



Solid Innovator

**InfoSupport**

**Nooddossier**  
**Inzicht in medische geschiedenis via declaratiegegevens**



Afstudeerder Joris van de Donk  
Studentnummer 2134012

# AFSTUDEERVERSLAG VOOR FONTYS HOGESCHOOL ICT

## Nooddossier

Inzicht in medische geschiedenis via declaratiegegevens

### Gegevens student

Naam: J.W.P. van de Donk  
Studentnummer: 2134012  
Afstudeerrichting: HBO ICT & Software Engineering (voltijd)  
Afstudeerperiode: 30 januari 2012 t/m 30 juni 2012 (100 werkdagen)

### Gegevens bedrijf

Naam bedrijf: Info Support  
Afdeling: Business Unit Zorg & Verzekeringen  
Plaats: Veenendaal  
Naam bedrijfsbegeleider: I. Diepstraten  
Functie bedrijfsbegeleider: Projectmanager

### Gegevens docentbegeleider

Naam M.J.W. van der Heijden

### Gegevens verslag

Titel afstudeerverslag: Nooddossier: Inzicht in medische geschiedenis via declaratiegegevens  
Datum afstudeerverslag: 7 juni 2012

---

Getekend voor gezien door bedrijfsbegeleider:

Datum

Handtekening

# Voorwoord

Dit afstudeerverslag is geschreven ter afsluiting van mijn opleiding ICT & Software Engineering aan Fontys Hogescholen Eindhoven.

Bij mijn vorige stage maakte ik kennis met de wondere wereld van web development. Sindsdien ben ik er o.a. bij mijn bijbaantje en in mijn vrije tijd erg veel bezig mee geweest. In eerste instantie focuste ik mij voornamelijk op het ontwerpen en realiseren van webservices waar andere ontwikkelaars vervolgens gebruik van zouden kunnen maken. Later ben ik steeds meer gaan ontwikkelen op gebied van grafische user interfaces, zodat ik ook zelf gebruik zou kunnen maken van de verschillende webservices die ik ontwikkeld had.

De opdracht van het Nooddossier sprak mij enorm aan. Het was voor mij namelijk mogelijk om te werken met de verschillende technieken van Microsoft. Ik had daar nog niet erg veel ervaring mee, maar wilde die ervaring wel opbouwen. Bovendien was het mogelijk om te werken met front-end web technieken, waar ik ook graag wat extra ervaring mee zou willen krijgen. Het Nooddossier-project leek dus een ideaal project voor mij; en dat is het uiteindelijk ook gebleken.

Graag zou ik enkele personen willen bedanken voor hun steun tijdens mijn afstudeerperiode. Ten eerste wil ik mijn begeleiding vanuit Info Support bedanken: Ivo Diepstraten, mijn opdrachtgever, Chris van Beek, mijn technisch begeleider, en Marieke Keurntjes, mijn procesbegeleider. Dankzij mijn begeleiders was het voor mij mogelijk om mijn afstudeerproject succesvol af te ronden. Verder wil ik mijn docentbegeleider, Rene van der Heijden, bedanken voor zijn steun vanuit Fontys. Ook wil ik nog twee collega's bedanken: Rik Meijer voor zijn ideeën over user interface frameworks, en Martijn Schuurmans voor zijn programmacode die ik in mijn project kon gebruiken. Als laatste wil ik alle andere Info Support afstudeerders bedanken voor hun feedback.

Joris van de Donk  
Uden, 7 juni 2012

# Inhoudsopgave

Samenvatting.....	5
Summary .....	6
Verklarende woordenlijst.....	7
1. Inleiding .....	9
2. Het bedrijf .....	10
2.1 Algemeen .....	10
2.2 De organisatiestructuur.....	10
2.3 Begeleiding.....	11
3. De Opdracht.....	13
3.1 Aanleiding.....	13
3.2 Probleemstelling .....	13
3.3 Doelstelling .....	15
3.4 Resultaat .....	15
3.5 Effect .....	15
4. Aanpak en planning .....	16
4.1 Ontwikkelmethode .....	16
4.2 Fasering .....	16
4.2.1 Inceptiefase.....	16
4.2.2 Hoofdphase.....	17
4.2.3 Documentatiefase .....	18
4.2.4 Eindfase .....	18
4.3 Tijdsplanning .....	19
5. Uitwerking van de opdracht.....	20
5.1 Inceptiefase .....	20
5.2 Hoofdphase .....	22
5.2.1 Onderzoek .....	22
5.2.2 Realisatie Nooddossier .....	34
5.3 Documentatiefase.....	49
6. Status van het project .....	50
7. Conclusie .....	51
8. Aanbevelingen .....	52

Evaluatie .....	54
Bronnenlijst .....	58
Bijlagen .....	60

## Samenvatting

Wanneer iemand op de spoedeisende hulp belandt, is het van belang dat medisch specialisten snel de medische geschiedenis van deze persoon op kunnen vragen. Wanneer medisch specialisten informatie hebben over bijvoorbeeld het medicijngebruik van een patiënt, kunnen ze hier bij hun behandelingen rekening mee houden. Het landelijk Elektronisch Patiënten Dossier was een initiatief welke gebruikt zou worden door zorgverleners om informatie met elkaar te delen, wat dit probleem op zou kunnen lossen, maar de wet die het Elektronisch Patiënten Dossier zou handhaven is door de Eerste Kamer verworpen.

In geval van een noodsituatie blijft het toch erg belangrijk dat men snel medische informatie van een onbekende patiënt toegankelijk kan maken voor de behandelende specialisten. Het snel zichtbaar maken van relevante medische informatie zou in een echt noodgeval levens kunnen redden.

Volgens Info Support, het bedrijf waar de afstudeerder stage heeft gelopen, zou er in de declaratiegegevens van de zorgverzekeraars veel informatie terug te vinden zijn over de medische geschiedenis van een patiënt. Een medisch nooddossier zou deze declaratiegeschiedenis op eenvoudige wijze voor een medisch specialist inzichtelijk moeten maken.

Info Support wilde graag een proof of concept zien van een medisch Nooddossier, welke gebruik zou maken van de declaratiegeschiedenis die bij de verschillende Nederlandse zorgverzekeraars bekend is. Om de declaratiegeschiedenis van een persoon te achterhalen, zou men eerst moeten weten bij welke zorgverzekeraars deze persoon verzekerd is (geweest). Hiervoor heeft men een Burger Service Nummer of geboortedatum plus achternaam nodig. Social media zou gebruikt kunnen worden om deze geboortedatum op te vragen, zodat men zelfs van een persoon waarvan het Burger Service Nummer onbekend is, een nooddossier zou kunnen weergeven.

De afstudeerder heeft dit probleem aangepakt door eerst een onderzoek te doen naar social media en de manier waarop declaratieverkeer in Nederland verloopt. Vervolgens heeft hij door middel van de ontwikkelmethode Scrum in een aantal iteraties een proof of concept gemaakt van het Nooddossier.

Op dit moment heeft de afstudeerder nog geen social media geïntegreerd in het Nooddossier. Functionaliteit welke gebruik maakt van social media zal geïmplementeerd worden in de periode tussen het inleveren van de scriptie en de uiteindelijke afstudeerzitting.

## Summary

When someone has a medical emergency, it's important that medical practitioners are quickly able to inspect the medical history of the person having the emergency. When a medical practitioner has information about the medicine usage of a patient, they can keep that information in mind when applying a specific treatment. In The Netherlands, a nationwide patient health record ("Elektronisch Patiënten Dossier") was proposed as a solution to this problem, but was shut down by the Senate later on.

Still, it remains extremely important that medical practitioners are able to quickly inspect the medical history of an unknown patient in case of an emergency. This could potentially save lives.

According to Info Support, the company in which the graduation internship took place, a lot of interesting information could be retrieved from the invoices of which insurance companies keep records. A medical emergency health record could present this information in a useful manner to medical practitioners.

Info Support wanted to see a proof of concept of a medical emergency health record, which would make use of the invoice records from the insurance companies. To retrieve these invoices from the insurance companies, some data has to be known in order to uniquely identify a person. The primary way of doing this would be to use the Dutch variation of the social security number, known as the "Burger Service Nummer". If this social security number isn't known, however, it would be possible to use a person's date of birth combined with their family name. If the date of birth of a person with a medical emergency isn't known, Social Media could potentially be used to retrieve it.

The intern has approached this problem by doing some research on social media and invoice traffic in The Netherlands. Once the necessary information had been obtained, the medical emergency health record application was built in a few iterations by making use of the Scrum process.

At the moment, social media hasn't yet been integrated with the medical emergency health record. Functionality which makes use of social media will be implemented in the time period between turning in this thesis and the end of the graduation internship.

## Verklarende woordenlijst

API	Application Programming Interface. Een specificatie van hoe bepaalde software met elkaar kan communiceren. In de context van een webapplicatie beschrijft een zogenaamde “Web-API” de manier waarop een server en een client met elkaar communiceren, en hoe die communicatie verloopt.
ASP.NET MVC	Een techniek van Microsoft waarmee men schaalbare webapplicaties kan ontwikkelen.
Bootstrap	UI-toolkit van Twitter, waarmee men snel webapplicaties kan ontwikkelen. Bootstrap biedt een aantal user interface componenten (buttons, dropdown-boxes, tabs) en CSS stijlen waarmee men een visueel aantrekkelijke en bruikbare applicatie kan bouwen.
COV-check	Controle op Verzekeringsrecht-check. Via de COV-check kan men achterhalen of iemand verzekerd is bij een zorgverzekeraar, bij welke zorgverzekeraars hij in het verleden verzekerd is geweest, en wat voor verzekering hij afgesloten heeft.
CSS	Cascading Style Sheet, een ‘taal’ welke gebruikt wordt om styling van een webpagina te definiëren.
DOM	De Document Object Model is een API voor XML en HTML-documenten. Met behulp van een DOM API kan men HTML en XML-documenten inlezen of wijzigen.
EPD	Elektronisch Patiënten Dossier, een landelijk systeem waarmee zorgverleners gegevens met elkaar zouden kunnen delen. De wet die het Elektronisch Patiënten Dossier zou handhaven is verworpen door de Eerste Kamer.
HTML	HyperText Markup Language, een ‘taal’ waarin men webpagina’s schrijft.
JavaScript	Scripting-taal waarmee men een webpagina dynamisch kan maken. Gestandaardiseerd onder de naam “ECMAScript”.
jQuery	Populaire JavaScript-library voor interactie met de HTML (DOM) van een webpagina.
SNS	Sociale Netwerk Site. Een sociale netwerk site is een website waar gebruikers kunnen communiceren met anderen op een persoonlijk niveau. Meervoud: SNSs.
TSP	Tien Stappen Plan, een gestructureerde methode om een afstudeerproject aan te pakken.
VECOZO	VECOZO is een organisatie, welke “een beveiligd internetportaal [faciliteert] waarin de communicatie tussen zorgaanbieders en zorgverzekeraars zo optimaal mogelijk kan verlopen. In feite vervult VECOZO een soort postbusfunctie. De verzonden en getoonde informatie door de zorgverzekeraars en zorgaanbieders wordt door VECOZO (inhoudelijk) niet gewijzigd”.
Vektis	Vektis is een organisatie die gegevens over de kosten en kwaliteit van de gezondheidszorg in Nederland verzamelt en analyseert. Verder beschikt Vektis over verschillende producten en diensten ter ondersteuning van de elektronische uitwisseling van berichten. Hiervoor ontwikkelen en beheren ze standaarden in samenspraak met zorgverzekeraars, zorgkantoren, zorgaanbieders en softwareleveranciers. Ook ontwikkelt en beheert Vektis diverse referentiesystemen.
WCF	Windows Communication Foundation, een techniek van Microsoft waarmee men webservices kan programmeren.



XSS	Cross Site Scripting, een techniek waarmee een kwaadwillende gebruiker scripts kan toevoegen aan een pagina. Met behulp van XSS kan een kwaadwillende gebruiker belangrijke gegevens stelen, zoals persoonsgegevens of inloggegevens.
-----	---

# 1. Inleiding

Het landelijk Elektronisch Patiëntendossier (EPD) was een voorstel voor een systeem waarbij zorgverleners informatie over patiënten automatisch bij collega's konden opvragen. Dit zou leiden tot verbeterde zorg <sup>(1)</sup> <sup>(2)</sup>; het aantal medische fouten zou zo verminderd worden, en mensen zouden niet meer bij elke zorgverlener hun medische geschiedenis hoeven te vermelden.

De wet die het landelijk EPD echter zou handhaven, is door de Eerste Kamer in april van 2011 verworpen <sup>(3)</sup>. Desalniettemin bleef de behoefte bestaan om het gemakkelijk te maken voor zorgverleners om informatie uit te wisselen.

Vanuit Info Support is het Nooddossier bedacht als afstudeeropdracht en als alternatief op het landelijk EPD. Het Nooddossier is primair bedoeld voor noodsituaties, en haalt patiëntgegevens niet rechtstreeks van de zorgverleners (wat bij het EPD het geval was), maar in plaats daarvan via de zorgverzekeraars. Zorgverleners declareren namelijk allerlei behandelingen bij de verschillende Nederlandse zorgverzekeraars, en dus zou het mogelijk zijn om via de zorgverzekeraars een declaratiegeschiedenis van een persoon op te kunnen vragen waarin deze behandelingen vermeld staan. Om aan enkele benodigde gegevens te komen indien deze onbekend zijn, zou verder gebruikt gemaakt kunnen worden van social media. De geboortedatum van een patiënt is een voorbeeld van een gegeven wat in sommige situaties nodig is om declaratiegegevens op te kunnen vragen.

Deze scriptie beschrijft de manier waarop de afstudeerder het Nooddossier ontwikkeld heeft. De volgende hoofdstukken komen aan bod:

**2. Het bedrijf** – In dit hoofdstuk wordt verdere informatie gegeven over Info Support. De organisatiestructuur zal aan bod komen, maar ook de positie van de afstudeerder binnen het bedrijf. Verder zal toegelicht worden welke begeleiding de afstudeerder vanuit het bedrijf heeft gehad.

**3. De Opdracht** – Hier zal verdere informatie gegeven worden over de opdracht. Onder andere de aanleiding en de doelstelling van het project zal in dit hoofdstuk verduidelijkt worden.

**4. Aanpak en planning** – In dit hoofdstuk zal verder uitgelegd worden hoe de afstudeerder het project aangepakt heeft, in welke fasen het project verdeeld was, en hoe alles concreet was ingepland.

**5. Uitwerking van de opdracht** – Dit hoofdstuk beschrijft op chronologische wijze hoe het project daadwerkelijk aangepakt is. Per fase zullen de werkzaamheden toegelicht worden. Onder andere een onderzoek dat uitgevoerd is om verdere kennis te verkrijgen over het declaratieverkeer en social media zal hier aan bod komen, maar ook de realisatie van het nooddossier komt aan bod.

**6. Status van het project** – Hier zal men kunnen lezen wat de huidige status van het project is: wat er al gedaan is, maar ook wat er nog niet gedaan is.

## 2. Het bedrijf

### 2.1 Algemeen

Info Support is opgericht in 1986 en ontwikkelt software, realiseert business intelligence en integratieoplossingen en draagt zorg voor het beheer en/of hosting van software. <sup>(4)</sup>

Naast dienstverlening voor andere bedrijven, biedt Info Support ruim 300 trainingen aan andere bedrijven, veelal gericht op softwareontwikkelaars.

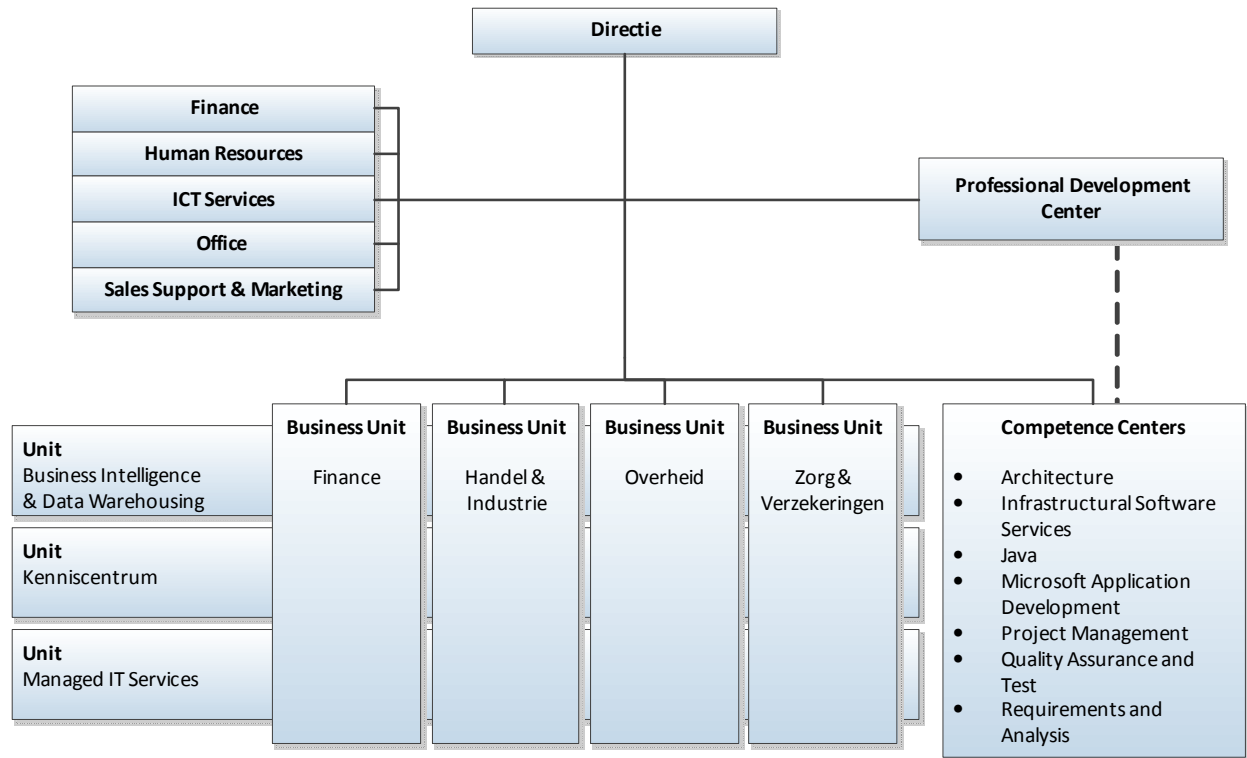
Info Support werkt onder andere samen met VGZ, CZ, VECOZO, NS, Transavia en de provincie Gelderland <sup>(5)</sup>.

Info Support staat voor solide en innovatief. Men is terecht trots op het feit dat men al meer dan 25 jaar zwarte cijfers schrijft, en dat als niet-beursgenoteerd bedrijf. Langetermijnrelaties en kwalitatief hoogwaardige oplossingen zijn belangrijk voor Info Support. Bovendien vindt men het belangrijk om mee te lopen met de ontwikkelingen in de IT. Op deze manier kan men er altijd voor zorgen dat opdrachtgevers aansluiting vinden bij de laatste marktontwikkelingen.

### 2.2 De organisatiestructuur

Info Support is intern onderverdeeld in verschillende *units*. Elke unit specialiseert zich ofwel horizontaal (verbredend binnen een vakgebied) of verticaal (verdiepend in een specifieke markt) <sup>(6)</sup>.

De afstudeerder is onderverdeeld in de Business Unit Zorg & Verzekeringen, aangezien de afstudeeropdracht zich plaats vindt binnen dit domein. Binnen de Business Unit is er veel kennis aanwezig over de zorgsector, en er zijn voldoende collega's die kennis hebben over de systemen waarmee de afstudeerder in aanraking zal komen.



figuur 1 – Schematische weergave van de organisatiestructuur van Info Support (6)

## 2.3 Begeleiding

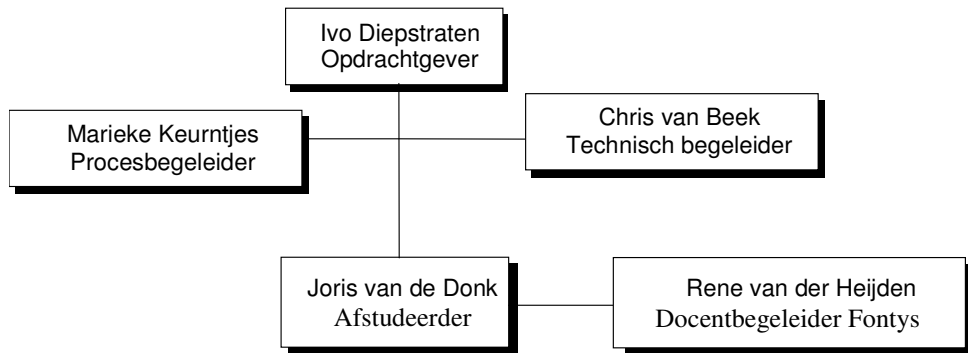
Vanuit Info Support krijgt de afstudeerder drie begeleiders toegewezen: een opdrachtgever, een technisch begeleider, en een procesbegeleider.

De *opdrachtgever* is de persoon die verantwoordelijk is voor de bewaking van de opdracht. Hij heeft de opdracht opgesteld en zal bepalen of de uitwerking van de opdracht voldoende is. De opdrachtgever is ook verantwoordelijk voor het bepalen van de requirements, en het stellen van prioriteiten.

De *technisch begeleider* kan geraadpleegd worden indien de afstudeerder technische ondersteuning nodig heeft bij de uitwerking van zijn opdracht. Bovendien kan de technisch begeleider om advies gevraagd worden.

De *procesbegeleider* kan geraadpleegd worden voor algemene vragen over het bedrijf of het afstudeerproces. Verder zal de procesbegeleider periodiek met de afstudeerder een vergadering hebben om de voortgang van het afstuderen te bespreken, en zal de procesbegeleider de afstudeerder wegwijs maken binnen het bedrijf.

In figuur 2 staat schematisch aangegeven wie de begeleiders vanuit Info Support zijn, en wie de docentbegeleider van Fontys is.



figuur 2 – Schematische weergave van de positie van de afstudeerder binnen het bedrijf

## 3. De Opdracht

### 3.1 Aanleiding

Wanneer iemand op de spoedeisende hulp beland, is het van belang dat men snel de medische geschiedenis van deze persoon op kan vragen. Wanneer medisch specialisten informatie hebben over bijvoorbeeld het medicijngebruik van een patiënt, kunnen ze hier bij hun behandelingen rekening mee houden.

Het landelijk Elektronisch Patiënten Dossier (EPD) zou tijdens een noodsituatie gebruikt kunnen worden door medisch specialisten om de medische geschiedenis van de patiënt op te vragen. De wet die het EPD zou handhaven is echter door de Eerste Kamer verworpen. Toch blijft de noodzaak bestaan om in geval van een noodsituatie medische informatie van een onbekende patiënt toegankelijk te maken voor de behandelende specialisten. Het snel zichtbaar maken van relevante medische informatie zou in een echt noodgeval levens kunnen redden, voornamelijk wanneer bij een bepaalde patiënt een specifieke behandeling levensbedreigend kan zijn.

Er is in de declaratiegegevens van de zorgverzekeraars veel informatie terug te vinden over de medische geschiedenis van een patiënt. Een medisch nooddossier zou deze declaratiegeschiedenis op eenvoudige wijze voor een medisch specialist inzichtelijk moeten maken.

Info Support zou graag een proof of concept willen zien van een medisch nooddossier welke gebruik maakt van de declaratiegegevens van zorgverzekeraars, zodat ze weten dat het concept werkt en wellicht doorontwikkeld kan gaan worden. Dit nooddossier zal een webapplicatie moeten worden. Verder willen ze weten met wat voor problemen men te maken gaat krijgen indien men in de toekomst een nooddossier-achtige service aan wil bieden aan derden.

### 3.2 Probleemstelling

Om van een persoon een nooddossier op te kunnen stellen, zal men bij alle verzekeraars waar de persoon bij verzekerd is (geweest) informatie moeten opvragen. Voordat men dat kan doen, moet men dus eerst weten bij welke zorgverzekeraars iemand verzekerd is geweest. Dit gebeurt via de *Controle op Verzekeringsrecht-service* van VECOZO. VECOZO is een organisatie welke “een beveiligd internetportaal [faciliteert] waarin de communicatie tussen zorgaanbieders en zorgverzekeraars zo optimaal mogelijk kan verlopen” <sup>(7)</sup>. Het gebruiken van de *Controle op Verzekeringsrecht-service* om de zorgverzekeraars van een persoon op te halen noemt men ook wel een COV-check.

Om een COV-check te kunnen doen, heeft men enkele persoonsgegevens nodig <sup>(8)</sup>. In figuur 3 staat weergegeven wat voor gegevens men kan gebruiken.

Optie	Benodigde persoonsgegevens voor COV-check
1	Burger Service Nummer (BSN nummer)
2	polisnummer zorgverzekeraar + geboortedatum
3	postcode + huisnummer + geboortedatum
4	achternaam + geboortedatum

figuur 3

Wanneer men geen BSN nummer weet, moet men dus de geboortedatum weten en het adres, een polisnummer of de achternaam van een persoon.

Via *social media* kunnen mensen tegenwoordig dit soort gegevens beschikbaar maken aan de buitenwereld. Het is de vraag of men verschillende sociale netwerken kan doorzoeken om de geboortedatum van een persoon op te halen. Op deze manier zou men in allerlei gevallen tóch kunnen achterhalen bij welke zorgverzekeraars een persoon verzekerd is geweest. Deze informatie kan men vervolgens gebruiken om een nooddossier op te stellen over een patiënt.

Het nooddossier zelf zal opgesteld moeten worden aan de hand van de declaratiegegevens die bij zorgverzekeraars bekend zijn. Deze declaratiegegevens worden door zorgverleners aangemeld bij de zorgverzekeraars volgens specifieke standaarden: de EI-standaarden. Elke beroepsgroep heeft zijn eigen EI-standaard voor het aanmelden van declaraties. Op dit moment bieden zorgverzekeraars echter geen service aan die andere partijen kunnen gebruiken om declaratiegegevens bij de zorgverzekeraars op te vragen. Deze functionaliteit is echter wel nodig voor de werking van een nooddossier op basis van declaratiegegevens.

### 3.3 Doelstelling

Er zijn twee doelstellingen:

*Doelstelling 1: Wensen van de opdrachtgever vervullen*

De opdrachtgever zou graag een *proof of concept* webapplicatie willen zien van een nooddossier waarmee medisch specialisten snel de declaratiegeschiedenis van een patiënt kunnen zien op basis van enkele persoonsgegevens, waarvan sommige via social media opgevraagd kunnen worden.

*Doelstelling 2: Afstuderen*

Met dit project tracht de afstudeerder aan te tonen aan Fontys Hogescholen te Eindhoven dat hij diplomawaardig is voor de opleiding ICT & Software Engineering.

### 3.4 Resultaat

- 1) Het Nooddossier wat aan de wensen en eisen van de opdrachtgever voldoet
- 2) Succesvol afstuderen van de afstudeer kandidaat

### 3.5 Effect

Een proof of concept van het Nooddossier wordt ontwikkeld welke aantoont dat het mogelijk is om op basis van verschillende persoonsgegevens een declaratiegeschiedenis van een persoon te verkrijgen, en dat men deze persoonsgegevens in sommige gevallen via social media kan verkrijgen.



## 4. Aanpak en planning

Vóór de ontwikkeling van het Nooddossier project is er een projectaanpak vastgesteld. De projectaanpak bestaat uit een keuze voor een ontwikkelmethode, de fasering die toegepast is op het gehele project, en een concrete tijdsplanning. Dit hoofdstuk beschrijft hoe het project aangepakt is. Voor meer informatie kan men de bijlage “Plan van Aanpak” lezen.

### 4.1 Ontwikkelmethode

Als ontwikkelmethode is Scrum gehanteerd. Scrum is een iteratief management-framework dat vaak gebruikt wordt om op een gestructureerde manier software te ontwikkelen <sup>(9)</sup>. Scrum is een iteratieve methode waarbij er steeds periodiek met de opdrachtgever een korte planning (voor 2 weken bijvoorbeeld) gemaakt wordt aan de hand van een grotere globalere planning. Aan de hand van deze korte planning wordt er vervolgens gewerkt aan een volgende versie van een product.

De voornaamste reden om Scrum te gebruiken, was dat de afstudeerder er al wat ervaring mee had vanuit zijn opleiding en werkervaring. Verder zou in de toekomst waarschijnlijk ook met Scrum gewerkt worden, dus zou het verstandig zijn om extra aandacht te besteden aan het Scrum-proces om er zo nog beter in te worden. Bovendien zou de opdracht zeer effectief in een aantal vaste iteraties uitgewerkt kunnen worden, omdat de opdracht in feite bestaat uit een aantal deelopdrachten. Als laatste vond men het verstandig om periodiek met de opdrachtgever de voortgang te bespreken en samen te bepalen waar men een periode aan zou gaan werken. Scrum biedt daarvoor een ideaal framework.

Scrum wordt over het algemeen gebruikt in teams van 5 tot 7 personen. Eén van de kenmerken van Scrum is dat men een dagelijkse meeting (de daily stand-up) houdt waarbij teamleden elkaar informeren over de stand van zaken. Aangezien de afstudeerder het enige projectlid is van het Nooddossier project, zal er geen dagelijkse meeting gehouden worden.

### 4.2 Fasering

Om het project voorspoedig te laten verlopen, zal het project ingedeeld worden in een aantal fasen. In elke fase zullen er een aantal taken verricht worden, en zullen er specifieke producten opgeleverd worden.

#### 4.2.1 Inceptiefase

Tijdens de inceptiefase zal er gewerkt worden aan het verder verduidelijken van de opdracht. De requirements zullen vastgesteld worden, en er zal gewerkt worden aan het Plan van Aanpak. Het doel van deze fase is de verduidelijking van het project en de manier waarop het project aangepakt zal worden.

Verder zullen er tijdens de inceptiefase concrete onderzoeksvragen opgesteld worden, die allemaal in de hoofdfase beantwoord dienen te worden.

De inceptiefase zal ongeveer 4 weken duren, en eindigt wanneer het plan van aanpak geaccepteerd wordt door de opdrachtgever, technisch begeleider en docentbegeleider.

#### 4.2.2 Hoofdfase

De hoofdfase is de fase in het project waar concreet volgens Scrum gewerkt wordt. In de hoofdfase zullen 6 sprints doorlopen worden van elk twee weken, welke gebruikt zullen worden om een proof of concept van het nooddossier te realiseren.

De hoofdfase zal voornamelijk gebruikt worden om backlog-items uit te voeren. Een backlog-item is een taak die uitgevoerd dient te worden met als uitgangspunt de realisatie van het eindproduct. Alle backlog-items samen vormen de backlog. De backlog kan ten alle tijden aangepast worden door de afstudeerder of opdrachtgever. Op deze manier kan men flexibel werken met veranderingen in de gestelde eisen of veranderingen aan een eerder vastgelegd ontwerp. Wanneer men de backlog aanpast door extra backlog-items toe te voegen, zal men echter wel rekening moeten houden met het feit dat daardoor andere backlog-items eventueel niet voldaan kunnen worden vanwege de beschikbare tijd. Vandaar dat het wijzigen van de backlog eigenlijk iets is waar de afstudeerder en de opdrachtgever samen toestemming voor moeten geven, aangezien ze beiden een stakeholder zijn van het algehele project.

