

# Afstudeerverslag

Evert van de Braak (20052468)

*"Het ontwikkelen van een  
geautomatiseerd statistieken  
weergeefstelsel bij Innospense"*



**Afstudeer periode:** 14/11/2011 – 19/03/2012

**Begeleidend docent:** Dhr. E.M. van Doorn

**Expert Docent:** Dhr. W. van Vliet

**Bedrijfs Mentorr:** Dhr. B. Timmermans

## Referaat

Dit document bevat een eindverslag van de ontwikkeling van een onderdeel van een webportaal (een systeemdeel) dat grafisch statistieken weergeeft op basis van een op te zetten gegevensanalyse. Het project is een opdracht van Innospense. Dit project fungeert tevens als afstudeerproject van Evert van de Braak, HBO Informatica student aan de Haagse Hogeschool. De afstudeerstage vond plaats van 14 november 2011 t/m 19 maart 2012, dat houdt in dat het afstudeertraject in totaal (40 uur \* 17 weken) 680 uur bedraagt.

Dit document is geschreven voor de examinatoren van de Haagse Hogeschool en de gecommitteerde.

Gebruikte afkortingen:

- |             |                                    |
|-------------|------------------------------------|
| • HHS       | De Haagse Hogeschool               |
| • UP        | Unified Proces                     |
| • UML       | Unified Modeling Language          |
| • MS Office | Microsoft Office                   |
| • SVM       | Support Vector Machine             |
| • IDE       | Integrated Development Environment |
| • PHP       | PHP Scripttaal                     |
| • WAMP      | Windows Apache MySQL PHP           |
| • AJAX      | Asynchronous Java And Xml          |
| • JSON      | JavaScript Object Notation         |
| • JSXGraph  | Een Javascript library             |

## Voorwoord

Dit verslag is geschreven in het kader van mijn afstudeerstage bij Innospense B.V. te Den Haag. De afstudeerstage is de laatste fase van mijn opleiding Informatica aan de Haagse Hogeschool, Academie ICT & Media te Den Haag. De stage heeft in de periode 14 november 2011 tot en met 19 Maart 2012 plaats gevonden.

Het verslag bevat achtereenvolgens een beschrijving van de organisatie waar de afstudeerstage heeft plaats gevonden. Een definitie van de afstudeeropdracht met een beschrijving van het plan van aanpak om deze opdracht uit te voeren. Vervolgens worden de uitgevoerde werkzaamheden beschreven. Ten slotte zijn het proces en de opgeleverde producten geëvalueerd.

De afstudeeropdracht is door mij als een zeer leerzame en intensieve periode ervaren. Het uitvoeren van de opdracht heeft van begin tot eind vele uitdagingen met zich meegebracht waarvoor telkens een passende oplossing gevonden moest worden. Mede door deze uitdagende elementen, en de manier waarop ik hierin ben ondersteund, heb ik deze opdracht met veel plezier uitgevoerd.

Graag maak ik gebruik van de gelegenheid om dit voorwoord af te sluiten met een dankbetuiging aan een aantal personen die mij tijdens deze afstudeerperiode hebben ondersteund en begeleid. De heer Bartel Timmermans en de heer Thijs van Nuenen zou ik willen bedanken voor de ondersteuning en de begeleiding die ze tijdens het gehele afstudeer traject hebben geboden. Naast mijn bedrijfsbegeleiders zou ik graag de medewerkers van Innospense willen bedanken voor hun aangename gezelschap en hun inbreng tijdens mijn afstudeertraject. Ik heb erg genoten van de tijd die ik bij Innospense heb doorgebracht.

Tot slot wil ik mijn twee examinatoren vanuit de Haagse Hogeschool de heer Ed van Doorn en de heer Willem van Vliet bedanken voor hun feedback en expertise tijdens het afstudeerproces.

Evert van de Braak,

Den Haag, 18 maart 2012

## Inhoudsopgave

Referaat.....	2
Voorwoord.....	3
1 Inleiding.....	1
2 De organisatie .....	2
2.1 Innospense.....	2
2.2 De plaats van de afstudeerder binnen de organisatie.....	3
3 De opdracht .....	4
3.1 Achtergrond .....	4
3.2 Probleemstelling .....	4
3.3 Doelstelling .....	5
3.4 Op te leveren producten.....	6
4 Inception fase.....	7
4.1 Opstellen plan van aanpak.....	8
4.1.1 Bepalen aanpak.....	9
4.1.2 Bepalen tijdsplanning.....	10
4.1.3 Bepalen van technieken.....	12
4.2 Uitvoeren van analyses en onderzoek.....	13
4.2.1 Oriënteren op statistische informatie.....	14
4.2.2 Grafisch statistieken concept ontwerp.....	16
4.2.3 (On-)geautomatiseerde patroonherkenning .....	19
4.3 Resultaat inception fase.....	21
5 Eerste iteratie.....	23
5.1 Vaststellen eisen .....	24
5.2 Use cases in kaart brengen .....	25
5.3 Bepalen van het architecturaal design .....	26
5.3.1 Client – Server architectuur .....	26
5.3.2 Server-side scripting vs. Client-side scripting.....	29
5.4 Resultaat eerste iteratie .....	32
6 Tweede iteratie .....	34

6.1	Wijzigingen ten opzichte van de eerste iteratie .....	35
6.2	Bepalen van het klassendiagram .....	37
6.2.1	Welke data is nodig voor de grafiek? .....	38
6.2.2	De communicatie structuur .....	41
6.2.3	Ontwerpen klassendiagram .....	45
6.3	Ontwikkelen .....	49
6.3.1	Ontwikkelomgeving en technieken .....	50
6.3.2	Ontwikkel knelpunten en oplossingen.....	53
6.3.3	Opstellen testplan .....	55
6.4	Resultaat tweede iteratie .....	56
7	Evaluatie.....	57
7.1	Procesevaluatie .....	58
7.1.1	Planning evaluatie .....	58
7.1.2	Interviews en onderzoek.....	60
7.1.3	Ontwerp fase.....	61
7.1.4	Ontwikkel fase.....	61
7.2	Productevaluatie .....	62
7.2.1	Plan van aanpak .....	62
7.2.2	Onderzoeksrapport.....	62
7.2.3	Elaboration rapport.....	63
7.2.4	Statistische grafiek .....	64
8	Te demonstreren beroepstaken .....	65
8.1	Uitvoeren analyse door definitie van requirements 1.4.....	66
8.2	Ontwerpen systeemdeel 3.2.....	66
8.3	Bouwen applicatie 3.3.....	66
9	Geraadpleegde literatuur .....	67
10	BIJLAGEN .....	68

# 1 Inleiding

Dit rapport is geschreven in de context van een afstudeerproject in opdracht van Innospense BV. Het bedrijf ontwikkelt producten die zorgverlening op afstand bevorderen. Het centrale product is de Medido Connected, een automatische medicatie dispenser, ofwel uitgifte machine. Door de dispenser vergeten cliënten minder vaak hun medicatie in te nemen en hierdoor vergroot de therapietrouw.

Apothekers kunnen op afstand door middel van het Innospense webportaal het uitgifte schema van hun eigen patiënten inzien en wijzigen. Naast de apotheek is er vanuit Innospense nauwe samenwerking met een thuiszorgcentrum en verschillende externe partijen om de kwaliteit van de dienstverlening te waarborgen. Dit netwerk bestaat uit de arts, cliënten, installatiebedrijf, zorgcentrum, thuiszorg, verpakker en Innospense zelf.

Het afstudeerproject heeft een duur van zeventien weken met als doel het ontwikkelen van een statistieken weergeefstelsel voor het Innospense webportaal. Mijn afstuderen is onder te verdelen in drie hoofdfasen: begin-, ontwerp- en bouwfase. De ontwerp(elaboration)fase neemt het meeste tijd in beslag. Het project is gebaseerd op de software ontwikkel methode UP , ofwel iteratief software ontwikkelen.

Alle vormen van externe communicatie worden eerst intern besproken met de opdrachtgevers.

## 2 De organisatie

### 2.1 Innospense

Innospense is een bedrijf dat een medicijndispenser, genaamd Medido, heeft ontwikkeld en dat als beheerder van een webportaal optreedt. Het webportaal fungeert als intermediair tussen de geplaatste Medido's en de betrokken hulpverleners.

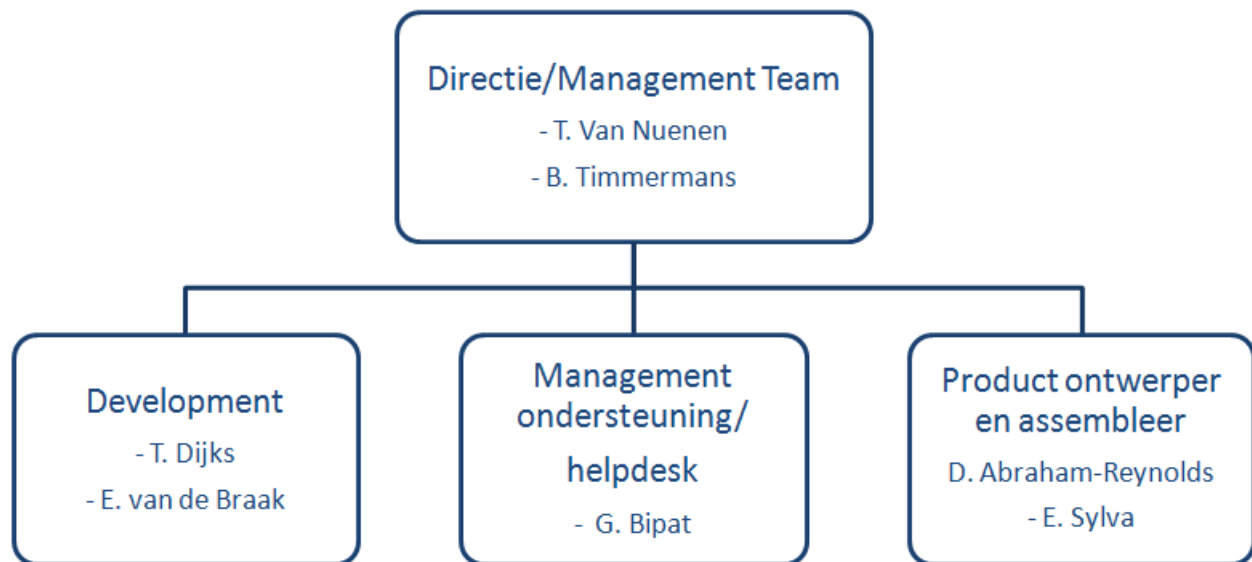
De Medido kan als medicijndispenser bij thuiszorgpatiënten worden geplaatst. De Medido geeft een geluidsignaal op een vooraf ingestelde tijd om de patiënt erop te attenderen dat het tijd is om de medicijnen in te nemen. Nadat de patiënt de knop op de Medido activeert, werpt de Medido een zakje met de in te nemen medicijnen uit met een inkeping om het zakje gemakkelijk open te kunnen maken.

De Medido is via een GPRS module aangesloten op het internet. De Medido kan hierdoor informatie versturen naar Innospense. Het kan bijvoorbeeld informatie sturen of de medicijnen op tijd zijn ingenomen, maar ook informatie ontvangen van het webportaal om het schema in te stellen wanneer medicijnen moeten worden ingenomen. De informatie in het webportaal is uit te lezen door medewerkers van thuiszorg, huisartsen en andere belanghebbenden. Er zijn plannen om (draadloos)randapparatuur aan te kunnen sluiten op de Medido, waardoor onder andere bloeddruk, glucose gehalte en gewicht opgemeten en verstuurd kan worden naar het webportaal.

Het bedrijf bestaat uit 5 personen die de volgende functies bekleden: directeur, projectmanager, systeembeheerder/ontwikkelaar, helpdeskmedewerker en assemblagemedewerker.

## 2.2 De plaats van de afstudeerder binnen de organisatie

Mijn twee begeleiders bij Innospense zijn Thijs van Nuenen en Bartel Timmermans die ook de oprichters zijn van Innospense. Wij zitten in dezelfde ruimte wat voor een amicale werksfeer zorgt en waardoor ik met vragen altijd terecht kan bij mijn begeleiders. Indien deze niet aanwezig zijn kan ik ze altijd bereiken via Skype, email of de telefoon. Met technische vragen wat betreft programmeren en of informatica gerelateerde vragen kan ik terecht bij de externe programmeur van Innospense Thomas Dijks. In Figuur 1 is een model te zien van de bedrijfshiërarchie.



Figuur 1 - Hierarchy Innospense



## 3 De opdracht

### 3.1 Achtergrond

Innospense is in bezit van een groot webportaal waarmee de Medido kan worden aangesproken. De Medido is via een GPRS module aangesloten op het internet. De Medido kan hierdoor informatie versturen naar het webportaal van Innospense. De Medido kan bijvoorbeeld informatie sturen of de medicijnen op tijd zijn ingenomen, maar ook informatie ontvangen van het webportaal om het uitgifteschema in te stellen wanneer medicijnen moeten worden uitgenomen. De uitgifte schema's worden bepaald door de apotheker. De ingestelde weekschema's zijn in het webportaal uit te lezen door medewerkers van Innospense, thuiszorg, huisartsen en andere belanghebbenden.

### 3.2 Probleemstelling

Het webportaal is toegankelijk voor de thuiszorg, huisartsen, apothekers en andere specialisten via een inlogstelsel. Door gebruik van het webportaal en feedback van gebruikers zijn er langzaam meer en meer functies aan het webportaal toegevoegd en is de hoeveelheid informatie samen met het aantal gebruikers gegroeid.

Op dit moment zijn meer dan honderd Medido-dispensers die gebruikt worden door patiënten in de intramurale en extramurale<sup>1</sup> zorg. Elke keer dat een Medido dispenser wordt gebruikt, wordt informatie vastgelegd in de database van het webportaal. De Medido registreert bijvoorbeeld de tijd wanneer de gebruiker een medicijn uit de Medido dispenser haalt, maar ook of het medicijn te laat, te vroeg of op schema is uitgenomen. In sommige gevallen kunnen medicijnen eerder uit de Medido-dispenser worden aangevraagd en dus eerder worden uitgenomen. Ongeacht de enorm harde piep die de Medido kan geven (50-100db), kan men ook vergeten om de medicijnen op tijd uit te nemen. Al deze informatie wordt per Medido-dispenser vastgelegd in de database van het webportaal.

Met al deze informatie die wordt opgeslagen in de database van het webportaal wordt echter nog niets gedaan. Statistieken zijn niet beschikbaar in een helder overzicht wat het ook lastiger maakt om een uitgiftepatroon te herkennen. Om een patroon te herkennen zal een zorginstelling eerst statistieken van de Medido-dispenser aanvragen bij het webportaal van Innospense. Een medewerker van Innospense zal de statistische informatie zonder gebruikersvriendelijk user-interface verzamelen door screendumps te maken van database informatie. Met de gegeven informatie moet een apotheker en/of zorginstelling een dergelijk patroon zelf ontdekken en wordt dit niet automatisch gedaan. Alleen al het verzamelen van statistische informatie is een tijdrovend, gebruikersonvriendelijk en onhandig proces, laat staan het herkennen van een patroon. Daarnaast heeft Innospense een groot probleem als bij schaalvergroting elke Medido-gebruiker op hetzelfde moment zou vragen naar een statistisch overzicht. Het

---

<sup>1</sup> Intramurale zorg is gezondheidszorg die gedurende een onafgebroken verblijf van meer dan 24 uur geboden wordt in een zorginstelling. Extramurale zorg is een vorm van zorg voor mensen met een verzorgingshuisindicatie die niet zijn opgenomen in een instelling.

automatiseren voor het aanvragen van statistische gegevens en het herkennen van gebruikspatronen is voor Innospense in dit proces dus een belangrijke voorwaarde voor groei.

Samengevat komt dit neer op de volgende probleemstelling:

*‘Er is momenteel geen geautomatiseerd statistisch overzicht van Medido-gegevens, waarbij automatisch patronen herkend worden, voor zowel de klanten als medewerkers van Innospense’*

De opdracht bestaat hiermee uit twee delen: Het eerste deel is het webportaal uitbreiden met de functionaliteit om een grafisch overzicht van verschillende Medido-statistieken weer te geven. Naast de eisen van de opdrachtgever zal daarbij de mening van Medido-gebruikers een belangrijke rol spelen. Het tweede deel bestaat uit een te ontwerpen systeemonderdeel dat gebruikspatronen herkent van de verzamelde data, hier grafische weergegeven statistieken van maakt voor presentatie via het webportaal en zonodig een melding verstuurt naar apothekers of zorginstellingen. De software zou in PHP worden ontwikkeld, getest en uiteindelijk geïmplementeerd worden in het webportaal.

### 3.3 Doelstelling

*‘De doelstelling van dit project is het ontwikkelen van een geautomatiseerd statistieken weergeefstelsel bij Innospense dat patronen herkent’*

Om dit uit te kunnen voeren zal het proces voor het verstrekken en toedienen van medicatie in kaart worden gebracht. Er zal een uitgebreide analyse uitgevoerd worden voor het antwoord op de vraag naar welke statistieken vraag is en hoe deze grafisch kunnen worden weergegeven voor de verschillende gebruikersgroepen. Apothekers en zorgverleners hebben immers behoefte aan andere of anders weergegeven gegevens dan bijvoorbeeld werknemers van Innospense.

Naast het ontwikkelen van een systeemdeel dat statistische gegevens grafisch weergeeft, zal het statistieken onderdeel ook patronen herkennen waardoor de Medido-dispenser optimaal door apothekers en zorgverleners kan worden ingezet. Voor het toepassen van patroonherkenning zal er uitgebreid onderzoek worden gepleegd naar de theorie, techniek en toepassing hiervan.

De ideale situatie die hieruit voort zou kunnen komen ziet er als volgt uit:

Innospense-medewerkers, zorgverleners en apotheken kunnen verschillende gebruiksstatistieken bekijken van de Medido-dispenser via het webportaal en krijgen daarbij een gemakkelijk te gebruiken en duidelijk grafisch overzicht. Uitgifte patronen van Medido gebruikers worden automatisch herkend waarop apothekers en zorgverleners kunnen reageren. Medewerkers van Innospense hoeven vrijwel geen tijd meer te besteden aan het verzamelen van statistische informatie voor klanten, omdat het webportaal de mogelijkheid biedt om dit zelfstandig te doen. Indien nodig kunnen medewerkers het webportaal ook gebruiken om specifieke informatie te verzamelen.

## 3.4 Op te leveren producten

In de eerste week van mijn afstudeer traject is een lijst gemaakt van op te leveren producten.

- Plan van aanpak
- Onderzoeksrapport
- Elaboration rapport
- Construction rapport
- Test plan
- Deployment plan

### ***Het plan van aanpak***

Dit document handelt over de activiteiten en middelen die voor het project nodig zijn en hoe deze met elkaar samenhangen.

### **Onderzoeksrapport**

Bevindingen van analyses en onderzoek worden vastgelegd in het onderzoeksrapport. Binnen deze afstudeeropdracht is de benaming Onderzoeksrapport vervangen voor het gebruikelijke Inception rapport in verband met de doelgroep waarvoor dit document bestemd is.

### ***Elaboration rapport***

In dit document worden de requirements vastgelegd en geeft het een volledige beschrijving van de functionele en niet-functionele eisen en het gedrag van het systeemdeel. Naast de eisen bevat dit document expliciete informatie over de architectuur, de structuur en de samenhang van systeemdelen.

### ***Construction rapport***

Dit document bevat informatie over de bouw/ontwikkeling en uitwerkingen van de in het elaboration rapport beschreven componenten.

### ***Testplan***

In dit document worden softwaretesten opgenomen om het systeemdeel uitvoerig te testen op bugs en errors. Het testplan wordt ontwikkeld tijdens de Construction fase.

### ***Deployment plan***

In dit document wordt een advies uitgegeven met betrekking tot de risico's, complicaties, aandachtspunten en vervolgstappen voor tijdens de transition fase

## 4 Inception fase

De inception fase is de eerste fase van mijn afstudeertraject. In deze fase waren er twee centrale hoofdvragen: *‘Hoe ga ik het project aanpakken wat betreft werkzaamheden en technieken?’* en *‘Welke informatie en kennis is nodig om de eisen voor dit project vast te stellen?’*. Door middel van een plan van aanpak, het plegen van analyses en onderzoek zijn deze hoofdvragen beantwoord. Als producten zou aan het eind van deze fase het plan van aanpak en het onderzoeksrapport worden opgeleverd.

In dit hoofdstuk wordt beschreven hoe deze fase van het project is aangepakt en is opgedeeld in drie paragrafen. Hierbij wordt de nadruk gelegd op de motivatie van keuzes tijdens specifieke beslismomenten.

In de eerste paragraaf, opstellen plan van aanpak, wordt beschreven hoe aanpak, tijdsplanning en de gebruikte technieken zijn bepaald.

In de tweede paragraaf, uitvoeren analyses en onderzoek, worden de keuzemomenten beschreven en de beslissingen die zijn genomen tijdens het uitvoeren van analyses en onderzoek.

In de derde paragraaf, resultaat inception fase, wordt beschreven tot welke conclusies en resultaten het onderzoek en de uitgevoerde analyses in de inception fase hebben geleid.

