

# Studentenhandleiding Showcase

Modules Client en Server Technology, Security for Webapplications

## Inhoud

Inleiding .....	3
1    Hoofdstuk 1 Big Picture .....	4
1.1    Idee .....	4
1.2    Samenhang vakken .....	5
1.3    Ontwikkelproces .....	5
1.4    Projecten, bouwen, deployen .....	6
2    Toetsing .....	9
2.1    Afspraken .....	9
2.1.1    Aftekenen .....	9
2.1.2    Definitief aftekenen .....	9
2.1.3    SSDLC .....	9
2.1.4    US1, US2 ... USn .....	9
3    Planning .....	11
4    Functionele en niet-functionele eisen US1 – Usn .....	12
4.1    US1 Profielpagina .....	12
4.1.1    Functionele requirements .....	12
4.1.2    Niet-functionele requirements .....	12
4.2    US2 Contactpagina .....	13
4.2.1    Functionele requirements .....	13
4.2.2    Niet-functionele requirements .....	13
4.3    US3 Showcase (week 3) .....	14
4.3.1    Niet-functionele requirements .....	14
4.4    US4 Showcase (week 4) .....	14
4.4.1    Functionele requirements .....	14
4.4.2    Niet-functionele requirements .....	14
4.5    US5 Showcase (week 5) .....	14
4.5.1    Niet-functionele requirements .....	14
4.6    US6 Showcase (week 6) .....	15
4.6.1    Niet-functionele requirements .....	15

## Inleiding

Dit document geeft een overzicht van de overkoepelende opdracht die je uitvoert in het semester Web Development. Het doel van de opdracht is om de technieken en methoden die je leert bij de drie betrokken vakken te laten gebruiken.

In het eerste hoofdstuk wordt de Big Picture geschetst. Hoofdstuk 2 beschrijft de toetsing, in dit hoofdstuk is ook de verzameling afspraken met betrekking tot de werkwijze opgenomen. Hoofdstuk 3 bevat een heel algemene planning van de opdracht.

Veel plezier!

Ernst Bolt




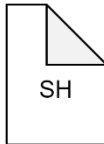
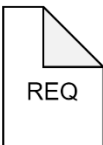

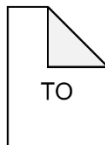
## 1 Hoofdstuk 1 Big Picture

Dit hoofdstuk biedt een overzicht over de hele opdracht. Hierbij komen documentatie, ontwikkelproces, toetsing en begeleiding aan de orde.

### 1.1 Idee

Het doel is om een systeem op te leveren. Het systeem is jouw Showcase. De Showcase ontwikkel je gedurende de eerste periode van het semester. Om je houvast te geven is de documentatie van de eerste twee userstories geheel uitgewerkt. Om je keuzevrijheid te geven ontwikkel je in de eerste weken een idee en legt dit vast in de documentatie. Dus, de gegeven documentatie gebruik je als basis en je breidt die uit op basis van een aantal userstories die je zelf bedenkt.

Hieronder een overzicht van alle documentatie die beschikbaar is:

Studentenhandleidingen			
	Client Technology	Server Technology	Security for Webapplications
studentenhandleiding			
	Overkoepelende opdracht - Showcase		
studentenhandleiding			
documentatie			

*Figuur 1 Documentatie*

Om je nog meer houvast te geven is ook de werkwijze van de opdracht in de volgende paragraaf vastgelegd en de relatie met de documentatie weergegeven.

## 1.2 Samenhang vakken

Het systeem wat je ontwikkelt in deze periode heeft eisen die voor alle drie de vakken gelden. De documentatie, requirements document (REQ), functioneel ontwerp (FO) en technisch ontwerp (TO) geldt voor alle vakken gezamenlijk.

## 1.3 Ontwikkelproces

Deze opdracht voer je individueel uit. Er is geen sprake van een opdrachtgever, specifieke events en de artefacts behorend bij SCRUM (backlog, sprintbacklog, retro verslag). De ontwikkelingmethode is dus geen Scrum, ook geen Agile. Maar wat dan wel?