Aan het begin van elke sprint zal er een sprintreview- en sprintplanning-sessie gehouden worden waarbij de afstudeerder en de opdrachtgever betrokken en aanwezig zijn. Tijdens deze sessie zullen de afstudeerder en opdrachtgever terugblikken op de vorige sprint, en bepalen wat er in de komende sprint gedaan gaat worden. Om te bepalen wat er in de volgende sprint gedaan wordt, kijkt men onder andere naar de snelheid van werken in de vorige sprints (de *velocity*), en de te verwachten hoeveelheid werk. Op deze manier zal men steeds accurater kunnen plannen.

Tijdens de hoofdfase zal voornamelijk gewerkt worden aan een onderzoek en aan het realiseren van een proof of concept van het nooddossier.

De hoofdfase heeft een aantal eindproducten, te weten:

- 1) Antwoorden op de onderzoeksvragen.
- 2) Proof of concept nooddossier (broncode en executables)
- 3) Ontwerpdocumentatie

### **4.2.3 Documentatiefase**

In de documentatiefase zal er een afstudeerscriptie geschreven worden. De documentatiefase loopt gedeeltelijk parallel met de hoofdfase, zodat de informatie in de afstudeerscriptie geschreven wordt op het moment dat de afstudeerder bezig is met datgene dat hij aan het documenteren is.

Het resultaat van de documentatiefase is een afstudeerscriptie.

### **4.2.4 Eindfase**

De eindfase omvat het eind van het afstudeerproject. In deze fase zal een presentatie opgesteld worden welke tijdens de afstudeerzitting gepresenteerd zal worden. Deze presentatie zal als proef intern gegeven worden, zodat er feedback ingewonnen kan worden. Verder zal de afstudeerder verschillende evaluatieformulieren met betrekking tot de afstudeerstage invullen. De eindfase eindigt met een afstudeerzitting op Fontys Hogescholen te Eindhoven. Deze afstudeerzitting betekent bovendien het einde van de afstudeerperiode.

## 4.3 Tijdsplanning

Van de verschillende fasen is er een tijdsplanning gemaakt. Deze kan men vinden in figuur 4. Vanuit Fontys Hogescholen word er gesteld dat het project volgens het zogenaamde Tien Stappen Plan (TSP) ingedeeld zou moeten worden. De tien stappen van het TSP zijn bedoeld voor het aanpakken van een afstudeeropdracht op een gestructureerde wijze. De afstudeerder, opdrachtgever en docentbegeleider hebben samen besloten om het Scrum-proces als leidraad te gebruiken, maar toch het TSP in gedachte te houden tijdens het gehele afstuderen. Er staat in figuur 4 daarom een kolom welke beschrijft op wat voor manier het project ingedeeld is volgens TSP.

Weeknr	Project week	Projectfase of onderdeel			Projectfase volgens TSP	Overige activiteiten	
5	1	Inceptiefase		Hoofdfase Documentatiefase	Oriënterende activiteiten		
6	2				Analyse		
7	3					-TechDays (2 dagen)	
8	4				Terugkoppeling		
9	5	Sprint 1	Ontwikkeling		Werkplanning en projectorganisatie	-cursus planning/documentatie (3 dagen)	
10	6	Sprint 2			Terugkoppeling	-bedrijfsbezoek 1	
11	7				Werkplanning en projectorganisatie		
12	8					-ADCSB Training (2 dagen)	
13	9	Sprint 3			Werkplanning en projectorganisatie		
14	10	Sprint 4					
15	11				Werkplanning en projectorganisatie		
16	12						
17	13	Sprint 5			Werkplanning en projectorganisatie		
18	14	Sprint 6					
19	15				Werkplanning en projectorganisatie		
20	16						
21	17	Uitloop			Oplossing		
22	18	Documentatiefase			Afronden en afstuderen	-scriptie inleveren	
23	19				Overdracht/invoering	-bedrijfsbezoek 2	
24	20	Eindfase	Eventuele laatste uitloop i.v.m. gemiste dagen				
25	21						
26	22				-Afstudeerzitting		

figuur 4

## 5. Uitwerking van de opdracht

Dit hoofdstuk beschrijft de manier waarop de afstudeerder de opdracht uitgewerkt heeft. Per fase wordt in dit hoofdstuk gekeken naar de activiteiten van die fase. Indien nodig worden de activiteiten verder toegelicht.

### 5.1 Inceptiefase

Tijdens de inceptiefase heeft men gewerkt aan het Plan van Aanpak. De opdracht is samen met de opdrachtgever verder verduidelijkt, en men heeft samen bepaald op welke dingen de afstudeerder zich gaat richten.

In de inceptiefase heeft men bijvoorbeeld besloten dat de security en audit trailing aspecten, welke in eerste instantie tot de opdracht behoorden, buiten de scope zouden vallen. De afstudeerder en opdrachtgever zijn samen tot deze beslissing gekomen.

Security is uiteraard erg belangrijk voor een project waarbij persoonsgegevens en medische gegevens gebruikt worden. Security is echter ook iets waarbij men zeer goed moet oppassen dat men alles volledig begrijpt en niet onverwachts een aantal steken laat liggen; één fout kan een lek veroorzaken die alle beveiligingsmaatregelen omzeilt. Bovendien kost het goed implementeren van beveiligingsmaatregelen aardig wat tijd, onder andere dankzij veiligheidseisen als NEN-7510 waar men aan moet voldoen, wat zou betekenen dat de afstudeerder zich niet kan richten op de relevantere en interessantere aspecten van het project, zoals de koppeling met Social Media en de extractie van gegevens uit de declaratiebestanden. Dat was de voornaamste reden dat men besloten heeft om geen aandacht te besteden aan het volledig veilig maken van het Nooddossier. Het Nooddossier zou bovendien slechts een proof of concept zijn, er was nooit sprake van het in productie nemen van het product. Als dat wel het geval zou zijn, was men uiteraard nooit tot de beslissing gekomen om security buiten de scope te laten vallen.

Audit trailing is ook een feature die zeer handig is, vooral om misbruik te kunnen detecteren, maar men heeft toch besloten om deze feature buiten de scope te laten vallen vanwege een mogelijk tekort aan tijd. Deze feature is afhankelijk van het bestaan van een werkend Nooddossier systeem, en zou pas tegen het eind van het project geïmplementeerd kunnen worden, vandaar dat er gekozen is om juist deze feature buiten de scope te laten vallen.

Ook heeft men tijdens de inceptiefase een aantal risico's ingeschat van het Nooddossier project. Eén van de risico's van het project was dat de afstudeerder niet zo veel ervaring had met C# en .NET. Samen met de technisch begeleider heeft de afstudeerder hier een aantal interne trainingen voor geselecteerd, en hebben ze bepaald wat voor studie men verder zelf nog zou kunnen doen om toch de benodigde kennis te vergaren.

Verder heeft de afstudeerder afspraken gemaakt met zijn begeleiders. Zo is afgesproken dat er wekelijks een voortgangsmail gestuurd zal worden naar de technisch begeleider, opdrachtgever, en docentbegeleider. Verder heeft men afgesproken dat de afstudeerder eens in de 2-3 weken samen met de procesbegeleider een gesprek zou hebben over het verloop van de afstudeerstage, en heeft men afgesproken dat eens in de 2 weken de opdrachtgever en afstudeerder een sprint review- & planning-sessie zouden houden.

Als laatste is de afstudeerder nog twee dagen naar de Microsoft TechDays geweest in Den Haag. Op de TechDays waren er allerlei interessante presentaties over verschillende

producten van Microsoft. Zo heeft men een presentatie bijgewoond over ASP.NET MVC3, Visual Studio 2010, de combinatie van Visual Studio 2010 en JavaScript, en een aantal presentaties over enkele toekomstige producten als MVC4.

## 5.2 Hoofdfase

Tijdens de hoofdfase van het project heeft men gewerkt aan het uitvoeren van de opdracht. De hoofdfase is gedaan volgens Scrum en grofweg onderverdeeld in twee delen: het onderzoek, en de realisatie van het Nooddossier.

### 5.2.1 Onderzoek

Voordat men begon aan de realisatie van het Nooddossier, heeft men een onderzoek gedaan. Er waren drie redenen om dit onderzoek uit te voeren:

- 1) Fontys Hogescholen stelde als eis aan het afstudeertraject dat er een onderzoek verricht zou worden.
- 2) Om de planning en prioritering van het Nooddossier project goed in te kunnen vullen was aanvullende informatie nodig over de haalbaarheid van verschillende onderdelen binnen het project. Vooral voor de verschillende EI-standaarden, welke horen bij verschillende beroepsgroepen, was het van belang om te kunnen bepalen welke EI-standaard de meest interessante informatie bevatte. Op deze manier zou men kunnen bepalen welke EI-standaard men als eerste zou ondersteunen binnen het Nooddossier.
- 3) Wanneer men de software van het nooddossier zou gaan ontwerpen, zou technische kennis nodig zijn over de verschillende standaarden en APIs die gebruikt zouden worden. Tijdens het onderzoekstraject zou deze kennis grotendeels opgedaan worden.

Het onderzoek had twee specifieke hoofdvragen:

- 1) Is het mogelijk om middels Social Media (Facebook, Hyves, Twitter, LinkedIn, Google Plus) van een persoon de geboortedatum op te halen, wanneer men van deze persoon alleen de naam en het geslacht weet?
- 2) Is het mogelijk om uit de declaratiegegevens van de zorgverzekeraars informatie te halen die voor een medisch specialist interessante informatie bevat?

In het onderzoek naar Social Media werd gekeken in hoeverre het mogelijk is om van een persoon de geboortedatum op te vragen via verschillende sociale netwerk sites (SNSs; enkelvoud: SNS) en wat voor data men nog meer van deze sites op kan halen. Denk hierbij aan foto's, statusupdates, enzovoorts. Ook werd hier gekeken naar de terms of service van de verschillende sociale netwerken, zodat men zou kunnen inschatten of een Nooddossier ook echt in productie zou kunnen draaien zonder dat men problemen zou krijgen met de SNSs.

In het onderzoek naar de declaratiegegevens is in de EI-standaarden gekeken naar informatie die voor medisch specialisten op de spoedeisende hulp interessant of relevant kunnen zijn.

Het onderzoek was iteratief uitgevoerd. De twee hoofdvragen kregen elk een aantal deelvragen, bedoeld om de hoofdvraag te kunnen beantwoorden. Al deze vragen kwamen op de Product Backlog te staan, zodat men het gehele onderzoek volgens het Scrum-proces in kon delen. Indien men tijdens het onderzoek extra vragen tegen zou komen, kwamen deze extra vragen ook weer op de Product Backlog te staan.

De afstudeerder heeft het onderzoek voornamelijk gedaan door het bestuderen van bestaande documentatie. Indien mogelijk heeft men zelf wat programmacode geschreven die het onderzoek gemakkelijk en praktisch maakt, zodat men dingen kon controleren. In feite was het onderzoek dus een combinatie van een literatuuronderzoek en een eigen onderzoek.

### *Social Media onderzoek*

Om de hoofdvraag over social media te kunnen beantwoorden, moesten de eerder opgestelde deelvragen beantwoord worden:

- a) Welke gegevens over gebruikers zijn beschikbaar op de verschillende sociale netwerk sites welke men kan gebruiken voor de COV-check of om een persoon te identificeren?
- b) Met welke invoerdata kan men zoeken naar profielen bij de verschillende sociale netwerken?
- c) Zijn er beperkingen bij het gebruik maken van de APIs behorende bij de sociale netwerk sites? Wat zijn globaal gezien de license agreements? Mag men ongelimiteerd gebruik maken van de APIs?
- d) Wat zijn de default privacy-settings voor de verschillende sociale netwerken? Zijn velden als geboortedatum standaard publiekelijk beschikbaar?
- e) In hoeverre is het mogelijk om gebruikers van verschillende sociale netwerken te 'koppelen' aan elkaar? Zijn er sociale netwerken die deze koppeling zelf ook al doen en beschikbaar maken aan derden?
- f) Wat is de kans dat men op verschillende sociale netwerken van een Nederlandse gebruiker de geboortedatum op kan halen
- g) Hoeveel Nederlanders hebben een account op de verschillende sociale netwerk sites?






Aan de hand van deze vragen is het mogelijk om te bepalen of men via sociale netwerk sites de benodigde gegevens voor een COV-check kan verkrijgen.

Een aantal van de antwoorden op sommige van deze vragen zullen kort toegelicht worden. Voor antwoorden op de andere vragen, bijvoorbeeld op vraag c, zie de bijlage "Onderzoeksresultaten".

### **Beschikbaarheid gegevens op sociale netwerk sites**



In figuur 5 staat per SNS weergegeven in hoeverre specifieke gegevens theoretisch gezien beschikbaar zijn via publieke APIs. Hier wordt geen rekening gehouden met privacy-instellingen.

	Achternaam	Geslacht	Geboortedatum	Postcode + huisnummer	Woonplaats	Profielfoto	Foto's	Werkgeschiedenis
 <b>Facebook</b>								
 <b>Hyves</b>								
 <b>Twitter</b>								
 <b>LinkedIn</b>								
 <b>Google Plus</b>								

#### Legenda

	Gegevens beschikbaar via API
	Gegevens niet beschikbaar via API
	Gegevens (eventueel) afleidbaar van een ander gegevensveld, of het formaat waarin de gegevens opgehaald kunnen worden is niet consistent of volledig genoeg.

figuur 5

Deze gegevens zijn opgesteld aan de hand van API-documentatie van de verschillende SNSs. Ter controle zijn na het lezen van de documentatie calls gemaakt naar de verschillende APIs, om te kijken of de documentatie wel klopt.

De geboortedatum van een persoon is alleen via via Facebook, Hyves en LinkedIn te achterhalen. De combinatie Facebook, Hyves en LinkedIn lijkt verreweg de sterkste combinatie te zijn waarmee men de meeste gegevens te weten kan komen.

De API van Google Plus claimt dat het mogelijk is om de geboortedatum van een persoon op te halen, maar wanneer men dat daadwerkelijk probeert zal men nooit een volledige geboortedatum als resultaat krijgen. Het lijkt er op dat er hier een fout zit in de API-documentatie of -implementatie van Google Plus. Wanneer iemand zijn eigen Google Plus profiel via de website bekijkt, zal men nergens zijn eigen geboortedatum kunnen vinden, dus het lijkt er op dat Google Plus deze gegevens nergens toonbaar maakt. Overigens is het wel verplicht om een geboortedatum in te vullen wanneer men een Google account aanmaakt <sup>(10)</sup>, omdat Google zijn services moet af kunnen schermen voor minderjarigen.






Voor meer informatie, zie de bijlage "Onderzoeksresultaten".

#### Standaard privacy-instellingen

De standaard privacy-instellingen van een sociaal netwerk zijn een goede indicatie over wat voor instellingen men kan verwachten bij een groot gedeelte van de gebruikers van het sociale netwerk in kwestie. Privacy-instellingen kunnen erg complex en onoverzichtelijk zijn,

vooral voor gebruikers die de web-interface van de SNS lastig vinden. Aan de hand van de standaard privacy instellingen krijgt men dus inzicht in de gegevens men waarschijnlijk realistisch gezien op kan vragen bij de verschillende SNSs.

In figuur 6 kan men zien wat de standaard privacy-instellingen zijn bij verschillende SNSs voor bepaalde gegevens. Deze gegevens zijn verkregen middels het aanmaken van een account en het handmatig kijken naar de instellingen. Verder zijn er een aantal externe bronnen geraadpleegd voor verificatie.

	Achternaam	Geslacht	Geboortedatum	Postcode + huisnummer	Woonplaats	Profielfoto	Foto's	Werkgeschiedenis
 <b>Facebook</b>	A	A	C	E	A	A	A	A
 <b>Hyves<sup>1</sup></b>	B	B	B	E	B	A	D	B
 <b>Twitter</b>	A	E	E	E	A	A	E	E
 <b>LinkedIn</b>	A	E	D	E	A	A	E	A
 <b>Google Plus</b>	A	A	E	E	A	A	D	A

#### Legenda

A	<b>Publiekelijk:</b> Zichtbaar voor iedereen
B	<b>Publiekelijk:</b> Zichtbaar voor alle ingelogde gebruikers
C	<b>Private:</b> Zichtbaar voor 2 <sup>e</sup> graads connecties (vrienden en vrienden van vrienden)
D	<b>Private:</b> Zichtbaar voor 1 <sup>e</sup> graads connecties (vrienden)
E	<b>Private:</b> Niet zichtbaar of niet beschikbaar

figuur 6

#### Zoekmogelijkheden

Op alle SNSs kan men zoeken naar voor en/of achternaam <sup>(11) (12) (13) (14) (15)</sup>. Bij Hyves en LinkedIn kan men bovendien zoeken naar woonplaats. LinkedIn biedt als extra zoekopties het zoeken naar postcode en werkgever.

De manier waarop via Hyves gezocht wordt kan echter voor problemen zorgen. Via Hyves zoekt men namelijk niet specifiek op voor- of achternaam. In plaats daarvan zoekt men op

<sup>1</sup> Let wel: op Hyves is de zichtbaarheid van alle gegevens voor mensen van 15 jaar of jonger standaard ingesteld op "alleen vrienden" <sup>(37)</sup>. De zichtbaarheid kan echter nog wel aangepast worden door de gebruiker.






voornaam, achternaam, bijnaam ('nickname') en plaatsnaam tegelijkertijd. Dit houdt in dat wanneer men zoekt op "Uden", men zowel mensen uit Uden vindt, of mensen die bijvoorbeeld "van Uden" heten met hun achternaam.

#### Percentage gebruikers waarvan men de geboortedatum bij een SNS op kan halen

De kans dat men van een gebruiker de geboortedatum bij een SNS op kan halen, is belangrijk voor het Nooddossier project. Met behulp van deze informatie kan men bepalen of men überhaupt SNSs gaat gebruiken om geboortedatums op te zoeken.

Om deze kans te bepalen, is een praktisch onderzoek uitgevoerd. Van elk van de sociale netwerk sites welke een geboortedatum via de APIs aanbieden (Facebook, Hyves, LinkedIn, Google Plus<sup>2</sup>), is via de APIs gezocht naar gebruikers met als achternaam "de Boer". Vervolgens is gekeken naar het aantal mensen waarvan men de volledige geboortedatum kon achterhalen. Indien men voor een SNS een significant aantal mensen vond waarvan dit mogelijk was (>5%), werd de grootte van de dataset vergroot om tot een accurater getal te komen, en zijn extra maatregelen genomen om niet-Nederlanders uit de dataset te filteren.

In figuur 7 staan de resultaten van dit korte onderzoek weergegeven. Voor gedetailleerde informatie over hoe men aan deze data gekomen is, zie de bijlage "Onderzoeksresultaten".

SNS	Percentage Nederlandse gebruikers waarvan men, indien ingelogd, de geboortedatum kan opvragen	Grootte dataset (aantal mensen)
 Facebook	<1%	125
 Hyves	78%	4953
 Twitter	n.v.t.	n.v.t.
 LinkedIn	<1%	220
 Google Plus	0%	373


figuur 7

Deze gegevens kan men waarschijnlijk verklaren door te kijken naar de standaard privacy-instellingen van de verschillende SNSs. Facebook staat standaard zó ingesteld dat alleen vrienden toegang hebben tot de geboortedatum. LinkedIn staat standaard zó ingesteld dat alleen directe connecties toegang hebben tot de geboortedatum. Hyves, echter, staat voor mensen ouder dan 15 jaar zó ingesteld dat alle andere hyvers (ingelogde gebruikers) toegang hebben tot de geboortedatum. Voor mensen van 15 of jonger staat de geboortedatum op Hyves standaard op "beschikbaar voor vrienden". Het lijkt er op dat mensen niet snel hun privacy-settings aanpassen, en dat men daarom op Hyves van 78% van de mensen een geboortedatum kan opvragen.

#### Conclusie






<sup>2</sup> Google Plus is óók in dit onderzoek opgenomen, ter verificatie van de eerdere bewering dat de API-documentatie niet lijkt te kloppen.

Om de hoofdvraag te kunnen beantwoorden, moeten we per SNS weten hoeveel gebruikers ze in Nederland hebben. In figuur 8 staat weergegeven per SNS hoeveel gebruikers ze hebben.

SNS	Aantal gebruikers in Nederland
 <b>Facebook</b>	6.263580 miljoen gebruikers <sup>(16)</sup>
 <b>Hyves</b>	9.7 miljoen gebruikers <sup>(17)</sup>
 <b>Twitter</b>	5 miljoen gebruikers (ongeveer) <sup>(18)</sup>
 <b>LinkedIn</b>	3 miljoen gebruikers (meer dan) <sup>(19)</sup>
 <b>Google Plus</b>	Onbekend aantal gebruikers

figuur 8

Wanneer men nu deze cijfers vermenigvuldigt met het percentage gebruikers waarvan men de geboortedatum te weten kan komen, kan men achterhalen van hoeveel mensen men de geboortedatum kan achterhalen:

SNS	Aantal gebruikers waarvan men de geboortedatum kan opvragen	Aandeel van Nederland (16.7 miljoen Nederlanders) <sup>(20)</sup>
 <b>Facebook</b>	Minder dan <b>62636</b> gebruikers	<b>0.38 %</b>
 <b>Hyves</b>	<b>7.57 miljoen</b> gebruikers	<b>45.33 %</b>
 <b>Twitter</b>	<b>0</b> gebruikers	<b>0.00 %</b>
 <b>LinkedIn</b>	Rond <b>30000</b> gebruikers	<b>0.18 %</b>
 <b>Google Plus</b>	<b>0</b> gebruikers	<b>0.00 %</b>

figuur 9

Men kan dus concluderen dat men theoretisch gezien in ongeveer **45%** van de gevallen van een Nederlands persoon de geboortedatum kan achterhalen op een SNS. Dit percentage is sterk afhankelijk van de sociale netwerk site Hyves, aangezien deze SNS de enige is waar men van een significant aantal mensen de geboortedatum op kan vragen. Het daadwerkelijke aantal in de praktijk bruikbare geboortedatum zal lager liggen, vanwege mensen die een verkeerde datum hebben ingevuld of mensen die niet vindbaar zijn op SNSs. Bovendien staat Hyves er op dit moment redelijk slecht voor; ze zijn erg veel gebruikers aan het verliezen aan het populaire Facebook <sup>(21)</sup>. Waarschijnlijk zal het percentage de komende jaren zakken, omdat mensen Hyves niet meer gebruiken en hun accounts opzeggen.

### *EI-Standaarden onderzoek*

Er is ook nog een onderzoek gedaan naar de EI-standaarden. Deze standaarden beschrijven de bestandsformaten welke zorgverleners en zorgverzekeraars gebruiken om onderling te communiceren over declaraties. Deze bestandsformaten zijn voor het Nooddossier relevant, aangezien ze wellicht interessante informatie kunnen bevatten over de behandeling of prestatie die een zorgverlener gedaan heeft bij een cliënt. Kennis over deze bestandsformaten en de inhoud daarvan is dus erg belangrijk.

Om de hoofdvraag over de EI-standaarden te kunnen beantwoorden, moesten de eerder opgestelde deelvragen beantwoord worden:

- a) Welke EI-standaarden zijn er voor declaraties en door wie worden deze gebruikt? Welke EI-standaarden zijn relevant voor het Nooddossier?
- b) Welke data in de relevante EI-standaarden moet geconverteerd worden zodat medisch specialisten ze kunnen begrijpen?

### Verschillende EI-standaarden voor aanmelden van declaraties bij zorgverzekeraars

Voordat men probeerde te achterhalen wat voor gegevens er beschikbaar zijn in de declaraties, was het nodig om te weten wat voor declaratiestandaarden er allemaal zijn, en door wie ze gebruikt worden. Op deze manier zou men inzicht kunnen krijgen in het doel van de verschillende standaarden, en hoe relevant ze zouden zijn voor het Nooddossier.

Er zijn een aantal EI-standaarden voor het aanmelden van declaraties <sup>(22)</sup>. Elke standaard wordt door een andere doelgroep gebruikt.

Doelgroep	Code van Standaard	Versienummer standaard
Apothekers	<a href="#">AP304</a>	7.2
AWBZ	<a href="#">AW319</a>	1.3
Eerstelijns Psychologie	<a href="#">EP301</a>	1.2
GGZ	<a href="#">GZ311</a>	2.0
Huisartsen	<a href="#">HA304</a>	4.2
Kraamzorg	<a href="#">KZ301</a>	3.2
Leveranciers hulpmiddelen	<a href="#">LH307</a>	5.2
Mondzorg	<a href="#">MZ301</a>	1.3
Overige sectoren	<a href="#">OS301</a>	1.0
Paramedici	<a href="#">PM304</a>	3.2
Vervoer	<a href="#">VE303</a>	4.2
Verloskundige hulp	<a href="#">VK301</a>	2.2
Ziekenhuizen	<a href="#">ZH308</a>	8.0

figuur 10 – EI-standaarden en de doelgroep die ze gebruikt

In figuur 10 wordt voor alle doelgroepen weergegeven middels welke EI-standaard zij declaraties aanmelden bij de zorgverzekeraars, en welke versienummers van deze standaarden men onderzocht heeft.

Er is nog een andere standaard beschikbaar; de standaard welke het declaratieverkeer voor de Forensische Zorg beschrijft (FZ301). Deze declaraties gaan echter niet naar zorgverzekeraars, maar naar het Ministerie van Justitie. Zorgverzekeraars zullen deze declaraties dus nooit beschikbaar kunnen stellen aan het Nooddossier. Om deze reden is de FZ301-standaard irrelevant voor het Nooddossier project, en zal deze standaard niet verder bekeken worden.

Het Ministerie van Volksgezondheid, Welzijn en Sport heeft bepaald dat een nieuw declaratiesysteem voor ziekenhuizen in zou moeten gaan op 1 januari 2012 <sup>(23)</sup>. Dit nieuwe declaratiesysteem, DOT geheten, zou een verbetering zijn ten opzichte van het oudere

systeem omdat het gebaseerd is op een internationaal bekend diagnosesysteem. Verder werd het aantal combinaties van diagnose en behandeling ingekort van ruim 30.000 via het oude systeem naar 4.400 via het nieuwe systeem <sup>(24)</sup>. Zorg zou door het invoeren van DOT herkenbaarder en transparanter moeten zijn.

De ZH308-standaard versie 8.0 (ZH308v8) zou ondersteuning bieden voor het nieuwe DOT-systeem. De eerste specificaties van ZH308v8 zijn in Maart van 2011 gepubliceerd. De laatste grote wijziging is in Juli 2011 doorgevoerd.

Ondanks dat ZH308v8 al een tijdje gepubliceerd was, draaide deze berichtstandaard nog niet in productie ten tijde van het onderzoek. Softwareleveranciers en zorgverzekeraars waren op dat moment nog druk bezig met de invoering van deze standaard, en ZH308v7.2 draaide op dat moment nog in productie.

In samenspraak met de opdrachtgever is besloten om ZH308v8 te hanteren binnen het onderzoek, omdat het belangrijk was om een recente versie van de standaard te gebruiken die bedoeld was om declaraties van ziekenhuizen gemakkelijker en duidelijker te maken. Indien men alleen ZH308v7.2 zou ondersteunen in het Nooddossier, zou dat betekenen dat het Nooddossier na de oplevering oude standaarden zou gebruiken.

Op 14 mei 2012 heeft Vektis bekend gemaakt dat ZH308v8 landelijk ingevoerd was <sup>(25)</sup>. Vektis is de organisatie welke verantwoordelijk is voor het beheer van de EI-standaarden. Bovendien voerde Vektis in opdracht van Zorgverzekeraars Nederland een programma uit om o.a. softwareleveranciers en zorgverzekeraars te ondersteunen bij de invoering van het nieuwe declaratiesysteem.

## **Structuur EI-bestanden**

Een belangrijk onderdeel van het Nooddossier is het succesvol kunnen lezen van declaratiebestanden, en deze bestanden kunnen omzetten naar voor medisch specialisten leesbare informatie. Om dit te kunnen doen, is informatie nodig over de manier waarop deze bestanden gestructureerd zijn.

EI-bestanden zijn tekstbestanden gestructureerd volgens het *flat file* principe. Flat files zijn bestanden waarin gegevens sequentieel weggeschreven worden.





## Gegevens-elementen die geconverteerd dienen te worden

Nadat kennis verkregen was over de EI-standaarden en hoe EI-berichten in elkaar zaten, heeft men gekeken naar de gegevens-elementen die geconverteerd dienen te worden om voor eindgebruikers leesbaar te zijn. Op deze manier zou men kunnen achterhalen wat voor data men concreet in het Nooddossier zou kunnen weergeven. Bovendien heeft men op deze manier kunnen kijken naar de technische haalbaarheid van het verlenen van support voor de verschillende standaarden, en naar eventuele complicaties die op zouden kunnen treden tijdens het implementeren van de support voor deze standaarden.

De gegevens-elementen zijn onderverdeeld in verschillende generieke types. Een generiek type beschrijft op welke manier men technisch gezien het gegevens-element in moet lezen. De types kunnen vergeleken worden met de primitieve types in veel programmeertalen. Alle gegevens-elementen hebben elk een unieke naam, welke volgens het formaat “<TYPE><IDNUMMER>-<ORGANISATIE>” beschreven is. In de naam van de gegevens-elementen (bijvoorbeeld “COD392-VEKT”) kan men dus de types herkennen. In figuur 13 kan men zien welke generieke types er gebruikt worden.

Type	Omschrijving
TEC	Waarde met een technische achtergrond die geen betekenis heeft binnen het domein.
ANT	Integer-waarde welke een percentage of aantal aangeeft.
DAT	Datum of tijd-waarde.
BED	Geldbedrag in 1/100 <sup>ste</sup> eenheden (centen).
NAM	Naam of alfanumerieke waarde.
NUM	Waarde welke een nummer voorstelt. Kan ook alfanumeriek zijn (telefoonnummers en dergelijke) of een referentie bevatten naar zorgverlener-specifieke interne data (referentienummers).
COD	Waarde waarvan de betekenis afgeleid moet worden uit een bepaalde datasource (codelijst of database).

figuur 13

Van alle gegevens-elementen behorende bij elk van deze generieke types heeft men gekeken of er iets geconverteerd zou moeten worden om de gegevens voor eindgebruikers begrijpelijk te maken.

Gegevens van het type TEC zijn technisch van aard, en hoeven nooit door mensen uitgelezen te worden.

Gegevens van type ANT hoeven nooit geconverteerd te worden, tenzij ze gebruikt worden om een percentage uit te drukken. In dat geval moet men de integer-waarde delen door 100 om achter het percentage te komen. 90% bijvoorbeeld wordt gecodeerd als “09000”.

Gegevens van type DAT beschrijven óf een dag, of een tijdstip. Deze zullen altijd omgezet moeten worden.



Gegevens van type BED beschrijven een geldbedrag in eurocenten, en moet men dus delen door 100 om het bedrag in euro's te krijgen.

Gegevens van type NAM beschrijven alleen maar alfanumerieke gegevens die al door mensen gelezen kunnen worden, en hoeven dus niet omgezet te worden. Een voorbeeld van een gegeven van type NAM is de voornaam van een persoon.

Gegevens van type NUM beschrijven gegevens die intern bij een zorgverlener gebruikt worden. Deze gegevens zijn al leesbaar en kan men nooit omzetten naar leesbare gegevens, omdat het hier om interne gegevens gaat.