## 4.1 Opstellen plan van aanpak

Tijdens het begin van het afstudeertraject is er een plan van aanpak geschreven. Het plan van aanpak is het document om het project in globale lijnen te identificeren en waarin de uit te voeren stappen staan beschreven.

Om het plan van aanpak te kunnen schrijven is er globale informatie rondom de opdracht nodig. Door de opdrachtdefinitie grondig door te nemen en de hoofdvraag in deelvragen op te delen, ontstonden vragen die dieper ingingen op het onderwerp rondom de opdracht:

- Hoe ziet de opdrachtgever het resultaat?
- Het Innospense webportaal wat is het en hoe werkt het?
- Wat /Wie zijn betrokken partijen: Medido/Patiënt/Apothekers/etc.?
- Medicatie uitgften hoe vindt wat en waar plaats?

Met het gebruik van deze deelvragen in interviews met mijn begeleider heb ik de wensen, eisen en ook ideeën van de begeleider vertaald naar het plan van aanpak. Hoe dit proces is verlopen wordt beschreven in de volgende drie subparagrafen. In deze subparagrafen wordt het proces beschreven wat het bepalen van de aanpak, tijdbepaling en gebruikte technieken beschrijft.

Het plan van aanpak is toegevoegd als bijlage, zie bijlage<sup>i</sup>.

### 4.1.1 Bepalen aanpak

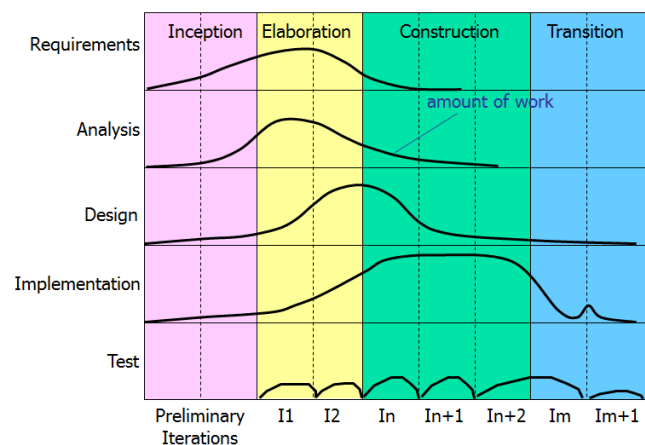
Tijdens het uitvoeren van een project binnen de IT zijn verschillende methoden die toegepast kunnen worden, zoals iteratief ontwikkelen of de watervalmethode. Voor dit project is voor het iteratief softwareontwikkelingsproces UP<sup>2</sup> en de modelleer taal UML<sup>3</sup> gekozen. Die keus is gemaakt omdat ik goede resultaten met deze methoden heb behaald bij projecten binnen mijn opleiding aan De Haagse Hogeschool. Naast positieve ervaringen komt UP in aanmerking vanwege de iteratieve mogelijkheden, zo kan er een scheiding worden gemaakt tussen het grafisch te ontwikkelen onderdeel en de patroon herkenning functionaliteit.

#### UP – Unified Process

UP is een iteratieve en incrementele ontwikkelingsmethode die bestaat uit vier fases:

- Inception (begin)
- Elaboration (uitwerking)
- Construction (bouw)
- Transition (overgang)

Deze fases zijn onderverdeeld in een serie van tijdgerelateerde iteraties. Elke iteratie is een opwaardering van het systeem ten opzichte van de vorige versie van het systeem door middel van toegevoegde of verbeterde functionaliteit. In



Figuur 2 - UP, bron: HHS U1d-20110\_2.ppt

Figuur 2 is te zien dat er per iteratie gewerkt wordt aan bijna alle procesdisciplines (requirements, analysis, design, implementation en test). De relevantie en inspanning is echter bij elke opvolgende iteratiestap anders.

#### Iteraties

Het project zal worden uitgevoerd door middel van iteraties. Het voordeel van iteraties is dat het eindproduct in gedeeltes wordt opgebouwd. Het is natuurlijk mogelijk, dat gedurende een project de gebruiker zijn wensen wil bijstellen bijvoorbeeld als gevolg van ontwikkelingen in de markt, zoals een nieuwe prospect. De opdrachtgever kan hierdoor na iedere iteratie het resultaat bekijken en eventuele nieuwe inzichten of eisen doorvoeren voor volgende iteraties. Dit scheelt veel tijd en geld in tegenstelling tot bijvoorbeeld de watervalmethode. Hierbij wordt het project lineair uitgevoerd en hebben veranderingen tijdens het project een grote impact op tijd en geld. Problemen zijn gemakkelijker op te lossen wanneer ze vroeg ontdekt worden. Het is overigens eenvoudiger fouten op te sporen in kleine stukken code dan in een gigantische codemassa die op het einde van een project wordt getest.

<sup>2</sup> [http://en.wikipedia.org/wiki/Unified\\_Process](http://en.wikipedia.org/wiki/Unified_Process)

<sup>3</sup> [http://en.wikipedia.org/wiki/Unified\\_Modeling\\_Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language)

De iteraties zijn opgesteld na het grondig doorlezen van de opdrachtdefinitie en de eerste blik op het te ontwikkelen onderdeel en documentatie. Met als doelstelling:

*‘het ontwikkelen van een geautomatiseerd statistieken weergeefstelsel dat patronen herkent’*

Hierin zijn twee onderdelen te onderscheiden : een geautomatiseerd statistieken weergeefstelsel en een patroonherkenning functie. Naast de voordelen die hierboven zijn beschreven is dit de voornaamste reden dat ik gekozen heb om de afstudeeropdracht in twee iteraties te verdelen.

#### 4.1.2 Bepalen tijdsplanning

Om een tijdsplanning te maken moet er helder zijn welke activiteiten er in het project plaats vinden en hoeveel tijd deze activiteiten in beslag nemen. Dit is grotendeels gedaan door middel van ervaringen uit schoolprojecten en uit de stageperiode in het derde jaar. Voor dit project bestaan de hoofdwerkzaamheden uit:

- Analyse en Onderzoek
- Ontwerpen
- Ontwikkelen en testen

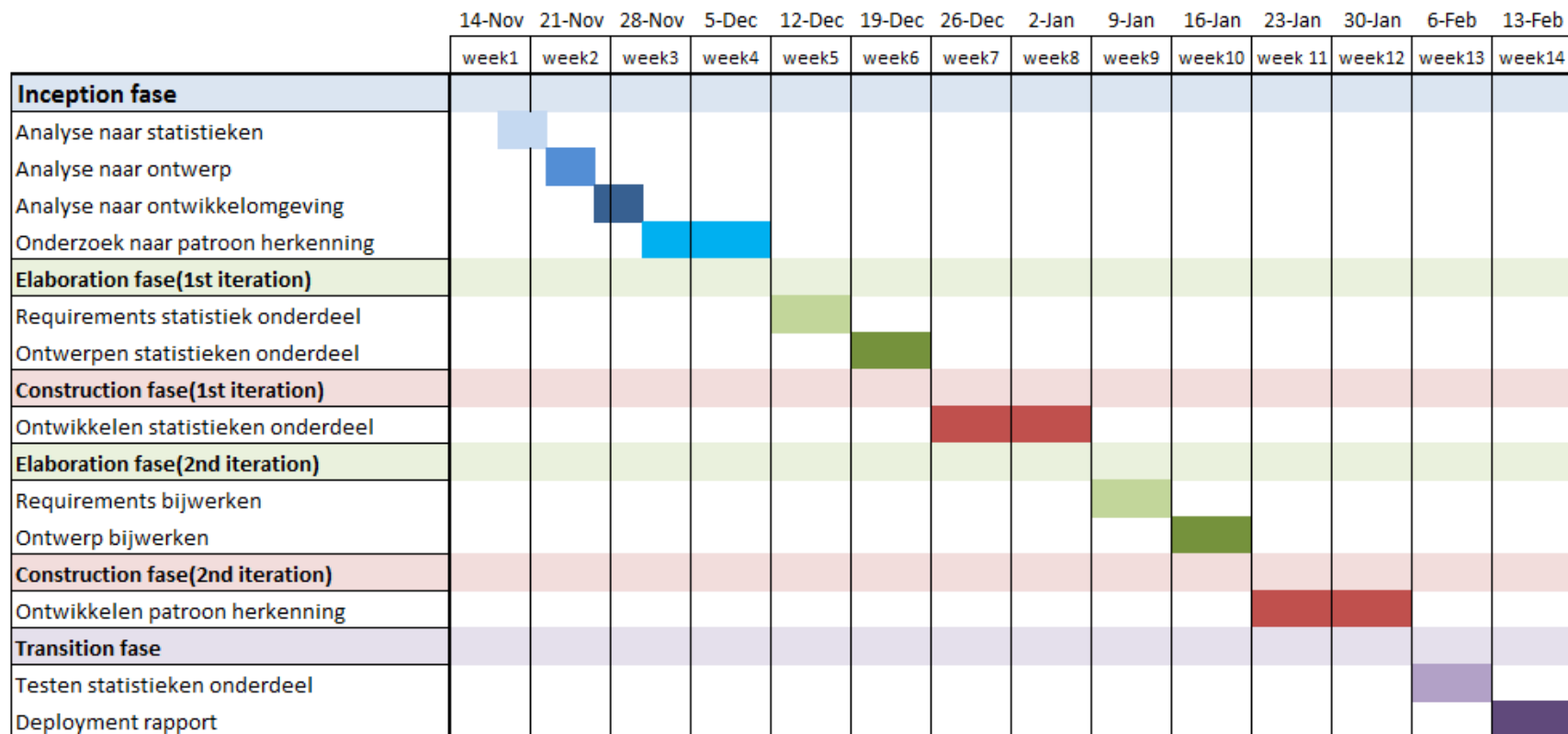
Deze werkzaamheden zijn uitgedrukt per product wat het volgende opleverde:

Product	Werkzaamheden
<b>Onderzoeksrapport</b>	Analyse statistische informatie Analyse grafisch ontwerp Analyse ontwikkelomgeving Onderzoek patroon herkenning
<b>Elaboration rapport</b>	Requirements opstellen Ontwerpen statistieken onderdeel Ontwerpen patroon herkenning functionaliteit
<b>Construction rapport</b>	Programmeren en code documenteren
<b>Testplan</b>	Test omgeving opzetten
<b>Deployment plan</b>	Aanbevelingen en conclusies rapporteren

Met het gebruik van de hierboven weergegeven werkzaamheden, de fasen van de gekozen aanpak en de twee vooraf bedachte iteraties is een globale planning opgesteld. Deze planning is te zien in Figuur 3 op de volgende pagina.

Planning: 17/11/2011

Innosense: ontwikkelen van een geautomatiseerd statistieken weergeefstelsysteem



Figuur 3 - Initiele planning



### 4.1.3 Bepalen van technieken

In deze subparagraaf wordt een beschrijving gegeven van de manier van werken die is aangehouden en de technieken die in de loop van het project zijn gebruikt. Voor de, in de vorige subparagraaf beschreven, werkzaamheden zijn vooraf een aantal technieken bepaald. Een aantal technieken waren bij aanvang al bekend en sommige zijn lopende het project toegevoegd.

#### ***Documentatie techniek***

Documenteren speelt in mijn afstudeertraject een grote rol voor zowel school, Innospense als voor mijzelf. Ik heb een logboek bijgehouden om mijzelf een geheugensteun te bieden bij het bedenken van oplossingen en de motivatie van mijn beslissingen/keuzes door het afstudeertraject heen. Naast het opschrijven van eigen bevindingen zal er ook gecommuniceerd worden met alle betrokken partijen. In mijn geval is dit voornamelijk met mijn bedrijfbegeleiders Bartel Timmermans en Thijs van Nuenen geweest. Punten uit gesprekken werden verwerkt als aantekeningen in het logboek. Naast aantekeningen is ook gebruik gemaakt van versiebeheer. De mogelijkheid bestaat om nieuwe met oude producten te vergelijken en indien nodig terug te vallen bij het maken van fouten. Voor het documenteren en figuren schetsen is gebruik gemaakt van MS Office.

#### ***Analyse/Onderzoek***

Voor het doen van analyses en onderzoek is gebruik gemaakt van verschillende bronnen waaronder interviews, internet en diverse literatuur.

#### ***Ontwerp techniek***

UML, wat wordt gebruikt voor het specificeren, visualiseren en modelleren van softwaresystemen, is een van de bekende technieken. Hoewel UML als onmisbaar wordt gezien binnen UP heb ik er voor gekozen om niet alle technieken van UML toe te passen gezien de omvang van dit project. UML technieken die zijn gebruikt voor dit project zijn UML Use Case diagram, klassendiagrammen en sequentiediagrammen.

#### ***Ontwikkel techniek***

In eerste instantie was het idee om PHP te gebruiken voor het ontwikkelen van het grafisch statistieken weergeef systeem. Dit omdat hier in de opdracht vanuit het bedrijf om werd gevraagd en om dat het bestaande webportaal van het bedrijf voor het grootste gedeelte in PHP is geschreven. Om met PHP te kunnen ontwikkelen wordt er gebruikt gemaakt van een locale web server (WAMP). Welke techniek gebruikt zou worden voor het herkennen van patronen was nog onduidelijk omdat hiervan geen kennis was. In een later stadium van het afstudeertraject is omgeschakeld naar een andere techniek om tot het gewenste resultaat te komen (zie 5.3.2). Deze andere techniek was het gebruik van Javascript en JSON, waarvan geen voorkennis aanwezig was.

## 4.2 Uitvoeren van analyses en onderzoek

Het uitvoeren van analyses en onderzoek had als doel informatie te verzamelen en kennis op te doen. Door deze informatie en kennis vast te leggen in het onderzoeksrapport werd er een basis gelegd en input gecreëerd voor de elaboration fase. Het plegen van onderzoek en analyses werd in de planning verdeeld in een viertal werkzaamheden met ieder hun eigen doel:

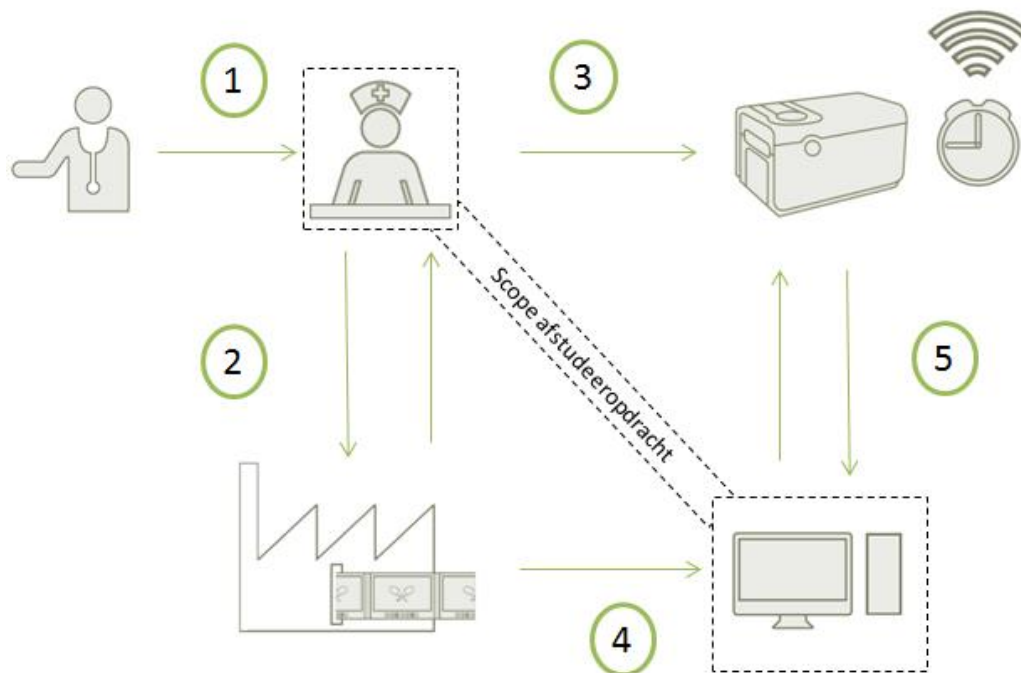
- Analyses statistische informatie: een globaal beeld krijgen van de beschikbare statische informatie en het in kaart brengen van het medicatie verstrekking proces
- Analyse grafisch ontwerp: feedback ontvangen op basis van een grafisch schets ontwerp
- Analyse ontwikkel omgeving: een globaal beeld krijgen van de webportaal architectuur en structuur
- Onderzoek patroon herkenning: kennis op doen op het gebied van patroon herkenning in het algemeen en binnen software ontwikkeling

Elke werkzaamheid is in eigen woorden en in de hierboven weergegeven volgorde uitgeschreven als een subparagraaf waarin het proces en de gemaakte keuzes worden beschreven. De werkzaamheid 'analyse ontwikkel omgeving' is niet als subparagraaf uitgewerkt omdat deze werkzaamheid lopende het project geschrapt is uit de planning (zie H.4.3 Resultaat inception fase).

Het onderzoeksrapport is toegevoegd als bijlage, zie bijlage<sup>ii</sup>.

### 4.2.1 Oriënteren op statistische informatie

Om een globaal beeld te krijgen welke statische informatie relevant en beschikbaar is, heb ik een statistische informatieanalyse uitgevoerd. De eerste stap was het in kaart brengen van het medicatieverstrekking proces. Innospense had hier zelf nog geen figuur of uitleg op papier vastgelegd. Daarom heb ik niet alleen voor de ICT doelgroep maar ook voor de stakeholder doelgroep een begrijpbaar model gemaakt. Dit is gedaan in samenwerking met mijn bedrijfsbegeleider aangezien hij de stappen van het proces wist. Hieronder is het resultaat van het in kaart gebrachte medicatieverstrekking proces weergegeven. In Figuur 4 is ook de scope is van mijn afstudeeropdracht aangegeven.



Figuur 4 - medicatie verstrekking proces

1. De huisarts schrijft medicatie aan patiënt X voor en speelt het recept door aan de apotheek.
2. De apotheek verstuurt de medicatievoorschriften inclusief uitgiftemomenten naar een verpakkingsfabrikant die de medicijnen verpakt per uitgiftemoment. Hierop retourneert de verpakkingsfabrikant een medicatierol aan de apotheek.
3. De apotheker of zorgverlener vult de Medido-dispenser wekelijks met de verkregen medicatierol.
4. De verpakkingsfabrikant stuurt tevens wekelijks een output bestand naar het webportaal van Innospense waar de gegevens van de desbetreffende patiënt in zijn verwerkt zoals de exacte, geplande uitgiftemomenten van de verpakte medicatierol.
5. Het webportaal zet de ontvangen uitgiftemomenten in een weekschema en stuurt deze naar de Medido-dispenser. De Medido-dispenser zal bij elk uitgiftemoment een alarm geven en informatie terug communiceren naar het webportaal om gebruikers statistieken op te slaan.

Het webportaal en de apotheker zitten in de scope omdat de apotheker uiteindelijk gebruik gaat maken van het in de webportaal weer te geven grafisch statistieken onderdeel.

In de hierboven gegeven beschrijving van het figuur is te lezen en te zien dat het webportaal communiceert met de Medido-dispenser. De communicatie tussen het webportaal en de Medido-dispenser is voor mij van belang om te weten welke informatie voor mij beschikbaar is om bijvoorbeeld weer te kunnen geven in de te ontwikkelen grafiek met statistische informatie. Wat ik bijvoorbeeld wilde weten was; wordt alle communicatie opgeslagen? Hoe ziet de inhoud van communicatie er uit van bijvoorbeeld medicatie die te laat, te vroeg, op tijd of niet is uitgenomen?

In een beschikbaar gesteld communicatie protocol document stond alle technische communicatie tussen webportaal en de Medido-dispenser in detail beschreven.

Met gebruik van dit document kon achterhaald worden welke informatie beschikbaar was. Met de vraag “welke gegevens/communicatie denk ik nodig te hebben om een statistieken grafiek mee weer te geven?” heb ik het communicatie protocol document gefilterd op relevante informatie. Een voorbeeld van deze gefilterde informatie<sup>4</sup> was bijvoorbeeld de status:

1. Medicatie uitgenomen: melding Mu=Taken
2. Medicatie niet uitgenomen: melding Mu=Not taken
3. Medicatie te vroeg uitgenomen: melding Mu=Unknown
4. Medicatie te laat uitgenomen: melding Mu=Vergeten

Aan de hand van deze informatie kan bijvoorbeeld de status worden gekoppeld aan een symbool/kleur in de weer te geven statistische grafiek. Een gebruiker zou dan gemakkelijk kunnen zien wat de status van een uitgifte is geweest.

