

PTM

The Next Iteration

Het bouwen van een nieuwe en verbeterde Project Time Management

Collin Maessen

AFSTUDEERSCRIPTIE VOOR FONTYS HOGESCHOOL ICT

Gegevens student:

Naam : Maessen, C.J.H.M.
Studentnummer : 2055738
Opleiding : Hogere Informatica
Afstudeerperiode : 3 september 2001 t/m 19 mei 2008 (160 werkdagen)

Gegevens bedrijf:

Naam bedrijf : ISAAC
Afdeling : Software Solutions
Plaats : Eindhoven
Bedrijfsbegeleider : Scholten V. Applicatiearchitect
Bedrijfsbegeleider : Peters K. Applicatiearchitect

Gegevens docentbegeleider:

Naam : Linnartz, P.L.L.J.

Gegevens scriptie:

Titel afstudeerverslag : PTM The Next Iteration
Datum uitgifte afstudeerverslag : 12 juni 2008

Getekend voor gezien door bedrijfsbegeleider:

Datum:

De bedrijfsbegeleider,

Voorwoord

Deze afstudeerscriptie heb ik geschreven in het kader van mijn afstudeerstage als informatica student aan de Fontys Hogeschool Informatica te Eindhoven.

Tijdens deze afstudeerstage heb ik een afstudeerproject van 800 uur uitgevoerd over een periode van 160 dagen. Dit project is getiteld “*PTM The Next Iteration*” en bestaat uit twee onderdelen, namelijk Onderzoek en Software Ontwikkeling. Beide onderdelen zullen in deze scriptie uitvoerig behandeld worden.

Graag wil ik in dit voorwoord gebruik maken van de gelegenheid het bedrijf ISAAC Software Solutions te bedanken voor het beschikbaar stellen van mijn afstudeerproject. Hierbij bedank ik in het bijzonder mijn bedrijfsbegeleiders Valentijn Scholten en Koen Peters voor de projectbegeleiding en technische ondersteuning tijdens mijn afstudeerstage. Ook wil ik mijn overige collega's bij ISAAC bedanken voor hun tijd en het delen van hun ervaring en kennis tijdens de uitvoering van mijn project.

Ook wil ik graag mijn docentbegeleider Paul Linnartz van Fontys Hogescholen bedanken voor zijn begeleiding tijdens het afstudeerproject, zijn tips en opbouwende kritiek.

Als laatste, maar zeker niet in het minst, bedank ik mijn coach Jos Boonen en afstudeercoördinator Frans Allebeek van Fontys Hogeschool Informatica voor alle begeleiding en ondersteuning tijdens het zoeken naar een geschikte afstudeerbedrijf met bijbehorende opdracht.

Collin Maessen
Eindhoven, juni 2008

Inhoudsopgave

VOORWOORD.....	I
LIJST VAN FIGUREN.....	V
LIJST VAN TABELLEN	VI
SAMENVATTING	VII
ENGLISH SUMMARY	VIII
VERKLARENDE WOORDENLIJST	IX
1 INLEIDING.....	1
I AFSTUDEEROPDRACHT.....	2
2 BEDRIJFSPROFIEL	3
2.1 GESCHIEDENIS	3
2.2 BEDRIJFSSTRUCTUUR	3
2.3 PRODUCTEN	4
2.4 WERKWIJZE	5
2.4.1 <i>Agile</i>	5
2.4.2 <i>Three-tier model</i>	6
3 OPDRACHTOMSCHRIJVING.....	7
3.1 ACQUISITIE.....	7
3.2 BEGINSITUATIE	8
3.3 PROBLEEMSTELLING.....	8
3.4 DOELSTELLING	9
4 OPSTARTEN PROJECT	10
4.1 KENNISMAKING	10
4.2 PLAN VAN AANPAK.....	10
4.3 PROBLEMEN	10
4.4 RESULTAAT & EVALUATIE	11
II ONDERZOEK.....	12
5 ONDERZOEKSONTWERP.....	13
5.1 DOELSTELLING	13
5.2 ONDERZOEKSMODEL	14
5.3 VRAAGSTELLING	15
5.4 BEGRIPSBEPALING	16
6 ONDERZOEKSPLANNING	17
6.1 FASERING	17
6.2 BESCHRIJVINGEN FASERING	17
7 PROCESVERSLAG ONDERZOEK	19
7.1 AANPAK.....	19
7.2 RESULTATEN	19

7.2.1	<i>Interviews functionaliteiten</i>	19
7.2.2	<i>Onderzoeksaanpak plan</i>	20
7.2.3	<i>Onderzoek statistieken weergave</i>	20
7.2.4	<i>Onderzoeksrapport</i>	21
7.3	CONCLUSIES	22
7.4	EVALUATIE.....	22
III	SOFTWARE ONTWIKKELING.....	23
8	PLANNING SOFTWARE ONTWIKKELING.....	24
8.1	FASERING	24
8.1.1	<i>Ontwerpfase</i>	25
8.1.2	<i>Bouwfase 1</i>	25
8.1.3	<i>Bouwfase 2</i>	25
8.1.4	<i>Testfase</i>	25
9	ARCHITECTUUR & TECHNIEK.....	26
9.1	J2EE/JAVA TECHNIEKEN	26
9.1.1	<i>EJB 3.0</i>	26
9.2	WEB TECHNIEKEN	27
9.2.1	<i>Flex</i>	27
9.3	DATA OVERDRACHT	28
9.3.1	<i>ValueObjects</i>	28
9.3.2	<i>Flex datamanager en RPChandlers</i>	30
9.3.3	<i>Servlets</i>	30
9.3.4	<i>XMLHelper</i>	31
10	PROCESVERSLAG SOFTWARE ONTWIKKELING.....	34
10.1.1	<i>Definitiefase</i>	34
10.1.2	<i>Ontwerpfase</i>	34
10.1.3	<i>Bouwfase 1</i>	35
10.1.4	<i>Bouwfase 2</i>	38
10.1.5	<i>Testfase</i>	38
10.2	PROCES & VERSIEBEHEER	39
10.3	CONCLUSIES SOFTWARE ONTWIKKELING.....	39
IV	AFRONDING.....	40
11	CONCLUSIES EN AANBEVELINGEN	41
11.1	ONDERZOEK	41
11.2	SOFTWARE ONTWIKKELING	41
12	EVALUATIE.....	43
12.1	ZELFREFLECTIE	43
12.2	DOELEN	43
13	NAWOORD	44
14	LITERATUURLIJST.....	45
V	BIJLAGEN.....	46
	BIJLAGE 1: PLAN VAN AANPAK	47
	BIJLAGE 2: ONDERZOEKSAANPAK PLAN	70

BIJLAGE 3: ONDERZOEKSRAPPORT.....	81
BIJLAGE 4: DOCUMENT VAN EISEN	108

Lijst van figuren

Figuur 1	Organogram ISAAC Software Solutions	4
Figuur 2	Onderzoeksmodel Rapportage en Charting API's	14
Figuur 3	Fasen Iterative Application Development	24
Figuur 4	Screenshot van agenda PTM	28
Figuur 5	ValueObject UML diagram	29

Lijst van tabellen

Tabel 1	Onderzoeksfasering	17
---------	--------------------	----

Samenvatting

Deze afstudeerscriptie behandelt de wijze waarop de afstudeeropdracht die uitgevoerd is bij ISAAC Software Solutions is aangepakt. Belangrijke beslissingen en keuzes zijn beargumenteerd en verantwoord. De scriptie beschrijft dus de uitvoering van de opdracht, de methode van werken die toegepast is en de resultaten van de verschillende onderdelen. Ook worden de oorspronkelijke planning en de daadwerkelijke chronologische uitvoering van de opeenvolgende fasen tijdens het project beschreven.

Hoewel de scriptie een procesverslag is kon er op enkele plaatsen aandacht worden besteed aan het technisch karakter van de afstudeerstage. Dit is gedaan om meer inzicht te geven in de softwareontwikkeling werkzaamheden en opgeleverde software. Deze technische informatie is in samenspraak met mijn bedrijfsbegeleider toegevoegd aan dit verslag (zie hoofdstuk 9) en betreft enkele voorbeelden uit de opgeleverde software en bevat geen volledig ontwerp.

Tijdens het onderzoek deel van het afstudeerproject is er onderzoek verricht naar een geschikt rapportage en charting API die voldoet aan de wensen van ISAAC Software Solutions voor de nieuwe versie van Project Time Management (PTM).

Het onderdeel softwareontwikkeling in de scriptie omschrijft het proces dat doorlopen is tijdens de ontwikkeling van de software en de daarbij tegengekomen problemen. Aan het einde van dit proces is een vernieuwde PTM applicatie opgeleverd.

In het verslag wordt geconcludeerd dat er een geschikt rapportage en charting API is gevonden voor het genereren van rapportages en grafieken, te weten BIRT. De conclusies en aanbevelingen van het softwareontwikkeling onderdeel worden verwoord. Tot slot wordt er een evaluatie gegeven over de wijze van werken van de afstudeerder en de opgeleverde resultaten.

English Summary

This thesis documents the course of the graduation assignment at ISAAC Software Solutions. Important decisions and choices are motivated. This thesis describes the execution of the assignment, the work method and the results of the different components of the assignment. The original planning and the chronological realisation of the different phases of the project are reported.

Although the thesis is a process-report there was room for attention to the technical part of the assignment. This enables insight in the software development activities in the delivered software. This technical information was added to the report after conference with my student company-counsellor entails no complete design but several examples of the delivered software.

Research of a suitable reporting and charting API, which fulfilled ISAAC Software Solutions' demands of a well functioning Project Time Management (PTM), took place during the research part of the thesis.

The software development part of the thesis describes the entire process of developing the software and the problems that occurred. At the end of the process a new PTM application was delivered.

The thesis concludes a suitable reporting and charting API, BIRT, for generating reports and statistics has been found. The conclusions and recommendations of the software development part are described. And a evaluation of the actions of the student and the end results of the assignment form the conclusion of the thesis.

Verklarende woordenlijst

Enterprise JavaBeans (EJB)	Deze specificatie is één van de Java APT's in de J2EE-standaard. EJB's zijn bedoeld om in een meerlagenmodel de zogenaamde businesslogica van een applicatie te bevatten.
Entity Bean	Dit is een type Enterprise JavaBean die een in een database opgeslagen gegevens representeert.
Servlet	Dit is een in Java geschreven programma dat binnen een J2EE webcontainer op een server draait.

1 Inleiding

Deze afstudeerscriptie behandelt de wijze waarop de afstudeerstage bij ISAAC Software Solutions is aangepakt. De keuzen en belangrijke beslissingen die gemaakt zijn tijdens deze stage zijn zo goed mogelijk beargumenteerd en omschreven in deze scriptie.

De volledige afstudeerscriptie is ingedeeld in een vijftal basisonderdelen, die hieronder in het kort worden toegelicht:

I Afstudeerproject

In dit onderdeel van het afstudeerverslag wordt allereerst gekeken naar het bedrijfsprofiel van het afstudeerbedrijf ISAAC Software Solutions. Hierbij wordt onder andere aandacht besteed aan de organisatiestructuur binnen ISAAC Software Solutions en de diensten en producten die zij levert. Vervolgens wordt het afstudeerproject inhoudelijk behandeld, waarbij onder andere de doelstellingen van het afstudeerproject nader worden beschreven. Als laatste komen de opstart en het verloop van het project aan bod.

II Onderzoek

Dit onderdeel van de afstudeerscriptie behandelt het onderdeel onderzoek van het afstudeerproject. Het onderzoeksontwerp met onderzoeksmodel, en de hiervan afgeleide vraagstelling worden behandeld. Als laatste wordt het verloop van het onderzoek, de gehanteerde aanpak, fasering en uiteindelijk resultaat behandeld.

III Software Ontwikkeling

In dit onderdeel van de afstudeerscriptie komt de ontwikkeling van de nieuwe PTM applicatie aan bod, waaronder de planning voor het ontwikkelen van de software, de gebruikte architectuur en technieken en het procesverslag software ontwikkeling.

IV Afronding

Dit onderdeel van de afstudeerscriptie behandelt de afronding van de afstudeerstage. Ten eerste wordt de afronding van het project behandeld met daaropvolgend conclusies en aanbevelingen. Ten tweede wordt een evaluatie gegeven van de gehele afstudeerperiode.

V Bijlagen

Het laatste onderdeel van het afstudeerverslag bevat de bijlagen die genoemd zijn in de scriptie of die belangrijk zijn als naslagwerk.

Als bijlage treft u onder andere aan het plan van aanpak, het onderzoeksrapport en het document van eisen.

I

Afstudeeropdracht

In dit onderdeel van het afstudeerverslag wordt allereerst gekeken naar het bedrijfsprofiel van het afstudeerbedrijf ISAAC Software Solutions. Hierbij wordt onder andere aandacht besteed aan de organisatiestructuur binnen ISAAC Software Solutions en de diensten en producten die zij leveren. Vervolgens wordt het afstudeerproject inhoudelijk behandeld, waarbij onder andere de doelstellingen van het afstudeerproject nader worden beschreven. Als laatste komen de opstart en het verloop van het project aan bod.

2 Bedrijfsprofiel

In dit hoofdstuk beschrijf ik het bedrijf ISAAC Software Solutions, in wiens opdracht ik mijn afstudeerstage heb uitgevoerd. Deze korte beschrijving van de core-business en structuur van de organisatie geeft een helder beeld van de omgeving waarin het afstudeerproject is uitgevoerd.

2.1 Geschiedenis

ISAAC is een relatief jong bedrijf dat in 1999 is opgericht door drie afgestudeerden van de Technische Universiteit Eindhoven (TU/e). De naam ISAAC is een acroniem dat staat voor Internet Strategy And Automation Company.

Uit het oorspronkelijk bedrijf zijn in 2006 de twee zusterbedrijven ISAAC Software Solutions en ISAAC Web Solutions ontstaan.

Software Solutions richt zich op het leveren van maatwerk voor het automatiseren van bedrijfsprocessen. De applicaties die geleverd worden aan klanten zijn met name internet/intranet applicaties. Web Solutions bouwt voornamelijk websites en website gerelateerde producten waarbij vaak gebruik wordt gemaakt van de softwareoplossingen van ISAAC Software Solutions.



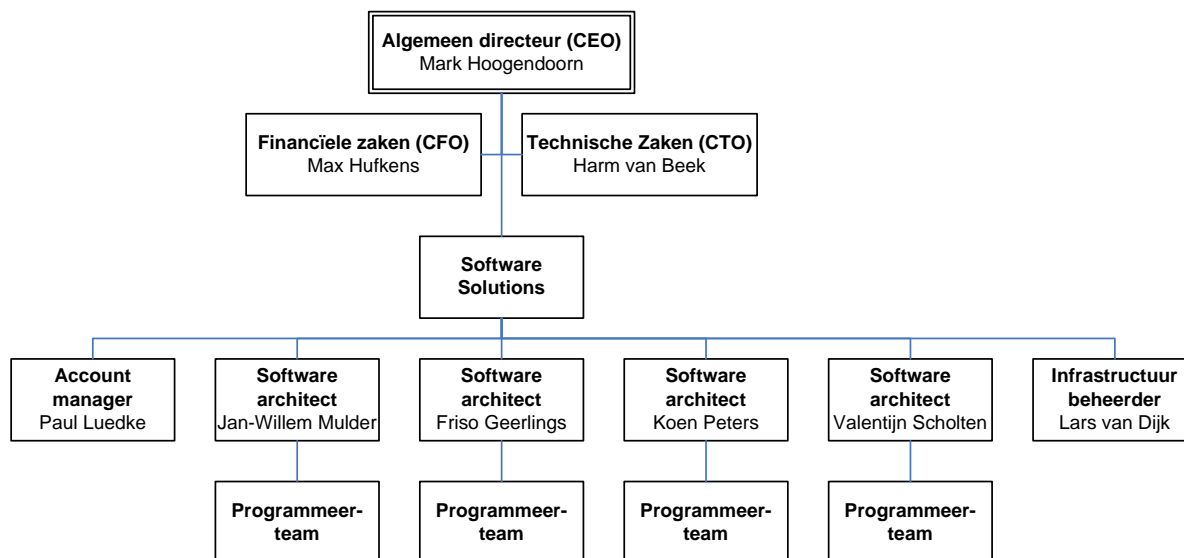
Uiteraard is er sprake van een nauwe samenwerking van de beide ISAAC bedrijven. Bovendien heeft ISAAC Web Solutions nauwe contacten met een aantal ontwerp- en reclamebureaus voor de grafische vormgeving van de nieuwe websites.

2.2 Bedrijfsstructuur

ISAAC beschikt momenteel over bijna dertig medewerkers en is, zoals eerder beschreven, onderverdeeld in twee zusterbedrijven, namelijk Software Solutions en Web Solutions. Het zusterbedrijf Software Solutions is het grootste en beschikt over tweederde van het totaal aantal medewerkers. Deze medewerkers zijn momenteel ingedeeld in vier programmeerteams die elk onder leiding staan van een software architect.

Het zusterbedrijf Web Solutions bestaat momenteel uit bijna tien medewerkers. Naast creative webdesigners beschikt Web Solutions, net zoals bij Software Solutions, over deskundige consultants.

Op de volgende pagina treft u een organogram aan van de bedrijfsstructuur van ISAAC Software Solutions (Figuur 1).



Figuur 1 Organogram ISAAC Software Solutions

De nog niet eerder genoemde accountmanager en infrastructuurbeheerder zijn zowel werkzaam voor Software Solutions als Web Solutions. Aangezien mijn afstudeerstage bij Software Solutions plaats vond, is de organisatiestructuur van Web Solutions niet in het organogram opgenomen en zijn de accountmanager en infrastructuur beheerder opgenomen in het organogram van ISAAC Software Solutions.

2.3 Producten

ISAAC Software Solutions heeft diverse standaardproducten ontwikkeld, waarvan ik hier enkele kort zal toelichten.

Adlinea - Online Publishing

Met het softwarepakket Adlinea kunnen klanten snel en gemakkelijk advertenties, folders, posters en ander drukwerk opmaken met behulp van in het systeem aanwezige sjablonen. Klanten kunnen zo gemakkelijk online in Adlinea gemaakte documenten naar hun drukker of uitgever sturen.

Pangaea / Enterprise Middleware¹

ISAAC biedt met Pangaea de mogelijkheid om bestaande 'legacy' backoffice systemen te laten voldoen aan de moderne informatiebehoefte met behulp van aansturing van interfaces. Deze interfaces kunnen verbinding maken met verschillende systemen en deze gegevens vervolgens aan andere applicaties voeden of presenteren op bijvoorbeeld een website of een internet connected desktop application.

Project Time Management

Project Time Management is een compleet urenregistratiesysteem waarin medewerkers snel en eenvoudig hun uren correct kunnen bijhouden. PTM kan ook worden gebruikt als hulpmiddel voor het bijhouden van een correcte administratie van de verlofuren en –dagen van medewerkers.

Content Management Systeem (CMS)

Met een Content Management Systeem (CMS) kunnen klanten zelf, eenvoudig en zonder technische kennis, hun website beheren en aanpassingen doorvoeren. ISAAC kan standaard oplossingen aanbieden voor het beheren van websites of een compleet pakket op maat maken.

¹ Middleware omvat systeemsoftware die de informatie-uitwisseling regelt tussen client-software en de software die de bedrijfsgegevens beheert.

Clientweb

Clientweb is een communicatieplatform waar klanten en hun afnemers toegang tot hebben. Hierdoor kan men onder andere zeer efficiënt informatie voor het beheren van orders intern of tussen verschillende klanten doorgeven.

Nieuwsbrief Generator

De door ISAAC ontwikkelde digitale nieuwsbrief software kan door klanten worden gebruikt om hun afnemers op de hoogte te houden van acties en evenementen. Door het bijhouden van gedetailleerde statistieken kan deze tevens gebruikt worden voor marketingdoeleinden.

Project Time Manager (PTM) is het product dat tijdens dit project herontworpen en ontwikkeld dient te worden, door onder meer het toevoegen van uitgebreide rapportage en charting mogelijkheden.

2.4 Werkwijze

In deze paragraaf wordt er gekeken naar de werkwijze binnen ISAAC Software Solutions. Een aantal onderdelen van de werkwijze worden hieronder kort toegelicht:

2.4.1 Agile

De Agile methode is een conceptueel framework dat binnen ISAAC wordt toegepast bij de ontwikkeling van applicaties. In tegenstelling tot traditionele methoden, zoals bijvoorbeeld de waterval methode waar men alle functionaliteit in een keer sequentieel implementeert, worden projecten bij de agile methode ingedeeld in zogenaamde sprints en wordt er iteratief gewerkt.

Een sprint neemt een periode van ongeveer 2 á 3 weken in beslag en wordt vooraf gegaan door een sprintplanning. Tijdens deze sprintplanning worden, eventueel in overleg met de opdrachtgever, de doelstellingen bepaald voor een aankomende sprint. De te volgen richting tijdens het ontwikkelingstraject wordt dus mede bepaald door de betreffende opdrachtgever. Tevens bekent het iteratief werken in de agile methode dat feedback en resultaten uit de huidige sprint, of ontwikkelingsfase, aanpassingen kunnen veroorzaken in de gebruikte documentatie en/of ontwikkelde applicatiecode.

Voor het maken van de sprintplanningen wordt het product backlog document gebruikt. Dit document bevat de gedetailleerde beschrijvingen van alle te implementeren requirements, wenslijsten, etc. Het omschrijft wat er gebouwd zal worden tijdens het project.

Tijdens de sprint zelf wordt een klein gedeelte van de uiteindelijk applicatie gerealiseerd. Door het in grotere mate betrekken van de opdrachtgever voldoet het uiteindelijk opgeleverde eindproduct in hogere mate voldoet aan de eisen van de opdrachtgever dan bij gebruik van traditionele ontwikkelingsmethoden.

Na afloop van een sprint vindt er een sprintreview plaats, waarbij het verloop van de sprint wordt geëvalueerd. Tijdens deze review wordt besproken wat goed en minder goed is verlopen en wordt gekeken welke verbeteringen mogelijk zijn.

2.4.2 Three-tier model

ISAAC Software Solutions gebruikt in al haar webapplicaties het three-tier model. Het three-tier model is een architectuur waarin de presentatielaag, logicalaag (business rules) en het opslaan en toegang tot data worden ontwikkeld en onderhouden als aparte modules.

Door het opzetten van de processen in verschillende lagen is het mogelijk om één van de lagen te vervangen of te upgraden als er nieuwe eisen gesteld worden of technologische veranderingen zijn.

Het three-tier model bevat de volgende tiers:

De presentatielaag

De presentatielaag bevat de grafische vormgeving van de applicatie, oftewel de user-interface. Deze wordt gescheiden van de andere elementen om eventuele veranderingen in de vormgeving onafhankelijk van de techniek door te kunnen voeren. Deze keuze bevordert de schaalbaarheid, beheersbaarheid en consistentie van zowel de vormgeving als de onderliggende data en logica lagen.

De data laag

De data laag bevat alle gegevens van de applicatie. Dit is meestal de inhoud van de site, maar ook gebruikersaccounts met hun toegangsrechten en zaken zoals bijvoorbeeld geplaatste orders en beschikbare magazijninhoud. Bij ISAAC wordt deze laag meestal geïmplementeerd met MS SQL 2005 database.

De logica laag

De logica laag koppelt de data laag aan de presentatielaag en zorgt er voor dat de inhoud op de juiste pagina getoond wordt. ISAAC implementeert de logicalaag met behulp van de programmeertaal Java.

3 Opdrachtomschrijving

In dit hoofdstuk worden de afstudeeropdracht en de afbakening van het project beschreven. Zo komen de aanleiding tot de opdracht, de beginsituatie en probleemstelling aan de orde. Ook worden de doelstelling, het beoogde projectresultaat en randvoorwaarden/risico's behandeld.

Deze informatie kan in meer detail ook in het Plan van Aanpak (bijlage I) gevonden worden.

3.1 *Acquisitie*

Ik was op zoek naar een bedrijf waar ik na de afstudeerstage de mogelijkheid zou hebben, mits de afstudeerstage naar tevredenheid is voltooid, een baan te krijgen. Daarnaast zocht ik ook een bedrijf dat past binnen mijn interesse in het ontwikkelen van webbased en intranetapplicaties.

Een vroege oriëntatie op een afstudeerplaats was voor mij van belang omdat ik door een urenbeperking (ik mag en kan maar 5 uur per dag werken) niet een 'normale' afstudeerperiode kan doorlopen, en dus een afstudeerbedrijf en project nodig had waar het mogelijk is om volgens deze uren te werken.

In het voorjaar ben ik naar de Fontys banenmarkt geweest. Tijdens de gesprekken met diverse recruiters op deze markt werd mij duidelijk dat er vaak mogelijkheden zijn om een bestaande afstudeeropdracht op een dergelijke manier uit te voeren of om een aangepaste afstudeeropdracht te maken.

Rond deze periode ben ik in het Stage en Afstuderen Informatie Systeem (SAIS) van Fontys gaan zoeken naar interessante bedrijven en/of afstudeeropdrachten. Helaas waren er niet veel interessant nieuwe opdrachten beschikbaar.

Uiteindelijk kreeg ik in juli 2007 op dezelfde dag van drie verschillende studenten, de tip om bij ISAAC Software Solutions te informeren naar een mogelijke afstudeerplaats. Diezelfde dag nog heb ik een e-mail gestuurd met de vraag of er een afstudeerplaats c.q. opdracht beschikbaar was.

Vrij kort daarna heb ik een kennismakingsgesprek gehad bij ISAAC Software Solutions waarin een mogelijke afstudeeropdracht voorgelegd werd: het uitbreiden van de huidige rapportage mogelijkheden in Project Time Management (PTM) en hiervan mooie en overzichtelijke rapportages te maken. Dit leek mij een interessante en uitdagende afstudeeropdracht die ik van harte heb aangenomen.

3.2 Beginsituatie

Project Time Management (PTM) is een webbased projectmanagement systeem. Voor ISAAC worden in PTM basisklantgegevens, projecten en projecturen van werknemers bijgehouden.

Door het bijhouden van deze gegevens in PTM kunnen diverse rapporten gegenereerd worden voor het inzichtelijk maken van de voortgang van projecten en de behaalde rendementen. Tevens worden deze gegevens gebruikt voor het factureren van klanten.

Voor klanten kan PTM in verschillende versies geleverd worden die toegespitst zijn op de taken urenregistratie, projectregistratie of taakregistratie.

PTM is geïmplementeerd als een webapplicatie en maakt gebruik van JavaBeans en Java Server Pages. De JavaBeans zijn als normale J2SE beans geïmplementeerd. Voor de gegevensopslag maakt PTM gebruik van Firebird.

PTM is via een browser benaderbaar. Hierdoor is geen installatie vereist op de pc van de gebruiker. Voor de weergave in de browser wordt gebruik gemaakt van HTML en CSS met een beperkt gebruik van JavaScript.

3.3 Probleemstelling

PTM werkt binnen ISAAC goed voor het bijhouden van klantgegevens, projecten en projecturen. Echter het systeem is technologisch verouderd waardoor de opgeslagen gegevens in PTM niet ten volle benut kunnen worden.

