Technisch Ontwerp

UC2 Contactpagina

Niveau 2 semester 2

Team SE  
2024

# Versiebeheer

|  |  |  |
| --- | --- | --- |
| Versie | Datum | Wijzigingen |
| 0.1 | 27-5-2024 | Initiële setup |
| 1.0 | 30-8-2024 | Zorgen dat het TO dezelfde opbouw heeft als het TO van UC1 |

## Distributie

|  |  |  |
| --- | --- | --- |
| Versie | Datum | Ontvangers |
| 0.1 | 27-5-2024 | Karen |
| 1.0 | 30-8-2024 | Ernst |

Inhoud

[Versiebeheer 2](#_Toc175918376)

[Distributie 2](#_Toc175918377)

[Inleiding 4](#_Toc175918378)

[2 Setup 5](#_Toc175918379)

[2.1 Repositories 5](#_Toc175918380)

[2.2 Runnen project 5](#_Toc175918381)

[3 Technieken 6](#_Toc175918382)

[3.1 Tools 6](#_Toc175918383)

[3.2 Programmeertalen 6](#_Toc175918384)

[3.3 Frameworks 6](#_Toc175918385)

[3.4 Standaarden 6](#_Toc175918386)

[4 Definition of Done 7](#_Toc175918387)

[5 Systeem Context 8](#_Toc175918388)

[6 Containers 9](#_Toc175918389)

[6.1 Applicaties 9](#_Toc175918390)

[6.2 Threat modelling op Container level 10](#_Toc175918391)

[7 Componenten 10](#_Toc175918392)

[7.1 Webapplicatie 10](#_Toc175918393)

[7.2 ShowcaseAPI 11](#_Toc175918394)

[8 Security Maatregelen 11](#_Toc175918395)

[9 Deployment 12](#_Toc175918396)

[9.1 Overzicht deployment 12](#_Toc175918397)

[10 Figuren 14](#_Toc175918398)

[11 Bibliografie 15](#_Toc175918399)

[12 Bijlage 1 Aanpak Technisch Ontwerp 16](#_Toc175918400)

[12.1 Nieuwe requirements 17](#_Toc175918401)

[12.2 Ontwerpen van C4 met Draw.io 17](#_Toc175918402)

[12.3 C4 level 1 en 2 17](#_Toc175918403)

[12.3.1 Threat Modeling 17](#_Toc175918404)

[12.3.2 Gebruik Threat List 18](#_Toc175918405)

[12.4 C4 level 3 en 4 18](#_Toc175918406)

[12.4.1 Threat Modeling 19](#_Toc175918407)

[13 Bijlage 3 Handleiding Threat Model Tool Microsoft 20](#_Toc175918408)

# Inleiding

In dit technisch ontwerp wordt een systeem beschreven dat zich richt op het ontwerp en de implementatie van de use case die is beschreven in het Functioneel Ontwerp. Dit systeem stelt geïnteresseerden in staat om, na het vormen van een beeld van de skills van een student-webdeveloper (UC1 – profielpagina), contact te leggen met de student-webdeveloper. Het ontwerp omvat een gedetailleerde analyse van de systeemcontext, container- en componentdiagrammen.

Veel plezier met ontwerpen!

Team SE

# Setup

In dit hoofdstuk een beschrijving hoe het systeem lokaal te runnen is.

## Repositories

Om het systeem te kunnen runnen is het noodzakelijk om de bijbehorende respositories te clonen.

De repository op Github: <https://github.com/Windesheim-HBO-ICT/webdev-showcase-studentversie/tree/main/US2-Contactpagina>

<https://github.com/Windesheim-HBO-ICT/webdev-showcase-v2>

## Runnen project

Open de solution in Visual Studio 2022 of een soortgelijke tool. Voor het runnen van het project heb je .NET 8 nodig. Vervolgens is het project direct te starten, zonder verdere configuratie. Als je voor het eerst een ASP.NET webapplicatie runt (in HTTPS), dan zal je gevraagd worden om een SSL-certificaat te installeren. Zorg dat je bij die melding op akkoord klikt, daarmee voorkom je dat je dit later alsnog handmatig moet doen.

# Technieken

In dit hoofdstuk wordt een overzicht gegeven van de gebruikte tools en standaarden.

## Tools

In deze paragraaf een overzicht van de gebruikte tools.