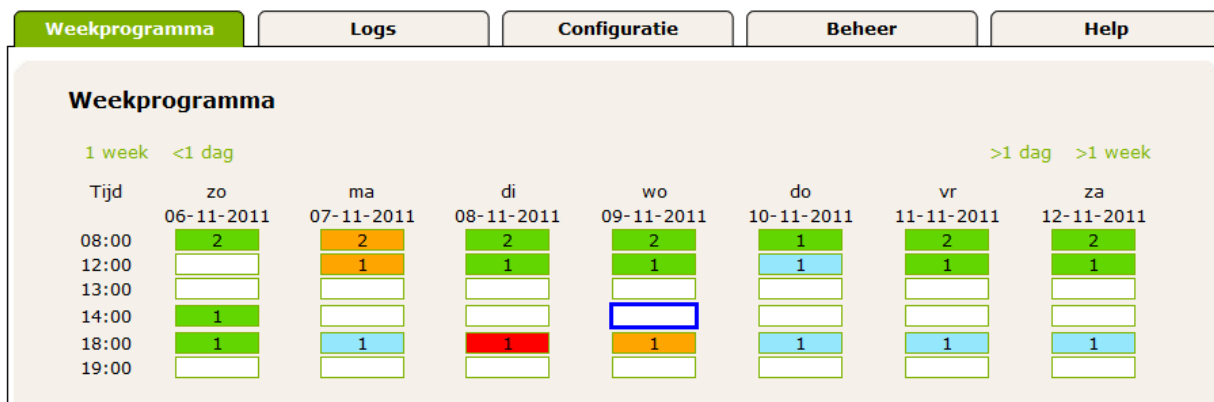
---

<sup>4</sup>Voor een compleet overzicht van de gefilterde informatie zie bijlage<sup>ii</sup> Onderzoeksrapport (H.2.3)

#### 4.2.2 Grafisch statistieken concept ontwerp

Nu het medicatieverstrekking proces in kaart is gebracht en bekend is welke statistische informatie beschikbaar is, wilde ik een grafisch concept ontwerp maken. Het maken van het concept ontwerp had als doel om feedback van de gebruikers(apothekers) en van de opdrachtgever te krijgen. Voordat het ontworpen en ontwikkeld wordt kan met de feedback de wensen van gebruikers en opdrachtgever genoteerd worden. Deze wensen zouden in de elaboration fase dan vertaald kunnen worden naar concrete requirements. Tevens zou ik na kunnen gaan in hoeverre mijn visie overeenkomt met die van mijn bedrijfsbegeleider; daar hecht ik zelf veel waarde aan..

Hieronder is Figuur 5 weergegeven, het huidige grafisch overzicht



Figuur 5 - Huidige grafisch overzicht van het Innospense webportaal

Dit is een weekschema overzicht van een willekeurige patiënt die de Medido-dispenser gebruikt. Links zijn de uitgifte tijden te zien met op elke dag een cijfer ingevuld van de hoeveelheid uit te nemen medicatie. De kleur geeft aan of medicatie op tijd(groen), te laat(blauw), te vroeg (oranje) of niet(rood) is uitgenomen. Uit deze figuur is op te maken dat de Medido-dispenser iedere avond om 18:00 een alarm zal geven om medicatie uit te nemen. Alleen op maandag heeft de patiënt rond die tijd de medicatie op tijd uitgenomen. Er kan ook genavigeerd worden naar de volgende/vorige dag of week.

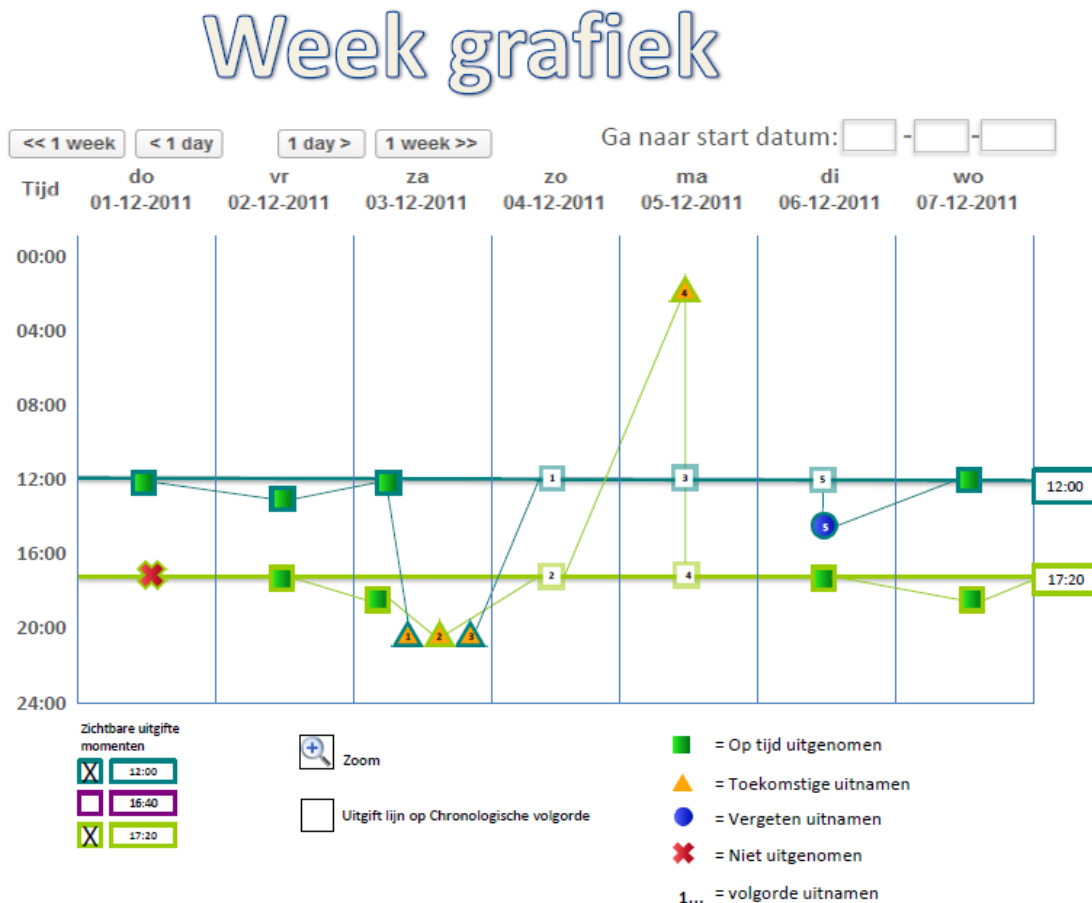
##### **Nadelen oud overzicht**

Qua weergaven en functies is dit een redelijk beperkt grafisch statistieken overzicht. Er kan bijvoorbeeld niet gezien worden hoe 'te vroeg' of 'te laat' medicatie is uitgenomen. Hierdoor is ook geen chronologie te zien in het uitgifte patroon. Het opzoeken van een specifieke dag of week is niet mogelijk. Als er twee maanden terug genavigeerd wil worden zal er acht keer op de knop moeten worden gedrukt tevens is er geen legenda aanwezig.

- Geen tijdspecificatie van te laat of te vroege uitgiften
- Ontbreken van de chronologie
- Geen zoek veld voor specifieke datum
- Geen legenda

### Concept ontwerp

Het doel van het grafisch overzicht is om gebruikers(apothekers) van zoveel mogelijk en verschillende Medido-dispenser gebruikersstatistieken te voorzien. Door de nadelen van Figuur 5 weg te nemen, heb ik een grafische concept versie van de grafiek in MS Office gemaakt die er als volgt uitzag. (het uiteindelijke concept ontwerp V3)



Figuur 6 - Grafisch concept ontwerp van grafiek

Door tijd en dag weer te geven op de x- en y-as worden de nadelen van chronologische volgorde en de precieze tijdsindicatie weggenomen. Een aantal toevoegingen zijn het in- en uitschakelen van zichtbare uitgifte momenten, het inzoomen op specifieke tijden van de dag en het zoeken op datum.

Het schakelen tussen zichtbaarheid is bedacht om zo een minder chaotisch overzicht te hebben. Het inzoomen is bedacht om op elkaar geclusterde uitgifte te kunnen onderscheiden.

Met dit overzicht zou het voor apothekers in een oogopslag duidelijk zijn wat precies de uitgifte momenten zijn geweest voor de desbetreffende patiënt. Wat bijvoorbeeld uit dit nieuwe overzicht wel op te maken valt en niet in het oude overzicht, is dat op zaterdag avond drie vervroegde uitgiften hebben plaats gevonden. Hier uit zou bijvoorbeeld afgeleid kunnen worden dat de patiënt zondag niet thuis aanwezig was.

### Invloed van interviews

Het uiterlijk en toegevoegde functies die hierboven zijn weergegeven is niet zomaar verzonnen. Dit is voortgekomen uit interviews en de verkregen feedback met apothekers en de opdrachtgever. Bij het houden van interviews met de opdrachtgever kon de basis worden geschetst. Met het houden van interviews bij verschillende apothekers kon met de feedback het schetsontwerp verbeterd worden. Twee voorbeelden van vragen die werden gesteld en waarop feedback is gegeven:

*“8. Zijn er nog op of aanmerkingen?”*

*De legenda laat geen kleuren zien en geeft geen uitleg over de cijfers die in de symbolen staan”*

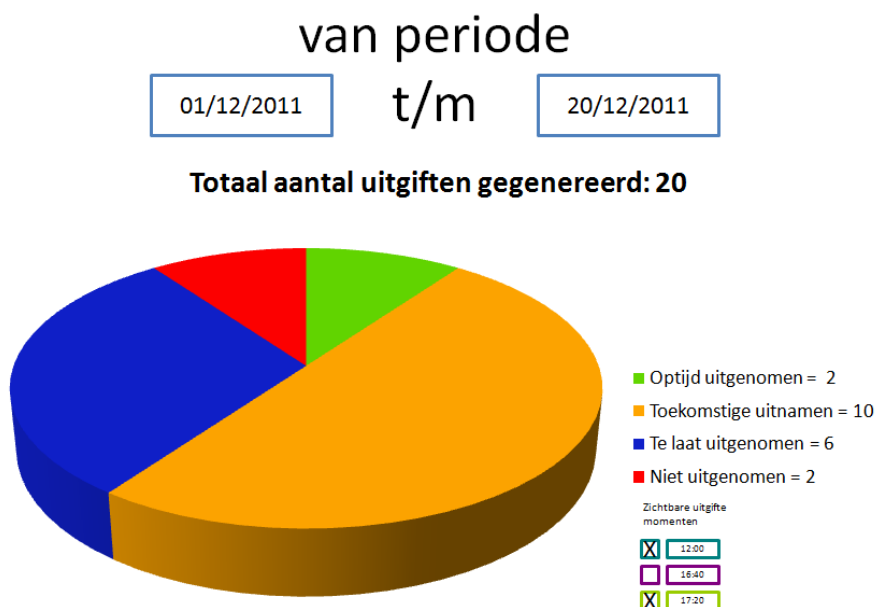
Hierop is de grafiek aangepast en zijn er kleuren toegekend aan de symbolen in de legenda en is de uitgifte volgorde toegevoegd aan de legenda.

*“8. Zijn er nog op of aanmerkingen?”*

*De week grafiek is het meest relevant maar voor het herkennen van patronen is het hebben van een maandgrafiek de meest toegevoegde waarde”*

Aangezien het herkennen van patronen zat verwerkt in mijn opdracht is extra aandacht besteed aan deze opmerking. Bij de concept ontwerpen is daarom ook een grafiek toegevoegd met een maandoverzicht. Voor een volledig overzicht van de gestelde vragen en verkregen feedback zie bijlage<sup>iii</sup>.

Naast grafieken zijn er ook een aantal andere grafische oplossingen geschetst voor het weergeven van specifieke statistieken, zoals een taartdiagram. Het volledige overzicht van geschetste grafieken is terug te zien in bijlage<sup>iv</sup>, grafieken.



Figuur 7 - Statistische gegevens in een taartdiagram

### 4.2.3 (On-)geautomatiseerde patroonherkenning

Een hoofd onderdeel van mijn afstudeeropdracht is naast het ontwikkelen van een geautomatiseerd statistieken weergeef systeem, het toevoegen van een patroonherkenning functionaliteit. Deze functionaliteit zou het mogelijk maken om specifieke uitgifte patronen te herkennen en hiervan de apothekers en/of zorgverlener op de hoogte te brengen. Als resultaat zou dit de therapietrouw van de patiënt bevorderen en Innospense medewerkers van een werklast afhelpen door apothekers zelfstandig het overzicht van patiënten te laten bekijken.

Het doel van het onderzoek naar patroon herkenning was om een globaal beeld te krijgen van patroon herkenning. De vragen die in dit onderzoek centraal lagen waren onder anderen: wat is een patroon, hoe wordt een patroon herkend en hoe kan softwareontwikkeling daarbij een rol spelen.

De opgedane kennis en informatie werd vastgelegd in het onderzoeksrapport. De gedachten was om daarmee de geautomatiseerde patroon herkenning functie te kunnen ontwerpen en ontwikkelen.

Tijdens het uitvoeren van het onderzoek naar patroon herkenning ben ik echter tegen een aantal problemen aangelopen.

#### ***Knelpunten binnen het onderzoek***

Aan het begin van het onderzoek ben ik gestart in Wikipedia om basiskennis op te doen . Het patroon herkenning onderwerp bleek echter veel groter te zijn dan ik van te voren zou kunnen bedenken. Daar heb ik mij niet door laten weerhouden, het onderwerp was een uitdaging geworden. Om dieper op de stof in te gaan is het boek gelezen van Sergios Theodoridis & Konstantinos Koutroumbas, *“Pattern Recognition 4th edition”*. Het wiskunde niveau was echter te hoog waarop hulp is gezocht bij derden (collega's/vrienden/kennissen) die ondersteuning konden bieden op het gebied van patroonherkenning. Helaas was er niemand bekend met het onderwerp patroonherkenning.

In de eerste planning was er twee weken gereserveerd voor het verrichten van onderzoek naar patroon herkenning, dit heeft echter bijna viereneenhalve week geduurd. In de eerste drie weken had ik veel moeite om mij door de grote hoeveelheid informatie te worstelen naar bruikbare informatie voor mijn onderzoek. Hoewel het veel tijd in beslag nam, vond ik het een erg fascinerend onderwerp en liet ik mij hierin erg meeslepen. Pas een week later realiseerde ik mij dat ik al vier weken bezig was met onderzoek verrichten naar patroon herkenning terwijl hier twee weken voor waren gereserveerd. Samen gevat kwamen de volgende knelpunten aan het licht:

- De patroonherkenning literatuur was te complex
- Geen externe bronnen/derden die ondersteuning of uitleg konden bieden op het gebied van patroonherkenning
- De verhouding van geïnvesteerde energie en opgebrachte kennis was te laag.
- Een opgelopen vertraging van 2,5 week.



### ***Voorlopige kennis en oplossing***

In verband met de zojuist genoemde knelpunten heb ik er voor gekozen om het onderzoek naar patroon herkenning stop te zetten. Ook al koos ik er voor om het onderzoek stop te zetten er was wel concrete informatie en kennis verzameld. Deze informatie is alsnog verwerkt in het onderzoeksrapport. Zo is er een set van nuttige en minder nuttige patronen gedefinieerd als training kenmerken.

Nuttig patroon

	Ma	Di	Wo	Do	Vrij	Za	Zo
Week 1	X	-	-	-	-	-	-
Week 2	X	-	-	-	-	-	-
Week 3	X	-	-	-	-	-	-
Week 4	-	-	-	X	-	-	-
Week 5	-	-	-	X	-	-	-
Week 6	-	-	-	X	-	-	-
Week 7	-	-	-	X	-	-	-

Nutteloos patroon

	Ma	Di	Wo	Do	Vrij	Za	Zo
Week 1	X	-	-	-	-	-	-
Week 2	-	X	-	-	-	-	-
Week 3	-	-	X	-	-	-	-
Week 4	-	-	-	X	-	-	-
Week 5	-	-	-	-	X	-	-
Week 6	-	-	-	-	-	X	-
Week 7	-	-	-	-	-	-	X

Deze informatie zou later nog van pas kunnen komen indien de ontwikkeling naar patroon herkenning wordt voortgezet.

Het eigenlijke doel was om het systeem automatisch patronen te laten herkennen en dus ook het systeem te laten definiëren wat een patroon is. De optie was om zo'n nuttig patroon alsnog te gebruiken bij het ontwikkelen van een patroon herkenning systeem. Het patroon zou niet (automatisch)gedefinieerd worden maar (statisch)ingebakken worden binnen de applicatie. Het 'nuttig patroon' wat hierboven staat weergegeven zou een voorbeeld kunnen zijn. Binnen code zouden dit soort patronen statisch gedefinieerd kunnen zijn:

```
Voor Elke(niet uitgenomen medicatie voor de afgelopen x aantal dagen){  
    Als( aantal vergeten medicatie binnen x dagen > y patroon grens){  
        patroon herkent, stuur melding;  
    }  
}
```

Uiteindelijk is er voor gekozen om ook de statische patronen te laten voor wat het was. Een statisch patroon zou namelijk niet aan de complexiteit voldoen zoals afgesproken was in het afstudeer plan. In de evaluatie wordt dit nader toegelicht (zie H.7.1.1).

### 4.3 Resultaat inception fase

Als resultaat uit de inception fase zijn het plan van aanpak en het onderzoeksrapport tot stand gekomen en is de planning gewijzigd.

Het plan van aanpak heb ik gebruikt om de opdrachtgever te laten zien wat mijn werkzaamheden waren en mijn visie op het project. Aan de hand van dit document heeft de opdrachtgever aangegeven dat we op één lijn zaten wat betreft het project.

Het onderzoeksrapport bevatte uiteindelijk veel informatie over de beschikbare statistische informatie, de feedback voor het grafisch concept ontwerp en gedeeltelijke informatie over patroon herkenning.

Het analyseren van de ontwikkel omgeving stond op de planning om uit te voeren voor het plegen van onderzoek. Echter vanwege de afwezigheid van de externe programmeur werd deze gepland na het plegen van onderzoek.

Bij het plegen van onderzoek naar patroon herkenning is er echter een vertraging opgelopen van 2,5 week door een aantal knelpunten. Dit had een aantal gevolgen:

- Vanwege de knelpunten en opgelopen vertraging is direct begonnen aan de elaboration fase
- Onduidelijk was of de tweede iteratie nog gebruikt zou kunnen worden voor het maken van een (statisch) patroon herkenning functionaliteit.
- Vanwege het opgelopen tijdverslies zullen er werkzaamheden geschrapt moeten worden

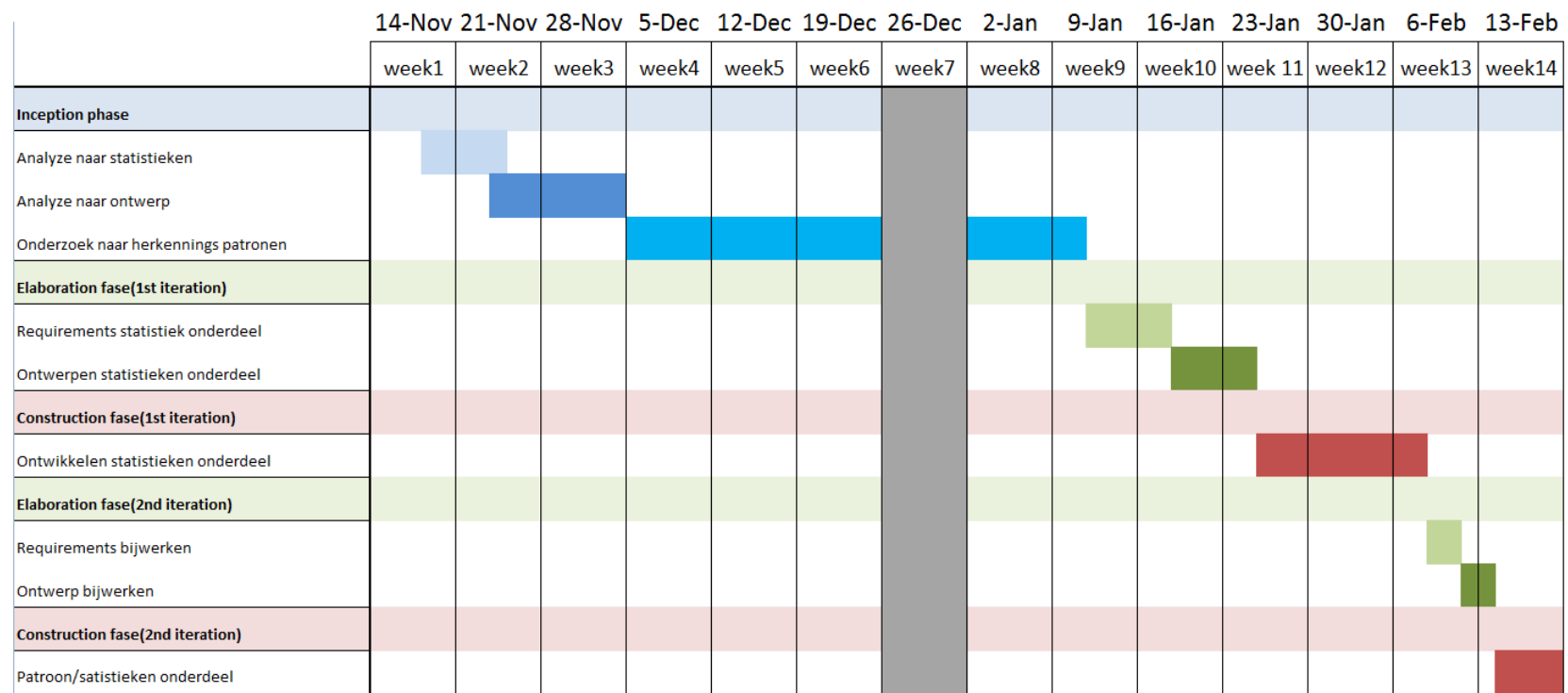
Het doel van analyse naar de ontwikkel omgeving was om de requirements nog voor de 1<sup>e</sup> iteratie aan te scherpen voor het grafisch statistieken onderdeel. Het was nog onzeker of de patroon herkenning functionaliteit ingebouwd zou worden. Ik had bedacht dat in plaats van patroon herkenning, de 2<sup>e</sup> iteratie kon dienen om de requirements voor het grafisch statistieken onderdeel alsnog aan te scherpen.