Zo moeten rapportages vaak buiten PTM bewerkt worden om de daarin weergegeven gegevens inzichtelijk te maken met behulp van berekeningen en/of grafieken, waardoor het inzichtelijk maken van deze gegevens onnodig bewerkelijk is.

Ook is het systeem vergeleken met de huidige Web 2.0 technieken gebruikersonvriendelijk. Met Web 2.0 technieken kan voorkomen worden dat gebruikers vaak tussen verschillende pagina's moeten wisselen. Dit kan door bijvoorbeeld invoer- en bewerktfunctionaliteit op de pagina zelf aan te bieden zodra de gebruiker nieuwe gegevens wil invoeren of oude gegevens wil bewerken.

Tevens is de achterliggende code al enkele jaren oud waardoor verschillende programmeurs aan PTM hebben gewerkt voor het toevoegen van een nieuwe functionaliteit. Hierdoor zijn verschillende programmeerstijlen gebruikt die de onderhoudbaarheid en leesbaarheid verminderen. Tevens is deze code door de introductie van nieuwe technieken flink verouderd en kan veel voordeel gehaald worden qua onderhoudbaarheid door het toepassen van deze nieuwe technieken.

De probleemstelling is daarom als volgt gedefinieerd:

“De huidige versie van PTM is technologisch verouderd en daardoor niet eenvoudig uitbreidbaar en onderhoudbaar. Verder is PTM door het gebruik van verouderde technieken, in vergelijking met de huidige technieken, gebruikersonvriendelijk. Daarnaast is het maken van rapportages met behulp van de gegevens in PTM onnodig bewerkelijk en daardoor tijdsconsumerend. Ook zijn de mogelijkheden tot het soort rapportages dat PTM genereert te beperkt.”

3.4 Doelstelling

De doelstelling is met behulp van de probleemstelling uit paragraaf 3.3 als volgt te formuleren:

“De door ISAAC gebruikte core-functionaliteit van PTM dient opnieuw te worden ontwikkeld met gebruik van de huidige technieken, met als doel het programma eenvoudiger uitbreidbaar, efficiënter en meer gebruikersvriendelijk op te zetten. De gegevens die in PTM opgeslagen zijn dienen eenvoudig opvraagbaar en inzichtelijk te zijn met behulp van rapportages en grafieken.”

4 Opstarten project

In dit hoofdstuk wordt het procesverloop van de uitvoering van het opstarten van het Afstudeerproject beschreven. Aan bod komen de eerste kennismaking en het vastleggen van de opdrachtomschrijving in de vorm van een Plan van Aanpak. Hieronder vallen de aanpak, de tegengekomen problemen en hoe hiermee is omgegaan, de resultaten en een korte evaluatie.

4.1 Kennismaking

De eerste werkdag van de afstudeerstage was op maandag 3 september 2007. Die dag startte met het kennismaken met de werknemers van ISAAC en een korte rondleiding binnen het bedrijf.

Onderdeel van het in gebruik nemen van de werkplek was het voor gebruik gereedmaken van de beschikbare computer. Dit was inclusief het verzamelen van enkele onderdelen, zoals geheugen, zodat de computer beter geschikt was voor het ontwikkelen en het testen van software. Daarnaast dienden ook de benodigde software te worden geïnstalleerd en geconfigureerd.

Ook diende administratieve werkzaamheden te worden verricht, zoals het doornemen en ondertekenen van de afstudeerovereenkomst en het aanmaken van login-accounts. Verder heeft een eerste verkenning plaatsgevonden van de gebruikte omgevingen, zoals JBOSS (locale test server) en Eclipse (ontwikkelomgeving) en het inlezen van de gebruikte J2EE technieken binnen het bedrijf zoals Enterprise JavaBeans 3.

4.2 Plan van Aanpak

Het eerste deel van de afstudeerstage was het omschrijven en definiëren van het project in het Plan van Aanpak, met behulp van de geschreven projectomschrijving een definitieve opdrachtomschrijving te maken en deze in te leveren bij de examencommissie.

Het schrijven van een Plan van Aanpak was een terugkerend en goed gedefinieerd proces binnen de opleiding, waardoor een raamwerk met vaste onderdelen reeds beschikbaar was. Dit raamwerk vormt de basis voor het opzetten van het Plan van Aanpak.

De invulling van de vaste onderdelen van het Plan van Aanpak zijn voor een groot gedeelte zelfstandig uitgevoerd, met zo nodig, navraag bij collega's of mijn bedrijfsbegeleider.

Het Plan van Aanpak is in deelstukken beoordeeld, hierdoor kon beoordeling en schrijven bijna gelijktijdig plaatsvinden. Na bespreking met de bedrijfsbegeleider zijn enkele algemene wijzigingen doorgevoerd in het Plan van Aanpak. Ook de docentbegeleider had een enkele aanpassing waarna de aangepaste versie van het Plan van Aanpak is geaccordeerd door bedrijfsbegeleider en de docentbegeleider.

4.3 Problemen

Het schrijven van het Plan van Aanpak verliep voorspoedig en zonder problemen. Het Plan van Aanpak is dan ook op de afgesproken datum opgeleverd

4.4 Resultaat & Evaluatie

Het resultaat van de eerste periode is het Plan van Aanpak (bijlage 1).

Het schrijven en opleveren van het Plan van Aanpak is goed verlopen. Complex was het indekken van projectrisico's en het zorgen voor een goede omschrijving van de begeleiding. De reden daarvoor was de mogelijkheid van langer durende uitval in verband met chronische ziekte. En de daarmee samenhangende andere aard van de begeleiding. Dit is uiteindelijk goed omschreven in het Plan van Aanpak.

II

Onderzoek

Dit onderdeel van de afstudeerscriptie behandelt het onderdeel onderzoek van het afstudeerproject. Het onderzoeksontwerp met onderzoeksmodel, en de hiervan afgeleide vraagstelling worden behandeld. Als laatste wordt het verloop van het onderzoek, de gehanteerde aanpak, fasering en uiteindelijk resultaat behandeld.

5 Onderzoeksontwerp

Het hoofdstuk Onderzoeksontwerp beschrijft de afbakening en definitie van het onderzoeksgebied. Zo wordt de doelstelling van het onderzoek formeel gedefinieerd en worden de onderzoeksmodellen voor de twee afzonderlijke subdoelen uitgewerkt.

Van de onderzoeksmodellen wordt de vraagstelling afgeleid en eenduidig opgesteld. Per deelvraag wordt het onderzoeksmateriaal aangegeven samen met de verwerkingsmethode. Als laatste wordt de onderzoeksstrategie behandeld.

5.1 Doelstelling

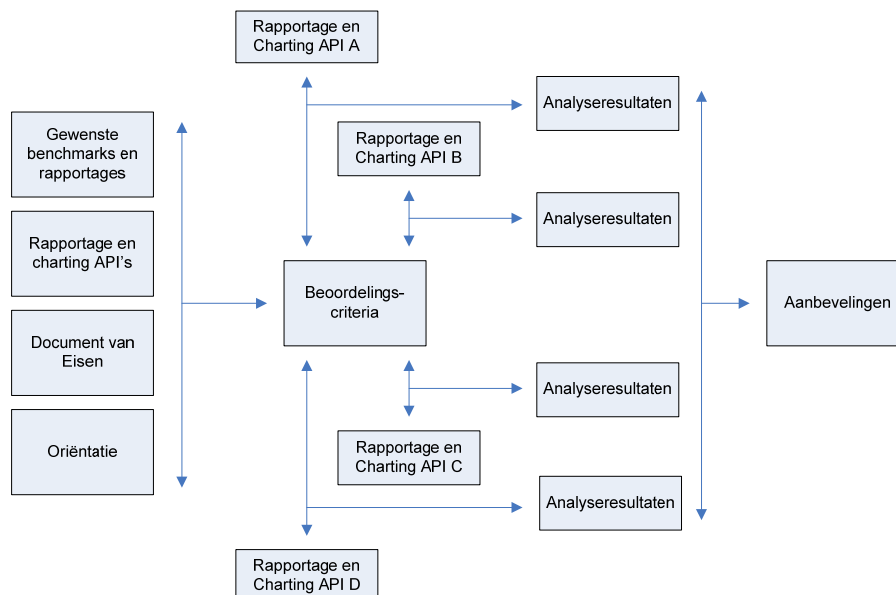
Het onderzoeksontwerp is opgesteld volgens de methode van Piet Verschuren en Hans Doorewaard (zie literatuurlijst) en begint met het vaststellen van het type onderzoek en het opstellen van de doelstelling hiervan.

Uit de algemene doelstelling in hoofdstuk 3.4 volgen twee subdoelen, te weten:

- D1 Opleveren van een vernieuwd en herontworpen PTM waarin de voor ISAAC belangrijke core-functionaliteit geïmplementeerd zijn.
- D2 Het inzichtelijk maken van de gegevens in PTM met behulp van rapportages en grafieken die de voor ISAAC belangrijke gegevens inzichtelijk maken.

5.2 Onderzoeksmodel

Het onderzoeksmodel voor de aanbevelingen met betrekking tot het te gebruiken Rapportage en Charting API is weergegeven in Figuur 2.



Figuur 2 Onderzoeksmodel Rapportage en Charting API's

De beoordelingscriteria zijn opgesteld op basis van bestuurderde gewenste benchmarks uit gesprekken met belanghebbenden, voorbestudering van de door charting en rapportage API's geboden mogelijkheden, op basis van (use cases en eisen uit) het document van eisen en een oriëntatie op de beschikbare charting en rapportage API's.

Voor een goed verloop van het onderzoek wordt allereerst een inventarisatie gemaakt van de eisen waaraan een rapportage en charting API moet voldoen. Hierdoor is het onderzoek opgesplitst in twee onderzoeksfases, namelijk in een oriëntatiefase en onderzoeksfase.

In de oriëntatiefase worden de informatiewensen van de belanghebbende onderzocht waardoor een lijst van gewenste rapportages ontstaat en van de gewenste inhoud van deze rapportages. Deze lijst wordt vervolgens vergeleken met de beschreven functionaliteit in het Document van Eisen van PTM waardoor duidelijk wordt welke gegevens beschikbaar zijn. Hierdoor ontstaat een overzicht van de verschillende rapportages met daarin vermeld of deze mogelijk zijn en zo ja in welke vorm. Aan de hand van deze lijst wordt een lijst van criteria opgesteld waarmee de Rapportage en Charting API's vergeleken worden.

Als laatste stap in de oriëntatiefase worden maximaal vier Rapportage en Charting API's geselecteerd. Dit om de scope van het onderzoek en de duur van het onderzoek te beperken.

De vier rapportage en charting API's worden geselecteerd aan de hand van de volgende eisen:

- Java gebaseerd
- Integreerbaar in een J2EE architectuur

In de onderzoeksfase worden de gekozen Rapportage en Charting API's met elkaar vergeleken met behulp van de opgestelde criterialijst en de onderzoeksvragen (zie hoofdstuk 5.3 Vraagstelling). Dit leidt uiteindelijk tot een advies voor de meest geschikte Rapportage en Charting API.

5.3 Vraagstelling

De kern van het onderzoek is een duidelijk geformuleerde vraagstelling. Deze dient als een controle voor de uitkomst van het onderzoek; alle vragen dienen te zijn beantwoord.

De vraagstelling is opgesplitst in drie centrale vragen. Dit zijn de hoofdvragen die beantwoord dienen te worden binnen het onderzoek. Waar nodig zijn deelvragen toegevoegd.

In de eerste centrale vraag wordt getracht te achterhalen wat de criteria zijn waaraan een Rapportage en Charting API moet voldoen voor gebruik binnen de vernieuwde PTM.

- I Wat zijn de criteria voor de beoordeling van deze rapportage en charting API's?
- a. Welke rapportages en grafieken wil men kunnen genereren in PTM?
 - b. Welke van deze rapportages en grafieken kunnen daadwerkelijk gegenereerd worden met de beschikbare gegevens in PTM?
 - c. Aan welke licentievoorwaarden en/of kostenvoorwaarden moet worden voldaan om een rapportage en charting API te gebruiken binnen de vernieuwde PTM?
 - d. Aan welke voorwaarden voor uitvoerformaat moet een rapportage en charting API voldoen?
 - e. Aan welke verdere gebruikerswensen en voorwaarden moet de rapportage en charting API voldoen?

In de tweede centrale vraag worden de vier gekozen rapportage en charting API's vergeleken en geëvalueerd.

- II Hoe worden de vier onderzochte rapportage en charting API's beoordeeld in het licht van de gestelde criteria?
- a. Heeft de rapportage en charting API de benodigde rapportage en charting functionaliteit die gewenst is voor de vernieuwde PTM?
 - b. Welke licentievoorwaarden en kostprijs zijn verbonden aan het gebruik van de rapportage en charting API?
 - c. Is de rapportage en charting API geschikt voor de integratie in een EJB3.0 programma met een rich client weergave?
 - d. Welke andere weergave formaten worden ondersteund door de rapportage en charting API?
 - e. Welke specifieke voor- en nadelen heeft het gebruik van de rapportage en charting API?

In de derde centrale vraag wordt op de basis van de vergelijking en evaluatie van de vier gekozen rapportage en charting API's een advies over de geschiktheid uitgebracht.

- III Welke aanbevelingen worden afgeleid uit de vergelijking van de vier onderzochte rapportage en charting API's?

5.4 Begripsbepaling

Om een goede begrenzing van de vraagstelling te kunnen bereiken wordt gebruik gemaakt van een afbakeningsmethode. Deze bestaat uit een afbakening van kernbegrippen die gebruikt zijn in de vraagstellingen.

Voor het bepalen van deze kernbegrippen is afgeweken van de standaard procedure zoals beschreven in de methode van Piet Verschuren en Hans Doorewaard (zie literatuurlijst). Mogelijk onduidelijk begrippen worden, zonder tussenstap, onderstaand beschreven.

Rapportage en charting API's

Dit zijn in Java geprogrammeerde softwarebibliotheken die in een EJB3.0 programma integreerbaar zijn. En tevens geschikt zijn voor het weergeven van gegevens in een intranet/internet applicatie.

Weergave formaten

Dit zijn de mogelijkheden om gegenereerde rapportages en grafieken weer te geven op een webpagina, als Excel of pdf bestand te exporteren. Verder wordt gekeken of het mogelijk is om grafieken in een voor een intranet/internet applicatie geschikte vorm weer te geven.

Specifieke voor en nadelen

Dit zijn de voor en nadelen van een rapportage en charting tool die tijdens het onderzoek naar voren zijn gekomen maar niet tot een van de andere gestelde vragen behoort.

6 Onderzoeksplanning

Dit hoofdstuk beschrijft het type fasering dat tijdens het onderzoek gebruikt is. De losse taken komen elk kort aan bod. De informatie in dit hoofdstuk dient als leidraad voor het volgen van het procesverslag in hoofdstuk 7.

6.1 Fasering

Voor de planning en fasering van het onderzoek is afgeweken van de methode beschreven door P. Verschuren en H. Doorewaard. Hiervoor is gekozen omdat deze methode, alhoewel grondig en duidelijk, te veel tijd zou innemen. Vanwege uitval door ziekte was het nodig om tijd in te halen en is de fasering vereenvoudigd opgezet.

Voor het onderzoek is de volgende vereenvoudigde fasering opgezet:

Omschrijving	Van	Tot	Werkdagen
Interviews functionaliteit	12-09-2007	28-09-2007	13
Onderzoeksaanpak plan	19-09-2007	22-10-2007	9
Onderzoek statistieken weergave	22-10-2007	13-11-2007	16
Onderzoeksrapport	1 11-2007	22-11-2007	15

Tabel 1: Onderzoeksfasering

De fases “*Interviews functionaliteit*” en “*Onderzoeksaanpak plan*” werden tegelijkertijd uitgevoerd met het opzetten van het document van eisen voor het software ontwikkeling traject.

Hiervoor is gekozen omdat gegevens uit de fase “*Interviews functionaliteit*” gebruikt werden voor zowel het onderzoek als het opzetten van het document van eisen. Vervolgens werd het document van eisen gebruikt in de fases “*Onderzoek statistieken weergave*” en “*Onderzoeksrapport*”.

Het document van eisen is een belangrijk document voor de vastlegging van de gewenste en beloofde functionaliteit van een software ontwikkelingsproject. En deze gegevens waren nodig voor een correcte evaluatie en selectie van een voor PTM geschikte rapportage en charting API.

6.2 Beschrijvingen fasering

De hierboven opgesomde fases uit tabel 6.1 bestaan uit de volgende taken:

I Interviews functionaliteiten

In kaart brengen van gewenste rapportages en functionaliteit.

II Onderzoeksaanpak plan

Projectkader, doelstelling, onderzoeksmodel, vraagstelling en kernbegripsomschrijvingen opstellen en bepalen. Onderzoeksmateriaal en strategie kiezen en het maken van een onderzoeksplanning.

Verzamelen van links, boeken, artikels en raadplegen zoeksystemen.

III Onderzoek statistieken weergave

Verzamelen van eventueel aanvullend materiaal als niet alle onderzoeksvragen uit de vraagstelling beantwoord kunnen worden. Ook is het mogelijk om in dat geval de vraagstelling aan te passen.

IV Onderzoeksrapport

Het uiteindelijke rapport waarin alle onderzoeksvragen beantwoord worden met beknopte managementsamenvatting. Zie bijlage 2 - onderzoeksrapport.

7 Procesverslag Onderzoek

In dit hoofdstuk wordt het procesverloop van de uitvoering van het onderzoek van de afstudeeropdracht behandeld. Hieronder vallen de aanpak, de tegengekomen problemen en hoe hiermee is omgegaan, de resultaten en een korte evaluatie.

7.1 *Aanpak*

Voor het ontwerpen en uitvoeren van dit onderzoek is gebruik gemaakt van de methode van P. Verschuren en H. Doorewaard welke methode binnen de opleiding Informatica gebruikt.

Deze methode is ontwikkeld voor het opzetten van grootschalige en langdurige onderzoeken. En wordt daarom veel gebruikt bij wetenschappelijk onderzoek, zoals medische trails, waar een goede definitie en vastlegging cruciaal is.

Voor een kleinschaliger onderzoek, zoals tijdens een afstudeerperiode, is de standaardvorm minder geschikt, door de tijd die benodigd is om elke fase correct en formeel te doorlopen en te documenteren. Daarom is, zoals aangegeven in hoofdstuk 5 en 6, voor de onderdelen begripsbepaling en fasering van het onderzoek afgeweken van de standaard methode.

De in de methode gebruikte stappen, zoals het onderzoeksontwerp, zijn al aan bod gekomen en in detail na te lezen in de hoofdstukken 5 en 6.

7.2 *Resultaten*

In deze paragraaf worden de afzonderlijke fasen van het onderzoek behandeld. Per fase wordt beschreven welke taken zijn uitgevoerd en op welke manier. Verder worden de eventuele problemen die zich voordeden behandeld evenals de ondernomen acties om deze problemen af te vangen.

7.2.1 Interviews functionaliteiten

De interviews voor het achterhalen van de gewenste functionaliteit en rapportages waren gepland van 13 september tot en met 15 oktober 2007, 13 werkdagen in totaal.

Deze fase is tevens gebruikt voor het winnen van informatie en feedback ten behoeve van het schrijven van het document van eisen.

De uiteindelijke interviews konden één dag eerder starten en waren ook eerder klaar dan de daarvoor ingeplande tijd. Dit kwam doordat er enkele openingen waren in de agenda's van de te interviewen personen en door het vroegtijdig maken van afspraken (twee weken voor de start van deze periode). In totaal zijn er interviews gehouden met vijf personen, te weten:

1. Dhr. M. Hufkens (Financiële zaken, CFO)
2. Dhr S. Daemen
3. Dhr. P. Luedke (account manager)
4. Dhr K. Peters (softwarearchitect)
5. Mw. K. Schengenga

Het houden van de interviews was een lopende taak met interviews van maximaal één uur, maar bij voorkeur dertig minuten, om zo het gesprek kort en bondig te houden. Tijdens deze gesprekken werd duidelijk dat de huidige opzet van PTM positief gewaardeerd werd en goed bruikbaar was, maar dat het

invoeren en bewerken van gegevens niet helemaal aansloot bij de werkwijze die tegenwoordig binnen het bedrijf gebruikt wordt.

Dit komt voornamelijk doordat de oude PTM nog opgezet was met behulp van losse pagina's waardoor er vaak gewisseld moest worden tussen schermen. Dit wilde men graag gecorrigeerd zien.

Tevens kwamen de wensen ten aanzien van de vorm van de rapportages naar voren. De huidige vorm van rapportages genereren binnen PTM was goed bruikbaar en leverde de gegevens op die men nodig had voor het maken van analyses en rapportages. Deze functionaliteit wilde men graag tevens in de vernieuwde PTM aangevuld met de mogelijkheid om bepaalde standaard rapportage en grafieken te maken binnen PTM. Een voorbeeld hiervan is het makkelijk inzichtbaar maken van factureerbare uren en niet factureerbare uren voor een project. Bij de huidige versie van PTM vindt dit plaats op basis van nacalculatie en door het beoordelen van ingevoerde opmerkingen bij gewerkte uren.

Ook kwamen toekomstige wensen aan bod zoals de mogelijkheid om taken aan te maken voor een project (met het aantal toegekende te werken uren) en deze toe te kunnen wijzen aan een medewerker. Er is voor gekozen om deze wens niet te implementeren omdat dit, gezien de tijdslimiet van de afstudeerstage en de opnieuw te implementeren functionaliteit, niet haalbaar was. Dit wordt meegenomen in een toekomstige uitbreiding.

De uiteindelijk vastgelegde wensen uit deze interviews kunt u vinden in het document van eisen (zie bijlage 4) en het onderzoeksrapport (zie bijlage 3).

7.2.2 Onderzoeksaanpak plan

Het schrijven van het onderzoeksaanpak plan was gepland van 19 september 2007 tot en met 22 oktober 2007, 9 werkdagen in totaal.

Het schrijven van het onderzoeksaanpak is uiteindelijk verplaatst naar 1 oktober met als uiteindelijke opleverdatum 6 december 2007.

De start van het schrijven van het onderzoeksaanpak plan is verschoven omdat die datum uiteindelijk niet realistisch bleek in verband met de benodigde tijd voor het schrijven van het document van eisen.

Ondanks bovengenoemde redenen en enkele zieke dagen was een eerste werkbaar versie beschikbaar op 17 oktober 2007. Inhoudelijk en qua opzet was deze goed en bruikbaar voor het onderzoek. De leesbaarheid voor derden was minder. Hierdoor zijn er tijdens het onderzoek nog enkele aanpassingen aan het onderzoeksaanpak plan gedaan.

Tevens werden de vragen en het onderzoeksmodel goed bekeken en opnieuw opgezet waardoor het onderzoek pas later kon beginnen.

De details van het onderzoeksaanpak plan kunt u lezen in hoofdstuk 5 en 6, tevens is het onderzoeksaanpak plan bijgevoegd als bijlage twee.

7.2.3 Onderzoek statistieken weergave

Het onderzoeken van de charting en rapportage API's was gepland van 22 oktober tot en met 13 november 2007, in totaal 16 werkdagen. Deze fase liep vanaf 1 november 2007 samen met het schrijven van het onderzoeksrapport.

Het onderzoeken van de charting en rapportage API's is uiteindelijk verplaatst naar 5 november 2007 en liep tot en met 7 december 2007 waar ook het schrijven van het onderzoeksrapport eindigde. Het heeft dus in totaal 20 werkdagen geduurd, vier werkdagen langer dan oorspronkelijk gepland. Deze verplaatsing is veroorzaakt door de in hoofdstuk 7.2.2 genoemde vertraging.

Het onderzoek zelf verliep zonder noemenswaardige problemen.

Tijdens het onderzoek zelf was jammer genoeg geen tijd om een klein werkend prototype van de verschillende rapportage en charting API's op te zetten. Dit kwam door problemen met het integreren van de rapportage en charting API's in een project, of door onduidelijkheden in de documentatie, of een combinatie van beide.

De tegengekomen problemen staan in detail omschreven in het onderzoeksrapport (zie bijlage drie) en zal in deze scriptie in het kort behandeld worden in hoofdstuk 7.3.

7.2.4 Onderzoeksrapport

Het onderzoek was opgesplitst in twee delen. Dit is ook terug te zien in het onderzoeksrapport zelf. Deze twee delen zijn als volgt opgezet:

1. In het eerste deel word onderzocht op welke punten rapportage en charting API's met elkaar vergeleken moeten worden en met behulp hiervan wordt een tabel met criteria opgezet waarmee de onderzochte API's eenvoudig vergeleken konden worden. Tevens werd aan het einde van de eerste centrale vraag een voorselectie gemaakt voor een aantal API's die volgens beschikbare documentatie geschikt zijn voor integratie in een J2EE project en de gewenste functionaliteit lijken te bieden.
2. In het tweede deel worden voor de geselecteerde rapportage en charting API's de vragen beantwoord, en de rapportage en charting API's met elkaar vergeleken en hieraan wordt uiteindelijk een conclusie aan verbonden. Tijdens het schrijven van het onderzoeksrapport is er voor gekozen bij elke vraag zelf de verwijzingen naar gebruikte internetpagina's en documenten toe te voegen, omdat er een groot aantal diverse bronnen zijn geraadpleegd tijdens het schrijven van het onderzoeksrapport. Op deze manier zijn de gebruikte bronnen per vraag te achterhalen.

De onderzocht rapportage en charting API's waren:

1. **BIRT**
Business Intelligence and Reporting Tools (BIRT) is geselecteerd doordat deze tool vaak in commerciële producten gebruikt wordt en zeer belovend uitziet qua mogelijkheden.
2. **JasperReports**
JasperReports is een open source Reporting & Charting Tool die ontwikkeld wordt door het bedrijf JasperSoft. Qua mogelijkheden is deze sterk te vergelijken met BIRT.
De standaard producten van JasperReports zijn vrij beschikbaar en te gebruiken, maar er worden tevens betaalde gespecialiseerde oplossingen aangeboden voor het optimaliseren van rapportage generatie.
3. **Crystal Reports for Eclipse (CR4E)**
Crystal Reports is de Reporting & Charting Tool van BusinessObjects. CR4E maakt gebruik van de Crystal Reports voor Java Engine. Qua mogelijkheden lijkt deze ook te voldoen aan de door ISAAC gestelde eisen.

Voor deze rapportage en charting API's is gekozen omdat ze alle drie in Java geschreven zijn en dus eenvoudig in een Java project (zoals PTM) te integreren zijn. Tevens leken deze rapportage en charting API's in een eerste korte vergelijking aan gestelde voorwaarden te voldoen.

Voor het in detail nalezen van het onderzoeksresultaat en conclusies kunt u het onderzoeksrapport raadplegen dat is bijgevoegd als bijlage drie.

7.3 Conclusies

Alle van de onderzochte rapportage en charting API's zijn uitgebreid genoeg om te voorzien in een breed scala aan rapportages. Qua grafieken is de uitzondering JasperReports die het minst aantal soorten grafieken ondersteunt waardoor bepaalde rapportages niet begeleid kunnen worden met een verduidelijkende grafiek.

