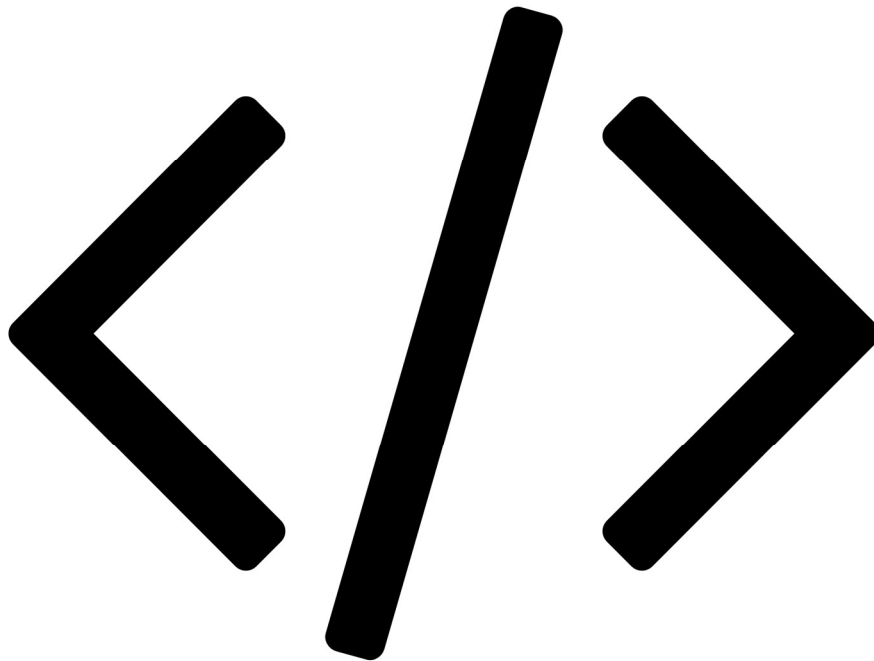


# PROJECT DOCUMENT

## APPLICATIE: HACK CHECK

Een online game waar je een database gaat hack doormiddel van een kleine minigame. Je kunt ook zelf een database maken die je met spelers kan delen.



**Gemaakt door:** Connor Timmerman

**Functie:** Student Fontys College

**Datum start project:** 8-2-2021

**Datum einde project:**

**Datum laatste wijziging:** 13-5-2021

**Versienummer:** 0.3

# INHOUDSOPGAVE

Project Document .....	1
Applicatie: Hack Check .....	1
Programma van eisen.....	4
Projectbeschrijving .....	4
Doelgroep(en) .....	4
Vergelijkbare Applicaties .....	4
MoSCow-rapportage .....	5
Vormgeving .....	5
Overig .....	6
Plan van aanpak.....	7
Inleiding.....	7
Doelstellingen.....	7
Benodigdheden .....	14
Takenlijst.....	14
Planning .....	15
Functioneel ontwerp .....	16
Inleiding.....	16
Functionaliteiten .....	16
Basis lay-out .....	17
Navigatiestructuur .....	20
Technisch ontwerp.....	21
Technische specificaties .....	21
Relationeel datamodel .....	21
datadictionary.....	22
Class Diagram.....	23
Ontwikkelomgeving.....	24
hardware .....	24
software .....	24
Realisatie.....	<b>Fout! Bladwijzer niet gedefinieerd.</b>
tesTPLAN .....	25

Testvoorbereiding .....	25
Testlog .....	27
Acceptatietest .....	28
testplan na aanpassingen.....	30
Productieomgeving .....	31
hardware .....	31
software .....	31
Versiebeheer .....	32
Evaluatieverslag .....	33
Betrokkenen .....	33
Evaluatie product .....	33
Evaluatie proces.....	33
Functieoverzicht.....	34
Gebruikershandleiding.....	35

# PROGRAMMA VAN EISEN

## PROJECTBESCHRIJVING

Dit project zal zijn voor mijn 2<sup>de</sup> semester op het Fontys College. Het zal een web applicatie worden die waarschijnlijk gebruik gaat maken van ASP.NET en MVC. Daarnaast zal het ook nog iets van een JavaScript Framework gebruiken dat verder beschreven zal worden in de technische details. Het zal er een beetje retro uit komen te zien met een groen en zwart kleuren schema, denk hierbij aan de Matrix bijvoorbeeld. Het hoofddoel van de applicatie zal entertainment zijn, het is tenslotte een soort game. Er zullen weinig tot geen echte hacker concepten gebruikt worden omdat dit waarschijnlijke te ingewikkeld zou zijn voor de gebruikers. Ik zie dit als een kleine puzzel game waar een potje maximaal 15 min zou moeten duren. Ik wil ook een bepaalde stijl uitstralen met deze game dat een beetje retro is zoals eerder beschreven, en dat dit ook het gene zal zijn dat de spelers het meeste zullen herinner

## DOELGROEP(EN)

Deze applicatie zal voornamelijk bedoeld zijn voor mensen die ten eerste een interesse hebben in puzzel games en die daarnaast wat interesse zullen hebben in ICT omdat er wel gebruikt zal gemaakt worden van bepaalde ICT termen zoals "Database" en meerdere dingen die relateren aan databasen. Ik denk dat wel de grootste doelgroep studenten zou zijn tussen de 16 en 24 jaar die zich een beetje hebben verdiept in ICT

## VERGELIJBARE APPLICATIES

Deze applicaties schetsen een goed beeld voor waar mijn applicatie op zal lijken:

### **HackRun**

[https://store.steampowered.com/app/378110/Hack\\_RUN/](https://store.steampowered.com/app/378110/Hack_RUN/)

HackRun gebruikt de stijl waar ik voor probeer te richten en heeft ook die snelle en simpele gameplay die ik in gedachte had.

### **Hacker Typer**

<https://hackertyper.net/>

Hacker typer gebruikt ook veel van de elementen die ik wil gaan gebruiken en is ook mijn grootste inspiratie geweest voor hoe het eruit moet komen te zien

### **Welcome to the Game**

[https://store.steampowered.com/app/485380/Welcome\\_to\\_the\\_Game/](https://store.steampowered.com/app/485380/Welcome_to_the_Game/)

Deze game heeft ook een hacker thema en ook veel verschillende hacker minigames die ik als inspiratie kan gebruiken om de minigame/ minigames te denken.