De keuze om de 2<sup>e</sup> iteratie voor het statistieken onderdeel of voor patroon herkenning te gebruiken hoefde niet perse in dit stadium vastgelegd te worden. Als het ontwikkelen van het grafische statistieken onderdeel bijvoorbeeld sneller gaat dan verwacht, zou als nog de 2<sup>e</sup> iteratie gebruikt kunnen worden voor het ontwikkelen van (statische) patroon herkenning.

Met deze gevolgen is er een wijziging gemaakt in de planning die is weergegeven op de volgende pagina. In deze planning is te zien dat het onderzoek naar patroon herkenning 2,5 week langer duurt en dat de analyse naar de ontwikkel omgeving is geschrapt van de planning. In een later stadium is er gekozen om de 2<sup>e</sup> iteratie te gebruiken voor het statistieken onderdeel, in H5.4 wordt duidelijk waarom. De transition fase is samen met het deployment plan weggefallen.

Planning: 27/12/2011

Innospense: ontwikkelen van een geautomatiseerd statistieken weergeefstelsysteem



Figuur 8 - Planning na inception fase

## 5 Eerste iteratie

Volgens de nieuwe planning, die hierboven zojuist is weergegeven, is eerst begonnen met de elaboration fase. In een elaboration fase worden de eisen van het systeem vastgesteld en de systeemarchitectuur in kaart gebracht met als product het elaboration rapport. Bij het opstellen van de requirements is gebruik gemaakt van UML modellering technieken. In dit hoofdstuk worden de volgende paragrafen behandeld:

In de eerste paragraaf 'Vaststellen eisen' wordt beschreven hoe het proces verliep voor het in kaart brengen en prioriteren van de requirements.

In de tweede paragraaf 'Use cases in kaart brengen' is beschreven hoe de requirements schematisch in kaart zijn gebracht met gebruik van onder anderen een Use Case diagram.

In de derde paragraaf 'ontwerp en ontwikkel omgeving verkennen' zijn specifieke werkzaamheden beschreven die invloed hebben gehad op het verloop en ook het beëindigen van de eerste iteratie.

In de vierde paragraaf 'Resultaat eerste iteratie', wordt beschreven wat de resultaten en gevolgen waren die uit de eerste iteratie volgden met terugblik op neven activiteiten rondom deze iteratie.

## 5.1 Vaststellen eisen

Voor het vaststellen van de requirements worden benodigde karakteristieken en kwaliteiten van het systeem geïdentificeerd, die voor het op te leveren eindproduct relevant zijn en meerwaarde bieden voor de ontwikkelaar/stakeholders. De requirements moeten testbaar zijn en getoetst worden op hun bruikbaarheid en haalbaarheid. Dit gebeurt door de requirements te verwerken in het testplan tijdens de Construction fase.

De opgestelde requirements zijn ontstaan door de volgende acties te ondernemen:

- Analyses en onderzoek vertalen naar requirements
- Requirements prioriteren

### ***Analyses en onderzoek vertalen naar requirements***

De informatie en kennis die is opgedaan uit de analyses en het onderzoek is vastgelegd in het onderzoeksrapport. In combinatie met de feedback van de interviews met apothekers en de opdrachtgever is die informatie vertaald naar requirements en vastgelegd.

Een voorbeeld van zo'n vertaling is bijvoorbeeld de feedback uit een interview om de grafiek in maandoverzicht weer te geven.

Gra04	Het grafisch overzicht heeft een maand overzicht	S
-------	--	---

Hierboven is een voorbeeld te zien van een enkele requirement. Hierin is als eerst een ID tag te zien, daarna een beschrijving van de requirement en als laatste een prioriteit. Voor een compleet overzicht van alle requirements zie bijlage<sup>v</sup>, elaboration rapport(H.2).

### ***Requirements prioriteren***

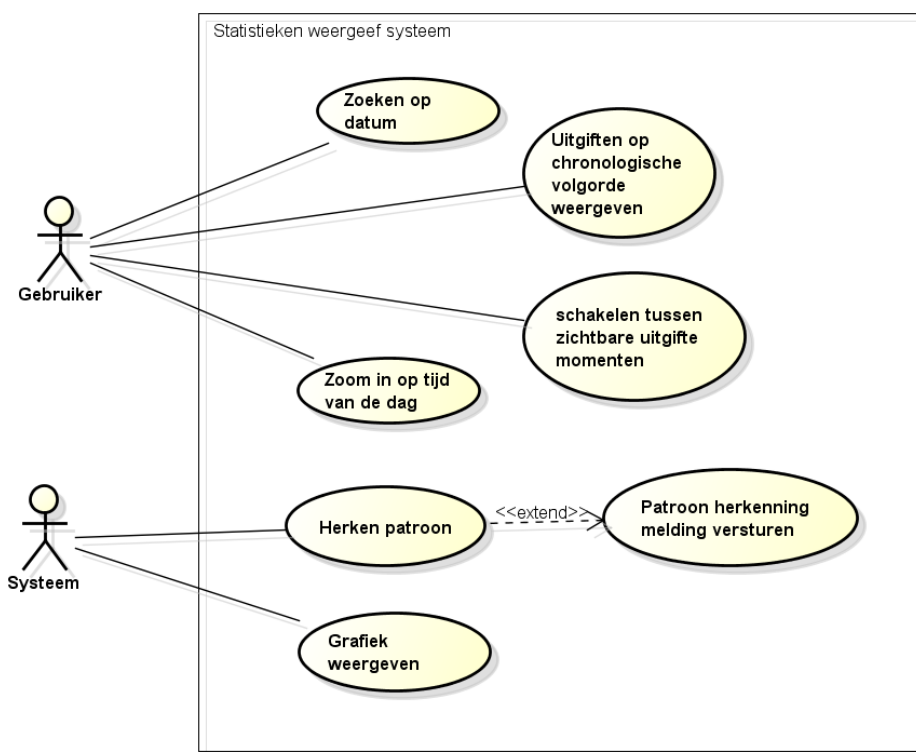
Zoals in het juist gegeven voorbeeld van een requirement staat er een 'S' voor de prioriteit. In samenwerking met mijn bedrijfsmentor heb ik per requirement prioriteit toegewezen met de MoSCoW<sup>5</sup> methode. De reden om prioriteit toe te kennen was voornamelijk om zo veel mogelijk in een volgorde te werken. Daarnaast konden minder belangrijke requirements een minder hoge prioriteit krijgen in verband met het opgelopen tijdverlies. Het voorbeeld Gra04 heeft als prioriteit een 'Should have' gekregen, het zou eigenlijk ontwikkeld moeten worden echter vanwege tijdverslies heeft het niet de hoogste prioriteit gekregen. Het voordeel is door deze requirement op te nemen kan er tijdens de ontwerp fase rekening mee gehouden worden in de ontwerp structuur. Dit om bijvoorbeeld gemakkelijk over te schakelen van week- naar een maandoverzicht.

---

<sup>5</sup> MoSCoW methode die de hoofdletters gebruikt om een prioriteit toe te kennen: Must have, Should have, Could have en Won't Have

## 5.2 Use cases in kaart brengen

Om de requirements schematisch in kaart te brengen heb ik ze gemodelleerd met behulp van UML. Zo is er een Use case diagram met bijbehorende Use case beschrijvingen gemaakt. Hiermee heb ik de processen binnen het systeem in kaart gebracht. Dankzij het maken van deze modellen word je gedwongen om na te denken over de processen en kom je weer achter nieuwe wensen en mogelijkheden. Daarnaast is het een goede manier om aan de opdrachtgever(en eventuele stakeholders) te laten zien welke functies er op het systeem kunnen worden uitgevoerd en hoe het ongeveer gaat werken. In Figuur 9 hier onder is het gemodelleerde Use case diagram te zien. De Use case beschrijvingen zijn opgenomen in het Elaboration rapport (H.4.2).



Figuur 9 - Use Case diagram

## 5.3 Bepalen van het architecturaal design

Nu de belangrijkste wensen en eisen boven water zijn, kunnen deze gemodelleerd worden in een architectuur. Een systeemarchitectuur is het conceptuele model dat de eigenschappen, het gedrag, de relaties en de weergaven van een systeem bepaalt. Het documenteren hiervan vergemakkelijkt de communicatie tussen stakeholders en de uitvoering van beheer en onderhoud en maakt het hergebruik mogelijk van componenten tussen verschillende projecten/producten.

Op basis van UML heb ik de systeemarchitectuur en systeemdelen kunnen modelleren en vast kunnen leggen <sup>6</sup>. Deze paragraaf is opgedeeld in een twee sub paragrafen waarin wordt beschreven hoe de architectuur precies tot stand is gekomen en wat daarbij de belangrijkste beslis en keuze momenten waren in de eerste iteratie.

### 5.3.1 Client – Server architectuur

Ergens binnen de bestaande Innospense architectuur zou mijn project geïntegreerd worden. Het was duidelijk dat externe partijen een grote rol spelen bij het compleet maken van de Innospense architectuur. Om meer duidelijkheid van deze architectuur te krijgen heb ik het design document van Innospense onder de loep genomen.. Op de volgende pagina is een schematisch overzicht van de Innospense architectuur gegeven.

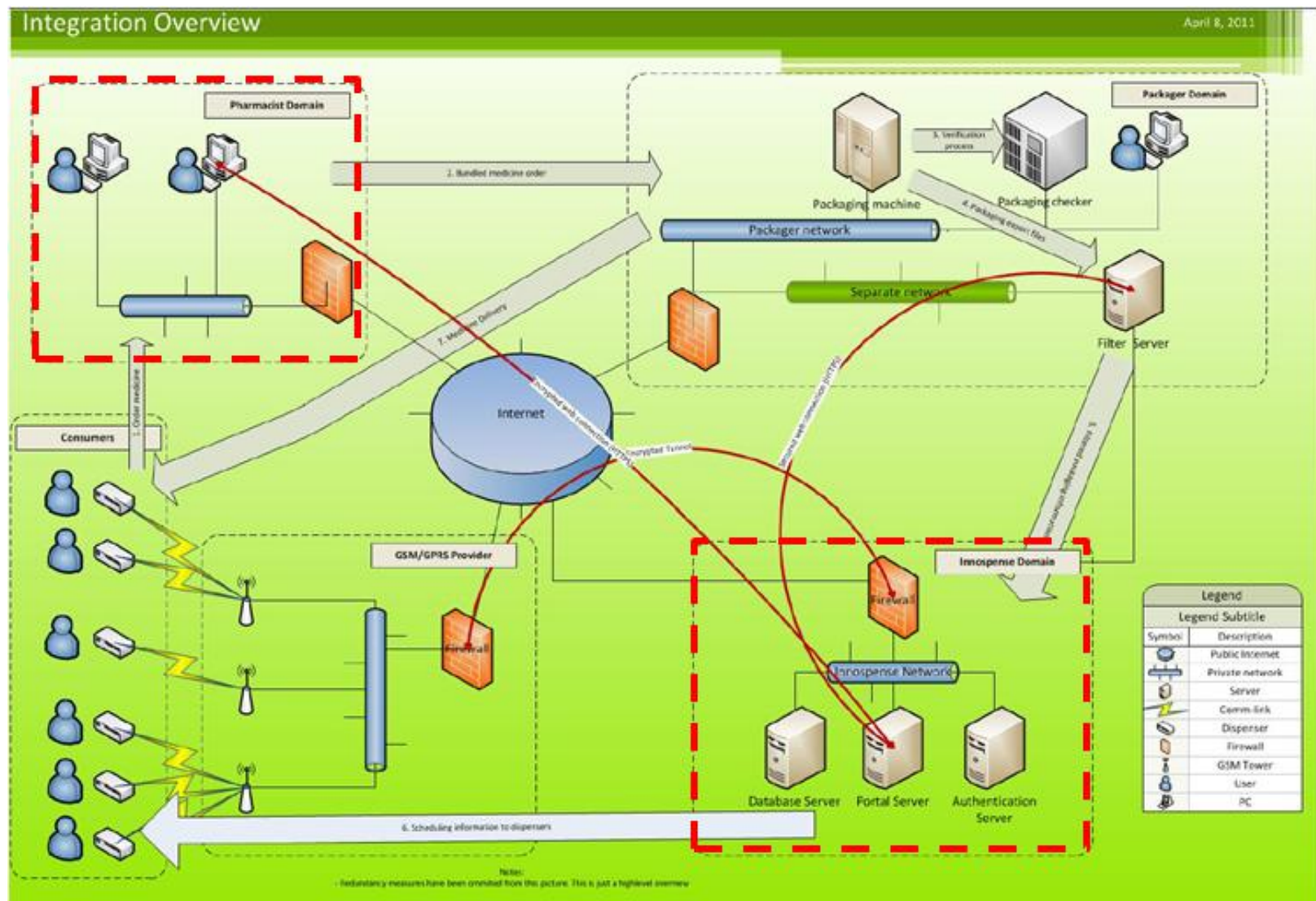
In Figuur 10 op de volgende bladzijde zijn een 5 tal domeinen te zien:

- Innospense
- Farmaceuten
- Consumenten(patiënten)
- GSM Providers
- Verpakking fabrikanten

Voor dit project is echter alleen de communicatie tussen het farmaceuten en Innospense domein van belang. Namelijk alle informatie die nodig is voor het tekenen van een grafiek met statistische gegevens is beschikbaar op de servers van het Innospense domein en de grafieken zijn tot dusver alleen bedoeld voor de farmaceuten.

---

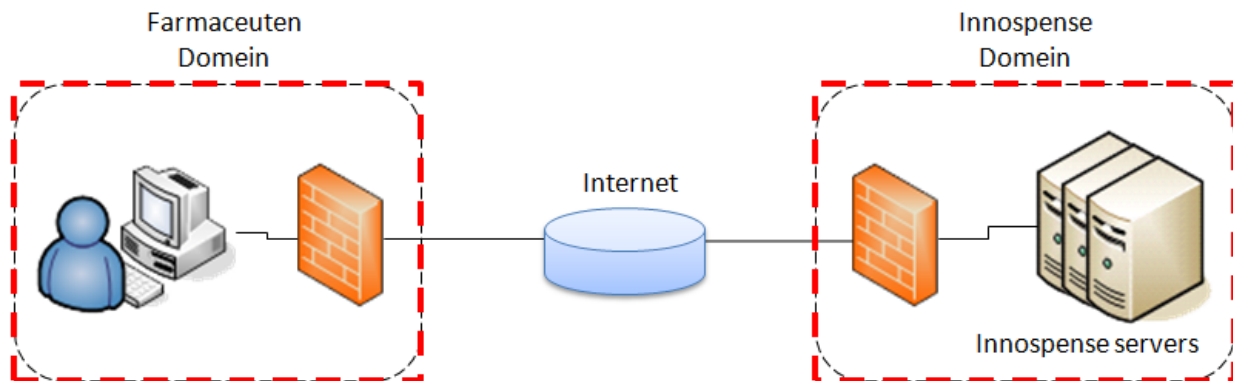
<sup>6</sup> Zie bijlage<sup>v</sup>, Elaboration rapport



Figuur 10 - Innospense architectuur, de rode arcerin geven de scope weer

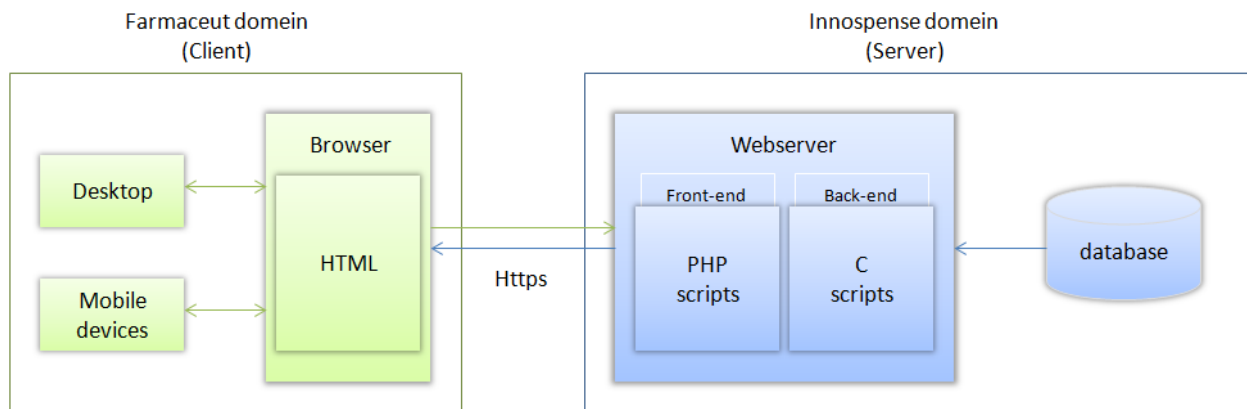


Omdat alleen het Innospense en het farmaceuten domein van toepassing is kan de architectuur voor dit project beknopter worden weergegeven.



In het figuur hierboven is een client-server architectuur ontwerp te zien. Bij een groter ontwerp zou er misschien nog een 'MultiTier' architectuur van toepassing zijn waarbij er een extra laag kwam voor de gegevensopslag. Echter om het beknopt te houden is er gekozen om het als een 2-tier(client-server) architectuur weer te geven.

Door te kijken naar de onderliggende structuur van de twee domeinen heb ik een basis architectuur model afgeleid die alleen de systeemdelen weergeeft die belangrijk zijn voor dit project.



**Figuur 11 - Architectural design**

Zo is in Figuur 11 hierboven te zien dat de Innospense webportaal wordt gehost op een web server die gebruik maakt van de PHP scripting taal als front-end en C gebruikt als back-end programmeertaal. De database bevat alle statistische informatie die nodig is voor dit project.

Het Innospense domein stelt de server voor waarop het Innospense webportaal draait. Het Farmaceuten domein kan het webportaal bereiken door op hun desktop of mobiel apparaat via hun browser naar de website te navigeren. Het doel van mijn opdracht is om het statistieken weergeef systeem in het webportaal te integreren die statistische Medido gebruikers informatie weergeeft. De data die nodig is om deze grafiek te tekenen zit opgeslagen in het database binnen het Innospense domein.

### 5.3.2 Server-side scripting vs. Client-side scripting

Tijdens het ontwerpen van het statistieken onderdeel was een van de nevenactiviteiten, het opdoen van kennis met de programmeertaal PHP. Dit had als doel om kennis op te doen van de ontwikkel omgeving voor het te ontwikkelen systeem. Het idee is immers om een grafisch statistieken weergeef systeem in het webportaal te ontwikkelen. Het probleem was echter dat PHP zeer inefficiënt bleek te zijn en dat er een andere oplossing was voor het architectuur ontwerp.

#### **Probleem**

Om een grafisch ontwerp te programmeren wordt bij PHP vaak gebruik gemaakt van externe bibliotheken, ook wel library of API genoemd. Bij PHP is de, voor mij, meest bekende library om grafisch tabellen of grafieken weer te geven, JpGraph. Bij het doornemen van de online documentatie van JpGraph<sup>7</sup> viel het volgende op: Omdat PHP en dus ook JpGraph server-side scripting is maakt JpGraph op de web server eerst een volledige grafiek, zet deze om in een afbeelding en stuurt deze terug naar de browser van de gebruiker als resultaat. Voor het grafisch concept ontwerp zou dit betekenen dat, voor elke interactie met de grafiek(denk aan zichtbaar maken van een uitgiftelij): er zal eerst een request naar de server gestuurd worden, de server zal een functie uitvoeren op de server en het resultaat omzetten naar een nieuwe grafiek, de grafiek wordt omgezet naar een plaatje en uiteindelijk wordt het plaatje als result terug sturen naar de cliënt. Hier kunnen de volgende nadelen uit worden afgeleid:

- De gehele website zal opnieuw ververst worden
- Vanwege het gebruik van plaatjes is er een hoge banbreedte nodig
- Het rekenwerk wordt door de server gedaan.

Met gebruik van een server-side scripting language zou het betekenen dat er zo goed als geen directe interactie mogelijk zou zijn zoals in de requirements als eis werd gesteld.

---

<sup>7</sup> Website online documentatie JpGraph <http://goo.gl/JgyqX>

### **Criteria**

Een client-side programmeertaal zou als oplossing kunnen dienen. Met het gebruik van een client-side programmeertaal zou het betekenen dat bij elke interactie waarop de grafiek grafisch anders wordt weergegeven het volgende gebeurt: een interactie wordt uitgevoerd, de browser wijzigt direct de grafische weergave van de grafiek en gebruikt hierbij de rekenkracht van de browser pc. De bandbreedte wordt bespaard en de werklast op de server vermindert omdat de browser pc de rekenkracht op zich neemt. Indien er toch nieuwe data nodig is afkomstig van de server kan een client-side scripting language gebruik maken van AJAX<sup>8</sup>. Hierbij worden gegevens asynchroon opgehaald van een web server. Hierdoor hoeven webpagina's niet meer in hun geheel verversd te worden. Dit is een efficiënte manier van omgaan met bandbreedte wat vooral een grote rol speelt bij het gebruik van mobiele apparaten zoals smartphones. Tevens zal de gebruikerservaring positiever zijn omdat niet de gehele webpagina opnieuw geladen hoeft te worden.