De keuze voor een rapportage en charting API wordt daarom voornamelijk gebaseerd op beschikbare documentatie en hoe gebruikersvriendelijk een rapportage en charting API is. Op dit gebied is van de onderzochte rapportage en charting API's BIRT de beste keuze door een combinatie van de volgende factoren.

Van de onderzochte rapportage en charting API biedt BIRT een gebruikersvriendelijke interface voor het ontwerpen. Met deze interface kan een nieuwe programmeur vrij snel standaard rapportages ontwikkelen en zijn zelfs gevorderde functionaliteiten vrij snel te vinden en eenvoudig te gebruiken.

De documentatie is niet de beste van de onderzochte rapportage en charting API's maar komt zeer dicht in de buurt van de kwaliteit van JasperReports. In combinatie met een levendige community op het internet is er veel informatie hierover beschikbaar.

De integratie van BIRT in een applicatie is redelijk complex door de hoeveelheid configuratiewerk die verzet moet worden. De enige uitzondering hierop is Crystal Reports die zeer eenvoudig te installeren en te gebruiken is binnen een applicatie (wel met enig inventief gebruik van de Crystal Reports plugin door ontbrekende documentatie). Eenmaal geïntegreerd is de documentatie van BIRT zeer goed en is het gebruik van de rapportages ook eenvoudig.

Het oorspronkelijke advies van het rapport is daarom gebruik te maken van BIRT voor het toevoegen van uitgebreide rapportage mogelijkheden in Java projecten.

Sinds het schrijven van het onderzoek is door MyEclipse een aangepaste versie gelanceerd van BIRT, genaamd MyEclipse Reports. Meestal zijn de producten van MyEclipse eenvoudig te integreren. Het is daarom de moeite waard om te kijken of MyEclipse interessant is om te gebruiken.

De details van het onderzoek zijn na te lezen in het onderzoeksplan en het onderzoeksrapport, respectievelijk bijgevoegd als bijlage 2 en 3.

7.4 Evaluatie

Het voeren van interviews leverde geen problemen op en was zeer verhelderend met betrekking tot de wensen. Tijdens het interview werd niet gewerkt met een vaste vragenlijst maar was de geïnterviewde vrij om zijn wensen uit te drukken en werd waar nodig om verduidelijking gevraagd. Dit was mogelijk omdat de geïnterviewde personen goed verwoorde meningen hadden over wat ze wilden in de vernieuwde PTM en wat ze vonden van de huidige PTM versie.

Het maken van het onderzoeksplan was moeilijker, doordat er tijdens deze periode enkele keren sprake was van uitval door ziekte en overbelasting. Het bleek dat de ingeschaalde zes werkuren per dag niet haalbaar waren. Haalbaar is maximaal vijf werkuren per dag. Het was zeer prettig dat de medewerkers van ISAAC hier begrip, ondersteuning en de ruimte voor gaven.

Het opzetten van een onderzoek volgens de methode van P. Verschuren en H. Doorewaard was complex, maar leverde wel resultaat op.

III

Software Ontwikkeling

In dit onderdeel van de afstudeerscriptie komt de ontwikkeling van de nieuwe PTM applicatie aan bod, waaronder de planning voor het ontwikkelen van de software, de gebruikte architectuur en technieken en het procesverslag software ontwikkeling.

8 Planning Software Ontwikkeling

In dit hoofdstuk wordt de gebruikte systeemontwikkelingmethode en de opgezette planning voor het onderdeel software ontwikkeling besproken. Hierbij wordt kort ingegaan op het voordeel van iteratief ontwikkelen. De informatie in dit hoofdstuk dient als leidraad voor het volgen van het procesverslag in hoofdstuk 10.

8.1 Fasering

Het software ontwikkeling deel is opgesplitst in de volgende vier fases:

1. Ontwerpfase
2. Bouwfase 1
3. Bouwfase 2
4. Testfase

In de hoofdstukken 8.1.1 tot en met 8.1.4 wordt de inhoud van deze fases behandeld worden.

Voor de bouwfases 1 en 2 van de software ontwikkeling wordt een aangepaste vorm van de IAD systeemontwikkelingmethode gebruikt. IAD staat voor “Iterative Application Development” waarbij het kernwoord Iterative is. Deze methode is gekozen omdat deze bij ISAAC vele malen met succes is gebruikt en ook uitermate geschikt is voor een project van deze aard.

De IAD methode steunt op het verscheidene malen doorlopen van dezelfde (meestal korte) lifecycle. Hierdoor kunnen tijdens de uitvoering van het project nog wijzigingen in de requirements doorgevoerd worden zonder dat dit veel vertraging veroorzaakt. Een groot voordeel is tevens dat elke iteratie een werkend prototype oplevert, met steeds verder uitgebreide functionaliteit.



Figuur 3 Fasen Iterative Application Development

Normaal gesproken wordt aan het begin van elke iteratie besloten wat de op te leveren functionaliteit is en volgens welke planning deze functionaliteit geïmplementeerd wordt. Dit levert een werkdokument op waarmee de deelfunctionaliteit ontworpen en geïmplementeerd zal worden. Met dit document wordt de voortgang bewaakt en de uiteindelijk opgeleverde functionaliteit besproken.

Voor dit project is ervoor gekozen om de planning en opsplitsing in iteratie voor de start van de bouwfases uit te voeren en toe te voegen aan het plan van aanpak (zie bijlage 1 plan van aanpak en hoofdstuk 8.4). Deze planning is vervolgens gebruikt voor het bewaken van de voortgang en het bespreken van de resultaten met de bedrijfsbegeleider.

8.1.1 Ontwerpfase

In de ontwerpfase wordt het domeinmodel en databasemodel ontworpen aan de hand van het document van eisen (dat geschreven is tijdens de start van het onderzoek). Het ontwerp wordt getest met behulp van prototype code, die eventueel aan het einde van de fase een kleine werkende prototype oplevert.

8.1.2 Bouwfase 1

In bouwfase 1 wordt de vernieuwde PTM ontwikkeld en getest aan de hand van de in de ontwerpfase ontwikkelde ontwerpen. Het testen gebeurt zoveel mogelijk tijdens het ontwikkelen van de nieuwe code. Een grondige testronde zal plaatsvinden in de testfase (zie hoofdstuk 8.1.4)

Het bouwen van de software gebeurt in acht iteraties (waarvan twee optioneel). Deze iteraties variëren afhankelijk van de complexiteit in duur (één week tot en met drie weken lang).

8.1.3 Bouwfase 2

In bouwfase 2 wordt de gekozen rapportage en charting API geïmplementeerd (zie deel II onderzoek van dit verslag voor meer details) voor het maken van de gewenste rapportages.

8.1.4 Testfase

Doel van de Testfase is het compleet testen van de software en het corrigeren van de gevonden fouten zodat de software opgeleverd kan worden.

Dit bestaat uit:

- Testen van de PTM-software en de rapportage en charting module voor statistieken volgens het Acceptatie Test Plan.
- Corrigeren van gevonden fouten in de software.
- Schrijven van een testrapport waarin beschreven staan de gevonden fouten, oorzaak daarvan en welke acties ondernomen zijn om de fout te corrigeren.

9 Architectuur & Techniek

In dit hoofdstuk komt de architectuur van de vernieuwde PTM aan bod. Het is voornamelijk een technisch hoofdstuk dat door de lezers die enkel in procesbeschrijving geïnteresseerd zijn probleemloos overgeslagen kan worden.

9.1 J2EE/Java Technieken

In dit hoofdstuk wordt de voor PTM gebruikte Java techniek besproken, die gebruikt wordt voor de server.

9.1.1 EJB 3.0

Vrij snel tijdens de start van de afstudeerperiode werd duidelijk dat het gebruik van EJB 3.0 voor het project een groot voordeel zou opleveren.

EJB 3.0 ondersteunt net zoals zijn voorgangers zogenaamde Entity Beans die gebruikt worden om de tabellen in de database te mappen naar POJO's (Plain Old Java Objects). Het grote voordeel van de EJB 3.0 Entity Beans is dat deze eenvoudiger te mappen zijn naar de tabellen in de database. Dit komt doordat in de vorige versie van de EJB, versie 2.2, deze mappings in losse XML bestanden omschreven moesten worden en deze meestal erg verbose waren.

In EJB 3.0 worden de mappings met zogenaamde annotations omschreven. Als voorbeeld van een simpele Entity Bean gebruiken wij de NationalHoliday klasse:

```
package nl.isaac.ptm.ejb.nationalholiday;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;

import org.hibernate.annotations.Type;
import org.hibernate.validator.Length;
import org.joda.time.DateTime;

@Entity
public class NationalHoliday {

    private Integer Id;
    private DateTime day;
    private String name;
    private String description;

    @Id
    public Integer getId() {
        return Id;
    }
    public void setId(Integer id) {
        Id = id;
    }

    @Column(nullable=false)
    @Type(type="org.joda.time.contrib.hibernate.PersistentDateTime")
}
```

```

    public DateTime getDay() {
        return day;
    }
    public void setDay(DateTime day) {
        this.day = day;
    }

    @Column(nullable=false)
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }

    @Length(max=4000)
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
}

```

Er zijn maar zes annotations gebruikt, te herkennen aan de @, om deze klasse compleet te mappen en te koppelen aan de database.

Om van een POJO een Entity Bean te maken zijn alleen de keywords @Entity nodig voor de klasse en @Id voor het aangeven van het ID veld van de tabel. Hierdoor kan men heel snel en eenvoudige Entity Beans opzetten en gebruiken.

Daarnaast worden de overige annotations gebruikt voor het toevoegen van restricties, zoals een maximale veldlengte, of het mappen van een speciaal type naar een veld, zoals een JodaTime² klasse.

Het project LoyaltySuite binnen ISAAC Software Solutions werd al ontwikkeld met behulp van EJB 3.0 technieken. Hierdoor was er al enige ervaring binnen ISAAC waarvan tijdens dit project gebruik gemaakt kon worden. Ook betekende dit dat er al oplossingen beschikbaar waren, bijvoorbeeld standaard code voor het opvragen en opslaan van Entity Beans. Hiervan kan gebruik gemaakt worden tijdens het ontwikkelen van de nieuwe PTM.

9.2 Web Technieken

In dit hoofdstuk worden de technieken besproken die voor de client worden gebruikt.

9.2.1 Flex

Oorspronkelijk was voor de gebruikersinterface voor een combinatie van HTML en JSP pagina's gekozen die gebruik maken van de DOJO Toolkit voor het dynamisch updaten van de schermen (om het herladen van complete pagina's te voorkomen). Waarom hiervan afgeweken is kunt u in hoofdstuk 10 lezen.

Door het overstappen op Flex werden de boven genoemde problemen opgelost en kon er vrij snel gestart worden met het opzetten van een gebruikersinterface.

² JodaTime is een Java datum en tijd API die krachtiger en eenvoudiger in gebruik is dan de standard in JAVA beschikbare klassen voor datum en tijd, en is daarom ook gebruikt binnen de PTM applicatie.

Echter doordat ik nog nooit gebruik heb gemaakt van Flex en de daarbij behorende taal ActionScript 3 is en deze een andere structuur heeft en manier van programmeren heeft is er een vertraging opgetreden door het leren van de Flex. Nadat eenmaal bekend was hoe bepaalde dingen gebruikt en/of opgezet moesten worden in ActionScript werden de mogelijkheden en kracht van deze taal duidelijk.

Tevens heeft het gebruik van Flex het voordeel dat de PTM applicatie eenvoudiger uit te voeren is op de client. Dit komt doordat Flex gebruik maakt van de standaard Flash Player van Adobe die op bijna elke pc geïnstalleerd is.

Uiteindelijk heeft het gebruik van Flex er voor gezorgd dat een mooie en goed werkende gebruikersinterface opgezet is.



Figuur 4 Screenshot van agenda PTM

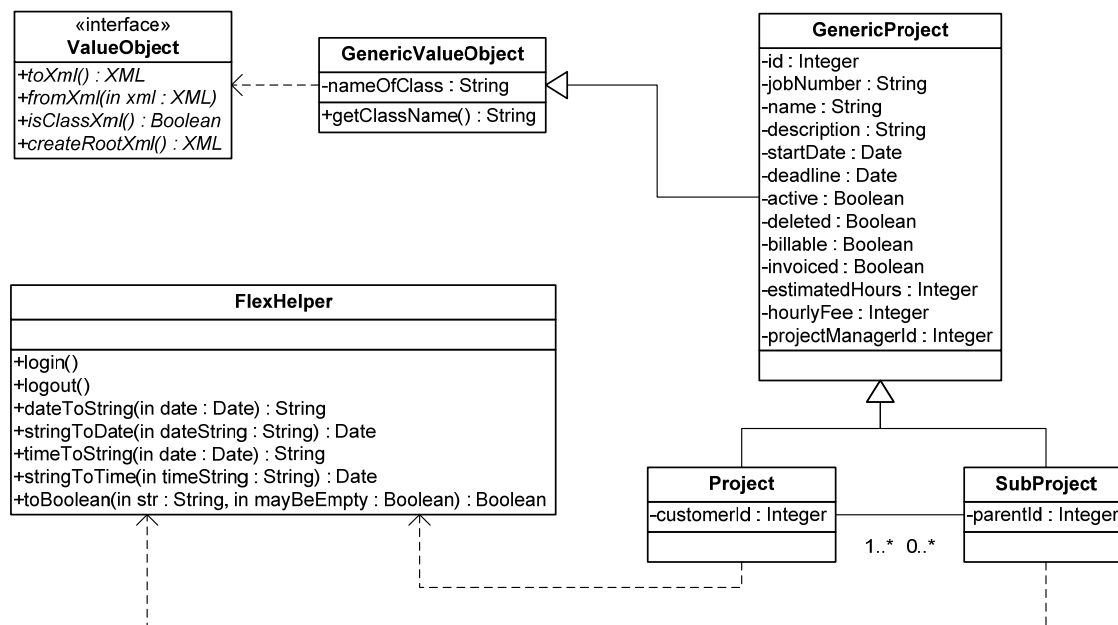
9.3 Data overdracht

Doordat er gebruik is gemaakt van Flex en niet van een Java oplossing zoals JSP pagina's is het niet mogelijk direct gebruik te maken van de logica en Entity Beans op de server.

Hier is een losse oplossing voor geschreven die zowel in de Flex client als in de server geïmplementeerd is..

9.3.1 ValueObjects

Doordat de Entity Beans niet beschikbaar waren binnen de Flex applicatie moest een oplossing gezocht worden voor het tijdelijk opslaan van gegevens in de applicatie om te voorkomen dat continue met de server gecommuniceerd moet worden voor het opvragen van gegevens. Hiervoor zijn de zogenaamde ValueObjects geschreven.



Figuur 5 ValueObject UML diagram

De interface ValueObject wordt alleen gebruikt voor het doorgeven van ValueObjects en om methodes te definiëren die een ValueObject moet bevatten.

De klasse GenericValueObject implementeert de methodes gedefinieerd in de interface en voegt logica toe. Deze logica zorgt ervoor dat een ValueObject altijd correct XML aanmaakt en correct XML data inleest. Hiervoor gebruikt de GenericValueObject de nameOfClass variabele die de naam bevat van de Entity Bean tegenhanger (zie hoofdstuk 9.3.4 XMLHelper voor details).

Een “real world” ValueObject, zoals bijvoorbeeld een project, extends deze GenericValueObject en voegt benodigde variabelen en functionaliteit toe aan de fromXml() en toXml() methodes. Tevens wordt er logica toegevoegd die als verplichte data, zoals een id van een ValueObject, niet ingeladen kan worden een als exception wordt gegenereerd.

Voor het vullen van de ValueObjects met de gegevens die in de aangeleverde XML staan wordt voor bepaalde types gebruik gemaakt van de FlexHelper. Deze klasse bevat methodes voor het converteren van datums en tijden zodat deze correct ingeladen en verzonden worden.

Hiervoor kon geen gebruik gemaakt worden van de standaard functionaliteiten binnen JodaTime en ActionScript omdat beide in formaten verzonden die niet compatible met elkaar waren.

9.3.2 Flex datamanger en RPCHandlers

Voor het bijhouden, opvragen en verzenden van de ValueObjects worden RPCHandlers gebruikt. Deze RPC handlers communiceren met Servlets (zie hoofdstuk 9.3.3. Servlets voor details) op de server waar ze XML documenten opvragen of naar versturen. Elke RPCHandler is verantwoordelijk voor een beperkt aantal, bij elkaar horende, ValueObjecten. Voor de ValueObjecten Project en SubProject is dit de RPCHandler ProjectManager.

Een RPCHandler zoekt in een XML document, in het geval van de project ValueObject, naar de project tag. Voor elke project tag die de RPCHandler vindt wordt een nieuw project ValueObject aangemaakt waar de gevonden XML data naar toe wordt gestuurd. De ValueObject regelt vervolgens het converteren van de XML data en het aanmaken van de subproject ValueObjecten.

Dit kan in bepaalde gevallen ook voor individuele ValueObjects waar het efficiënter is om maar één ValueObject bij te werken in plaats van een complete update.

Voor het updaten van bestaande records in de database, of voor het aanmaken van nieuwe, wordt een enkele ValueObject geconverteerd naar XML en verzonden naar de Servlet die verantwoordelijk is voor het verwerken van de XML data (zie hoofdstuk 9.3.3 Servlets voor details).

De RPCHandler is dus alleen verantwoordelijk voor het bijhouden van ValueObjecten en de communicatie met de server. Voor het toevoegen, updaten en synchronisatie van gegevens is de DataManager verantwoordelijk.

Deze DataManager bevat de instanties van de diverse RPCHandlers en schermt deze af voor de rest van de applicatie. Dit om te voorkomen dat er gegevens bijgewerkt of opgevraagd worden zonder dat de DataManager hier van af weet. Als dit zou gebeuren kan de situatie ontstaan dat bepaalde data niet meer klopt met de gegevens die in de database staan.

Voor bijvoorbeeld het aanpassen of toevoegen van een ProjectPeriod (die de gewerkte tijd voor een dag bijhoudt voor één project van één werknemer) is niet alleen een update nodig van de huidige ProjectPeriod ValueObjects, maar ook van het project waar deze ProjectPeriod aan toegevoegd is. Dit is nodig omdat het toevoegen van ProjectPeriods gevolgen heeft voor gebruikte en beschikbare uren voor een project/subproject.

Als de DataManager een update of toevoeging van een dergelijke ValueObject detecteert zorgt deze er ook voor dat de ProjectManager een update uitvoert.

Voor het bekend maken van wijzigingen in de bijgehouden data in de RPCHandlers heeft de DataManager een aantal Events waar de gebruikersinterface van gebruik kan maken. Dit zorgt er voor dat als er wijzigingen ontstaan in de bijgehouden gegevens deze ook in de weergave in de gebruikersinterface te zien zijn.

9.3.3 Servlets

De servlets zijn verantwoordelijk voor het verzenden van de gegevens van de server naar de Flex client en het ontvangen van gegevens van de Flex client.

Tijdens het versturen of opvragen van gegevens van een servlet wordt altijd een action code meegeven. Aan de hand van deze action code weet de servlet of er gegevens opgevraagd of verzonden worden, en welke acties uitgevoerd moeten worden.

Aan de hand van de action code weet de servlet ook of er extra gegevens verzonden zijn naar de servlet die niet in de XML data staat. Dit komt voornamelijk voor bij het opvragen van gegevens waar bijvoorbeeld een ID voldoende is om de complete gegevens te krijgen.

Door de grote verschillen die er nu gaan optreden tussen het verzenden en ontvangen van gegevens zullen deze apart behandeld worden in “Gegevens ontvangen” en “Gegevens opgevraagd”.

Gegevens ontvangen

Als uit de action code blijkt dat er gegevens ontvangen zijn wordt uit de verzonden request naar de servlet de XML data gehaald en verzonden naar de FlexFacade voor afhandeling en decoding.

In de FlexFacade wordt de XML data met behulp van de XMLHelper (zie voor details hoofdstuk 9.3.4 XMLHelper) die de ontvangen XML data omzet naar een Entity Bean en koppelt deze aan een record in de database (mits deze bestaat).

Als dit correct verloopt, alle verplichte velden zijn gevuld en voldoen aan de in de Entity Bean gedefinieerde voorwaarden, dan wordt de Entity Bean verstuurd naar de EJBFacade die verdere business logic uitvoert en de complexere voorwaarden controleert die niet in de Entity Bean gedefinieerd kunnen worden.

De FlexFacade communiceert het succesvol of niet succesvol uitvoeren van de acties naar de Servlet die dit verzendt naar de Flex Applicatie. Hierdoor kan de Flex Applicatie reageren op fouten als gegevens niet correct verwerkt kunnen worden op de server en wordt bij falen niet gereageerd met een voor de gebruiker onbegrijpelijke foutmelding.

Gegevens opgevraagd

Het opvragen van gegevens gebeurt aan de hand van action code en de servlet waar het naar gestuurd is. Bijvoorbeeld het sturen van de action code 8 met het ID van een bestaand project naar de ProjectServlet zorgt ervoor dat de werknemers die toegevoegd zijn aan het project als XML data naar de Flex Applicatie verzonden worden.

De ProjectServlet stuurt na het ontvangen van de action code het ID van het project door naar de ProjectFlexFacade die vervolgens de werknemers van een project opvraagt bij de ProjectFacade. De ProjectFlexFacade krijgt zo een Collection aan Employees die bij een project horen. Deze worden één voor één naar XML gecodeerd door de XMLHelper (voor details zie hoofdstuk 9.3.4 XMLHelper) en verzonden naar de ProjectServlet. Deze maakt van de rauwe XML data een correct gevormd XML document en stuurt dit door naar de Flex Applicatie.

Als er ergens in het proces een fout optreedt, bijvoorbeeld het project dat bij het opgegeven ID hoort kan niet gevonden worden, wordt in de XML die teruggestuurd wordt een speciale XML tag toegevoegd met daarin de fout informatie. De Flex applicatie kan hierop vervolgens reageren en indien nodig een foutmelding tonen aan de gebruiker.

9.3.4 XMLHelper

Tijdens het ontwikkelen van de applicatie werd snel duidelijk dat er generieke code opgezet moest worden voor het afhandelen van de encoding van en naar XML van de Entity Beans.

In de oude code werd voor elke Entity Bean die gebruikt werd aparte code geschreven voor het coderen en decoderen naar XML. Samen met het afvangen van alle uitzonderingen levert dit veel code op die niet goed te onderhouden was en zeer bewerkelijk en tijdsconsumerend is om op te zetten.

Hier is toen een oplossing voor geschreven met behulp van Java Reflection. Reflection is een binnen Java beschikbare functionaliteit om van een object de klassen, interfaces, methodes, variabelen, etc op te vragen. Alle in de broncode omschreven informatie is via Reflection opvraagbaar en aan te passen.

Omdat er een groot verschil bestaat tussen het decoderen van XML en het encoderen naar XML worden deze twee processen los van elkaar behandeld in “Encoding” en “Decoding”

Encoding

De Entity Bean Project bestaat uit verschillende publiek beschikbare “get”, “set” en “is” methodes, met een bepaald type voor invoer of uitvoer. Voor het Encoderen van een project naar XML worden de publieke “get” en “is” methoden gebruikt.

Van elk van zo’n aangetroffen methode wordt de naam, min “get” of “is”, gebruikt voor het aanmaken van een XML tag. De methode wordt vervolgens aangeroepen om de gegevens die deze methode teruggeeft op te vragen en als tekst op te slaan in de tag. Dit levert dus de volgende XML op:

```
<methodenaam>data</methodenaam>
```

Voor het toevoegen van data aan een XML tag moet deze van het type String zijn. Omdat een “get” een breed scala aan typen kan teruggeven wordt door de XMLHelper gekeken welk type teruggeven wordt en aan de hand van het type wordt dan de benodigde converteer methode aangeroepen.

Op het moment kan de XMLHelper alle types omzetten die een toString methode hebben, JodaTime datums en tijden, en Entity Beans.

JodaTime wordt apart gecodeerd omdat de standaard toString methode geen bruikbare of eenvoudig te converteren String voor de Flex Applicatie oplevert. Aan de hand van de methodenaam wordt daarom ook een JodaTime DateTime object omgezet naar een tijd, dag of een dag en tijd String.

Een methode met de naam getDate met als returntype een JodaTime DateTime object wordt omgezet naar het formaat DD-MM-JJJJ, wat bijvoorbeeld de datum 12-03-2008 kan opleveren. Een zelfde soort conversie wordt uitgevoerd met methodes met in de naam “datetime”, “date”, “day”, “deadline” of “tijd”. Als de XMLHelper geen van de eerder genoemde woorden in de methodenaam vindt dan converteert hij standaard een DateTime naar een string met datum en tijd.

Voor Entity Beans is aparte encoding code geschreven omdat het in bijna alle gevallen alleen nodig is om een ID mee te geven. Bijvoorbeeld het Entity Bean Project heeft een relatie met de Entity Bean Customer. Het coderen van deze Customer Entity Bean is niet nodig omdat deze Customer bijgehouden wordt door andere managers in de Flex Applicatie en op de server andere servlets en facades verantwoordelijk zijn voor deze Entity Bean. De klant is daarom al beschikbaar binnen de Flex Applicatie of opvraagbaar indien nodig.

Alle tags die genegeerd worden aan de hand van de methode namen worden geplaatst in een tag die gemaakt wordt aan de hand van de klasse naam. In het geval van een Project Entity Bean levert dit de XML tag <project> op. Dit is dan ook de naam die ingesteld wordt in de “nameOfClass” variabele in de ValueObjects die de Flex Applicatie gebruikt.

Decoding

Voor het decoderen van XML data wordt een niet met data gevulde, en niet aan een record in de database gekoppelde Entity Bean doorgegeven met de ontvangen XML data.

De XMLHelper voert vervolgens het tegenovergesteld uit van het in “Encoding” omschreven proces. Als eerste controleert de XMLHelper of de hoofdtag van de doorgegeven XML data overeenstemt met de klasse naam van het doorgegeven object en decodeert de XML data alleen als deze twee overeenkomen. Deze controle wordt uitgevoerd omdat Reflection een vrij kostbaar proces is qua CPU verbruik en door deze controle onnodig decoderen wordt voorkomen.

Vervolgens worden alle “set” methodes afgelopen en gevuld met de gegevens die in XML tags staan met dezelfde naam.

Tijdens het decoderen van een tag met als tegenhanger een methode die een Entity Bean accepteert wordt deze Entity Bean opgezocht in de database en doorgegeven aan de setter. De XMLHelper is zo opgezet dat deze elk willekeurige Entity Bean kan opvragen en invoeren in een “set” methode.

Voor JodaTime wordt alleen gekeken naar de naam om een correcte conversie van String naar DateTime object te faciliteren. Omdat een XML tag een tijd, datum of beide kan bevatten is er verschillende logica nodig om dit correct om te zetten naar een JodaTime DateTime object dat *altijd* een datum en tijd verwacht.

Voor alle overige methodes wordt gekeken of deze een valueOf methode hebben of van het type String zijn. Alleen deze kunnen generiek afgehandeld worden door de XMLHelper.

Aan het einde van het decoderen zorgt EJB 3.0 er zelf voor dat niet ingevulde gegevens opgehaald worden uit de database. Als er opdracht voor wordt gegeven, worden de nieuwe gegevens opgeslagen in de database.