## MOSCOW-RAPPORTAGE

Must have's:	<ul style="list-style-type: none"><li>▪ Applicatie moet online benaderbaar zijn.</li><li>▪ Applicatie moet een login systeem hebben en een registreer systeem.</li><li>▪ Games moeten worden gespeeld vanuit een database.</li><li>▪ Een database game maker waar een gebruiker hun eigen puzzels kunnen maken.</li><li>▪ Gebruiker gemaakte databases moeten gedeeld kunnen worden.</li></ul>
Should have's:	<ul style="list-style-type: none"><li>▪ Data moet op een veilige manier verwerkt worden.</li><li>▪ Goed regels bij het maken van een wachtwoord</li><li>▪ Wachtwoord moet veranderd kunnen worden door de gebruiker</li><li>▪ Een manier om een sessie van een speler bij te houden</li></ul>
Could have's:	<ul style="list-style-type: none"><li>▪ Workshop voor verschillende databases om te spelen.</li></ul>
Won't have's:	<ul style="list-style-type: none"><li>▪ Geen uitgebreide gebruikersprofielen</li></ul>

## VORMGEVING

De font die voor de website gebruikt zal worden zal een oude retro stijl zijn daarom heb ik de volgende gekozen:

<https://www.fontspace.com/super-legend-boy-font-f34010>

Voor de kleuren wil ik voornamelijk #000000(zwart) gebruiken voor de achtergronden en #00ff01(groen)



# PLAN VAN AANPAK

## INLEIDING

Dit project is bedoeld voor een school project voor mijn 2<sup>de</sup> semester, Ik Connor Timmerman ben een student op het Fontys College in Eindhoven.

## DOELSTELLINGEN

Hier zal ik alle leeruitkomsten neerzetten wat de uiteindelijke doelstellingen zijn van dit project.

**Leeruitkomst 1:** Je baseert je keuzes op feedback van stakeholders en onderbouwt ze op een heldere en professionele wijze.

### **Stakeholder**

Iemand met een bepaalde rol en belang in het project (ongeacht of deze groot of klein is). Je kunt deze stakeholders identificeren en kunt hun belangen achterhalen en prioriteren.

### **Helder en professioneel**

Documentatie is compleet maar compact, niet-triviaal, gecontroleerd op spelfouten en toepasselijk voor de stakeholder waar deze voor bedoeld is.

### **Onderbouwen**

Geef betrouwbare en relevante bronnen voor alle beslissingen. Je beoordeelt bronnen op hun betrouwbaarheid en relevantie voor het project.

Reflectie:

Omdat ik zelf het project ha kiezen ben ik natuurlijk een van de belangrijke personen in dit project en zal ik ook mijn eisen hierin opschrijven. Daarnaast ga ik het beste deze eisen omzetten in een plan waar ik mijn keuze verantwoord en deze met de docent doorneem.

**Leeruitkomst 2:** Je werkt samen en communiceert met anderen op constructieve en professionele wijze.

### **Professioneel samenwerken.**

Je werkt samen aan een gemeenschappelijk doel en neemt initiatieven om het proces te verbeteren.

### **Professioneel communiceren.**

Je levert artefacten op aan de stakeholders en hebt zinvolle meetings met het team. Een artefact is een opgeleverd (deel)product dat waarde heeft voor de stakeholder. Voorbeelden zijn: analysedocumenten, ontwerpdocument, code en geïnstalleerde software.

### **Constructief**

Je reflecteert regelmatig op de manier waarop je werkt en hoe je handelen jou, anderen en het projectresultaat beïnvloedt. Hiervoor vraag je regelmatig feedback. Op basis van de informatie die je hieruit verkrijgt, maak je aanpassingen aan je gedrag.

Reflectie:

Ik blijf in contact met mijn docenten en zal minimaal 2 keer in de week feedback vragen over waar ik mee bezig ben daarnaast zal ik hun feedback noteren en toepassen waar dat nodig is.



**Leeruitkomst 3:** Je documenteert gevalideerde gebruikersspecificaties voor applicaties en vertaalt deze in correcte softwareontwerpen.

### **Gevalideerde gebruikersspecificaties**

Gevalideerd: je controleert dat requirements geaccepteerd zijn door de stakeholders en kunt ze zodanig prioriteren dat de eisen die de meeste waarde opleveren voor de stakeholders, de hoogste prioriteit krijgen.

### **Gebruikersspecificaties**

Het verwachte gedrag van het systeem, gespecificeerd in termen van interactie tussen de gebruiker en het systeem. Specificaties worden gevalideerd met behulp van uitvoerbare acceptatietests.

### **Correcte softwareontwerpen**

Je vertaalt specificaties naar relevante diagrammen waarin het technisch ontwerp beschreven wordt en die kunnen worden geïmplementeerd. Diagrammen die relevant kunnen zijn voor het implementeren van het product zijn onder andere: architectuurdiagrammen, domeinmodellen en databaseontwerpen.

Reflectie:

Ik maak alle benodigde documentatie dat bij een project hoort zoals een functioneel ontwerp en een technisch ontwerp, Dit heb ik al vaak genoeg gedaan op mijn vorige opleiding dus zal hier geen problemen mee hebben

**Leeruitkomst 4:** Je bouwt, ontwerpt en levert herhaaldelijk veilige en onderhoudbare applicaties op (waarvan er tenminste één web-gebaseerd is) die verbinding maken met een database en gebruik maken van OO-principes en standaard technieken gebaseerd op gevalideerde gebruikerseisen.

### **Herhaaldelijk**

Je maakt, breidt uit en onderhoud verschillende projecten.

### **Ontwerpen**

Het uitbreiden en onderhouden van projecten begint met herijken van de specificaties en het ontwerp.

### **Opleveren**

Je stelt de software op een dusdanige wijze beschikbaar dat de stakeholders er gebruik van kunnen maken.

### **Veilig**

Een softwaresysteem moet bescherm zijn tegen onbedoelde of onverwachte fouten. Onbedoelde fouten treden op als gebruikers het product gebruiken op een manier die niet was voorzien bij het opstellen van de specificaties. Onverwachte fouten treden op als iets faalt in het systeem, zoals bijvoorbeeld het niet beschikbaar zijn van een verbinding met de database.

### **Onderhoudbaar**

Een ontwerp moet klaar zijn voor toekomstige nieuwe eisen of aanpassingen aan bestaande specificaties.

### **OO Principes**

Gebruik gangbare OO-principes om het softwaresysteem te ontwerpen en dit ontwerp te onderbouwen.

Reflectie:

Het project zal gemaakt worden naar aanleiding van de documentatie die ik maak zal uitgebreid worden waar nodig is en als ik eerder klaar mocht zijn dat ik nog een ander klein project zal doen. Gebruikers data zal veilig opgeslagen door middel van encryptie.

**Leeruitkomst 5:** Je redeneert over computationele uitdagingen en implementeert algoritmisch complexe problemen in software.

### **Computationele Uitdagingen**

Je lost veelvoorkomende uitdagingen op, zoals een sorteerprobleem en onderbouwt waarom een bepaalde techniek geschikt is voor specifieke problemen.