In samenwerking met de externe programmeur zijn een aantal eisen opgesteld voor het kiezen van een geschikte programmeertaal:

- Een client-side programmeertaal
- Kan gebruik maken van AJAX(of soortgelijke techniek)
- Ondersteund directe gebruikers interactie
- Wordt ondersteund door verschillende (mobiele)internetbrowsers
- Is gratis om te gebruiken
- Bevat voldoende documentatie
- Heeft de mogelijkheid om aan de Use cases te voldoen

De reden voor het hebben van deze strenge eisen had te maken met de flexibiliteit van de omgeving waar de software draait. Het bleek dat apothekers een zeer stroeve instelling hebben wat betreft software die bij hun gebruikt wordt. Het downloaden van een nieuwe browser zoals Firefox of Google Chrome zou er dan ook niet inzitten. Uit verontrustende bevindingen is zelfs gebleken dat er bij sommige apothekers nog Windows 95 systemen worden gebruikt.

---

<sup>8</sup> <http://en.wikipedia.org/wiki/AJAX>

## Oplossing

Als oplossing kwam de cliënt-side scripting language Javascript<sup>9</sup> het meest in aanmerking om te gebruiken. Andere overwogen oplossingen vielen vrijwel direct af om de volgende redenen:

- Silverlight – De vereiste Microsoft ontwikkel omgeving is niet gratis en wordt niet gebruikt bij Innospense
- HTML 5 Canvas – Wordt niet (volledig)ondersteund door Internet Explorer 7 en 8 die bij de meeste apothekers wordt gebruikt.
- Flash – wordt niet ondersteund door Mobiele iOS apparaten.

Om te kunnen controleren of voldaan kon worden aan de gestelde eisen is als kort experiment ontwikkeld met JavaScript. Dit om te kijken welke externe library kon worden gebruikt om te voldoen aan de eisen die waren gesteld.

Libraries:

- JQuery – Deze Javascript library maakt het mogelijk om Asynchroon data te versturen en ontvangen van een webserver
- JSXGraph<sup>10</sup> – Deze Javascript library werkt op alle geteste browsers, bevat voldoende documentatie om mee te kunnen werken en bevat bestaande voorbeelden die dicht bij de te ontwikkelen Use cases komen. Dit in tegenstelling tot Google Chart Tools(ontbrekende documentatie en functies), JScharts(HTML5) en AMcharts(HTML5/Flash), en daardoor niet in aanmerking kwamen.

Omdat het wisselen van programmeertaal door mij werd gezien als een wijziging met veel invloed op de requirements en het ontwerp, heb ik besloten om de eerste iteratie af te sluiten. Specifieke wijzigingen en keuzen zouden dan kunnen doorgevoerd worden in de tweede iteratie.

---

<sup>9</sup> <http://nl.wikipedia.org/wiki/Javascript>

<sup>10</sup> <http://jsxgraph.uni-bayreuth.de/wp/examples/>

## 5.4 Resultaat eerste iteratie

Als resultaat uit de eerste iteratie is het elaboration rapport gemaakt. Tevens is er een klein ontwikkel experiment geweest waarin een alternatief is gevonden voor de inefficiënte server-side programmeertaal PHP.

### ***Elaboration rapport***

Het elaboration rapport bevat requirements met een prioriteit, een Use case diagram, Use case beschrijvingen en een architectuur. Dit rapport heb ik gebruikt om de opzet en architectuur van het project te laten zien zowel aan mijn opdrachtgever als aan de externe programmeur. Dit heb ik gedaan om eventuele feedback te ontvangen voor de tweede iteratie en te kijken in hoeverre ik en de opdrachtgever op één lijn zaten. Er was nogal wat (positieve) kritiek over onduidelijkheden betreffende de modellering van de architectuur. Een voorbeeld was dat er geen verantwoordelijkheden toegekend waren aan delen van de architectuur en dus niet duidelijk was welk systeem waarvoor verantwoordelijk was. Dit was feedback die meegenomen kon worden naar de tweede iteratie.

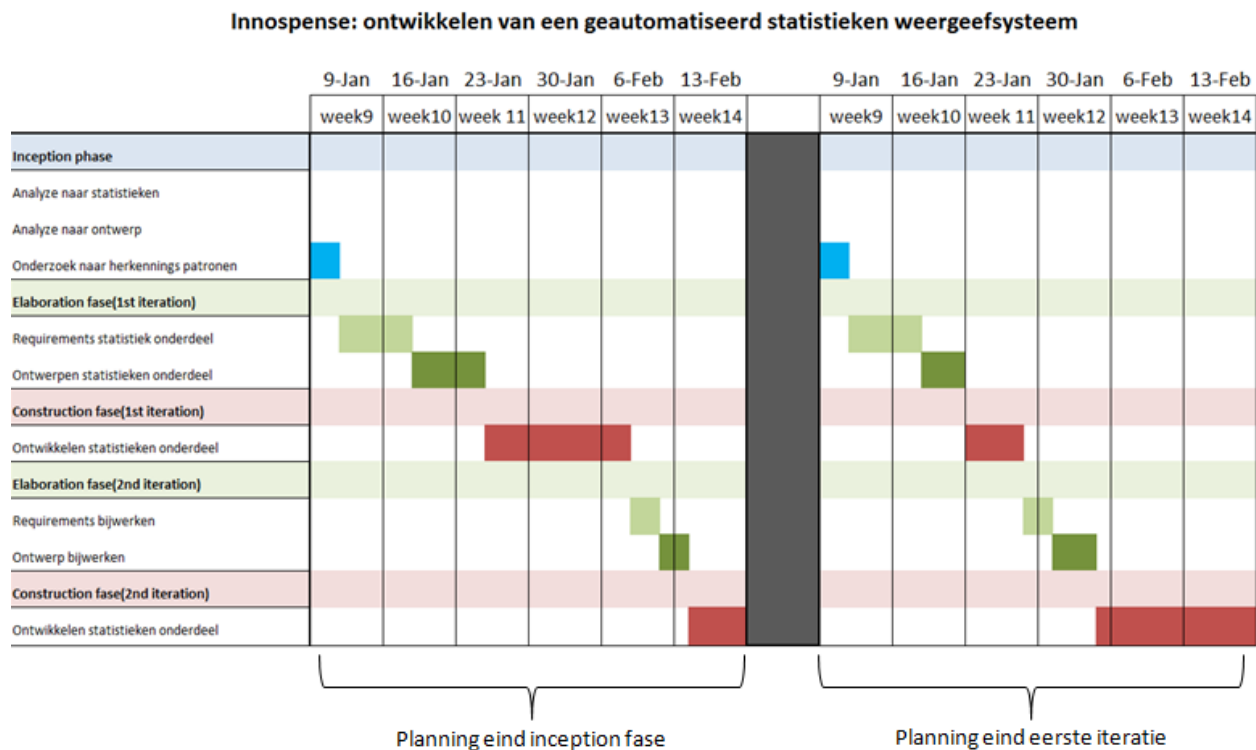
### ***Kort ontwikkel onderzoek***

Door de bevindingen die waren gedaan op het gebied van efficiënt ontwerp bleek het gebruik van PHP niet geschikt te zijn voor dit project. In plaats daarvan is er gekozen voor een client-side scripting language Javascript die een aantal voordelen had ten opzichte van PHP (zie H.5.3.2 Server-side scripting vs. Client-side scripting). Door te kiezen voor een andere programmeertaal kwamen er gelijk een aantal requirements bij. Een voorbeeld zijn de niet-functionele requirements die beschrijven dat de grafiek ondersteund wordt op verschillende internetbrowsers. Met de verandering die het ontwikkel onderzoekje met zich mee bracht heb ik de keuze genomen om een punt te zetten achter de eerste iteratie. Alle wijzigingen bij elkaar opgeteld vond ik dit namelijk een iteratie waardig.

Door de keuze te maken om nu al (niet volgens planning) door te gaan naar de tweede iteratie, zal de patroonherkenning functionaliteit hoogst waarschijnlijk niet meer ingebouwd worden. Dit had ook te maken met de nieuwe omgeving die de externe Javascript library JSXGraph met zich mee bracht.

De nieuwe planning die was opgesteld aan het einde van de inception fase was dus niet meer van toepassing. Op papier zou de planning er voor de tweede iteratie als volgt uitzien. Links is de planning die was opgemaakt aan het eind van de inception fase en rechts is de planning die is opgemaakt aan het einde van de eerste iteratie.

**Planning: 25/01/2012**



**Figuur 12 - Vergelijking van planning aan het eind van de inception fase ten opzichte van het einde eerste iteratie**

## 6 Tweede iteratie

De tweede iteratie is gehouden in verband met wijzigingen op het gebied van requirements, ontwerp en ontwikkel omgeving. Tijdens de tweede iteratie waren de hoofdwerkzaamheden het doorvoeren van wijzigingen in de requirements en ontwerp en is er ontwikkeld.

In de eerste paragraaf wordt beschreven wat de belangrijkste wijzigingen waren ten opzichte van de eerste iteratie.

Omdat er veel aandacht is besteed aan het ontwerp van het systeem is er een tweede paragraaf gemaakt voor het opstellen van het klassendiagram. Om de totstandkoming van het klassendiagram in detail te beschrijven komen naast een klassendiagram ook sequentiediagrammen en een domein model aanbod.

In de derde paragraaf wordt het ontwikkel proces onder de loep genomen waarbij de ontwikkel technieken, knelpunten met bijbehorende oplossingen en het opzetten van een testplan worden beschreven.

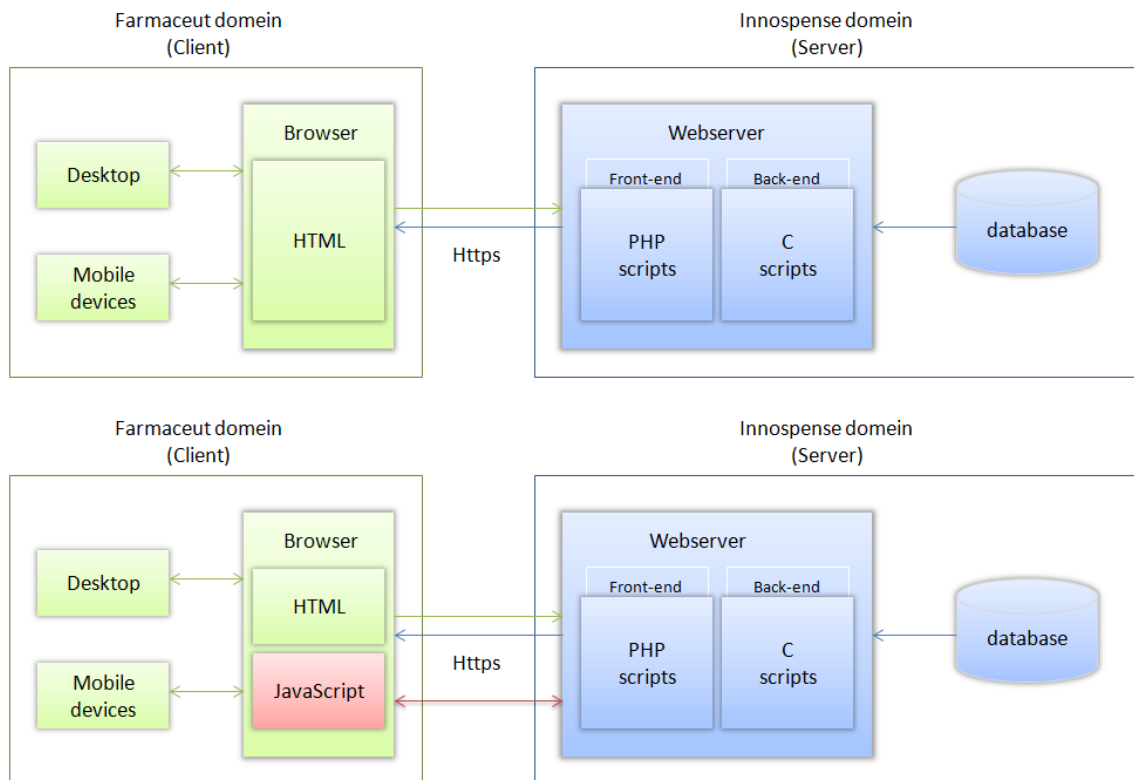
In de vierde paragraaf 'resultaat tweede iteratie' wordt beschreven wat de resultaten waren die uit de tweede iteratie volgden met terugblik op het ontwerp en ontwikkel proces.

## 6.1 Wijzigingen ten opzichte van de eerste iteratie

Met de conclusie die is getrokken in de eerste iteratie bleek dat Javascript de oplossing zou bieden. Deze keuze zou gevolgen hebben voor de requirements en het client-server architectuur ontwerp. In deze paragraaf worden de belangrijkste wijzigingen en keuze momenten beschreven.

### **De architectuur**

Het doel van dit project is om een grafisch statistisch overzicht op de browser van de gebruiker weer te geven. Om een interactieve grafiek weer te geven is gekozen voor de client-side scripting language Javascript. Dat betekent dat het Javascript gedeelte aan de client-side, op de browser van de gebruiker, zal worden gedraaid. Javascript maakt het tevens mogelijk om via AJAX asynchroon gegevens met de server uit te wisselen. Het oude architectuur model(boven) verandert in een nieuw model(onder):



**Figuur 13 - Architectural design zonder Javascript(boven) en met Javascript(onder)**

Met de rode lijn die loopt tussen het Javascript onderdeel naar de web server wordt de AJAX asynchrone communicatie aangegeven. Binnen het Javascript onderdeel zal de grafiek gebouwd en weergegeven worden aan de gebruiker waarbij directe interactie mogelijk is.



### ***De verantwoordelijkheden***

Binnen dit client-server model zullen er verschillende verantwoordelijkheden worden belegd.

De verantwoordelijkheden voor de server:

- Het authenticeren van de gebruiker die gegevens opvraagt
- De gegevens verzamelen uit het database
- De gegevens beschikbaar stellen voor de client

De verantwoordelijkheden van de client:

- Gegevens opvragen bij de server
- De grafiek bouwen
- De grafiek interactief laten reageren op input van de gebruiker
- Asynchroon data opvragen aan de server d.m.v. AJAX

Voor het authenticeren van de gebruiker en het beschikbaar stellen van gegevens voor de client stelde de externe programmeur zich verantwoordelijk in verband met de veiligheidsintegriteit. Dit betekende dat ik geen directe controle zou hebben over de beschikbare data uit het database.

## 6.2 Bepalen van het klassendiagram

Het klassendiagram is de belangrijkste bouwsteen in object georiënteerd modelleren. Het wordt gebruikt zowel voor algemene conceptuele modellering van het systeem als voor het vertalen van de architectuur modellen naar programmeerbare code. Om zo'n klassendiagram te ontwerpen zal een aantal andere zaken ook in kaart gebracht worden, zoals hoe ziet de uitwisseling van gegevens er uit tussen verschillende systeemdelen en wat is de structuur van deze communicatie. Omdat ik zelf geen toegang zal krijgen tot de database en niet zelf de benodigde data kan samenstellen zal extra nadruk worden gelegd op welke data ik nodig zal hebben en in welke structuur dit is. Om dit volledig te kunnen beschrijven is deze paragraaf opgedeeld in een drietal sub paragrafen die het proces voor het bepalen van het klassendiagram volledig in kaart brengen.

In de eerste sub paragraaf wordt gekeken naar welke data waar nodig is door gebruik te maken van een domeinmodel.

In de tweede sub paragraaf wordt gekeken hoe deze data wordt gecommuniceerd van bijvoorbeeld server-to-client.

In de derde sub paragraaf wordt beschreven hoe alle informatie tot een ontwerp van het klassendiagram leidde.

### 6.2.1 Welke data is nodig voor de grafiek?

Binnen informatica worden voor het bepalen van de data opslag domeinmodellen en database modellen gemaakt. Binnen een domeinmodel<sup>11</sup> wordt ontworpen wat de structuur van het datamodel is en wat de onderlinge relaties binnen de structuur zijn. Deze worden in een databasemodel omgezet naar tabellen met velden.

Voor het opstellen van een klassendiagram was nog niet duidelijk welke data in welke vorm beschikbaar was en hoe deze beschikbaar zou worden gesteld. Om dit beter in kaart te brengen had ik bedacht om een domein model te ontwerpen die een samengevat en gemakkelijk overzicht weergeeft van de benodigde data. Tevens zou het domeinmodel gebruikt kunnen worden om aan de externe programmeur aan te geven welke data nodig is voor het maken van de grafiek. Dit zou belangrijk zijn omdat de externe programmeur mij geen toegang verleent voor het aanspreken van het database.

#### ***Benodigde data en beschikbare data***

Door te kijken naar de resultaten van de analyse naar statistische informatie (zie H.4.1.2 Bepalen tijdsplanning) heb ik een idee gekregen welke data de grafiek nodig had. Door een aantal data queries te schetsen heb ik mijn idee vastgelegd op papier, zie bijlage<sup>vi</sup> Data queries. Hierdoor ontstond de vraag naar de volgende gegevens:

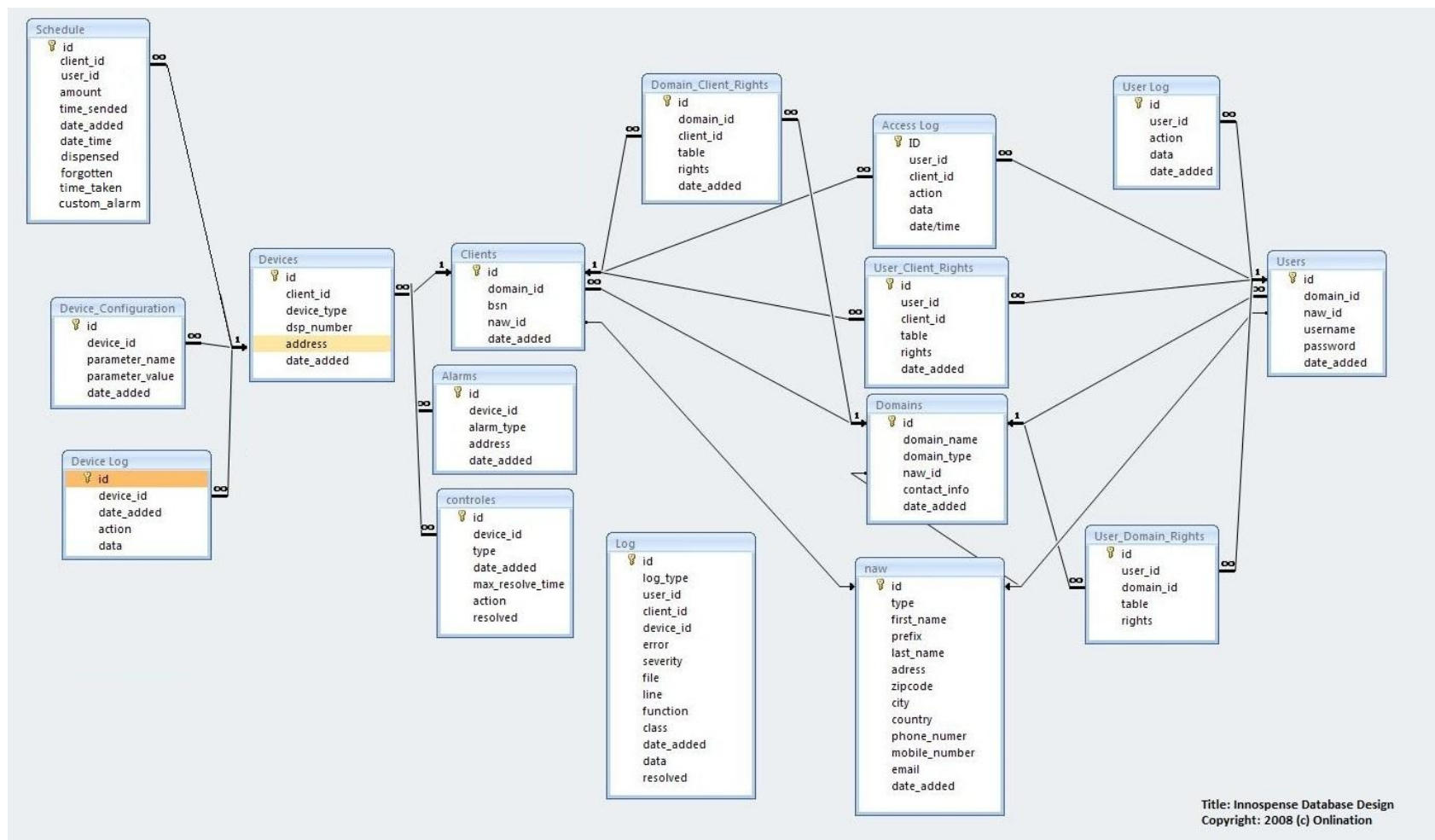
- Om welke patiënt gaat het?
- Wat is de datum waarop elke uitgifte is gepland
- Wat is de datum waarop elke uitgifte is uitgenomen
- Wat is de tijd waarop elke uitgifte is gepland
- Wat is de tijd waarop elke uitgifte is uitgenomen
- Een waarde of een uitgifte is vergeten, op tijd, te laat of te vroeg is uitgenomen
- Het limiet<sup>12</sup> van een op tijd uitgenomen medicatie

Met gebruik van Figuur 14 Innospense databasemodel is gekeken in welk domein deze gegevens zich bevonden. Door nauwe communicatie met de ontwikkelaar van het hieronder weergegeven databasemodel kon na worden gegaan welke velden welke informatie vasthielden.