Gegevens van type COD zijn waarden waarvan de betekenis afgeleid moet worden uit een bepaalde datasource; zij het een codelijst of een database. Deze gegevens moeten bijna altijd geconverteerd worden via de datasource; alleen in uitzonderlijke gevallen kan men de informatie direct halen uit de waarde zelf en hoeft men daar geen aparte datasource voor te gebruiken. Gegevensen van het type COD zijn erg interessant voor het Nooddossier. Deze worden o.a. gebruikt om aan te geven of iemand mannelijk of vrouwelijk is, maar ook om een medicijn of een diagnose aan te duiden in een behandeling die bij een persoon hoort.

Sommige gegevenselementen van het type COD maken gebruik van datasources welke door Vektis beheerd worden via publiekelijk beschikbare codelijsten <sup>(27)</sup>. Deze gegevenselementen kan men op een standaardwijze omzetten naar leesbare gegevens, en wanneer men deze gegevenselementen ondersteunt in één EI-standaard ondersteunt men ze ook meteen voor alle andere EI-standaarden. Andere gegevenselementen van type COD maken gebruik van datasources welke op een aparte wijze ondersteund dienen te worden. Deze gegevenselementen zijn enorm interessant voor het Nooddossier, aangezien het hier vaak gaat over gegevens die specifiek door één beroepsgroep gebruikt worden. Diagnoses, gebruikte medicijnen en geleverde hulpmiddelen zijn voorbeelden van dit soort gegevenselementen. Voor elk van deze gegevenselementen moet men aparte code schrijven. Voor meer informatie over deze gegevenselementen, zie de bijlage "Onderzoeksresultaten".

## Conclusie

Het onderzoek naar de EI-standaarden gaf erg veel inzicht in de technische haalbaarheid van de implementatie van de verschillende EI-standaarden. De ZH308-standaard bleek technisch gezien de moeilijkste standaard om ondersteuning voor te bieden. Deze standaard bevat veel gegevenselementen welke gebruik maken van datasources waarvoor men specifieke code moet schrijven om de gegevens leesbaar te maken.

Naast dat er informatie verkregen was over de technische haalbaarheid van de verschillende standaarden, kwam men ook meer te weten over wat er in de declaratiegegevens precies staat. Het was zo mogelijk om te bepalen wat functioneel gezien de meest interessante standaard was. Dit bleek ook de ZH308-standaard te zijn. Hierin staat namelijk beschreven wat voor diagnoses er over een patiënt gesteld zijn, en wat voor behandelingen er gepleegd zijn in een ziekenhuis. AP304, LH307 en HA304 bleken ook erg interessant te zijn,

aangezien deze gaan over medicijnen (AP304), hulpmiddelen (LH307) en huisartsenbezoeken (HA304).

Aan de hand van het onderzoek naar het declaratieverkeer hebben de afstudeerder en opdrachtgever er samen voor gekozen om ZH308 de hoogste prioriteit te geven, aangezien deze standaard zowel functioneel als technisch gezien de meest interessante standaard was.

### 5.2.2 Realisatie Nooddossier

In de 3<sup>e</sup> sprint begon dan echt de uitwerking van het Nooddossier. Het onderzoek had men achter de rug, en het was mogelijk om vanaf dat moment volledig te focussen op het realiseren van het product. Aangezien de afstudeerder een training Advanced C# gevolgd had vóór dat hij daadwerkelijk ging programmeren, was er al veel kennis aanwezig over de ontwikkelomgeving en de verschillende technieken die men zou kunnen gebruiken voor de verdere ontwikkeling.

Aan de hand van de onderzoeksresultaten en de prioriteiten van de opdrachtgever, heeft men bepaald dat de hoogste prioriteit kwam te liggen bij het zichtbaar krijgen van de declaratiegeschiedenis. De integratie met social media zou in een latere sprint gedaan worden.

#### *Web API voor declaratiegegevens*

Als eerste is men een Web API gaan ontwerpen welke het Nooddossier de mogelijkheid zou geven om declaratiegegevens op te halen van een specifieke zorgverzekeraar. Men heeft er voor gekozen om de Web API als eerste te maken, vóórdat men iets aan de front-end van het Nooddossier zou doen, omdat men dacht dat 't het eenvoudigst zou zijn om een front-end te maken aan de hand van al bestaande communicatiekanalen. Een andere mogelijkheid was om als eerste de front-end te maken en vervolgens daar een communicatiekanaal voor te maken welke de data zou aanleveren, maar de afstudeerder dacht dat zijn front-end werk het ontwerp van de Web API te veel zou beïnvloeden.

Deze API zou, indien het Nooddossier in de toekomst daadwerkelijk in productie genomen zou worden, door de zorgverzekeraars geïmplementeerd dienen te worden. Mede daarom heeft men samen met de opdrachtgever een aantal uitgangspunten bepaald voor deze API:

- 1) De API moet gemakkelijk te implementeren zijn voor de zorgverzekeraars, zodat de 'barrier to entry' laag blijft en het gemakkelijker zou zijn om zorgverzekeraars te overtuigen om mee te doen aan het Nooddossier-project.
- 2) Men gaat er van uit dat zorgverzekeraars de rauwe EI-bestanden behorende bij een declaratie ergens opgeslagen hebben. De opdrachtgever heeft aangegeven dat dit hoogstwaarschijnlijk het geval is.
- 3) Als input voor een API-call wordt het BSN nummer van een persoon gebruikt. Het BSN nummer is immers een unieke identificatie van een persoon. Men zoekt dus specifiek op de declaraties van één persoon; zoeken op meerdere personen is niet mogelijk. Het BSN nummer kan men altijd ophalen via de COV-check.

Aan de hand van deze uitgangspunten is men een ontwerp gaan maken voor de Web API voor declaratiegegevens. Tijdens het ontwerpen heeft men een aantal belangrijke beslissingen moeten maken.

De belangrijkste beslissing ging over de verantwoordelijkheid voor het inlezen en omzetten van de EI-bestanden naar leesbare gegevens. Deze omzetting zou óf gedaan moeten worden door de zorgverzekeraars, óf door de server van het Nooddossier.

Nadat de voor- en nadelen van beide mogelijkheden op een rijtje gezet waren, kwam men er achter dat 't het beste zou zijn als de Nooddossier-server de omzetting zou doen. Dit zou namelijk als resultaat hebben dat de Web API de rauwe declaratiebestanden zou kunnen versturen naar het Nooddossier. Zo zou het gemakkelijker zijn voor de zorgverzekeraars om de Web API te implementeren binnen hun eigen systemen, omdat ze nu zelf geen gegevens hoeven om te zetten en ze maar weinig over de EI-standaarden hoeven te weten. Een nadeel van deze beslissing is wel dat zorgverzekeraars de API onmogelijk kunnen implementeren als blijkt dat ze nergens de rauwe EI-bestanden meer hebben.

Bij het ontwerpen van de Web API drong het tot de afstudeerder door dat het mogelijk was dat er meerdere prestaties in één declaratiebestand beschreven stonden. Dit had redelijk grote gevolgen voor het ontwerpproces van de Web API, en was het grootste probleem dat men ondervond tijdens het ontwerp van deze API. Voor meer informatie over dit probleem en de manier waarop men dit opgelost heeft, zie de bijlage “Ontwerp Web-API voor declaratiegegevens”.

Als laatste heeft men een XML Schema Definition (XSD) geschreven welke het retourbericht van de Web API exact definieert. Zo staat er in de XSD beschreven welke gegevens er in het retourbericht staan, en van welk formaat ze moeten zijn. Men kan een XSD gebruiken om te controleren of een bericht zich aan de specificaties houdt, maar XSDs kunnen ook gezien worden als een vorm van documentatie voor mensen die XSDs kunnen lezen. Let wel: de XSDs zeggen alleen iets over het formaat waarin gegevens aangeleverd moeten worden; niet iets over de functionele validiteit van deze gegevens.

Voor meer informatie over de API, en een uitdraai van de XSDs, zie de bijlage “Ontwerp Web-API voor declaratiegegevens”.

Van de Web API heeft de afstudeerder zelf ook een implementatie geschreven. Deze implementatie is geschreven in C#, en maakt gebruik van Windows Communication Foundation (WCF). WCF is een framework voor het schrijven van webservices<sup>(28)</sup>. In de eigen implementatie wordt gebruik gemaakt van dummy-data volgens ZH308 die door de afstudeerder zelf gecreëerd zijn aan de hand van testbestanden van Vektis<sup>(29)</sup>.

## Het Nooddossier

Nadat de Web-API ontworpen en geïmplementeerd was, was het tijd om het Nooddossier te ontwikkelen. Dit heeft men in een aantal stappen gedaan.

### Basisopzet: back-end en front-end.

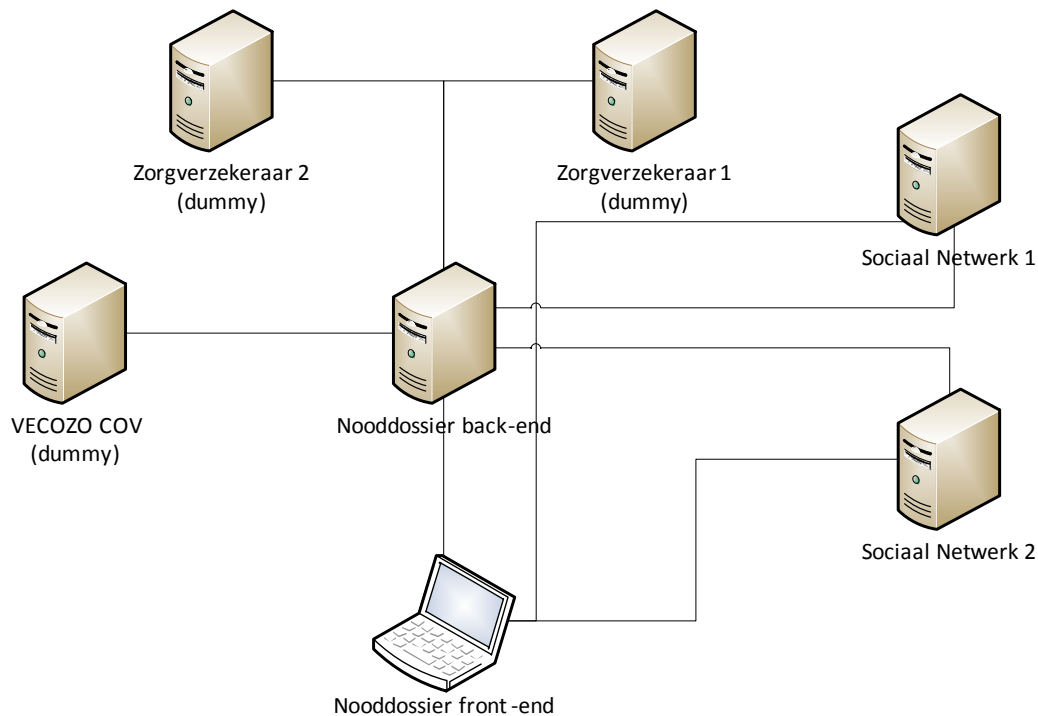
In sprint 3 en sprint 4 heeft men gewerkt aan de basisopzet van het Nooddossier. Vanwege de persoonlijke leerdoelen van de afstudeerder om wat meer met .NET te werken, heeft men ervoor gekozen om het Nooddossier te maken met ASP.NET MVC3 (hierna te noemen MVC3) in combinatie met WCF. MVC3 is een framework om webapplicaties te ontwikkelen bovenop het .NET framework, en WCF kan gebruikt worden om webservices te maken.

Het Nooddossier kan in feite gezien worden als een *back-end*, en een *front-end*. De back-end is het gedeelte van het Nooddossier dat de feitelijke business-logica bevat, en wordt geprogrammeerd in C#. De back-end kan men zien als de server voor de front-end. De front-end is eigenlijk de grafische interface die de gegevens van de back-end op een mooie en bruikbare manier weergeeft. De front-end wordt geprogrammeerd in JavaScript en C#, en maakt verder gebruik van HTML en CSS voor de styling en opbouw.

Voor de communicatie tussen back-end en front-end wordt gebruik gemaakt van REST. REST, wat staat voor “Representational State Transfer”, is een vorm van softwarearchitectuur voor gedistribueerde systemen die vaak gebruikt wordt voor webservices. Door middel van REST kan men werken met data welke gerepresenteerd is middels JSON (JavaScript Object Notation), een compacte objectnotatie die door de meeste JavaScript libraries ondersteund wordt en daarom tegenwoordig vaker gebruikt wordt dan XML. Omdat men met REST kan werken met gedistribueerde systemen zal het in een later traject van het Nooddossier project mogelijk zijn om gemakkelijk ondersteuning te bieden voor een grotere groep eindgebruikers.

De back-end heeft als primaire doel om de communicatie tussen de front-end en de zorgverzekeraars te verzorgen en af te schermen. Bovendien zal de back-end communiceren met de verschillende sociale netwerken en de Controle op Verzekeringsrecht-service van VECOZO. Dankzij de back-end is het mogelijk om de communicatie met de zorgverzekeraar op een veilige wijze te doen, en om de data van de zorgverzekeraars om te zetten naar data die iets leesbaarder is. Bovendien kan de back-end gegevens tijdelijk opslaan, waardoor men deze gegevens sneller aan kan bieden aan de eindgebruikers.

In figuur 14 staat schematisch aangegeven met welke systemen de front-end en back-end communiceren.



**figuur 14 – Schematische weergave van de verbindingen tussen de verschillende systemen.**

In sprint 3 had men al wat belangrijke functionaliteit in de back-end kunnen implementeren: het ophalen van declaraties bij een zorgverzekeraar en het inlezen van deze declaraties was aan het eind van sprint 3 al mogelijk. Deze functionaliteit was redelijk eenvoudig te implementeren omdat men gebruik kon maken van een EI-bestand-parser van een collega, Martijn Schuurmans. Het enige wat men vervolgens hoefde te doen, was het omzetten van de gegevens naar een formaat dat men in de front-end kon lezen. Dit is gedaan via een library, Json.NET. Json.NET kan men gebruiken om data te serializeren naar JSON. Een van de vele <sup>(30)</sup> voordelen van Json.NET ten opzichte van de JSON serializers van .NET zelf, is dat Json.NET veel sneller is. Bovendien is Json.NET veel flexibeler in het gebruik; zo kan men bijvoorbeeld specificeren op welke manier men datums codeert <sup>(31)</sup>.

Toch had men ook een parser voor de EI-bestanden geschreven in JavaScript, om te kijken of dat misschien handiger of sneller was in het gebruik. Dit bleek niet zo te zijn, aangezien men dan de volledige datasources voor gegevens van type COD moest doorsturen naar de front-end of voor elke conversie een call moest maken naar de back-end.

## UI toolkit: Twitter Bootstrap

Toen de afstudeerder begon aan de afstudeeropdracht had hij niet het idee dat hij een UI toolkit zou gaan gebruiken. Men was eigenlijk van plan om de standaard layout en styling van ASP.NET MVC3 te gebruiken. Indien nodig voor de bruikbaarheid van de applicatie zou men kleine aanpassingen maken aan de layout of styling. Aan het eind van de afstudeerstage, na het inleveren van de scriptie, zou men dan nog wat aanpassingen kunnen maken aan de *looks* van het Nooddossier.

Tijdens de onderzoeksfase liet een collega echter Bootstrap<sup>(32)</sup> zien. Bootstrap is een toolkit van Twitter waarmee men het maken van webapplicaties gemakkelijk kan maken. Bootstrap biedt een front-end webdeveloper enkele handige GUI-elementen zodat men vlug aan de slag kan. Verder is Bootstrap voorzien van een consistente style guideline, zodat de webapplicatie er ook nog eens goed uit ziet. Dankzij Bootstrap is het mogelijk om snel een mooie webapplicatie te ontwikkelen, voorzien van GUI-elementen die men tegenwoordig van een moderne webapplicatie verwacht.

Twitter Bootstrap bevat o.a. het volgende:

- Mooie CSS styling voor tabellen, buttons, badges, labels, dropdown-buttons, tabs, breadcrumbs, alert dialogs, en progress bars.
- Meer dan 130 kleine icoontjes die voor gebruikers erg herkenbaar zullen zijn. Deze icoontjes kunnen op buttons of hyperlinks gebruikt worden. Er zijn bijvoorbeeld icoontjes voor het zoeken binnen een document, het downloaden van een bestand, of het sturen van een e-mail.
- JavaScript plugins voor moderne webapplicatie GUI elementen als tooltips, carousels, modale dialogen, typeahead-invoervelden en accordions.
- Compatibiliteit met alle gangbare moderne browsers: Internet Explorer 6-9, Firefox, Chrome, Safari, Opera, en nog vele anderen.
- Compatibiliteit met allerlei verschillende schermresoluties. Bootstrap zal ook goed werken op telefoons en tablets, omdat de layout van de pagina automatisch aangepast kan worden aan de schermresolutie.

Bootstrap is open-source en vrijgegeven onder de Apache 2.0 licentie. Op GitHub, een soort van sociale netwerk site voor softwareontwikkeling, is Bootstrap het meest populaire project. Bootstrap heeft de meeste 'volgers'<sup>(33)</sup> (mensen die interesse hebben in het project) en ook de meeste 'forks'<sup>(34)</sup>. Een fork is een soort van afsplitsing van een bestaand project. Veel forks houdt in dat veel mensen actief bezig zijn om aan Bootstrap mee te werken. Op GitHub is Bootstrap populairder dan bijvoorbeeld jQuery.

De afstudeerder heeft wat met Bootstrap geëxperimenteerd, en kwam er al snel achter dat men de CSS styling voor tabellen en buttons goed zou kunnen gebruiken. In eerste instantie had men het idee om Bootstrap alleen op specifieke plaatsen te gebruiken, bijvoorbeeld om tabellen mooi te maken of modale dialogen tevoorschijn te toveren. De rest van het nooddossier zou gestyled worden volgens de standaard MVC3-styling.

---

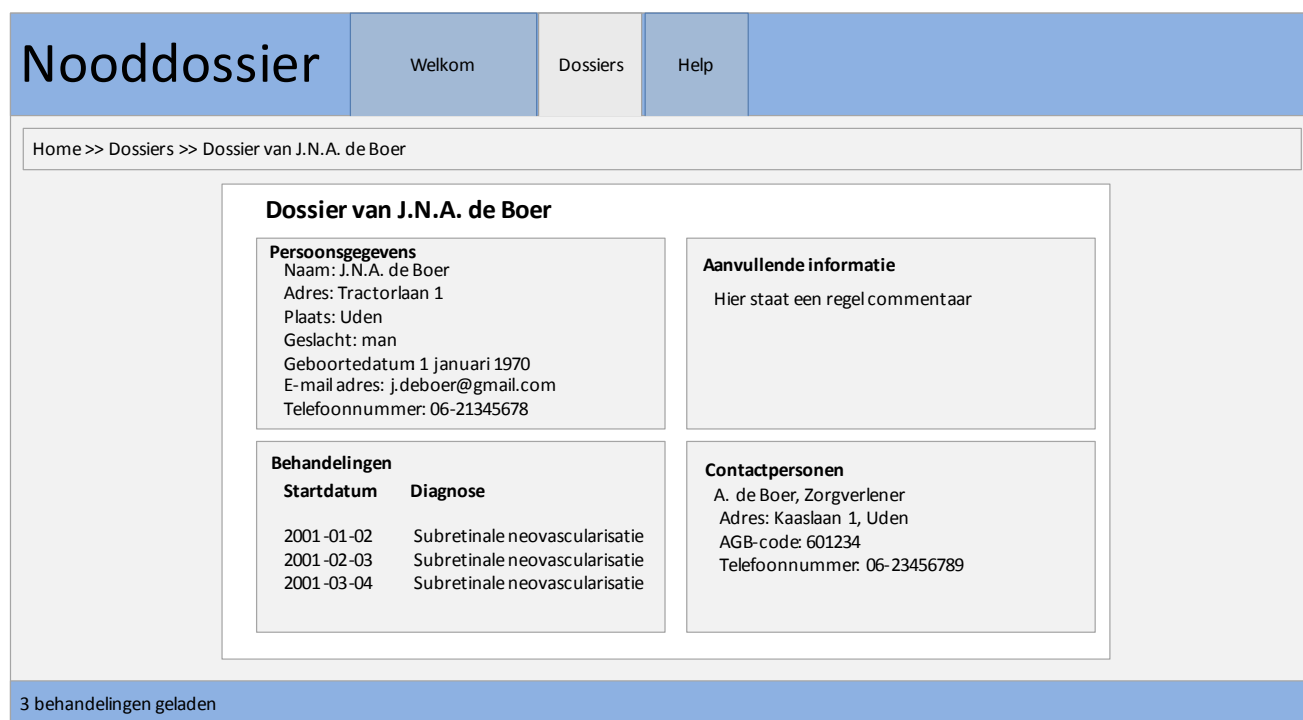
<sup>4</sup> In de lijst van meest 'geforkte' projecten staat op nummer één een project getiteld "Spoon-Knife". Dit is echter een testproject van GitHub bedoeld voor demonstraties, en is dus geen echt project.

Redelijk snel nadat men de beslissing gemaakt had om de standaard styling van MVC3 te gebruiken, met Bootstrap om sommige dingen iets mooier te maken, kwam de afstudeerder er achter dat het geheel veel beter tot zijn recht zou komen als alles dezelfde stijl heeft. In een ander project van de afstudeerder, waar hij in zijn vrije tijd aan gewerkt had, had hij ook gebruik gemaakt van Bootstrap. In dat project, wat overigens geen gebruik maakte van MVC3, had hij Bootstrap ook voor de pagina-layout en het hoofdmenu gebruikt. De afstudeerder merkte dat alles er erg goed uit zag zonder dat men veel moeite hoefde te doen. Daarom besloot men om de standaard styling van MVC3 toch niet te gebruiken, en om voor de basis-styling van het nooddossier óók Bootstrap te gaan gebruiken.

## GUI-ontwerp

Toen men eenmaal besloten had om Bootstrap overal voor te gaan gebruiken; kon men een ontwerp gaan maken voor de GUI.

In een aantal stappen is de GUI ontworpen. Men is begonnen met een hoofdontwerp voor de layout en de menu-indeling, en heeft vervolgens stapsgewijs detail aangebracht aan de rest van de applicatie. Het resultaat van dit traject staat weergegeven in figuur 15. Vervolgens heeft men dit ontwerp gerealiseerd met behulp van MVC3. In deze uitwerking waren de meeste gegevens (persoonsgegevens en dergelijke) rechtstreeks in de HTML uitgeschreven; men maakte nauwelijks gebruik van de back-end. Een screenshot van het resultaat staat weergegeven in figuur 16.



The screenshot displays a web application titled 'Nooddossier'. The top navigation bar includes links for 'Welkom', 'Dossiers', and 'Help'. Below the navigation bar, a breadcrumb trail reads 'Home >> Dossiers >> Dossier van J.N.A. de Boer'. The main content area is titled 'Dossier van J.N.A. de Boer' and is divided into four sections:

- Persoonsgegevens:**
  - Naam: J.N.A. de Boer
  - Adres: Tractorlaan 1
  - Plaats: Uden
  - Geslacht: man
  - Geboortedatum: 1 januari 1970
  - E-mail adres: j.deboer@gmail.com
  - Telefoonnummer: 06-21345678
- Aanvullende informatie:**
  - Hier staat een regel commentaar
- Behandelingen:**

Startdatum	Diagnose
2001-01-02	Subretinale neovascularisatie
2001-02-03	Subretinale neovascularisatie
2001-03-04	Subretinale neovascularisatie
- Contactpersonen:**
  - A. de Boer, Zorgverlener
  - Adres: Kaaslaan 1, Uden
  - AGB-code: 601234
  - Telefoonnummer: 06-23456789

At the bottom of the interface, a status bar indicates '3 behandelingen geladen'.

figuur 15 – Het ontwerp van het hoofdscherm van het Nooddossier.



localhost:51335/Home/Dossiers?bsn=145133813#

Nooddossier Welkom Dossiers Help Sitemap

Home » Dossiers » BSN:145133813

## Dossier van J.N.A. (Jan) de Boer

### Persoonsgegevens

Achternaam	de Boer
Voornaam	Jan
Voorletters	J.N.A.
Geslacht	man
Geboortedatum	21 maart 1971
Adres	Tractorlaan 1, 5403 BE in Uden
E-mail adressen	j.deboer@gmail.com
Telefoonnummers	06-12345678

### Aanvullende informatie

Hier staat commentaar

### Laatste behandelingen

Begindatum	Einddatum	Diagnose
2010-11-25	2010-11-26	Astma
2011-02-12	2011-03-25	Subretinale neovascularisatie

### Contactpersonen

naam: D. Heijn  
werkzaam bij Ministerie van Defensie in Veghel

Naam: H DE JONG  
Geslacht: MAN  
Adres: KAASLAAN 11 IN UDEN  
Postcode: 5403WE  
Landcode:  
AGB-Code: 6012345  
Telefoonnummer: 06-23456789

2 behandelingen geladen

© Info Support 2012

figuur 16 – De gerealiseerde versie van het ontwerp

Behandeling van J.N.A. de Boer

Postcode: 5403WE  
AGB-Code: 6041234  
Telefoonnummer: 06-12345678

Diagnose en behandeling

Diagnose	Subretinale neovascularisatie
Zorgproduct	Laserbehandeling bij subretinale neovascularisatie

Begin- en einddatum(s)

	Begindatum	Einddatum
Prestatie	12-02-2011	25-03-2011
Zorgtraject	22-01-2011	25-03-2011

Commentaar

De behandeling was voorspoedig voorlopen.

Annuleren

Opslaan

figuur 17 – De realisatie van het details-scherm. Het details-scherm wordt weergegeven wanneer men op een behandeling klikt op het hoofdscherm.

Nadat men van dit gehele ontwerp een uitwerking gemaakt had, had men eigenlijk nog maar één taak open staan: het omzetten van alle statische, handmatig ingevulde data naar ‘live’ data die van de back-end van het nooddossier opgehaald zou worden.

#### Dynamische content door middel van templates

Een van de problemen waar men al snel tegenaan liep bij het integreren van data welke opgehaald zou worden via de back-end, was dat het ontwikkelen van een webapplicatie met behulp van de JavaScript HTML DOM API nogal vervelend was. De DOM API is een API waarmee men HTML programmatisch kan bewerken vanuit JavaScript.

Stel dat men het volgende JavaScript object heeft welke een persoon representeert:

```
var persoon = {voornaam: "Jan", achternaam: "de Boer"}
```

figuur 18

..en men wil op basis van deze gegevens de volgende tabel genereren:

```
<table class="table table-striped table-bordered">
  <tbody>
    <tr>
      <th>voornaam</th>
      <td>Jan</td>
    </tr>
    <tr>
      <th>achternaam</th>
      <td>de Boer</td>
    </tr>
  </tbody>
</table>
```

figuur 19

Om deze tabel te genereren via de JavaScript HTML DOM API, moet men al gauw vele regels code schrijven:

```
var table = document.createElement("table");
table.setAttribute("class", "table table-striped table-bordered");

var tbody = document.createElement("tbody");
table.appendChild(tbody);

var tablerow1 = document.createElement("tr");
var tablerow2 = document.createElement("tr");
var tableheader1 = document.createElement("th");
var tableheader2 = document.createElement("th");
var tabledata1 = document.createElement("td");
var tabledata2 = document.createElement("td");

tableheader1.innerHTML = "Voornaam";
tabledata1.innerHTML = persoon.voornaam;
tableheader2.innerHTML = "Achternaam";
tabledata2.innerHTML = persoon.achternaam;

tablerow1.appendChild(tableheader1);
tablerow1.appendChild(tabledata1);
tablerow2.appendChild(tableheader2);
tablerow2.appendChild(tabledata2);
tbody.appendChild(tablerow1);
tbody.appendChild(tablerow2);
```

figuur 20

Het grote probleem van deze API is dat het lastig is om uit de code af te leiden wat het resultaat is wanneer men de code draait; men schrijft namelijk erg veel code die voornamelijk gebruikt wordt om de structuur aan te geven van de tabel, terwijl de inhoud wellicht veel belangrijker is.

Gelukkig is er een andere manier om dit soort zaken simpeler te maken. jQuery biedt een mogelijkheid om dit veel mooier te doen:

```
var table = $("<table class=\"table table-striped table-bordered\"><tbody></tbody></table>");
table.append("<tr><th>Voornaam</th><td>" + persoon.voornaam + "</td></tr><tr><th>Achternaam</th><td>" + persoon.achternaam + "</td></tr>");
```

figuur 21

De jQuery-oplossing is veel netter dan de JavaScript HTML DOM oplossing. Ten eerste is de code compact; er worden weinig variabelen geïntroduceerd en de documentstructuur wordt voornamelijk door jQuery afgehandeld.

Beide oplossingen zijn echter niet ideaal. Zo “escapen” ze beide niet de invoertekst. Indien men HTML-markup toevoegt aan de persoonsgegevens kan dit gevolgen hebben voor de tabelweergave. Deze kan bijvoorbeeld compleet verkeerd weergegeven worden, wat ernstige gevolgen heeft voor de bruikbaarheid van de applicatie. Bovendien kan iemand met kwade bedoelingen JavaScript-code injecteren in de pagina, wat mogelijk kan leiden tot account-diefstal en het lekken van belangrijke persoonsgegevens! Dit type veiligheidslek wordt ook wel ‘Cross-Site-Scripting’ (XSS) genoemd. Het is mogelijk om dit server-side op te lossen, dusdanig dat de server alleen maar data aanlevert die ge-escaped is, maar het is het ook mogelijk om dit client-side op te lossen in JavaScript, door ‘key’ en ‘persoon[key]’ in de bovenstaande codelijsten te escapen middels een escape-functie. Indien een ontwikkelaar echter vergeet om dit ergens te doen, is de hele applicatie kwetsbaar voor een XSS-aanval!

Wat men eigenlijk zou willen, is een mogelijkheid om op een veilige wijze HTML te genereren zonder dat men veel code hoeft te schrijven, en zonder dat het onduidelijk is wat het resultaat zal zijn. *Template engines* bieden de mogelijkheid om dit te doen. Een template engine is een stuk software welke als doel heeft om markup (bijvoorbeeld HTML) te genereren volgens een specifieke manier. De exacte manier waarop deze markup gegenereert dient te worden noemt men de *template*. Programmeurs of designers schrijven templates in een specifieke *templating language*, welke door de template engine ingelezen wordt en gecombineerd wordt met data om zo markup te kunnen genereren. Het genereren van markup aan de hand van een template en data noemt men ook wel *renderen*.

Een groot voordeel van het gebruiken van een template engine is dat men de template gemakkelijk kan aanpassen. Verder is de code veel netter, heeft men een goede scheiding tussen de weergave van de data en de data zelf, en kan een template engine zorgen voor een veiliger systeem door middel van het automatisch controleren van invoervelden op een XSS-aanval.

#### *Keuze templating system*

Tegenwoordig zijn er enorm veel templating systemen beschikbaar voor allerlei verschillende programmeertalen. Het was meteen al duidelijk dat men een keuze moest maken tussen allerlei verschillende templating systemen.