10 Procesverslag Software Ontwikkeling

In dit hoofdstuk komt het procesverloop van de uitvoering van het Software Ontwikkeling onderdeel van de afstudeeropdracht aan bod. Hieronder vallen de incrementele ontwikkeling, de status op het moment dat deze scriptie gedrukt is en overige van toepassing zijnde zaken zoals het bijhouden van de urenverantwoording en het versiebeheer.

10.1.1 Definitiefase

Tijdens de definitiefase, die gelijktijdig liep met het onderzoek, is het document van eisen geschreven waarmee de applicatie is ontwikkeld. Het document van eisen is zoals eerder vermeld ook gebruikt voor het opzetten en het uitvoeren van het onderzoek naar de rapportage en charting API's.

Problemen & Oplossingen

Tijdens het schrijven van het document van eisen zijn geen noemenswaardige problemen opgetreden.

Resultaat & Evaluatie

Het resultaat van deze fase is het document van eisen (bijlage 4).

Het schrijven van het document van eisen was redelijk complex doordat de al bestaande functionaliteit in de oude PTM versie geïnventariseerd en opnieuw omschreven moest worden. Daarnaast moest ook de nieuwe functionaliteit, zoals aangegeven door diverse belanghebbenden in de interviews, omschreven worden in het document van eisen.

Een volgende taak die veel tijd en werk in beslag nam was het selecteren van de functionaliteit die geïmplementeerd zou worden. Dit werd veroorzaakt door de scope van de omschreven functionaliteit en de tijdsbeperking die geldt tijdens een afstudeerperiode.

Uiteindelijk is een opzet geschreven voor de functionaliteit, en te implementeren functionaliteit, waarin alle partijen zich konden vinden en is het Document van Eisen gebruikt voor het ontwerpen van de vernieuwde PTM.

10.1.2 Ontwerpfase

Tijdens de ontwerpfase is het domeinmodel ontworpen met het daarbij behorende datamodel. Doordat de applicatie gebruik maakt van Entity Beans en deze bijna een op een gekoppeld zijn aan tabellen in de database is er voor gekozen om deze gelijk aan de hand van het domeinmodel te implementeren. Daardoor liep de iteratie datamodel implementeren vrijwel gelijk aan de ontwerpfase.

Problemen & Oplossingen

Tijdens de ontwerpfase zijn geen noemenswaardige problemen opgetreden.

Resultaat & Evaluatie

Tijdens de ontwerpfase is het domeinmodel ontwikkeld en opgeleverd.

Het ontwerpen van het domeinmodel heeft zelfstandig plaatsgevonden. De tussentijdse ontwerpen zijn ter beoordeling ingeleverd bij mijn bedrijfsbegeleider en met behulp van zijn feedback is het ontwerp verbeterd tot de uiteindelijke versie.

10.1.3 Bouwfase 1

In hoofdstuk 8 van software ontwikkeling staat al beschreven dat een aangepaste vorm van Iterative Application Development methode gebruikt wordt. Deze methode schrijft voor dat elke iteratie een deel van de functionaliteit bevat. In de volgende paragrafen worden uitgevoerde iteraties van de Software Ontwikkeling beschreven aan de hand van de aanpak, de tegengekomen problemen en hoe hiermee is omgegaan, de resultaten en een korte evaluatie per iteratie.

Door problemen met het ontwikkelen in Flex en de daaraan gekoppelde leercurve is het uiteindelijk niet gelukt om de geplande iteraties te volgen. Iteratie 1 is de enige iteratie die volledig volgens de planning is afgerond.

Voor de rest van de ontwikkeling was het doel zoveel mogelijk functionaliteit te implementeren en dit is op incrementele manier uitgevoerd. Dit proces echter dusdanig onderhevig aan verandering en anders dan de gemaakte planning dat dit als één geheel behandeld wordt.

10.1.3.1 Iteratie 1 – Datamodel implementeren

In de eerste iteratie werd het datamodel geïmplementeerd dat voor de applicatie zou worden. Voor het datamodel zelf was geen database ontwerp gemaakt, omdat deze direct gekoppeld is aan het ontwerp van het domeinmodel. Daarom is het domeinmodel ook gebruikt voor het maken van de Entity Beans en aan de hand van de Entity Beans is vervolgens de database ontwikkeld.

Het datamodel diende zo goed herkenbaar mogelijk gekoppeld te worden en zo efficiënt mogelijk opgezet te worden. Hierbij diende er mee rekening gehouden worden dat bepaalde beslissingen voor de opzet van de Entity Beans en de tabellen in de database gevolgen kunnen hebben voor de performance.

Zo maakt de applicatie overal gebruik van de Inheritance Strategy SingleTable voor klassen Entity Beans die gebruik maken van overerving. In een dergelijk geval worden de super en alle subklassen in een tabel opgeslagen in de database. Dit zorgt ervoor dat voor het ophalen van een Entity Bean maar één round trip naar de database nodig is. Voor andere manieren van opzetten zijn er meerdere round trips naar de database nodig omdat de gegevens dan in diverse tabellen opgeslagen worden.

Problemen & Oplossingen

Het implementeren van de Entity Beans verliep zonder noemenswaardige problemen omdat veel mogelijke problemen al onderzocht en opgelost waren met het maken van kleine prototypes voor Entity Beans.

Resultaat & Evaluatie

Deze iteratie werd binnen de gestelde tijdslimiet van drie weken afgerond. Toen waren de voor de eerste opzet en verder ontwikkelen van PTM alle basis klassen en functionaliteit beschikbaar.

Er zijn tijdens het verder ontwikkelen van PTM enkele veranderingen toegepast aan de Entity Beans en een enkele nieuwe toegevoegd. Dit was verwacht tijdens het ontwikkelen en hiermee is rekening gehouden tijdens het inplannen van de iteraties.

10.1.3.2 Verdere ontwikkeling van PTM

Van de oorspronkelijk geplande functionaliteit van PTM voor bouwfase 1 is ongeveer 50% geïmplementeerd. Er is vooral gefocust tijdens het bouwen op het implementeren van de functionaliteit van de agenda en voor het beheren van projecten. De agenda is op het moment van schrijven voor 95% af (één functie is nog niet geïmplementeerd). Het beheren van projecten is voor 75% geïmplementeerd (hier kunnen via de client nog geen nieuwe projecten worden aangemaakt).

De verdere schermen en functionaliteit van PTM zijn voor ongeveer 30% af. Hier is de nadruk gelegd op het opvragen van de basis informatie en het bewerken van bestaande informatie. De informatie is dus grotendeels opvraagbaar en bewerkbaar maar er moet nog redelijk veel werk verzet worden om de resterende functionaliteit te implementeren. In problemen & oplossingen kunt u de details over de tegengekomen problemen lezen.

Problemen & Oplossingen

Oorspronkelijk was voor de gebruikersinterface voor een combinatie van HTML en JSP pagina's gekozen die gebruik maken van de DOJO Toolkit voor het dynamisch updaten van de schermen (om het herladen van complete pagina's te voorkomen). De DOJO Toolkit is een verzameling aan AJAX klassen die de communicatie met de server en de weergave van gegevens voor de programmeur regelt. Het grootste voordeel van de DOJO Toolkit is dat deze al veel kant en klare klassen en grafische interfaces, zogenaamde widgets, heeft die allemaal cross browser compatible zijn. Dit bespaard een programmeur veel debug en programmeerwerk omdat er al zoveel beschikbaar is.

Voor dit project is uiteindelijk niet gebruik gemaakt van de DOJO Toolkit omdat bepaalde combinaties van widgets voor problemen zorgden die niet op te lossen waren. Het vervelendste was dat de complete gebruikersinterface kon verdwijnen tijdens gebruik van bepaalde widgets en de gebruiker vervolgens met een leeg scherm geconfronteerd werd. Dit probleem en enkele andere problemen, gecombineerd met het moeilijk uitbreiden van deze toolkit door gebrekkige documentatie leidden er toe dat er voor gekozen is om over te stappen op Flex. Hierdoor zijn twee weken verloren gegaan tijdens het ontwikkelen van de gebruikersinterface.

Daarnaast was een van de grootste problemen tijdens het ontwikkelen van de gebruikersinterface de onbekendheid met het ontwikkelen van Flex. Binnen ISAAC Software Solutions is al redelijk ervaring opgedaan met het ontwikkelen met Flex, maar PTM was complexer dan de al ontwikkelde software en maakte gebruik van technieken en Flex Componenten die niet, of bijna niet gebruikt waren binnen ISAAC Software Solutions. Dit zorgde ervoor dat er veel uitzoekwerk gedaan moest worden en vaak code herschreven moest worden omdat oorspronkelijke opzetten niet werkten.

Alhoewel ik vaak werkte met Flex code en componenten die nog nooit waren gebruikt binnen ISAAC Software Solutions was de beschikbare kennis wel voldoende om mij te helpen met het oplossen van problemen en het aanleveren van hulpcode. Bijvoorbeeld oor uitleg van een collega over hoe gegevens verzonden en ontvangen kunnen worden met Flex. Deze kennis heeft veel geholpen tijdens het ontwikkelen van de vernieuwde Flex applicatie.

Het tweede grote probleem tijdens het ontwikkelen van Flex was dat deze niet altijd even goed en makkelijk uitbreidbaar is voor het maken van custom componenten. Flex zelf is heel krachtig en goed uitbreidbaar, maar er zijn door Adobe tijdens de ontwikkeling van Flex enkele beslissingen genomen die het zeer moeilijk kunnen maken om custom componenten te schrijven.

Een van die beslissingen is dat bijvoorbeeld bijna alles in componenten en klassen binnen Flex gemarkeerd is als `private` en dus niet uitbreidbaar of aanpasbaar is in een eigen implementatie. Meestal is hier wel een omweg voor te schrijven maar dit koste vaak veel tijd of was uiteindelijk niet te implementeren.

Een van de ontdekkingen tijdens het proberen uit te breiden van bestaande Flex functionaliteit was dat er redelijk vaak gebruik wordt gemaakt van het keyword `mx_internal`. `Mx_internal` is een manier van `private`

markeren van methoden of variabelen in Flex code. Maar vaak werden deze methodes of variabelen toch gebruik op plaatsen waar dit niet mogelijk zou zijn als ze gemarkeerd waren met `private`. Met enige uitzoekwerk en geëxperimenteer met code, is het gelukt om gebruik te maken van code gemarkeerd als `mx_internal`. De manier om deze code te gebruiken is als volgt:

```
package nl.isaac.ptm.controls
{
    use namespace mx_internal;

    public class PtmDateChooser extends DateChooser
    {
        private function dateRenderer(day:int, bold:Boolean):void
        {
            var startRow:int = 1;
            var startCol:int =
getOffsetOfMonth(this.displayedYear, this.displayedMonth);
            var lastDay:int =
getNumberOfDaysInMonth(this.displayedYear, this.displayedMonth);

            var row:int = Math.floor((day-1 + startCol) / 7) +
1;
            var col:int = (day-1 + startCol) % 7;

            var textField:UITextField =
this.mx_internal::dateGrid.mx_internal::dayBlocksArray[col][row];
            var format:TextFormat = textField.getTextStyles();

            format.bold = bold;
            textField.setTextFormat(format);
            this.mx_internal::dateGrid.validateNow();
        }
    }
}
```

Bovenstaande codesnippet is van de aangepaste `DateChooser` die binnen PTM gebruikt wordt voor het weergeven van dagen waarop gewerkt zijn of feestdagen. Bovenstaande methode is verantwoordelijk voor de aangepaste rendering van een geselecteerde datum in de `PtmDateChooser`. Aan het einde van die methode wordt de methode `validateNow()` aangeroepen om er voor te zorgen dat het vetgedrukt maken van de geselecteerde datum ook weergegeven wordt.

De enige aanpassing die nodig was om gebruik te maken van deze methode is aan de klasse `PtmDateChooser` “use namespace `mx_internal`” toe te voegen. Alle als `mx_internal` gemarkeerde methodes en variabelen zijn dan benaderbaar.

In bovenstaand voorbeeld zijn deze methodes ook gemarkeerd met “`this.mx_internal::`” om aan te geven waar een dergelijk object gebruikt is. Tevens is dit ook een alternatieve methode om dergelijke methodes en variabelen te gebruiken.

Verder bleek ook tijdens het ontwikkelen met Flex dat deze niet op alle vlakken goed uitontwikkeld is. Dit bleek uit het feit dat verschillende keren code anders werkte of op een ander manier gebruikt moest worden dan omschreven in de documentatie van Flex. Dit leverde vertraging en problemen op met het uitzoeken waarom functionaliteit niet of anders werkte dan verwacht.

Een van de noemenswaardige problemen hierbij was het vinden van een bug in de compiler van Flex. Flex biedt de mogelijkheid om get en set methodes voor de programmeur er uit te laten zijn als publiek beschikbare variabelen. Dus in plaats van `persoon.setNaam(“Collin”)` kan een dergelijke methode gebruikt worden als `persoon.naam = “Collin”`. Voor het setten en getten van de naam van een persoon wordt dezelfde functienaam gebruikt, maar voor het getten of setten zijn wel twee apart geïmplementeerde methodes gedefinieerd. De compilerbug komt naar voren als de getter in een andere namespace, of vice

versa, dan geeft de compiler de foutmelding “ambiguous reference”. Dit betekent dat er tijdens het overriden van een bestaande functie in de superklasse een fout is gemaakt en de compiler geen werkende Flex applicatie kan compileren. Toen dit zich voordeed werd er echter geen gebruik gemaakt van overridding van een methode in de superklasse. Hier is wel een workaround voor te schrijven maar het heeft enig zoekwerk gevergd naar de oorzaak van deze compiler foutmelding.

Resultaat & Evaluatie

Aan het einde van de afstudeerperiode is het niet gelukt om een volledig werkende PTM versie op te leveren. Wel is er een goede basis opgeleverd voor het verder ontwikkelen van de PTM die voldoet aan de gestelde eisen van onderhoudbaarheid en uitbreidbaarheid.

Tevens heeft het werken met Flex voor de PTM applicatie een schat aan informatie opgeleverd over de werking van Flex en hoe componenten uitgebreid kunnen worden.

10.1.4 Bouwfase 2

Deze fase omvatte het herontwerpen en bouwen van al bestaande rapportage functionaliteiten in de huidige PTM versie en het ontwerpen en bouwen van de nieuwe rapportages, met behulp van de geselecteerde rapportage en charting API, voor de nieuwe PTM versie.

Door de opgelopen vertragingen en de veranderingen aan het ontwerp van de nieuwe PTM is deze fase uitgesteld tot een later moment. Doordat Flex gebruikt werd gebruikt voor de gebruikersinterface kan de geselecteerde rapportage en charting API niet direct of eenvoudig gebruikt worden.

Voor de nieuwe opzet in Flex van PTM is het waarschijnlijk beter om gebruik te maken van de Flex Charting om de benodigde rapportages te genereren. De voordelen van het gebruiken van de Flex charting in de nieuwe situatie zijn onder andere:

- Eenvoudiger te integreren in de huidige Flex Applicatie dan een op Java gebaseerde rapportage en charting API.
- De load van het genereren van de rapportages en charts wordt gedeeld door zowel de server en de client. De gegevens worden op de server verzameld en klaargemaakt. De client voert vervolgens de grafische weergave en opmaak uit.

Tijdens de verdere ontwikkeling van PTM zal hier een besluit over genomen worden, tijdens het schrijven van dit verslag was hier nog geen informatie over beschikbaar.

10.1.5 Testfase

Door tijdgebrek is een duidelijk en goed gedocumenteerde testfase niet uitgevoerd. Tijdens het implementeren van de functionaliteit van de vernieuwde PTM is er voor gekozen om deze zo goed mogelijk te testen. Een van de maatregelen was dat mijn bedrijfsbegeleider enkele momenten heeft gehad waarin hij het programma doornam en besproken heeft. Dit leverde een lijst van verbeterpunten op die vervolgens meegenomen zijn in de verdere ontwikkeling. Tevens heeft een andere afstudeerder, Maarten Daalder, de code getest waardoor enkele fouten zijn gevonden en hersteld.

10.2 Proces & Versiebeheer

In het begin van de afstudeerstage is er gewerkt aan verschillende kleine test projecten voor het verkrijgen van bekendheid met de ontwikkelomgeving en de te gebruiken technieken. Dit waren voornamelijk kleine testcases voor functionaliteit zoals test EJB entitybeans en het gebruik hiervan.

Voor het bijhouden van een urenregistratie is gebruik gemaakt van de binnen ISAAC Software Solutions beschikbare PTM versie. Hierin zijn de stageuren en uren voor extra taken, zoals kleine aanpassingen aan websites van klanten, bijgehouden.

Voor het bekend zijn met alle mogelijkheden binnen de software en hoe deze geïmplementeerd in de gebruikers interface beschikte de afstudeerder over alle rechten binnen PTM. Dit was behulpzaam bij het opzetten van de vernieuwde gebruikersinterface van PTM.

Als versiebeheersysteem voor de software is gebruik gemaakt van Subversion die standaard binnen ISAAC Software Solutions gebruikt wordt. Subversion is een opensource version control system en geeft een ontwikkelaar de mogelijkheid om meerdere versies van een software pakket bij te houden en een oudere versie terug te zetten of op te vragen voor inzage.

Binnen het project is dit gebruikt voor de nieuwe PTM applicatie en voor een groter prototype.

10.3 Conclusies software ontwikkeling

EJB 3.0 en met name diverse daaromheen opgebouwde technieken uit de LoyaltySuite, zijn zeer goed te gebruiken in andere projecten, doordat ze veel werk besparen tijdens het uitbreiden, onderhouden en opzetten van code.

Als wij kijken naar de gebruikte techniek voor de frontend, Flex, is dit ook zeer aan te raden voor het maken van zeer interactieve, mooi ogende en gebruikersvriendelijke applicaties. Er moet tijdens de ontwikkeling wel zeer goed rekening gehouden worden dat Flex nog niet helemaal volwassen is. Dit is te merken aan diverse fouten binnen Flex waardoor code niet werkt zoals omschreven in de documentatie of helemaal niet werkt. Ik raad dan ook aan om hiervoor extra tijd uit te trekken omdat dit een van de redenen was dat er uitloop was tijdens het ontwikkelen van PTM.

Dit wil niet zeggen dat Flex een slechte oplossing is, deze techniek is zeer veelbelovend en in de huidige vorm al zeer krachtig. Daarom kan ik deze oplossing dan ook aanraden waar een mooie en goed te gebruiken gebruikersinterface nodig is.

IV

Afronding

Dit onderdeel van de afstudeerscriptie behandelt de afronding van de afstudeerstage. Ten eerste wordt de afronding van het project behandeld met daaropvolgend conclusies en aanbevelingen. Ten tweede wordt een evaluatie gegeven van de gehele afstudeerperiode.

11 Conclusies en aanbevelingen

In dit hoofdstuk wordt een eindconclusie en aanbeveling gegeven met betrekking tot het volledige afstudeerproject. Deze zijn uitgesplitst in de onderdelen onderzoek en software ontwikkeling en heeft u al kunnen lezen in de conclusies van elk deel. Voor het gemak zijn deze hier nogmaals vermeld.

11.1 *Onderzoek*

Alle van de onderzochte rapportage en charting API's zijn uitgebreid genoeg om te voorzien in een breed scala aan rapportages. Qua grafieken is de uitzondering JasperReports die het minst aantal soorten grafieken ondersteunt waardoor bepaalde rapportages niet begeleid kunnen worden met een verduidelijkende grafiek.

De keuze voor een rapportage en charting API wordt daarom voornamelijk gebaseerd op beschikbare documentatie en hoe gebruikersvriendelijk een rapportage en charting API is. Op dit gebied is van de onderzochte rapportage en charting API's BIRT de beste keuze door een combinatie van de volgende factoren.

Van de onderzochte rapportage en charting API biedt BIRT een gebruikersvriendelijke interface voor het ontwerpen. Met deze interface kan een nieuwe programmeur vrij snel standaard rapportages ontwikkelen en zijn zelfs gevorderde functionaliteiten vrij snel te vinden en eenvoudig te gebruiken.

De documentatie is niet de beste van de onderzochte rapportage en charting API's maar komt zeer dicht in de buurt van de kwaliteit van JasperReports. In combinatie met een levendige community op het internet is er veel informatie hierover beschikbaar.

De integratie van BIRT in een applicatie is redelijk complex door de hoeveelheid configuratiewerk die verzet moet worden. De enige uitzondering hierop is Crystal Reports die zeer eenvoudig te installeren en te gebruiken is binnen een applicatie (wel met enig inventief gebruik van de Crystal Reports plugin door ontbrekende documentatie). Eenmaal geïntegreerd is de documentatie van BIRT zeer goed en is het gebruik van de rapportages ook eenvoudig.

Het oorspronkelijke advies van het rapport is daarom gebruik te maken van BIRT voor het toevoegen van uitgebreide rapportage mogelijkheden in Java projecten.

Sinds het schrijven van het onderzoek is door MyEclipse een aangepaste versie gelanceerd van BIRT, genaamd MyEclipse Reports. Meestal zijn de producten van MyEclipse eenvoudig te integreren. Het is daarom de moeite waard om te kijken of MyEclipse interessant is om te gebruiken.

11.2 *Software ontwikkeling*

EJB 3.0 en met name diverse daaromheen opgebouwde technieken uit de LoyaltySuite, zijn zeer goed te gebruiken in andere projecten, doordat ze veel werk besparen tijdens het uitbreiden, onderhouden en opzetten van code.

Als wij kijken naar de gebruikte techniek voor de frontend, Flex, is dit ook zeer aan te raden voor het maken van zeer interactieve, mooi ogende en gebruikersvriendelijke applicaties. Er moet tijdens de ontwikkeling wel zeer goed rekening gehouden worden dat Flex nog niet helemaal volwassen is. Dit is te merken aan diverse fouten binnen Flex waardoor code niet werkt zoals omschreven in de documentatie of helemaal niet werkt. Ik raad dan ook aan om hiervoor extra tijd uit te trekken omdat dit een van de redenen was dat er uitloop was tijdens het ontwikkelen van PTM.

Dit wil niet zeggen dat Flex een slechte oplossing is, deze techniek is zeer veelbelovend en in de huidige vorm al zeer krachtig. Daarom kan ik deze oplossing dan ook aanraden waar een mooie en goed te gebruiken gebruikersinterface nodig is.

12 Evaluatie

In dit hoofdstuk wordt mijn functioneren tijdens de afstudeerperiode kritisch geëvalueerd en beschreven. Verder wordt aandacht besteed aan de mate van het bereiken van de vooraf gestelde doelen en de toepassing van de tijdens de opleiding opgedane kennis en vaardigheden. Tot slot wordt gekeken naar de opgedane kennis en vaardigheden tijdens de afstudeerperiode.

12.1 Zelfreflectie

Kritisch gekeken naar mijn functioneren tijdens de afgelopen afstudeerperiode zijn er een aantal positieve en negatieve punten op te noemen.

De manier van planning en het bijhouden van de planning is minder goed gelukt dan verwacht. De aanpak werkte goed binnen de opleiding met de kleinere projecten, maar voor een afstudeerperiode was deze niet strak genoeg en is er daardoor op bepaalde punten uitloop ontstaan. Deze vertragingen werden grotendeels ook veroorzaakt door het leren omgaan met Flex en de vele nukken van Flex.

Tijdens de afstudeerstage was er sprake van een hoge mate van zelfredzaamheid qua werken en het oplossen van problemen. Waar mogelijk is gebruik gemaakt van kennis van collega's. Was dit niet mogelijk dan waren oplossingen zelfstandig ook goed te vinden.

De stage is op een positieve manier afgesloten en wordt vervolgt door een baan bij ISAAC Software Solutions.

12.2 Doelen

Het belangrijkste doel, losstaand van het project, was het opdoen van zoveel mogelijk nieuwe kennis en ervaring. Met alle nieuwe geleerde technieken en alle verder opgedane ervaringen en indrukken tijdens de afstudeerperiode is dit doel volledig geslaagd. Het doel van de afstudeerproject is als volgt gedefinieerd:

“De door ISAAC gebruikte core-functionaliteit van PTM dient opnieuw te worden ontwikkeld met gebruik van de huidige technieken, met als doel het programma eenvoudiger uitbreidbaar, efficiënter en meer gebruikersvriendelijk op te zetten. De gegevens die in PTM opgeslagen zijn dienen eenvoudig opvraagbaar en inzichtelijk te zijn met behulp van rapportages en grafieken.”

Dit doel is niet volledig behaald. Van de afgesproken functionaliteit moet nog een groot deel geïmplementeerd worden. Het geïmplementeerde deel van de afgesproken functionaliteiten is wel beter te onderhouden, eenvoudiger uit te breiden en eenvoudiger te gebruiken voor programmeurs en eindgebruikers van PTM.

Het toevoegen van de rapportages in de vernieuwde PTM is niet uitgevoerd doordat dit gebaseerd was op het uitgangspunt dat de vernieuwde PTM in een vorm van JSP pagina's met AJAX opgezet zou worden. Doordat er gebruik is gemaakt van Flex, en door tijdgebruik, kon de gekozen rapportage en charting API niet geïmplementeerd worden.

Het project heeft wel veel nieuwe kennis en ervaring met Flex opgeleverd binnen ISAAC Software Solutions. Deze kennis wordt nu al gebruikt voor het ontwikkelen van nieuwe Flex applicaties. Tevens is de basis van de nieuwe PTM op een dergelijke manier opgezet dat deze eenvoudig verder uitbreidbaar is.

Ook al is de oorspronkelijke doelstelling niet volledig gehaald, is het project zelf wel een succes dankzij alle opgeleverde kennis en de reeds ontwikkelde solide basis voor het verder ontwikkelen van PTM.

13 Nawoord

Nu het einde van deze afstudeerperiode steeds dichterbij komt kan ik, ondanks de vele tegengekomen problemen, terugkijken op een redelijk succesvolle periode. Ik heb erg veel geleerd in de tijd die ik bij ISAAC Software Solutions heb doorgebracht. Ook kon ik veel gebruik maken van de in mijn opleiding opgedane kennis. Het werken aan een dergelijk project als een afstudeerproject is een groot verschil met de redelijk kleine projecten die ik uitgevoerd heb tijdens mijn opleiding. Dit was een goede leerschool.

Hoewel de nieuwe PTM versie nog niet alle benodigde functionaliteit heeft en een enkele bug in bestaande code nog gecorrigeerd moet worden, is de nieuwe versie qua gebruikersgemak een grote stap voorwaarts in vergelijking met de vorige versie. Tevens is de nieuwe code beter te onderhouden en makkelijker te begrijpen voor nieuwe programmeurs.

Na mijn afstuderen start ik met mijn nieuwe baan bij ISAAC Software Solutions waarin ik deze versie van PTM verder zal ontwikkelen.

Binnen deze versie van PTM zijn voor de klassen, interfaces, methodes, etc, zo eenduidig mogelijke namen gebruikt. Zo is voor andere programmeurs de codebase eenvoudig te gebruiken. Tevens is binnen de applicatie zoveel mogelijk documentatie geschreven (jammer genoeg is deze op het moment van schrijven nog niet compleet) en waar nodig zijn waarschuwingen en verdere uitleg gegeven als er ‘speciale’ code nodig is voor het bereiken van gewenst gedrag.