### **Algoritmisch complexe problemen**

Je ontwerpt en implementeert algoritmes zodanig dat ze rekening houden met de randvoorwaarden van de stakeholders.

Reflectie:

Er zal een stuk complexiteit in komen te zitten voor het genereren van de user databases en natuurlijk in het gameplay stuk

**Leeruitkomst 6:** Je ontwerpt, bouwt en be vraagt een relationeel databasesysteem en integreert dit met een applicatie.

### **Ontwerpen**

Maak onderscheid tussen database- en softwareontwerpen. Een databaseontwerp bevat meerdere soorten relaties, zoals 1-op-veel en veel-op-veel.

### **Bevragen**

Je voert CRUD-operaties (Create, Read, Update en Delete) uit op data in een relationele database, en houdt rekening met performantie.

Reflectie:

Omdat de game volledig afhankelijk zal zijn van de database die er mee verbonden is. En het zal de standaard CRUD operaties bevatten

**Leeruitkomst 7:** Je verbetert en toont de kwaliteit van je software continue aan, gebruikmakend van standaard technieken en hulpmiddelen.

### **Continue**

Je werkt op een iteratieve wijze zonder bestaande functionaliteit te verstoren en waarbij veranderingen worden bijgehouden.

### **Verbeteren**

Gebruik standaard hulpmiddelen en technieken om de kwaliteit van je code te bewaken en te verbeteren.

### **Aantonen**

De code moet getest worden voor zowel het geplande gebruik, verwachte en onverwachte foutsituaties. Deze testen moeten meerdere malen kunnen worden uitgevoerd in meerdere fasen van het project. Verwachte fouten kunnen voortkomen uit de specificaties of van externe afhankelijkheden van de software.

### **Standaard technieken en hulpmiddelen.**

Gebruik bijvoorbeeld een versiebeheersysteem, acceptatietests en unit-tests.

Reflectie:

Iedere sprint werk ik om een deel van de applicatie op te leveren en stel deze doelen aan het begin van de sprint. Ook zal mijn code getest worden door een simpel test plan te maken

## BENODIGDHEDEN

- Computer:
  - 16 GB RAM, I7 Quad Core, 250GB opslag en geen overige eisen.
- Rand apparatuur:
  - Muis (Geen voorkeur), Toetsenbord (Geen voorkeur), Beeldscherm (Voorkeur naar 1920x1080 resolutie ), 2 van deze zou helpen.
- Werkruimte:
  - Een kantoor is prima zolang het maar een comfortabele stoel heeft met een bureau. Met stroom en een stabiele internet verbinding.
- Windows 10:
  - Het besturingssysteem voor de ontwikkel omgeving.
- Visual Studio Community:
  - De IDE die gebruikt gaat worden om met ASP.NET Core 5 te gaan werken
- MS SQL Server Express
  - De server die database gaat hosten voor de webapplicatie
- SQL Server 2019 Configuration Manager
  - De tool die gebruikt wordt om de MS SQL server te gebruiken
- Google Chrome
  - Voor testen en gebruiken voor het opzoeken van informatie
- Balsamiq 3
  - Voor het maken van de mock-ups en voor het maken van de ERD's
- Office 365
  - Dit pakket zal gebruikt worden voor het maken van de documentatie
- Microsoft Teams
  - Voor communicatie met de docent

## TAKENLIJST

Voor de alle taken en een dagelijks overzicht maak ik gebruik van een trello bord dat je hier kan vinden: <https://trello.com/b/k2NwoyNX>

## PLANNING

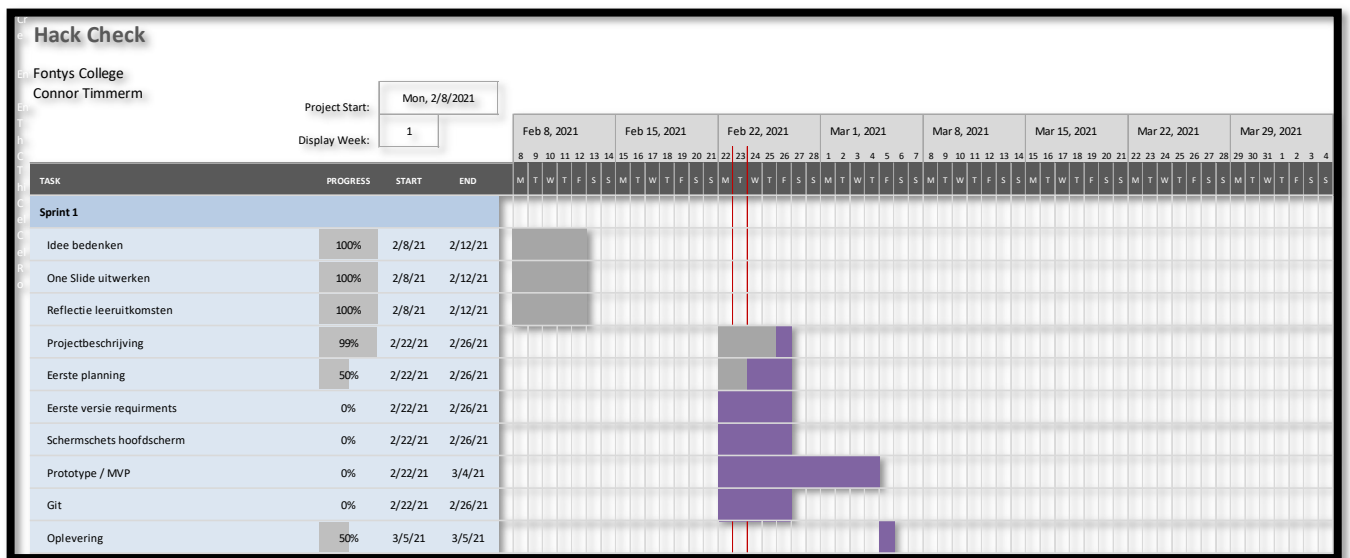


Foto 3: Een afbeelding van de GANTT Chart voor dit project

Voor dit project wordt er een GANTT Chart gemaakt aan het begin van iedere sprint met de hoofd taken verdeeld in tijdvakken en in hoe ver dit tijdens de sprint voltooid is.

Het volledige bestand kun je hier vinden: [HIER](#)

Voor de alle taken en een dagelijks overzicht maak ik gebruik van een trello bord dat je hier kan vinden: <https://trello.com/b/k2NwoyNX>

# FUNCTIONEEL ONTWERP

## INLEIDING

In dit ontwerp vind je een overzicht van de functionaliteiten waaraan de applicatie moet voldoen.

