

# Software Design Document (SDD)

Eindopdracht Gemeente Almere – Software Development Project



Luuk Steltenpool, Jesper Meuzelaar, Ryan Dürr

Projectleider: Hans Pieters

Gemeente Almere

7-5-2025, versie 1

## Versie's

Versie	Auteur	Datum	Veranderingen
1.1	Ryan	7-5-2025	Start van SDD
1.2	Luuk, Ryan	12-5-2025	Inleiding/Algemene beschrijving van de opdracht
1.2	Luuk	14-5-2025	Algemene planning
1.4	Jesper	19-5-2025	Gebruikerseisen en technische vereisten
1.5	Luuk	21-5-2025	Wireframes + toelichting gemaakt
1.6	Ryan	23-5-2025	Advies
1.7	Ryan	2-6-2025	Datamodel + toelichting
1.8	Luuk	4-6-2025	Algemene architectuur
1.9	Jesper	10-6-2026	Deployments en onderhoud
2.0	Luuk	10-6-2025	Implementatie details toegevoegd.
2.1	Jesper	14-6-2025	Testmethoden en resultaten.

## Inhoudsopgave

Versie's .....	2
1. Inleiding .....	4
1.1. Doel .....	4
1.2. Project definitie .....	4
1.3. Product Visie .....	4
1.4. Scope .....	4
1.5. Stakeholders .....	5
2. Gebruikerseisen .....	5
3. Architectuuroverzicht .....	6
3.1. Technische eisen .....	6
3.2. Algemene architectuur .....	7
3.3. Technologieën en frameworks .....	7
4. Gedetailleerd Ontwerp .....	8
4.1. Interface-ontwerp .....	8
4.2. Datamodel .....	11
4.3. Advies .....	12
4.4. Module-overzicht .....	12
5. Implementatie Details .....	13
5.1. Programmeertaal en coding standaarden .....	13
5.2. Beveiliging en authenticatie .....	13
6. Testing en Validatie .....	14
6.1. Teststrategie .....	14
7. Deployments en Onderhoud (optioneel) .....	15
7.1. Deployment strategie .....	15
7.2. Monitoring en onderhoud .....	15
7.3. Schaalbaarheid .....	15
8. Globale planning .....	16

# 1. Inleiding

## 1.1. Doel

**Beschrijf het doel van dit document en voor wie het bedoeld is.**

De Lief & Leed Pot ondersteunt medewerkers bij bijzondere gelegenheden zoals jubilea, ziekte of pensionering. Momenteel verloopt het proces handmatig, wat inefficiënt is. Dit project beoogt het proces te automatiseren, inclusief:

- Digitale aanvraagverwerking
- Automatische controle op basis van leeftijd en dienstjaren

Het doel van dit document is om dit project te documenteren zodat de opdrachtgever een duidelijk beeld heeft van het product.

## 1.2. Project definitie

Dit project is opgezet in samenwerking met de personeelsvereniging van Gemeente Almere om het proces rondom de Lief & Leed Potje te digitaliseren. Medewerkers kunnen maandelijks vrijwillig bijdragen aan deze pot, waarvan bij bijzondere gebeurtenissen een geldbedrag beschikbaar wordt gesteld. Dit project gaat ervoor zorgen dat de aanvraag van het geldbedrag automatisch verloopt, waardoor de administratieve lasten afnemen en efficiëntie toeneemt.

## 1.3. Product Visie

Hoe gaat het product er uit zien en wat zal de werking zijn? Waarom wordt de software ontwikkeld en welke waarde biedt de software als het af is?

## 1.4. Scope

**Beschrijf de grenzen van het project, de belangrijkste functies en de context waarin het systeem zal opereren.**

**In scope:**

- Digitaal aanvraagportaal
- Automatische controles (leeftijd, dienstjaren)
- Handmatige beoordeling bij ziekte
- Integratie van betaling via rekeningnummer.