Op het moment van schrijven is een nieuwe afstudeerder reed gestart met het schrijven en integreren van code zodat het mogelijk wordt om een zogenaamde data push uit te voeren. Met de huidige versie is het alleen mogelijk om informatie op te vragen in de client. Wijzigingen opgeslagen door een andere client ziet men nog niet gelijk terug in alle andere clients. Het gebruik van deze toevoeging aan PTM zal veel nieuwe functionaliteit mogelijk maken en het programma nog robuuster en foutongevoeliger maken.

14 Literatuurlijst

In dit hoofdstuk wordt een overzicht gegeven van de gebruikte informatiebronnen tijdens de afstudeerperiode.

Titel	Auteur / website	Uitgave
Enterprise JavaBeans 3.0	Bill Burke & Richard Monson-Haefel	2006
Core J2EE Patterns	Deepak Alur, John Crupi & Dan Malks	2003
Het ontwerpen van een onderzoek	Piet Verschuren & Hans Doorewaard	2005
Rapport over rapporteren	Wim Hoogland & Roel Dik	vierde druk 2002
Refactoring to Patterns	Joshua Kerievsky	2004

V

Bijlagen


Het laatste onderdeel van het afstudeerverslag bevat de bijlagen die genoemd zijn in de scriptie of die belangrijk zijn als naslagwerk.

Als bijlage treft u onder andere aan het plan van aanpak, het onderzoeksrapport en het document van eisen.

Bijlage 1:

Plan van Aanpak

Plan van Aanpak PTM

Auteur:	Collin Maessen	 ISAAC
Datum:	12 november 2007	
Versie / status:	1.3 / Definitief	
File:	PvA PTM - 20071112 - v1.3.doc	

Versiebeheer

Datum	Versie	Belangrijkste wijzigingen	Auteur
05-09-2007	0.1	Eerste opzet.	Collin Maessen
06-09-2007	0.2	Verklarende woordenlijst toegevoegd, enkele tekstuele aanpassingen.	Collin Maessen
11-09-2007	0.3	Feedback van bedrijfsbegeleider verwerkt	Collin Maessen
02-10-2007	1.0	Feedback van docentbegeleider en bedrijfsbegeleider verwerkt en planning aangepast.	Collin Maessen
17-10-2007	1.1	Aangepast aan de hand van feedback van het maken van het Onderzoeksaanpak Plan	Collin Maessen
01-11-2007	1.2	Planning aangepast aan de hand van de resultaten van het Document van Eisen	Collin Maessen
12-11-2007	1.3	Kleine tekstuele aanpassingen	Collin Maessen

Voorwoord

Voor de afronding van mijn opleiding Informatie bij Fontys Hogescholen voer ik een afstudeeropdracht uit bij ISAAC Software Solutions BV. Voor deze afstudeeropdracht heb ik het Plan van Aanpak geschreven dat voor u ligt.

In dit document is de opzet en gebruikte aanpak van mijn afstudeerproject beschreven.. Dit document zal tijdens het uitvoeren van het project aangepast worden als er wijzigingen zijn in het project.

Collin Maessen
Eindhoven, september 2007

Inhoudsopgave

SAMENVATTING.....	52
1 INLEIDING	55
1.1 DOELGROEP.....	55
2 BEDRIJFSPROFIEL	56
3 PROJECTBESCHRIJVING	57
3.1 BEGINSITUATIE MET UITGANGSPUNTEN.....	57
3.2 PROBLEEMSTELLING.....	57
3.3 PROJECTDOELSTELLINGEN	58
3.4 FORMULERING PROJECTRESULTAAT	58
3.5 WAT BEHOORT WEL TOT HET PROJECTRESULTAAT	58
3.6 WAT BEHOORT NIET TOT HET PROJECTRESULTAAT	58
3.7 PROJECTRANDVOORWAARDEN	59
3.8 PROJECTRISICO'S.....	59
3.9 RISICOBEPERKENDE MAATREGELEN	59
4 PROJECTFASERING.....	60
4.1 INITIËLE FASE	60
4.2 ONDERZOEK- EN DEFINITIEFASE.....	60
4.3 ONTWERPFASE	61
4.4 BOUWFASE 1	61
4.4.1 Increment 1	62
4.4.2 Increment 2	62
4.4.3 Increment 3	62
4.4.4 Increment 4	63
4.4.5 Increment 5	63
4.4.6 Increment 6	63
4.4.7 Increment 7 (optioneel)	64
4.4.8 Increment 8 (optioneel)	64
4.5 BOUWFASE 2	64
4.6 TESTFASE	64
4.7 RAPORTEER & AFRONDINGSFASE	65
5 BEHEERSASPECTEN	66
5.1 BEHEERSASPECT INFORMATIE	66
5.1.1 Bronnen.....	66
5.1.2 Gebruik en rapportage.....	66
5.2 BEHEERSASPECT TIJD	66
5.2.1 Globale planning.....	66
5.2.2 Registratie, tijdverantwoording en bewaking.....	67
5.3 BEHEERSASPECT KWALITEIT	67
5.3.1 Criteria.....	67
5.3.2 Registratie	67
5.3.3 Controle	67
BIJLAGE 1: PROJECTPLANNING	68

Samenvatting

In dit document wordt zowel de opzet als de gebruikte aanpak binnen het afstudeerproject beschreven.

Projectbeschrijving

ISAAC gebruikt voor het registreren van projecturen een in eigen huis ontwikkelde applicatie, het zogenaamde Project Time Management (PTM). De huidige versie is technologisch verouderd en sluit daardoor niet meer aan binnen de veranderde processen binnen ISAAC en de taken waarvoor PTM gebruikt wordt.

Tijdens het project wordt onderzocht welke functionaliteiten tot de core-functionaliteit behoren van PTM. Deze core-functionaliteiten zullen opnieuw geïmplementeerd worden zodat PTM weer voldoet aan de huidige stand van de techniek.

Een tweede onderdeel van het project is het inzichtelijk maken van de gegevens die in PTM aanwezig zijn. Er zal onderzocht worden welke gegevens gewenst zijn bij het management en de werknemers en welke weergavemethode en weergavetechniek hiervoor het meest geschikt is.

Projectfasering

Het project is opgedeeld in de volgende fases:

<i>Initiële fase</i>	In deze fase wordt er kennis gemaakt met het bedrijf en de collega's en zal aan de hand van ingewonnen informatie een Plan van Aanpak opgesteld worden.
<i>Onderzoek- en definitiefase</i>	In de onderzoek- en definitiefase worden de functionaliteitwensen en informatiewensen in kaart gebracht. En wordt er onderzocht welke informatiewensen er zijn, welke weergave techniek en weergave oplossing het beste geschikt is voor PTM.
<i>Ontwerpfase</i>	In de Ontwerpfase worden de specificaties voor het programma omgezet naar een blauwdruk.
<i>Bouwfase 1</i>	In bouwfase 1 wordt PTM opnieuw ontwikkeld en getest.
<i>Bouwfase 2</i>	In bouwfase 2 wordt het weergeven van rapportages en grafieken ontwikkeld en geïntegreerd in PTM.
<i>Testfase</i>	In de Testfase wordt de software compleet getest en gevonden fouten worden gecorrigeerd.
<i>Rapporteur & afrondingsfase</i>	In de rapporteer- en afrondingsfase worden de uitgevoerde taken in een procesverslag vastgelegd en wordt de eindpresentatie voorbereid.

Beheersaspecten

Om tot een succesvolle afronding van het project te komen, wordt er gebruik gemaakt van beheersaspecten. Deze beheersaspecten dienen voor het garanderen van de kwaliteit van werken, gemaakte documenten en ontwikkelde software. Tevens zorgen de beheersaspecten ervoor dat tijdens het project de projectplanning gevolgd wordt en op tijd ingegrepen wordt als de ingeplande tijd overschreden dreigt te worden.

Verklarende woordenlijst

Acceptatie Test Plan	Het Acceptatie Test Plan beschrijft hoe en welke functionaliteiten van een computerprogramma getest worden.
Broncode	De broncode van een computerprogramma is de door mensen leesbare omschrijving hoe een computerprogramma moet werken.
Codestandaard	Een codestandaard beschrijft onder andere hoe een programmeur zijn broncode moet opmaken en aan welke naamconventies hij zich moet houden.
CSS	Cascading Style Sheets (CSS) is een techniek die gebruikt wordt voor de vormgeving van webpagina's.
Databasemodel	Het databasemodel is het ontwerp van een database.
Document van Eisen	In het document van eisen staan de uitgewerkte eisen van de opdrachtgever waaraan een computerprogramma moet voldoen.
Domeinmodel	Het domeinmodel is een grafische weergave van de interne werking van een computerprogramma.
GUI	GUI staat voor Graphical User Interface, dit is het gedeelte van een computerprogramma dat een gebruiker krijgt te zien en bedient.
HTML	HyperText Markup Language (HTML) is een taal die gebruikt wordt voor de structuur van een webpagina te beschrijven.
Iteratie	Een iteratie wordt gebruikt om de ontwikkeling van een computerprogramma op te delen in kleine overzichtelijke stukken.
J2SE	J2SE staat voor Java Platform 2, Standard Edition. J2SE is een platform dat een volledige omgeving voor het ontwikkelen en draaien van applicaties op desktop systemen en servers biedt.
Java	Java is een programmeertaal waarmee computerprogramma's ontwikkeld kunnen worden.
JavaScript	JavaScript is een door de browser geïnterpreteerde programmeertaal. Deze taal wordt voornamelijk gebruikt op het wereldwijde web. Deze taal is niet een variant van de Java programmeertaal.
JavaBeans	Een JavaBean is een softwarecomponent dat bepaalde afspraken volgt. Door het volgen van deze afspraken en het gebruikmaken van de faciliteiten van de JavaBeans API, worden JavaBeans tot herbruikbare componenten die configureerbaar zijn op de plaats waar ze uiteindelijk toegepast worden, gemanipuleerd kunnen worden met behulp van softwaretools en die gekoppeld kunnen worden aan andere stukken software (JavaBeans, andere componenten of clientsoftware).
Java Server Pages	Java Server Pages (JSP) is een manier om dynamisch HTML, XML of andere inhoud te genereren op basis van ingevoerde gegevens of instellingen.
MoSCoW-methode	Met de MoSCoW-methode wordt met Must have, Should have, Could have en Won't have aangegeven welke functionaliteit in een computerprogramma geïmplementeerd worden en hoe belangrijk de functionaliteit is.

Prototype	Met een prototype wordt de mogelijkheden en onmogelijkheden van een ontwerp of techniek onderzocht.
Scenario	Scenario's zijn uitgeschreven stappen/handelingen die gebruikt om een programma te testen op een correcte werking.
Software bibliotheek	Een software bibliotheek is een verzameling functionaliteit die gebruikt kunnen worden in een computerprogramma.
Unit test	Een Unit test wordt gebruikt om functionaliteit van een computerprogramma geautomatiseerd te testen.

1 Inleiding

Onderliggend document dient als plan van aanpak voor het project. Het beschrijft de projectuitvoering in resultaat en manier van werken, en beschrijft de randvoorwaarden waaronder dit moet plaatsvinden.

Voor een succesvol project dient er in de beginfase van het project overeenstemming te zijn over het plan van aanpak. Alle partijen moeten het eens zijn over de opdracht, de aanpak, hun taken, de risico's inclusief de risicobeperkende maatregelen.

1.1 Doelgroep

Dit document is bestemd voor de bedrijfsbegeleider, de docentbegeleider en de afstudeerder.

Functie	Naam
Bedrijfsbegeleider	Valentijn Scholten
Docentbegeleider	Paul Linnartz
Afstudeerder	Collin Maessen

2 Bedrijfsprofiel

ISAAC is een jong en groeiend bedrijf dat zich richt op de nieuwste ontwikkelingen op het internet. Hiervoor creëren de mensen bij ISAAC middelen voor een effectieve internetcommunicatie. Het bedrijf is gevestigd in Eindhoven en bestaat uit twee afdelingen, genaamd ISAAC Software Solutions en ISAAC Web Solutions.



ISAAC bestaat momenteel 8 jaar en is ontstaan op de Technische Universiteit Eindhoven (TU/e). Gezien deze achtergrond hebben de medewerkers van ISAAC een sterke affiniteit met techniek.

Het bedrijf telt momenteel ongeveer 25 medewerkers en beschikt over deskundige consultants, ervaren programmeerteams, deskundige webdesigners en een sterke Research & Development afdeling. Hierdoor is ISAAC als geen ander in staat bedrijven optimaal te begeleiden, en een complex proces om te zetten tot eenvoudig te gebruiken en praktische softwareoplossingen.

3 Projectbeschrijving

In de projectbeschrijving wordt beschreven wat de beginsituatie is van het project en wat het project dient te bereiken. Ook wordt verduidelijkt wat wel en niet tot het projectresultaat behoort en welke producten aan het einde worden opgeleverd.

Daarnaast wordt beschreven aan welke randvoorwaarden voldaan moeten worden zodat het project tot een succesvol einde gebracht kan worden, welke risico's er zijn en hoe deze afgedekt worden.

3.1 *Beginsituatie met uitgangspunten*

Project Time Management (PTM) is een webbased projectmanagement systeem. Voor ISAAC worden in PTM basisklantgegevens bijgehouden, projecten en uren die werknemers maken voor de projecten. Door het bijhouden van deze gegevens in PTM kunnen diverse rapporten gegenereerd worden voor het inzichtelijk maken van de voortgang van projecten, behaalde rendementen van projecten en behaalde rendementen van werknemers. Ook worden deze gegevens gebruikt voor het factureren van klanten.

Voor klanten kan PTM in verschillende versies geleverd worden die toegespitst zijn op de taken urenregistratie, projectregistratie of taakregistratie.

PTM is geïmplementeerd als een Java Webservice en maakt gebruik van JavaBeans en Java Server Pages. De JavaBeans zijn als normale J2SE beans geïmplementeerd. Voor de gegevens opslag maakt PTM gebruik van Microsoft SQL server.

Door de implementatie als Java Webservice is PTM via een browser benaderbaar waardoor er geen installatie vereist is op de pc van de gebruiker. Voor de weergave in de browser wordt gebruik gemaakt van HTML en CSS met een beperkt gebruik van JavaScript.

3.2 *Probleemstelling*

PTM werkt binnen ISAAC goed voor het bijhouden van klantgegevens, projecten en projecturen. Echter is het systeem technologisch verouderd waardoor de opgeslagen gegevens in PTM niet ten volle benut kunnen worden.

Zo moeten rapporten vaak nog buiten PTM bewerkt worden om de daarin weergegeven gegevens inzichtelijk te maken met behulp van berekeningen en grafieken. Dit maakt het inzichtelijk maken van de gegevens in PTM onnodig bewerkelijk.

Ook is het systeem met de huidige Web 2.0 technieken onnodig gebruikersonvriendelijk. Met Web 2.0 technieken kan voorkomen worden dat gebruikers vaak tussen verschillende pagina's moeten wisselen. Dit kan door bijvoorbeeld invoer en bewerktfunctionaliteiten op de pagina zelf aan te bieden zodra de gebruiker nieuwe gegevens wil invoeren of oude gegevens wil bewerken.

Ook is de achterliggende code al enkele jaren oud waardoor verschillende programmeurs aan PTM hebben gewerkt voor het toevoegen van nieuwe functionaliteiten.

De probleemstelling is daarom dan ook als volgt gedefinieerd:

“De huidige versie van PTM is technologisch verouderd en daardoor niet eenvoudig uitbreidbaar en onderhoudbaar. Verder is PTM door het gebruik van verouderde technieken, in vergelijking met de huidige technieken, gebruikersonvriendelijk. Daarnaast is het maken van rapportages met behulp van de gegevens in PTM onnodig bewerkelijk en daardoor tijdsconsumerend. Ook zijn de mogelijkheden tot het soort rapportages dat PTM genereert te beperkt.”

3.3 Projectdoelstellingen

De doelstelling is met behulp van de probleemstelling als volgt te formuleren:

“De door ISAAC gebruikte core-functionaliteit van PTM dient opnieuw te worden ontwikkeld met gebruik van de huidige technieken, met als doel het programma eenvoudiger uitbreidbaar, efficiënter en meer gebruikersvriendelijk op te zetten. De gegevens die in PTM opgeslagen zijn dienen eenvoudig opvraagbaar en inzichtelijk te zijn met behulp van rapportages en grafieken.”

Deze algemene doelstelling is op te splitsen in 2 doelen voor het project:

- D1: Opleveren van een vernieuwd en herontworpen PTM waarin de voor ISAAC belangrijke core-functionaliteiten geïmplementeerd zijn.
- D2: Het inzichtelijk maken van de gegevens in PTM met behulp van rapportages en grafieken die de voor ISAAC belangrijke gegevens inzichtelijk maken.

3.4 Formulering projectresultaat

Aan het einde van het project zijn de volgende producten opgeleverd:

- Plan van Aanpak
- Onderzoeksrapport
- Document van Eisen
- Domeinmodel
- Databasemodel
- Acceptatie Test Plan
- Vernieuwde PTM-software
- Rapportage en charting module
- Gebruikershandleiding

Het precieze aantal te doorlopen iteraties tijdens de ontwikkeling van de software is op het moment van schrijven van het Plan van Aanpak nog niet bekend. Daarom zullen deze iteraties op het moment ook niet benoemd worden in de planning.

3.5 Wat behoort WEL tot het projectresultaat

Binnen de scope van dit project vallen:

- Het opleveren van een compleet vernieuwd en werkend PTM
- Een werkend en compleet rapportage en charting module

3.6 Wat behoort NIET tot het projectresultaat

Buiten de scope van dit project vallen:

- Het implementeren van de ontwikkelde software in een productieomgeving
- Onderhoud van de ontwikkelde software

3.7 Projectrandvoorwaarden

Aan de volgende projectrandvoorwaarden dienen voldaan te worden voor een succesvolle afronding van het project:

- Binnen ISAAC dient er te allen tijde een geschikte werkplek beschikbaar te zijn voor de student.
- ISAAC dient er voor te zorgen dat de afstudeerder voldoende begeleiding krijgt tijdens zijn afstudeerperiode.
- Er dient door Fontys een docentbegeleider beschikbaar te zijn gesteld voor de student.

3.8 Projectrisico's

- R1: Bij de student is er een beperkte kennis aanwezig over de gebruikte technieken bij ISAAC. Dit kan voor een onderschatting van de complexiteit van de software zorgen en uitloop in de implementatie veroorzaken.
- R2: De student is beperkt in zijn inzet en energie door een chronische ziekte. Dit kan er voor zorgen dat de student tijdens het uitvoeren van het project overbelast raakt en uitvalt.
- R3: Door het redelijk aantal belanghebbende die geraadpleegd moeten worden voor het ontwikkelen van de nieuwe software kunnen er tegenstrijdigheden en/of onduidelijkheden ontstaan. Dit ten opzichte van de gebruikerswensen, hoe deze geïmplementeerd dienen te worden en hoe belangrijk de diverse functionaliteiten zijn. Dit kan voor vertragingen in de ontwikkeling zorgen doordat gedeeltes opnieuw geïmplementeerd moeten worden om te voldoen aan de gestelde eisen.

3.9 Risicobeperkende maatregelen

- R1 Bij de planning van het project zal de benodigde tijd ruim ingeschat worden. Hierdoor is er tijd beschikbaar voor het eigen maken van kennis en voor het afvangen van vertragingen door problemen.
Tevens zal er zorg voor gedragen worden dat tijdig gebruik wordt gemaakt van de aanwezige kennisexpertise binnen ISAAC en daarbuiten. Tevens zal regelmatig de status/voortgang van het project besproken worden met de bedrijfsbegeleider.
- R2 Om het risico van overbelasting en uitval af te dekken is er contact met het UWV opgenomen om begeleiding tijdens het afstuderen te regelen. Dit omdat er bij de student en binnen het bedrijf de benodigde kennis en ervaring ontbreken voor een gedegen begeleiding op dat gebied.
Binnen het bedrijf zal tijdig contact opgenomen worden met de bedrijfsbegeleider als er overbelasting en daaropvolgend uitval dreigt zodat er tijdig ingegrepen kan worden.
- R3 De gebruikerswensen en andere functionaliteiten worden zo duidelijk mogelijk vastgelegd en teruggekoppeld aan de diverse belanghebbenden. Dit zal uiteindelijk een Document van Eisen opleveren waarin met behulp van de MoSCoW methode aangegeven wordt hoe belangrijk de diverse functionaliteiten zijn.
Met verschillende gesprekken en terugkoppelingsmomenten naar de bedrijfsbegeleider, en eventueel naar belanghebbende, zal bewaakt worden dat het programma voldoet aan de gestelde eisen.

4 Projectfasering

In de projectfasering wordt beschreven welke fases er doorlopen worden tijdens het project. Per fase wordt aangegeven wat het doel is, wat de op te leveren resultaten zijn en welke activiteiten er ondernomen moeten worden.

De fases die doorlopen worden zijn:

- Initiële fase.
- Onderzoek- en definitiefase.
- Ontwerpfase.
- Bouwfase 1.
- Bouwfase 2.
- Testfase.
- Rapporteer & afrondingsfase.

Het precieze aantal te doorlopen iteraties tijdens de ontwikkeling van de software is op het moment van schrijven van het Plan van Aanpak nog niet bekend. Daarom zullen deze iteraties op het moment ook niet benoemd worden in de planning.

4.1 *Initiële fase*

Doel van de initiële fase is het bekend raken met het bedrijf, de werknemers en de probleemstelling. Aan de hand van de ingewonnen gegevens wordt een Plan van Aanpak opgesteld.

Resultaten:

- Plan van Aanpak.
- Definitieve Opdrachtomschrijving.
- Communicatieplan.

Activiteiten:

- Kennismaking met collega's en het bedrijf.
- Werkplek inrichten.
- Inlezen in relevante documentatie en informatie.
- Schrijven van het Plan van Aanpak.
- Schrijven van de Definitieve Opdrachtomschrijving.

4.2 *Onderzoek- en definitiefase*

Doel van het onderzoek- en definitiefase is het in kaart brengen van de core-functionaliteit van PTM en de informatiewensen van het management en de gebruikers.

Voor het onderzoeksrapport wordt gebruik gemaakt van een aangepaste variant van de methode beschreven door Piet verschuren en Hans Doorewaard.

Resultaten:

- Document van Eisen.
- Onderzoeksplan
- Onderzoeksrapport.

Activiteiten:

- Management en key-gebruikers van PTM interviewen om gebruikerswensen te achterhalen.
- Overleg met management welke wensen daadwerkelijk geïmplementeerd worden en hoe belangrijk wensen zijn, hiervoor zal de MoSCoW methode gebruikt worden.
- Onderzoek naar welke statistische weergave het meest geschikt is voor de informatiewensen van het management.
- Onderzoek naar welke methode/software bibliotheken het meest geschikt zijn voor het grafisch weergeven van de gewenste statistieken.

4.3 Ontwerpfase

Doel van de Ontwerpfase is het omzetten van de specificaties naar een domeinmodel en databasemodel. Aan de hand van deze modellen wordt de software ontwikkeld.

Resultaten:

- Domeinmodel.
- Databasemodel.
- Prototype (eventueel).

Activiteiten:

- De eisen van het management en de gebruikers die beschreven staan in het Document van Eisen worden omgezet naar een domeinmodel en databasemodel
- Als er onduidelijkheden zijn in hoe de software geïmplementeerd moet worden zal er een prototype ontwikkeld worden die dit verduidelijkt.

4.4 Bouwfase 1

In bouwfase 1 wordt PTM opnieuw ontwikkeld en getest.

Het implementeren van de use cases uit het Document van Eisen verloopt in 8 increments. In increments 1 tot en met 6 worden de use cases geïmplementeerd die men als must have ziet voor het programma.

Waarin in increment 1 configuratie van PTM, rechten en medewerkers use cases worden geïmplementeerd. In increment 2 de benodigde use cases voor het bijhouden van projecten en projecturen. Tijdens increment 3 zullen de use cases voor verlofdagen, ziekmeldingen en externe kosten geïmplementeerd worden.

De use cases in increment 7 en 8 kunnen gezien worden als optioneel. Waar tijd beschikbaar is binnen de rest van de projectverloop zullen deze zo veel mogelijk geïmplementeerd worden.

Resultaten:

- PTM-software.
- Software Documentatie.

Activiteiten:

- GUI PTM-software ontwikkelen.
- PTM-software ontwikkelen.
- Testen met behulp van Unit tests.
- Schrijven van software documentatie.

4.4.1 Increment 1

Totale duur van increment 1 is drie weken en loopt samen met prototyping.
Effectieve duur van 1,5 weken.

Domeinmodel implementeren (5 werkdagen):
Implementatie van complete EJB domeinmodel

Databasemodel implementeren (3 werkdagen):
Implementatie van de database tabellen voor de implementatie van het EJB domeinmodel.

4.4.2 Increment 2

Totale duur van increment 2 is drie weken.

Periodes (4 werkdagen):
Periode toevoegen (must have)
Periode bewerken (must have)
Periode verwijderen (must have)

Agenda scherm(10 werkdagen):
DateChooser implementeren (must have)
Inzien van agenda's van andere werknemers implementeren (must have)
Selectiemogelijkheden voor periodes toevoegen (must have)

4.4.3 Increment 3

Totale duur van increment 3 is drie weken.

Medewerkers (5 werkdagen):
Medewerker toevoegen (must have)
Medewerker bewerken (must have)
Medewerker verwijderen (must have)
Medewerker inactiveren (must have)
Medewerker activeren (must have)
Medewerker aan project toevoegen (must have)

Rechten (1 werkdag):
Inloggen (must have)
Uitloggen (must have)

Medewerkersscherm, inclusief in- en uitloggen (10 werkdagen)

4.4.4 Increment 4

Totale duur van increment 4 is drie weken.

Projecten (5 werkdagen)

- Project aanmaken (must have)
- Project bewerken (must have)
- Project verwijderen (must have)
- Project inactiveren (must have)
- Project activeren (must have)
- Project markeren als gefactureerd (must have)
- Project markeren als niet gefactureerd (must have)
- Medewerker aan project toevoegen (must have)

Projectschermb maken (10 werkdagen)

4.4.5 Increment 5

Totale duur van increment 5 is drie weken.

Klanten (10 werkdagen):

- Klant toevoegen (must have)
- Klant bewerken (must have)
- Klant verwijderen (must have)
- Klant inactiveren (must have)
- Klant activeren (must have)
- Aangepaste klantenlijst tonen (must have)

Klanten scherm implementeren (5 werkdagen):

4.4.6 Increment 6

Totale duur van increment 6 is één week.

Externe kosten (3 werkdagen):

- Externe kosten toevoegen aan (sub)project (must have)
- Externe kosten bewerken (must have)
- Externe kosten verwijderen (must have)
- Externe kosten markeren als niet factureerbaar (must have)
- Externe kosten markeren als factureerbaar (must have)

Accountmanagers (1 werkdag):

- Accountmanager toewijzen aan klant (should have)
- Accountmanager van klant verwijderen (should have)

Contactpersonen (1 werkdag):

- Contactpersoon aan klant toevoegen (should have)
- Contactpersoon van klant verwijderen (should have)
- Contactpersoon van klant bewerken (should have)

4.4.7 Increment 7 (optioneel)

Totale duur van increment 7 is één week.