## FUNCTIONALITEITEN

Functionele requirements hoge prioriteit:

- Applicatie moet online benaderbaar zijn.
- Gebruikers moeten kunnen inloggen via een account.
- Gebruikers moeten een account kunnen aanmaken.
- Gebruikers moeten hun wachtwoord kunnen resetten.
- Gebruikers moeten kunnen uitloggen.
- Gebruikers moeten een level/database kunnen selecteren.
- Gebruikers moet het geselecteerde level kunnen spelen.
- Gebruikers moeten het geselecteerde level kunnen stopzetten.
- Gebruikers moeten een eigen level/database kunnen maken.
- Gebruikers moeten hun eigen level kunnen delen.
- Applicatie moet een sessie kunnen aanmaken voor een gebruiker

Functionele requirements lage prioriteit:

- Gebruikers kunnen een workshop open waar gebruiker gemaakte levels staan.
- Gebruikers moeten een level/database kunnen spelen uit de workshop.

Non-functionele requirements:

- Website design moet zich houden aan de aangegeven vormgeving
- Gevoelige gebruiker gegevens worden op een veilige manier opgeslagen
- Alles zal in het Engels zijn



## BASIS LAY-OUT

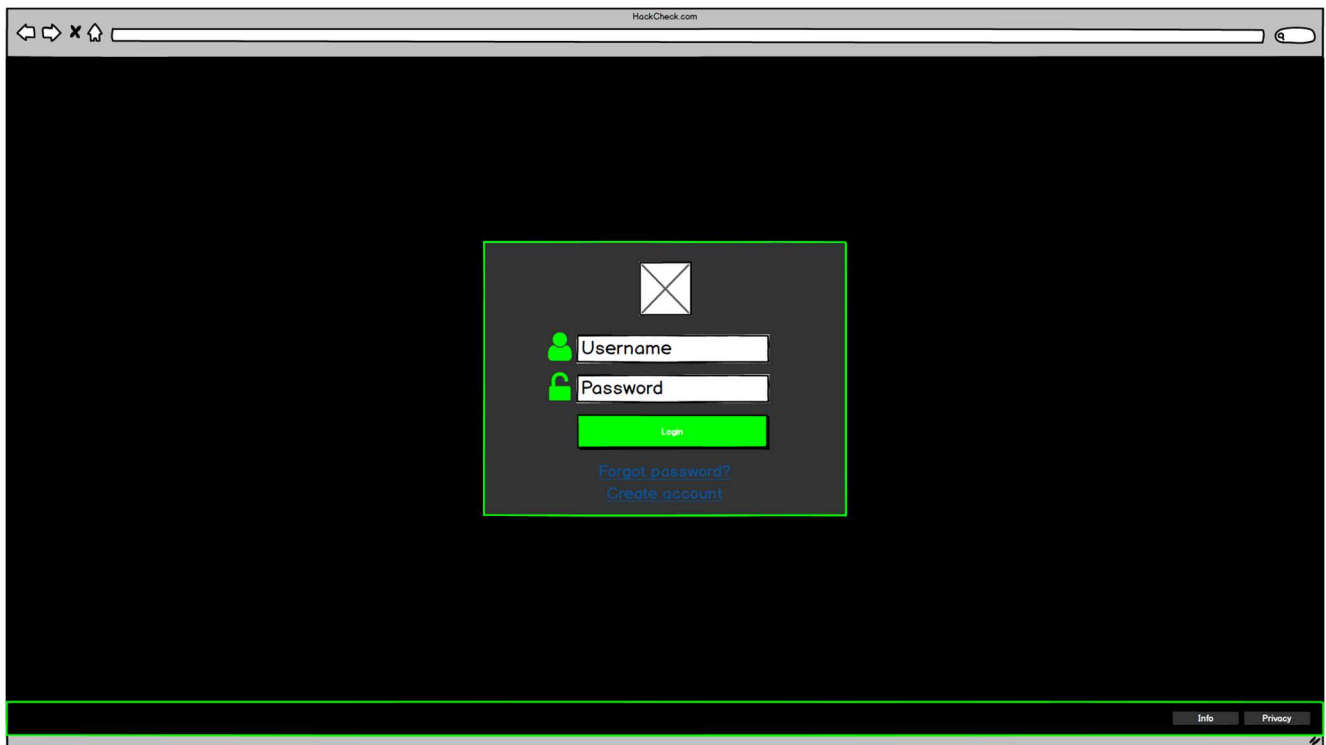


Foto 4: De login pagina om verder komen op de website

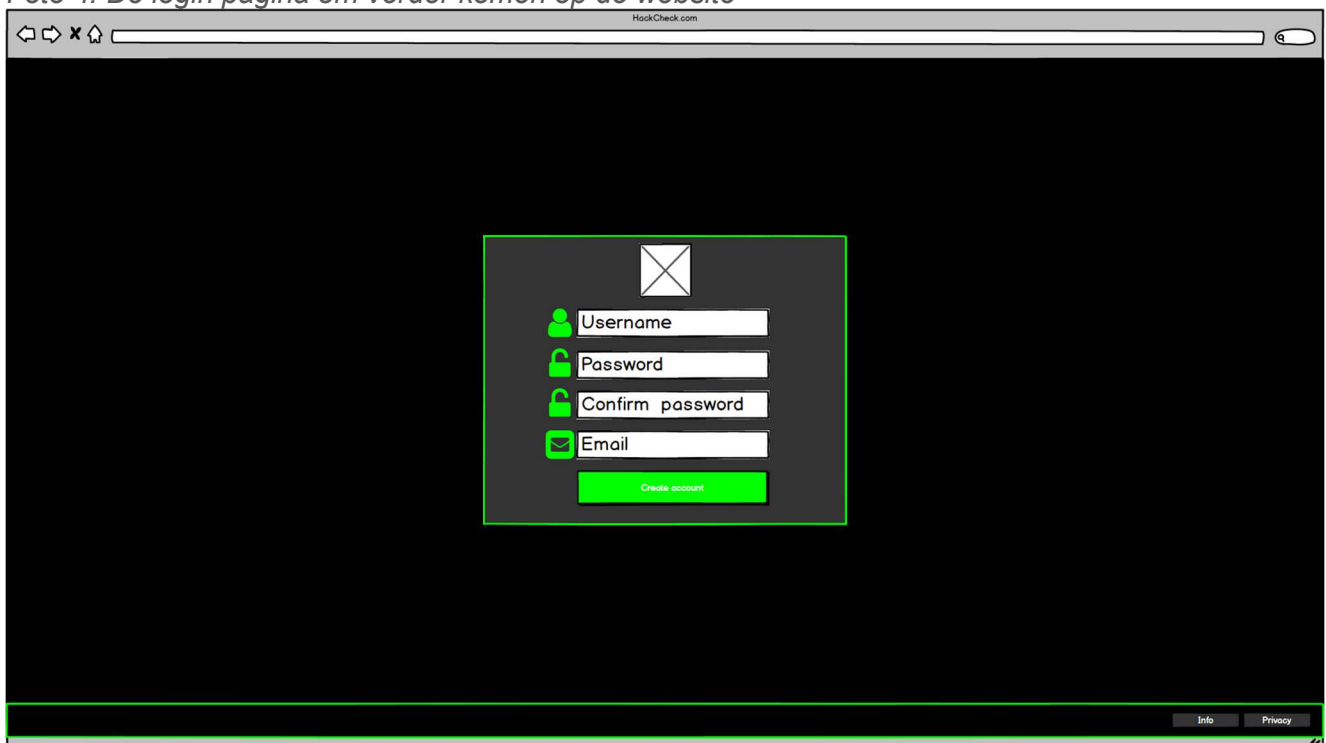


Foto 5: De pagina waar je nieuw account kan aanmaken

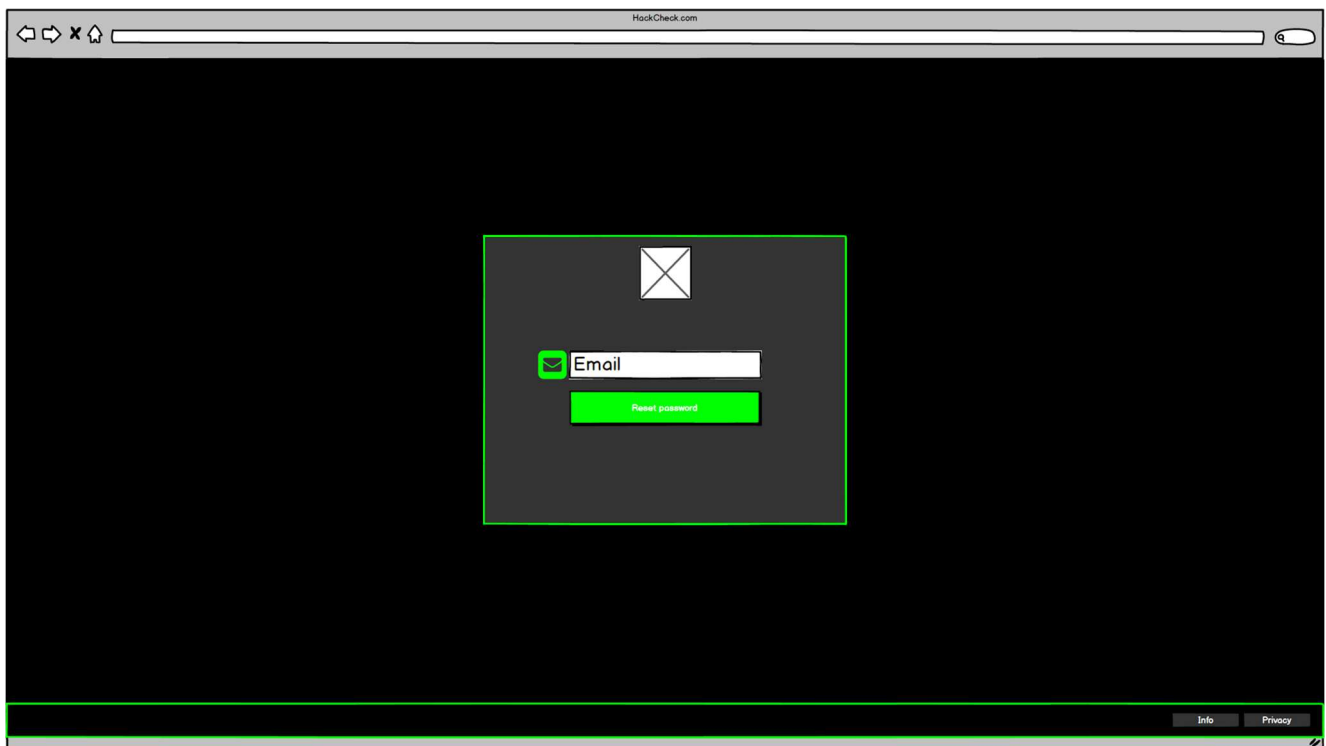