**Buiten scope:**

- Wijzigingen in contributiestructuur
- Integratie met externe HR-systemen (m.u.v. geboortedata en dienstjaren)

## 1.5. Stakeholders

- Medewerkers Gemeente Almere
  - HR-afdeling
  - IT-afdeling
  - Financiële administratie
  - Mollie (betalingsverwerker)
- 

## 2. Gebruikerseisen

Hier wordt beschreven wat het systeem moet doen:

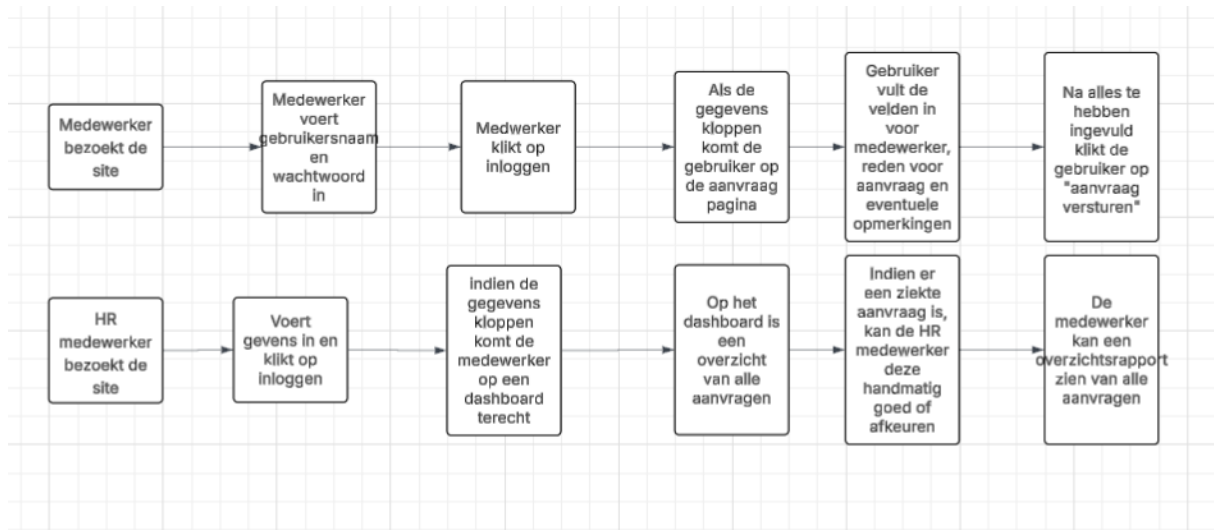
F1	Gebruikers moeten aanvragen kunnen indienen via een portal
F2	Een automatisch systeem wat automatisch controles voor leeftijd en dienst jubilea verwerkt
F3	Invoeren type gebeurtenis (verjaardag, ziekte, etc..)
F4	Mogelijkheid voor handmatige controle van aanvragen bijvoorbeeld: ziektegevallen
F6	Een rapportage module voor inzicht in en aanvragen
F7	Er moet een apart beheerder dashboard beschikbaar zijn voor HR
F8	Validatie op basis van geboortedata en dienstjaren
F9	Bij ziekte aanvragen worden ze niet automatisch goedgekeurd
F10	Mogelijkheid tot opmerkingen plaatsen bij de aanvraag

## 3. Architectuuroverzicht

### 3.1. Technische eisen

Nf1	Systeem moet beschikbaar zijn via een beveiligde web omgeving	Must	Privacy gevoelige gegevens
NF2	Alleen geautoriseerde medewerkers hebben toegang tot het HR-dashboard	Must	Voor avg compliance
NF3	Gebruikersinterface moet intuïtief zijn	Should	Bevordert Acceptatie
NF4	Systeem moet schaalbaar zijn voor toekomstig gebruik	could	Voor toekomstigheid
NF5	Gegevensopslag moet privacy correct zijn	Must	Het betreft privacygevoelige gegevens
NF6	Minimale laadtijd van paginas: < 2 seconden	Must	Bevordert acceptatie
NF7	Data moet worden opgeslagen in een MySQL database	Must	Maakt de data-opslag makkelijk schaalbaar

## 3.2. Algemene architectuur



## 3.3. Technologieën en frameworks

In dit project hebben wij gebruik gemaakt van het Framework Laravel. Laravel geeft ons een fijne basis om mee te werken en was erg handig voor ons aangezien we het log-in systeem goed konden gebruiken.