PTM beveiliging (3 werkdagen):

- Limiteren aantal inlogpogingen (could have)
- Limiteren aantal inlogpogingen uitzetten (could have)
- Limiteren aantal inlogpogingen bewerken (could have)
- Blokking ip-adres opheffen (could have)
- Blokking gebruikersprofiel opheffen (could have)

Factureren (2 werkdagen):

- Factuur toevoegen (could have)
- Factuur bewerken (could have)
- Factuur verwijderen (could have)
- Factuur markeren als betaald (could have)
- Factuur markeren als niet betaald (could have)

4.4.8 Increment 8 (optioneel)

Factureren (5 werkdagen):

- Projectkosten dashboard (could have)

4.5 Bouwfase 2

In bouwfase 2 worden de rapportages ontwikkeld en in PTM geïntegreerd.

Resultaten:

- PTM-software met geïntegreerde rapportage én chartingfunctionaliteit.
- Software Documentatie.

Activiteiten:

- Ontwikkelen van statistiek weergave module.
- Integreren van statistiek weergave module in PTM.
- Testen met behulp van Unit tests.
- Schrijven van software documentatie.

4.6 Testfase

Doel van de Testfase is het compleet testen van de geïmplementeerde functionaliteiten en het corrigeren van de gevonden fouten zodat de software foutloos opgeleverd kan worden.

Resultaten:

- Getest en gecorrigeerd PTM.
- Getest en gecorrigeerd rapportage en charting module.
- Testrapport.

Activiteiten:

- Testen van de PTM-software en de rapportage en charting module voor statistieken volgens het Acceptatie Test Plan.
- Corrigeren van gevonden fouten in de software.
- Schrijven van testrapport waarin de gevonden fouten beschreven staan, oorzaak van de fout en welke acties ondernomen zijn om de fout te corrigeren.

4.7 Rapporteer & afrondingsfase

Het doel van het rapporteer & afrondingsfase is het vastleggen van de uitgevoerde taken in een procesverslag en het voorbereiden op de eindpresentatie.

Resultaten:

- Afstudeerverslag.
- PTM gebruikershandleiding.

Activiteiten:

- Afronden van het schrijven van het afstudeerverslag.
- Schrijven van de PTM gebruikershandleiding.
- Voorbereiding eindpresentatie.

5 Beheersaspecten

In het huidige hoofdstuk worden de beheersaspecten die toegepast worden binnen het project beschreven. Deze beheersaspecten dienen voor het garanderen van de kwaliteit van werken, gemaakte documenten en ontwikkelde software. Tevens zorgen de beheersaspecten ervoor dat tijdens het project aan de projectplanning voldaan wordt en op tijd ingegrepen wordt als de ingeplande tijd overschreden dreigt te worden.

5.1 *Beheersaspect informatie*

In het beheersaspect informatie wordt beschreven welke bronnen gebruikt worden en hoe het gebruik van bronnen verantwoord wordt.

5.1.1 Bronnen

Tijdens de uitvoering van het project zal voornamelijk het internet als informatiebron gebruikt worden voor het onderzoek. Waar nodig zal eventueel navraag gedaan worden bij de leveranciers van producten.

Binnen ISAAC kan voor het project gebruik gemaakt worden van diverse kennisexperts. Verder zal ook gebruik gemaakt kunnen worden van JBoss gold partner ondersteuning.

Daarnaast beschikt ISAAC over diverse boeken die ondersteunend zijn voor het project en die waar nodig geraadpleegd worden.

5.1.2 Gebruik en rapportage

Naarmate het project vordert zal de informatiebehoefte en gebruik steeds veranderen en zal waar nodig gebruik gemaakt worden van bovenstaande bronnen.

Waar nodig zullen gebruikte bronnen in documenten vermeld worden.

5.2 *Beheersaspect tijd*

In de beheersaspect tijd wordt beschreven welke planning gevolgd wordt voor het project en hoeveel dagen beschikbaar zijn voor het project. Tevens staat beschreven hoe de gemaakte uren geregistreerd, bewaakt en verantwoord worden.

5.2.1 Globale planning

Voor de globale planning zie bijlage 1: projectplanning

De globale planning van het project zal afwijken van de standaard 100 werkdagen doordat de student maximaal 6 uur per dag kan werken. Omdat er minimaal 800 uur besteed moet worden aan het afstuderen zal daardoor het aantal project werkdagen oplopen naar 134 werkdagen.

5.2.2 Registratie, tijdverantwoording en bewaking

Voor het registreren van gemaakte uren zal gebruik gemaakt worden van de huidige versie van PTM. Daarnaast zal er een logboek bijgehouden worden die met de planning een goed overzicht geeft van de ingeplande uren en de daadwerkelijk bestede uren. In het afstudeerverslag zal in detail beschreven worden waar de planning gehaald is en/of niet en wat hiervoor de redenen waren. Ook zal er overleg gepleegd worden met de bedrijfsbegeleider als er onvoorziene omstandigheden voor vertraging zorgen of dreigen te zorgen.

5.3 *Beheersaspect kwaliteit*

5.3.1 Criteria

De opgeleverde broncode dient te voldoen aan de binnen ISAAC gehanteerde codestandaard. Dit betekent dat de broncode moet voldoen aan de Java codestandaard met een paar ISAAC specifieke uitzonderingen

Documentatie dient opgeleverd te worden volgens het binnen het project gehanteerde opmaak en dient vrij te zijn van spelling- en grammaticafouten.

5.3.2 Registratie

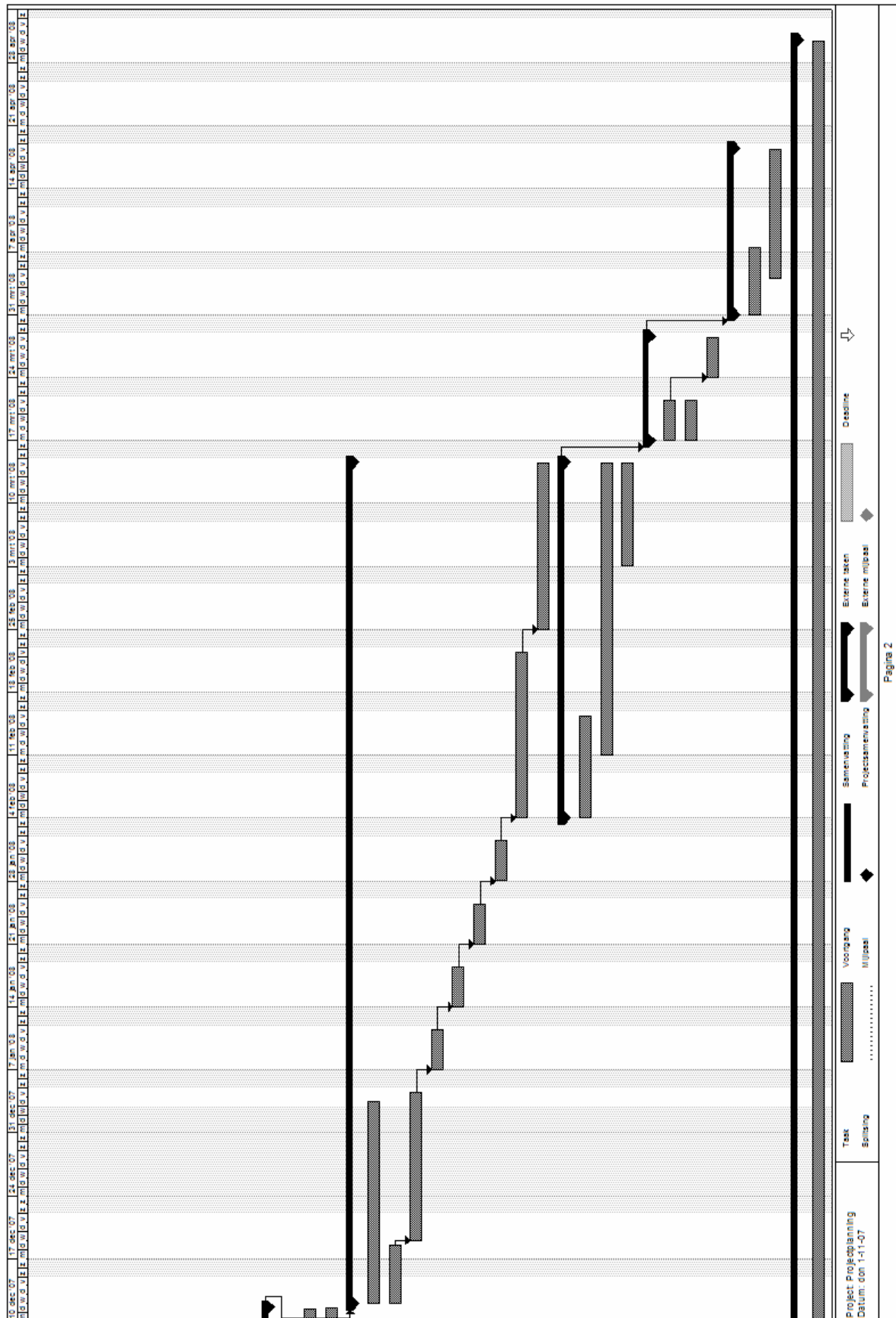
In elk document is een versiebeheer aanwezig waarin men kan zien welke versie en status het document heeft, en welke wijzigingen er zijn toegepast.

Voor software wordt gebruik gemaakt van Subversion om wijzigingen bij te houden en eventueel ontstane fouten te corrigeren.

5.3.3 Controle

Per fase van het project zal een voortgangsrapportage naar zowel de bedrijfsbegeleider en docentbegeleider gestuurd worden. In de voortgangsrapportage staat een kort verslag van afgeronde taken, tegengekomen problemen en oplossingen.

Per week zal, indien nodig, een (kort) gesprek gehouden worden met de bedrijfsbegeleider over de voortgang van het project en de kwaliteit van de verrichte werkzaamheden.




Bijlage 2:

Onderzoeksaanpak Plan

Onderzoeksaanpak Plan

PTM Onderzoek

Auteur:	Collin Maessen	 ISAAC
Datum:	6 december 2007	
Versie / status:	1.2 / Definitief	
File:	Onderzoeksaanpak Plan - 20071206 - v1.2.doc	

Versiebeheer

Datum	Versie	Belangrijkste wijzigingen	Auteur
02-10-2007	0.1	Eerste opzet.	Collin Maessen
16-10-2007	0.2	Eerste conceptversie	Collin Maessen
17-10-2007	1.0	Definitieve versie	Collin Maessen
25-10-2007	1.1	Aangepast aan de hand van feedback	Collin Maessen
06-12-2007	1.2	Aangepast aan de hand van feedback	Collin Maessen

Inhoudsopgave

1	CONCEPTUEEL MODEL.....	74
1.1	DOELSTELLING	74
1.1.1	<i>Projectkader.....</i>	74
1.1.2	<i>Onderzoekstype</i>	75
1.1.3	<i>Doelstelling</i>	75
1.2	ONDERZOEKSMODEL	76
1.2.1	<i>Onderzoeksoptiek</i>	76
1.2.2	<i>Visualisatie onderzoeksmodel</i>	77
1.3	VRAAGSTELLING	78
1.4	KERNBEGRIPPEN.....	79
2	ONDERZOEKSTECHNISCH ONTWERP.....	80
2.1	ONDERZOEKSSTRATEGIE	80
2.2	ONDERZOEKSPLANNING	80

1 Conceptueel Model

In het conceptueel model beschrijven wij wat, waarom en hoeveel wij gaan onderzoeken. Het conceptueel bestaat uit drie delen, de doelstelling, onderzoeksmodel, vraagstelling en de gebruikte kernbegrippen.

In de doelstelling beschrijven wij wat wij willen bereiken met het onderzoek en binnen welk kader het onderzoek zal plaatsvinden, het zogenaamde projectkader.

Het onderzoeksmodel beschrijft in woord en beeld hoe het onderzoek opgezet is, welke middelen/informatiebronnen gebruikt zullen worden en in welke stappen het onderzoek doorlopen zal worden.

Met behulp van de beschrijving in het onderzoeksmodel worden de vragen voor het onderzoek opgesteld in de vraagstelling.

Als laatste worden de binnen het onderzoek te gebruiken kernbegrippen omschreven en geconcretiseerd.

1.1 Doelstelling

In de projectkader omschrijven wij wat de huidige situatie is waarbinnen het onderzoek zal plaatsvinden, en welke resultaten men wenst.

Met behulp van het projectkader wordt het onderzoekstype bepaald. Het onderzoekstype beschrijft het soort onderzoek dat zal plaatsvinden en waarom hiervoor gekozen is.

Bovenstaande informatie wordt vervolgens gebruikt voor het opstellen van een doelstelling waarin beschreven wordt wat men wil bereiken met het onderzoek.

1.1.1 Projectkader

Binnen ISAAC werkt Project Time Management (PTM) goed voor het bijhouden van klantgegevens, projecten en projecturen. Echter is het systeem technologisch verouderd waardoor er de volgende problemen zijn:

- Rapportages worden vaak nog buiten PTM bewerkt om de gegevens inzichtelijk te maken met behulp van berekeningen en grafieken. Dit maakt het creëren van rapportages met behulp van PTM onnodig bewerkelijk.
- Ook is het systeem vergeleken met de huidige beschikbare Web 2.0 technieken gebruikersonvriendelijk. Met Web 2.0 technieken kan onder andere voorkomen worden dat gebruikers vaak tussen verschillende pagina's moeten wisselen. Dit kan door bijvoorbeeld invoer- en bewerktfunctionaliteiten op de pagina zelf aan te bieden.
- Tevens is de achterliggende code enkele jaren oud en hebben diverse programmeurs op verschillende momenten gewerkt aan PTM. Dit heeft er voor gezorgd dat niet overal dezelfde programmeerstijl wordt gebruikt. Verder is het niet eenvoudig om nieuwe functionaliteiten toe te voegen door het gebruik van oudere programmeertechnieken.

Dit onderzoek is specifiek gericht op het onderzoeken van mogelijkheden die er zijn met een vernieuwd PTM voor het genereren van duidelijke rapportages en grafieken. Hiervoor zal worden onderzocht welke informatie beschikbaar is binnen PTM, hoe deze het beste weergegeven kan worden in rapporten en grafieken en welke rapportage en charting tools hiervoor het meest geschikt is.

De oplossingen voor de overige omschreven problemen zullen met behulp van het normale software ontwikkelingstraject oplossingen voor gezocht worden. Dit houdt in dat in samenwerking met gebruikers en belanghebbende gekeken zal worden wat men wenst, welke technieken hiervoor beschikbaar zijn om dit te realiseren en zullen er keuzes gemaakt worden voor de te gebruiken technieken.

1.1.2 Onderzoekstype

Voor de opzet van het onderzoek is gekozen voor een praktijkgericht onderzoek. Dit houdt in dat dit onderzoek voor een bestaande praktijksituatie (zie projectkader) een bijdrage zal leveren tot het komen van een oplossing van die bestaande situatie.

Als type van het praktijkgericht onderzoek is gekozen voor het diagnostisch onderzoek. Dit houdt in dat er tijdens het praktijkgericht onderzoek de huidige problemen onderzocht en in kaart gebracht zullen worden, en naar mogelijke oplossingen gezocht worden voor deze problemen. Hiermee wordt bedoelt dat binnen het onderzoek gekeken zal worden welke rapportages gewenst/mogelijk zijn binnen de vernieuwde PTM en welk rapportage en charting tool hiervoor het meest geschikt is.

Het daadwerkelijk ontwerpen en implementeren van de mogelijke oplossing zal plaatsvinden binnen het normale software ontwikkelingstraject.

1.1.3 Doelstelling

Het doel van het onderzoek is het in kaart brengen van de informatiewensen binnen ISAAC, en de beschikbare gegevens binnen de vernieuwde PTM en hoe deze gegevens het beste ontsloten kunnen worden met behulp van rapportages.

Met behulp van deze informatie en van de onderzoeksvragen zullen vier rapportage en charting tools onderzocht en vergeleken worden.

Aan het einde van het onderzoek zal een overzicht gegeven worden over de mogelijkheden van de vier verschillende rapportage en charting tools, en zal een advies gegeven worden welke van deze rapportage en charting tools het meest geschikt is voor gebruik binnen PTM.

1.2 Onderzoeksmodel

Met het onderzoeksmodel wordt een globaal overzicht gegeven de te nemen stappen binnen het onderzoek. Dit gebeurt met een schematische weergave van de te nemen stappen die leiden naar het doel van het onderzoek. Dit visueel onderzoeksmodel zal ondersteund worden met een verwoording van de genomen stappen.

Het onderzoeksmodel wordt vervolgens gebruikt voor het opstellen van de onderzoeksvragen.

1.2.1 Onderzoeksoptiek

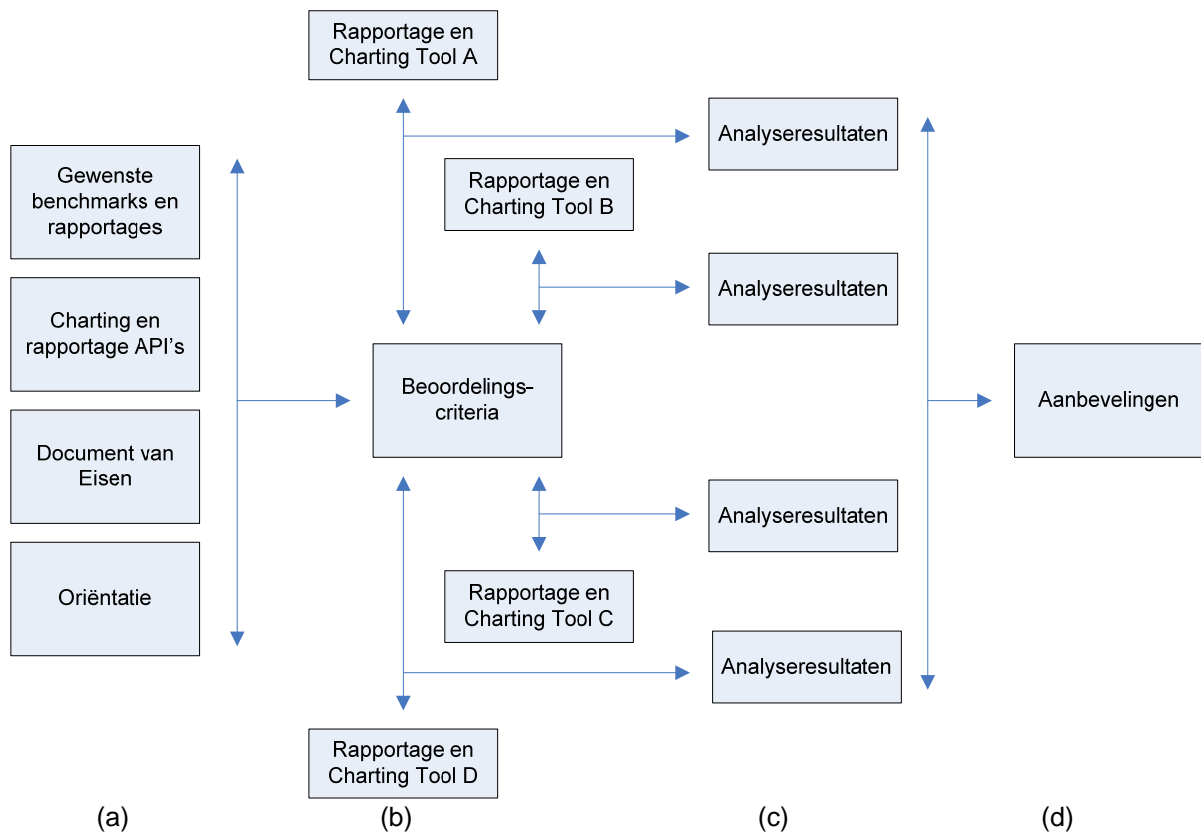
Het onderzoeksoptiek bepaalt welke facetten van het onderzoeksobject (vernieuwde rapportage mogelijkheden van PTM) worden bestudeerd en vanuit welke invalshoek.

Dit betekend dat gekozen wordt voor een manier van onderzoeken die bepalend is voor de opzet van het onderzoek, de te doorlopen stappen, en hoe de vragen opgesteld dienen te worden.

Tijdens dit onderzoek zal gebruik gemaakt worden van twee onderzoeksoptieken:

1. Een diagnostisch onderzoeksoptiek met als doel om te inventariseren welke rapportages gewenst zijn om de huidige bedrijfsprocessen te ondersteunen. Ook zal onderzocht worden welke functionaliteiten de rapportage en charting tools moeten ondersteunen om geschikt te zijn voor gebruik binnen de vernieuwde PTM. De resultaten zullen verwerkt worden in een lijst van criteria.
2. Een evaluatieonderzoeksoptiek waarin de opgestelde criteria gebruikt zullen worden om de vier rapportage en charting tools met elkaar te vergelijken en te evalueren. Dit levert uiteindelijk een overzicht op van de mogelijkheden van de verschillende rapportage en charting tools waarna een advies gegeven zal worden over de te gebruiken rapportage en charting tool.

1.2.2 Visualisatie onderzoeksmodel



(a) Een bestudering van mogelijke rapportage en charting tools voor integratie in de vernieuwde PTM versie, gebaseerd op gesprekken met domeindeskundigen, en op een oriëntatie van de beschikbare rapportage en charting tools en het document van eisen leveren de beoordelingscriteria, (b) waarmee vier soorten rapportage en charting tools kunnen worden geëvalueerd. (c) Een vergelijking van de resultaten van deze vier evaluaties resulteert in (d) aanbevelingen over welke rapportage en charting API het beste gebruikt kan worden.

1.3 Vraagstelling

De vraagstelling is opgesplitst in drie centrale vragen. Dit zijn de hoofdvragen die beantwoord dienen te worden binnen het onderzoek. Waar nodig zijn deelvragen toegevoegd.

In de eerste centrale vraag wordt getracht te achterhalen wat de criteria zijn waaraan een Rapportage en Charting API moet voldoen voor gebruik binnen de vernieuwde PTM.

- I Wat zijn de criteria voor de beoordeling van deze rapportage en charting API's?
- a. Welke rapportages en grafieken wil men kunnen genereren in PTM?
 - b. Welke van deze rapportages en grafieken kunnen daadwerkelijk gegenereerd worden met de beschikbare gegevens in PTM?
 - c. Aan welke licentievoorwaarden en/of kostenvoorwaarden moet worden voldaan om een rapportage en charting API te gebruiken binnen de vernieuwde PTM?
 - d. Aan welke voorwaarden voor uitvoerformaat moet een rapportage en charting API voldoen?
 - e. Aan welke verdere gebruikerswensen en voorwaarden moet de rapportage en charting API voldoen?

In de tweede centrale vraag worden de vier gekozen rapportage en charting API's vergeleken en geëvalueerd.

- II Hoe worden de vier onderzochte rapportage en charting API's beoordeeld in het licht van de gestelde criteria?
- a. Heeft de rapportage en charting API de benodigde rapportage en charting functionaliteit die gewenst is voor de vernieuwde PTM?
 - b. Welke licentievoorwaarden en kostprijs zijn verbonden aan het gebruik van de rapportage en charting API?
 - c. Is de rapportage en charting API geschikt voor de integratie in een EJB3.0 programma met een rich client weergave?
 - d. Welke andere weergave formaten worden ondersteund door de rapportage en charting API?
 - e. Welke specifieke voor- en nadelen heeft het gebruik van de rapportage en charting API?

In de derde centrale vraag wordt op de basis van de vergelijking en evaluatie van de vier gekozen rapportage en charting API's een advies over de geschiktheid uitgebracht.

- III Welke aanbevelingen worden afgeleid uit de vergelijking van de vier onderzochte rapportage en charting API's?

1.4 Kernbegrippen

Kernbegrippen dienen om gebruikte begrippen in het onderzoek te definiëren en af te bakenen zodat er geen onduidelijkheden en/of misverstanden ontstaan over de gebruikte terminologie.

Voor het bepalen van de kernbegrippen is afgeweken van de standaard procedure zoals beschreven volgens in de methode van Piet Verschuren en Hans Doorewaard. Er is gekozen om de begrippen waar mogelijk verwarring over kan ontstaan zonder tussenstappen uit de voorgaande teksten te halen en te beschrijven.

Rapportage en charting toolss:

Met rapportage en charting tools worden in JAVA geprogrammeerde programmeerbibliotheken bedoelt die in een EJB3.0 programma integreerbaar zijn, en geschikt zijn voor het weergeven van gegevens in een intranet/internet applicatie.

Weergave formaten:

Met weergave formaten van een rapportage en charting tool bedoelen wij de mogelijkheden om gegenereerde rapportages en grafieken weer te geven op een webpagina, als Excel bestand te exporteren en als een pdf bestand te exporteren. Verder zal gekeken worden of het mogelijk is om grafieken in een voor een intranet/internet applicatie geschikte vorm weer te geven.

Specifieke voor en nadelen:

Met de specifieke voor en nadelen van een rapportage en charting tool bedoelen wij de voor en nadelen van een rapportage en charting tool die tijdens het onderzoek naar voren zijn gekomen maar niet tot een van de andere gestelde vragen behoort.

2 Onderzoekstechnisch ontwerp

In het onderzoekstechnisch ontwerp wordt beschreven welke onderzoeksmaterialen gebruikt worden tijdens het onderzoek, welke onderzoeksstrategie gebruik zal worden en wat de planning is van het onderzoek.

In dit onderzoekstechnisch ontwerp zal geen beschrijving van de te gebruiken onderzoeksmaterialen aanwezig zijn, en wordt dus afgeweken van de door Piet Verschuren en Hans Doorewaard beschreven methodiek. Dit door de grote onzekerheid over de te gebruiken onderzoeksmaterialen die samenhangen met de te onderzoeken rapportage en charting tools.

Van de rapportage en charting tools is zeer weinig literatuur beschikbaar die deze stof behandelt. Hierdoor zal voornamelijk internetbronnen geraadpleegd worden die opgezocht zullen worden met behulp van een zoekmachine.

Dit zorgt ervoor dat vooraf niet te bepalen is welke bronnen gebruik zullen worden en hierdoor is het onmogelijk om een goede lijst van de te gebruiken bronnen op te zetten.

In de onderzoeksstrategie wordt beschreven hoe het onderzoek uitgevoerd zal worden en waarom het onderzoek op deze manier uitgevoerd wordt.

De onderzoeksplanning beschrijft de planning die gebruikt zal worden tijdens het onderzoek.

2.1 Onderzoeksstrategie

Een onderzoeksstrategie is de manier waarop binnen een onderzoek informatie wordt verzameld en het soort bronnen dat geraadpleegd wordt. Voor dit onderzoek is gekozen voor een Bureauonderzoek met als variant Literatuuronderzoek.

Dit houdt in dat het onderzoek als het ware achter een bureau plaatsvindt waarbij voornamelijk literatuur (in het geval van dit onderzoek internetbronnen) geraadpleegd wordt.