Foto 6: De pagina waar je een wachtwoord kan resetten

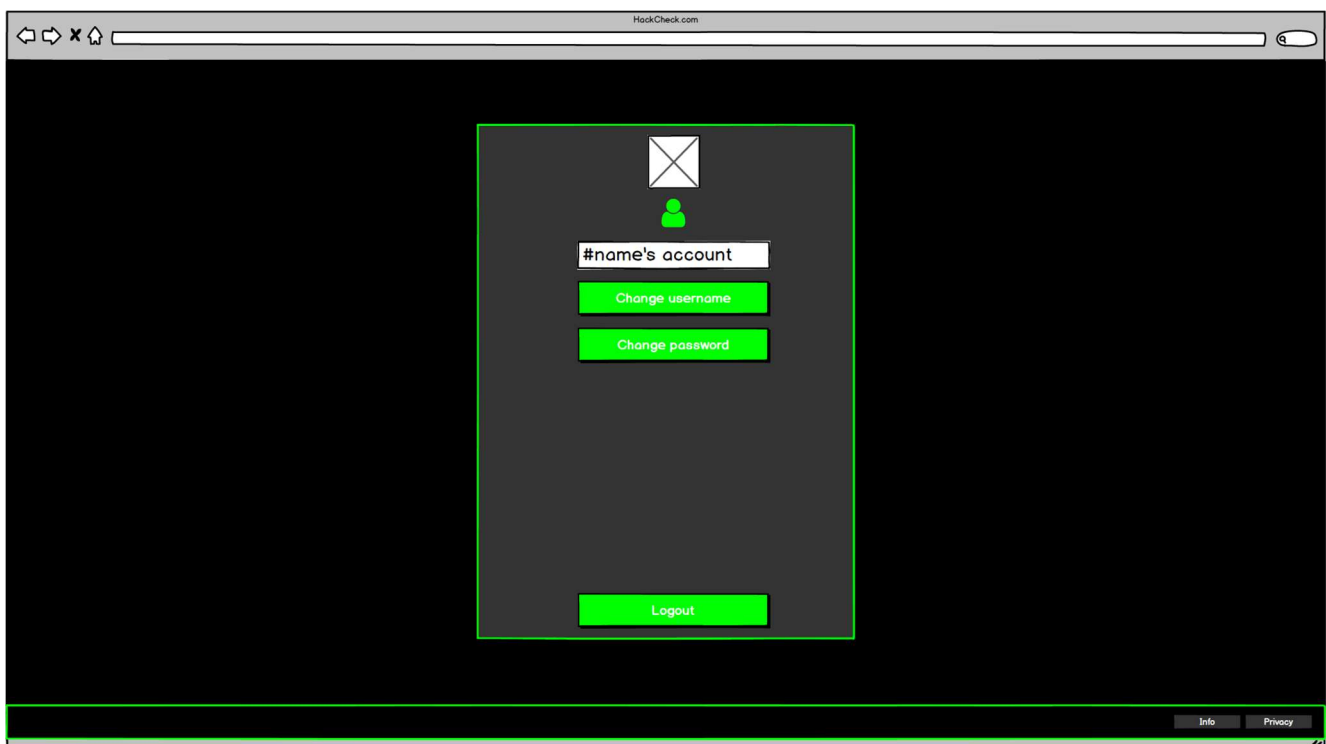


Foto 7: Account overzicht pagina met logout

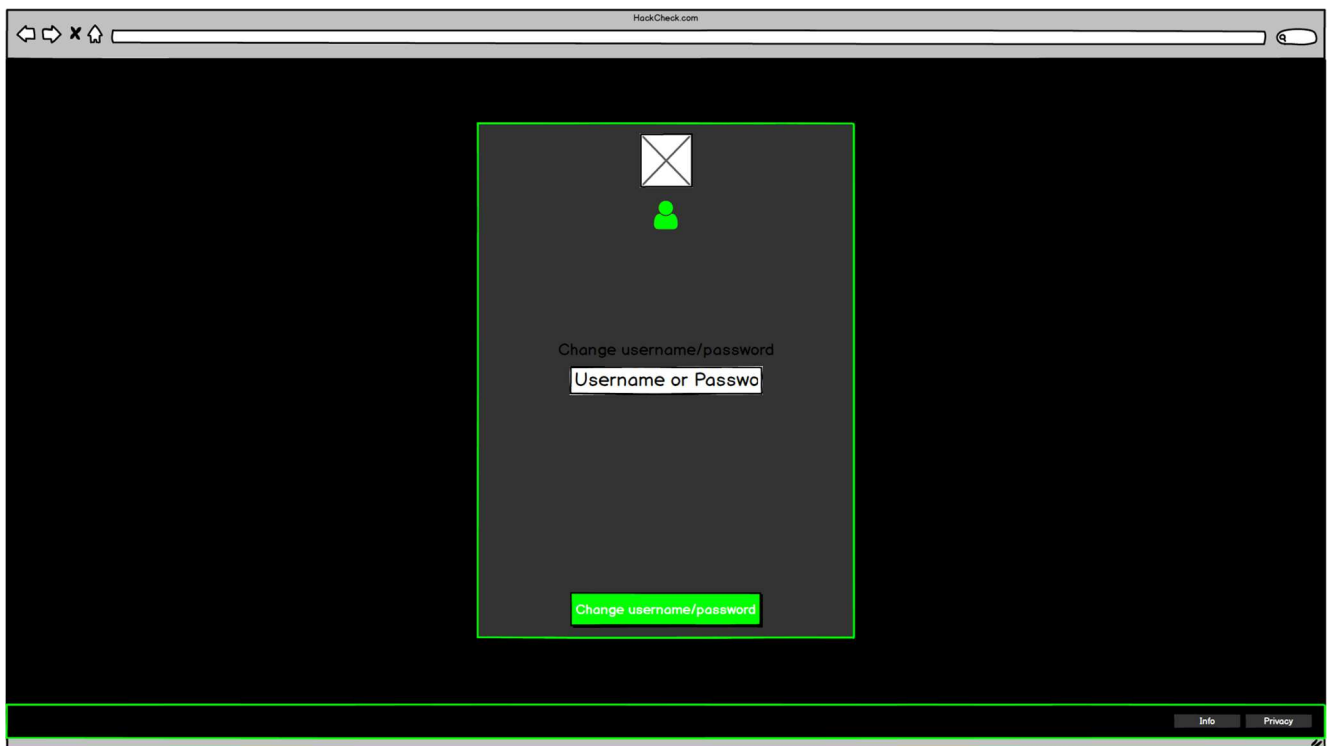


Foto 8: Verander wachtwoord of gebruikersnaam

## NAVIGATIESTRUCTUUR

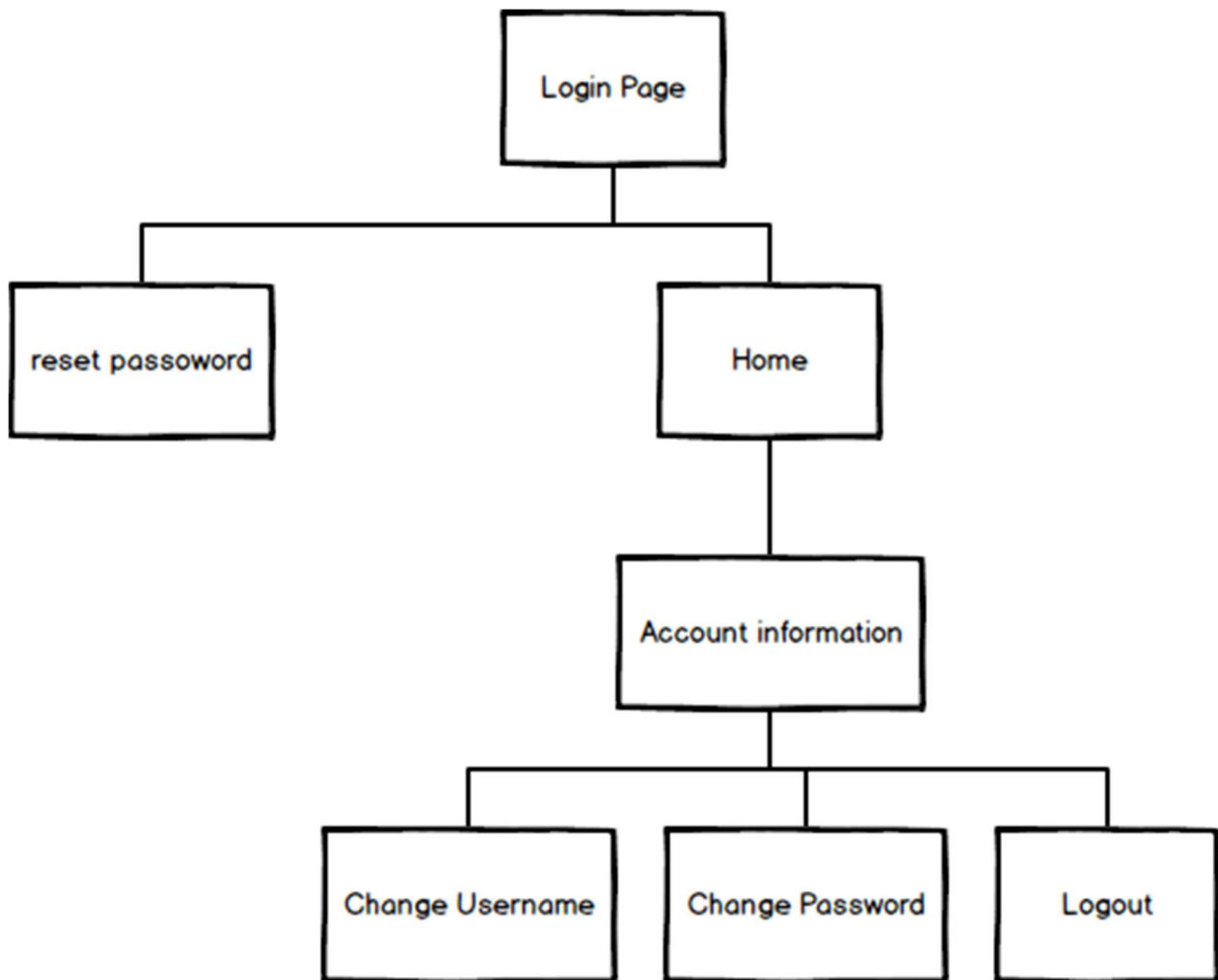


Foto 9: De structuur van de pagina's

# TECHNISCH ONTWERP

## TECHNISCHE SPECIFICATIES

Werk het functioneel ontwerp uit naar technische specificaties. Gebruik, indien gewenst, schematechnieken (bijv. Use Case, Activity Diagram, Class Diagram). Denk er ook over na hoe de beveiliging van de applicatie geregeld moet worden. (Security by design).