Programmeertalen die wij hebben gebruikt zijn als volgt:

- PHP
- HTML/Blade
- CSS Tailwind

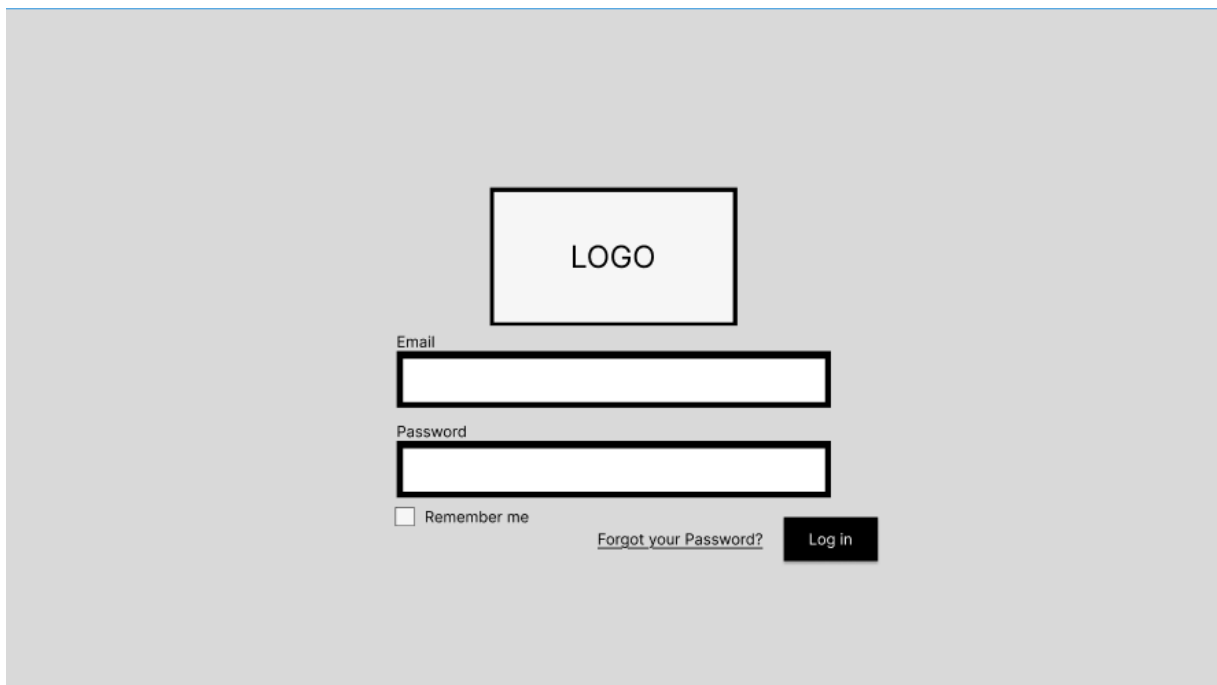
## 4. Gedetailleerd Ontwerp

### 4.1. Interface-ontwerp

In dit project hadden wij 3 nodige pagina's. Een pagina voor het inloggen, een pagina voor een medewerker om een aanvraag te doen. Een pagina voor de HR-medewerker om een aanvragen overzicht te hebben en aanvragen i.v.m ziekte goed of af te keuren.

Voor deze pagina's moesten wij als eerste wireframes opstellen.

**De inlog pagina:**

A wireframe of a login page on a light gray background. At the top center is a rectangular box labeled "LOGO". Below it is a text input field with the label "Email" to its left. Underneath the email field is another text input field with the label "Password" to its left. Below the password field is a checkbox with the text "Remember me" to its right. To the right of the checkbox is a link that says "Forgot your Password?". At the bottom right is a dark gray button with the text "Log in" in white.

Op de inlogpagina is in het midden van het scherm het logo te vinden. Hieronder staat een textvak waar de gebruiker zijn/haar email adres in voert.