Hieronder wordt de ontwikkeling van de Showcase beschreven in relatie tot een Secure Software Development Life Cycle (SSDLC). Een SSDLC geeft in een aantal stappen weer hoe software wordt ontwikkeld. Een methodologie van een SSDLC is Scrum, Kanban, RUP en Prince2.


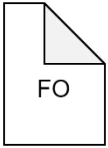
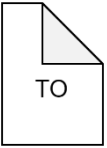
De SCLC van Snyk wordt bij deze opdracht aangehouden. Let nu vooral op de binnenste cirkel en leer die uit je hoofd: Requirements, Design, Development, Testing, Deployment. De buitenste cirkel houdt rekening met het security aspect, ook daar ga je rekening mee houden. Dus elk segment van de binnenste cirkel heeft een security segment in de buitenste cirkel.

Secure Software Development Life Cycle (SSDLC)



Figuur 2 Gehanteerd SSDLC

De eerste twee userstories zijn uitgewerkt, volgens onderstaande indeling:

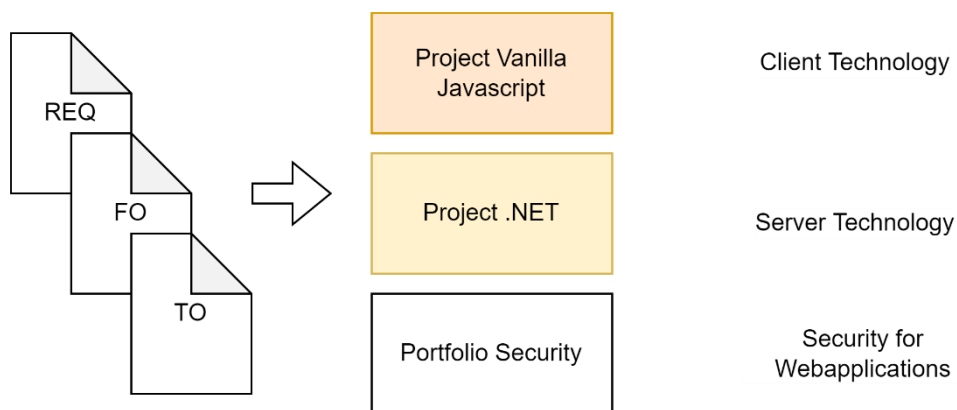
SSDLC			
Requirements	Beschrijving requirements inclusief risicoanalyse		
Design		Functionele beschrijving inclusief Threat Modeling	Technisch ontwerp inclusief Threat Modelling
Development			Statische analyse in IDE en

			CICD
Testing			Unittesten, vulnerability testen (ZAP)
Deployment			Deployment naar Cloudflare en Skylab Secure configuration

*Figuur 3 Documentatie per fase*

## 1.4 Projecten, bouwen, deployen

In deze periode leer je bij de drie vakken een aantal technieken en vaardigheden. Om deze te kunnen inzetten is gekozen voor een de volgende opzet. In deze paragraaf een eenvoudig overzicht van het totaal met een beschrijving. In het TO is een uitgebreidere technische uitwerking van het systeem.