Voor de keuze van een templating systeem heeft de afstudeerder een blogpost van LinkedIn<sup>(35)</sup> gelezen over verschillende templating systemen, en heeft hij een aantal aandachtspunten gesteld waar hij de verschillende templating systemen op zou beoordelen:

- 1) Het gekozen template systeem moest implementaties hebben voor .NET en JavaScript. Toen er een keuze moest gemaakt worden voor een templating systeem, wist men nog niet zeker waar men de templates zou gaan gebruiken: in de back-end of in de front-end. Aangezien irritaties bij het ontwikkelen van de front-end de oorzaak waren voor het kiezen van een templating systeem, zou men in eerste instantie templates gebruiken in de front-end. Men wilde echter graag een 'fallback-optie' hebben, zodat het mogelijk was om toch nog de transitie te maken naar het gebruik van templates in de back-end. Dit zou namelijk kunnen zorgen voor snelheidswinst en grotere compatibiliteit met browsers.
- 2) Het template systeem moet zogenaamde *partials* ondersteunen. Partials zijn kleine stukken templates die men in andere templates kan invoegen. Op deze manier voorkomt men dat men bepaalde dingen meerdere keren gaat opschrijven. Met andere woorden: het template systeem moet het DRY ("Don't Repeat Yourself") principe goed ondersteunen.

Uiteindelijk kwam men terecht bij Mustache, een eenvoudige templating engine die voor allerlei verschillende programmeertalen implementaties heeft, waaronder JavaScript en .NET. Mustache wordt beschreven als een 'logic-less' templating engine. Het is niet mogelijk om met behulp van Mustache geavanceerde logica te gebruiken in een template; het enige wat mogelijk is, is het conditioneel een of meerdere keren weergeven van bepaalde data. Het logica-loze aspect van Mustache zorgt er voor dat Mustache templates zeer leesbaar zijn.

Een ander voordeel van Mustache is dat er een grote developer community achter zit. Veel mensen maken er gebruik van op allerlei verschillende platformen, en er bestaan ook aardig wat aanpassingen voor Mustache die meer functionaliteit bieden. Handlebars is een derivatieve van Mustache die extra functionaliteit introduceert in de vorm van 'block helpers' ten koste van de platform-independente<sup>(36)</sup>. De syntax daarentegen blijft nagenoeg gelijk, en Handlebars is dan ook compatible met de Mustache syntax. Handlebars is alleen beschikbaar voor JavaScript, en is zeer populair onder JavaScript-developers vanwege de hoge mate van functionaliteit en de simpele syntax.

Uiteindelijk heeft men er voor gekozen om Mustache te gebruiken voor de templates in de front-end, met als uitvalsmogelijkheden Handlebars op de front-end en Mustache op de back-end. Op deze manier zou men altijd nog kunnen kiezen om templates te renderen in de back-end zonder dat men daarvoor de templates zou hoeven aan te passen. Men heeft verder gekozen om Handlebars als extra uitvalsmogelijkheid te gebruiken voor het geval dat Mustache te weinig functionaliteit bood.

Verder heeft men besloten om een extra 'hulp-library' te gebruiken voor Mustache; ICanHaz geheten. ICanHaz zorgt er voor dat men templates kan embedden in een HTML-document met behulp van script-tags van type "text/html". Zonder ICanHaz zou men templates moeten definiëren in een JavaScript String, wat vervelend werkt aangezien men dan rekening moet houden met newlines en het escapen van quotes. Indien men Handlebars zou gebruiken,

zou de hulplibrary ICanHandlebarz gebruikt kunnen worden. ICanHandlebarz levert dezelfde functionaliteit als ICanHaz, maar dan voor Handlebars.

De template-definitie voor de tabel gespecificeert in figuur 19 is, voor zowel Mustache als Handlebars gebruikmakende van ICanHaz of ICanHandlebarz:

```
<script type="text/html" id="template_persoon">
  <table class="table table-striped table-bordered">
    <tbody>
      <tr>
        <th>Voornaam</th>
        <td>{{voornaam}}</td>
      </tr>
      <tr>
        <th>Achternaam</th>
        <td>{{achternaam}}</td>
      </tr>
    </tbody>
  </table>
</script>
```

figuur 22

Om deze template te renderen, hoeft men alleen maar ICanHaz of ICanHandlebarz aan te roepen:

```
$("#mainContent").append(ich.template_persoon(persoon)); //"#mainContent" is het element waaronder de template gerenderd wordt.
```

figuur 23

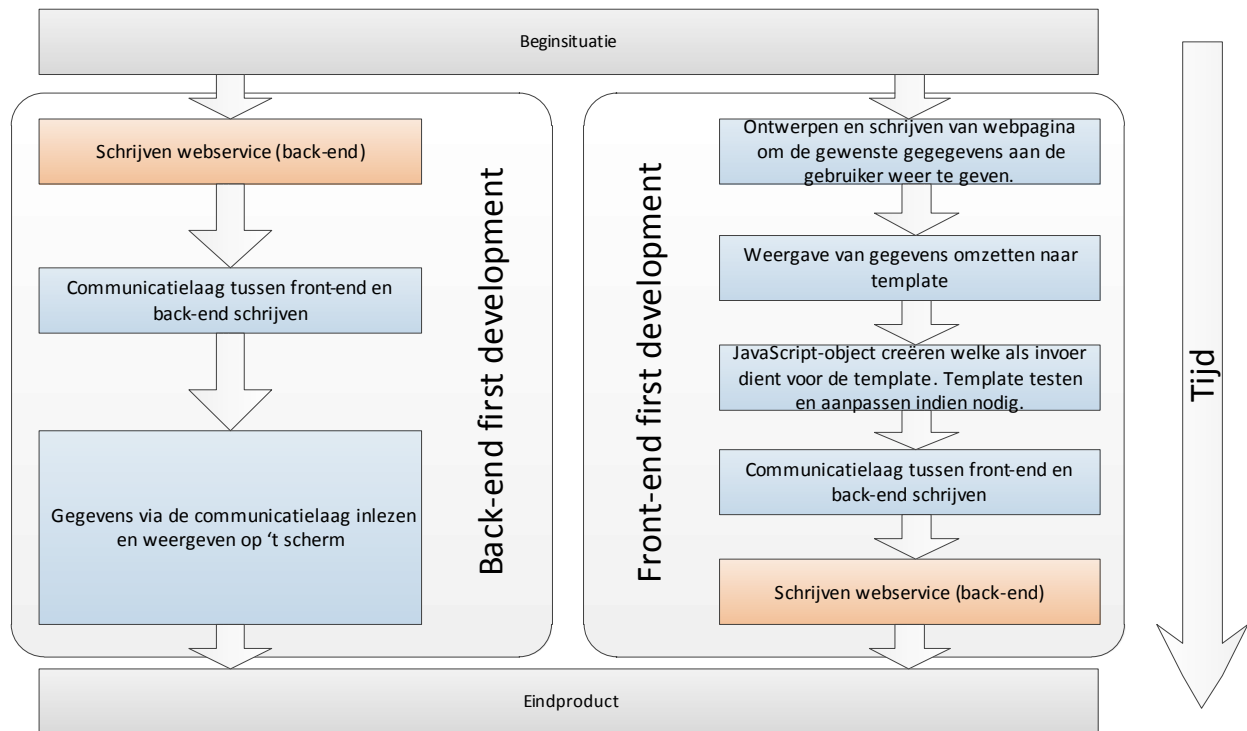
## Front-end first development

Dankzij het gebruik van templating engines, heeft de afstudeerder een erg handige manier ontdekt om webapplicaties te ontwerpen en te ontwikkelen.

Vóórdat de afstudeerder gebruik maakte van templating engines, implementeerde hij meestal eerst de communicatielaag tussen de front-end en back-end, zodat hij alle gegevens tot zijn beschikking zou hebben in de front-end. Vervolgens zou hij de code schrijven die de gegevens aan de gebruiker zou weergeven. Dit werkte meestal erg goed, totdat hij er achter kwam dat hij bepaalde andere gegevens toch op een andere manier zou willen weergeven. Indien dat het geval was, zou men delen van de back-end én van de front-end opnieuw moeten schrijven. Deze manier van werken, die de afstudeerder *back-end first development* noemde, leek niet altijd even goed te werken.

De nieuwe manier van ontwikkelen die de afstudeerder ontdekt had, welke *front-end first development* genoemd werd, draaide eigenlijk het hele proces om. Men zou beginnen met het maken van de weergave van de gegevens die men aan de gebruiker wilde laten zien. Dit deed men door een (deel van een) webpagina te maken waarin alle gegevens alvast handmatig ingevuld werden. Op deze manier kon de afstudeerder zich focussen op het design van de webpagina en hoe hij de gegevens zou willen weergeven. Vervolgens zou men de webpagina converteren naar een template. Dit was erg eenvoudig; in de meeste

gevallen hoefde de afstudeerder alleen maar de relevante HTML-markup te kopiëren en template annotaties toe te voegen op de plaatsen waar dat nodig was. Daarna zou men een JavaScript-object creëren welke als input gegeven werd voor het templating systeem. Op deze manier zou men kunnen controleren of de template op een juiste manier werkte. Als laatste zou men de code schrijven die de back-end en de front-end met elkaar zou verbinden.



figuur 24 - Schematische weergave van front-end first en back-end first development door de tijd heen.

De voornaamste reden waarom de afstudeerder front-end first development zo fijn vond, was dat hij maar één keer de frontend, backend en communicatielaag hoefde te maken, en hier verder geen aanpassingen aan zou hoeven te doen. Men wist immers al hoe men de data precies zou willen weergeven. Bij back-end first development kwam het toch een aantal keren voor dat men de communicatielaag zou moeten herschrijven, of de webservice aan zou moeten passen om de gegevens op een andere manier te structureren. Bovendien was frontend-first een stuk makkelijker voor het indelen van de dag middels Scrum: er waren meer taken die men moest doen, maar elke taak duurde een stuk korter. Vandaar dat, sinds dat de afstudeerder het frontend-first principe ontdekt heeft, hij alleen nog maar volgens dat principe gewerkt heeft.

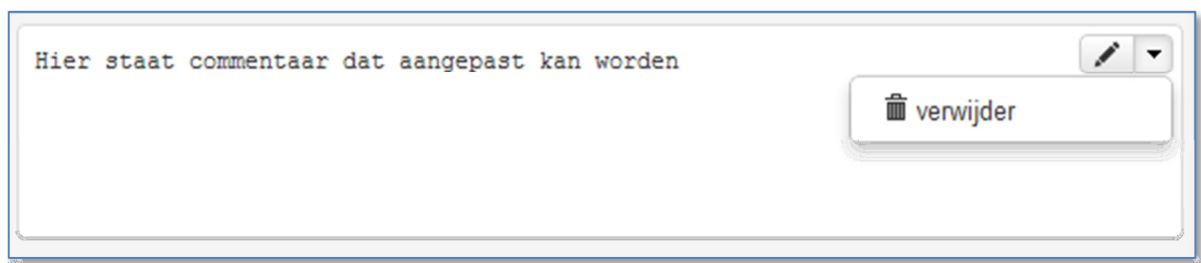
#### Aanpasbaar maken nooddossier

Eén van de features die de opdrachtgever graag in het nooddossier zou zien, was de mogelijkheid om van een persoon het dossier aan te passen. Zo zou het handig zijn als men van een persoon de persoonsgegevens kon wijzigen, als men extra aanvullend commentaar op het algemene dossier van een persoon kon leveren, en als men commentaar kon toevoegen bij behandelingen die een persoon ondergaan heeft. Het aanpassen van de details van een behandeling, zoals bijvoorbeeld de diagnose of de behandeldatum,

hoorde specifiek niet bij de gewenste features. Het zou immers een beetje vreemd zijn dat men de details van een behandeling zou kunnen aanpassen nadat deze behandeling plaats heeft gevonden; alsof men daardoor de behandelingsprestatie die uitgevoerd is zou kunnen aanpassen.

Het implementeren van deze feature heeft de afstudeerder aangepakt volgens de *frontend-first* methode. Men ben begonnen met het uitwerken van de GUI, heeft vervolgens code geschreven in de frontend die aanpassingen doorstuurt naar de backend, en heeft als laatste de backend-code geschreven die aanpassingen opslaat in een database. Deze keuze had men voornamelijk gemaakt vanwege de voordelen van front-end first development, waarover men meer kan lezen in de voorgaande paragraaf “*Front-end first development*”,

Voor de GUI heeft men gekozen voor speciale edit/delete buttons die zichtbaar gemaakt worden wanneer de gebruiker met de muis over een wijzigbaar element heen gaat.



figuur 25 - De edit/delete buttons

Wanneer de gebruiker een bepaald element wijzigt of toevoegt, wordt er intern een lijst van aanpassingen bijgehouden. Pas als de gebruiker op een ‘opslaan’ knop klikt, worden de wijzigingen doorgestuurd naar de server. Wanneer de gebruiker op ‘annuleren’ klikt, wordt de lijst van aanpassingen leeggemaakt en wordt de GUI opnieuw getekend. Omdat het dossier in het geheugen van de webbrowser pas aangepast wordt wanneer de gebruiker op ‘opslaan’ klikt, hoeft men geen complexe rollback-operatie te doen. Dit zorgt er voor dat de code compact blijft en men niet gemakkelijk fouten maakt.

De manier waarop de backend de aanpassingen opslaat is nog niet ideaal. In feite worden de aanpassingen als rauwe JSON opgeslagen in de database. Om de aanpassingen aan de front-end door te sturen, worden alle aanpassingen geconcateneerd en rauw met een dossier van een persoon mee gestuurd als lijst van wijzigingen. De wijzigingen worden dan in de front-end (client-side) doorgevoerd. Dit zorgt er voor dat de back-end erg simpel en snel werkt, maar kan voor security-problemen zorgen in de front-end, bijvoorbeeld wanneer iemand een manier vind om code te injecteren. Desalniettemin heeft men voor deze aanpak gekozen, omdat het een snelle manier was om de aanpasbaarheids-feature te implementeren in het nooddossier. Het nooddossier is immers maar een proof of concept, en security is eigenlijk buiten de scope gesteld in overeenstemming met de opdrachtgever. De mogelijkheid om meerdere features toe te kunnen voegen aan het uiteindelijke product was belangrijker dan het implementeren van goede security. Uiteraard zal men, wanneer men het nooddossier ook echt in productie wil gaan nemen, hier rekening mee moeten



houden; in dat geval zal men de backend moeten herschrijven zodat deze de aanpassingen op een goede wijze opslaat.

## 5.3 Documentatiefase

De documentatiefase was primair bedoeld om aan de scriptie te werken.

Volgens de planning zou de documentatiefase tegelijkertijd doorlopen worden met de hoofdfase van het project; één keer per week zou er een aantal uren aan de scriptie gewerkt worden. In de eerste paar sprints zou er maar een aantal uren per week aan gewerkt worden, maar in de laatste paar sprints zou dat oplopen tot wel één dag per week. Vervolgens zou er, als het gehele project af was, full-time gewerkt worden aan de scriptie om alles af te ronden.

Helaas bleek aan het eind van de hoofdfase dat het toch wat lastiger was om naast het programmeren aan de scriptie te werken. De afstudeerder vond het lastig om het programmeerwerk neer te leggen om even een hoofdstuk van de scriptie te schrijven. In plaats daarvan hield men korte notities bij waarin beschreven werd wat men gedaan had, en waarom men een bepaalde beslissing gemaakt had.

Aan het begin van de 6<sup>e</sup> sprint bleek dus dat men wellicht in tijdnood zou raken indien pas ná de uitloopweek aan de scriptie gewerkt zou worden. Vanwege het inlevermoment zou men dan precies anderhalve week hebben om de scriptie af te ronden. Vandaar dat samen met de opdrachtgever besloten was om het schrijven van de scriptie officieel op de Product Backlog te zetten als onderdeel van het Scrum-proces. De scriptie kreeg hoge prioriteit, en werd dus gekozen als het onderdeel van het project waarin men in de 6<sup>e</sup> sprint aan zou gaan werken. De uitloopweek en de twee documentatiefase-weken werden vervangen door een extra 7<sup>e</sup> en 8<sup>e</sup> sprint. Op deze manier zou men ná de scriptie gewoon weer verder kunnen gaan met het project.

In feite kan men dus zeggen dat aan het begin van de 6<sup>e</sup> sprint, de documentatiefase volledig ondergebracht werd in het proces van de hoofdfase.

## 6. Status van het project

Ten tijde van het schrijven van deze scriptie was men nog niet in de eindfase van het project beland. Er moet dus nog het een en ander geïmplementeerd worden. Daarom zal men in dit hoofdstuk kort samenvatten wat er op dit moment allemaal geïmplementeerd is, en wat men nog gaat toevoegen aan het project in de periode tussen het inleveren van de scriptie en de afstudeerzitting.

Het volgende is op dit moment gerealiseerd:

- Er is een specificatie geschreven voor een Web-API welke zorgverzekeraars kunnen implementeren om declaratiegegevens toegankelijk te maken voor het Nooddossier. Hiervan is een implementatie gemaakt welke gebruik maakt van dummy-gegevens. Zie voor deze specificatie de bijlage *“Ontwerp web API voor declaratiegegevens”*.
- Er is een webservice gemaakt welke de Controle op Verzekeringsrecht-check (COV-check) van VECOZO nabootst. Deze webservice maakt gebruik van dummy-gegevens, maar volgt (met uitzondering van de security) het communicatiepatroon zoals deze ook echt door VECOZO gebruikt wordt.
- Er is een werkend Nooddossier gerealiseerd waarmee men van een persoon kan zien wat voor behandelingen en diagnoses hij heeft gehad in een ziekenhuis. Binnen dit nooddossier wordt gebruik gemaakt van de twee producten hierboven. Van het dossier van een persoon kan men de persoonsgegevens bovendien aanpassen en aanvullen.

Het volgende is op dit moment nog niet geïmplementeerd, maar staat op de planning voor de periode tussen het inleveren van deze scriptie en de uiteindelijke afstudeerzitting:

- Integratie met Social Media: via het Nooddossier zal men extra details over een persoon kunnen zien die via Social Media opgehaald worden, en men zal middels Social Media de geboortedatum van een persoon (indien beschikbaar) automatisch op kunnen zoeken voor de COV-check.

## 7. Conclusie

Er is een proof of concept van een Nooddossier gerealiseerd. Dit Nooddossier is een webapplicatie welke verbinding maakt met enkele door de afstudeerder ontworpen systemen welke tezamen declaratiegegevens aanbieden aan het Nooddossier. Met het gehele systeem kan men de volgende taken verrichten:

- Men kan van een persoon de declaratiegegevens inzien welke betrekking hebben op behandelingen in ziekenhuizen. Tot de declaratiegegevens horen o.a. welke diagnose er gesteld is, wat voor een behandeling een persoon ondergaan heeft, en wanneer dit gebeurt is.
- Men kan van een persoon aanvullende persoonsgegevens inzien.
- Men kan persoonsgegevens aanpassen en aanvullen.
- Men kan commentaar leveren op de behandelingen.
- Men kan van een persoon een lijst van contactpersonen inzien. De contactpersonen bestaan uit zorgverleners welke een behandeling bij de persoon uitgevoerd hebben. Op deze manier zal men snel andere personen kunnen contacteren tijdens een noodgeval.

Verder is er, ter voorbereiding van de realisatie van het Nooddossier, een onderzoek gedaan naar o.a. social media. Uit dit onderzoek blijkt dat men theoretisch gezien bij 45% van alle Nederlanders de geboortedatum op kan halen via social media. Deze geboortedatum is nodig om van een persoon een nooddossier op te kunnen stellen wanneer men van deze persoon geen Burger Service Nummer kent.

In de periode tussen het inleveren van deze scriptie en de afstudeerzitting zal verder nog social media functionaliteit toegevoegd worden aan het Nooddossier. Middels deze functionaliteit moet het mogelijk zijn om van een onbekende persoon extra details te krijgen vanuit social media, en moet het mogelijk zijn om van een persoon het nooddossier te raadplegen zelfs wanneer men de geboortedatum niet kent. Ook extra contactpersonen zullen weergegeven worden, zodat behandelende artsen snel vrienden of familie kunnen raadplegen voor extra informatie.

De opdrachtgever wilde graag weten met wat voor problemen men kampte tijdens de realisatie van het Nooddossier. De volgende problemen zijn door de afstudeerder ondervonden:

- Men heeft bij slechts één sociaal netwerk een redelijke kans van slagen om de geboortedatum van een gebruiker op te halen. Dit sociale netwerk is Hyves. Hyves staat er op dit moment echter slecht voor qua marktpenetratie, en is al geruime tijd terrein aan het verliezen ten opzichte van Facebook.
- De documentatie van de APIs van verschillende sociale netwerk sites klopt niet altijd. De documentatie van Google Plus claimt dat men de geboortedatum van een persoon op kan vragen via de API, maar dit is incorrect.

## 8. Aanbevelingen

Ondanks dat men erg tevreden was over het eindresultaat, zijn er toch een aantal aanbevelingen. De meeste aanbevelingen zijn gericht op taken die de afstudeerder tijdens zijn afstuderen wel had kunnen volbrengen, maar waar hij samen met de opdrachtgever van bepaald had dat deze niet gedaan zouden worden vanwege de beschikbare tijd en andere taken die hogere prioriteit hadden.

- Implementeer security goed; dit is buiten de scope gesteld maar is desalniettemin zeer belangrijk. NEN 7510, “Informatiebeveiliging in de zorg”, geeft hier een aantal minimumeisen en richtlijnen voor.
- Zorg ervoor dat mensen zelf kunnen kiezen wat voor data ze beschikbaar willen stellen voor het Nooddossier. Zorg ervoor dat mensen eerst toestemming moeten geven voordat hun informatie via het nooddossier geraadpleegd kan worden; op deze manier kan men voorkomen dat mensen een slecht gevoel krijgen bij het nooddossier.
- Kijk naar de legaliteit van alle dingen die men in het nooddossier heeft geïmplementeerd of nog wil implementeren. Het kan zijn dat er bepaalde zaken verplicht op een bepaalde manier afgehandeld moeten worden, bijvoorbeeld beveiliging van persoonsgegevens. Kijk hierbij ook meteen naar de terms of service van de verschillende sociale netwerken, zodat men kan bepalen wat het risico is dat een sociaal netwerk de toegang van het nooddossier tot hun API blokkeert.
- Vraag aan de verschillende sociale netwerk sites om exclusieve toegang te krijgen tot extra gegevens of hogere datalimieten. Op deze manier zal het nooddossier meer informatie kunnen geven aan eindgebruikers, of zal het mogelijk zijn om meer eindgebruikers tegelijkertijd te ondersteunen.
- Schrijf een privacy policy en voeg die toe aan het Nooddossier. Dit is een vereiste van het gebruik van de verschillende sociale netwerken.
- Implementeer de andere EI-standaarden, zodat men ook van declaraties afkomstig van andere soorten zorgverleners (bijvoorbeeld huisartsen of apothekers) verdere informatie kan bekijken.
- Kijk naar unit-testing. Vooral kritieke code zal goed getest dienen te worden. Sommige libraries en frameworks die gebruikt worden in het nooddossier hebben zelf al unit tests, dus die hoeven niet meer getest te worden, tenzij de tests niet de juiste dekkinggraad bieden.
- Schrijf een betere backend voor aanpassingen. Op dit moment wordt rauwe JSON opgeslagen, wat niet erg flexibel is en bovendien onveilig kan zijn. De backend moet kunnen begrijpen wat voor aanpassingen de gebruiker heeft gedaan, en deze aanpassingen netjes opslaan.
- Kijk naar de bruikbaarheid van het nooddossier op tablets of telefoons. Waarschijnlijk werkt het nooddossier goed op tablets en telefoons, omdat Bootstrap daar support voor biedt, maar met wat kleine aanpassingen kan men het Nooddossier wellicht perfect laten draaien op deze platformen zonder dat het veel moeite kost! Een voorbeeld van een aanpassing die men zou moeten doen, is het zichtbaar maken van de edit-buttons wanneer men het scherm aanraakt. Op dit moment worden deze

buttons zichtbaar gemaakt wanneer men met de muis over een element zweeft, wat natuurlijk niet mogelijk is op een telefoon of tablet.

- Kijk naar de performance van het algehele nooddossier project. Het is wellicht mogelijk dat er een bepaalde zwakke plek in het systeem zit die er voor kan zorgen dat het nooddossier niet altijd snel genoeg reageert wanneer er erg veel medisch specialisten tegelijkertijd gebruik maken van het nooddossier. Let hierbij ook op de schaalbaarheid: wellicht is het mogelijk dat er meerdere nooddossier-servers naast elkaar worden gezet om zo de communicatie met de nooddossier-webapplicaties te verdelen over meerdere servers. Dit zal er voor zorgen dat het nooddossier sneller reageert, en men sneller informatie op kan vragen van een patiënt.

## Evaluatie

Als backend-webdeveloper en iemand die het werken met Java erg fijn vindt, was ik erg benieuwd hoe het werken met ASP.NET MVC3 mij zou bevalen, en wat ik zou vinden van het werken met JavaScript op de front-end.

### MVC3

Ik was in het begin erg cynisch over MVC3. Ik had het idee dat het schrijven van een webapplicatie erg vervelend zou zijn met MVC3, en dat het resultaat een 'oubollige' webapplicatie zou zijn waarbij er weinig gebruik gemaakt zou worden van moderne technieken op web-gebied.

Ik was dan ook erg verrast toen ik er achter kwam dat het standaard MVC3-project geleverd wordt met populaire open-source libraries als jQuery. Deze JavaScript libraries zorgen er voor dat developers gemakkelijk kunnen werken aan dynamische webpagina's die werken op allerlei verschillende browsers. Verder zijn er ook erg veel andere open-source libraries beschikbaar voor .NET, zoals een library die het mogelijk maakt om te communiceren met de Twitter API. Al deze libraries kunnen geïnstalleerd worden via NuGet, een open-source package management tool voor applicaties die compleet met Visual Studio geïntegreerd is. Via NuGet kon ik gemakkelijk allerlei functionaliteit toevoegen aan mijn webservices en mijn webapplicatie.

Ook had ik niet verwacht dat men MVC3 op verschillende manieren kan gebruiken. Ik had verwacht dat MVC3-webapplicaties redelijk 'strikt' geschreven moesten worden volgens het Model-View-Controller design pattern, omdat de naam daarvan afgeleid is. Ik was blij verrast toen ik er achter kwam dat je ook erg veel met JavaScript kan doen, of dat je er voor kan kiezen om het Model-View-Controller patroon op slechts een paar punten te gebruiken.

### Front-end web development

Toen ik begon aan het Nooddossier project, had ik niet erg veel ervaring met front-end web development. Ik had tijdens mijn opleiding en vorige stage wel het een en ander met CSS, HTML en JavaScript gedaan, maar de focus lag er nooit echt op; het was altijd een bijzaak. Voor het nooddossier project was het echter van belang dat ik er veel mee zou gaan doen, vooral wanneer ik een echt mooie webapplicatie zou willen maken.

Het werken met HTML, JavaScript en CSS beviel me enorm. Dankzij Twitter Bootstrap kon ik er zelf voor kiezen op welk niveau ik met de verschillende technologieën werkte. Zo had ik er voor gekozen om iets minder met de styling (CSS) te doen, maar juist iets meer met HTML en JavaScript (de achterliggende opmaak/functionaliiteit) aangezien daar mijn sterktepunten lagen. Desalniettemin heb ik toch aardig wat met de CSS moeten doen. Zo heb ik zelf de onderste navigatiebalk en de breadcrumb-bar van het Nooddossier moeten implementeren. Bovendien heb ik wat aanpassingen gemaakt om tabellen en de edit/delete knoppen mooi weer te geven.

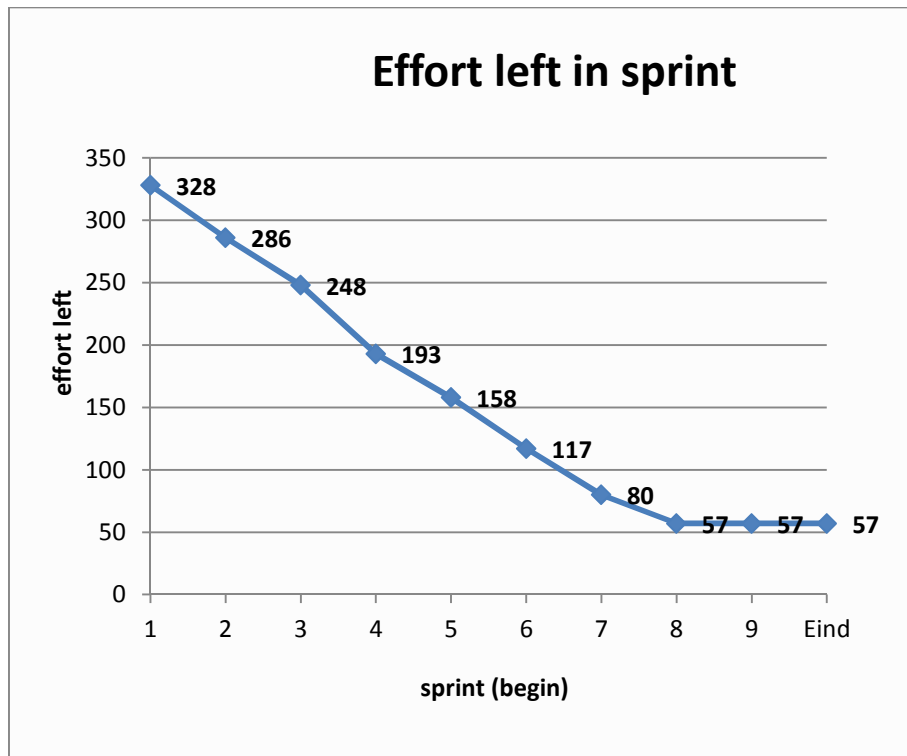
Uiteindelijk heb ik dankzij het nooddossier project erg veel geleerd over JavaScript en CSS. Zo snap ik nu veel beter hoe JavaScript werkt. Ik had voorheen nooit echt door dat men in JavaScript vaak data geeft aan functies, terwijl in de gangbare object-geïntendeerde programmeertalen men meestal functies geeft aan data. Toen ik dit eenmaal door had, begreep ik andere concepten die men in JavaScript vaak ziet, zoals closures, een stuk beter.

Voordat ik intensief met HTML, JavaScript en CSS ging werken als onderdeel van het nooddossier project, programmeerde ik in mijn vrije tijd voornamelijk in Java. Ik heb Java altijd al een fijne programmeertaal gevonden om mee te werken, en de mogelijkheid om Java-code op allerlei verschillende operating systemen te draaien vond ik erg nuttig en belangrijk. Sinds dat ik me heb verdiept in HTML, JavaScript en CSS, heb ik bijna geen regel Java-code meer geschreven. Ik vond het vanaf dat moment namelijk een stuk handiger om alles in JavaScript te doen, mede dankzij Twitter Bootstrap. De user interfaces werden redelijk gemakkelijk om te maken, het werd voor gebruikers eenvoudiger om mijn applicaties in gebruik te nemen, en ik vond het een stuk makkelijker om iets echt indrukwekkends te maken. Bovendien vond ik veel dingen ook een stuk leuker om aan te werken, wellicht omdat ik met wat nieuwere technieken aan de slag kon gaan.