Onder het textvak voor het emailadres staat nog een textvak, hier voert de gebruiker zijn/haar wachtwoord in.

Daaronder staat een aanvinkbaar vak, indien dit aangevinkt is onthouden het systeem uw gegevens de volgende keer dat u wilt inloggen.

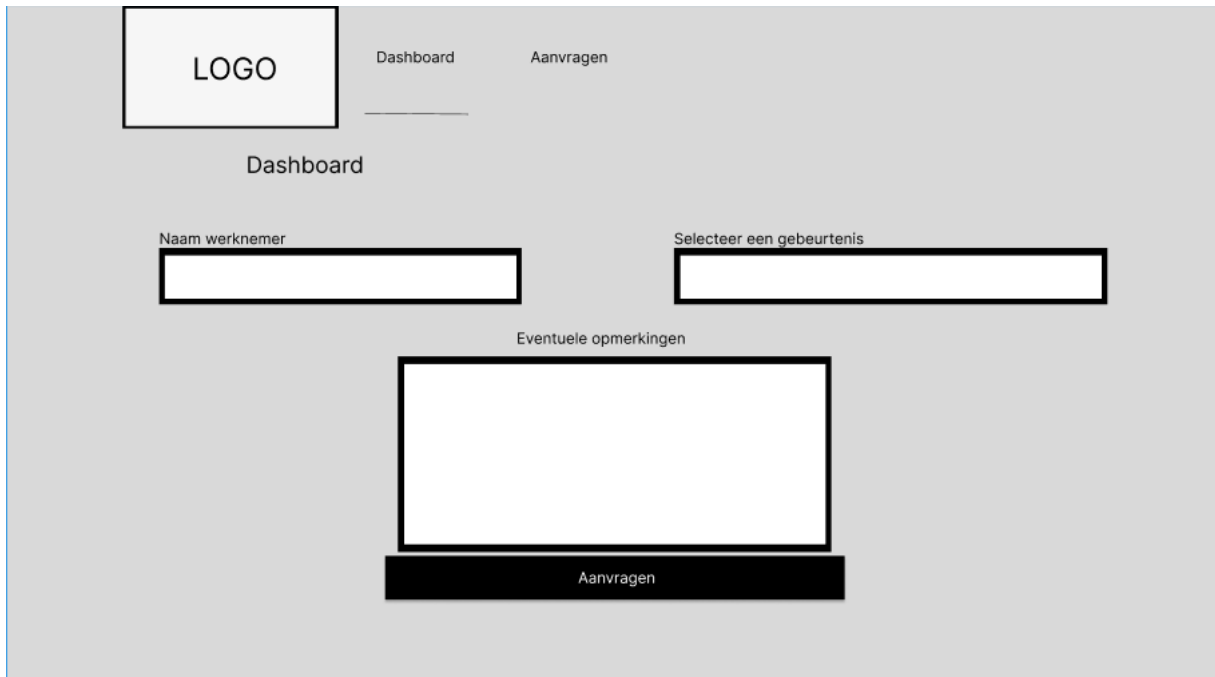
Verder is er een link met "Forgot your password?". Indien u hierop klikt wordt de gebruiker naar een pagina geleid via waar ze zijn/haar wachtwoord kunnen veranderen als ze dit vergeten zijn.

Als laatste is er een login knop. Als deze ingedrukt wordt controleert het systeem of de in het textvak ingevoerde gegevens overeenkomen met gegevens uit de



database. Zo ja, wordt de gebruiker doorgestuurd naar het desbetreffende portaal. Indien ze incorrect zijn krijgt de gebruiker een melding met dat de gegevens niet overeenkomen.

### Aanvragen dashboard:



Op het dashboard voor het doen van aanvragen zijn een aantal dingen te vinden.

Linksboven is het logo te vinden. Rechts daarvan 2 knoppen. Als de gebruiker een HR-medewerker is, zijn zowel de dashboard als de aanvragenoverzichts sectie beschikbaar.

Als u een normale medewerker bent heeft u alleen toegang tot het dashboard om aanvragen te doen.