---

<sup>11</sup> [http://en.wikipedia.org/wiki/Domain\\_model](http://en.wikipedia.org/wiki/Domain_model)

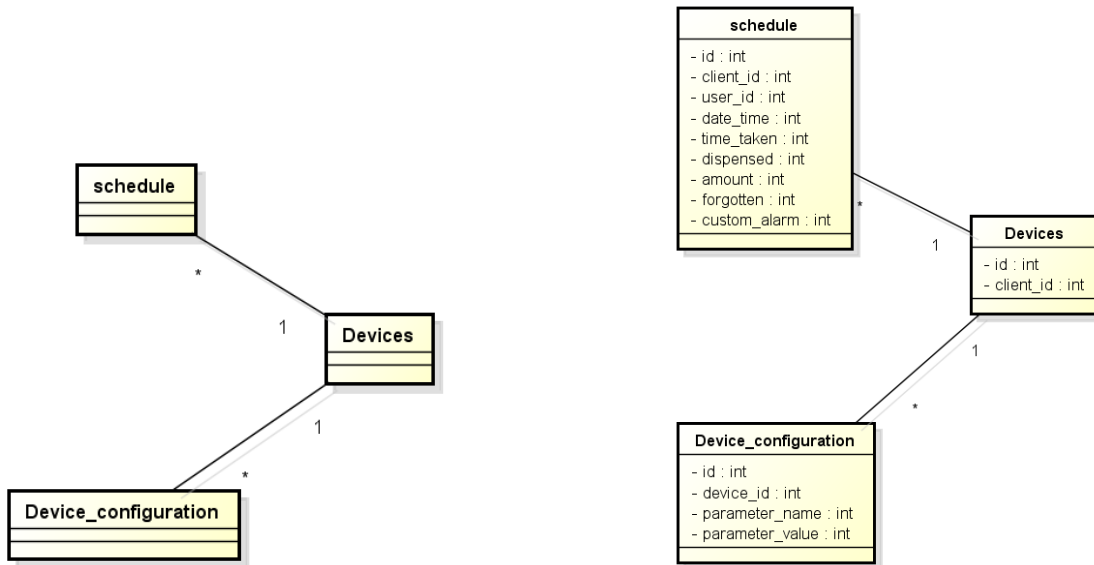
<sup>12</sup> Het limiet is het x aantal minuten dat medicatie nog als op tijd wordt geregistreerd bij het uitnemen van medicatie na het alarm. Dit kan per patiënt verschillen.



Figuur 14 - Bestaande Innosense databasemodel

### Opstellen domeinmodel

Uit het hierboven weergegeven databasemodel is de gewenste data verspreid over verschillende databasetabellen. Om in kaart te brengen welke domeinen nodig zijn uit het database heb ik een domeinmodel opgesteld(links), dit domeinmodel heb ik daarna gevuld met de originele database velden die ik nodig dacht te hebben voor mijn klassendiagram:



Figuur 15 - Domein model zonder velden(links) en met velden (rechts)

In Figuur 15 is onder worden velden uit de schedule tabel voornamelijk gebruikt om de grafiek te voorzien van uitgifte data. De tabel device wordt gebruikt voor de identificatie van de patiënt en de bijbehorende Medido-dispenser. De tabel device\_configuration wordt gebruikt om specifieke configuratie parameters van de Medido te verkrijgen. Voor een uitgebreide toelichting van de parameters zie het Elaboration rapport in bijlage<sup>v</sup>.

Dit model is voornamelijk gemaakt om de externe programmeur, Thomas Dijks, van informatie te voorzien zodat hij weet welke data ik nodig denk te hebben. Thomas Dijks zal naast het verzamelen en beschikbaar stellen van de hierboven gewenste data, de authenticatie van de gebruiker voor zijn rekening neemt in verband met de veiligheidsintegriteit.

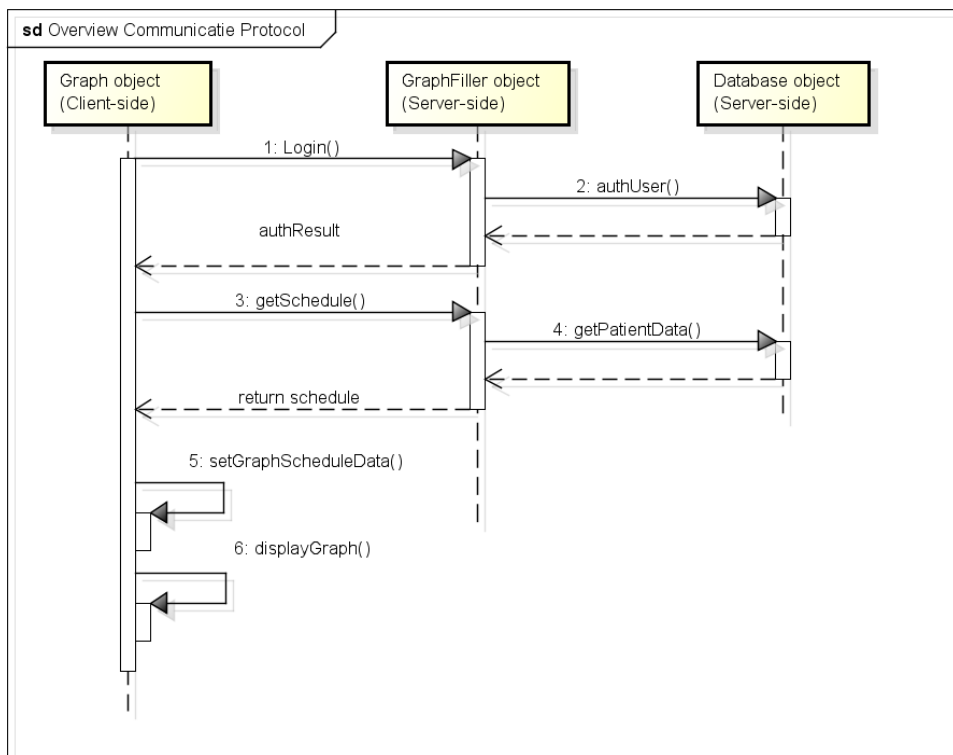
## 6.2.2 De communicatie structuur

Door het domeinmodel is vast gelegd welke data nodig is voor het maken van een grafiek. Het is echter nog niet bekend hoe de communicatie verloopt voor het doorgeven van deze data tussen onderlinge systeem delen. Het doel van het in kaart brengen van de communicatie structuur was om een overzicht te creëren die weergeeft hoe welke systeemdelen met elkaar communiceren en wat voor/welke data hierbij wordt uitgewisseld.

In deze subparagraaf wordt beschreven hoe deze communicatie in kaart is gebracht en wordt de globale-, server-database- en client-server-communicatie in detail besproken. Voor het in kaart brengen van de communicatie wordt gebruik gemaakt van UML sequentie diagrammen. Een sequentie diagram<sup>13</sup> geeft de interacties weer tussen verschillende objecten die een bepaalde functionaliteit (of een deel ervan) implementeren.

### **Globaal communicatie overzicht**

Om een duidelijk beeld van de gehele structuur te behouden is er een globaal communicatie overzicht gemaakt. Hieronder is een sequentie diagram weergegeven die een overzicht geeft van de globale communicatie.



Figuur 16 - Sequentie diagram Overview Communicatie Protocol

<sup>13</sup> [http://en.wikipedia.org/wiki/Sequence\\_diagram](http://en.wikipedia.org/wiki/Sequence_diagram)

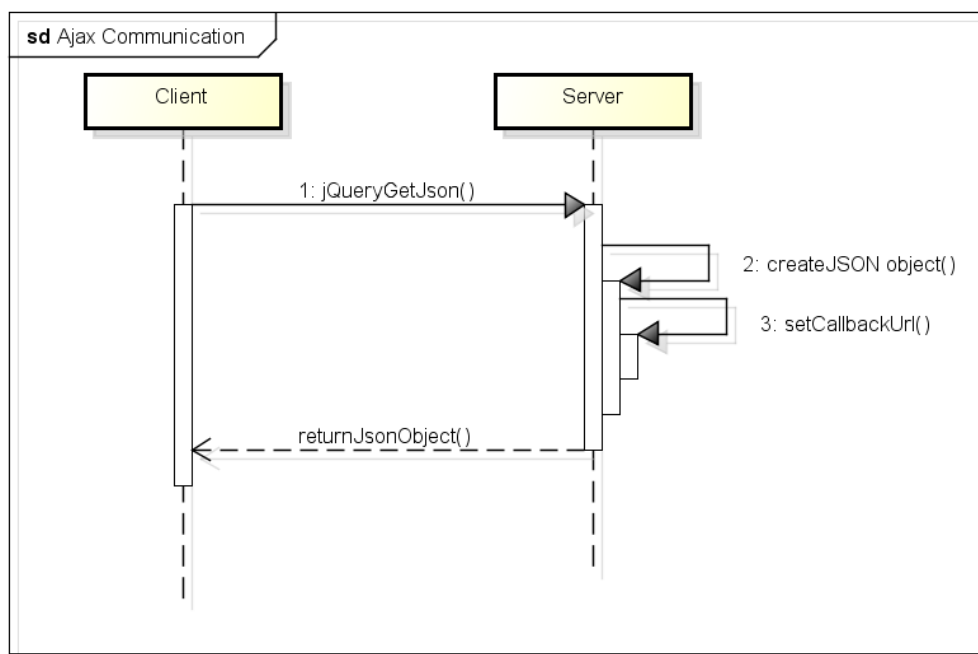
In het sequentie diagram van Figuur 16 is te zien dat een gebruiker zich eerst dient te authenticeren wat wordt gecheckt met het database. Om een grafiek weer te kunnen geven aan de kant van de gebruiker zal er eerst statistieken data nodig zijn, ook deze wordt op gehaald uit het database. Vervolgens zal de browser van de gebruiker de data in de grafiek zetten en deze weergeven aan de gebruiker.

### ***Server-database communicatie***

Zoals eerder beschreven heeft Thomas Dijks in verband met veiligheidsintegriteit er voor gekozen om de authenticatie en het ophalen van statistieken data uit het database op zich te nemen. Intern hebben ik en Dhr. Dijks afgesproken dat hij de data, zoals aangegeven in het door mij opgestelde domein model, beschikbaar stelt in de vorm van een PHP array. Verdere architectuur/structuur rondom dit gebied is daarom onbekend.

### ***Client-server communicatie***

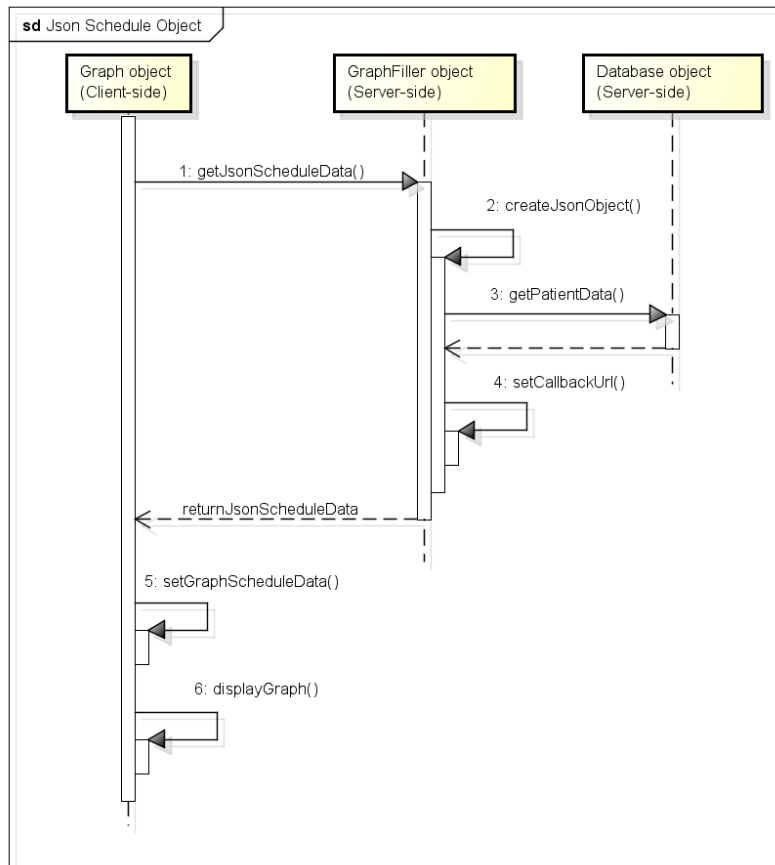
De actie `getSchedule()` uit het globaal communicatie sequentiediagram is een van de belangrijkste communicaties binnen dit project. Hierbij zal namelijk de server (die gebruik maakt van PHP) data versturen naar de client (die gebruik maakt van Javascript). Voor het uitwisselen van datastructuren tussen client en server wordt gebruik gemaakt van JSON<sup>14</sup>(JavaScriptObjectNotation), een deelverzameling van de Javascript programmeertaal. Bij het gebruik van AJAX voor het asynchroon ophalen van data wordt er om een JSON object gevraagd. Deze communicatie loopt via het https protocol en kan in een sequentiediagram als volgt worden aangegeven:



**Figuur 17 - Sequentie diagram Ajax Communication**

<sup>14</sup>Voor een uitgebreidere uitleg over de werking en structuur van JSON zie, <http://en.wikipedia.org/wiki/JSON>

De client zal met gebruik van de jQuery library de asynchrone GetJson() call doen richting de server. De server maakt in het sequentiediagram hierboven eerst een JSON object aan. Met de actie setCallbackUrl() wordt een adres in het JSON object aangemaakt zodat de server gemakkelijk het beschikbare webadres kan wijzigen voor het ophalen van een JSON object. Door het globale communicatie sequentie diagram te combineren met het hierboven weergegeven sequentie diagram kan het sequentie diagram zoals in Figuur 18 weergegeven worden.



**Figuur 18 - Sequentie diagram Json Schedule Object**

De functie om data asynchroon op te halen wordt niet alleen de eerste keer uitgevoerd als een gebruiker gegevens wil inzien maar zit ook in de Use case 'zoeken op datum verwerkt'. Omdat er niet telkens opnieuw hoeft worden ingelogd zal de authenticatie binnen het sequentiediagram niet worden getekend.



### ***Communicatie binnen de client***

Als de schemadata eenmaal binnen is aan de client-side zijn er nog een aantal functies die uitgevoerd kunnen worden op de grafiek(zie H.5.1Vaststellen eisen).

- Schakelen tussen zichtbare uitgiftelijnen
- Uitgiften sorteren op chronologische volgorde
- Inzoomen op een specifieke tijd van de dag.

Omdat voor het uitvoeren van deze functies geen aanvullende informatie van de server nodig is zal er ook geen communicatie met de server zijn. De sequentie van deze functies zijn opgenomen in het elaboration rapport, zie bijlage<sup>v</sup>.

### 6.2.3 Ontwerpen klassendiagram

De klassen in een klassendiagram vertegenwoordigen zowel de belangrijkste objecten met hun attributen, operaties en relaties als de objecten die vertaald worden naar programmeerbare code. Nu de communicatie van gegevens tussen verschillende systeemonderdelen in kaart is gebracht kan er een klassendiagram worden ontworpen. In deze subparagraaf wordt besproken hoe de objecten, relaties en attributen tot stand zijn gekomen.

#### ***Het creëren van objecten en relaties***

Belangrijk tijdens het maken van het ontwerp is dat er onderscheid gemaakt wordt tussen de client en de server. Grafische functies gerelateerd aan de grafiek zullen in Javascript worden geschreven voor de client-side en functies voor het beschikbaar stellen van de data wordt geschreven in PHP voor de server-side.

Door de schetsen van de grafieken te bestuderen heb ik de grafiek opgedeeld in objecten, hieruit volgde:

- De webpagina bevat een grafiek object
- Een grafiek bevat een of meer schema object(en)
- Een schema bevat een of meer uitgiftelij object(en)
- Een uitgiftelij bevat een of meer uitgifte object(en)



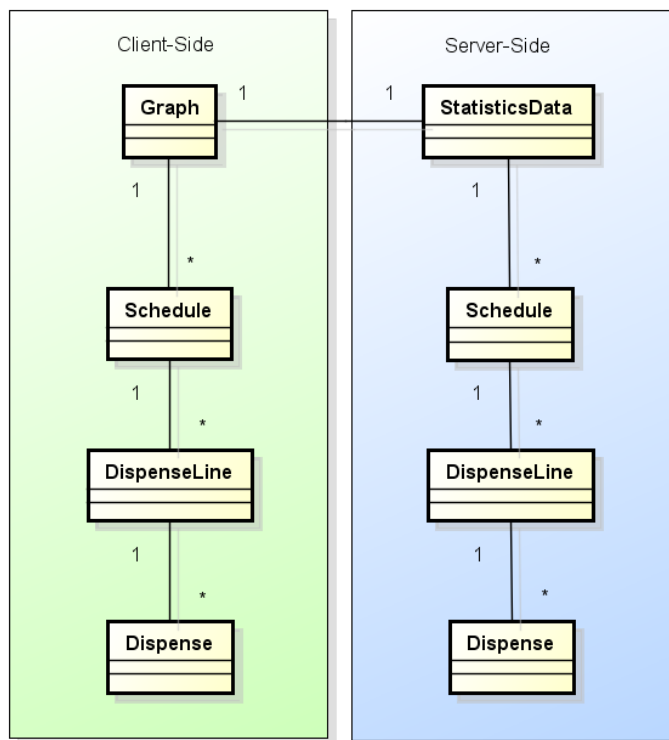
Figuur 19 - Concept van grafiek opgedeeld in objecten

Uit het figuur hierboven is op te maken dat een webpagina een grafiek bevat, de grafiek kan verschillende schema's tekenen, het schema kan worden onderverdeeld in een of meer uitgifte lijnen en per uitgifte lijn kunnen er een of meerdere uitgifte zijn.

Ik heb ervoor gekozen om een schema onder te verdelen in een uitgiftelijijn en uitgiften. Dit heb ik gedaan met oog op de functies/acties die zijn terug te zien in de vorm van Use cases zoals bijvoorbeeld, 'zoom in op tijd van de dag'. Vanwege die functie is er voor gekozen om het object uitgiftelijijn toe te voegen om zo makkelijk te kunnen schakelen tussen de zichtbaarheid van een complete uitgiftelijijn.

Door te definiëren dat er een of meer instanties van een object zijn is er ook gelijk een relatie tussen de objecten gelegd. Omdat het grafiek object grafisch en interactief is zal deze alleen voorkomen aan de client-side. De objecten schema, uitgiftelijijnen en uitgiften betreffen data en zullen ook voorkomen aan de server-side, de data is immers afkomstig van de server. Er is met opzet gekozen voor een structuur die gelijk is aan zowel de client- als server-side. Hierdoor ontstaat er in de complete systeem architectuur een uniforme structuur.