**Visual Studio**Visual Studio is een IDE die de ontwikkelaar ondersteunt bij het schrijven van code, onder andere door middel van auto-completion en het genereren van code (co-pilot plugin).

**Github**Github wordt gebruikt voor de opslag en het beheer van de Git repository. Ook kan gebruikgemaakt worden van de CI/CD om code te deployen naar Cloudflare en Skylab.

## Programmeertalen

Bij het ontwikkelen van het systeem zijn een aantal talen gebruikt. Hieronder volgen de talen die gebruikt zijn.

**C#**De .NET applicatie is geschreven in C#.

**Javascript, HTML, CSS**  
Deze talen zijn gebruikt om een deel van het front-end te ontwikkelen.

## Frameworks

Bij het ontwikkelen van het systeem is gebruik gemaakt van twee frameworks. Deze zijn hieronder beschreven.

**.NET** (Microsoft, 2023)Voor het ontwikkelen van de back-end is gebruik gemaakt van .NET Core met een MVC project.

## Standaarden

Om de code kwaliteit te garanderen is gebruikt van linting. Hieronder de beschrijving van de gebruikte linters. Deze linters zijn onderdeel van de pipeline van het Project Vanilla Javascript.

**ESLint** (ESLint, 2023)  
Voor het ontwikkelen van kwalitatief goede code is ESLint geconfigureerd voor het front-end.

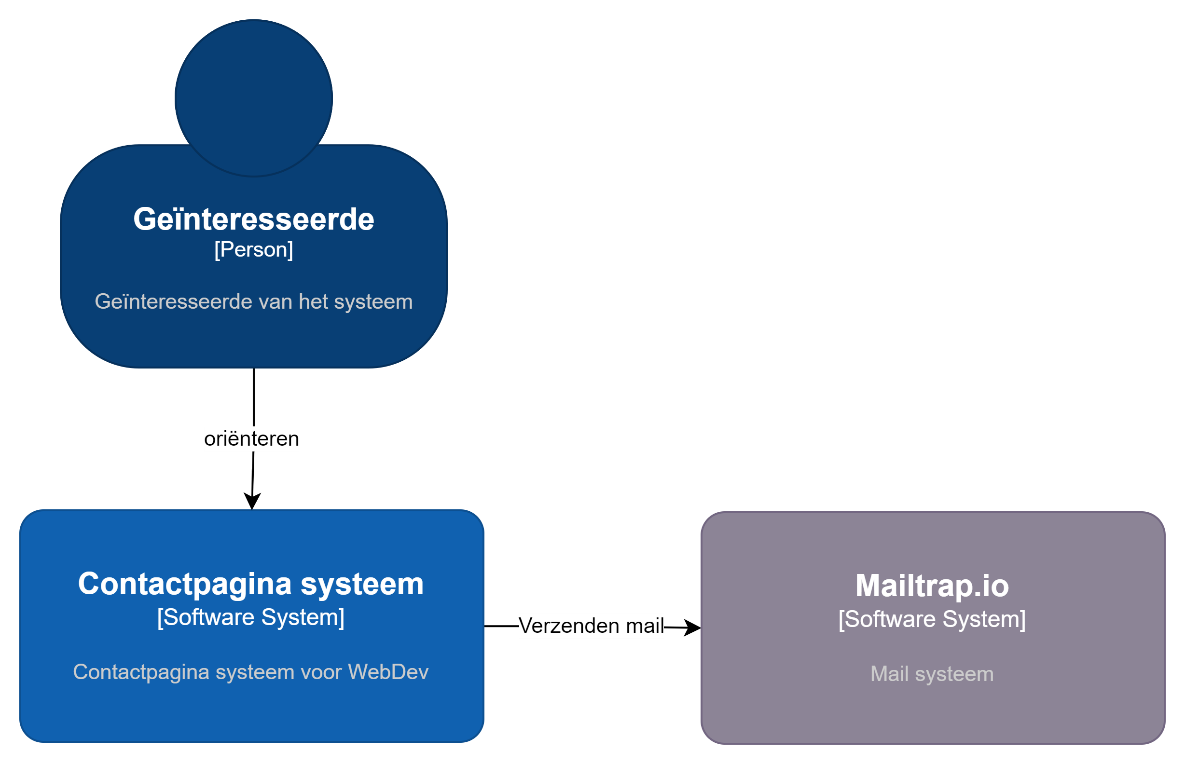
# Definition of Done

In dit hoofdstuk is de definition of done uitgewerkt. Dit zijn de eisen, waar nieuwe functionaliteit (technisch) aan moet voldoen, voordat deze kan worden afgerond. Zoals vastgelegd in het projectplan zijn deze eisen voorgelegd aan de opdrachtgever. Deze DOD zou in plaats van in het TO opgenomen kunnen worden in andere projectdocumentatie.