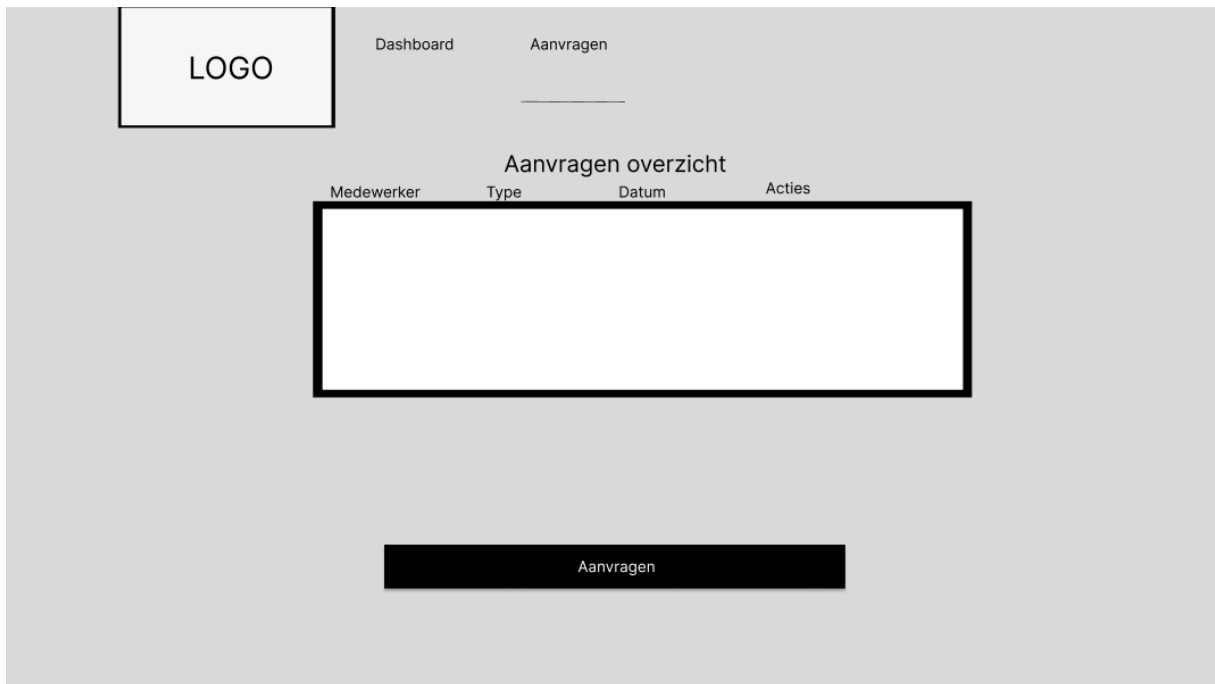
Hieronder vindt u een veld met "naam medewerker". Hier kan de gebruiker een naam intypen. Het systeem vindt dan automatisch namen die hier mee overeen komen, waardoor de gebruiker eenvoudig de gewenste collega kan vinden.

Na de juiste collega te hebben gevonden kan de gebruiker in het veld rechts hiervan een gebeurtenis selecteren. Dit is reden voor de aanvraag, bijvoorbeeld: 40 jaar in dienst bij de gemeente, 12,5 jaar huwelijk.

Hier zitten vaste geldprijzen aan verbonden.