Door dit uit te werken in een concept versie van een klassendiagram heb ik een basis gecreëerd die er als volgt uit zag:



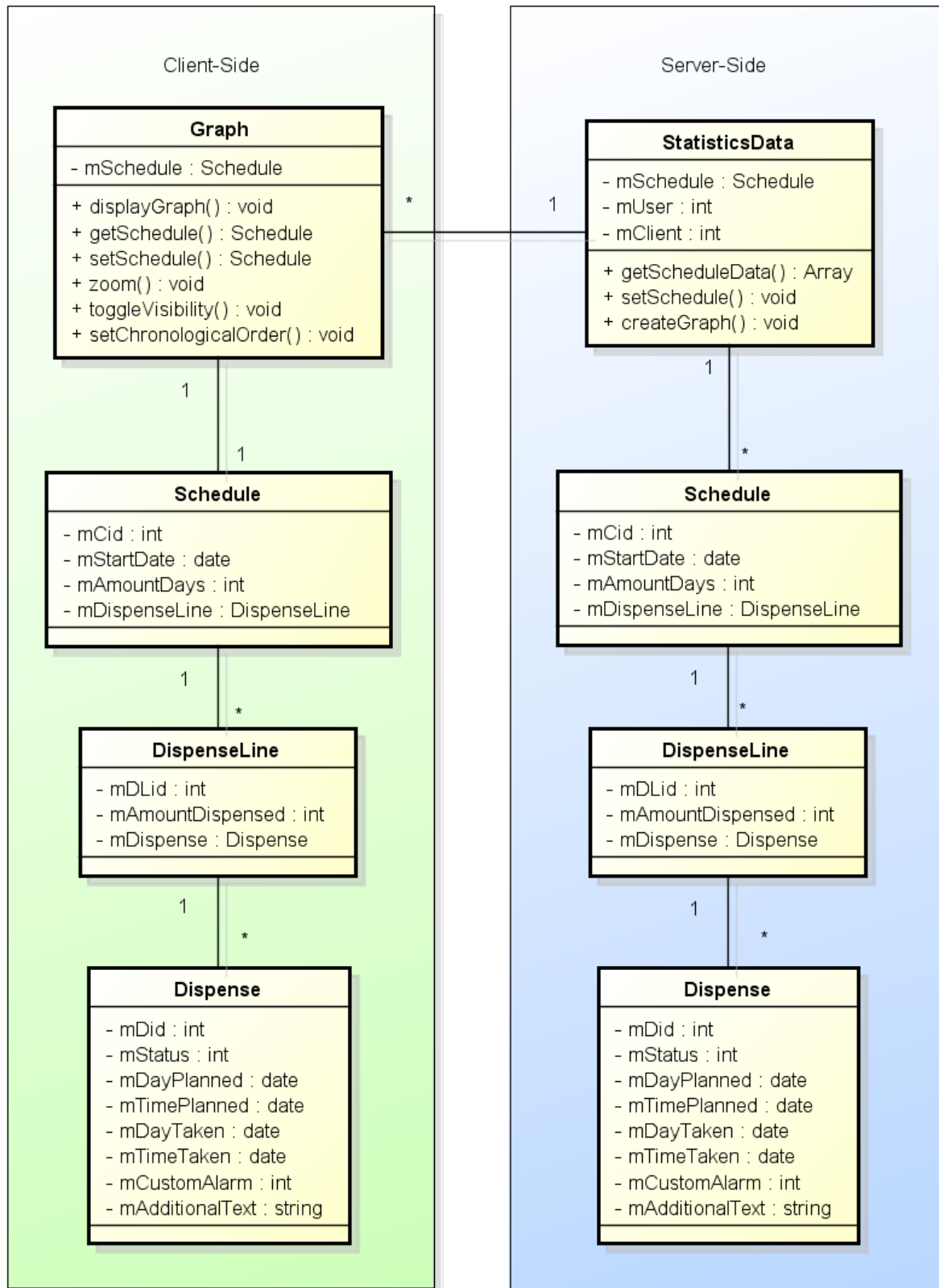
Figuur 20 - Klassendiagram opzet/skelet

De reden dat dit concept zoveel afwijkt van het domein model is omdat het schema object is opgedeeld in subobjecten uitgiftelijns en uitgifte. Tevens omdat Thomas Dijks het authenticatie- en data uitleesproces op zich neemt. De klasse statisticsData is verantwoordelijk voor het initialiseren van het grafiek object en het ontvangen van de data die Thomas Dijks uit het database zal halen.

### ***Attributen en Operaties***

Voor het invullen van de operaties en klassen is gekeken naar de data die door Thomas Dijks wordt opgehaald uit het database. Voor het omzetten van de databaserecords naar attributen hebben er een aantal wijzigingen plaatsgevonden omdat sommige database velden bijvoorbeeld een dubbele waarde of een onduidelijke benaming hadden. Een goed voorbeeld hiervan is het veld 'date\_time' die de datum EN tijd waarde bevat van wanneer een uitgifte gepland is. De benaming 'date\_time' is onduidelijk omdat er ook nog een dag en tijd is waarop een uitgifte daadwerkelijk is uitgenomen. Hierop heb ik ervoor gekozen om de waarde te scheiden en te veranderen naar, 'day\_planned' en 'time\_planned'. Welke velden waarnaar zijn veranderd is terug te zien in bijlage<sup>v</sup>, elaboration rapport(H.5.4). Voor het scheiden van zo'n veld zal waarschijnlijk een functie worden geschreven. Dit soort functies zijn niet opgenomen in het klassendiagram om te voorkomen dat het een te onoverzichtelijk diagram wordt. Er is daarom gekozen om alleen de attributen en operaties weer te geven die belangrijk zijn voor de structuur. In Figuur 21 - Het uiteindelijke klassendiagram op de volgende pagina is het huidige klassendiagram weergegeven.

Voor een volledige beschrijving van het ontwerp zie bijlage<sup>v</sup>, elaboration rapport.



Figuur 21 - Het uiteindelijke klassendiagram

## 6.3 Ontwikkelen

Het primaire doel van de Construction fase is het bouwen van het softwaresysteem. In deze fase ligt de nadruk op de ontwikkeling van componenten en andere kenmerken van het systeem. Dit is de fase waarin het daadwerkelijk programmeren van het grafisch statistieken weergeef systeem plaatsvindt. Het product wat normaliter bij deze fase hoort is het construction rapport waar de documentatie van de code in staat en het testplan in is verwerkt. Vanwege de vertraging en de uitgebreide online documentatie van JSXGraph is echter gekozen om de eventuele documentatie te integreren in de broncode en het testplan als een apart document op te stellen. Deze paragraaf is onderverdeeld in een aantal subparagrafen waarin de belangrijkste beslis en keuze momenten op een rij zijn gezet.

De eerste subparagraaf beschrijft welke technieken zijn gebruikt tijdens het ontwikkelproces.

De tweede subparagraaf behandelt een aantal knelpunten met de bijbehorende oplossing.

De derde subparagraaf beschrijft het proces waarin het testplan is opgesteld.

### 6.3.1 Ontwikkelomgeving en technieken

Om het ontwikkelen mogelijk te maken is er een gepaste en beschikbare ontwikkelomgeving nodig. Tijdens het ontwikkelen worden er verschillende technieken gebruikt om te komen tot het ontwerp zoals die staan beschreven in het elaboration rapport. In deze subparagraaf wordt beschreven hoe deze ontwikkelomgeving is opgezet en gemotiveerd waarom is gekozen voor het gebruik van specifieke technieken.

#### ***Opzetten ontwikkel omgeving***

Het webportaal bestaat uit een client-server architectuur. Voor het programmeren van server-side scripting is een web server nodig. Omdat er op het moment van ontwikkelen werd gewerkt aan het operationeel maken van het nieuw webportaal interface was de gebruikelijke ontwikkel omgeving in gebruik. Om niet tot last en/of afhankelijk te zijn voor/van de andere programmeur heb ik er voor gekozen om een lokale web server op te zetten als ontwikkel omgeving.

Als web ontwikkelomgeving heb ik gekozen voor het WAMP-server<sup>15</sup> pakket dat bestaat uit de applicaties: Apache2, PHP, MySQL. Dit was op dat moment de meest voor de hand liggende keuze omdat vrijwel geen instellingen nodig zijn om direct een lokale web server draaiend te hebben. Om te programmeren is de IDE<sup>16</sup> Eclipse in eerste instantie gebruikt echter werd geadviseerd door Dhr. Dijks om Netbeans te gebruiken als IDE omdat hij deze zelf ook gebruikt en het dan makkelijker zou zijn om programma code met elkaar uit te wisselen of om mij te ondersteunen. Daarom is gekozen om Netbeans 7.1 als IDE te gebruiken.

---

<sup>15</sup> <http://nl.wikipedia.org/wiki/WAMP>

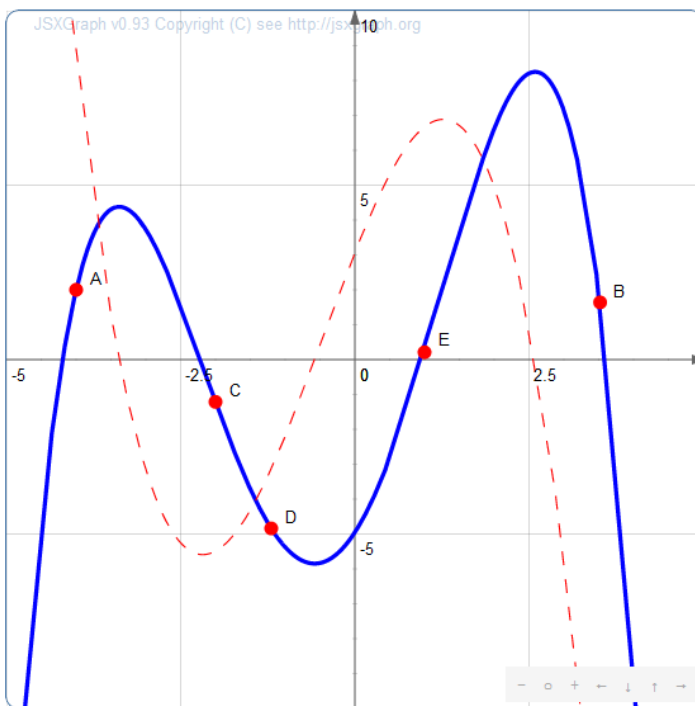
<sup>16</sup> [http://nl.wikipedia.org/wiki/Integrated\\_development\\_environment](http://nl.wikipedia.org/wiki/Integrated_development_environment)

### Gebruikte technieken

In de eerste iteratie is door middel van een klein experiment besloten (zie H.5.3.2) om voor het maken van de grafiek gebruik te maken van Javascript in combinatie met de grafische library JSXGraph. De voornaamste reden dat JSXGraph is gekozen is vanwege de compatibiliteit met verschillende soorten browser. Dit heeft te maken met de verschillende tekentechnieken die zitten verwerkt in de JSXGraph library. Naast SVG<sup>17</sup> wordt namelijk ook VML<sup>18</sup> en Canvas<sup>19</sup> gebruikt om data te visualiseren. In Figuur 22 is een voorbeeld grafiek te zien die met gebruik van JSXGraph is getekend.

Voor het verkrijgen van data zonder de volledige webpagina te verversen wordt gebruik gemaakt van de Javascript library jQuery. Door bijvoorbeeld de functie `jQuery.getJSON()` te gebruiken kan er een JSON object geladen worden door gebruik van een GET http request te maken. Er is gekozen om deze techniek te gebruiken omdat er een gestructureerd object geleverd wordt. Dit object is beschikbaar in de vorm van een array met (meerdere)subarrays.

Naast de programmeertaal Javascript is nog steeds een server-side scripting language, zoals PHP, nodig om de client-side van data te voorzien. Het standaard framework van PHP biedt de mogelijkheid om JSON objecten te creëren en beschikbaar te stellen. Dit sloot dan ook goed aan bij de jQuery functie `getJSON`.



Figuur 22 - Een voorbeeld grafiek, bron: <http://goo.gl/xmZLo>

<sup>17</sup> [http://nl.wikipedia.org/wiki/Scalable\\_Vector\\_Graphics](http://nl.wikipedia.org/wiki/Scalable_Vector_Graphics)

<sup>18</sup> Vector markup language

<sup>19</sup> [http://en.wikipedia.org/wiki/Canvas\\_element](http://en.wikipedia.org/wiki/Canvas_element)





### 6.3.2 Ontwikkel knelpunten en oplossingen

Tijdens het ontwikkelen van het grafisch statistieken weergeef systeem met PHP en Javascript kwam ik af en toe knelpunten tegen. Er worden een aantal voorbeelden gegeven die laten zien wat motivaties zijn bij specifieke ontwikkel keuzen. Bij wat complexere functies is documentatie aan de code toegevoegd om de gebruikte functie te voorzien van uitleg en motivatie.

#### ***Creëren van een JSON object met PHP***

Voor asynchrone communicatie tussen client en server is gekozen om de jQuery functie `getJSON()` te gebruiken. Hierbij wordt er een http get request gedaan naar een webadres die een JSON object aanbiedt. De structuur van een JSON object kan gezien worden als een array met subarrays. Omdat de benodigde data nog niet direct beschikbaar was gesteld door de ontwikkelaar ben ik met eigen test data gaan werken. De test data bestond echter uit een array met vijf waardes. De array versturen van server naar client verliep zonder problemen evenals het uitlezen. Echter de client was ontworpen voor het ontvangen van een array die maar vijf waardes had. Er was geen rekening mee gehouden dat de array die de schema data zou bevatten een flink complexere structuur zou hebben dan vijf losse waardes. Als oplossing heb ik een voorbeeld JSON object met de hand geschreven om zo te visualiseren hoe het JSON object er uit zou komen te zien. Het resultaat van het visualiseren van de structuur liet mij beter nadenken over de loops om data in of uit een JSON object te lezen.

```
{ "Schema": {
  "user_id": "1",
  "client_id": "1",
  "start_date": "20/01/2012",
  "amount_days": "7",
  "total_dispenses": "4",
  "total_on_time": "4",
  "total_too_late": "0",
  "total_too_early": "0",
  "total_forgotten": "0",
  "total_error": "0",
  "dispense_lines": {
    "dispense_line": {
      "time": "12:00",
      "amount_dispenses": "2",
      "line_on_time": "2",
      "line_too_late": "0",
      "line_too_early": "0",
      "line_forgotten": "0",
      "line_error": "0",
      "dispense_moments": {
        "dispense": [
          {
            "id": "001",
            "day_taken": "20/01/2012",
            "time_taken": "12:07",
            "status": "on_time",
            "additional_text": "optijd",
            "custom_alarm": "false"
          },
          {
            "id": "002",
            "day_taken": "21/01/2012",
            "time_taken": "12:10",
            "status": "on_time",
            "additional_text": "optijd",
            "custom_alarm": "false"
          }
        ]
      }
    }
  }
}
```

## Met JSXGraph de x- en y-as bepalen

JSXGraph is gebaseerd op een tekenbord/platvlak met x en y coördinaten.

```
var board = JXG.JSXGraph.initBoard('box', {boundingbox: [-2, 3, 2, -3]});
```

Het board is in principe oneindig groot, de boundingbox waardes geven aan welke gedeelte van de grafiek zichtbaar is. De cijfers die in de bounding box staan, geven de volgende grenzen aan: x negatief, y positief, x positief, y negatief. Om te tekenen op dit board worden eigenlijk alleen maar 'points' getekend, deze hebben een x en y waarde. Een lijn (dus ook de beide assen) wordt dus niet gedefinieerd in lengte maar wordt gedefinieerd door een lijn tussen twee 'points', beginpunt[x,y] en eindpunt[x,y].

Het doel was om het skelet van de grafiek na te bootsen zoals die is weergegeven in bijlage<sup>iv</sup>, grafieken. Dit hield in dat de dagen van de week op de x-as zouden komen te staan en het aantal uren van een dag op de y-as. Voor een grafiek van zeven dagen had ik ontwikkeld dat er een lijn werd getrokken tussen twee statische punten, een beginpunt (dag 1) en het eindpunt (dag 7). Bij het tekenen van de x-as werd echter helemaal geen rekening gehouden met de 'should have' requirement voor een maand weergave. Het probleem was dat als later het programma wordt ontwikkeld om een maandoverzicht te geven er een heel stuk nieuwe code geschreven zou moeten worden.

De oplossing en uitdaging was om de code zo te schrijven dat een maand overzicht niet direct implementeerbaar hoefde te zijn maar dat er rekening mee werd gehouden voor later. Dit wil zeggen door in plaats van vaste punten te definiëren werden de x en y coördinaten van begin- en eindpunt gedefinieerd door ze te koppelen aan het daadwerkelijk opgegeven aantal dagen. Hiermee werd niet alleen rekening gehouden dat er een maandoverzicht was maar ook dat die schaalbaar was. Een gedeelte van de functie zou dus goed gebruikt kunnen worden voor het in en uitzoomen op specifieke uitgifte momenten. Zo heb ik een functie geschreven die de datum en tijd in leest van de als eerst geplande uitgifte datum en de laatst geplande uitgifte datum die in de ontvangen data zit en op basis daarvan wordt de x- en y-as getekend.

```
function drawGridLines(){
  var amountOfDays = 7;
  var amountOfHours = 24;
  var xAxStart = board.create('point',[0.5,0], {name:'xStart', visible:false});
  var xAxEnd = board.create('point',[amountOfDays+0.5,0], {name:'xEnd', visible:true, fixed:true});
  var xAx = board.create('line',["xStart", "xEnd"],{straightFirst:false, straightLast:false});
  var yAxStart = board.create('point',[0.5,0], {name:'yStart', visible:false});
  var yAxEnd = board.create('point',[0.5,amountOfHours], {name:'yEnd', visible:true});
  var yAx = board.create('line',["yStart", "yEnd"], {straightFirst:false, straightLast:false});
  for(var i=1;i<=amountOfDays;i++){
    var d = new Date();
    d.setDate(d.getDate()-(amountOfDays-i));
    var monthOfYear = d.getMonth();
    var dayOfMonth = d.getDate();
    var dayOfWeek = d.getDay();
    var dayText = board.create('text',[i-1, mDayNames[dayOfWeek]], {fontSize:8});
    var dateText = board.create('text',[i-2, dayOfMonth+" "+mMonthNames[monthOfYear]], {fontSize:8});
  }
  for(var i=0;i<=amountOfHours;i++){
    var dayText = board.create('text',[0,i,"-00"], {fontSize:8});
  }
}
```

### 6.3.3 Opstellen testplan

Software wordt getest om de kwaliteit van een systeem te bepalen. Hoe uitgebreider de tests, hoe hoger de kwaliteit van het product is. Zo test je of er aan de requirements wordt voldaan maar ook of er nog fouten in software zitten (bugs of business logica). Een belangrijk aspect aan testen is het in een vroeg stadium kunnen opsporen van fouten. Want hoe vroeger je in het ontwikkeltraject een fout ontdekt, hoe goedkoper het is om het op te lossen. Daarnaast kan je aan de tests meten wat de kwaliteit is van het product. Het testen kan op verschillende manieren worden uitgevoerd.

Voor dit project stond gepland om te testen door middel van Unit tests. In deze Unit tests wordt gekeken of de output aan het verwachtingspatroon voldoet. Ik koos voor Unit testen omdat het makkelijk uit te voeren is, ook al is er weinig tot geen kennis van het systeem. Zo kan een buitenstaander een nieuwe functionaliteit inbouwen en de oude functionaliteiten testen door de unit tests uit te voeren. Slagen deze tests dan werken de oude functionaliteiten nog zoals het hoort. Daarnaast moet de applicatie voorbereid zijn op automatisch testen hiervoor en moet je hiervoor abstractie in je code gebruiken. Dit maakt in mijn optiek de code ook netter, begrijpelijker en overzichtelijker.

Vanwege de vertraging die is opgelopen tijdens de inception fase (zie H.4.3) bleef er maar een beperkte tijd over. Om toch niet helemaal met lege handen te komen heb ik ervoor gekozen om in ieder geval een testplan achter te laten voor de persoon die verder zal gaan met mijn project. Voor het opstellen van het testplan zijn alle requirements, Use cases en de belangrijkste interne code functies opgesomd. Door een beschrijving per requirement, Use case en functie te geven kan worden gecontroleerd waarop ze precies getest zijn en of ze al getest zijn. Zie bijlage<sup>vii</sup>, testplan

Door de drukte ben ik helaas niet meer toegekomen om het testplan uit te voeren. Ik ben echter wel van mening dat het testplan dusdanig is opgebouwd dat de volgende ontwikkelaar met dit zelfde testplan uit de voeten kan om dergelijke testen uit te voeren.

## 6.4 Resultaat tweede iteratie

Als resultaat uit de tweede iteratie is het elaboration rapport volledig afgerond. Tevens is het grafisch statistieken weergeefstelsel voor een gedeelte ontwikkeld.

### ***Afgerond elaboration rapport***

Het elaboration rapport bevatte vanuit de eerste iteratie al requirements met een prioriteit, een Use case diagram, Use case beschrijvingen. In dit rapport zijn in de tweede iteratie een aantal kleine wijzigingen in de requirements gemaakt en een aantal grote betreft de ontworpen architectuur. Zo is er een domein model gemaakt die de beschikbare data in kaart brengt. Met gebruik van UML diagrammen is de onderlinge communicatie tussen systeemdelen weergegeven en er is een klassen diagram ontworpen om vanuit te programmeren. Feedback vanuit de eerste iteratie is tevens verwerkt zo zijn er bijvoorbeeld in het ontwerp verantwoordelijkheden toegekend aan de delen van de architectuur.

### ***Grafisch statistieken weergeefstelsel***

Het oplossen van een vertraging en het wisselen naar een onbekende programmeertaal had meer invloed op de planning dan van te voren gedacht. Naast de opgelopen vertraging en het wisselen van de programmeertaal zijn er nog veel andere factoren die een rol hebben gespeeld tijdens de tweede iteratie. Dit is in de evaluatie terug te lezen. Tot aan het eind van het afstudeertraject wordt er door ontwikkeld aan het grafisch statistieken weergeefstelsel.

## 7 Evaluatie

In dit hoofdstuk wordt het hele proces dat ik deze afstudeerperiode heb doorlopen geëvalueerd. Deze evaluatie is verdeeld over een procesevaluatie en een product evaluatie.

In de procesevaluatie bespreek ik onder andere hoe ik het hele proces van de uit te voeren afstudeeropdracht heb beleefd en welke moeilijkheden ik hierbij ben tegengekomen. Uiteraard geef ik hier ook aan wat goed ging, wat slecht en hoe het beter had gekund.

In de productevaluatie draait het vooral om het grafisch statistieken weergeef systeem. Hierin gaat het over hoe tevreden (of niet) ik ben met het resultaat. Ik bespreek hier wat beter had gemoeten, waar ik trots op ben en wat in de toekomst er nog bij kan komen. Hier worden ook de andere producten besproken die ik opgeleverd heb, zoals de RUP documenten en andere documenten die van belang waren.