|  |  |
| --- | --- |
| Design | |
| 1. | Het FO en TO weerspiegelen de gerealiseerde functionaliteit |
| Development and Testing | |
| 2. | De gerealiseerde functionaliteit voldoet aan alle acceptatiecriteria (vastgelegd in het functioneel ontwerp) |
| 3. | De gerealiseerde functionaliteit voldoet aan de eisen gesteld in het issue ([zie issue templates](https://github.com/Windesheim-HBO-ICT/webdev-showcase/tree/master/2324%20semester%201/DOD/issue-templates)) |
| 4. | De testen van de gerealiseerde functionaliteit slagen allen |
| Deployment[[1]](#footnote-1) | |
| 5. | De pipeline van de staging omgeving slaagt |
| 6. | Alle secure parameters zijn opgenomen als enviroment variabelen |

# Systeem Context

Dit hoofdstuk geeft een overzicht van het systeem. De geïnteresseerde is iemand die contact wil leggen met de student-webdeveloper. Dit is bijvoorbeeld een bedrijf die een webdeveloper wil inhuren.



Figuur 1 Level 1 Systeem Context van het Contactpagina Systeem

# Containers

Dit hoofdstuk beschrijft de applicaties waaruit het systeem bestaat. Ook is de communicatie tussen de applicaties en de database beschreven.[[2]](#footnote-2)

## Applicaties

Het Container Diagram in Figuur 2 Container Diagram van de Showcase laat zien dat het systeem uit één applicatie bestaat een .NET Webapplicatie.

Afbeelding met tekst, schermopname, Lettertype, ontwerp

Automatisch gegenereerde beschrijving

Figuur 2 Container Diagram van de contactpagina

## Threat modelling op Container level

Op basis van de het Container Diagram van het systeem zou een Threat Model gemaakt kunnen worden. Deze kun je hier uitwerken.

# Componenten

In dit hoofdstuk wordt per applicatie getoond hoe de architectuur is vormgegeven. In dit geval is dit slechts 1 applicatie. Hierin is voor de architectuur MVC toegepast (Model-View-Controller).

## Webapplicatie

Afbeelding met tekst, schermopname, Lettertype, ontwerp

Automatisch gegenereerde beschrijving

Figuur 3 Componenten van de Webapplicatie container

## ShowcaseAPI

Afbeelding met tekst, schermopname, Lettertype, ontwerp

Automatisch gegenereerde beschrijving

Figuur 3 Componenten van de ShowcaseAPI container

# Security Maatregelen

In dit hoofdstuk moet een Threat Model worden ontwikkeld. Daartoe zijn in elk geval twee dringende redenen: gegevens vanuit het formulier in de Browser worden door de webapplicatie ontvangen en verwerkt. Dit moet nader onderzocht worden. Verder wordt met een externe partij gecommuniceerd. Ook daarvoor moet geïnventariseerd worden welke risico’s aanwezig zijn. De aanpak is beschreven in hoofdstuk *12 Bijlage 1 Aanpak Technisch Ontwerp*.

Afbeelding met diagram, tekst, lijn, cirkel

Automatisch gegenereerde beschrijving

Figuur 4 Threat Model

De bedreigingen en de gekozen maatregelen moeten verder uitgewerkt worden. Het Threat Model en de Threat List bevinden zich in de map met bestanden.

# Deployment

Het systeem staat in productie op Skylab. Deployen wordt gedaan voor Security for Web Applications, maar is niet verplicht voor Server Technology of Client Technology.

## Overzicht deployment

Het builden wordt lokaal binnen Visual Studio gedaan. Vervolgens wordt de build geplaatst in Skylab.