## Proces: Scrum

Het werken volgens Scrum was me erg goed bevallen. Omdat ik een burndown-chart had, kon ik ten alle tijden zien of ik tijd tekort zou komen voor het realiseren van het Nooddossier. De burndown-chart, en de tweewekelijkse vergaderingen met mijn opdrachtgever, gaven me erg veel feedback tijdens het werken en inzicht in de projectplanning. Op deze manier kon ik samen met de opdrachtgever goede beslissingen maken over de prioritering, en was het mogelijk om een goede inschatting te maken over de taken die ik in één sprint uit zou kunnen voeren. Bovendien had ik zo een goed overzicht over de taken die ik in korte termijn zou moeten volbrengen, en taken die ik op lange termijn zou moeten volbrengen.





figuur 26 – De burndown-chart

Zoals men uit bovenstaande burndown-chart kan afleiden, loop ik op schema en zal het project op tijd afgerond worden. Op dit moment zit ik namelijk in sprint 8; ik heb nog twee sprints over om de rest van de taken te verrichten.

## Competenties

Binnen Fontys word er gebruik gemaakt van een aantal competenties, welke gebruikt worden om van een student te bepalen of hij gereed is om de beroepsmarkt te betreden. Tijdens het afstuderen heb ik van een aantal competenties extra kunnen aantonen dat ik deze goed onder de knie heb.

Zo heb ik bijvoorbeeld *Professioneel handelen* aangetoond toen ik er achter kwam dat de manier waarop ik delen van het nooddossier maakte voor problemen zorgde, en vervolgens besloot om gebruik te maken van een templating engine om deze problemen te verhelpen. Ik heb dus gereflecteerd op mijn eigen werk en besloten om hierin een verbetering door te voeren. Zie voor meer informatie over deze beslissing de paragraaf “*Dynamische content*”.

Verder heb ik ook *Methodisch handelen* aan kunnen tonen door middel van het Scrum-proces zelf. Elke twee weken heb ik met mijn opdrachtgever een planning- en reviewsessie gehouden. Tijdens deze sessie liet ik aan mijn opdrachtgever zien wat ik gedaan had in de afgelopen twee weken, en wat ik in de komende twee weken zou gaan doen. Het plannen werd gedaan aan de hand van de kennis die ik verkregen had over mijn werksnelheid in de vorige sprints. De keuze over welke taken ik in een bepaalde sprint zou gaan doen, was afhankelijk van de wensen van de opdrachtgever en de inschatting die ik zelf had gemaakt over hoe lang een bepaalde taak zou gaan duren. Met behulp van al deze informatie werd

uiteindelijk voor elke periode van twee weken een takenlijst opgezet, die haalbaar was en voldeed aan de behoeften van de opdrachtgever. Op deze wijze heb ik dus methodisch gehandeld. Voor meer informatie, zie mijn evaluatie van het Scrum proces in de paragraaf “*Proces: Scrum*”.

Ook van de competentie *Ontwerpen* heb ik aardig wat aan kunnen tonen. Zo heb ik een specificatie van een Web API ontworpen en uiteindelijk gerealiseerd. Voor meer informatie hierover, zie de paragraaf “*Web API voor declaratiegegevens*”.

Van de competentie *Realiseren* heb ik ook flink wat aan kunnen tonen. Ik heb namelijk software kunnen realiseren welke voldeed aan de eisen van de opdrachtgever, zonder dat ik veel ervaring had met de programmeertalen waar ik mee ging werken. Voor meer informatie over wat ik uiteindelijk gerealiseerd heb, zie paragraaf “*6. Status van het project*”.

## Eindresultaat

Ik ben erg tevreden met het eindresultaat. Het Nooddossier ziet er goed uit, en ik ben tevreden over de functionaliteit die ik geïmplementeerd heb.

Ik ben voornamelijk erg tevreden met de grafische user interface (GUI) van het Nooddossier. Deze ziet er, dankzij het gebruik van Twitter Bootstrap, erg goed uit. Omdat ik veel van de functionaliteit geïmplementeerd heb in JavaScript, en omdat ik veel gebruik maak van asynchrone communicatie met de backend, is het bovendien prettig werken met het Nooddossier: bepaalde gegevens hoeven alleen maar ingeladen te worden wanneer ze nodig zijn, en vanaf het moment dat ze ingeladen zijn kan men verder bladeren zonder dat men hoeft te wachten.

## Bronnenlijst

1. **Ministerie van Volksgezondheid, Welzijn en Sport.** Brief elektronisch patiëntendossier. [Online] 31 Oktober 2008. <http://www.rijksoverheid.nl/bestanden/documenten-en-publicaties/brieven/2008/10/31/brief-elektronisch-patientendossier/brief-zorgconsumenten.pdf>.
2. **NICTIZ, CIBG en Ministerie van Volksgezondheid, Welzijn en Sport.** Het elektronisch patiëntendossier. [Online] 5 September 2008. <http://www.rijksoverheid.nl/bestanden/documenten-en-publicaties/brochures/2008/09/08/het-elektronisch-patientendossier/meva-2875251b.pdf>.
3. **Schippers, mw. drs. E.I. (Minister van Volksgezondheid, Welzijn en Sport).** Kamerbrief stand van zaken EPD. [Online] 11 april 2011. <http://www.rijksoverheid.nl/bestanden/documenten-en-publicaties/kamerstukken/2011/04/11/kamerbrief-stand-van-zaken-epd/kamerbrief-stand-van-zaken-epd.pdf>.
4. **Info Support B.V.** Over Info Support. *Info Support website*. [Online] <http://www.infosupport.com/OverInfoSupport>.
5. **Info Support B.V.** Referenties. *Info Support website*. [Online] [http://www.infosupport.com/referentie\\_root](http://www.infosupport.com/referentie_root).
6. **Info Support B.V.** De structuur van Info Support BV. [Online] <http://carriere.infosupport.com/BVKwOrganisatie>.
7. **VECOZO.** Over VECOZO. *VECOZO website*. [Online] [Citaat van: 30 5 2012.] [http://www.vecozo.nl/web/Over\\_VECOZO/Default.aspx](http://www.vecozo.nl/web/Over_VECOZO/Default.aspx).
8. **VECOZO.** Gebruikershandleiding Controle op Verzekeringsrecht. [Online] 2.9a, 13 Augustus 2010. [https://www.vecozo.nl/web/download.aspx?bin=Handleiding\\_COV](https://www.vecozo.nl/web/download.aspx?bin=Handleiding_COV).
9. **Danube Technologies.** The Essential Scrum Cheat Sheet. *DZone Refcardz*. [Online] <http://refcardz.dzone.com/refcardz/scrum>.
10. **Google.** Age requirements on Google Accounts. *Google*. [Online] <http://support.google.com/plus/bin/answer.py?hl=en&answer=1350409>.
11. **Facebook.** Graph API. *Facebook developer website*. [Online] <http://developers.facebook.com/docs/reference/api/>.
12. **Google.** Google+ platform: people-search API documentation. *Google Developers*. [Online] <https://developers.google.com/+/api/latest/people/search>.
13. **Hyves.** Hyves users.search API documentation. *Hyves developer documentation*. [Online] <http://www.hyves-developers.nl/documentation/data-api/methods/2.1/users.search>.
14. **LinkedIn.** People Search API. *LinkedIn Developers website*. [Online] <https://developer.linkedin.com/documents/people-search-api>.
15. **Twitter.** Users/search API Documentation. *Twitter Developers website*. [Online] <https://dev.twitter.com/docs/api/1/get/users/search>.
16. **Socialbakers.** Netherlands Facebook Statistics. *Socialbakers.com*. [Online] [Citaat van: 6 Maart 2012.] <http://www.socialbakers.com/facebook-statistics/netherlands#chart-intervals>.
17. **Hyves.** Over Hyves. *Hyves.nl*. [Online] [Citaat van: 6 Maart 2012.] <http://www.hyves.nl/over/facts/?&pageid=3LJWZGWBQ9ESKWKC8>.
18. **SemioCast.** Brazil becomes 2nd country on Twitter, Japan 3rd; Netherlands most active country. *SemioCast.com*. [Online] 31 Januari 2012. [Citaat van: 6 Maart 2012.] [http://semioCast.com/publications/2012\\_01\\_31\\_Brazil\\_becomes\\_2nd\\_country\\_on\\_Twitter\\_s\\_uperseds\\_Japan](http://semioCast.com/publications/2012_01_31_Brazil_becomes_2nd_country_on_Twitter_s_uperseds_Japan).

19. **LinkedIn**. LinkedIn goes Orange. *LinkedIn Press Center*. [Online] 27 Januari 2012. [Citaat van: 6 Maart 2012.] <http://press.linkedin.com/node/1082>.
20. **Centraal Bureau voor de Statistiek**. Bevolkingsteller. *cbs.nl*. [Online] [Citaat van: 6 Maart 2012.] [http://www.cbs.nl/nl-nl-menu/themas/bevolking/cijfers/extra/bevolkingsteller.htm](http://www.cbs.nl/nl-nl/menu/themas/bevolking/cijfers/extra/bevolkingsteller.htm).
21. **Nawcom Research & Consultancy**. Social Media gebruik in Nederland 2012. *Slideshare.net*. [Online] 2 Mei 2012. <http://www.slideshare.net/newcomresearch/newcom-research-consultancy-gebruik-social-media-nl-mei-2012>.
22. **VECOZO**. Mogelijkheden EDP. [Online] 10 Mei 2012. <http://www.vecozo.nl/web/download.aspx?bin=MogelijkhedenEDP2012.pdf>.
23. **Nederlandse Zorgautoriteit**. DBC/DOT Dossier. *Website van de Nederlandse Zorgautoriteit*. [Online] <http://www.nza.nl/zorgonderwerpen/dossiers/dbc-dossier/>.
24. **Nederlandse Zorgautoriteit**. Invoering DOT per 2012 definitief. *Website van de Nederlandse Zorgautoriteit*. [Online] 30 Juni 2011. <http://www.nza.nl/publicaties/nieuws/Invoering-DOT-per-2012-definitief/>.
25. **Vektis C.V.** Project Invoering DOT Declaratie: Ondersteuning invoering van de nieuwe ZH-standaard versie 8. *Website Vektis C.V.* [Online] <http://www.vektis.nl/index.php/invoering-dot-declaratie>.
26. **Vektis C.V.** EI-standaard voor Ziekenhuiszorg ZH308. *Webapplicatie EI-standaardisatieproducten*. [Online] 15 Maart 2011. [http://ei.vektis.nl/WespStandaardenOverzichtDetail.aspx?is\\_iber=ZH308&is\\_versie=8.0&showstructuur=true](http://ei.vektis.nl/WespStandaardenOverzichtDetail.aspx?is_iber=ZH308&is_versie=8.0&showstructuur=true).
27. **Vektis C.V.** Overzicht codelijsten. *Webapplicatie EI-standaardisatieproducten*. [Online] <http://ei.vektis.nl/WespCodelijstenOverzicht.aspx>.
28. **Microsoft**. What Is Windows Communication Foundation. *MSDN*. [Online] <http://msdn.microsoft.com/en-us/library/ms731082.aspx>.
29. **Vektis C.V.** ZH308 versie 8 testset release 4. *Webapplicatie EI-standaardisatieproducten*. [Online] [http://ei.vektis.nl/Help/ZH308v8.0/ZH308v8.0\\_ZV\\_Testset\\_u4.zip](http://ei.vektis.nl/Help/ZH308v8.0/ZH308v8.0_ZV_Testset_u4.zip).
30. **Newton-King, James**. About JSON.NET. *CodePlex*. [Online] <http://json.codeplex.com/>.
31. **Newton-King, James**. Good (Date)Times with Json.NET. *Website van James Newton-King*. [Online] <http://james.newtonking.com/archive/2009/02/20/good-date-times-with-json-net.aspx>.
32. **Twitter**. Bootstrap, from Twitter. [Online] <http://twitter.github.com/bootstrap/>.
33. **GitHub**. GitHub popular watched repositories. *GitHub Repositories*. [Online] [Citaat van: 17 Mei 2012.] <https://github.com/popular/watched>.
34. **GitHub**. GitHub popular forked repositories. *GitHub Repositories*. [Online] <https://github.com/popular/forked>.
35. **LinkedIn**. The client-side templating throwdown: mustache, handlebars, dust.js, and more. *LinkedIn Engineering*. [Online] Januari 2012. <http://engineering.linkedin.com/frontend/client-side-templating-throwdown-mustache-handlebars-dustjs-and-more>.
36. **Handlebars Developers**. *Handlebars website*. [Online] <http://handlebarsjs.com/>.
37. **Hyves**. Hyves Privacy Policy. *Hyves*. [Online] Oktober 2011. [Citaat van: 29 Mei 2012.] hoofdstuk 8. <http://www.hyves.nl/index.php?l1=ut&l2=pr&l3=hm>.

## Bijlagen

- I Plan van Aanpak
- II Onderzoeksresultaten
- III Ontwerp Web API voor declaratiegegevens



Solid Innovator  
Solid Innovator



## Bijlage I: Plan van Aanpak

### **Nooddossier**



## Colofon

Titel	Plan van Aanpak: Nooddossier
Versie, datum	1.1, 6-6-2012
Samengesteld door	Joris van de Donk
Project	Nooddossier

© Info Support, Veenendaal 2012

Niets uit deze uitgave mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, microfilm of op welke andere wijze ook, zonder voorafgaande toestemming van Info Support.

No part of this publication may be reproduced in any form by print, photo print, microfilm or any other means without written permission by Info Support.

Prijsopgaven en leveringen geschieden volgens de leveringsvoorwaarden van Info Support b.v., gedeponeerd bij de K.v.K. te Utrecht onder nr. 30135370



## Historie

Versie	Datum	Verandering	Pagina's
0.1	9 feb 2012	Initiële versie	20
0.2	20 feb 2012	<ul style="list-style-type: none"> <li>- Feedback van Ivo Diepstraten en Chris van Beek verwerkt</li> <li>- Backlog toegevoegd</li> <li>- Onderzoeksvragen toegevoegd</li> <li>- Planning aangepast (invoering sprints, uitloop verkleind en verschoven)</li> </ul>	23
1.0	12 maart 2012	<ul style="list-style-type: none"> <li>-Indeling en bewoording onderzoeksvragen aangepast</li> <li>-Backlog aangepast</li> <li>-Bedrijfsbeschrijving en begrippenlijst toegevoegd</li> </ul>	27
1.1	5 juni 2012	Versie voor als bijlage in de Afstudeerscriptie. Veranderingen: <ul style="list-style-type: none"> <li>-Bijlages verwijderd, omdat deze ook al in de afstudeerscriptie staan</li> </ul>	20

## Distributie

Versie	Datum	Aan
0.1	9 feb 2012	<ul style="list-style-type: none"> <li>-Ivo Diepstraten</li> <li>-Chris van Beek</li> </ul>
0.2	20 feb 2012	<ul style="list-style-type: none"> <li>-Ivo Diepstraten</li> <li>-Chris van Beek</li> <li>-Rene van der Heijden</li> <li>-Marieke Keurtjes</li> </ul>
1.0	12 maart 2012	<ul style="list-style-type: none"> <li>-Ivo Diepstraten</li> <li>-Chris van Beek</li> <li>-Rene van der Heijden</li> <li>-Marieke Keurtjes</li> </ul>

## Management samenvatting

Dit document beschrijft kort het plan van aanpak van het Nooddossier project, een afstudeerproject welke door Joris van de Donk uitgevoerd wordt. De opdrachtgever van het project is Ivo Diepstraten. De afstudeerder wordt technisch begeleid door Chris van Beek, en procesmatig begeleid door Marieke Keurntjes.

Het Nooddossier project is een project met het uitgangspunt om voor medisch specialisten in tijden van nood een medisch dossier aan te kunnen bieden dat de declaratiegeschiedenis van een onbekende patiënt weergeeft. Deze declaratiegeschiedenis zal opgevraagd worden bij de zorgverzekeraars waar de patiënt bij verzekerd is (geweest).

Het eindresultaat van het Nooddossier project is een proof-of-concept van een Nooddossier inclusief de bijbehorende (ontwerp)documentatie.

Het Nooddossier zal in grote lijnen geproduceerd worden volgens de Scrum-methode. Er zal iteratief gewerkt worden zodat de kans zeer groot is dat er een werkend proof-of-concept opgeleverd kan worden. Indien er tijd beschikbaar is zal het proof-of-concept uitgebreid worden met extra functionaliteiten.

# Inhoudsopgave

<b>COLOFON .....</b>	<b>1</b>
<b>HISTORIE .....</b>	<b>2</b>
<b>DISTRIBUTIE .....</b>	<b>2</b>
<b>MANAGEMENT SAMENVATTING.....</b>	<b>3</b>
<b>1. OPDRACHT .....</b>	<b>5</b>
1.1 OPDRACHTGEVER EN OPDRACHTNEMER .....	5
1.2 OPDRACHTDEFINITIE .....	6
1.3 AFBAKENING.....	7
1.4 AFHANKELIJKHEDEN.....	8
1.5 KWALITEITSEISEN.....	8
1.6 UITGANGSPUNTEN .....	9
1.7 RANDVOORWAARDEN.....	9
<b>2. RISICOMANAGEMENT .....</b>	<b>10</b>
<b>3. AANPAK.....</b>	<b>11</b>
3.1 INCEPTIEFASE.....	11
3.2 HOOFDFASE.....	11
3.3 DOCUMENTATIEFASE (PARALLEL MET 3.2).....	12
3.4 EINDFASE .....	12
<b>4. BEHEERSASPECTEN .....</b>	<b>13</b>
4.1 ORGANISATIE.....	13
4.2 TIJD.....	13
4.3 GELD .....	15
4.4 MIDDELEN.....	15
4.5 KWALITEIT.....	16
4.6 COMMUNICATIE.....	16

# **1. Opdracht**

## **1.1 Opdrachtgever en opdrachtnemer**

De opdrachtnemer voor de in dit plan van aanpak beschreven delen is:

Adres: Info Support b.v.  
Kruisboog 42  
3905 TG Veenendaal  
Info@infosupport.com  
www.infosupport.com  
Contactpersoon: Joris van de Donk  
06-30517993

De opdrachtgever voor de in dit plan van aanpak beschreven delen is:

Adres: Info Support b.v.  
Business Unit Zorg  
Kruisboog 42  
3905 TG Veenendaal  
[Info@infosupport.com](mailto:Info@infosupport.com)  
[www.infosupport.com](http://www.infosupport.com)  
Contactpersoon: Ivo Diepstraten  
06-45378523

## 1.2 Opdrachtdefinitie

### Aanleiding

Het landelijk Elektronisch Patiënten Dossier (EPD; een systeem welke zorgverleners in staat stelt om informatie uit te wisselen over patiënten) is door de Eerste Kamer verworpen, maar toch blijft de noodzaak bestaan om in geval van een noodsituatie medische informatie van een onbekende patiënt toegankelijk te maken voor de behandelende specialisten. Het snel zichtbaar maken van relevante medische informatie zou in een echt noodgeval levens kunnen redden, voornamelijk wanneer een bepaalde patiënt een aandoening heeft welke een specifieke behandeling levensbedreigend kan maken.

Er is in de declaratiegegevens van de zorgverzekeraars veel informatie terug te vinden over de medische geschiedenis van een patiënt. Een medisch nooddossier zou deze declaratiegeschiedenis op eenvoudige wijze voor een medisch specialist inzichtelijk moeten maken.

Info Support zou graag een proof-of-concept willen zien van een medisch nooddossier, zodat ze weten dat het concept werkt en wellicht doorontwikkeld kan gaan worden. Verder willen ze weten met wat voor problemen men te maken gaat krijgen indien men in de toekomst een nooddossier-achtige service aan zal bieden aan derden.

### Probleemstelling

Indien men alleen een naam en geslacht weet, zou men onvoldoende informatie hebben om de medische geschiedenis van een patiënt op te vragen. Daarvoor zijn namelijk extra gegevens nodig, zoals de geboortedatum van de patiënt. Via *social media* kunnen tegenwoordig dit soort gegevens beschikbaar gemaakt worden aan de buitenwereld, en het is de vraag of men verschillende sociale netwerken kan doorzoeken om dit soort gegevens van een persoon op te halen. Op deze manier kan men in allerlei gevallen toch een nooddossier opvragen van een patiënt.

### Doelstelling

Er zijn twee doelstellingen:

#### *Doelstelling 1: Afstuderen*

Met dit project tracht de afstudeerder om aan te tonen aan Fontys Hogescholen te Eindhoven dat hij diplomawaardig is voor de opleiding Software Engineering.

#### *Doelstelling 2: Wensen van de opdrachtgever vervullen*

De opdrachtgever zou graag een *proof of concept* willen zien van een nooddossier waarmee medisch specialisten snel de declaratiegeschiedenis van een patiënt kunnen zien op basis van enkele gegevens, waarvan sommige via sociale media opgevraagd kunnen worden.

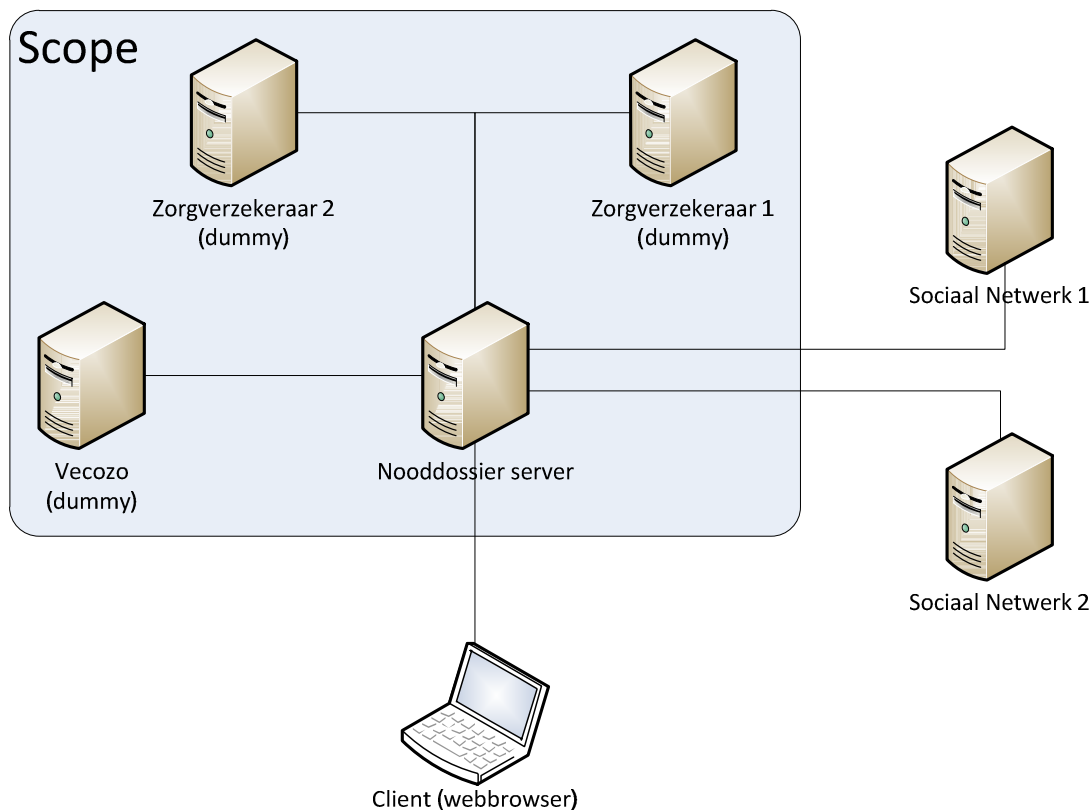
### Resultaat

- 1) succesvol afstuderen van de afstudeerkandidaat
- 2) het nooddossier wat aan de wensen en eisen van de opdrachtgever voldoet

### Effect

Een proof-of-concept van een nooddossier is ontwikkeld welke aan toont dat het mogelijk is om op basis van verschillende gegevens een declaratiegeschiedenis van een persoon te verkrijgen, wat men moet doen om dit proof-of-concept in een productieomgeving werkend te krijgen, en welke problemen men kan verwachten.

## 1.3 Afbakening



### Binnen scope:

- 1) Nooddossier-webapplicatie bouwen welke de volgende taken verricht:
  - Details van een persoon ophalen via verschillende sociale netwerken (Facebook, Twitter, LinkedIn, Hyves, Google Plus).
  - Zorgverzekeraars behorende bij een persoon ophalen bij VECOZO.
  - Declaratiegegevens van een persoon opvragen bij de verschillende zorgverzekeraars om deze vervolgens aan een medisch specialist beschikbaar te maken.
- 2) Dummy applicatie bouwen welke gegevens van VECOZO aanbiedt aan het Nooddossier. Vanwege de gevoeligheid van de informatie zal hiervoor een dummy applicatie gemaakt worden welke VECOZO's Controle op Verzekeringrecht-functionaliteit simuleert, zodat men kan werken met dummy data.
- 3) Dummy applicatie bouwen welke declaratiegeschiedenis van de verschillende zorgverzekeraars aanbiedt aan het nooddossier. Hier zijn geen standaarden voor, dus die moeten ook nog ontwikkeld worden.

### Buiten scope:

- 1) Audit-trail implementeren in het nooddossier zodat men alle informatie-aanvragen kan loggen en inzien. Dit stond wél vermeld in de originele opdrachtbeschrijving, maar men heeft uiteindelijk samen met de opdrachtgever besloten om dit buiten de scope te houden.
- 2) De certificaatbeveiliging zoals deze bij VECOZO wordt gebruikt, zal niet geïmplementeerd worden in het Nooddossier. Deze keuze is gemaakt samen met de opdrachtgever. Deze beveiliging is namelijk redelijk standaard en voor de opdracht niet erg interessant, terwijl het implementeren van deze beveiliging aardig wat tijd in beslag zal nemen.

## **1.4 Afhankelijkheden**

- 1) Er is een afhankelijkheid met de data zoals deze door verschillende Social Media ter beschikking worden gesteld. Het project is afhankelijk van het formaat waarin deze data ter beschikking wordt gesteld, en de beschikbaarheid van deze data. Denk hierbij aan een server die down gaat, of een API welke gewijzigd wordt.
- 2) Er is een afhankelijkheid met de tool geschreven door Martijn Schuurmans. Deze tool kan gebruikt worden om verschillende bestanden van verschillende bestandsformaten te genereren. Indien deze tool niet beschikbaar is, kan men eventueel zelf handmatig bestanden genereren, maar dit wordt niet aangeraden.
- 3) Er is een afhankelijkheid met de beschikbaarheid en accuraatheid van documentatie over de bestaande specificaties die gebruikt worden door onder andere VECOZO.

## **1.5 Kwaliteitseisen**

### **Kwaliteitseisen voor Doelstelling 1:**

- 1) Er dient een afstudeerscriptie gemaakt te worden. Hiervan zullen vier gedrukte exemplaren en één digitaal exemplaar ingeleverd worden. De precieze deadline voor inleveren wordt op het Fontys intranet bekend gemaakt, maar voor zover nu bekend is dient de afstudeerscriptie in de week van 4 juni ingeleverd te worden. De afstudeerscriptie dient aan de eisen gesteld door Fontys te voldoen. Deze eisen hebben met name betrekking op de vorm van de afstudeerscriptie, en kunnen op het Fontys intranet gevonden worden. Bovendien moet het Plan van Aanpak als bijlage toegevoegd worden aan de afstudeerscriptie.
- 2) Een definitief portfolio dient ingeleverd te worden. Het portfolio bestaat uit het CV, een definitieve cijferlijst, en een motivatie voor de minor-keuze.

### **Kwaliteitseisen voor Doelstelling 2:**

- 1) Het nooddossier moet gebruik maken van bestaande standaarden die binnen VECOZO en door zorgverzekeraars gebruikt worden. Het betreft hier om EI-standaarden VZ37 en VZ38, de verschillende EI-standaarden welke gebruikt worden voor declaraties, en eventuele security-standaarden die gebruikt worden binnen het VECOZO systeem.
- 2) Het nooddossier moet op zo veel mogelijk browsers kunnen draaien. De nieuwste versies van Internet Explorer, Firefox en Chrome zijn vereist, maar ondersteuning voor browsers als Opera en Safari is een pré.

## **1.6 Uitgangspunten**

Voor uitvoering van de in dit plan van aanpak beschreven delen zijn de onderstaande uitgangspunten van toepassing:

- 1) Het nooddossier moet benaderbaar worden via een webbrowser en zal dus een webapplicatie moeten worden.
- 2) Het nooddossier zal geïmplementeerd worden door gebruik te maken van verschillende technieken van Microsoft. ASP.NET, MVC3, WCF en C# spelen hier een belangrijke rol.
- 3) De back-end en front-end van het nooddossier zullen erg sterk met elkaar geïntegreerd worden. Er zal géén onderscheid gemaakt worden tussen back-end en front-end, en er zal dus ook géén API ontwikkeld worden.
- 4) Gebruikers van het nooddossier hebben zelf al beschikking tot de *controle op verzekeringsrecht*-service die door VECOZO aangeboden wordt.

## **1.7 Randvoorwaarden**

Aan uitvoering van de in dit plan van aanpak beschreven delen zijn de volgende randvoorwaarden gesteld:

- 1) Indien het nooddossier in productie genomen gaat worden, is het de bedoeling dat zorgverzekeraars een webservice aanbieden die gelijk is aan de specificaties die binnen dit project opgesteld worden voor de dummy-zorgverzekeraars.



## 2. Risicomanagement

Voor uitvoering van de in dit plan van aanpak beschreven delen zijn de onderstaande risico's onderkend. Bij de risico's zijn de oorzaken en bijbehorende maatregelen ter beheersing opgenomen.

Nr.	Risico omschrijving				
	Oorzaak	Maatregel	P/S	Wie	Wanneer
1	<b>Er is onvoldoende kennis aanwezig bij de afstudeerder om de opdracht succesvol te kunnen volbrengen</b>				
	Onvoldoende kennis van geavanceerde C# features	Cursus Advanced C# volgen (ADCSB)	P	Afstudeerder	15/16 maart
	Onvoldoende kennis van het .NET Framework	Cursus .NET Foundation Framework volgen (NETFOUNB)	P	Afstudeerder	19 t/m 22 maart
	Onvoldoende kennis van MVC	MVC in vrije tijd verder bestuderen	S	Afstudeerder	Indien nodig
	Onvoldoende kennis van WCF	WCF zelf bestuderen, of advies vragen bij de technisch begeleider; eventueel cursus	P/S	Afstudeerder en Technisch begeleider	Indien nodig
2	<b>Het systeem voldoet onverwachts niet aan de gestelde eisen of vertoont onverwachte fouten</b>				
	Sociale netwerk verandert van API of heeft downtime	Veranderingen anticiperen, defensief programmeren en meerdere sociale netwerken aanroepen	P	Afstudeerder	Tijdens ontwikkelingsfase
	Eisen kunnen niet voldaan worden vanwege technische redenen	Eisen aanpassen in overleg met de opdrachtgever	S	Afstudeerder en opdrachtgever	Indien nodig
	Bestandsgenerator-tool van Martijn Schuurmans genereert bestanden welke niet aan de specificaties voldoen	Vroegtijdig controleren of de bestanden aan de specificaties voldoen, zodat indien nodig deze bestanden handmatig aangemaakt kunnen worden en men daar niet achter komt in een (te) laat stadium	P	Afstudeerder	Voordat men de gegenereerde bestanden definitief gaat gebruiken
3	<b>De planning van het project loopt in gevaar; een bepaalde fase van het project kan niet op tijd afgerond worden</b>				
	Onvoldoende kennis over projectplanning	Cursus projectplanning en documentatie voor afstudeerders IS volgen	P	Afstudeerder	27 februari t/m 2 maart
	Een van de betrokkenen wordt ziek.	Uitloop-periode inplannen	P	Afstudeerder	Tijdens het maken van een planning
	Verkeerde inschatting gemaakt	Planning en/of prioriteiten aanpassen zodat de belangrijkste onderdelen toch gemaakt kunnen worden	S	Afstudeerder en opdrachtgever	Indien nodig

P/S = Preventief of Schadebeperkend

## **3. Aanpak**

De aanpak van uitvoering van de in dit plan van aanpak beschreven delen is in grote lijnen volgens de Scrum-methode. Voor de ontwikkeling van het product zullen 6 sprints van elk twee weken gebruikt worden.