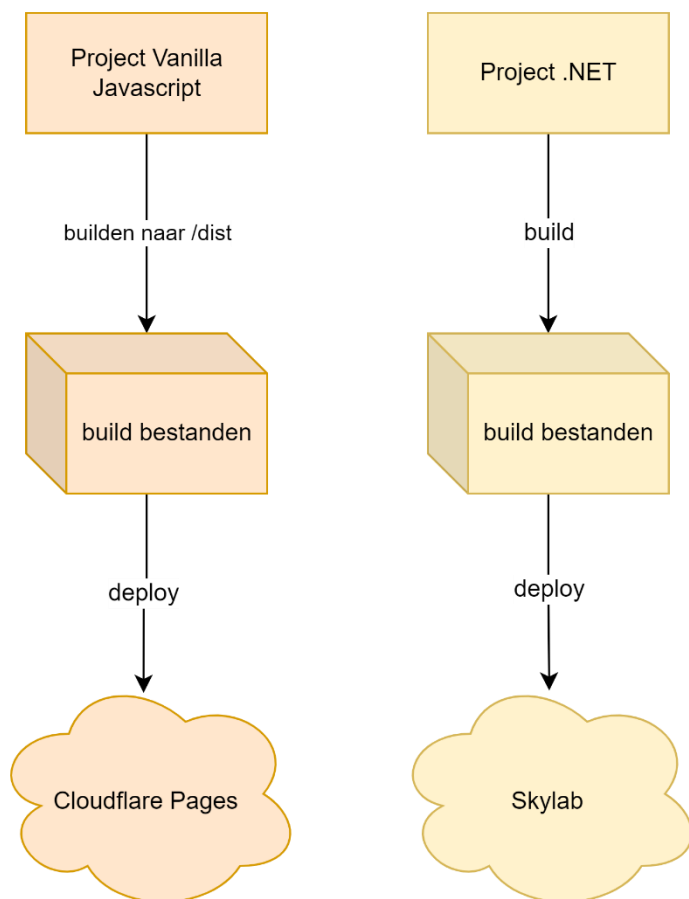
*Figuur 4 Voorbeelden van producten*

Vanuit de documenten ontwikkel je:

- een webapplicatie voor Client Technology,
- een webapplicatie voor Server Technology,
- een portfolio voor Security for Webapplications met daarin bewijsmateriaal waarmee je aantoont te voldoen aan de leeruitkomsten van dit vak.

Het systeem bestaat dus uit twee applicaties die samenwerken. De achterliggende techniek is uitgewerkt in het TO. In het gegeven ontwerp is een uitwerking gemaakt met een specifieke techniek, maar je bent vrij om dit te wijzigen naar andere frameworks.

Beide applicaties worden apart gebuild en gedeployed volgens onderstaand diagram:



*Figuur 5 Projecten en deployen*

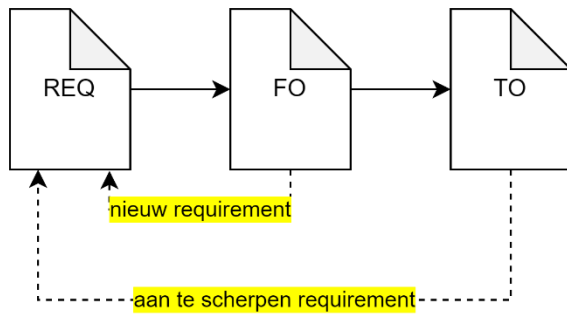
Beide applicaties hebben een deel van de functionaliteit van het systeem. Daarbij loop je wel tegen problemen aan, zoals:

- Hoe kun je geauthenticeerd zijn op beide systemen?
- Hoe communiceren beide applicaties met elkaar?

Deze vragen worden voor een deel al beantwoord in het TO. Een deel van de oplossing moet je zelf verder uitwerken.

Om te komen tot een veilig systeem worden in de requirements fase en de design fase al maatregelen genomen, security by design genoemd. Voor de eerste twee user stories is dit al uitgewerkt. Ook is beschreven welke stappen genomen zijn om te komen tot de genomen veiligheidsmaatregelen. De verantwoording van de toegepaste maatregelen neem je op in je security portfolio.

De documenten van de eerste twee user stories zijn na elkaar ontwikkeld: Requirements Analyse, Functioneel Ontwerp, Technisch Ontwerp. Tijdens de ontwikkeling van het Functioneel Ontwerp en het Technisch Ontwerp zijn nieuwe requirements ontdekt of is gebleken dat requirements aangescherpt moesten worden. Deze requirements zijn niet doorgevoerd in de Requirements Analyse, wel zijn ze expliciet **geel gemarkeerd**. Het doorvoeren hiervan dien je zelf te doen. In onderstaande diagram is deze werkwijze weergegeven:



*Figuur 6 Requirements in het proces toevoegen en aanscherpen*

Voor de andere userstories werk je deze documenten zelf uit. Ook kan het zijn dat je tijdens het ontwikkelen of testen ontdekt dat je meer security maatregelen moet nemen. Die leg je dan vast in je documentatie (REQ, FO, TO) en het portfolio van Security for Webapplications.