## 7.1 Procesevaluatie

### 7.1.1 Planning evaluatie

Over de opzet van de eerste planning was ik in eerste instantie redelijk tevreden, er zaten veel werkzaamheden in verwerkt. Echter na de jaarwisseling viel mij een aantal dingen op:

- In week twee was ik een halve week ziek
- De kerstvakantie week was niet opgenomen in de planning
- Met het onderzoek had ik twee en een halve week vertraging opgelopen

Dit telde bij elkaar een vertraging van 4 weken. Dit zou betekenen dat als die van de originele planning zou worden afgetrokken er niet eens een tweede ontwikkel iteratie zou zijn.

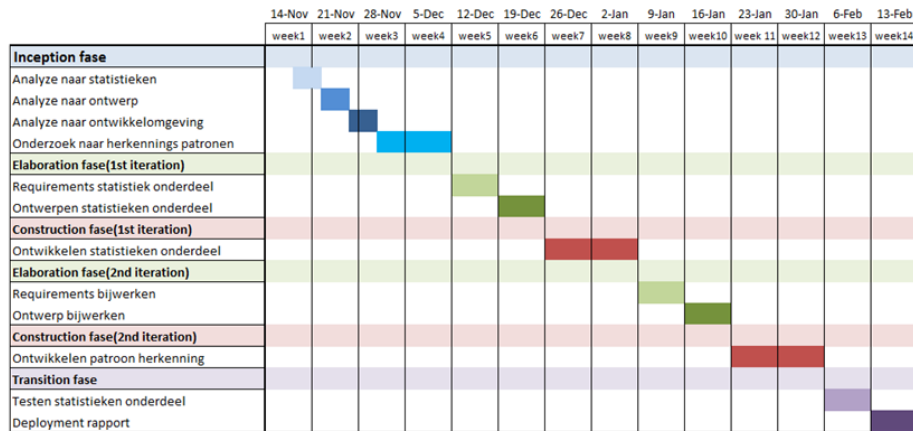
In de opnieuw opgestelde planning heb ik ondanks de flinke vertraging er een tweede iteratie in verwerkt. Dit omdat ik het belangrijk vond dat het iteratief werken in ieder geval zou worden nageleefd. Besloten was om het automatisch herkennen van patronen niet te ontwerpen en ontwikkelen. Met de resterende tijd was het echter nog wel mogelijk om statische patronen te ontwikkelen. Mijn voorkeur ging hier echter niet naar uit omdat ik het statisch herkennen van patronen niet zag als een uitdaging en het de tijd die het zou kosten niet waard vond. Maar ondanks mijn voorkeur liet ik de optie open om of het statische patroon herkenning te ontwerpen en ontwikkelen of het statistieken weergeef systeem nogmaals te doorlopen. Dit zou afhankelijk zijn van hoe vlot en soepel de eerste iteratie zou verlopen.

Bij de tweede keer aanpassen van de planning aan het eind van de eerste iteratie heb ik bewust gekozen om af te wijken van de planning. Met de bevindingen betreffende de inefficiëntie van PHP liep de eerste iteratie minder vlot dan gehoopt en vond ik het belangrijker de tweede iteratie te gebruiken voor de nodige ontwerp wijzigingen dan het ontwerpen en ontwikkelen van statische patroon herkenning functionaliteiten.

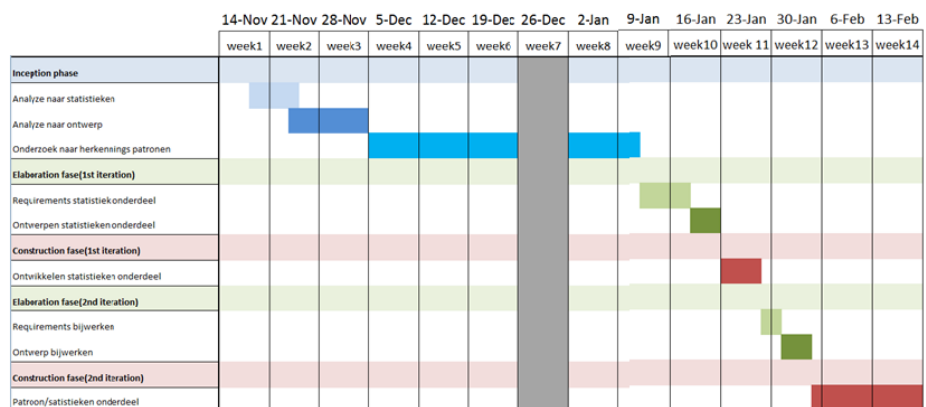
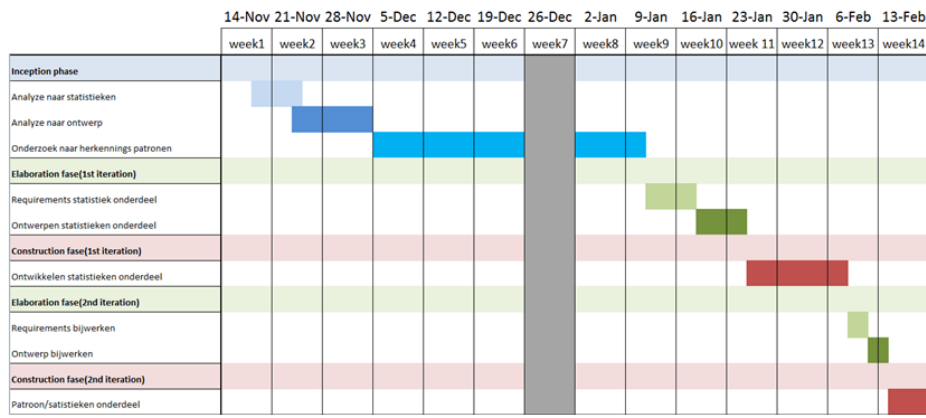
Achter af gezien had ik de vertraging wel in de gaten maar had ik het niet tijdig gevisualiseerd in een planning, hierdoor besepte ik niet wat de impact van die vertraging betekende. Tevens zat de opgelopen vertraging niet zozeer in een verkeerd opgestelde planning maar meer in de werkzaamheden zelf. In de toekomst zou ik de planning niet alleen binnenst hoofd bij moeten houden maar ook op papier. Waar ik erg tevreden mee was is de communicatie die ik had met zowel mijn bedrijfsbegeleider als de examinerator begeleider die ik direct op de hoogte had gesteld van de opgelopen vertraging.

In Figuur 23 is een overzicht te zien van de veranderingen in planningen die hebben plaatsgevonden.

Planning: 17/11/2011  
Innospense: ontwikkelen van een geautomatiseerd statistieken weergeefstelsysteem



Planning: 27/12/2011  
Innospense: ontwikkelen van een geautomatiseerd statistieken weergeefstelsysteem



Planning eind eerste iteratie

Figuur 23 - Alle planningen met elkaar vergeleken, Initiële planning(boven), planning einde inception fase(midden) en de planning aan het einde van de eerste iteratie(onder)



### 7.1.2 Interviews en onderzoek

Om achter de wensen en eisen van de opdrachtgever en de gebruikers van het Innospense webportaal te komen heb ik in de Inception fase interviews afgenomen. Deze interviews waren open gesprekken waar ik een paar vragen had. Wat ik vooral belangrijk vond was de verschillende meningen van de gebruikers in kaart brengen. Het leuke en tevens de uitdaging hieraan was om hiermee tot een oplossing te komen die voor iedereen geschikt is. Wat misschien minder goed was is dat ik een voorgekauwd ontwerp liet zien waar al erg goed door mij over nagedacht was. Hiermee werd de creativiteit van feedback of ideeën een beetje weggenomen. Echter denk ik dat het toch de goede keus is geweest omdat apothekers meer commercieel gericht zijn.

Het onderzoek naar patroon herkenning ging goed van start. Wat echter niet goed ging is dat ik geen duidelijk doel voor mijzelf had genomen om naar toe te werken. Ik had verwacht dat ik met alleen kennis opdoen wel een oplossing voor mijn ongedefinieerde probleem zou vinden. Omdat ik echter geen duidelijk doel voor ogen had verdwaalde ik in alle beschikbare informatie. Daar kwam bij dat ik het onderwerp patroon herkenning erg fascinerend en uitdagend vond vanwege zijn complexe karakter. Dit was ook een van de oorzaken dat ik niet goed besepte dat ik vertraging opliep. Met een op wat later moment ontdekte literatuur kwam ik er achter dat de complexiteit nog hoger lag dan ik dacht. Hierdoor besepte ik dat ik mij moest terug trekken uit de theorie en op zoek moest gaan naar bestaande software die voor mij patronen zou herkennen. Op dat moment was er echter al zoveel tijd verstreken dat ik daar niet meer aan toe ben gekomen. Daarnaast kon ik niemand vinden, ook extern niet, die mij kon helpen met het onderwerp patroon herkenning. Op dat moment heb ik er voor gekozen om niet alleen het onderzoek naar patroon herkenning te stoppen maar om ook het herkennen van patronen te schrappen van de planning.

Achteraf gezien, na onder anderen het 80% voortgang gesprek, is het misschien zonde geweest om niets met de stof en opgedane kennis te doen die is opgedaan tijdens het plegen van onderzoek. Wat ik wel als positief ervaar is dat ik juist gemotiveerd raak voor het vinden van een oplossing voor een complex onderwerp.

### 7.1.3 Ontwerp fase

Ik ben erg tevreden over hoe de ontwerp fase is gelopen ongeacht de grote wijziging in het ontwerp en de daardoor opgelopen vertraging. Door met het gebruik van theoretische modellen en technieken er achter te komen dat specifiek gekozen ontwerp keuzen niet geschikt zijn geeft het gevoel van kennis en verstandig gebruik. Tevens laat dit zien dat de theorie die op school wordt geleerd nuttig is voor het uitvoeren van een opdracht in de praktijk. Het ontwerp van het grafisch statistieken weergeef systeem springt er misschien niet boven uit vanwege zijn complexiteit, maar ik ben trots als ik kijk naar de aanpak, hoe de theorie is gebruikt en hoe gedocumenteerd is.

Wat iets minder ging was de aandacht die ik bestede aan de communicatie met de externe programmeur. Zijn planning was erg strak, zo had hij alleen tijd op woensdag en vrijdag. Achteraf gezien ging ik hier soms misschien iets te laks mee om wat de workflow toch flink kan beïnvloeden. In de toekomst is het verstandig om direct actie te ondernemen als er iets mist binnen jou proces maar waar iemand anders verantwoordelijk voor is om te leveren.

### 7.1.4 Ontwikkel fase

Het initiële idee was om een grafisch statistieken weergeef systeem en een patroon herkenning functionaliteit te ontwikkelen en te implementeren in het Innospense webportaal. De patroon herkenning functionaliteit is echter geschrapt van de planning (zie H.7.1.2 en H.7.1.3.) en het grafisch statistieken weergeef systeem is nog niet zo ver ontwikkeld dat het gereed is om geïmplementeerd te worden.

Met terugblik op de ontwikkelfase denk ik er beter aan te doen om de gekozen programmeertaal voor het te ontwikkelen systeem niet te onderschatten wat betreft de tijd die geïnvesteerd moet worden om het onder de knie te krijgen. Mijn kennis op het gebied van PHP en Javascript was dusdanig lager dan verwacht dat er toch nog flink wat tijd in is gaan zitten om echt lekker op gang te komen in de ontwikkelfase.

Als er meer tijd was over geweest had ik misschien het grafisch statistieken weergeefstelsysteem kunnen testen en kunnen implementeren voor het einde van de afstudeerperiode van de opleiding. Helaas is de construction fase vrij laat begonnen en was er weinig tijd over. Ook in het aanpassen van dit afstudeerverslag heeft de laatste twee weken van het project heel veel tijd gezeten. Hierover heb ik met de opdrachtgever gesproken en aangegeven dat het verslag verbeterd moest worden en dat ik de construction fase even op pauze moest zetten. Zelf heb ik dan ook aangeboden om na de afstudeerperiode van 19 maart nog tot 19 april 2012 te werken om in ieder geval de construction fase naar behoren af te ronden. Op dit moment heb ik afgesproken dat na het inleveren van dit verslag verder gewerkt gaat worden aan de constructie en eventueel nog de implementatie ervan.

## 7.2 Productevaluatie

### 7.2.1 Plan van aanpak

Het Plan van aanpak is bedoeld om de projectgroep(ik) en de opdrachtgever op één lijn te krijgen. Hierin staan ook weer een hoop dezelfde onderwerpen als in het Afstudeerplan, alleen nu gericht op de opdrachtgever. De gekozen oplossing was ook in zijn optiek een goede oplossing en hij had eigenlijk alleen commentaar op wat fouten in het document of onduidelijk beschreven punten.

### 7.2.2 Onderzoeksrapport

In de Inception fase zijn analyses uitgevoerd en is onderzoek gedaan naar het probleem en de oplossing. De bevindingen zijn vervolgens vastgelegd in het onderzoeksrapport. Vooral de interviews met de apothekers en de opdrachtgever waren erg belangrijk en voegden veel toe aan het onderzoeksrapport.

Wel vind ik het soms lastig om objectief te blijven als ik zelf een uitgesproken mening over een onderwerp heb. Vanwege het abrupt stoppen met het onderzoek naar patroon herkenning en het schrappen van patroon herkenning in de planning is het onderzoeksrapport niet volledig afgerond. Desalniettemin ben ik evenals mijn bedrijfsbegeleider blij het met het onderzoeksrapport als product. Dit omdat er veel bruikbare informatie in is verwerkt en het product goed tot zijn recht is gekomen in de elaboration fase.

Bij het Inception Rapport zat ook de eerste detailplanning, welke zoals eerder gezegd niet al te realistisch was. De detailplanning die ik later gemaakt heb sloot beter aan bij het proces. Deze detailplanning is nu uiteraard ook meegenomen in het Inception Rapport. Wat ik alleen nog steeds niet goed vind hieraan is het gebrek aan overzicht dat je hebt als je mijn planning ziet. Als ik per week keek was het goed te doen, maar je kan niet in één oogopslag zien wat je de komende weken gaat doen. Wel heb ik alles nog wat ik wanneer heb gedaan bijgehouden in een Excel bestand.

### 7.2.3 Elaboration rapport

In het Elaboration Rapport werk je de bevindingen van de Inception fase verder uit en wordt er meer geconcentreerd op de uitvoer van de oplossing. In het elaboration rapport zijn onder andere de requirements en het ontwerp vastgelegd. Het meeste tijd heeft gezeten in het onderzoek naar een geschikte alternatief voor PHP en het opstellen van het klassendiagram, dit vond ik ook het meest interessant. Ik merkte hoe moeilijk het is om van te voren een goed klassendiagram te maken. Je weet wel welke klassen je nodig hebt, en wat die ongeveer moeten kunnen, maar om dit allemaal uit te schrijven blijft lastig (vooral the methodes die je nodig hebt). Pas als ik eenmaal aan het programmeren ben weet ik vaak beter waar ik mee bezig ben en wat ik nog nodig heb. Desondanks ben ik erg blij en trots op het elaboration rapport vanwege zijn volledigheid.

#### 7.2.4 Statistische grafiek

Uiteindelijk draaide het allemaal om het maken van het grafisch statistieken weergeef systeem. Op dit moment wordt er nog volop aan het statistieken weergeef systeem gewerkt. Daarom heb ik zoals eerder aangegeven (zie H.7.1.4), aan mijn bedrijfsleider voor gesteld om nog een maand langer door te werken indien nodig voor het ontwikkelen van de statistische grafiek. Desondanks ben ik erg blij dat ik zover ben gekomen met de opdracht zoals die nu is.

Als op of aanmerkingen voor in de toekomst denk ik dat Innospense goed op weg is in het uitbreiden van functionaliteiten in het webportaal. Misschien nog een aantal ideeën voor in de toekomst die interessant kunnen zijn als product of functie:

- Het grafisch statistieken onderdeel uitbreiden met tabellen met overige statistische gegevens
- De database uitbreiden met tabellen waarin statistieken zijn terug te vinden.
- Alsnog een (statisch)patroon herkenning functionaliteit voor het herkennen van simpele patronen.

## 8 Te demonstreren beroepstaken

In dit hoofdstuk wordt beschreven hoe de door mij gekozen beroepscompetenties behaald zijn. Deze beroepscompetenties zijn er om in voorkomende beroepssituaties op adequate, doelbewuste en gemotiveerde wijze proces- en resultaatgericht te handelen. Dat wil zeggen passende procedures te kiezen en toe te passen om de juiste resultaten te bereiken. De vier beroepscompetenties/taken die ik heb gekozen zijn:

- Uitvoeren analyse door definitie van requirements (1.4)
- Ontwerpen systeemdeel (3.2)
- Bouwen applicatie (3.3)
- ~~Beheren van applicatie (4.1)~~

Voor elke beroepstaak is een paragraaf opgesteld waarin de desbetreffende beroepstaak wordt beschreven hoe deze behaald zijn. De beroepstaak Beheren van applicatie is echter afgefallen omdat ik hier niet aan toe gekomen ben in verband met de opgelopen vertraging in de inception fase.

## 8.1 Uitvoeren analyse door definitie van requirements 1.4

Tijdens mijn afstudeertraject heb ik uitvoerig geïnventariseerd door onderzoek te doen op verschillende gebieden, raadplegen van literatuur, online literatuur en interviews gehouden. Met een uitvoerige analyse op de geïnventariseerde onderdelen heb ik al mijn bevindingen beschreven in een onderzoeksrapport en daarna omgezet naar tastbare requirements in het elaboration rapport, zie bijlage<sup>v</sup>. Hierbij is gebruik gemaakt van UML technieken als Use case diagrammen en Use case beschrijvingen. De MoSCoW methode is toegepast voor het toekennen van prioriteit om zo de requirements nog gemakkelijker te kunnen testen op hun haalbaarheid.

## 8.2 Ontwerpen systeemdeel 3.2

Bij het bewijzen van de competentie ontwerpen systeemdeel zou de nadruk in eerste instantie liggen op het ontwerpen van een patroon herkenning functionaliteit. Echter het geautomatiseerd herkennen van een patroon leek toch complexer te zijn dan tevoren gedacht. Desalniettemin is wel het systeemdeel tot in detail ontworpen betreffende het grafisch statistieken weergeef systeem. In de eerste iteratie heeft zelfs een enorme wijziging plaatsgevonden in het ontwerp van het systeem wat voor mij geen probleem was om daar een passend ontwerp voor te maken. Hierbij heb ik niet alleen laten zien dat ik een systeemdeel kan ontwerpen maar dat ik ook flexibel ben in het bedenken van een beter ontwerp. Verschillende technieken zijn toegepast zoals het maken van een communicatie overzicht met sequentiediagrammen en is er een klassendiagram ontworpen.

## 8.3 Bouwen applicatie 3.3

In eerste instantie was het de bedoeling dat het grafisch statistieken weergeef systeem in het webportaal zou worden gebouwd met PHP. Door het ontwerp goed te bestuderen is er achter gekomen dat PHP zeer inefficiënt bleek te zijn en is er gekozen om als alternatief met Javascript te programmeren. Naast Javascript zijn er externe bibliotheken gebruikt voor de grafische mogelijkheden en het Asynchroon versturen van data tussen de server en de client. Aan het testen van de applicatie ben ik helaas niet toegekomen echter heb ik wel de extra moeite genomen om een testplan op te stellen waarin requirements en Use cases zijn verwerkt. Eventuele toekomstige programmeurs zouden het systeem dan alsnog kunnen testen op de goede werking en de kwaliteit.

## 9 Geraadpleegde literatuur

### Literatuur

- Een boek over patroon herkenning op hoog wiskundig niveau  
Pattern Recognition van Sergios Theodoridis  
[goo.gl/WBZlf](http://goo.gl/WBZlf)

### Online literatuur

- Stack Overflow is een gratis vraag en antwoord websites voor ontwikkelaars voor verschillende ontwikkel omgevingen. Gerelateerde vragen wat betreft broncode kunnen daar gepost worden waar vaak binnen 5 werkdagen wel een antwoord op wordt gegeven.  
<http://stackoverflow.com/questions>
- De website van jQuery is waar de Javascript library op te vinden is om zonde de volledige webpagina te verversen (AJAX) data te versturen tussen client en server. Op de website is ook de online documentatie beschikbaar.  
<http://jquery.com/>
- De website van JSXGraph is waar de Javascript library op te vinden is om grafieken mee te plotten. Op de website is ook de online documentatie beschikbaar.  
<http://jsxgraph.uni-bayreuth.de/wp/>
- De grootste vrije internetencyclopedie van de wereld die meer over veel informatie beschikt wat betreft verschillende methoden en technieken binnen de informatica en helpt met verschillende steekwoorden het zoek bereik enorm te vergroten.  
<http://nl.wikipedia.org/wiki/Wikipedia>



## 10 BIJLAGEN

De bijlagen zijn meegeleverd als apart document en bevatten de volgende producten:

- 
- <sup>i</sup> Plan van aanpak V1.0
  - <sup>ii</sup> Onderzoeksrapport V1.0
  - <sup>iii</sup> Interviews met apothekers
  - <sup>iv</sup> Grafieken V3.1
  - <sup>v</sup> Elaboration rapport V1.0
  - <sup>vi</sup> Data queries
  - <sup>vii</sup> Testplan