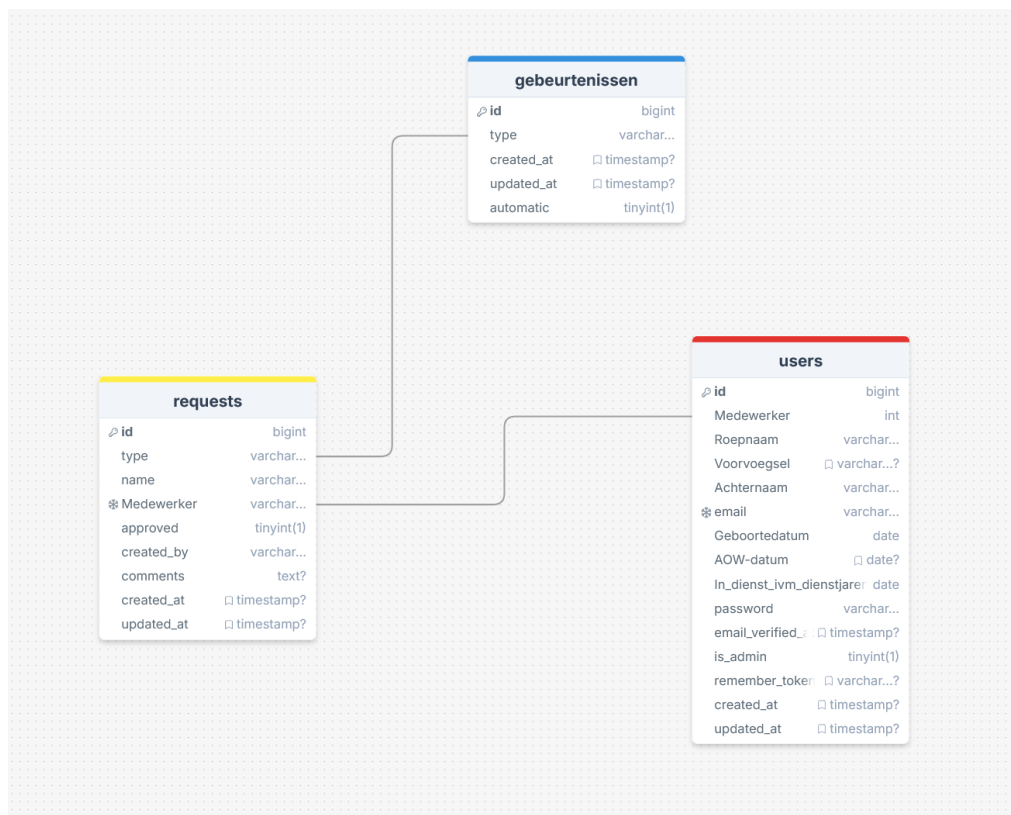
In het tekstveld eronder kan de gebruiker nog eventuele extra opmerkingen invoeren zoals een kleine toelichting. Onder dit veld staat de aanvraag knop, wanneer deze wordt ingedrukt controleert het systeem of alle velden correct zijn ingevuld, zo ja wordt de aanvraag doorgestuurd.

### Aanvragenoverzicht pagina:



Op dit scherm kan een HR-medewerker een overzicht zien van alle aanvragen. Bij elke aanvraag is de medewerker naam te zien, het type aanvraag(aanleiding voor de aanvraag), de datum van de aanvraag en eventuele acties. Bij ziekte aanvragen kan de aanvraag goed of afgekeurd worden door de gebruiker.

## 4.2. Datamodel



Hoe  
het werkt:

De **users**  
tabel bevat  
alle

medewerker gegevens zoals naam, e-mail, geboortedatum en of iemand beheerder is. Elke medewerker krijgt een uniek medewerker-ID dat wordt meegegeven aan requests wanneer er een aanvraag wordt gedaan.

De **gebeurtenissen** tabel houdt verschillende soorten gebeurtenissen bij waar beroep op kan worden gedaan, bijvoorbeeld ziekte, een verjaardag of geboorte. Hierdoor wordt de juiste gebeurtenis gekoppeld met de juiste medewerker.

De **requests** tabel zorgt ervoor dat alle aanvragen worden opgeslagen die door medewerkers zijn aangemaakt. Elke aanvraag is gekoppeld aan een medewerker en een bepaalde gebeurtenis. De aanvraag heeft een type (bijvoorbeeld "huwelijk"), de volledige naam van de medewerker waar het voor is en zijn/haar medewerker ID, een boolean de "approved" status op true of false te zetten. Verder kan er nog extra informatie bij gezet worden in "comments".

**Praktisch voorbeeld:** Een medewerker wil vakantie aanvragen. Het systeem maakt een nieuwe record aan in de requests tabel, koppelt deze aan de medewerker en aan een "vakantie" gebeurtenis. Een manager kan de aanvraag bekijken, opmerkingen toevoegen en deze goedkeuren of afwijzen.