## 2 Toetsing

De overkoepelende opdracht is onderdeel van alle drie de vakken. In dit hoofdstuk staat beschreven wat getoetst wordt en op welk wijze dit plaats vindt.

Voor de vakken Client Technology en Server Technology worden de documentatie en de ontwikkelde userstories beoordeeld. Voor Security for Webapplications worden de maatregelen die genomen zijn en vastgelegd in de documentatie beoordeeld.

Overzicht van onderdelen die getoetst worden				
#	Onderdeel	Client Technology	Server Technology	Security for Webapplications
1	Development US1 & US2	V	V	V bewijzen in portfolio
2	Req & Design US Showcase	FO & TO	FO & TO	
3	Risk & Threat Modeling			FO & TO
<b>Dringend advies: vraag akkoord van je docent op #1 tot en met #3 voordat je #4 gaat ontwikkelen!</b>				
4	Development US Showcase	Code	Code	V bewijzen in portfolio

Figuur 7 Toetsonderdelen van de opdracht

### 2.1 Afspraken

#### 2.1.1 Aftekenen

De toetsonderdelen laat je aftekenen bij je docenten. Het dringend advies is om #1 tot en met #3 te laten aftekenen voordat je verder gaat met #vier.

Voorbeeld:

Als je klaar bent met toetsonderdeel #2 laat je deze aftekenen bij Client Technology en Server Technology. Ook heb je de security maatregelen vastgelegd en voorgelegd aan je docent. Daarna begin je met ontwikkelen.

#### 2.1.2 Definitief aftekenen

In de toetsweek en in de herkansingsweek is een aftekenmoment gepland. Na deze momenten is er pas weer einde periode 4 mogelijkheid tot aftekenen.

#### 2.1.3 SSDLC

Tijdens deze periode ontwikkel je volgens de SSDLC zoals is beschreven in dit document. Dus eerst ontwikkeling van REQ, FO en TO met bijbehorende security onderdelen, voorafgaand aan ontwikkelen en testen.

#### 2.1.4 US1, US2 ... USn

- US1 en US2 zijn uitgewerkt, maar je mag naar eigen inzicht hiervan afwijken, natuurlijk bijbehorende documentatie wijzigen en veiligheid in acht nemen.

- De werkwijze in de gegeven documenten kun je óf aanhouden óf op een zelfde kwaliteitsniveau wijzigen.
- De daarop volgende userstories kun je voorleggen aan je docenten

### 3 Planning

In dit hoofdstuk een overzicht van de planning van de Showcase.

week	Userstory/onderwerp		Aftekenmoment
1	US1 Profielpagina		
2	US2 Contactpagina		
3	Uitwerken idee voor Showcase		Aftekenmoment 1
4	Eigen US#	Rollen en rechten	
5	Eigen US#	Deployment	
6	Eigen US#	Testen	
7			Aftekenmoment 2
8			Aftekenmoment 3

## 4 Functionele en niet-functionele eisen US1 – Usn

### 4.1 US1 Profielpagina

#### 4.1.1 Functionele requirements

Beschrijving
Het profiel bevat een overzicht van skills
Het profiel bevat een beschrijving/introductie van de developer
Het profiel bevat een afbeelding van de developer: meerdere afbeeldingen toegestaan, slideshow met afbeeldingen is toegestaan

#### 4.1.2 Niet-functionele requirements

Vak	Beschrijving
Client	De gebruikte HTML-tags zijn semantisch waar dit mogelijk is
	Het design is Mobile First
	Het design is Responsive
	De profiel pagina bevat een GDPR
	De GDPR keuze wordt opgeslagen in een cookie (round trip)
	GDPR wordt alleen getoond als er geen consent is gegeven
	Styling GDPR past bij pagina
Server	Er wordt gebruikgemaakt van MVC of een ander architectural pattern. Je moet de keuze kunnen onderbouwen.
	De view dient strongly typed te zijn
	Tel het aantal navigaties (requests) van een gebruiker met behulp van een cookie