### **3.1 Inceptiefase**

Tijdens de inceptiefase zal er gewerkt worden aan het verder verduidelijken van de opdracht. De requirements zullen vastgesteld worden, en er zal gewerkt worden aan het Plan van Aanpak. Het doel van deze fase is de verduidelijking van het project en de manier waarop het project aangepakt zal worden.

Verder zullen er tijdens de inceptiefase concrete aanvullende onderzoeksvragen opgesteld worden, die allemaal in de hoofdfase beantwoord dienen te worden.

De inceptiefase zal ongeveer 15 dagen (3 weken) duren, en eindigt wanneer het plan van aanpak geaccepteerd wordt door de opdrachtgever en technisch begeleider.

### **3.2 Hoofdfase**

De hoofdfase is de fase in het project waar concreet volgens Scrum gewerkt wordt. In de hoofdfase zullen 6 sprints doorlopen worden van elk twee weken, welke gebruikt zullen worden om een proof-of-concept van het nooddossier te realiseren.

De hoofdfase zal voornamelijk gebruikt worden om backlog-items uit te voeren. Een backlog-item is een taak die uitgevoerd dient te worden om het eindproduct te kunnen realiseren. Alle backlog-items samen vormen de backlog. De backlog kan ten alle tijden aangepast worden door de afstudeerder of opdrachtgever. Op deze manier kan men flexibel werken met veranderingen in de gestelde eisen of veranderingen aan een eerder vastgelegd ontwerp.

Aan het begin van elke sprint zal er een sprintplanning-sessie gehouden worden waarbij de afstudeerder en de opdrachtgever betrokken en aanwezig zijn. Deze sessie zal in twee delen opgesplitst zijn. In het eerste gedeelte bepalen de afstudeerder en opdrachtgever welke backlog-items er tijdens de komende sprint uitgevoerd zullen worden. In het tweede gedeelte, waar alleen de afstudeerder bij betrokken is, zal er een concrete planning gemaakt worden voor deze backlog-items.

Zoals uit de huidige backlog blijkt, zal de hoofdfase ook gebruikt worden om een onderzoek uit te voeren. Dit onderzoek is verplicht om doelstelling 1 (het afstuderen; zie 1.2) succesvol te behalen, en zal waarschijnlijk in de eerste sprint uitgevoerd worden.

Het onderzoek heeft twee belangrijke eindproducten, te weten:

- 1) Antwoorden op de onderzoeksvragen.
- 2) Kennis en informatie over de protocollen en dataformaten die in het eindproduct gebruikt dienen te worden. Deze kennis en informatie zal voornamelijk terug komen in de ontwerpdocumentatie.

De hoofdfase zelf heeft, naast de eindproducten van het onderzoek, de volgende eindproducten:

- 1) Proof-of-concept nooddossier (broncode en executables)
- 2) Ontwerpdocumentatie (functionele- en technische ontwerpen)

Deze eindproducten zullen gedurende de sprints gemaakt worden.

### **3.3 Documentatiefase (parallel met 3.2)**

In de documentatiefase zal er een afstudeerscriptie geschreven worden. De documentatiefase loopt gedeeltelijk parallel met de hoofdfase, zodat de informatie in de afstudeerscriptie geschreven wordt op het moment dat de afstudeerder bezig is met datgene dat hij aan het documenteren is.

Het resultaat van de documentatiefase is een afstudeerscriptie dat aan de door Fontys opgestelde eisen voldoet.

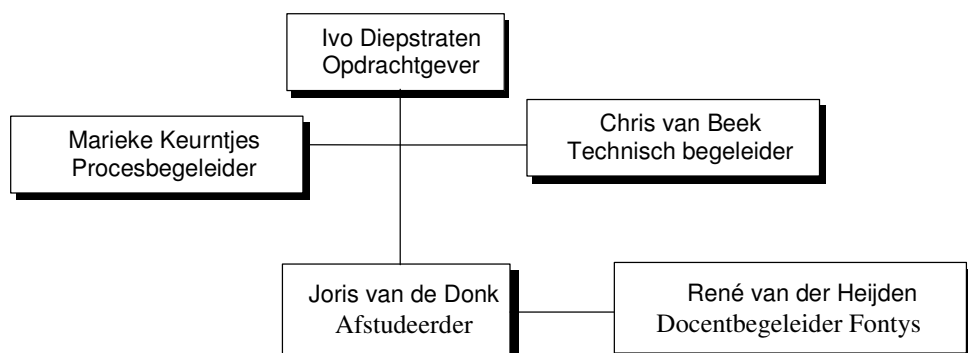
### **3.4 Eindfase**

De eindfase omvat het eind van het afstudeerproject. In deze fase zal er binnen Info Support een presentatie gegeven worden. Verder zal de afstudeerder verschillende evaluatieformulieren met betrekking tot de afstudeerstage invullen. De eindfase eindigt met een afstudeerzitting op Fontys Hogescholen te Eindhoven.

## 4. Beheersaspecten

### 4.1 Organisatie

De betrokkenen van dit project zijn weergegeven in onderstaand diagram:



### 4.2 Tijd

#### 4.2.1 Normstelling

De afstudeerperiode loopt van 30 januari 2012 tot en met 30 juni 2012. Binnen deze periode moeten 85 werkdagen (17 volledige werkweken) volgemaakt worden. Terugkomdagen van school worden wél als werkdagen geteld; ziekte, verlof en feestdagen niet.

De precieze tijdsplanning zal aan het begin van een sprint opgesteld worden aan de hand van de op dat moment geldende backlog. Hier onder staat een globale planning vermeld, zodat men inzicht heeft in de manier waarop men zal gaan werken en de manier waarop de tijd ingedeeld is:

Weeknr.	Project-week	Projectfase / onderdeel		Overige activiteiten
5	1	Inceptiefase		
6	2			
7	3			-TechDays (do, vr)
8	4			
9	5	Sprint 1	Hoofdfase Documentatiefase	-cursus planning/documentatie (ma, di, vr)
10	6			-bedrijfsbezoek 1
11	7	Sprint 2		
12	8			-ADCSB Training (do, vr)
13	9	Sprint 3		-NETFOUNB Training (ma, di, wo, do)
14	10			
15	11	Sprint 4		
16	12			
17	13	Sprint 5		
18	14			
19	15	Sprint 6		
20	16			
21	17	Uitloop		
22	18	Documentatiefase		
23	19			-scriptie inleveren
24	20	Eindfase	Eventuele laatste uitloop i.v.m. gemiste dagen	-bedrijfsbezoek 2
25	21			
26	22			-Afstudeerzitting

In de verschillende sprints zal er met mijlpalen gewerkt worden, zodat men periodiek een versie van een prototype kan opleveren. De precieze invulling van de mijlpalen kan eventueel nog veranderen, maar de algemene indeling zal zo blijven.

#### Mijlpaal 1:

- 1) Dummy controle op verzekeringsrecht-service van VECOZO welke werkt volgens de standaarden VZ37 en VZ38, en gefingeerde informatie terugstuurt op basis van enkele gefingeerde persoonsgegevens.
- 2) Nooddossier: Communicatie met social media: zoeken naar voor en achternaam, en vervolgens alle details van alle gebruikte social media weergeven in een lijst zodat de gebruiker zelf extra benodigde details kan opschrijven/onthouden.
- 3) Nooddossier: Communicatie met de dummy VECOZO webservice. Op basis van zoekcriteria als voornaam, achternaam, geboortedatum, huisnummer en postcode moeten van één persoon alle verzekeringen weergegeven worden.

#### Mijlpaal 2:

- 1) Dummy zorgverzekeraars-webservice welke de declaratiegeschiedenis van een bepaald persoon opstuurt.
- 2) Nooddossier: Integratie met de zorgverzekeraars-services in het Nooddossier. Het Nooddossier moet nu de declaratiegeschiedenis van een persoon kunnen weergeven.

**Mijlpaal 3 (could have):**

- 1) Nooddossier: Automatische integratie met social media. Indien de VECOZO-service meldt dat er niet genoeg informatie gegeven is om van een bepaald persoon de zorgverzekeraars op te kunnen halen, wordt social media slim ingezet om deze gegevens te proberen op te zoeken. De gebruiker zal zo min mogelijk gevraagd worden om input.
- 2) Nooddossier: mogelijkheid toevoegen die patiënten in staat stelt om hun gegevens in het nooddossier in te zien en aan te passen.

## **4.2.2 Voortgangscontrolle**

Periodiek zal er een voortgangscontrolle zijn. De afstudeerder zal wekelijks kijken of hij nog steeds *on schedule* is. Indien er problemen zijn van dusdanige aard dat de planning aangepast dient te worden, zal de afstudeerder dit melden bij de opdrachtgever en zal er gekeken worden hoe men de problemen het beste op kan lossen, of op wat voor manier de planning het beste aangepast kan worden. Eventueel wordt de technisch begeleider om advies gevraagd.

## **4.3 Geld**

### **4.3.1 Normstelling**

Niet van toepassing

### **4.3.2 Voortgangscontrolle**

Niet van toepassing

## **4.4 Middelen**

### **4.4.1 Normstelling**

De volgende middelen zullen van belang zijn in dit project:

- 1) Standaardwerkplek met standaardsoftware (Visual Studio, Office)
- 2) Training *projectplanning en documentatie voor afstudeerders IS* (intern)
- 3) Training *ADCSB* (intern)
- 4) Training *NETFOUNB* (intern)

De afstudeerder is zelf verantwoordelijk voor het regelen van benodigde trainingen. De procesbegeleider en technisch begeleider zullen voor sommige trainingen aangeven of deze nuttig/verplicht zijn, en kunnen assisteren bij het verder regelen van trainingen.

### **4.4.2 Voortgangscontrolle**

De afstudeerder zal één week voor aanvang van een training controleren of hij aangemeld is bij de training. Indien er door de afstudeerder om andere middelen gevraagd wordt, zal hij moeten controleren of de middelen aanwezig zijn en naar behoren functioneren.

## 4.5 Kwaliteit

### 4.5.1 Normstelling

Het belangrijkste kwaliteitsaspect binnen het Nooddossier project is dat het resulterende product voldoet aan de gestelde eisen met betrekking tot functionaliteit.

Aangezien het hier gaat om een proof-of-concept, is code kwaliteit niet zo van belang. Wel is belangrijk dat aan het Nooddossier werkt volgens de gangbare standaarden en regels, en dat het Nooddossier de standaarden zo concreet mogelijk volgt. Indien de standaarden specificeren dat bepaalde combinaties van gegevens nooit gebruikt mogen worden, dient dat ook zo te zijn in het Nooddossier.

De standaarden die zelf opgesteld worden voor de zorgverzekeraars, moeten goed gedocumenteerd worden. Goede documentatie zorgt er voor dat zorgverzekeraars in de toekomst gemakkelijk de declaratiegeschiedenis-webservice kunnen aanbieden aan medisch specialisten.

### 4.5.2 Voortgangscntrole

Aan het eind van elke iteratie zal samen met de opdrachtgever gekeken worden naar de status op dat moment van het product. De opdrachtgever kan zo commentaar geven op het product en aandachtspunten of verbeteringen aangeven die in de volgende iteratie verbeterd kunnen worden.

## 4.6 Communicatie

De volgende periodieke communicatievormen worden gebruikt:

Wat	Wie	Wanneer
<b>Voortgangsrapportage Product</b>	-Afstudeerder -Opdrachtgever -Technisch begeleider	Wekelijks op vrijdag via e-mail, beginnend vanaf de eerste sprint.
<b>Voortgangsrapportage Proces Intern</b>	-Afstudeerder -Procesbegeleider	Eens in de 2-3 weken.
<b>Voortgangsrapportage Proces Extern</b>	-Afstudeerder -Docentbegeleider	Wekelijks op vrijdag via e-mail, beginnend vanaf de eerste sprint.

De volgende éénmalige communicatievormen worden gebruikt:

Wat	Wie	Wanneer
<b>Bedrijfsbezoek 1 docentbegeleider</b>	-Afstudeerder -Docentbegeleider	9 maart 2012
<b>Bedrijfsbezoek 2 docentbegeleider (+demo presentatie)</b>	-Afstudeerder -Docentbegeleider -Opdrachtgever -Technisch begeleider	Nader te bepalen (week 24?)

De volgende periodieke overlegvormen worden gebruikt:

Wat	Wie	Wanneer
Sprintplanning	-Afstudeerder -Opdrachtgever	Aan het begin van een nieuwe sprint



**Afstudeerder:**

Joris van de Donk

[jorisdo@infosupport.com](mailto:jorisdo@infosupport.com)**Opdrachtgever:**

Ivo Diepstraten

[ivod@infosupport.com](mailto:ivod@infosupport.com)**Procesbegeleider:**

Marieke Keurntjes

[mariekek@infosupport.com](mailto:mariekek@infosupport.com)**Technisch begeleider:**

Chris van Beek

[chrisb@infosupport.com](mailto:chrisb@infosupport.com)**Docentbegeleider Fontys:**

Rene van der Heijden

[r.vanderheijden@fontys.nl](mailto:r.vanderheijden@fontys.nl)

De communicatie binnen dit project zal voornamelijk verlopen via e-mail. Eventueel kan men telefonisch contact opnemen, maar dit zal alleen in uitzonderlijke en dringende situaties voorkomen.

Indien er belangrijke beslissingen gemaakt worden zal de afstudeerder hier een notitie van aanmaken en dit opnemen in zijn eindverslag.

In onderstaande tabel is weergegeven wie welk document ter inzage en/of beoordeling krijgt:

	Afstudeerder	Opdrachtgever	Procesbegeleider	Technisch begeleider	Docentbegeleider Fontys
Plan van aanpak	O	I+F	I	I+F	I+F+B
Planning	O	I+F		I+F	
Ontwerpdocumentatie	O	I (+F)		I+F	
Afstudeerscriptie (draft)	O	I+F	I	I+F	I+F
Afstudeerscriptie (final)	O	B	I	I	B

**O:** Opstellen**I:** Inzage**F:** Feedback**B:** Beoordelen



Solid Innovator

**InfoSupport**

## Bijlage II: Onderzoeksresultaten

**Nooddossier**

## Inleiding

Dit document beschrijft het onderzoek dat gedaan is voor het Nooddossier project. Voor meer informatie over het Nooddossier project, zie het Plan van Aanpak van dit project.

## Wijzigingen

Datum	Versie	Wijzigingen
11-04-2012	1.0	Initiële versie
15-05-2012	1.1	<p>Deelvraag '2b' verandert naar '2c', en deelvraag '2b' geïntroduceerd. De nieuwe deelvraag '2b' heeft betrekking op de structuur en inhoud van de EI-berichten. De structuur van de EI-berichten was wel onderzocht ten tijde van het uitvoeren van het onderzoek, maar hier was in dit document niets over geschreven. Het stuk over de inhoud van de EI-berichten was destijds ondergebracht in de oude deelvraag '2b', maar hoort daar (in retrospect) duidelijk niet te staan.</p> <p>Om dit document leesbaarder te maken is de nieuwe deelvraag geïntroduceerd. Deze nieuwe deelvraag zorgt voor een logischere opbouw van dit document, en zorgt er ook voor dat het een en ander gemakkelijker terug te vinden is. Bovendien is dit document nu iets leesbaarder, zonder dat men voorgaande kennis over de EI-berichten hoeft te hebben.</p> <p><b>N.B.:</b> de introductie van de nieuwe deelvraag 2b heeft dus <b>niet</b> geleid tot een verandering van de <i>inhoud</i> van het gedane onderzoek. Alleen de <i>representatie</i> van de onderzoeksresultaten is hierdoor gewijzigd.</p>

## Onderzoeksplan

In het onderzoek zullen de volgende vragen beantwoord getracht te worden:

- 1) Is het mogelijk om middels Social Media (Facebook, Hyves, Twitter, LinkedIn, Google Plus) van een persoon de geboortedatum op te halen, wanneer men van deze persoon alleen de naam en het geslacht weet?

Deelvragen:

- a) Welke gegevens over gebruikers zijn beschikbaar op de verschillende sociale netwerk sites welke men kan gebruiken voor de COV-check of om een persoon te identificeren?
  - b) Met welke invoerdata kan men zoeken naar profielen bij de verschillende sociale netwerken?
  - c) Zijn er beperkingen bij het gebruik maken van de APIs behorende bij de sociale netwerk sites? Wat zijn globaal gezien de license agreements? Mag men ongelimiteerd gebruik maken van de APIs?
  - d) Wat zijn de default privacy-settings voor de verschillende sociale netwerken? Zijn velden als geboortedatum standaard publiekelijk beschikbaar?
  - e) In hoeverre is het mogelijk om gebruikers van verschillende sociale netwerken te 'koppelen' aan elkaar? Zijn er sociale netwerken die deze koppeling zelf ook al doen en beschikbaar maken aan derden?
  - f) Wat is de kans dat men op verschillende sociale netwerken van een Nederlandse gebruiker de geboortedatum op kan halen
  - g) Hoeveel Nederlanders hebben een account op de verschillende sociale netwerk sites?
- 2) Is het mogelijk om uit de declaratiegegevens van de zorgverzekeraars informatie te halen die voor een medisch specialist interessante informatie bevat?

Deelvragen:

- a) Welke EI-standaarden zijn er voor declaraties en door wie worden ze gebruikt?
- b) Wat voor gegevens staan er in de EI-standaarden en hoe zijn deze gestructureerd?
- c) Welke data in de EI-standaarden moeten geconverteerd worden zodat medisch specialisten ze kunnen begrijpen?

## Waarom deze vragen gesteld worden

Het onderzoek wordt verricht om drie redenen:

- 1) Fontys Hogescholen stelt als eis aan het afstudeertraject dat er ook een onderzoek verricht wordt
- 2) Om de planning en prioritering van het Nooddossier project goed in te kunnen vullen is aanvullende informatie nodig over de haalbaarheid van verschillende onderdelen binnen het project.
- 3) Wanneer men de software van het nooddossier gaat ontwerpen, is kennis nodig over de verschillende standaarden en APIs die gebruikt dienen te worden. Tijdens het onderzoekstraject zal deze kennis gedeeltelijk al opgedaan worden.

De individuele vragen zullen om verschillende redenen gesteld worden:

Vraag 1 wordt gesteld om te achterhalen wat de mogelijkheden zijn van het Social Media gedeelte van het Nooddossier project. Wanneer vraag 1 beantwoord is, weet men in hoeverre het social media aspect van het Nooddossier project technisch haalbaar is, en kan men de risico's beter inschatten. Ook heeft men wat meer informatie over de technische invulling van het social media gedeelte van het project.

Vraag 2 wordt gesteld om te achterhalen in hoeverre het mogelijk is om uit de declaratiebestanden nuttige informatie te halen. Deze vraag zal inzicht en informatie opleveren over de verschillende EI-standaarden.

## 1a: Welke gegevens over gebruikers zijn beschikbaar op de verschillende sociale netwerk sites welke men kan gebruiken voor de COV-check of om een persoon te identificeren?

De verschillende sociale netwerk sites (SNSs) bevatten allerlei verschillende data over hun gebruikers. Alle sociale netwerk sites waar men in het Nooddossier project mee gaat werken, stellen een API beschikbaar welke applicatieprogrammeurs kunnen gebruiken om data over gebruikers op deze sites op te kunnen vragen. Voor het nooddossier project is het van belang dat er enkele gegevens beschikbaar zijn van deze gebruikers. Om de COV-check te kunnen doen, is bijvoorbeeld de geboortedatum van een persoon of zijn postcode plus huisnummer nodig.

De volgende tabel laat zien wat voor data er op de verschillende sociale netwerken beschikbaar is. Bovendien wordt aangegeven welke data er middels APIs opgevraagd kan worden. Deze tabel is gebaseerd op de API-documentatie welke door de verschillende SNSs beschikbaar gesteld wordt. Ter verificatie heeft men vervolgens calls gemaakt naar de verschillende APIs om te controleren of de documentatie wel klopt. Let op: deze tabel laat *niet* zien of dat de data in de meeste gevallen ook echt beschikbaar is. Gebruikersinstellingen met betrekking tot privacy kunnen er voor zorgen dat bepaalde gegevens niet beschikbaar zijn.

	Achternaam	Geslacht	Geboortedatum	Postcode + huisnummer	Woonplaats	Profielfoto	Foto's	Werkgeschiedenis
Facebook [1]								
Hyves [2]								****
Twitter [3]	*				**			
LinkedIn [4]					Via postcode			
Google Plus [5]			***		**			

### Legenda

	Gegevens beschikbaar via API
	Gegevens niet beschikbaar via API
	Gegevens (eventueel) afleidbaar van een ander gegevensveld.

\* - Bij Twitter geven gebruikers niet concreet hun voor- en achternaam op. In plaats daarvan kan men 'naam' invullen. Deze naam kan dus een fictieve naam ('nickname') zijn, of de echte naam van de gebruiker. Men zal dus uit het 'naam' veld de achternaam van een gebruiker kunnen halen.

\*\* - De beschikbaarheid van deze velden hangt af van de input van de gebruiker. De gebruiker kan hier zelf een regio of een plaatsnaam invullen. Indien de gebruiker een regio invult ("Noord-

Brabant”) is de woonplaats dus niet beschikbaar, maar als een gebruiker een plaatsnaam invult (“Eindhoven”), is dat wel het geval.

\*\*\* - Alhoewel Google Plus een geboortedatum van een persoon beschikbaar lijkt te stellen volgens de API-documentatie, is er geen enkele mogelijkheid om deze gegevens van jezelf of anderen te bekijken. Het lijkt er op dat deze gegevens niet beschikbaar gesteld worden aan eindgebruikers. In één geval was het mogelijk om van een gebruiker de geboortedag en –maand op te vragen.

\*\*\*\* - Via Hyves kan men meerdere bedrijven of verenigingen toevoegen aan een account, maar men kan niet aangeven wat voor functie men binnen een bedrijf voldoet, of wanneer men binnen het bedrijf gewerkt heeft. De link tussen persoon en bedrijf is dus niet helemaal duidelijk, maar dat de link bestaat kan wel kenbaar gemaakt worden.

[1] – <http://developers.facebook.com/docs/reference/api/user/>

[2] – <http://www.hyves-developers.nl/documentation/data-api/methods/2.1/users.get>

[3] – <https://dev.twitter.com/docs/api/1/get/users/show>

[4] – <http://developer.linkedin.com/documents/profile-fields>

[5] – <https://developers.google.com/+/api/latest/people>



## 1b: Met welke invoerdata kan men zoeken naar profielen bij de verschillende sociale netwerken?

Om effectief en gericht te kunnen zoeken naar een persoon, is het van belang dat je verschillende invoervelden kan gebruiken in je zoek-query. Wanneer je weet dat een persoon die je zoekt Jan heeft, en in Amsterdam woont, zou je een stuk gericht kunnen zoeken wanneer je beide gegevens in je zoek-query opneemt, dan wanneer je alleen de voornaam gebruikt.

Onderstaande tabel laat zien met welke gegevens je kan zoeken op de verschillende sociale netwerken. Indien anders aangegeven, kan men kiezen voor elke willekeurige combinatie van gegevens.

	Voor- en/of achternaam	Plaats	Postcode	Bedrijf
Facebook [6]	X			
Hyves [7]	X*	X*		
Twitter [8]	X			
LinkedIn [9]	X**	X***	X	X
Google Plus [10]	X			

\* - Binnen Hyves kan men zoeken op naam en plaats, maar men gebruikt altijd dezelfde API call. Dit heeft als nadelig effect dat wanneer men zoekt naar mensen uit Uden, men ook mensen vindt die als achternaam bijvoorbeeld "van Uden" hebben. Zoekt men echter een specifiek persoon ("Jan Klaassen") uit Uden, dan kan men zoeken op "Jan Klaassen Uden" en men zal dan deze specifieke persoon vinden.

\*\* - Men kan apart op voor- en achternaam zoeken binnen LinkedIn.

\*\*\* - Zoeken via woonplaats is niet mogelijk, maar men kan wel zoeken via postcode+afstand.

[6] – <http://developers.facebook.com/docs/reference/api/>

[7] – <http://www.hyves-developers.nl/documentation/data-api/methods/2.1/users.search>

[8] – <https://dev.twitter.com/docs/api/1/get/users/search>

[9] – <http://developer.linkedin.com/documents/people-search-api>

[10] – <https://developers.google.com/+/api/latest/people/search>

## 1c: Zijn er beperkingen bij het gebruik maken van de APIs behorende bij de sociale netwerk sites? Wat zijn globaal gezien de license agreements? Mag men ongelimiteerd gebruik maken van de APIs?

### Twitter

Twitter heeft over het algemeen zeer specifieke technische beperkingen [11].

Voor Twitter's REST API kan men in totaal 350 requests per uur maken. De user search REST API is gelimiteerd op 60 requests per uur[11b]. Elke call naar de user search API zal bovendien tellen voor de globale rest API request limits. Er is geen mogelijkheid om ge-whitelist te worden of om extra resources te krijgen van Twitter. Het was vroeger wel mogelijk om door Twitter ge-whitelist te worden, maar sinds Februari 2011 is dit niet meer mogelijk[12]. Sindsdien kan men voor high-volume datatoegang bij Gnip terecht, een *authorized reseller* van Twitter data.

Verder zijn er ook enkele legale beperkingen waar men rekening mee moet houden.

Zo moet men de 'Developer Rules of the Road'[13] kennen en toepassen. Deze regels beschrijven in het kort wat men met de Twitter APIs mag doen en hoe men met Twitter als bedrijf, Tweets en gebruikers van Twitter moet om gaan.

De volgende belangrijke punten staan in de Developer Rules of the Road:

- Indien men een API aanbiedt aan derden welke content van Twitter beschikbaar maakt, mag men alléén de Twitter ID-nummers versturen, niet de content van Twitter zelf.
- Men mag geen data van Twitter opslaan in een database.
- Men moet de vier hoofdprincipes in acht nemen: "Verras eindgebruikers niet", "Creëer of distribueer geen spam", "Respecteer de privacy van gebruikers" en "Wees een goede partner van Twitter".
- Men moet een Privacy Policy hebben, waarin duidelijk vermeld staat wat er gedaan wordt met informatie van gebruikers.

[11] – <https://dev.twitter.com/docs/rate-limiting>

[11b] – <https://dev.twitter.com/docs/api/1/get/users/search>

[12] – [http://www.readwriteweb.com/archives/twitter\\_kills\\_the\\_api\\_whitelist\\_what\\_it\\_means\\_for.php](http://www.readwriteweb.com/archives/twitter_kills_the_api_whitelist_what_it_means_for.php)

[13] – <https://dev.twitter.com/terms/api-terms>

## Facebook

De APIs van Facebook hebben geen officiële rate limits. Verschillende developers melden echter dat alle API calls toch wel een rate limit hebben, namelijk van 600 calls per 600 seconden[14].

Verder moet men voor het gebruik van het Facebook Platform de Facebook Platform Policies[15] in acht nemen. Deze policies leggen uit wat men met het gehele Facebook Platform als applicatie-developer mag doen.

Voor het gebruik van het Facebook Platform gelden de volgende algemene principes:

- Men moet een goede *user experience* creëren. Dit houdt in dat men sociale en “engaging” applicaties moet ontwikkelen welke gebruikers keuzes en controle geven en ze de kans geven om expressieve en relevante content te delen.
- Men moet betrouwbaar zijn. Privacy moet gerespecteerd worden, men mag gebruikers niet misleiden of verrassen, en men mag geen spam versturen.

Verder zijn er enkele specifieke relevante policies:

- Men moet een expliciete uitlogfunctie implementeren indien gebruikers in je applicatie inloggen bij Facebook.
- Men moet een privacy policy hebben welke aangeeft hoe men data van gebruikers gebruikt, weergeeft, deelt en verstuurd. Deze privacy policy moet in de applicatie benaderbaar zijn.
- Men mag geen data van gebruikers opslaan tenzij gebruikers de mogelijkheid hebben om deze data op verzoek te laten verwijderen.
- Men moet een commercieel billijke geo-filtering dienst gebruiken om je content af te schermen aan mensen waar je het niet mag aanbieden. Voor het Nooddossier houdt dit dus in dat men de applicatie moet afschermen voor niet-Nederlanders.

[14] – <http://forum.developers.facebook.net/viewtopic.php?id=56950>

[15] – <http://developers.facebook.com/policy>

## Hyves

Binnen Hyves wordt er gebruik gemaakt van rate limits. Desktopapplicaties welke de Hyves API aanspreken hebben vanuit de API een lagere limiet dan webapplicaties. De exacte rate limits worden nergens aangegeven, maar wanneer men een limiet overschreden heeft krijgt men wel een foutmelding terug welke dit vermeldt.

Wanneer men een Hyves-applicatie ontwikkeld, dient men de Terms of Use[16] te accepteren. In de Terms of Use staat beschreven wat men allemaal mag, en wat er van een developer verwacht wordt.

Enkele relevante regels uit de Terms of Use zijn:

- Men mag geen onderdeel of service van Hyves vervangen of dupliceren.
- Men mag de identiteit van een gebruiker of het doel van de applicatie niet verbergen.
- Men mag content van Hyves niet langer opslaan dan nodig is, met een maximum van 24 uur, met uitzondering van de 'identificatie velden' uit de API-documentatie en de timestamp welke aangeeft wanneer bepaalde velden voor het laatst geüpdate zijn.
- Men moet de volgende tekst ergens opnemen in de helptekst van de applicatie: 'This application uses the Hyves API but has not been developed, certified or otherwise approved on behalf of or on the instructions of Hyves.'
- Men moet de Hyves privacy policy[17] gebruiken en implementeren binnen de applicatie.

[16] - <http://hyves.nl/api/termsfuse>

[17] - <http://www.hyves.nl/index.php?l1=ut&l2=pr&l3=hm>

## Google Plus

De Google Plus API heeft bepaalde rate limits. Zo is er een per-user limiet van 5 requests per seconde, en is er een zogenaamde 'courtesy limit' van 10000 queries per dag. Google biedt echter wel de mogelijkheid aan om extra API capaciteit aan te vragen of te kopen.

Voor het gebruik maken van de Google Plus APIs moet men met de Terms of Service[18] rekening houden. De Terms of Service verwijzen naar andere documenten, zoals de Platform Developer Policies[19]. Ook deze documenten behoren tot de collectie regelgeving-documenten waar men als Google Plus developer rekening mee moet houden.