## 4.3. Advies

Door de website te hosten op een server naar keuze (bijvoorbeeld via een Cloud provider zoals DigitalOcean of een eigen server), wordt de webapplicatie toegankelijk op het internet.

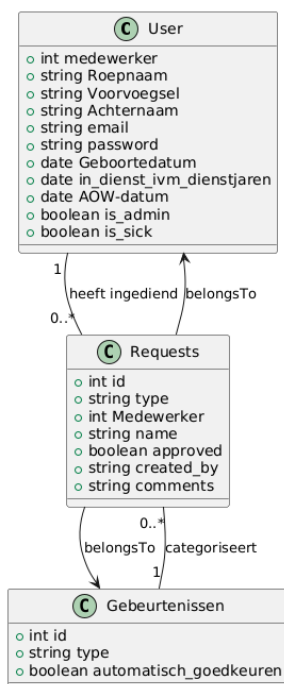
Na het online zetten van de website, kunt u inloggen met de **testgebruiker** om de website te gebruiken, op de website kunt u alle personeelsgegevens importeren. Deze testgebruiker heeft volledige beheerrechten.

### Inloggegevens testgebruiker:

- **E-mailadres:** test@example.com
- **Wachtwoord:** liedenleed

Zodra de personeelsgegevens zijn geïmporteerd, kunnen medewerkers met hun eigen account en wachtwoord inloggen om gebruik te maken van de website.

## 4.4. Module-overzicht



Hierbij een overzicht van de verschillende klassen. En de relaties ertussen.

## **5. Implementatie Details**

### **5.1. Programmeertaal en coding standaarden**

**Programmeertalen gebruikt:**

- PHP
- HTML
- Tailwind CSS

**Gevolgde coding conventies:**

- Microsoft Coding Convention
- KISS, DRY, YAGNI

### **5.2. Beveiliging en authenticatie**

Het systeem werkt met de inbegrepen beveiliging van Laravel, wat een goede basis aanbiedt voor een veilig systeem.

## 6. Testing en Validatie

### 6.1. Teststrategie

Wat we testen via ons testplan is Het importeren van data en ook de Request logica, ook testen we of de routes correcte autorisatie hebben. En als laatste testen we of de dashboard correct word ingeladen en er geen errors zijn. Via deze testen kunnen we bepalen of de core principes van onze applicatie correct werken.

Testresultaten:

```
PASS Tests\Feature\ImportTest
✓ import success 0.58s

PASS Tests\Feature\RequestControllerTest
✓ only admin can view requests index 0.04s
✓ admin can approve request 0.01s
✓ admin can reject request 0.01s

PASS Tests\Feature\RouteProtectionTest
✓ admin routes require admin role 0.02s
✓ employee search is accessible to authenticated users 0.01s

PASS Tests\Feature\DashboardTest
✓ dashboard shows all gebeurtenissen 0.01s
✓ employee not found returns error 0.01s

Tests: 8 passed (28 assertions)
Duration: 0.75s
```

## 7. Deployments en Onderhoud (optioneel)

### 7.1. Deployment strategie

Het systeem kan worden gehost in de Cloud via **DigitalOcean**, met automatische deploys via **Laravel Forge**. Wijzigingen in de code worden via Git gedeployed naar de live-omgeving.

### 7.2. Monitoring en onderhoud

Het systeem werkt met de inbegrepen beveiliging van Laravel, hierdoor is er al een veilige basis aanwezig voor het monitoren, verder zou een ICT-specialist van tijd tot tijd check-ups doen en als er iets fout is dit oplossen.

### 7.3. Schaalbaarheid

Bij toevoeging van medewerkers kan het systeem eenvoudig opschalen door extra servercapaciteit toe te voegen of over te stappen naar betere hosting. Laravel ondersteunt caching en queueing om prestaties te verbeteren bij veel gebruik.