Je dient ervoor te zorgen dat de oplossing/uitwerking haalbaar en realistisch is.

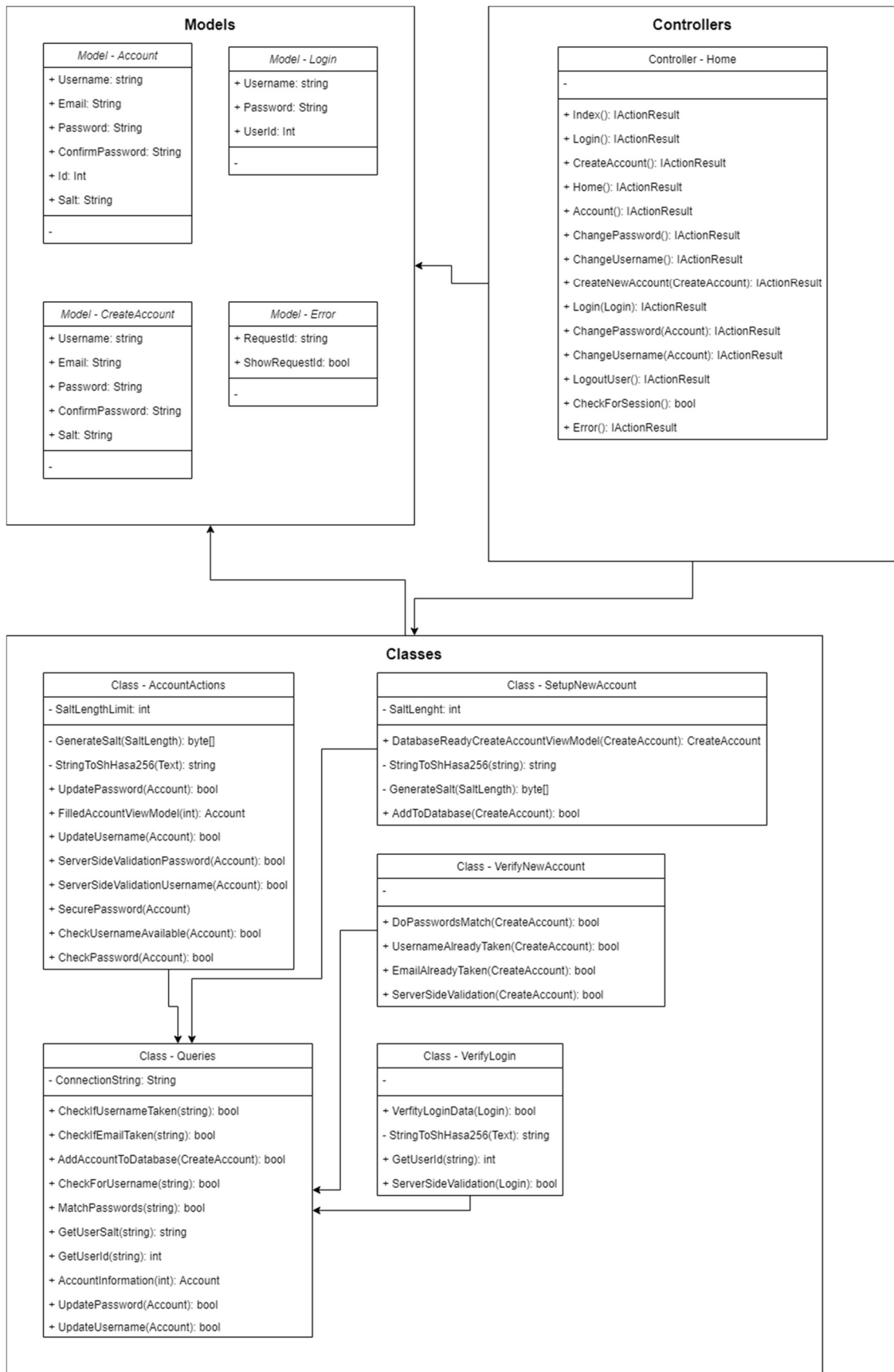
## RELATIONEEL DATAMODEL

Users
<u>PK Id (INT)</u>
[ ] Username (VARCHAR, 16)
[ ] Email (NVARCHAR, 320)
[ ] Password (CHAR, 64)
[ ] Salt (CHAR, 64)

## DATADITIONARY

Naam entiteit	Users	
Definitie	Hier staat de algemene informatie van de gebruikers	
Attribuutnaam	Datatype	Toelichting
Id	Int	Primary key, Identity
Username	VarChar	16
Email	NVarChar	320
Password	Char	64
Salt	Char	64
Relatie	Toelichting	
R1	...	
R2	...	

# CLASS DIAGRAM



# ONTWIKKELOMGEVING

In dit gedeelte staat beschrijven welke hard- software is gebruikt om de ontwikkel- en test omgeving in te richten.