Het onderzoek zal geen puur bureauonderzoek zijn zoals beschreven in de methodiek van Piet Verschuren en Hans Doorewaard. Dit omdat tijdens het onderzoek ook gesprekken zullen plaatsvinden met belanghebbenden en gebruikers, en er onderzoek gepleegd zal worden met behulp van lichte prototyping. Dit zijn onderdelen die niet standaard behoren tot een bureauonderzoek maar in dit onderzoek een goede aanvulling geven.


2.2 Onderzoeksplanning

De onderzoeksplanning is opgenomen in het de totaalplanning voor het project. U vindt deze totaalplanning in het Plan van Aanpak.

Bijlage 3:

Onderzoeksrapport

Onderzoeksrapport PTM

Auteur:	Collin Maessen	 ISAAC
Datum:	13 december 2007	
Versie / status:	1.0 / Definitief	
File:	Onderzoekrapport - 20071213 - v1.0.doc	

Versiebeheer

Datum	Versie	Belangrijkste wijzigingen	Auteur
30-11-2007	0.1	Eerste conceptversie.	Collin Maessen
03-12-2007	0.2	Tweede conceptversie	Collin Maessen
04-12-2007	0.3	Derde conceptversie	Collin Maessen
07-12-2007	0.4	Vierde conceptversie	Collin Maessen
12-12-2007	0.5	Verklarende woordenlijst toegevoegd en enige tekstuele aanpassingen	Collin Maessen
13-12-2007	1.0	Definitieve versie	Collin Maessen

Inhoudsopgave

1	INLEIDING	86
2	REPORTING & CHARTING TOOLS	87
2.1	CENTRALE VRAAG: CV1.....	87
2.1.1	Deelvraag 1: CV1DV1.....	87
2.1.2	Deelvraag 2: CV1DV2.....	89
2.1.3	Deelvraag 3: CV1DV3.....	90
2.1.4	Deelvraag 4: CV1DV4.....	91
2.1.5	Deelvraag 5: CV1DV5.....	91
2.1.6	Opstellen criteria	93
2.1.7	Selecteren Rapportage & Charting Tools	93
2.2	CENTRALE VRAAG 2: CV2.....	95
2.2.1	Deelvraag 1: CV2DV1	95
2.2.2	Deelvraag 2: CV2DV2	98
2.2.3	Deelvraag 3: CV2DV3	99
2.2.4	Deelvraag 4: CV2DV4	101
2.2.5	Deelvraag 5: CV2DV5	101
2.3	CENTRALE VRAAG 3: CV3.....	103
3	CONCLUSIES EN AANBEVELINGEN	106
	LITERATUURLIJST	107

Verklarende woordenlijst

J2EE	Dit is een Java versie die speciaal gericht is voor het ontwikkelen van enterprise software.
Use case	Een use case beschrijft de stappen die doorlopen worden tijdens het gebruik van bepaalde functionaliteit binnen een applicatie.
JSP	Java Server Pages (JSP) is een onderdeel van de J2EE-standaard. JSP is een manier om dynamisch HTML, XML of andere inhoud te genereren met behulp van Java code.
JDBC	JDBC staat voor Java DataBase Connectivity, en kan door een Java-programma gebruikt worden om te communiceren met een database.
POJO	POJO is een acroniem voor Plain Old Java Object om aan te geven dat de geschreven Java code normale Java code is en niet bijvoorbeeld de J2EE versie.
Hibernate	Hibernate is een voor Java geschreven taal om vanuit een database gegevens op te vragen.
EJB3 Beans	Dit staat voor de derde versie van Enterprise JavaBeans en is onderdeel van J2EE/
Collection	Collection is een verzameling van gegevens die aan bepaalde voorwaarden voldoet zodat software weet hoe met een verzameling gecommuniceerd moet worden.
Java	Java is een programmeertaal waarmee computerprogramma's ontwikkeld kunnen worden.
IDE	Een Integrated Development Environment (IDE) is computersoftware die een softwareontwikkelaar ondersteunt bij het ontwikkelen van computersoftware.

1 Inleiding

Applicaties vergaren over de tijd dat deze gebruikt worden een zeer grote hoeveelheid aan nuttige gegevens. Het probleem dat vaak speelt met een dergelijke hoeveelheid aan gegevens is dat deze voor gebruikers niet in een nuttige en overzichtelijke vorm in te zien is. De standaard overzichtslijsten die gebruikt kunnen worden in een applicatie voldoen niet aan de informatiewensen van gebruikers.

Om aan deze informatiebehoefte te voldoen zijn er diverse Reporting & Charting Tools beschikbaar die gegevens kunnen omzetten tot informatierijke rapportages. In dit Onderzoek zijn de basis informatiebehoefte onderzocht en met de daaruit verkregen wensen zijn drie Reporting & Charting Tools onderzocht op hun mogelijkheden.

In hoofdstuk 2.1 besteden wij aandacht aan de criteria waaraan een Reporting & Charting Tool moet voldoen. Er vindt een inventarisatie plaats van de informatiebehoefte, gebruikerswensen en voorwaarden die gesteld worden voor het gebruik van een Reporting & Charting Tool. Dit levert een lijst van beoordelingscriteria op die in de daaropvolgende hoofdstukken en paragrafen gebruikt wordt voor het vergelijken van de Reporting & Charting Tools.

Daarna worden in hoofdstuk 2.2 de geselecteerde Reporting & Charting tools onderzocht op hun mogelijkheden en onmogelijkheden en wordt gekeken aan welke criteria ze voldoen.

Dit levert in hoofdstuk 2.3 een overzichtslijst van de mogelijkheden van een Reporting & Charting Tool met daarbij een korte uitleg over de verschillende Reporting & Charting tools.

Als laatste treft u de conclusies en aanbeveling aan die getrokken zijn uit het onderzoek met daarin een duidelijk advies voor een Reporting & Charting Tool.

2 Reporting & Charting Tools

In dit hoofdstuk probeer ik antwoord te geven op de centrale vragen.

2.1 Centrale vraag: CV1

De centrale vraag CV1 is als volgt opgesteld:

“Wat zijn de criteria voor de beoordeling van deze Reporting & Charting Tools?”

Deze centrale vraag is opgesplitst in vijf deelvragen die hieronder beantwoord worden, alvorens de centrale vraag te beantwoorden in hoofdstuk “2.1.6 Opstellen criteria”.

2.1.1 Deelvraag 1: CV1DV1

Deelvraag DV1 is als volgt geformuleerd:

“Welke rapportages en grafieken wil men kunnen genereren in PTM?”

Om deze vraag te beantwoorden zijn de volgende bronnen geraadpleegd:

- Notities van gesprekken met Dhr. M. Hufkens, Dhr S. Daemen, Dhr. P. Luedke, Dhr K. Peters en Mw. K. Schengenga
- Document van Eisen PTM - 20071106 - v0.6

Tijdens gesprekken is naar voren gekomen dat men de huidige rapportage functionaliteiten van PTM behouden wil zien. Deze rapportage functionaliteiten variëren van overzichtslijsten van bijvoorbeeld klanten tot dynamisch te genereren overzichten die geëxporteerd kunnen worden naar Excel.

De Overzichtslijsten van klanten zijn eenvoudig te genereren met de aanwezige functionaliteiten binnen de te ontwikkelen applicatie. Daarnaast is het gebruik van een Reporting & Charting Tool voor dergelijke lijsten af te raden ten opzichte van gebruikersvriendelijkheid. In hoofdstuk 2.2.3 Deelvraag 3: CV2DV3 ga ik verder in op wat wel en niet mogelijk is op dit gebied.

De dynamisch te genereren lijsten in PTM vallen in de huidige implementatie onder de namen “Uren” en “Kosten”. In de uren overzichten is het mogelijk om voor klanten, projecten en medewerkers overzichten te genereren over de gewerkte uren. Deze overzichten worden dan geanalyseerd om geïnformeerde beslissingen mee te maken of voor het genereren van rapportages. Deze functionaliteit wil men graag behouden binnen PTM zodat het mogelijk blijft, om waarnodig en waar gewenst, eigen rapportages te kunnen generen. Deze functionaliteit is ook te implementeren door gebruik te maken van bestaande functionaliteiten binnen J2EE.

Echter mist men de mogelijkheid in PTM om standaard rapportages en grafieken standaard te kunnen genereren. De rapportages en grafieken die men mist zijn vaak rapportages die erg bewerkelijk zijn om te genereren, of met de huidige rapportage mogelijkheden niet de volledige gewenste informatie geven.

De vraag is nu welke rapportages en grafieken men mist voor het ondersteunen van de bedrijfsprocessen binnen ISAAC. De meest voorkomende taken waarbij PTM wordt gebruikt zijn de volgende:

- Factureren van klanten
- Bijhouden van projecten
- Projectmanagement

Facturatie

Facturatie verloopt op het moment door het genereren van een overzicht van de gewerkte uren van medewerkers voor een project. Met het gegenereerde Excel bestand worden de gemaakte uren geanalyseerd en besloten welke uren factureerbaar zijn naar de klant en welke niet. Dit levert een aangepast Excel bestand op waarmee de financiële administratie de klanten vervolgens factureert.

Om dit proces te ondersteunen zijn de volgende rapportages voorgedragen:

Projecturen staat

Een projecturen staat is bedoeld om de beschikbare gegevens over gewerkte uren te presenteren, zodat hiermee beslissingen te maken zijn over kosten, projectlooptijd en het rendement van een project (factureerbare en niet factureerbare uren). Er zal per werknemer een overzicht zijn van de gewerkte uren, zowel factureerbaar als niet factureerbaar, en de beschikbare uren. Met voor het project de totalen van de uren.

Project rapportage

Een projectrapportage geeft inzicht in gewerkte uren (factureerbare en niet factureerbare), gemaakte kosten voor een project, ingeplande uren, behaalde rendementen en behaalde resultaten.

Met deze rapportages zijn de gegevens beschikbaar voor een snelle afhandeling van het factureren van projecten aan klanten.

Voor het genereren van deze rapportages zijn de volgende gegevens nodig:

- Gewerkte uren per werknemer.
- Gegevens over factureerbare en niet factureerbare uren.
- Geschatte uren per werknemer.
- Bijhouden van gemaakte kosten.
- Ontvangen betalingen van klant.

De Reporting & Charting Tool moet de volgende functionaliteiten ondersteunen om dit mogelijk te maken:

- Pie charts.
- Barcharts.
- Combineren van meerdere databronnen voor het genereren van gegevens en overzichten.

Registratie projecturen

Het registreren van projecturen wordt door alle ISAAC werknemers uitgevoerd voor alle werkzaamheden. Hierdoor is een schat aan gegevens beschikbaar over wat een werknemer gedaan heeft, voor welk project, en hoeveel uren hij hieraan gewerkt heeft.

Deze gegevens worden op het moment inzichtelijk gemaakt met behulp van de overzichtslijsten die de huidige versie PTM kan genereren. Vervolgens worden deze gegevens handmatig bewerkt tot grafieken en overzichten waarmee procesbewaking uit te voeren is (projectleiders en directie) of medewerkers beoordeelt kunnen worden op hun prestaties.

In de voorgaande paragraaf genoemde "Projecturen staat" kan de hier gewenste gegevens complimenteren.

Om dit proces te ondersteunen zijn de volgende rapportages voorgedragen:

Medewerkers totaalrapportage

De medewerkers rapportage toont een totaaloverzicht van de rendementen die werknemers behaald hebben over hun projecten

Dit overzicht bestaat uit een overzicht van de factureerbare en niet factureerbare uren voor alle projecten (niet per project gespecificeerd), de beschikbare uren voor projecten (niet per project gespecificeerd) en de daarmee berekende rendement van de werknemers.

Medewerker rapportage

Een medewerker rapportage toont voor een medewerker een overzicht van de behaalde rendementen over zijn projecturen.

Dit overzicht bestaat uit een overzicht van de factureerbare en niet factureerbare uren voor alle projecten, de beschikbare uren voor projecten en de daarmee berekende rendementen van de werknemer.

De gegevens zijn in dit overzicht wel per project op detailniveau beschikbaar.

Voor het genereren van deze rapportages zijn de volgende gegevens nodig:

- Gewerkte uren per werknemer.
- Gegevens over factureerbare en niet factureerbare uren.
- Geschatte uren per werknemer.

De Reporting & Charting Tool moet de volgende functionaliteiten ondersteunen om dit mogelijk te maken:

- Pie charts
- Scatter plot
- Combineren van meerdere databronnen tot een overzicht
- Geneste tabellen van meerdere databronnen.

Projectmanagement

Projectmanagement wordt voornamelijk uitgevoerd door de projectleiders van de projecten. Hierbij is het van belang dat de projectleiders inzicht hebben in de gewerkte uren, beschikbare uren en de deadlines van projecten om de voortgang van het project te kunnen bewaken.

In de huidige versie van PTM worden gegenereerde overzichten gebruikt in combinatie met Microsoft Project en Excel voor het bewaken van de voortgang.

De in de vorige paragrafen genoemde rapportages ondersteunen projectleiders op een dergelijke manier dat er op het moment geen verdere rapportages zijn aangedragen om aan een bepaalde informatiebehoefte te voldoen.

2.1.2 Deelvraag 2: CV1DV2

Deelvraag DV2 is als volgt geformuleerd:

“Welke van deze rapportages en grafieken kunnen daadwerkelijk gegenereerd worden met de beschikbare gegevens in PTM?”

Om deze vraag te beantwoorden zijn de volgende bronnen geraadpleegd:

- Notities van gesprekken met Dhr. M. Hufkens, Dhr S. Daemen, Dhr. P. Luedke en Dhr K. Peters
- Document van Eisen PTM - 20071106 - v0.6
- PvA PTM - 20071112 - v1.3

In het Plan van Aanpak staat beschreven welke use cases in het Document van Eisen geïmplementeerd worden en wanneer. Met deze gegevens en de omschreven benodigde gegevens van Deelvraag 1 kan per rapportage bekeken worden of de rapportages mogelijk zijn en in welke vorm ze mogelijk zijn.

Projecturen staat

De gegevens:

- Factureerbare / niet factureerbare uren
- Gewerkte uren
- Beschikbare uren

Deze gegevens staan omschreven in de desbetreffende use cases als benodigde invoer. Hierdoor is het mogelijk om deze rapportage te maken.

Project rapportage

Voor een project zijn de gegevens:

- Ingeplande uren
- Factureerbare / niet factureerbare uren
- Gewerkte uren
- Kosten

Zijn omschreven in de desbetreffende use cases als benodigde invoer.

Echter staan de gegevens over de ontvangen betalingen in een optionele use case. Deze use case zal worden geïmplementeerd als hiervoor tijd beschikbaar is. Dit betekent dat deze rapportage beschikbaar zal zijn binnen de vernieuwde PTM, maar dat dit in een aangepaste vorm kan zijn doordat bepaalde gegevens niet beschikbaar zijn.

Medewerkers totaalrapportage

De gegevens:

- Factureerbare / niet factureerbare uren
- Gewerkte uren
- Beschikbare uren

Staan omschreven in de desbetreffende use cases als benodigde invoer. Hierdoor is het mogelijk om deze rapportage te maken.

Medewerker rapportage

De gegevens:

- Factureerbare / niet factureerbare uren
- Gewerkte uren
- Beschikbare uren

Staan omschreven in de desbetreffende use cases als benodigde invoer. Hierdoor is het mogelijk om deze rapportage te maken.

2.1.3 Deelvraag 3: CV1DV3

Deelvraag DV3 is als volgt geformuleerd:

“Aan welke licentievoorwaarden en/of kostenvoorwaarden moet worden voldaan om een rapportage en charting API te gebruiken binnen de vernieuwde PTM?”

Om deze vraag te beantwoorden zijn de volgende bronnen geraadpleegd:

Notities van gesprekken met Dhr. M. Hufkens

De voorkeur bij ISAAC gaat uit naar open source oplossingen die zonder restricties te gebruiken zijn in een commercieel product. Dit betekent dat het gebruik van een Reporting & Charting Tool er niet voor mag zorgen dat de broncode van PTM openbaar gemaakt moet worden of bijvoorbeeld onder dezelfde licentie als de Reporting & Charting tool gedistribueerd moet worden.

Daarnaast gaat de voorkeur ook uit naar een Reporting & Charting tool waarbij geen kosten berekend worden voor het gebruik ofwel verspreiding van de Reporting & Charting Tool in combinatie met PTM.

Indien restricties gelden op het gebruik van, en/of er kosten berekend worden voor gebruik, is dit bespreekbaar mits er een grote meerwaarde aanwezig door het gebruik van de Reporting & Charting Tool.

2.1.4 Deelvraag 4: CV1DV4

Deelvraag DV4 is als volgt geformuleerd:

“Aan welke voorwaarden voor uitvoerformaat moet een rapportage en charting API voldoen?”

Om deze vraag te beantwoorden zijn de volgende bronnen geraadpleegd:

- Notities van gesprekken met Dhr. M. Hufkens, Dhr S. Daemen, Dhr. P. Luedke en Dhr K. Peters
- Document van Eisen PTM - 20071106 - v0.6

Voor het gebruik binnen de webbased omgeving waarin de vernieuwde PTM komt te draaien moet de Reporting & Charting Tool ondersteuning bieden voor deze omgeving. Daarnaast verwacht men de mogelijkheid om rapporten in het Excel formaat te kunnen downloaden. Hier wil men tevens dat het naar Excel geëxporteerde rapport zoveel mogelijk Excel functies gebruikt zoals formules, grafieken en veldopmaak. Dit zodat rapporten eenvoudig buiten PTM te bewerken zijn. Daarnaast wil men graag dat rapportages ook in pdf formaat beschikbaar gesteld kunnen worden voor gemakkelijke verspreiding van rapporten en voor het eenvoudig afdrukken.

Een export functie voor het exporteren van rapportages naar Word documenten is een handige extra, echter is dit niet specifiek genoemd als eis. Deze functionaliteit is gekozen als “zou leuk/handig zijn” doordat men binnen ISAAC al aangaf dat men rapportages buiten PTM wil kunnen bewerken en gebruiken. De rapportage in Word formaat sluit hier goed op aan maar is geen vereiste voor een Reporting & Charting Tool.

Alle andere export formaten die een Reporting & Charting Tool ondersteund zal gezien worden als een bonus functionaliteit. Deze zijn echter niet doorslaggevend voor de keuze voor een Reporting & Charting Tool.

2.1.5 Deelvraag 5: CV1DV5

Deelvraag DV5 is als volgt geformuleerd:

“Aan welke verdere gebruikerswensen en voorwaarden moet de rapportage en charting API voldoen?”

Om deze vraag te beantwoorden zijn de volgende bronnen geraadpleegd:

- Notities van gesprekken met Dhr. M. Hufkens, Dhr S. Daemen, Dhr. P. Luedke en Dhr K. Peters
- Document van Eisen PTM - 20071106 - v0.6

Uit gesprekken en werkwijzen bij ISAAC zijn de volgende verdere wensen en voorwaarden voor Reporting & Charting Tools samengesteld:

Eenvoudig integreerbaar in gebruikersinterface

De ontworpen rapportages moeten op een voor de programmeur op een eenvoudige manier integreerbaar zijn in de grafische user interface

Daarnaast moet het mogelijk zijn om gegenereerde rapportages te exporteren naar Excel en PDF. Dit via een link/verwijzing die toe te voegen is door de programmeur of een standaard functionaliteit van de rapportage viewer zelf.

Rapportages moeten minimaal op een JSP pagina, of variant van, beschikbaar moeten zijn zodat rapportages in de grafische user interface van de vernieuwde PTM geïntegreerd kunnen worden. Dit moet op een dergelijke manier mogelijk zijn zodat de gebruiker kan blijven navigeren met behulp van de standaard PTM navigatie.

De meest ideale variant van bovenstaande functionaliteit is de integratie van de rapportage en/of grafieken in een JSP pagina. Hierdoor is Reporting & Charting Tool makkelijk te integreren in de normale navigatie en layout van de vernieuwde PTM versie.

Het moet mogelijk zijn om opties door te geven aan rapportages zodat ze kunnen reageren op gebruikersinput. Hiermee is het bijvoorbeeld mogelijk om een medewerker rapportage te laten genereren aan de hand van de keuze voor een medewerker. De rapportage gebruikt dan de beschikbare gegevens over de doorgegeven medewerker voor het genereren van de rapportage.

Ondersteuning voor EJB Entity Beans

In de vernieuwde PTM zal gebruik gemaakt worden van zogenaamde EJB Entity Beans. Deze beans regelen de uitwisseling van gegevens tussen de database en de applicatie. Hierdoor is er binnen de applicatie geen code aanwezig die direct een bepaalde database aanspreekt en zodoende er ook van afhankelijk is.

Om te voorkomen dat een Reporting & Charting Tool door direct communiceren met de database deze afhankelijkheid introduceert zal deze EJB Entity Beans als databron moeten ondersteunen.

Bij voorkeur IDE integratie

Voor het ontwikkelen van Rapportages kan er veel ontwikkeltijd en onderzoekwerk bespaart worden als er voor de programmeur een goed opgezette IDE integratie beschikbaar is. Met een dergelijke integratie kunnen bijvoorbeeld basale handelingen of veel uitgevoerd bewerkingen geautomatiseerd zijn.

Omdat binnen ISAAC de Eclipse ontwikkelomgeving gebruikt wordt is er een voorkeur voor een directe integratie met deze omgeving, dit zal in de vorm van een plug-in zijn. Een losse ontwikkelomgeving voor het ontwikkelen en ontwerpen van rapportages is ook een mogelijkheid, maar heeft niet de voorkeur omdat dit integratie en testen met de applicatie ingewikkelder maakt.

2.1.6 Opstellen criteria

Met de beantwoorde vragen in de voorgaande paragrafen is de volgende tabel samengesteld waarmee de Reporting & Charting Tools vergeleken zullen worden:

	Reporting & Charting Tool A	Reporting & Charting Tool B	Reporting & Charting Tool C	Reporting & Charting Tool D
Ondersteunde invoer				
JDBC				
XML				
EJB				
POJO				
Hibernate				
Overige				
Uitvoerformaten				
HTML				
Excel				
PDF				
Overige				
Report delivery				
Web				
Download				
IDE integratie				
Eclipse				
JBOSS IDE				
MyEclipse				
Anders				
Reports				
Ondersteunde charts				
Kruistabel				
Ondersteunen parameters				
Overige				
Realtime				
J2EE				
Licentie				
Kosten				

2.1.7 Selecteren Rapportage & Charting Tools

Om deze vraag te beantwoorden zijn de volgende bronnen geraadpleegd:

- Notities van gesprekken met Dhr. M. Hufkens, Dhr S. Daemen, Dhr. P. Luedke en Dhr K. Peters
- Document van Eisen PTM - 20071106 - v0.6
- Aantekeningen van oriëntatie op onderwerp
- BIRT website <http://www.eclipse.org/birt/phoenix/>
- Crystal Reports website <http://www.businessobjects.com/>
- JasperReports website http://www.jasperforge.org/jaspersoft/opensource/business_intelligence/jasperreports/
- Onderzoek Reporting Tools door Nick Martens

Tijdens de oriëntatie op het onderzoekskader en onderwerp zijn al diverse Reporting & Charting Tools bekeken om bekendheid te krijgen met het onderwerp en de beschikbare pakketten.

Hierdoor is het mogelijk om een selectie te maken voor een aantal te onderzoeken Reporting & Charting Tools. Deze zijn geselecteerd op bekende wensen en eisen waaraan ze moeten voldoen. De geselecteerde Reporting & Charting Tools zijn:

BIRT

Business Intelligence and Reporting Tools (BIRT) is geselecteerd doordat deze tool vaak in commerciële producten gebruikt wordt en zeer belovend uitziet qua mogelijkheden.

JasperReports

JasperReports is een open source Reporting & Charting Tool die ontwikkeld wordt door het bedrijf JasperSoft. Qua mogelijkheden is deze sterk te vergelijken met BIRT

De standaard producten van JasperReports zijn vrij beschikbaar en te gebruiken, maar er worden tevens, betaalde, gespecialiseerde oplossingen aangeboden voor het optimaliseren van rapportage generatie.

Crystal Reports for Eclipse (CR4E)

Crystal Reports is de Reporting & Charting Tool van BusinessObjects, CR4E maakt gebruik van de Crystal Reports voor Java Engine. Qua mogelijkheden lijkt deze ook te voldoen aan de door ISAAC gestelde eisen.

In het Onderzoeksaanpak Plan staat aangegeven dat vier Reporting & Charting Tools geselecteerd en onderzocht zullen worden. Echter door voorselectie op ondersteunde functionaliteiten die genoemd worden op de productwebsites van de diverse Reporting & Charting Tools, en doordat er maar enkele grote en goed ondersteunde Reporting & Charting Tools bestaan, zijn er uiteindelijk drie geselecteerd.

2.2 Centrale vraag 2: CV2

De centrale vraag CV2 is als volgt opgesteld:

“Hoe worden de vier onderzochte rapportage en charting API's beoordeeld in het licht van de gestelde criteria?”

Deze centrale vraag is opgesplitst in vijf deelvragen die hieronder beantwoord worden, waarvan de antwoorden gebruikt zullen worden voor het beantwoorden van de derde centrale vraag in Hoofdstuk 2.1.

2.2.1 Deelvraag 1: CV2DV1

Deelvraag DV1 is als volgt geformuleerd:

“Heeft de rapportage en charting API de benodigde rapportage en charting functionaliteit die gewenst is voor de vernieuwde PTM?”

Om deze vraag te beantwoorden zijn de volgende bronnen geraadpleegd:

- BIRT website <http://www.eclipse.org/birt/phoenix/>
- Crystal Reports website <http://www.businessobjects.com/>
- JasperReports website http://www.jasperforge.org/jaspersoft/opensource/business_intelligence/jasperreports/

BIRT

Charts

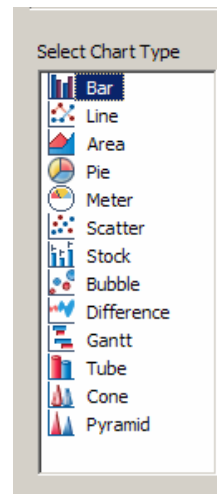
Alle benodigde grafieken worden ondersteund door BIRT. Samen met Crystal Reports ondersteund BIRT de meeste soorten en combinaties van grafieksoorten. Met de ondersteunde grafieken van BIRT is het geen probleem om gedegen rapportages op te zetten.

Voor alle charts binnen BIRT kan het formaat ingesteld worden, BIRT is de enige rapportage tool die deze functionaliteit ondersteund. Beschikbaar zijn PNG, BMP, JPG en SVG.

BIRT is daarnaast ook de enige tool waarbij wijzigingen in een grafiek direct zichtbaar zijn. Dus wijzigingen in de opmaak of bijvoorbeeld soort grafiek zijn direct zichtbaar in de rapportage. Zelfs als er nog geen gegevens beschikbaar zijn voor het genereren van de grafiek/rapportage.

Rapportages

Vanaf BIRT 2.2 is het ook mogelijk om zogenaamde kruistabellen te gebruiken. Met deze kruistabellen is het mogelijk om niet direct gerelateerde gegevens met elkaar te vergelijken en conclusies mee te trekken. Dit maakt onder andere het genereren van de medewerkers rapportages gemakkelijker.