## 4.2 US2 Contactpagina

### 4.2.1 Functionele requirements

Beschrijving
De pagina bevat de naam van de developer
Het formulier bevat een invoer voor het onderwerp
Het formulier bevat een invoer voor e-mail
Het formulier bevat een text input voor het bericht
Wanneer het formulier verstuurd wordt, ontvangt de developer een mail met de ingevulde gegevens
Het formulier bevat een captcha

### 4.2.2 Niet-functionele requirements

Vak	Beschrijving
Client	Het formulier wordt alleen verstuurd als de inputs valid zijn
	Voorwaarden voor de inputs: a. Onderwerp, niet langer dan 200 tekens, b. E-mail, valide emailadres, c. Bericht, niet langer dan 600 tekens
	Het formulier bevat een captcha (simpel met een som óf ingewikkelder bijvoorbeeld Recaptcha v2/3)
Server	Er is een API endpoint beschikbaar die de data uit het formulier afvangt
	De data uit het formulier wordt gecontroleerd op een aantal voorwaarden. Deze voorwaarden zijn vastgelegd in een model: a. Onderwerp, niet langer dan 200 tekens, b. E-mail, valide emailadres, c. Bericht, niet langer dan 600 tekens
	Het endpoint geeft de juiste statuscode terug wanneer de waarden niet voldoen aan de voorwaarden
	De gevalideerde data uit het formulier wordt opgeslagen in een database (maak zelf de keuze tussen een relationele of niet-relationele database)
	Voor het versturen van de mail wordt gebruikgemaakt van een mail delivery service zoals SendGrid
Security	De contactpagina mag wel een beperkte set HTML opmaak attributen toestaan zoals opsommingen, bold of headers maar in ieder geval geen scripts, verborgen tekst of tekst in de kleur van de achtergrond.

### 4.3 US3 Showcase (week 3)

#### 4.3.1 Niet-functionele requirements

Vak	Beschrijving
Server	Er wordt tenminste één tabel in de database geseed met (test)data
	Asynchroon programmeren is toegepast waar nodig

### 4.4 US4 Showcase (week 4)

#### 4.4.1 Functionele requirements

Beschrijving
Er moet onderscheid worden gemaakt tussen ingelogde gebruikers en gastgebruikers. Wat een gastgebruiker/ingelogde gebruiker wel/niet mag is zelf te bepalen.
Als admin wil ik een gebruiker een rol kunnen geven (admin, moderator, user). Er moet onderscheid zijn gemaakt tussen de (on)mogelijkheden die deze rollen hebben. Wat een rol wel/niet mag is zelf te bepalen.
Als admin wil ik de rol van een gebruiker kunnen wijzigen

#### 4.4.2 Niet-functionele requirements

Vak	Beschrijving
Server	De authenticatie bestaat uit óf cookie based óf token based óf third party access (OAuth/OIDC)
	De autorisatie strategie is zelf te bepalen (policy based, claim based)

### 4.5 US5 Showcase (week 5)

#### 4.5.1 Niet-functionele requirements

Vak	Beschrijving
Server	Er is een ontwikkelstraat aanwezig die automatisch getriggerd wordt, bijvoorbeeld op het moment dat er gepushed wordt naar een branch, of wanneer er een merge request wordt uitgevoerd (zelf te bepalen).
	Er wordt automatisch gedeployed naar de productieomgeving ( <b>optioneel</b> )

## 4.6 US6 Showcase (week 6)

### 4.6.1 Niet-functionele requirements

Vak	Beschrijving
Server	De testen zijn toegevoegd aan de workflow (CI/CD) die je eerder gemaakt hebt. Als de testen niet slagen, moet het deploymentproces afgebroken worden.