## HARDWARE

Onderstaande hardware is nodig om de applicatie te kunnen maken:

Apparaat	Vereisten
Computer	

## SOFTWARE

*Hieronder staat de software die gebruikt is om de applicatie in de vorm van een webapplicatie te bouwen:*

<i>Software</i>	<i>Opmerkingen</i>
<i>Besturingssysteem</i>	
<i>IDE</i>	
<i>Ondersteunende frameworks en servers</i>	
<i>Database</i>	
<i>Browsers</i>	
<i>Pentesting</i>	



# TESTPLAN

## TESTVOORBEREIDING

Het testplan bestaat uit een gegevensset en een lijst met te testen onderdelen van de applicatie. Zorg ook voor een gegevensset voor het toevoegen en het wijzigen van gegevens. De gegevensset bestaat uit de entiteiten uit de ERD.

Maak gebruik van onderstaande tabel om een gegevensset te maken.

Tabelnaam 1			
Entiteit 1	Entiteit 2	Entiteit 3	Entiteit 4

Tabelnaam 2			
Entiteit 1	Entiteit 2	Entiteit 3	Entiteit 4

Maak gebruik van de onderstaande tabel om te bepalen welke onderdelen van de applicatie getest moeten worden.

Onderdeel	Formulier	Functie	Afwijkend paginaontwerp
	JA/NEE	Beheer van de toegangsrechten	
	JA/NEE		
	JA/NEE		
	JA/NEE		
	JA/Nee		

## TESTLOG

Het tweede deel van het testrapport bestaat uit een testlog. Maak gebruik van onderstaande tabel bij het testen van de onderdelen van de applicatie.

Testlog							
Pagina	Datum test	Tester	Defect	Prioriteit	Verbeteractie	Datum actie	Afgehandeld door

De prioriteit van problemen wordt door middel van een getal aangeduid.

0 = Geen prioriteit

1 = Lage prioriteit voor een probleem waar niet meteen een oplossing voor hoeft te worden gevonden.

2 = Prioriteit voor een probleem dat opgelost dient te worden, maar waar voorlopig mee gewerkt kan worden.

3 = Hoogste prioriteit voor een probleem dat onmiddellijk opgelost dient te worden.

# ACCEPTATIETEST

<b>Functionaliteit</b>	Welke functie uit je functioneel ontwerp ga je testen?				
<b>Actie</b>	Welke actie/handeling moet gedaan worden om de werking van die functie te laten zien?				
<b>Scenario</b>	Wat moet er gebeuren? Wat kan je als gebruiker verwachten dat er gebeurt?				
<b>Verwacht resultaat</b>	Wat zou het resultaat hiervan moeten zijn?				
<b>Werkelijk resultaat</b>	Wat was tijdens de acceptatietest het werkelijke resultaat?				
<b>Aanpassingen</b>	Zijn er, en zo ja, welke aanpassingen zijn er nog nodig aan deze functionaliteit?				
<b>Uitvoering</b>	<b>Uren</b>		<b>Prioriteit*</b>		<b>Door</b>

<b>Actie</b>	Zijn er nog meer acties, die bij deze functionaliteit horen?				
<b>Scenario</b>					
<b>Verwacht resultaat</b>					
<b>Werkelijk resultaat</b>					
<b>Aanpassingen</b>					
<b>Uitvoering</b>	<b>Uren</b>		<b>Prioriteit*</b>		<b>Door</b>

\*Maak voor de prioriteit gebruik van de volgende waarden:

- 0 = geen
- 1 = laag
- 2 = middel
- 3 = hoog

<b>Functionaliteit</b>						
<b>Actie</b>						
<b>Scenario</b>						
<b>Verwacht resultaat</b>						
<b>Werkelijk resultaat</b>						
<b>Aanpassingen</b>						
<b>Uitvoering</b>	<b>Uren</b>		<b>Prioriteit*</b>		<b>Door</b>	

<b>Actie</b>						
<b>Scenario</b>						
<b>Verwacht resultaat</b>						
<b>Werkelijk resultaat</b>						
<b>Aanpassingen</b>						
<b>Uitvoering</b>	<b>Uren</b>		<b>Prioriteit*</b>		<b>Door</b>	

\*Maak voor de prioriteit gebruik van de volgende waarden:

- 0 = geen
- 1 = laag
- 2 = middel
- 3 = hoog

## TESTPLAN NA AANPASSINGEN.

Maak gebruik van de onderstaande tabel om aan te geven welke aangepaste onderdelen van de applicatie getest moeten worden.

Onderdeel	Formulier	Functie	Afwijkend paginaontwerp
<input type="checkbox"/> <b>Alle overige functies werken na deze aanpassingen.</b>			
	JA/NEE		
	JA/NEE		
	JA/NEE		
	JA/NEE		

# PRODUCTIEOMGEVING

## HARDWARE

Onderstaande hardware is nodig om de applicatie te demonstreren:

Apparaat	Vereisten

## SOFTWARE

*Hieronder staat de software die gebruikt is om de applicatie in de vorm van een webapplicatie te bouwen:*

<i>Software</i>	<i>Beschrijving</i>
<i>Besturingssysteem</i>	
<i>Applicatie</i>	
<i>Database</i>	
<i>Databaseverbinding</i>	

# VERSIEBEHEER

<b>Versienummer</b>	<b>1.00</b>
<b>Releasedatum</b>	
Tickets	
<b>Ontwikkelomgeving</b>	
Broncode	
Database	
Database-verbinding	
Ontwikkeltool	
<b>Productieomgeving</b>	
Applicatie	
Database	
Database-verbinding	
<b>Wijziging per versie</b>	
<b>Versie</b>	<b>Wijziging(en)</b>



# EVALUATIEVERSLAG

## BETROKKENEN

[Noteer welke betrokkenen uit het proces jij om feedback vraagt.]

## EVALUATIE PRODUCT

[Beschrijf de punten die goed zijn gegaan en benoem eventuele verbeterpunten.]

## EVALUATIE PROCES

[Beschrijf de punten die goed zijn gegaan en benoem eventuele verbeterpunten.]

# FUNCTIEOVERZICHT

Beschrijf welke (nieuwe) functie(s) er in de applicatie zitten.

Funcienaam	Vanaf versie	Beschrijving

# GEBRUIKERSHANDLEIDING

Benoem alle functies, de werking ervan en hoe deze functie bereikt kan worden binnen de applicatie.

Functie	
Beschrijving	
Bereikbaar via	
Mogelijke handelingen	
Schermafdruck	
Opmerkingen	

Functie	
Beschrijving	
Bereikbaar via	
Mogelijke handelingen	
Schermafdruck	
Opmerkingen	

Functie	
Beschrijving	
Bereikbaar via	
Mogelijke handelingen	
Schermafdruck	
Opmerkingen	

Functie	
Beschrijving	
Bereikbaar via	
Mogelijke handelingen	
Schermafdruck	
Opmerkingen	