Afbeelding met tekst, schermopname, Lettertype, ontwerp

Automatisch gegenereerde beschrijving

Figuur 5 Deployment .NET applicatie

# Figuren

[Figuur 1 Level 1 Systeem Context van het Profielpagina Systeem 8](#_Toc175138706)

[Figuur 2 Container Diagram van de profielpagina 9](#_Toc175138707)

[Figuur 3 Componenten van de Webapplicatie container 11](#_Toc175138708)

[Figuur 4 Deployment .NET applicatie 12](#_Toc175138709)

[Figuur 5 Ontwikkelstappen Technisch Ontwerp 15](#_Toc175138710)

[Figuur 6 Threat Modeling in het Technisch Ontwerp 15](#_Toc175138711)

[Figuur 7 Het diagram in de Threat Model tool van Microsoft 17](#_Toc175138712)

[Figuur 8 Threat Modeling op C4 level 3 en 4 18](#_Toc175138713)

[Figuur 9 pijl wijst naar de optie om een nieuwe Threat Model aan te maken 19](#_Toc175138714)

[Figuur 10 Voorbeeld threat model pijl wijst naar componentenlijst 20](#_Toc175138715)

[Figuur 11 Pijl wijst naar rapport genereren 20](#_Toc175138716)

[Figuur 12 Pijl wijst naar knop 'Analysis View' 21](#_Toc175138717)

[Figuur 13 Pijl wijst naar knop 'Export to csv' 21](#_Toc175138718)

[Figuur 14 Threat List geïmporteerd in Excel 21](#_Toc175138719)

# Bibliografie

Cypress.io. (2023). *Why Cypress.* Opgehaald van Cypress: https://docs.cypress.io/guides/overview/why-cypress

ESLint. (2023). *Documentation.* Opgehaald van ESLint: https://eslint.org/docs/latest/

Microsoft. (2023). *.NET documentation.* Opgehaald van Microsoft: https://learn.microsoft.com/en-us/dotnet/

NUnit. (2023). *NUnit.* Opgehaald van NUnit Documentation Site: https://docs.nunit.org/

# Bijlage 1 Aanpak Technisch Ontwerp

In deze bijlage een overzicht van de stappen die genomen zijn om te komen tot een technisch ontwerp.

Eerst een schematische weergave van de stappen:

Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

Figuur 6 Ontwikkelstappen Technisch Ontwerp

In het SSDLC ziet dit er als volgt uit:

Afbeelding met tekst, schermopname, cirkel

Automatisch gegenereerde beschrijving

Figuur 7 Threat Modeling in het Technisch Ontwerp

## Nieuwe requirements

Bij het uitwerken van het ontwerp is gelet op nieuwe requirements. Deze zijn afgestemd met de stakeholders. Normaal gesproken wordt bij het vaststellen van een nieuwe requirement de requirementsanalyse aangepast. Om de ontwikkeling van requirements zichtbaar te maken zijn nieuwe requirements/ aan te scherpen requirements gemarkeerd.

## Ontwerpen van C4 met Draw.io

Het programma Draw.io is gebruikt voor de C4 diagrammen. Voor specifieke C4 layout is een [library beschikbaar](https://github.com/kaminzo/c4-draw.io).

## C4 level 1 en 2

Level 1 en 2 zijn eerst ontworpen. Bij het ontwerpen zijn de [regels van C4 toegepast](https://c4model.com/). Daarna is voldoende duidelijk om een Threat Model op te stellen.

### Threat Modeling

In plaats van handmatig het Threat model op stellen is gebruik gemaakt van een tool. Na het instellen van de tool wordt een Report en een Threat List gegenereerd. Deze lijst is te exporteren naar .csv. In Bijlage 3 Samenvatting Threat Model

Op basis van deze lijst kan een prioritering en keuze worden gemaakt welke security maatregelen worden genomen.

Afbeelding met tekst, diagram, Plan, schermopname

Automatisch gegenereerde beschrijving

Figuur 8 Het diagram in de Threat Model tool van Microsoft

Alle bestanden zijn te vinden op Github onder docs/Threat Model. Bijlage 3 is een handleiding voor het installeren en gebruiken van de Microsoft tool.

