Technisch ontwerp showcase

Sil Gosker

**Versiebeheer**

|  |  |  |
| --- | --- | --- |
| **Wie** | **Wat** | **Wanneer** |
| Sil Gosker | Opzet van het document | 2025-02-05 |
| Sil Gosker | Initialiseren van UC1 | 2025-02-05 |
| Sil Gosker | Initialiseren van UC2 | 2025-02-07 |
|  |  |  |

Contents

[2 Inleiding 4](#_Toc189818745)

[2.1 Probleemstelling 4](#_Toc189818746)

[2.2 Oplossing 4](#_Toc189818747)

[3 Systeem context 5](#_Toc189818748)

[3.1 Actoren 5](#_Toc189818749)

[3.2 Website 5](#_Toc189818750)

[3.3 Mailserver 6](#_Toc189818751)

[4 Container context – website 7](#_Toc189818752)

[4.1 Component context – Frontend 8](#_Toc189818753)

[4.1.1 UC1: Code context – Homepagina 9](#_Toc189818754)

[4.1.2 UC2: Code context – Contactpagina 10](#_Toc189818755)

[4.2 Component context – Backend API 11](#_Toc189818756)

[4.2.1 UC2: Code context – Mailing module 12](#_Toc189818757)

# Inleiding

Dit document bevat de technische documentatie. Dit omvat visueel en tekstueel vormgegeven ontwerpen op basis van requirements en ontwerpen uit het functioneel ontwerp. Er zal van een globale scope naar detail genavigeerd worden om overzicht te behouden. In de inleiding zal naast deze introductie ook de probleemstelling- en oplossing geïntroduceerd worden. Het verkleinen en detailleringsniveau van de scope zal middels het C4-model vormgegeven worden. De vierde laag zal echter hybride vormgegeven zijn om ruimte te geven de use cases hierin te verwerken.

## Probleemstelling

Stage aanbieders, werkgevers en recruiters zijn allemaal geïnteresseerd in nieuwe ontwikkelaars en willen oppervlakkig in contact met hen komen. Dit wordt in hoge mate vermoeilijkt door het feit dat er online geen verbinding gelegd kan worden met deze ontwikkelaars. Hierdoor wordt teruggevallen op sociale media platformen zoals LinkedIn, wat niet wenselijk is.

## Oplossing

De ontwikkelaar in kwestie zal een website ontwikkelen waarin duidelijk is wat hij kan en waar hij geïnteresseerd in is. Daarnaast moet een contactformulier geïntegreerd worden in de website waardoor oppervlakkig contact opgenomen kan worden.

# Systeem context

A diagram of software

Description automatically generatedIn dit hoofdstuk zal de globale systeemcontext uitgelegd worden. Deze kan gevonden worden in Figuur 1: Systeemcontext.

Figuur 1: Systeemcontext

## Actoren

In het onderstaande overzicht kan gevonden worden welke actoren relevant zijn voor het huidige systeem:

|  |  |
| --- | --- |
| **Actor** | **Beschrijving** |
| Geïnteresseerde | Een bezoeker op de website die geïnteresseerd is in de ontwikkelaar van de website zelf. |

## Website

De website is waar de geïnteresseerde naar gaat als hij meer wilt weten over de ontwikkelaar. De website bevat verschillend componenten die toekomstige uitbreidingen en ontwikkelingen mogelijk maken.

## Mailserver

Om berichten te versturen en contact te leggen moeten e-mails verstuurd worden. Dit zal gedaan worden middels een externe mailserver. Omdat mailserving en het onderhouden van zo’n server buiten de scope van het project ligt zal hier niet dieper op ingegaan worden.

# Container context – website

In dit hoofdstuk zal de context van de website zelf beschreven worden. Dit omvat de frontend, backend en databases. Ieder container zal globaal beschreven worden, waarna in detail gekeken zal worden naar de mogelijkheden van het container middels de laag drie en vier van het C4 model.

A blue rectangular sign with white text

Description automatically generated

Figuur 2: Container context website

**Frontend**

De frontend omvat alles wat de gebruiker ziet; dit is waar de wireframes in het functionele ontwerp gerealiseerd worden. Echter, hier zitten geen functionaliteiten aan vast. Er is minimale interactie mogelijk. Voor uitgebreide interactie en volledige realisatie van de requirements is een backend nodig. Om consistentie te verhogen is gebruik gemaakt van TypeScript. Daarnaast maken het ontwikkelingsproces webcomponents makkelijker, consistenter en sneller. Hiervoor is svelte gekozen. Om de UI consistent te houden, is tailwind gekozen als CSS framework.

**Backend API**

De backend API verwerkt alle data. Denk aan het versturen van emails middels het contactformulier. Het verwerkt dan ook alleen data. Om consistentie te verhogen is gekozen voor een gecompileerde, hard gecodeerde programmeertaal zoals C# en ASP.NET. Voor het beveiligen van data is gekozen om gebruik te maken van Entity Framework Core in combinatie met het NpgSQL pakket. Toegangspunten (endpoints) worden onderhouden door FastEndPoints.

## Component context – Frontend

Dit hoofdstuk zal dieper ingaan op de verschillende componenten die gevonden kunnen worden in de Frontend-container. Een visuele weergave hiervan kan gevonden worden in Figuur 2: Container context website. Ieder component zal kort besproken worden, waarna dieper ingegaan zal worden op de individuele componenten middels de vierde laag van het C4 model. Deze laag is vormgegeven d.m.v. UML klassendiagrammen. Ondanks dat er geen klassen worden gerealiseerd, kan een klassendiagram wel de relatie tussen verschillende componenten en hun eigenschappen weergeven.

