop ontwikkeld voor Toets 1 van de module First Full Stack. De applicatie biedt gebruikers een intuïtieve ervaring met functies zoals het bekijken van producten, beheren van een winkelmandje en plaatsen van bestellingen. De back-end is gebouwd met Java Spring Boot en maakt gebruik van PostgreSQL voor gegevensbeheer. De front-end gebruikt Angular en Bootstrap voor responsieve webervaringen. JWT-tokens worden toegepast voor veilige authenticatie.

# Inleiding

De webshop is ontworpen voor klanten die regelmatig online winkelen en een soepele, veilige ervaring zoeken. Het systeem maakt gebruik van een moderne layered architecture, waarbij de presentatie-, business-, en datalaag duidelijk gescheiden zijn. Deze structuur maakt de applicatie schaalbaar en onderhoudbaar.

Inhoudsopgave

[1 Management Samenvatting 3](#_Toc159947091)

[2 Inleiding 5](#_Toc159947092)

[3 Klassendiagrammen 5](#_Toc159947093)

[3.1 Frontend 5](#_Toc159947094)

[3.2 Backend 5](#_Toc159947095)

[4 Sequentiediagrammen 5](#_Toc159947096)

[4.1 Sequentiediagram UC1 5](#_Toc159947097)

[4.2 Sequentiediagram UC4 5](#_Toc159947098)

[4.3 Sequentiediagram UC12 5](#_Toc159947099)

[4.4 Sequentiediagram UC13 5](#_Toc159947100)

[4.5 Sequentiediagram UC15 5](#_Toc159947101)

[5 Database 5](#_Toc159947102)

[5.1 Databasemanagementsysteem 5](#_Toc159947103)

[5.2 ER Diagram 5](#_Toc159947104)

[6 REST 5](#_Toc159947105)

[6.1 Endpoints 5](#_Toc159947106)

[6.2 HTTP Statuscodes 5](#_Toc159947107)

[7 Security 6](#_Toc159947108)

[7.1 Rollensysteem 6](#_Toc159947109)

[7.2 Authenticatie 6](#_Toc159947110)

[7.2.1 JWT tokens 6](#_Toc159947111)

[8 Bronnen 6](#_Toc159947112)

# Inleiding

# Klassendiagrammen

U kunt de volledige diagram via deze link open voor een wat minder wazig overzicht.

<https://drive.google.com/file/d/18xI1PO4DdbiYvgUq9F-qHcCnxgrz148n/view?usp=drive_link>

Afbeelding met tekst, diagram, nummer, Parallel

Automatisch gegenereerde beschrijving

## Frontend

Hoog niveau diagram waar je overzicht geeft in welke Angular components je gemaakt hebt. We zoeken naar overzicht / inzicht in het model.

We vragen je om een klasse diagram te maken.  
Let op dat je niet apart de HTML of CSS files hoeft op te schrijven – deze graag uit het diagram houden voor beter overzicht. Dit hoort bij een component in Angular maar hoeft dus niet in het klasse diagram.

Probeer je in diagram / model de lagen te tonen van onze “layered architecture” (n-tier architecture) Dus : de view of presentatie laag bovenaan, controller / business laag daaronder en de data opslag / communicatie helemaal onderaan.

(mag natuurlijk ook van links naar recht i.p.v. boven naar beneden)

## Backend

Applicatie model – ook hier graag in de vorm van lagen indien mogelijk.

# Sequentiediagrammen

*Hier maak je per use case een sequentie diagram. We verwachten minimaal vijf sequentiediagrammen per student. Zet bij elk sequentiediagram de naam van de use case erbij.*

## Sequentiediagram UC1

## Sequentiediagram UC4

## Sequentiediagram UC12

## Sequentiediagram UC13

## Sequentiediagram UC15

# Database

## Databasemanagementsysteem

*Licht hier toe welk databasemanagementsysteem wordt gebruikt.  
In principe is dit voorgeschreven – MySQL database & PG Admin 4 als client*

## ER Diagram

*Voeg het ER Diagram in. Zorg dat deze leesbaar blijft en op een pagina past. Deze mag zowel in Crowfeet als Chen notatie.*

# REST

## Endpoints

*Voeg hier alle endpoints (URI’s) toe die worden gemaakt. Inclusief request type (POST/GET/DELETE/UPDATE/..) , verwachte responses ( in menselijke taal wat je als antwoord moet verwachten) en eventueel response types (JSON by default) .*

## HTTP Statuscodes

*Licht hier alle statuscodes toe die teruggegeven moeten worden per response, denk hier ook aan error handeling en wat voor responses je naar je gebruikers terugstuurt.*

# Security

## Rollensysteem

*Jullie applicatie heeft waarschijnlijk te maken met meerdere rollen zoals administrator, beheerder (moderator) of eindgebruiker. Een moderator mag waarschijnlijk niet het logo van de applicatie veranderen of een nieuwe administrator toevoegen terwijl een administrator dat wel mag. In deze paragraaf beschrijf je welke rollen er zijn en hoe het rollensysteem wordt geïmplementeerd zodat autorisatie op de juiste manier wordt ingevoerd.*

## Authenticatie

*Beschrijf de wijze en de mate van authenticatie en autorisatie die het beoogde systeem gebruikt. Verwijs hierbij eventueel naar een klassendiagram of een sequentiediagram.*

### JWT tokens

*Als je gebruik gaat maken van access tokens, refresh tokens e.a. beschrijf hier dan de wijze waarop ze gebruikt worden en de inhoud ervan (claims). Je voegt een diagram toe die dit proces verduidelijkt.*

# Bronnen

Als je goede bronnen op internet vind en gebruikt, dan lever je deze hier ook.

In principe probeer je de APA notatie te gebruiken voor bronnen.