**Het aantal maatregelen in de tool is veel beperkter dan die van de** [**ASVS Quick Reference**](https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/)**. Daarom wordt een gecombineerde aanpak geadviseerd, die hierna beschreven wordt.**

### Gebruik Threat List

Op basis van het ingegeven model (C4 level 2) geeft de tool 60 threats, en daarbij 29 unieke maatregelen geplaatst. In deze lijst kan een prioritering worden aangebracht (Excel) en worden bepaald welke security maatregelen direct worden vastgelegd. Deze moeten ook in het Risk Assessment worden toegevoegd.

## C4 level 3 en 4

Per applicatie is een component diagram gemaakt. Vervolgens zijn op level 4 een aantal relevante onderwerpen beschreven (Deployment en Mail).

### Threat Modeling

Per onderwerp op level 3 en 4 is de Threat List en de ASVS geraadpleegd. Op basis van een eerste inschatting door de stakeholders zijn risico’s in het Risk Assessment opgenomen en voorzien van risico inschatting. In geval van een hoog risico is een Security Maatregel opgenomen en vervolgens een uitwerking in het Technisch Ontwerp opgenomen. In onderstaande diagram is dit proces beschreven.

Afbeelding met tekst, schermopname, Rechthoek, Lettertype

Automatisch gegenereerde beschrijving

Figuur 9 Threat Modeling op C4 level 3 en 4

# Bijlage 3 Handleiding Threat Model Tool Microsoft

**Stap 1: Download Microsoft Threat Modeling Tool.**   
Om toegang te krijgen tot de Microsoft Threat Modeling Tool, kun je de onderstaande link openen. Scroll vervolgens helemaal naar beneden op de pagina en onder het kopje "Next Steps" vind je de downloadlink voor de tool. Klik erop om de tool te downloaden.

Link: <https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-releases-73002061>

**Stap 2: Maak een nieuwe model aan.**

**Afbeelding met tekst, schermopname, software, Webpagina

Automatisch gegenereerde beschrijving**

Figuur 10 pijl wijst naar de optie om een nieuwe Threat Model aan te maken

**Stap 3: Ontwerp een thread model**

Aan de rechterzijde van het scherm bevinden zich de beschikbare componenten waarmee het thread model kan worden opgebouwd. In de onderstaande afbeelding heb ik de volgende componenten gebruikt:

* Generic Data Flow (verzoek en respons) om de stroom van verzoeken te illustreren.
* Generic TrustLine Boundary om de locaties van de niet-vertrouwde grenzen aan te geven.
* Browser
* Webapplicatie
* Web-API
* Database

**Een handige tip**: door op een component te klikken, kunt u aan de rechterzijde van het scherm, onder het gedeelte van het component, de naam wijzigen.

Afbeelding met tekst, schermopname, diagram, Perceel

Automatisch gegenereerde beschrijving

Figuur 11 Voorbeeld threat model pijl wijst naar componentenlijst

**Stap 4: Rapport generen**Om een volledig rapport te genereren, klik je bovenaan op "Reports". Selecteer vervolgens de optie "Generate full rapport". Zodra je een naam hebt ingevoerd, wordt het rapport automatisch gegenereerd. In dit rapport worden alle mogelijke kwetsbaarheden op basis van het **STRIDE-model** beschreven, samen met mogelijke maatregelen om ze te verminderen.

Afbeelding met tekst, schermopname, diagram, software

Automatisch gegenereerde beschrijving

Figuur 12 Pijl wijst naar rapport genereren

**Stap 5: Threat List genereren**Om de Threat List te genereren die je kunt exporteren naar csv klik je bovenaan op Analysis View.

Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

Figuur 13 Pijl wijst naar knop 'Analysis View'

Klik vervolgens op ‘Export to csv’.

Afbeelding met tekst, schermopname, software, Webpagina

Automatisch gegenereerde beschrijving

Figuur 14 Pijl wijst naar knop 'Export to csv'

Importeer het .csv bestand in Excel:

Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

Figuur 15 Threat List geïmporteerd in Excel

1. Dit komt pas aan de orde in een volgend semester (niet verplicht voor Web Development) [↑](#footnote-ref-1)
2. Op basis van dit hoofdstuk zou een Threat Model ontwikkeld kunnen worden. De bedreigingen en de gekozen maatregelen zouden dan vastgelegd moeten worden, bijvoorbeeld in een Excel bestand. Deze uitkomsten moeten vervolgens worden opgenomen in het Risk Assessment document. [↑](#footnote-ref-2)