## 8. Globale planning

Taak	Wanneer het af moet	Tijdsinschatting
Globaal overzicht project	Tweede project week	Aantal uren
Wireframes	Voor eerste versie	Aantal uur mee bezig
Eerste versie SDD	26-5-2025	Alle weken tot aan dit moment.
Basis van het programma functioneel	26-5-2025	Aantal weken.
Grootste deel SDD af.	1-6-2025	Paar uur aan bijwerken
Laatste versie programma	14-6-2025	3 project week tot hier.
Definitieve inlevering	16-6-2026	Alle weken tot dit moment.



## Azure omgeving

To Do

+ New item

6 Mockup

To Do

Doing

1/5

1 Sdd

Doing

Ryan Dürr

Done

15 Import logic

Done

Jesper Meuzelaar (student)

16 dashboard logic

Done

Jesper Meuzelaar (student)

17 GUI

Done

Jesper Meuzelaar (student)

14 responsiveness

Done

Jesper Meuzelaar (student)

13 frontend

Done

Luuk Steltenpool

12 procesarchitectuur

Done

Luuk Steltenpool

11 Planning

Done

Luuk Steltenpool

10 Admin Dashboard

Done

Ryan Dürr

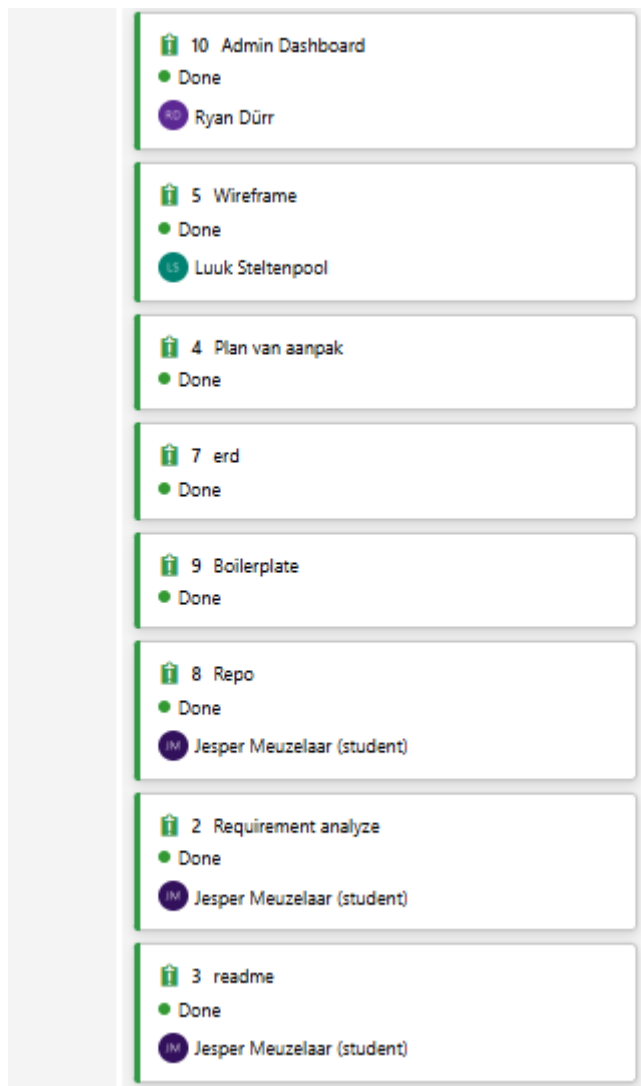
5 Wireframe

Done

Luuk Steltenpool

4 Plan van aanpak

Done



Op deze afbeeldingen is onze Azure omgeving te zien. Hierin hebben wij voornamelijk gewerkt om overzicht te houden over het project en wie waaraan werkt. Wij hebben ons project voltooid in 3 sprints.

Sprint 1: In de eerste sprint hebben wij verkenning gedaan naar het project en ideeën bedacht en uitgewerkt. Het doel hiervan was vooral om een basis te leggen.

Sprint 2: In de tweede sprint hebben we gewerkt richting de eerste versie van het SDD en de eerste versie van een werkend programma.

Sprint 3: In de derde sprint hebben we gefocust op het afronden van alles en zorgen dat we zowel in de documentatie als in het programma alles hadden.