**A blue square with white text

Description automatically generatedHomepagina**

Figuur 3: Component context Frontend

De homepagina omvat UC1: bekijken profiel. Denk hierbij aan het inzien van foto’s, de introductie tot de ontwikkelaar of vaardigheden/eigenschappen.

**Contactpagina**

De contactpagina is de pagina waar, wanneer relevant, de geïnteresseerde contact kan leggen met de ontwikkelaar. Dit gebeurt middels een contactformulier waar een aantal verplichte data ingevoerd moet worden.

### UC1: Code context – Homepagina

De homepagina bestaat uit een aantal componenten die teruggevonden kunnen worden in de code. Denk hierbij aan de slideshow of de beschrijving. Hou er rekening mee dat dit geen klassen representeren, maar (svelte)componenten. Een visuele weergave van het diagram kan gevonden worden in Figuur 4: code context Home pagina. Per component zal een beschrijving gegeven worden aan de functie van dat component.

A blue screen with black text

Description automatically generated

**HomePage**

De HomePage is het “root”- component. Dit betekent dat dit component initieel wordt ingeladen als naar “/” genavigeerd wordt. Dit component laadt vervolgens 3 andere componenten in: AboutMeDescription, SlideShow en SkillGraph

Figuur 4: code context Home pagina

**AboutMeDescription**

Dit component bevat een nette, stijlvolle beschrijving voor de geïnteresseerde.

**SlideShow**

Dit is een container-component waar een slideshow gestart kan worden. Alle elementen die hier in terecht komen worden als individuele slides gezien die horizontaal voorbij “scrollen”.

**Slide**

Dit representeert een individuele slide. Dit omvat een standaardgrootte- en breedte en andere stijlen.

**SkillGraph**

Dit is het component dat verantwoordelijk is voor het tonen van eigenschappen van de ontwikkelaar. Dit gaat door middel van een naam (bijv. C#) en een percentage (0-100), waarbij 0 compleet onbekend is, en 100 expert/senior-niveau aan kennis.

### UC2: Code context – Contactpagina

De contactpagina bestaat uit een aantal componenten die teruggevonden kunnen worden in de code. Denk hierbij aan input-velden met automatische validatie. Een visuele weergave van het diagram kan gevonden worden in Figuur 5: Code context Contact pagina.

**A blue rectangular object with text

AI-generated content may be incorrect.**

**ContactPage**

De ContactPage is het “root”-component. Dit betekent dat dit component initieel wordt ingeladen als naar “/contact” genavigeerd wordt. Dit component laadt vervolgens het ContactForm component in.

**ContactForm**

Het contactformulier bevat een aantal invoervelden en is verantwoordelijk voor het onderhouden van de gecombineerde data. Als het formulier verstuurd wordt, wordt een aanvraag naar de Backend API gedaan, waarnaar een melding getoond wordt dat de e-mail verstuurd is.

Figuur 5: Code context Contact pagina

**FormField**

Een herbruikbaar component dat gebruikt wordt voor het automatisch valideren van input. In de huidige situatie maakt ContactForm gebruik van 3 van deze velden: één voor de email; één voor het onderwerp en één voor het bericht zelf.

## Component context – Backend API

Dit hoofdstuk zal dieper ingaan op de verschillende componenten die gevonden kunnen worden in de Backend API-container. Een visuele weergave hiervan kan gevonden worden in Figuur 4: Component context Backend API. Ieder component zal kort besproken worden, waarna dieper ingegaan zal worden op de individuele componenten middels de vierde laag van het C4 model. Deze laag is vormgegeven d.m.v. UML klassendiagrammen.

**Mailing module**

De emailmodule is verantwoordelijk voor het versturen van e-mails op basis van de informatie die is meegegeven in het formulier op de contactpagina.

A diagram of a diagram

Description automatically generated

Figuur 6: Component context Backend API

### UC2: Code context – Mailing module

A close-up of several folders

AI-generated content may be incorrect.Dit hoofdstuk zal dieper ingaan op de inhoud van de mailing module. Dit is vormgegeven d.m.v. een klassendiagram. Alle relevante klassen zullen individueel besproken worden om hun rol en relevantie te bespreken. Het klassendiagram is gevisualiseerd in Figuur 7: Code context Mailing module.

Figuur 7: Code context Mailing module

**SendMailRequest**

Dit representeert het dataformaat dat verwerkt wordt. Daarom zitten hier alle aspecten in die relevant zijn voor het versturen van een e-mail, zoals de verstuurder, het bericht en het onderwerp.

**SendMailEndpoint**

Het aanvraagpunt (endpoint) wordt gedefinieerd door de SendMailEndPoint klasse. Deze bevat de IEmailService om de email te versturen. Dit wordt d.m.v. Dependency Injection resolved.

**IEmailService**

De IEmailService is de interface die definieert wat er met emails gedaan kan worden. Door een expliciete interface te gebruiken, wordt de mogelijkheid opengesteld om de SendMailEndpoint te unit testen.

**SmtpEmailService**

De EmailService-klasse is de implementatie van de IEmailService, en gebruikt de SmtpClient om emails te versturen.

**EmailOptions**

De EmailOptions-klasse representeert alle configuratie die mogelijk is bij het versturen van emails. Het is mogelijk om hier later extra properties aan toe te voegen wanneer meer configuratie nodig is.