De Platform Developer Policies beschrijven in het kort een aantal guidelines en policies waar men als developer rekening mee moet houden. De volgende algemene guidelines zijn van kracht:

- De gebruiker is het belangrijkste. Maak iets dat Google+ gebruikers leuk zullen vinden, en geef ze betekenisvolle keuzes over met wie ze dingen delen.
- Wees transparant. Wees eerlijk over de bedoelingen van je applicatie. Laat gebruikers zien wat de applicatie aan het doen is en vraag de gebruikers expliciet om toestemming.
- Respect de data van gebruikers. Houd de persoonlijke (private) informatie van gebruikers persoonlijk en verspreid deze niet. Gebruik data om de user experience te verbeteren, niet te verslechteren.

De volgende specifieke policies zijn ook van kracht:

- Bied gebruikers een privacy policy aan, houd deze up-to-date, en geef gebruikers een waarschuwing vooraf indien je de privacy policy gaat wijzigen.
- Verzamel, deel of sla geen gevoelige persoonlijke informatie op over gebruikers. Denk bij dit soort informatie aan credit card gegevens en sofi-nummers.
- Verleen geen toegang tot de persoonlijke informatie van een gebruiker aan *third parties* zonder toestemming van de gebruiker.
- Gebruik geen oude data. Je mag data cachen, maar het is de bedoeling dat je indien mogelijk verse data van de APIs gebruikt. Indien de APIs aangeven dat bepaalde data verwijderd is, moet je je eigen opgeslagen data óók verwijderen.
- Geef de gebruiker de mogelijkheid om zijn data op je systemen te verwijderen (inclusief accounts die gekoppeld zijn aan een Google profiel/account). Voor non-persoonlijk-identificeerbare informatie hoeft dit niet.
- Je mag de gebruiker geen valse informatie geven over het doel of de werking van je applicatie.

[18] – <https://developers.google.com/+/terms>

[19] – <https://developers.google.com/+/policies>

## LinkedIn

LinkedIn heeft zeer specifieke API rate limits. Zo heeft een applicatie bepaalde limits, maar hebben gebruikers zelf ook limits. Dagelijks kan een applicatie bijvoorbeeld 100000 'people search' calls maken, en kunnen gebruikers van deze applicatie 100 van deze calls maken. Andere calls hebben andere limits en worden ook apart geteld. Voor meer informatie, zie [20]. Om extra capaciteit aan te vragen, zie [21]. Er zijn geen specifieke technische beperkingen waar men echt rekening mee moet houden in het Nooddossier.

LinkedIn heeft een uitgebreid Terms of Use[22] document voor de LinkedIn APIs. Dit document beschrijft de regels die developers in acht moeten nemen wanneer ze gebruik gaan maken van de APIs. Het Terms of Use document maakt referenties naar onder andere de Platform Guidelines[23], die ook in acht genomen moeten worden.

De volgende relevante regels gelden er voor de LinkedIn APIs en het LinkedIn Platform:

- Men mag geen applicaties schrijven en deze vervolgens distribueren naar andere bedrijven. Indien men dit van plan is, dan moet men contact opnemen met LinkedIn's *Enterprise Relations Team*. Men mag wel intern een applicatie draaien, maar zodra men deze applicatie distribueert naar anderen gelden andere terms of use.
- Geen data van LinkedIn mag opgeslagen worden, tenzij de gebruiker hier expliciete toestemming voor heeft gegeven middels een door LinkedIn aangeleverde user interface. Member IDs zijn uitgesloten van deze regel.
- Men mag de data van een LinkedIn account niet aanleveren aan een andere LinkedIn gebruiker die geen deel uitmaakt van hetzelfde 'netwerk', tenzij deze data publiekelijk is. Persoonlijke data mag niet publiekelijk gemaakt worden.
- Men mag geen API access aanbieden aan gebruikers.
- Men mag geen features creëren die de professionele reputatie van LinkedIn gebruikers schaadt.
- Men mag geen LinkedIn data naar derden versturen. De enige persoon die de data mag zien, is de gebruiker van je applicatie.
- Wanneer je data aan een gebruiker toont die je van LinkedIn gehaald hebt, moet je duidelijk maken aan de gebruiker dat je deze data van LinkedIn gehaald hebt door middel van tekst of een icoontje in de linker-bovenhoek van het gegevens-element. Bovendien moet je een link toevoegen aan de data, zodat de gebruiker de data op LinkedIn zelf kan bekijken.
- Wanneer je een bedrijf weergeeft, moet je de gebruiker de mogelijkheid geven om dit bedrijf op LinkedIn te volgen.
- Men mag geen data van LinkedIn uitleveren aan derden.

-Men mag de APIs niet gebruiken voor iets waarbij het falen van de APIs kan leiden tot de dood of verwonding.

-Men moet een privacy policy en/of een user agreement hebben, en deze duidelijk aan de gebruikers laten zien.

[20] - <https://developer.linkedin.com/documents/throttle-limits>

[21] - <https://developer.linkedin.com/themes/linkedin-home/form-api.html>

[22] - <https://developer.linkedin.com/documents/linkedin-apis-terms-use>

[23] - <https://developer.linkedin.com/documents/linkedin-platform-guidelines>

## Samenvatting

In onderstaande tabellen staan enkele belangrijke requirements schematisch weergegeven. Voor meer informatie, zie de voorgaande paragrafen over de verschillende sociale netwerk sites.

SNS	Data mag opgeslagen worden in een database	Applicatie moet een privacy policy hebben voor eindgebruikers
Facebook	Nee, tenzij gebruikers ingelicht worden en ze de mogelijkheid hebben om deze data op verzoek te laten verwijderen.	Ja
Hyves	Alleen identificatie-velden en timestamps. De rest mag maximaal 24 uur opgeslagen worden als caching.	Nee, maar men moet wel de Hyves privacy policy implementeren.
Twitter	Nee	Ja
LinkedIn	Alleen member IDs. Andere gegevens moet de gebruiker expliciet toestemming voor verlenen.	Ja
Google Plus	Ja, als caching. Wanneer de APIs zeggen dat iets verwijderd is, moet je dat zelf ook verwijderen.	Ja

SNS	Globale rate limit over alle calls heen	User search API call rate limit	Whitelisting of extra API calls mogelijk (offici�el)
Facebook	Volgens developers: 600 requests per 10 minuten	Onbekend	Nee.
Hyves	50000 req/dag (per applicatie)	Aanwezig, maar niet gedocumenteerd.	Op aanvraag.
Twitter	350 req/uur (per gebruiker)	60 req/uur (per gebruiker)	Nee. Koop data in bij <i>Gnip</i> , een reseller van twitter data.
LinkedIn	Geen	Per applicatie: 100000 req/dag. Per gebruiker: 100 req/dag.	Op aanvraag.
Google Plus	10000 per dag, 5 per seconde.	Geen	Op aanvraag of in te kopen.



## 1d: Wat zijn de standaard privacy-instellingen voor de verschillende sociale netwerken? Zijn velden als geboortedatum standaard publiekelijk beschikbaar?

De standaard privacy-instellingen van een sociaal netwerk zijn een redelijk goede indicatie over wat voor instellingen men kan verwachten bij een groot gedeelte van de gebruikers van het sociale netwerk in kwestie. Privacy-instellingen kunnen erg complex en onoverzichtelijk zijn, vooral voor gebruikers die de web-interface van de SNS sowieso al lastig vinden.

Onderstaande tabel laat zien wat de standaard privacy-instellingen zijn bij verschillende SNSs op verschillende gebieden. Des te donkerder de kleur van een cel in de tabel is, des te minder mensen toegang hebben tot het gegeven op de SNS.

	Achternaam	Geboortedatum	Postcode + huisnum	Geslacht	Profielfoto	Foto's	Woonplaats	Werkgeschiedenis
Facebook [24]	Iedereen	Vrienden van vrienden	n.v.t.	Iedereen	Iedereen	Iedereen	Iedereen	Iedereen
Hyves [25]	Hyvers*	Hyvers*	n.v.t.	Hyvers*	Iedereen*	Vrienden	Hyvers*	Hyvers*
Twitter [26]	Iedereen**	n.v.t.	n.v.t.	n.v.t.	Iedereen	n.v.t.	Iedereen	n.v.t.
LinkedIn	Iedereen	Directe connecties***	Niemand	n.v.t.	Iedereen	n.v.t.	Iedereen	Iedereen
Google Plus	Iedereen	n.v.t.	n.v.t.	Iedereen	Iedereen	Circles	Iedereen	Iedereen

\* - De zichtbaarheid van gegevens is voor mensen onder de 16 op Hyves standaard ingesteld op "Alleen Vrienden", maar dit kan vervolgens wel aangepast worden. [25]

\*\* - Twitter heeft eigenlijk geen namen voor personen. Bij Twitter kan men ook alleen een voornaam of een nickname gebruiken. Toch ziet men vaak dat mensen hun volledige naam gebruiken in hun Twitter profiel.

\*\*\* - Binnen LinkedIn specificeert men zowel het geboortjaar als de verjaardag (combinatie geboortemaand- en dag). Het geboortjaar is standaard zichtbaar voor directe connecties; de verjaardag is standaard zichtbaar voor directe connecties (1stegraads), connecties van je eigen directe connecties (2egraads), en connecties van de connecties van je eigen directe connecties (3egraads). De volledige geboortedatum is dus standaard alleen zichtbaar voor directe connecties.

[24] – <http://mattmckeeon.com/facebook-privacy/>

[25] - <http://www.hyves.nl/index.php?l1=ut&l2=pr&l3=hm>, hoofdstuk 8

[26] – <http://support.twitter.com/articles/14016>: "When you sign up for Twitter, you have the option of keeping your Tweets public (the default account setting) or protecting your Tweets. [...] Accounts with public Tweets have profile pages that are visible to everyone."

## **1e: In hoeverre is het mogelijk om een persoon te herkennen en te identificeren op verschillende sociale netwerken? Zijn er sociale netwerken die deze koppeling zelf ook al doen en beschikbaar maken aan derden?**

Voor het Nooddossier is het van belang dat men van een persoon de profielen op verschillende sociale netwerken kan weergeven.

Er zijn enkele manieren om een gebruiker te herkennen binnen meerdere sociale netwerken. Helaas zijn een aantal van deze manieren te geavanceerd om in het Nooddossier op te nemen, aangezien ze gebruik maken van geavanceerde data-processing technieken.

De gemakkelijkste manier om user-accounts op verschillende SNSs aan elkaar te koppelen, is om binnen een gebruikersprofiel te zoeken naar referenties naar andere SNSs. Google Plus bijvoorbeeld laat gebruikers willekeurige URLs aan hun profiel koppelen. Soms zie je hierbij ook referenties staan naar andere sociale netwerken. Vooral Facebook en Twitter komen redelijk vaak voor in Google Plus profielen. Binnen LinkedIn kan men ook twitter-accounts toevoegen.

Een andere manier is om te kijken naar de gegevens van de gebruiker. Wanneer een gebruiker een profiel heeft op verschillende SNSs, is het waarschijnlijk dat hij op alle netwerken dezelfde informatie invult. Op basis van de data op de verschillende profielen kan men bepalen hoe waarschijnlijk het is dat profielen behoren bij dezelfde persoon. Sommige van deze data kan door een applicatie geanalyseerd worden (naam, bedrijf, vrienden). Andere data, zoals foto's en video's, kunnen te complex zijn om door een computer te laten analyseren. Een mens kan echter erg goed en accuraat een persoon herkennen in meerdere foto's op verschillende SNSs.

Verder kan men gebruik maken van de vrienden van een persoon. Als men twee profielen bekijkt op twee verschillende SNSs, en de relaties lijken hetzelfde te zijn, dan is men waarschijnlijk twee profielen aan het bekijken die behoren bij dezelfde persoon. Hetzelfde geldt voor de interesses en de werkgeschiedenis: als deze gelijk zijn op twee profielen, dan is het aannemelijk dat men naar twee profielen aan het kijken is welke bij dezelfde persoon horen.

Het is dus zeker mogelijk om dezelfde persoon te herkennen in meerdere sociale netwerken. Men kan gebruik maken van tekstuele data als naam en woonplaats, maar ook relaties tussen meerdere personen. Des te meer data beschikbaar is, des te gemakkelijker het wordt om met enige zekerheid te zeggen dat twee profielen bij dezelfde persoon horen. Mensen kunnen deze herkenning gemakkelijker doen dan computerprogramma's, omdat ze veel beter zijn in het herkennen van gezichten en patronen.

## 1f: Wat is de kans dat men op verschillende sociale netwerken van een Nederlandse gebruiker de geboortedatum op kan halen?

Voor het Nooddossier project is het van belang dat men weet hoe groot de kans is dat men van een Nederlander die op een bepaalde sociale netwerk site een account heeft, de geboortedatum op kan vragen. Deze kennis is van belang om te kunnen bepalen welke sociale netwerken het nuttigst zijn om support voor aan te bieden.

Deze deelvraag zal voor elk sociaal netwerk beantwoord worden.

### Facebook

Facebook heeft een netjes gedocumenteerde API voor het ophalen van gebruikersinformatie. Deze API, de 'Graph API' genoemd, stelt developers in staat om de zogenaamde *social graph* te inspecteren. De API kan zowel geauthenticeerd als niet-geauthenticeerd aangeroepen worden, maar de resultaten die men krijgt hangen af van de authenticatie; publiekelijk beschikbare informatie kan men altijd opvragen, maar niet-publiekelijke informatie kan men niet opvragen tenzij de geauthenticeerde gebruiker daar toegang toe heeft.

Facebook heeft een handig hulpprogramma, de Graph API Explorer (<http://developers.facebook.com/tools/explorer>) welke een developer de mogelijkheid geeft om zonder problemen geauthenticeerde en non-geauthenticeerde calls te maken naar de Graph API.

Het onderzoek naar Facebook is verricht door gebruik te maken van de Graph API Explorer. Geauthenticeerde calls zijn gemaakt naar de search-functionaliteit om te zoeken naar gebruikers welke in hun naam de tekst 'de Boer' hebben. Vervolgens is er handmatig door de profielen gezocht (ook via de Graph API) om te kijken of de geboortedatum publiekelijk was.

Na 125 resultaten doorzocht te hebben, bleek dat er in slechts één geval een geboortedatum beschikbaar was; minder dan 1% van de gevallen. Een handmatige zoektocht via de Facebook website zelf (dus niet via de API) gaf hetzelfde resultaat.

Men kan dus concluderen dat men via Facebook in **minder dan 1%** van de gevallen van een gebruiker de geboortedatum kan opvragen.

### Hyves

Hyves heeft een redelijk simpele API waarbij men gemakkelijk naar gebruikers kan zoeken en in één keer een XML kan ontvangen waarin alle gegevens van de gebruikers in staan. Indien men ingelogd is, kan men van een aantal gebruikers de geboortedatum opvragen (zie deelvraag 1d).

Deze XML kan men gemakkelijk analyseren met behulp van XPath. XPath is een query language waarmee men door middel van het opstellen van een bepaalde query een aantal nodes in een XML document kan selecteren. In de query kan men specificeren aan welke voorwaarden de te selecteren

nodes moeten voldoen, bijvoorbeeld of dat een bepaald attribuut een bepaalde waarde moet hebben.

Een aantal XML-bestanden welke gebruikersdata van Hyvers bevatten zijn geanalyseerd middels XPath. In totaal gaat het om 5000 Hyvers die allemaal gevonden waren wanneer men op 'de boer' zocht. Door een XPath query op te stellen en het aantal geselecteerde nodes te tellen, kan men redelijk simpel deze documenten analyseren.

Om te bepalen hoeveel Nederlandse Hyvers er in de dataset aanwezig zijn, moet men de niet-nederlanders er eerst uit filteren. Hyves kent een systeem waarbij gebruikers het land waarin zij wonen kunnen aangeven. Elk land krijgt vervolgens in het Hyves-systeem een identificatienummer, welke in Hyves het 'countryid' genoemd wordt. Door een XPath query te creëren welke alleen zoekt naar Hyvers welke voor 'countryid' het Nederlandse countryid ingevuld hebben, kan men alle buitenlanders wegfilteren.

Binnen Hyves heeft Nederland het ID "00da0d7e3a352207e824bf85791df6cbcb". Hieruit kan men de volgende XPath query opstellen: `//user[countryid="00da0d7e3a352207e824bf85791df6cbcb"]`

Wanneer men deze query draait over de volledige dataset, zal men er achter komen dat er in totaal 4953 gebruikers uit Nederland in de dataset aanwezig zijn.

Hyves splitst de geboortedatum op in twee delen: het geboortjaar, en de verjaardag. De verjaardag is een combinatie van geboortedag en geboortemaand. Zowel de verjaardag als het geboortjaar kunnen afzonderlijk publiekelijk gemaakt worden, dus het kan zo zijn dat men van een gebruiker wel de geboortedag en maand weet, maar niet het geboortjaar. Ook kan het zo zijn dat men van een gebruiker het geboortjaar wel weet, maar vervolgens niet de geboortedag en maand.

Voor het nooddossier is zowel de dag, maand als jaar nodig. Met een XPath query kunnen we controleren of voor een gebruiker deze gegevens aanwezig zijn. De volgende query kan men hiervoor gebruiken:

`//user[countryid="00da0d7e3a352207e824bf85791df6cbcb"][[birthday/year/node()]][[birthday/month/node()]][[birthday/day/node()]]`

Wanneer men deze query draait, krijgt men in totaal 3864 gebruikers waarvan zowel de jaar, maand en dag van de geboortedatum beschikbaar zijn.

3864 van 4953 gebruikers staat gelijk aan 78% van de gebruikers. Op Hyves is dus van **78%** van de geregistreerde gebruikers de volledige geboortedatum bekend.

## Twitter

Aangezien Twitter geen geboortedatum weet van gebruikers, zal men in 0% van de gevallen van een Twitter gebruiker de geboortedatum kunnen opvragen. De enige manier waarop men wellicht van een Twitter gebruiker de geboortedatum kan achterhalen, is om zijn tweets te analyseren om te kijken of hij wellicht ooit een referentie naar zijn verjaardag gemaakt heeft. Dit valt echter buiten de scope van dit onderzoek, en dus kan men zeggen dat in **0%** van de gevallen men op Twitter van een gebruiker de geboortedatum kan opvragen.

## LinkedIn

LinkedIn geeft aan dat het mogelijk is om van gebruikers de geboortedatum op te vragen, maar standaard is deze datum alleen beschikbaar voor directe connecties. Het is dus de vraag hoeveel mensen hun instellingen aangepast hebben om hun geboortedatum publiekelijk te maken.

Middels de Developer REST Console (<https://developer.linkedin.com/rest-console>) is een handmatige test gedaan. Er is gezocht op LinkedIn gebruikers met als achternaam 'de Boer' en 'Janssen' (voorbeeld-URL: [http://api.linkedin.com/v1/people-search:\(people:\(id,date-of-birth,first-name,last-name\)\)?last-name=janssen&count=20&start=0](http://api.linkedin.com/v1/people-search:(people:(id,date-of-birth,first-name,last-name))?last-name=janssen&count=20&start=0)). Men heeft ingelogd met een gebruiker zonder connecties.

Van de 220 mensen die gevonden waren, hadden 26 hun geboortedag en geboortemaand publiekelijk gemaakt. 2 mensen hadden geboortetejaar publiekelijk gemaakt. 1 iemand had geboortedag, geboortemaand, en geboortetejaar publiekelijk gemaakt.

Op LinkedIn kan men dus in **minder dan 1%** van de gevallen van een gebruiker de geboortedatum ophalen. Waarschijnlijk is dit een resultaat van de default privacy instellingen op LinkedIn. Deze zijn redelijk goed gekozen, en het is aannemelijk dat veel mensen deze instellingen niet aanpassen.

## Google Plus

De documentatie van de Google Plus APIs zegt dat het mogelijk is om van een gebruiker de geboortedatum op te vragen. Een eerdere handmatige test lijkt echter aan te geven dat men *nooit* van een gebruiker de geboortedatum op kan vragen.






Een goed uitgewerkte test is dus gewenst om te bepalen of dit echt zo is. Middels programmacode welke op Google Plus zoekt naar gebruikers met de achternaam 'de Boer' is getracht om deze vraag goed te beantwoorden. Om de kans van het vinden van een geboortedatum te maximaliseren, heeft men ingelogd met een bestaande Google Plus gebruiker en is de officiële Google Plus Java API Library gebruikt om naar geboortedatums te zoeken.

Uiteindelijk heeft men van 373 gebruikers slechts 1 keer een geboortedatum kunnen ontvangen. De enige geboortedatum die men heeft kunnen ontvangen, bevatte echter geen geboortetejaar. Uiteindelijk kan men dus concluderen dat men in **0%** van de gevallen op Google Plus van een gebruiker een geboortedatum (jaar + maand + dag) kan opvragen.

Verder lijkt het ook onmogelijk te zijn om als Google Plus gebruiker je eigen geboortedatum in te zien, of om de privacy-instellingen van het weergegeven van je geboortedatum aan te passen.

## Conclusie

De volgende tabel beschrijft de kans dat men automatisch op een bepaalde SNS van een bestaande gebruiker de geboortedatum op kan vragen:

SNS	Percentage Nederlandse gebruikers (ongeveer) waarvan men de geboortedatum kan opvragen	Sample size
 Facebook	<1% (indien ingelogd)	125
 Hyves	78% (indien ingelogd)	4953
 Twitter	n.v.t.	n.v.t.
 LinkedIn	<1% (indien ingelogd)	110
 Google Plus	0% (indien ingelogd)	373

Natuurlijk kan men ook handmatig door gebruikersprofielen bladeren op zoek naar berichten of foto's die aangeven dat een gebruiker op een bepaalde datum jarig is geweest, en hoe oud hij of zij is geworden.

## Addendum

Indien men op een SNS geen geboortedatum kan ophalen, kan men er altijd nog voor kiezen om het gebruikersprofiel handmatig te doorzoeken naar informatie. Veel gebruikers zullen bijvoorbeeld een post maken op hun profiel waarin ze melden dat ze jarig zijn en hoe oud ze zijn geworden, of vrienden kunnen deze gebruiker feliciteren met hun verjaardag, of ze kunnen een foto plaatsen van een verjaardagstaart waarop staat hoe oud iemand geworden is. Op deze manier is het wellicht toch mogelijk om een indicatie te krijgen van de geboortedatum. Middels programmacode kan dit echter lastig zijn vanwege de complexiteit van de Nederlandse taal, en het is dus verstandig om dit door een eindgebruiker handmatig te laten doen.

## 1g: Hoeveel Nederlanders hebben een account op de verschillende sociale netwerk sites?

De vraag over hoeveel Nederlanders een account op verschillende SNSs hebben, is lastiger te beantwoorden dan men wellicht in eerste instantie zou denken. De benodigde informatie wordt nu eenmaal niet altijd publiekelijk gemaakt, of kan ambigu zijn. Ambigüiteit treedt op wanneer men op een website bijvoorbeeld statistieken laat zien over gebruikers die Nederlands spreken, in plaats van gebruikers uit Nederland. Verder is ook de vraag of men statistieken geeft over het totaal aantal accounts, over actieve gebruikers die een account hebben en de site maandelijks bezoeken, en bezoekers aan de site die zelf geen accounts hebben. Verder is het ook maar de vraag of een account behoort aan een gebruiker, of dat organisaties wellicht zelf ook een account op een SNS hebben. Vooral bij Twitter gebeurt dit vaak.

De volgende data is beschikbaar over de verschillende SNSs:

Facebook	6.263580 miljoen gebruikers [27]
Hyves	9.7 miljoen gebruikers [28]
Twitter	5 miljoen gebruikers (ongeveer) [29]
LinkedIn	3 miljoen gebruikers (meer dan) [30]
Google Plus	Onbekend aantal gebruikers*

\* - Google Plus accounts zijn tegenwoordig verplicht voor mensen die een Gmail account aanmaken. Het is erg waarschijnlijk dat erg veel mensen al een Google Plus account hebben, omdat ze deze 'gratis' kregen bij een Gmail account. Desalniettemin is er geen concrete data beschikbaar over het aantal Google Plus gebruikers in Nederland, maar de algemene gedachte op het Internet is dat Google Plus maar weinig gebruikt wordt in vergelijking met o.a. Facebook.

[27] – <http://www.socialbakers.com/facebook-statistics/netherlands#chart-intervals>, geraadpleegd op 6 maart 2012

[28] – <http://www.hyves.nl/over/facts/>, geraadpleegd op 6 maart 2012

[29] – [http://semioCast.com/publications/2012\\_01\\_31\\_Brazil\\_becomes\\_2nd\\_country\\_on\\_Twitter\\_surpasses\\_Japan](http://semioCast.com/publications/2012_01_31_Brazil_becomes_2nd_country_on_Twitter_surpasses_Japan) (31 januari 2012)

[30] – <http://press.linkedin.com/node/1082> (27 januari 2012)

## 2a: Welke EI-standaarden zijn er voor declaraties en door wie worden ze gebruikt?

De volgende EI-standaarden zijn er in gebruik en worden door de meeste zorgverzekeraars ondersteund[31]:

Doelgroep	Code
Apothekers	<a href="#">AP304</a>
AWBZ	<a href="#">AW319</a>
Eerstelijns Psychologie	<a href="#">EP301</a>
GGZ	<a href="#">GZ311</a>
Huisartsen	<a href="#">HA304</a>
Kraamzorg	<a href="#">KZ301</a>
Leveranciers hulpmiddelen	<a href="#">LH307</a>
Mondzorg	<a href="#">MZ301</a>
Overige sectoren	<a href="#">OS301</a>
Paramedici	<a href="#">PM304</a>
Vervoer	<a href="#">VE303</a>
Verloskundige hulp	<a href="#">VK301</a>
Ziekenhuizen	<a href="#">ZH308</a>

Binnen de EI-standaarden voor declaraties is er ook een standaard voor de declaraties van de Forensische Zorg (standaard [FZ301](#)). Aangezien deze declaraties echter naar het ministerie van justitie gaan in plaats van de zorgverzekeraars, is deze standaard niet relevant voor het Nooddossier.

[31] – <http://www.vecozo.nl/web/download.aspx?bin=MogelijkhedenEDP2012.pdf>





De EI-standaarden definiëren voor elk recordtype een aantal gegevenselementen. Een gegevenselement is in principe een ‘veld’ in het record waar het bij hoort.

Om flat files te lezen, moet men weten volgens welke specificaties de bestanden geschreven zijn. Belangrijk is namelijk de positie waarop een bepaald gegevenselement staat. Aan de hand van de positie en lengte van een gegevenselement kan men het gegevenselement lezen uit een EI-bestand. Het datum-veld in het plaatje op de vorige pagina is een voorbeeld van zo’n gegevenselement.

De gegevenselementen zijn onderverdeeld in verschillende generieke types. Een generiek type beschrijft op welke manier men technisch gezien het gegevenselement in moet lezen. Men kan de types een beetje vergelijken met de primitieve types in veel programmeertalen.

Alle gegevenselementen hebben elk een unieke naam, welke volgens het formaat “<TYPE><IDNUMMER>-<ORGANISATIE>” beschreven is. In de naam van de gegevenselementen (bijvoorbeeld “COD392-VEKT”) kan men dus de types herkennen. De volgende types zijn er in de EI-standaarden aanwezig:

Type	Omschrijving
COD	Integer-waarde waarvan de betekenis afgeleid moet worden uit een codelijst.
TEC	Waarde met een technische achtergrond die geen betekenis heeft binnen het domein.
ANT	Integer-waarde welke een aantal of percentage aangeeft.
DAT	Datum of tijd-waarde.
BED	Geldbedrag in 1/100 <sup>ste</sup> eenheden (centen).
NAM	Naam of alfanumerieke waarde.
NUM	Waarde welke een nummer voorstelt. Kan ook alfanumeriek zijn (telefoonnummers en dergelijke) of een referentie bevatten naar organisatie-specifieke interne data (referentienummers).

Van het type TEC zijn er geen relevante gegevenselementen. Al deze elementen hebben een technische achtergrond, en bevatten geen informatie die relevant is voor medische specialisten.

Voor sommige elementen van het type ANT geldt dat er een conversie-slag gedaan moet worden om ze voor mensen goed leesbaar te kunnen maken. Percentages bijvoorbeeld worden opgeslagen als percentage in honderdste procenten (90% is dus gecodeerd als 09000). Over het algemeen geldt echter dat de inhoud van gegevenselementen van type ANT niet geconverteerd hoeven te worden om begrepen te kunnen worden, aangezien ze een gewoon aantal weergeven.

Van het type DAT worden er slechts twee soorten gebruikt: datums volgens het formaat EEJJMMDD (DAT039-NEN; “20120102” voor 2 januari 2012) en tijden volgens het formaat UUMM (DAT126-VEKT; “2015” voor kwart over acht). Deze moeten dus wel omgezet worden zodat mensen ze kunnen lezen, vooral omdat ze nogal relevant kunnen zijn.

Er zijn slechts twee soorten BED-velden: BED025-VEK1 ("Totaalbedrag 11 posities") en BED015-VEKT ("Standaardbedrag"). Het verschil tussen deze twee is dat BED015 uit 8 karakters bestaat, en BED025 uit 11 karakters bestaat. Beide soorten velden bevatten integers welke aangeven uit hoeveel eurocenten een bedrag bestaat. Ook hier geldt dus dat men enige conversie moet doen om deze bedragen voor mensen leesbaar te maken.

NAM-velden worden voornamelijk gebruikt voor alfanumerieke data die al door mensen gelezen kan worden. Veel van de NAM-velden zijn globaal en worden in alle standaarden gebruikt, aangezien ze betrekking hebben op de gegevens van een patiënt (tussenvoegsel, patiëntnaam). Een aantal velden zijn specifiek voor LH307 en VE303 en hebben betrekking op een straatnaam, woonplaatsnaam, en merknaam. Aangezien elementen van het type NAM alfanumeriek zijn en de te weergeven informatie al in het goede formaat opgeslagen worden, hoeven NAM-velden niet geconverteerd te worden.

Velden van het type NUM worden gebruikt voor referentienummers welke intern door verschillende instanties gebruikt worden, zoals intern patiëntnummer en dergelijke. Over het algemeen gezien lijken gegevens van het type NUM voornamelijk relevant te zijn voor interne doeleinden en efficiënte communicatie tussen zorgverzekeraar en zorgverlener, en kunnen ze niet geconverteerd worden naar relevantere informatie.

Van het type COD wordt veelal gebruik gemaakt van codelijsten. Velden in de COD categorie bevatten meestal integers welke verwijzen naar een zogenaamde codelijst. De codelijsten bevatten vervolgens informatie die gebruikt worden om de verschillende integers betekenissen te geven. Een veelgebruikte codelijst is bijvoorbeeld de codelijst voor geslacht, welke specificeert welke integer-waardes behoren bij welk geslacht. Praktisch alle velden van type COD moeten omgezet worden via een codelijst om voor mensen leesbaar te zijn.

## **2c: Welke gegevenselementen in de EI-standaarden moeten omgezet worden via databases of codelijsten zodat medisch specialisten ze kunnen begrijpen?**

Alleen van de gegevenselement-types COD zijn er gevallen waarbij men met een database of codelijst verbinding moet maken om de betekenis te kunnen achterhalen.

De COD-elementen welke beheerd worden door Vektis (CODxxx-VEKT) hebben over het algemeen gesproken een codelijst beschikbaar welke volgens een algemene standaard beschreven zijn. Tijdens het programmeren zou het dus mogelijk zijn om een algemene converter te schrijven welke deze gegevenselementen automatisch omzet via de codelijsten zonder enige problemen.

Er zijn echter ook een aantal velden waarvan men de bijbehorende codelijsten niet zomaar kan ophalen, omdat deze extern beheerd worden, omdat de te gebruiken codelijst afhangt van de inhoud van een ander gegevenselement. Bij deze velden zal men extra moeite moeten verrichten om de betekenis te achterhalen. Deze velden zijn over het algemeen erg interessant, aangezien ze vaak met branche-specifieke data werken. Zo bevat de LH307-standaard bijvoorbeeld gegevenselementen waarin men hulpmiddelen kan aangeven volgens de GPH-code ("Generieke Product Code"). Men kan een database met GPH-codes downloaden, zodat men GPH-codes om kan zetten naar producten als rolstoelen of wandelstokken.

Over het algemeen kan men stellen dat de gegevenselementen die men moet omzetten via een *externe* codelijst of database veel interessanter zijn dan de gegevenselementen waarvan men dat kan doen door middel van door Vektis aangeleverde codelijsten.

Hier onder staat schematisch weergegeven bij welke gegevenselementen men een externe database of codelijst moet gebruiken. Bovendien staat aangegeven in welke EI-standaarden ze gebruikt worden.

	AP304	AW319	EP301	GZ311	HA304	KZ301	LH307	MZ301	OS301	PM304	VE303	VK301	ZH308
COD040-CTS													
COD089-CSIZ													
COD099-DBCO													
COD329-DBCO													
COD988-DBC1													
COD989-DBCO													
COD990-DBCO													
COD014-KPN													
COD001-VEK4													
COD009-VEKT													
COD038-VEKT													
COD061-VEKT													
COD068-VEKT													
COD069-VEKT													
COD127-VEK1													
COD181-VEKT													
COD185-VEK1													
COD388-*													
COD400-VEKT													
COD413-VEKT													
COD434-VEKT													
COD695-VEK1													
COD788-VEKT													
COD789-VEKT													
COD810-VEKT													
COD811-VEKT													
COD860-VEKT													
COD973-VEKT													
COD978-ZIND													

### Legenda

	Gegevens-element dat in slechts één EI-standaard gebruikt wordt
	Gegevens-element dat in slechts een aantal EI-standaarden gebruikt wordt, maar niet alle EI-standaarden
	Gegevens-element dat in alle EI-standaarden gebruikt wordt.
	Irrelevant gegevens-element; niet interessant voor het Nooddossier.
	Gegevens-element is al leesbaar voor mensen. Extra informatie over de inhoud van dit element kan echter wel via externe datasources opgehaald worden.

Hier onder staat aangegeven welke COD-gegevenselementen er zijn, wie de bijbehorende codelijsten beheert, wat de gegevenselementen betekenen, en waar men de codelijst vandaan kan halen.

Naam	Beheerder	Betekenis
<a href="#">COD040-CTS</a>	Samenwerking NMT-ZN	Aanduiding gebitselementen conform het "Vademecum Tandheelkunde"
<a href="#">COD089-CSIZ</a>	Coördinatiepunt Standaardisatie Informatievoorziening in de Zorgsector	WCC-Code zorgaanbieder
<a href="#">COD099-DBCO</a>	DBC Onderhoud	Zorgactiviteitidentificatie (DOT)
<a href="#">COD329-DBCO</a>	DBC Onderhoud	Zorgactiviteitidentificatie
<a href="#">COD988-DBC1</a>	DBC Onderhoud	Zorgproduct gebaseerd op DOT
<a href="#">COD989-DBCO</a>	DBC Onderhoud	Hashtotaal DOT prestaties van grouper (ter controle)
<a href="#">COD990-DBCO</a>	DBC Onderhoud	Afsluitreden zorgtraject/subtraject
<a href="#">COD014-KPN</a>	TPG (voorheen KPN)	Postcode
<a href="#">COD001-VEK4</a>	Vektis	Kenmerk record
<a href="#">COD009-VEKT</a>	Vektis	AGB-Code (Zorgverlener)
<a href="#">COD038-VEKT</a>	Vektis	Intern patiëntnummer
<a href="#">COD061-VEKT</a>	Vektis	UZOVI-nummer
<a href="#">COD068-VEKT</a>	Vektis	Artikelcode hulpmiddel (Zi-nummer/G-standaard of NVAB-artikelcodering)
<a href="#">COD069-VEKT</a>	Vektis	Type hulpmiddel indien geen code bekend / het type niet via een artikelcode te achterhalen is
<a href="#">COD127-VEK1</a>	Vektis	Typerende diagnose DBC-zorgproducten
<a href="#">COD181-VEKT</a>	Vektis	AGB-Code (Praktijk)
<a href="#">COD185-VEK1</a>	Vektis	Verwijsdiagnose gecodeerd door paramedicus
<a href="#">COD388-*</a>	Vektis / Nederlandse Zorgautoriteit	Prestatiecode
<a href="#">COD400-VEKT</a>	Vektis	Paramedische diagnosecode
<a href="#">COD413-VEKT</a>	Vektis	Hulpmiddel welke niet voorkomt in de G-standaard
<a href="#">COD434-VEKT</a>	Vektis	Postcode buitenland
<a href="#">COD695-VEK1</a>	Vektis	Prestatiecode
<a href="#">COD788-VEKT</a>	Vektis	Zorgtypecode
<a href="#">COD789-VEKT</a>	Vektis	Typerende zorgvraag
<a href="#">COD810-VEKT</a>	Vektis	Postcode buitenland
<a href="#">COD811-VEKT</a>	Vektis	Postcode buitenland
<a href="#">COD860-VEKT</a>	Vektis	GPH-Code
<a href="#">COD973-VEKT</a>	Vektis	DBC declaratiecode
<a href="#">COD978-ZIND</a>	Z-Index	WMG-Tariefcode

Naam	Waar codelijst vandaan halen
<a href="#">COD040-CTS</a>	<a href="http://nl.wikipedia.org/wiki/Internationale_tandnummering">http://nl.wikipedia.org/wiki/Internationale_tandnummering</a>
<a href="#">COD089-CSIZ</a>	<a href="http://www.kiwaprismant.nl/prismant/expertise/WCC-lijst-instellingen.aspx">http://www.kiwaprismant.nl/prismant/expertise/WCC-lijst-instellingen.aspx</a>
<a href="#">COD099-DBCO</a>	<a href="http://www.dbconderhoud.nl/index.php?option=com_docman&amp;task=doc_download&amp;gid=3140&amp;Itemid=404">http://www.dbconderhoud.nl/index.php?option=com_docman&amp;task=doc_download&amp;gid=3140&amp;Itemid=404</a>
<a href="#">COD329-DBCO</a>	<a href="http://www.dbconderhoud.nl/index.php?option=com_docman&amp;task=doc_download&amp;gid=3140&amp;Itemid=404">http://www.dbconderhoud.nl/index.php?option=com_docman&amp;task=doc_download&amp;gid=3140&amp;Itemid=404</a>
<a href="#">COD988-DBC1</a>	<a href="http://www.dbconderhoud.nl/index.php?option=com_docman&amp;task=doc_download&amp;gid=3141&amp;Itemid=404">http://www.dbconderhoud.nl/index.php?option=com_docman&amp;task=doc_download&amp;gid=3141&amp;Itemid=404</a>
<a href="#">COD989-DBCO</a>	Geen (irrelevante data met technische achtergrond)
<a href="#">COD990-DBCO</a>	<a href="http://www.dbconderhoud.nl/index.php?option=com_docman&amp;task=doc_download&amp;gid=3078&amp;Itemid=404">http://www.dbconderhoud.nl/index.php?option=com_docman&amp;task=doc_download&amp;gid=3078&amp;Itemid=404</a>
<a href="#">COD014-KPN</a>	Geen (alfanumerieke data is al leesbaar). Om extra informatie weer te geven: <a href="http://postcode.nl">http://postcode.nl</a>
<a href="#">COD001-VEK4</a>	Geen (irrelevante data met technische achtergrond)
<a href="#">COD009-VEKT</a>	Database AGB-gegevens. Hiervoor moet men een licentie aanvragen bij Vektis.
<a href="#">COD038-VEKT</a>	Niet beschikbaar (is intern bij de zorgverlenende organisatie)
<a href="#">COD061-VEKT</a>	Relevante UZOVI-codes: <a href="http://www.vecozo.nl/web/download.aspx?bin=MogelijkhedenEDP2012.pdf">http://www.vecozo.nl/web/download.aspx?bin=MogelijkhedenEDP2012.pdf</a>
<a href="#">COD068-VEKT</a>	<a href="http://www.acousoft.nl/Downloads/NVABArtikelbestand.aspx">http://www.acousoft.nl/Downloads/NVABArtikelbestand.aspx</a>
<a href="#">COD069-VEKT</a>	Geen (fabrikant bepaald; invulling is al leesbaar voor mensen)
<a href="#">COD127-VEK1</a>	ICD-10 of DBC typeringslijst (volgens COD074-VEKT). ICD-10: <a href="http://www.icd-10.nl/downloadcentrum/">http://www.icd-10.nl/downloadcentrum/</a> DBC: <a href="http://www.dbconderhoud.nl/index.php?option=com_docman&amp;task=cat_view&amp;gid=477&amp;Itemid=32">http://www.dbconderhoud.nl/index.php?option=com_docman&amp;task=cat_view&amp;gid=477&amp;Itemid=32</a>
<a href="#">COD181-VEKT</a>	Database AGB-gegevens. Hiervoor moet men een licentie aanvragen bij Vektis.
<a href="#">COD185-VEK1</a>	<a href="http://www.fysionet.nl/kennisplein/organisatie-bedrijfsvoering-en-pz/ict-en-epd/financieel-administratieve-processen/ict_dcsph-per-01-07-2010.pdf">http://www.fysionet.nl/kennisplein/organisatie-bedrijfsvoering-en-pz/ict-en-epd/financieel-administratieve-processen/ict_dcsph-per-01-07-2010.pdf</a>
<a href="#">COD388-*</a>	Zie <a href="http://tog.vektis.nl/WebInfo.aspx?ID=Prestatiecodelijsten">http://tog.vektis.nl/WebInfo.aspx?ID=Prestatiecodelijsten</a> gebaseerd op <a href="http://ei.vektis.nl/WespGegevensElementenOverzichtDetail.aspx?ei_ge_code=COD367&amp;ei_or_code=VEKT">http://ei.vektis.nl/WespGegevensElementenOverzichtDetail.aspx?ei_ge_code=COD367&amp;ei_or_code=VEKT</a>
<a href="#">COD400-VEKT</a>	<a href="http://www.fysionet.nl/kennisplein/organisatie-bedrijfsvoering-en-pz/ict-en-epd/financieel-administratieve-processen/ict_dcsph-per-01-07-2010.pdf">http://www.fysionet.nl/kennisplein/organisatie-bedrijfsvoering-en-pz/ict-en-epd/financieel-administratieve-processen/ict_dcsph-per-01-07-2010.pdf</a>
<a href="#">COD413-VEKT</a>	ISO 9999 (bestellen bij NEN) of GPH ( <a href="http://ei.vektis.nl/Help/LH307v5.2/047-GPH%20versie%20011b-20111221.zip">http://ei.vektis.nl/Help/LH307v5.2/047-GPH%20versie%20011b-20111221.zip</a> )
<a href="#">COD434-VEKT</a>	Geen (alfanumerieke data is al leesbaar)
<a href="#">COD695-VEK1</a>	Voor de codelijst, zie voorgaand gegevenselement (COD367-VEKT). Mogelijke codelijsten: 041, 016, 990
<a href="#">COD788-VEKT</a>	<a href="http://www.dbconderhoud.nl/index.php?option=com_docman&amp;task=doc_download&amp;gid=3084&amp;Itemid=32">http://www.dbconderhoud.nl/index.php?option=com_docman&amp;task=doc_download&amp;gid=3084&amp;Itemid=32</a>
<a href="#">COD789-VEKT</a>	<a href="http://www.dbconderhoud.nl/index.php?option=com_docman&amp;task=doc_download&amp;gid=3383&amp;Itemid=404">http://www.dbconderhoud.nl/index.php?option=com_docman&amp;task=doc_download&amp;gid=3383&amp;Itemid=404</a>
<a href="#">COD810-VEKT</a>	Geen (alfanumerieke data is al leesbaar)
<a href="#">COD811-VEKT</a>	Geen (alfanumerieke data is al leesbaar)
<a href="#">COD860-VEKT</a>	<a href="http://ei.vektis.nl/Help/LH307v5.2/047-GPH%20versie%20011b-20111221.zip">http://ei.vektis.nl/Help/LH307v5.2/047-GPH%20versie%20011b-20111221.zip</a>
<a href="#">COD973-VEKT</a>	<a href="http://www.dbconderhoud.nl/index.php?option=com_docman&amp;task=cat_view&amp;gid=454&amp;Itemid=462">http://www.dbconderhoud.nl/index.php?option=com_docman&amp;task=cat_view&amp;gid=454&amp;Itemid=462</a>
<a href="#">COD978-ZIND</a>	<a href="http://www.apotheek.nl/Upload/Tarieven%20per%201%20januari%202010.pdf">http://www.apotheek.nl/Upload/Tarieven%20per%201%20januari%202010.pdf</a> (Zoek Z-index document??)

## Extra: Hoeveel mensen belanden er dagelijks op de spoedeisende hulp die niet aanspreekbaar zijn?

Om te bepalen of de technische beperkingen van verschillende sociale netwerken zoals beschreven in deelvraag 1c een groot risico met zich meebrengen, is het van belang om te weten hoeveel keer per dag het Nooddossier theoretisch gezien gebruikt kan gaan worden.

Om deze vraag te beantwoorden, moeten we eerst weten hoeveel mensen er dagelijks op de spoedeisende hulp-afdeling van een ziekenhuis belanden.

Volgens Consument en Veiligheid [32] worden er jaarlijks zo'n 2 miljoen behandelingen uitgevoerd op de afdeling spoedeisende hulp van de verschillende ziekenhuizen in Nederland.

Natuurlijk zijn er van deze 2 miljoen behandelingen een hoop behandelingen waarbij de patiënt bij kennis is en hij de behandelend verpleger cruciale informatie kan vertellen indien daar om gevraagd wordt. Van deze 2 miljoen behandelingen zullen er dus waarschijnlijk een hoop behandelingen zijn waarbij het Nooddossier niet gebruikt gaat worden. Om te bepalen over hoeveel gevallen dit gaat, moet men weten bij hoeveel behandelingen de patiënten buiten kennis zijn.

Helaas is hier weinig concrete informatie hier over te vinden. Wel kan men aannemen dat, tenzij er sprake is van een acuut levensbedreigend probleem wat binnen een aantal uren opgelost dient te worden, er voldoende tijd is voor een patiënt bij de spoedeisende hulp om de gewenste informatie te leveren. Wanneer men weet hoe vaak per jaar er een acuut probleem zich op treedt bij de spoedeisende hulp, heeft men dus een indicatie van hoeveel mensen er op de spoedeisende hulp belanden die potentieel niet aanspreekbaar zijn maar waarvan men wel snel alle relevante informatie wil hebben.

Uit [33] blijkt dat in 6.1% van de gevallen een behandeling bij de spoedeisende hulp als 'levensbedreigend' of 'acuut' aangemerkt wordt, waarbij men vreest dat vitale functies van het lichaam in gevaar lopen als men niet snel in actie komt. Dit komt neer op 122000 gevallen per jaar, of 334 gevallen per dag.

Uiteraard zullen er van deze 334 gevallen per dag een aantal gevallen zijn waarbij de patiënten wel aanspreekbaar zijn en waarbij men via de patiënt aan de benodigde informatie kan komen. Het werkelijke aantal keren waarbij het nooddossier erg goed van pas komt is dus lager dan 334 keer per dag.

Men kan dus concluderen dat de technische beperkingen met betrekking tot het aantal API calls dus waarschijnlijk geen invloed gaan hebben op de functionaliteit van het nooddossier.

[32] –

[http://www.veiligheid.nl/csi/veiligheid.nsf/wwwAssets/543CD58D60F286AEC12576DC0036D246/\\$file/Ziekte\\_vs\\_uitwendige\\_oorzaken.pdf](http://www.veiligheid.nl/csi/veiligheid.nsf/wwwAssets/543CD58D60F286AEC12576DC0036D246/$file/Ziekte_vs_uitwendige_oorzaken.pdf)

[33] – <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2564227/> (tabel 1; U1+U2)



## Antwoord op de hoofdvragen

- 1) Is het mogelijk om middels Social Media (Facebook, Hyves, Twitter, LinkedIn, Google Plus) van een persoon de geboortedatum op te halen, wanneer men van deze persoon alleen de naam en het geslacht weet?**

Ja, dit is mogelijk, alhoewel niet altijd even makkelijk of haalbaar. Om van VECOZO de zorgverzekeraars van een persoon op te halen, moet men de geboortedatum en achternaam weten, of de geboortedatum en postcode+huisnummer. Postcode+huisnummer kan men eigenlijk nergens vinden. De achternaam van een persoon zou men in principe moeten weten op de afdeling spoedeisende hulp door de persoonlijke bezittingen van een persoon te doorzoeken. De geboortedatum kan men in sommige gevallen via SNSs ophalen. Voornamelijk op Hyves kan men de geboortedatum gemakkelijk ophalen: 78% van de gebruikers stelt deze informatie beschikbaar aan derden, en 9.7 miljoen Nederlanders hebben een account op Hyves. Dit betekent dat men van 45% van de Nederlanders (bevolkingsaantal: 16.7 miljoen) de geboortedatum via Hyves kan opvragen, indien deze Nederlanders hun volledige voor- en achternaam op Hyves hebben staan. Dit is een significant aantal, alhoewel men er wel rekening mee moet houden dat niet alle gebruikers op Hyves vindbaar zullen zijn met alleen een achternaam, vooral wanneer deze achternaam zeer vaak voorkomt of wanneer gebruikers niet herkenbaar zijn aan hun profielfoto. Helaas zijn de resultaten op andere SNSs niet zo spectaculair: alle andere onder de loep genomen SNSs hebben de geboortedatum van gebruikers goed afgeschermd. Verder moet men uiteraard rekening houden met het feit dat niet iedereen zijn geboortedatum juist invult.

- 2) Is het mogelijk om uit de declaratiegegevens van de zorgverzekeraars informatie te halen die voor een medisch specialist interessante informatie bevat?**

Ja, dit is mogelijk. Er is aardig wat data beschikbaar in de verschillende declaratiegegevens. Naast persoonsgegevens en gegevens over behandelende artsen, welke erg interessant kunnen zijn voor artsen op de spoedeisende hulp, zijn er ook gegevens beschikbaar over diagnoses en medicinatie. Al de gegevens samen kunnen van een persoon een erg gedetailleerde declaratiegeschiedenis tonen, waaruit behandelende artsen relevante informatie kunnen halen over de medische geschiedenis van een persoon.

Helaas is het niet mogelijk om bijvoorbeeld de bloedgroep van een persoon te achterhalen via de declaratiegegevens.



Solid Innovator

*InfoSupport*

Bijlage III:  
Ontwerp Web-API voor Declaratiegegevens

**Nooddossier**

## Inleiding

Dit document beschrijft het ontwerp van de Web-API voor Declaratiegegevens, en de manier waarop dit ontwerp tot stand is gekomen. Deze API zal ingezet worden door zorgverzekeraars om declaratiegegevens van verzekerden door te geven aan het Nooddossier, welke vervolgens op zijn beurt de gegevens toonbaar maakt aan zijn eindgebruikers.

Voor meer informatie over het Nooddossier project, zie het bijbehorende Plan van Aanpak.

Het gekozen ontwerp zal, indien het Nooddossier-project een vervolg krijgt, geïmplementeerd dienen te worden door zorgverzekeraars. Zij zullen op basis van dit ontwerp software moeten schrijven en vervolgens deze software in productie moeten nemen. Het uiteindelijke gekozen ontwerp heeft hier rekening mee gehouden, om de implementatie zo gemakkelijk mogelijk te laten verlopen.

Voor de proof-of-concept versie van het Nooddossier wordt dit ontwerp één keer gerealiseerd, maar wordt er gebruik gemaakt van dummy-declaratiedata.

## Requirements

De Web-API zal de volgende functionaliteit moeten bieden:

- Declaraties van één persoon beschikbaar stellen aan het Nooddossier
- Filtreren op: BSN (verplicht), PrestatieDatum (begin- en eind-datum), Declaratiestandaard en versienummer
- Support voor paginering middels 'skip' (offset) en 'take' (limit) om de declaratiegegevens op een goede wijze weer te kunnen geven binnen het Nooddossier zonder dat men (te) veel data in één keer hoeft op te halen.
- Sorteren van declaraties op prestatie-begindatum (ascending en descending)

De volgende zaken zijn geen echte requirements, maar kunnen wel invloed uitoefenen op bepaalde beslissingen. In feite zijn dit dus de 'would-like-to-haves':

- Eenvoudig te implementeren API welke gemakkelijk onderhoudbaar is of maar weinig onderhoud nodig heeft, zodat zorgverzekeraars gemakkelijk en goedkoop de API kunnen implementeren binnen hun bestaande systemen indien het Nooddossier-project een vervolg krijgt.
- Zo veel mogelijk de bestaande standaarden voor declaratieverkeer vasthouden, en zo min mogelijk bestaande data van formaat wijzigen, zodat alles gemakkelijk implementeerbaar is en zodat men bestaande architecturen/software kan hergebruiken.

## Problemen

Helaas zijn er tijdens het ontwerpen van de API een aantal problemen gevonden welke men in het uiteindelijk gekozen ontwerp heeft getracht op te lossen, maar die wellicht gezorgd hebben voor wat vreemde ontwerpkeuzes.

Het grootste probleem dat ondervonden was, is dat één declaratiebestand kan bestaan uit meerdere zogenaamde 'prestaties'. Een prestatie is een bepaalde handeling die verricht is op een patiënt; de precieze inhoudelijke betekenis van een prestatie hangt af van het type EI-bericht. Over het algemeen spreekt men over een behandeling wanneer men het heeft over een prestatie.

Omdat een declaratiebestand uit meerdere prestaties kan bestaan, is het niet altijd mogelijk om de begin- en eind-datum van een declaratie goed te bepalen. Wanneer een bestand bijvoorbeeld bestaat uit twee prestaties, waarvan één uit het jaar 2009 en een ander uit het jaar 2012, is het lastig om van dit bestand te bepalen over welke datum het precies gaat, en waar men deze declaratie in een lijst van gesorteerde declaraties op datum moet plaatsen. Voor het uiteindelijke Nooddossier product is het van belang dat de verrichte *prestaties* gesorteerd zijn op datum, niet de verrichte *declaraties*. Op deze manier kan men inzicht krijgen in het tijdstip waarop een bepaalde behandeling is uitgevoerd, of het tijdstip waarop iemand een bepaald medicijn heeft gehaald bij de apotheker. Wanneer men declaraties sorteert op datum kan men hier niet altijd van uit gaan, aangezien er meerdere prestaties in hetzelfde declaratiebestand kunnen zitten.

In de gekozen oplossing heeft men dit probleem opgelost door alle prestaties van metadata te voorzien, welke de prestatiedatums (begin en eind) afzonderlijk beschrijven. Het volledige declaratiebestand wordt verzonden als bijlage van de prestatie. Dit heeft als nadeel dat hetzelfde bestand meerdere keren opgestuurd wordt naar de Nooddossier-servers in het geval dat een bestand meerdere prestaties heeft. Deze keuze heeft wel als voordeel dat men zonder al te veel file-parsing (aan de zijde van de Zorgverzekeraar/Declaratie webservice) toch declaraties kan aanbieden aan het Nooddossier, en dat deze declaraties aangeboden worden met metadata welke gaat over de prestaties in deze declaraties.

## **Uiteindelijk API-ontwerp**

Het uiteindelijke ontwerp is dat de rauwe EI-bestanden middels Base64-encoding in een XML-document gezet worden, waar omheen een prestatie-XML-element gebouwd wordt welke enkele zeer belangrijke prestatiegegevens gemakkelijk beschikbaar stelt aan het Nooddossier. Indien er meerdere prestaties gedefinieerd zijn binnen een EI-bestand, wordt nog steeds het hele EI-bestand meegestuurd. Op deze manier hoeft de Declaraties-webservice bijna geen parsing te doen, wat alles een stuk onderhoudbaarder maakt. Mocht een zorgverzekeraar toch willen dat andere declaraties en prestaties niet meegestuurd worden indien ze in hetzelfde bestand gedefinieerd zijn, dan mogen ze het declaratiebestand zelf aanpassen, maar de specificaties vermelden en verwachten niet dat dit gebeurt.

Het transporteren van de XML-documenten gebeurt via HTTP. Hieronder staat beschreven hoe alles precies in zijn werk gaat.

## GET /getPrestaties

### Invoerparameters:

Parameter	Type	Default indien niet opgegeven	Dataformaat (+ voorbeeld)	Verplicht (+ errorcode indien niet opgegeven)	Omschrijving
bsn	Datafilter (==)		BSN-nummer (145133813)	Ja (1000)	Burger Service Nummer van de patiënt waarop men filtert.
standaardcode	Datafilter (==)		Integer (101)	Nee	Bericht-standaard waarop men filtert. Voor ZH308-declaraties: 101
versienummer	Datafilter (==)		String (8.0)	Nee	Versienummer van de berichtcode (standaard) waarop men filtert.
begindatum	Datafilter (>=)		String; yyyyMMdd (20120325)	Nee	Alle prestaties die <i>niet</i> na of op deze datum plaats gevonden hebben, worden weg-gefilterd
einddatum	Datafilter (<=)		String; yyyyMMdd (20120325)	Nee	Alle prestaties die <i>niet</i> voor of op deze datum plaats gevonden hebben, worden weg-gefilterd
Id	Datafilter (==)		Integer (42)	Nee	Uniek bericht-ID waarop gefilterd dient te worden. Indien opgegeven zal er slechts één resultaat geretourneerd worden.
skip	Weergave-optie + datafilter	0	Integer (12)	Nee	Aantal records dat overgeslagen dient te worden in het eindresultaat
take	Weergave-optie + datafilter		Integer (13)	Nee	Aantal records dat teruggegeven dient te worden in het eindresultaat
sortdate	Weergave-optie + datafilter	DESC	String; "ASC" (ascending) of "DESC" (descending)	Nee	Specificeert de manier waarop de records op <i>begindatum</i> gesorteerd dienen te worden. Staat standaard op 'descending' (records met meest recente begindatum eerst)
blob	Weergave-optie	1	Boolean; 1 (true) of 0 (false)	Nee	Indien '1': Laat base64-encoded binary van het bron-EI-bestand zien. Indien '0': laat deze base64-encoded binary niet zien.

## Retourformaat (XML):

```
<?xml version="1.0" encoding="utf-8"?>
<prestaties>
  <prestatie bsn="$BSN" begindatum="$BEGINDATUM" einddatum="$EINDDATUM"
prestatievolgnummer="$PRESTATIEVOLGNUMMER" standaardcode="$STANDAARDCODE"
versienummer="$VERSIENUMMER" id="$ID">
    <blob>
      $BLOB
    </blob>
  </prestatie>
</prestaties>
```

## Data-formaten:

Gegeven	Dataformaat	Voorbeeld
\$BSN	String (BSN)	145133813
\$BEGINDATUM	String (yyyyMMdd)	20120325
\$EINDDATUM	String (yyyyMMdd)	20120325
\$PRESTATIEVOLGNUMMER	Integer	1
\$STANDAARDCODE	Integer	101
\$VERSIENUMMER	String (Major.Minor)	8.0
\$ID	Integer	12
\$BLOB	Base64-geëncodeerde binary van een bestand van de EI-standaard. Base64 volgens RFC 1521	MDExMDEwODAwVDk5OTgwMTEgICAgICAgICAw... (voorbeeld te lang om hier weer te geven)

Aantal voorkomens elementen:

Element	Aantal voorkomens in retour-XML
prestaties	1 (root)
prestatie	0..* (1 element per prestatie)
blob	0 of 1 (hangt af van 'blob' invoerparameter)



## Schema voor validatie:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      Schema om Prestaties te valideren welke door een Zorgverzekeraar
      geretourneerd worden
    </xsd:documentation>
  </xsd:annotation>

  <!-- Root element -->
  <xsd:element name="prestaties" type="PrestatiesRootType"/>

  <!-- Types -->

  <xsd:complexType name="PrestatiesRootType">
    <xsd:sequence>
      <xsd:element name="prestatie" type="PrestatieType" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="PrestatieType">
    <xsd:sequence>
      <xsd:element name="blob" type="xsd:base64Binary" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="bsn" type="xsd:int"/>
    <xsd:attribute name="begindatum" type="SimpleDate-yyyyMMdd"/>
    <xsd:attribute name="einddatum" type="SimpleDate-yyyyMMdd"/>
    <xsd:attribute name="prestatievolgnummer" type="xsd:int"/>
    <xsd:attribute name="standaardcode" type="xsd:int"/>
    <xsd:attribute name="versienummer" type="xsd:string"/>
    <xsd:attribute name="id" type="xsd:int"/>
  </xsd:complexType>

  <xsd:simpleType name="SimpleDate-yyyyMMdd">
    <xsd:annotation>
      <xsd:documentation xml:lang="en">
        Deze simpleType controleert niet of dat de gegeven datum ook echt bestaat voor de
        gegeven maand.
        20110231 (31 februari 2011) is dus valid volgens deze type, terwijl die datum niet
        bestaat.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[0-9]{4}((0[1-9])|(1[0-2]))((0[1-9])|([1-2][0-9])|(3[0-1]))"/>
    </xsd:restriction>
  </xsd:simpleType>

</xsd:schema>
```

## Foutafhandeling:

Indien er een fout optreedt, wordt HTTP Status code 200 ("OK") teruggegeven, en de message body zal het volgende zijn:

```
<?xml version="1.0" encoding="utf-8"?>
<error errorCode="$ERRORCODE">$ERRORBESCHRIJVING</error>
```

De volgende errorcodes en beschrijvingen zijn op dit moment gedefinieerd. Meerdere errorcodes en beschrijvingen zullen later gedefinieerd worden:

Errorcode	Error beschrijving
0	Onbekende fout opgetreden
100	Onjuist formaat één of meerdere invoerparameters
1000	Geen BurgerServiceNummer (BSN) opgegeven. Dit is een verplicht parameter. Voer a.u.b. een BSN in (parameter: 'bsn')

HTTP Status code 500 is gereserveerd voor algemene fouten buiten de API om. Het gebruik van errorcode 0 wordt afgeraden en zal alleen tijdens development/debugging gebruikt worden.

De errorcodes zijn over het algemeen zo opgesteld dat een errorcode met een hoger aantal significante cijfers een specifiekere error aanduidt. Errorcodes met slechts één significant cijfer (bijvoorbeeld '0') zijn dus helemaal niet specifiek, terwijl errorcodes met een hoog aantal significante cijfers (bijvoorbeeld '1000') erg specifiek zijn.

## Schema voor validatie error:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      Schema om Errors te valideren
    </xsd:documentation>
  </xsd:annotation>

  <!-- Root element -->
  <xsd:element name="error" type="ErrorRootType"/>

  <!-- Types -->
  <xsd:complexType name="ErrorRootType">
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="errorCode" type="xsd:int" />
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:schema>
```