Daarnaast zijn de volgende functionaliteiten beschikbaar voor het ontwerpen van rapportages:

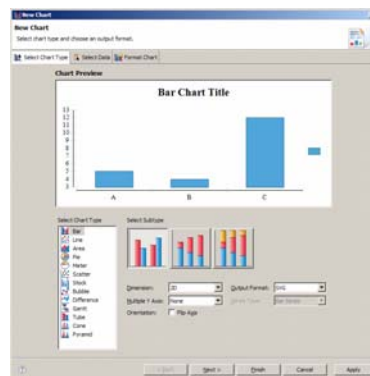
- Tabellen
- Lijsten
- Labels
- Dynamische teksten
- Afbeeldingen
- Grid

Met voor alle van deze weergave mogelijkheden opmaak opties en mogelijkheden.

BIRT laat ontwerpers van rapportages helemaal vrij in de indeling van rapporten, dit doordat er geen secties aanwezig zijn in een standaard BIRT rapport. Hierdoor kan elke gewenste lay-out gemaakt worden, dit maakt het opzetten van een rapportage wel iets bewerklijker.

Algemeen

Over het algemeen zijn de functionaliteiten van BIRT zeer toegankelijk en overzichtelijk ingedeeld in de plugin. Hierdoor is het makkelijk om bijvoorbeeld een grafiek te maken die qua opmaak voldoet aan de wensen, zeer eenvoudig om databronnen toe te wijzen en teksten in te stellen.



Crystal Reports

Charts

Van alle rapportage tools biedt Crystal Reports de meeste chart types die te gebruiken zijn in rapportages.

Grafieken zijn net zoals in BIRT eenvoudig in te stellen door in de chart het onderdeel aan te wijzen, bijvoorbeeld de titel, waarna deze direct te wijzigen zijn en alle andere opties via een properties panel beschikbaar zijn.

Rapportages

Net zoals BIRT ondersteund Crystal Reports kruistabellen in zijn rapportages waardoor ook met deze tool bepaalde rapportages makkelijk te maken zijn.

Ook qua verdere functionaliteiten die beschikbaar zijn voor het ontwerpen van rapportages biedt Crystal Reports een vergelijkbare set aan functionaliteiten.

Voor het ontwerpen van rapportages maakt Crystal Reports gebruik van secties. Dit biedt voordelen aan de ontwerpen omdat het makkelijker is om een rapportage op te maken. Echter is er wel een beperking met het maken van rapportages door het gebruik van secties: grafieken mogen niet in de body of page header sectie voorkomen van de rapportage. In alle andere secties, zoals bijvoorbeeld de report header, mag een grafiek wel voorkomen.

Algemeen

Qua gebruikersvriendelijkheid en overzichtelijkheid komt Crystal Reports in de buurt van BIRT. Functionaliteiten zijn makkelijk toegankelijk en eenvoudig in te stellen, wel in een minder mooi uitzijnde interface dan bijvoorbeeld bij BIRT, maar niet minder functioneel. Wel is het direct doorvoeren van wijzigingen in grafieken een gemis.

JasperReports

Charts

Van alle rapportage tools ondersteund JasperReports het minste aantal soorten charts, dit zijn wel de meest gebruikte charts zoals:

bar, line, pie, meter, area, scatter, stock, bubble en thermometer.



Voor het ontwerpen en opmaken van grafieken is JasperReports de minst intuïtieve van de drie Reporting Tools. Het samenstellen en opmaken van een grafiek is niet een kwestie van point and click zoals bij de voorgaande reporting tools. Binnen JasperReports moet via contextmenu's, of via het hoofdmenu, de diverse instellingen ingegeven worden. Een enkele keer kan het invoeren met behulp van point and click, en vervolgens een properties pane, echter de meeste instellingen moeten via het menu ingevoerd worden.

Rapportages

Ook JasperReports ondersteund kruistabellen voor het genereren van rapportages.

Qua verdere functionaliteiten die beschikbaar zijn voor het ontwerpen van rapportages biedt JasperReports een vergelijkbare set aan functionaliteiten.

JasperReports gebruikt voor het ontwerpen van rapportages secties en heeft ook de meeste secties in zijn rapportages. Echter wordt de gebruiker niet beperkt in welke weergave elementen hij waar plaatst in de rapportage. Enige voorwaarde is dat elementen niet meerdere secties mogen overlappen om weergave problemen te voorkomen, dit wordt duidelijk aangegeven door JasperReports als dit gebeurt.

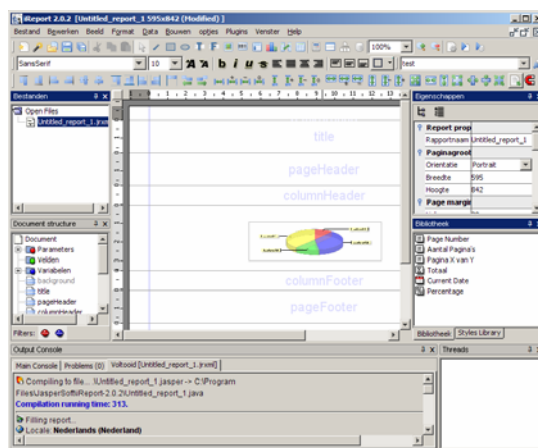
Algemeen

Qua gebruikersvriendelijkheid en overzichtelijkheid is JasperReports duidelijk de mindere vergeleken met Crystal Reports en BIRT.

Over het algemeen zijn instellingen en opties wel duidelijk qua gebruik en de feedback bij fouten is van JasperReports zeer goed te noemen.

De overzichtelijkheid van alle mogelijkheden en het gebruiksgemak tijdens het ontwerpen van rapportages laat te wensen over.

Ook in JasperReports is het direct doorvoeren van wijzigingen een gemis.



2.2.2 Deelvraag 2: CV2DV2

Deelvraag DV2 is als volgt geformuleerd:

“Welke licentievoorwaarden en kostprijs zijn verbonden aan het gebruik van de rapportage en charting API?”

Om deze vraag te beantwoorden zijn de volgende bronnen geraadpleegd:

- BIRT website <http://www.eclipse.org/birt/phenix/>
- Crystal Reports website <http://www.businessobjects.com/>
- Gesprek met Mw. A van Gastel, VCD Automatisering, Crystal Reports Reseller.
- JasperReports website http://www.jasperforge.org/jaspersoft/opensource/business_intelligence/jasperreports/

BIRT

BIRT wordt gedistribueerd onder Eclipse Public License (EPL) en is daardoor vrij en zonder voorwaarden te gebruiken in een commercieel ontwikkelde applicatie.

Echter staat in de EPL voorwaarden wel een afwijzing van alle aansprakelijkheid voor de gebruikte code en dat gemaakte claims, bijvoorbeeld over prestaties, voor verantwoordelijkheid zijn van de producent die BIRT gebruikt in zijn product.

Crystal Reports

De Free Edition van Crystal Reports for Eclipse is vrij te gebruiken binnen een commercieel ontwikkeld product. Het gebruik van Crystal Reports moet wel geregistreerd worden om een watermerk te verwijderen die aanwezig is in de niet geregistreerde versie, aan deze registratie zijn geen kosten verbonden.

Voor de Free Edition geldt daarnaast ook de restrictie dat deze schaalbaar is naar ongeveer 5 gelijktijdige gebruikers. Hoe dit getal berekent is staat niet vermeld in de aangeleverde documentatie en op de website van Crystal Reports. De betaalde versie zou schaalbaar zijn tot “single server”, wat dit betekend qua prestaties en wat de specificaties zijn van de schaalbaarheid op deze single server is ook niet te achterhalen.

JasperReports

De reporting tool JasperReports is verkrijgbaar onder de GNU Lesser General Public License (LGPL). Dit betekent dat JasperReports vrij te gebruiken is in een commercieel ontwikkeld pakket, mits duidelijk in PTM, of in een bijgevoegd document, vermeld staat dat JasperReports in de software gebruikt wordt.

2.2.3 Deelvraag 3: CV2DV3

Deelvraag DV3 is als volgt geformuleerd:

“Is de rapportage en charting API geschikt voor de integratie in een EJB3.0 programma met een rich client weergave?”

Om deze vraag te beantwoorden zijn de volgende bronnen geraadpleegd:

- BIRT website <http://www.eclipse.org/birt/phoenix/>
- Crystal Reports website <http://www.businessobjects.com/>
- JasperReports website http://www.jasperforge.org/jaspersoft/opensource/business_intelligence/jasperreports/
- Using the BIRT Report Viewer <http://www.eclipse.org/birt/phoenix/deploy/viewerUsage2.2.php>
- Crystal Reports, Create a report with Plain Old Java Objects (POJO) http://devlibrary.businessobjects.com/businessobjectsxir2/en/en/JRC_SDK/jrc_java_dg_doc/doc/jrcsdk_java_dg/StartHere16.html
- JasperReports Tutorial <http://struts.apache.org/2.x/docs/jasperreports-tutorial.html>
- Creating a Report with JasperReports http://www.regdeveloper.co.uk/2006/10/24/jasperreports_tutorial/

BIRT

Voor het genereren van rapportages kan BIRT gegevens lezen uit Entity Beans, POJO's en EJB3 Beans. Hieraan is wel de voorwaarde verbonden dat gebruik gemaakt wordt van een POJO waarin een readData() functie geïmplementeerd is. Deze functie wordt gebruikt voor het inlezen van een collection die vervolgens in een Scripted Datasource gebruikt wordt voor het genereren van rapportages.

Reports van BIRT zijn met behulp van een view te integreren in pagina's of zijn als pagina opvraagbaar binnen een webomgeving. Integratie van de view van een report kan met behulp van Java code of met behulp van de standaard JSP Tag Library van BIRT.

Daarnaast is het mogelijk met behulp van parameters om bijvoorbeeld gebruikers de mogelijkheid te geven de sorteervolgorde te veranderen van een overzicht in een rapportage. Standaard zorgt dit ervoor dat rapportages opnieuw gegenereerd en getoond moet worden door de view. Echter staat in de documentatie een verwijzing dat het veranderen van de sorteervolgorde mogelijk is zonder de rapportage opnieuw te genereren. Hoe dit werkt is op het moment van schrijven nog niet duidelijk.

Crystal Reports

Voor het genereren van rapportages ondersteund Crystal Reports volgens de maker Business Objects het lezen van Entity Beans, POJO's en EJB3 beans.

Echter is hier zo goed als geen informatie over te vinden hoe dit geprogrammeerd moet worden. Er is één voorbeeld gevonden waar Crystal Reports gebruikt maakt van een POJO voor het genereren van rapportages. Hierin is te zien hoe het mogelijk is om een POJO te laten vullen met database informatie (bijvoorbeeld met behulp van Entity Beans) en hoe daarmee de rapportage genereert kan worden.

Echter is deze informatie erg summier waardoor het niet te zeggen is hoe makkelijk deze functionaliteit te gebruiken is binnen de te ontwikkelen applicatie (er is zo goed als geen documentatie beschikbaar). Er is wel een werkend voorbeeld beschikbaar waarmee met behulp van reverse engineering waarschijnlijk een gedeelte van de informatie te achterhalen is.

Met behulp van de Crystal Reports for Eclipse plugin is het zeer eenvoudig om een viewer JSP pagina te maken. Hierin zijn bepaalde standaard functionaliteiten toe te voegen aan een viewer JSP pagina zoals export naar PDF.

Het enige probleem dat bij het genereren van de viewer page speelt is dat er geen duidelijke documentatie beschikbaar is over de benodigde libraries die toegevoegd moeten worden aan een bestaand project. Hierdoor vergt het enig zoekwerk om een viewer page werkend te krijgen. Op het moment van schrijven zijn de benodigde libraries bekend voor het maken van een JSP viewer page. Alleen bepaalde instellingen en interne afhankelijkheden van de benodigde libraries niet waardoor op het moment nog niet gelukt is om een werkende viewer page te implementeren.

Zodra dit achterhaald is het geen probleem om een rapportage te tonen op een JSP pagina en zelfs geïntegreerd in de normale lay-out van de vernieuwde PTM.

Net zoals in BIRT is het mogelijk om rapportages te laten reageren op basis van user input met behulp van parameters. Zoals bijvoorbeeld het anders laten sorteren van kolommen. Het is alleen onduidelijk of dit kan zonder de rapportage compleet opnieuw te laten genereren. Ook hier is weinig informatie over beschikbaar.

JasperReports

Net zoals voorgaande pakketten ondersteund JasperReports het lezen van Entity Beans, POJO's en EJB3.0 beans.

Dit werkt op eenzelfde manier zoals bij BIRT waarbij een POJO object als interface dient tussen JasperReports en de POJO objecten of de EJB3.0 beans.

Net zoals alle voorgaande rapportage tools is JasperReports te integreren in een webomgeving. Dit gebeurt net zoals bij de voorgaande rapportage tools met behulp van een viewer die op de pagina geïntegreerd wordt.

Over hoe dit geprogrammeerd moet worden is zeer duidelijke documentatie op internet te vinden en is met de beschikbare informatie zonder al te grote problemen te integreren in een bestaande applicatie.

Ook voor JasperReports is het mogelijk om bijvoorbeeld rapportages te laten reageren op user input. Waardoor mogelijkheden ontstaan zoals het sorteren van kolommen. Er is geen informatie gevonden of dit ook kan zonder de rapportage opnieuw te genereren en weer te geven.

Algemeen

Alle rapportage tools zijn niet geschikt voor het genereren van lijsten waarbij er de mogelijkheid moet zijn voor het bewerken van de weergegeven gegevens op de pagina zelf. Het is wel mogelijk om besturingselementen toe te voegen zodat men gegevens kan bewerken en/of toevoegen, maar dit gebeurt dan op een andere pagina. Dergelijke lijsten zullen dan ook via een eigen implementatie gegenereerd moeten worden om de gewenste gebruikersfunctionaliteit beschikbaar te stellen.

2.2.4 Deelvraag 4: CV2DV4

Deelvraag DV4 is als volgt geformuleerd:

“Welke andere weergave formaten worden ondersteund door de rapportage en charting API?”

Om deze vraag te beantwoorden zijn de volgende bronnen geraadpleegd:

- JasperReports overzicht pagina http://www.jaspersoft.com/JasperSoft_JasperReports.html
- Crystal Reports for Eclipse Release Notes <http://diamond.businessobjects.com/node/447>

BIRT

Vanaf versie 2.2 ondersteund BIRT standaard HTML, gepagineerde HTML, PDF, WORD, XLS, en PostScript. Vergeleken met versie 2.1 waar HTML, gepagineerde HTML en PDF standaard ondersteund werden is dit een grote verbetering.

In de onderzochte versie 2.2.1 van BIRT is het ook mogelijk om PPT bestanden te genereren. Dit gaat echter nog niet helemaal vlekkeloos. Test rapportages worden uiteindelijk wel in PPT formaat getoond aan de gebruiker maar er wordt wel een foutmelding gegeven dat het bestand beschadigt is. Als PPT met de huidige versie gebruikt moet kunnen worden dan is het aan te raden om een van de beschikbare emitters voor BIRT te gebruiken. Deze buiten het BIRT project ontwikkelde emitters kunnen extra output formaten toevoegen.

Crystal Reports

De gratis versie van Crystal Reports for Eclipse ondersteund alleen HTML, PDF, CSV en RTF. Voor de mogelijkheid om meer ondersteunde formaten toe te voegen, zoals onder andere de gevraagde Excel support, zal men de betaalde versie moeten kopen. Deze versie is te koop voor 495,- Dollar met voor ISAAC een korting van 11%.

JasperReports

JasperReports ondersteund standaard PDF, HTML, XLS, CSV en XML voor het generen c.q. exporteren van rapportages. Daarnaast is dit het enige pakket dat voor alle output formaten pixel perfecte weergave garandeert. Dit betekent dat een rapportage in elk weergave formaat hetzelfde uitziet.

2.2.5 Deelvraag 5: CV2DV5

Deelvraag DV5 is als volgt geformuleerd:

“Welke specifieke voor- en nadelen heeft het gebruik van de rapportage en charting API?”

Om deze vraag te beantwoorden zijn de volgende bronnen geraadpleegd:

- BIRT website <http://www.eclipse.org/birt/phoenix/>
- Crystal Reports website <http://www.businessobjects.com/>
- JasperReports website http://www.jasperforge.org/jaspersoft/opensource/business_intelligence/jasperreports/

BIRT

IDE integratie

Voor BIRT is een plugin beschikbaar die zonder problemen werkt onder Eclipse, MyEclipse en de JBOSS IDE.

Documentatie

Via de BIRT website is meestal de basale informatie beschikbaar voor het gebruik van BIRT, echter is soms wel enig zoekwerk vereist om de benodigde informatie te vinden. De gevonden informatie legt vaak wel duidelijk uit hoe bepaalde functionaliteit werkt, de informatie is vaak wel net te summier om direct toepasbaar te zijn binnen een applicatie. Hiervoor is dan verdere raadpleging van de API documentatie vereist. De API documentatie is door de integratie met Eclipse beschikbaar via de Eclipse Help functie.

Verder is op internet redelijk veel, en vaak duidelijke, informatie beschikbaar over BIRT. Echter is dit vaak wel informatie die over versie 2.1 van BIRT gaat, en kan daardoor misschien niet direct toepasbaar zijn op het ontwikkelen met BIRT 2.2.

Crystal Reports

IDE integratie

Voor Crystal Reports is een Eclipse plugin beschikbaar, de zogenaamde Crystal Reports for Eclipse die gedurende dit gehele onderzoek onderzocht is. Deze plugin werkt zonder problemen onder Eclipse, MyEclipse en de JBoss IDE.

Voor het installeren van deze plugin is het wel aan te raden om dit via de Eclipse Update manager uit te voeren. Dit doordat de plugging veel afhankelijkheden heeft met andere benodigde plugins. Deze zijn standaard niet geïnstalleerd in een Eclipse installatie. Doordat de Crystal Reports plugin op een andere discovery site staat dan de benodigde plugins zijn deze niet automatisch selecteerbaar door Eclipse. Alle benodigde plugins moeten handmatig geselecteerd worden en het afwezig zijn van een van de benodigde plugins kan vreemde problemen veroorzaken tijdens het gebruik van de Crystal Reports plugin.

Documentatie

Tijdens het onderzoek is naar voren gekomen dat de informatie op de officiële Crystal Reports website zeer gefragmenteerd en onduidelijk is. Het is zeer moeilijk om duidelijke informatie te vinden over producten en mogelijkheden. Ook de aanwezige documentatie over het gebruik van Crystal Reports lijdt onder dezelfde problemen.

Er is verder op het internet redelijk veel informatie te vinden over het werken met Crystal Reports. Echter gaat veel van deze informatie over de .NET versie van Crystal Reports en is niet overal even duidelijk welke versie van Crystal Reports gebruikt wordt (er zijn grote verschillen qua mogelijkheden tussen de versies). Vooral informatie die specifiek gaat over de Eclipse plugin voor Crystal Reports is weinig over beschikbaar en moeilijk te vinden.

JasperReports

IDE integratie

Voor JasperReports is standaard geen IDE integratie beschikbaar, alle rapportages moeten gemaakt worden in de iReport designer. Er is een Eclipse plugin beschikbaar om iReport te gebruiken binnen Eclipse maar door niet aanwezige documentatie is het niet gelukt om deze werkend te krijgen.

Daardoor moeten ontworpen rapportages handmatig toegevoegd worden aan een project of webomgeving waar een applicatie draait. Daarnaast maakt deze opzet het ook onmogelijk om direct met POJO's te werken en te testen in de ontwikkelomgeving. Van deze POJO's moet de gecompileerde versie binnen iReport bekend gemaakt moeten worden om hiermee te testen.

Documentatie

Op de website van JasperReports is uitzonderlijk duidelijke documentatie beschikbaar over het gebruik van JasperReports. Er is zelfs informatie beschikbaar over het maken van Excel vriendelijke rapportages die ook op andere Rapportage tools van toepassing is. Onder andere duidelijke uitleg aan welke eisen een ontwerp van een rapportage dient te voldoen om deze zo goed mogelijk converteerbaar naar Excel te maken.

2.3 Centrale vraag 3: CV3

De centrale vraag CV3 is als volgt opgesteld:

“Welke aanbevelingen worden afgeleid uit de vergelijking van de vier onderzochte rapportage en charting API's?”

Om deze vraag te beantwoorden zijn de volgende bronnen geraadpleegd:

- BIRT 2.2 Project Plan http://www.eclipse.org/birt/phenix/project/project_plan_R2_2_0.php
- JasperReports Requirements
http://www.jasperforge.org/jaspersoft/opensource/business_intelligence/jasperreports/requirements.html

BIRT

De installatie en gebruik van de BIRT viewer is het meest complexe en gebruikersonvriendelijke onderdeel van de tool (dit in schril contrast met de BIRT Eclipse plugin die zeer gemakkelijk te installeren is). Er is documentatie aanwezig met daarin veel voorkomende scenario voorbeelden maar niet genoeg informatie om BIRT volledig in een applicatie te integreren. Dit is uiteindelijk te realiseren door goed de documentatie te lezen en gegevens uit de diverse voorbeelden te combineren. Als BIRT eenmaal geïntegreerd is in de applicatie zijn de rapportages gemakkelijk te gebruiken.

Het gebruik van de report designer van BIRT is de makkelijkste en gebruikersvriendelijkste van de onderzochte Reporting en Charting tools. De report designer heeft functionaliteiten op een logische en overzichtelijke manier ingedeeld. Deze bieden ook de verwachte functionaliteit voor het genereren van rapportages (net zoals alle andere onderzochte Reporting en Charting tools).

De documentatie van BIRT is een van de betere van de Reporting en Charting Tools (JasperReports is de beste). Het enige probleem dat bij de BIRT documentatie speelt is dat deze enig gefragmenteerd is over de BIRT website en daardoor zoekwerk nodig is om alle benodigde informatie te achterhalen. Ook speelt op het moment het probleem dat veel informatie nog geschreven is voor versie 2.1 van BIRT terwijl BIRT 2.2 al enige tijd gereleased is.

Sinds de release van versie 2.2 van BIRT is deze een echte serieuze Reporting en Charting tool geworden. Veel functionaliteiten zijn onder handen genomen door de ontwikkelaars om het gebruik er van te vergemakkelijken en te verbeteren. Ook zijn er nieuwe functionaliteiten toegevoegd zoals onder andere de mogelijkheid om rapportages te exporteren naar Excel of het gebruik van een extern CSS bestand voor het instellen van de opmaak.

Crystal Reports

De Crystal Reports designer for Eclipse is zeer moeilijk om te installeren door de afwezigheid van duidelijke documentatie die de afhankelijkheden van de plugin noemt. Dit moet tijdens de installatie en gebruik door de gebruiker zelf achterhaald worden en het ontbreken van een van de benodigde libraries kan zeer vreemd gedrag veroorzaken bij de designer.

Het gebruik van de Crystal Viewer in een project is daarentegen weer redelijk eenvoudig, ondanks de zeer slechte documentatie op dit gebied. Met zoekwerk op internet kan men redelijk eenvoudig vinden hoe de viewer gebruikt moet worden. En met behulp van het aanmaken van een Crystal Reports project en de daarin gebruikte libraries te kopiëren naar de te ontwikkelen applicatie is crystal reports te gebruiken. Alhoewel iets complexere handelingen dan het tonen van een rapportage in een viewer moeilijk te realiseren is door het gebrek aan documentatie (hierover zijn veel onopgeloste vragen te vinden op internet).

Het gebruik van de report designer is redelijk gemakkelijk en is nummer twee (na BIRT) qua gebruikersvriendelijkheid. De functionaliteiten zijn niet zo goed georganiseerd en overzichtelijk als bij BIRT (het vergt soms enig zoekwerk om een functie te vinden) maar is redelijk goed mee te werken.

Zoals alle andere Reporting en Charting tools biedt Crystal Reports de verwachte functionaliteiten voor het ontwerpen van rapportages.

De documentatie van Crystal Reports is de slechtste van alle onderzochte Reporting en Charting Tools. Deze is zeer gefragmenteerd en er is zeer weinig informatie te vinden over de Crystal Reports for Eclipse versie. Voor andere versies van Crystal Reports is wel enige documentatie te vinden maar deze is niet altijd direct toepasbaar binnen Crystal Reports for Eclipse door de grote verschillen qua mogelijkheden tussen de verschillende versies.

JasperReports

IReport de JasperReports designer is eenvoudig te installeren op een computer met de erbij geleverde installatie programma. Van alle Reporting en Charting tools is JasperReports de enige die niet standaard IDE integratie met Eclipse biedt. Er is in IReport wel een plugin directory voor het gebruik van IReport in Eclipse, echter is hier geen informatie over te vinden hoe deze geconfigureerd en geïnstalleerd moet worden (na de standaard plugin installatie is IReport niet te gebruiken in Eclipse).

De installatie van de viewer in een applicatie is zeer eenvoudig door de goede documentatie en door een paar voorbeelden die de basis laten zien over hoe een rapportage getoond kan worden.

IReport is het meest gebruikersonvriendelijk programma van de onderzochte Reporting en Charting tools. De functionaliteiten worden niet op een overzichtelijke en intuïtieve manier aangeboden aan de gebruiker. Voor veel standaard acties is het nodig om de documentatie te raadplegen om te achterhalen hoe iets precies gebruikt kan worden.

De documentatie van JasperReports is de beste van de onderzochte Reporting en Charting tools. Er is vaak duidelijke documentatie te vinden over hoe functionaliteit te gebruiken.

Op de volgende pagina treft u een overzicht van de eisen waaraan de onderzochte Reporting en Charting tools voldoen.

	BIRT	Crystal Reports	JasperReports
Ondersteunde invoer			
JDBC	√	√	√
XML	√	√	√
EJB	x ¹	x ²	√
POJO	√	√	√
Hibernate	√	√	√
Overige	√	√	√
Uitvoerformaten			
HTML	√	√	√
Excel	√	x ³	√
PDF	√	√	√
Overige	DOC, PPT, PostScript,	RTF, Word RTF, CSV	Excel, CSV, RTF, OpenOffice Document
Report delivery			
Web	√	√	√
Download	√	√	√
IDE integratie			
Eclipse	√	√	x
JBOSS IDE	√	√	x
MyEclipse	√	√	x
Anders	x	x	√ ⁴
Reports			
Ondersteunde charts	Bar, Line, Area, Pie, Meter, Scatter, Stock, Bubble, Difference, Gantt, Tube, Cone, Pyramid	Bar, Line, Area, Pie, Doughnut, Scatter, Radar, Bubble, Stock, Gauge, Gant, Funnel, Histogram	Bar, Line, Pie, Meter, Area, Scatter, Stock, Bubble, Thermometer
Kruistabel	√	√	√
Ondersteunen parameters	√	√	√
Overige			
Realtime	√	√	√
J2EE	√	√	√
Licentie	EPL	Commercial	LGPL
Kosten	€ 0,-	€ 0,-	€ 0,-

¹ Alleen mogelijk met een POJO die als interface dient tussen de EJB en BIRT

² Alleen mogelijk met een POJO die als interface dient tussen de EJB en Crystal Reports

³ Beschikbaar in de betaalde versie van Crystal Reports.

⁴ Er is een Eclipse plugin beschikbaar voor IReport, deze wordt niet vermeld doordat er geen documentatie beschikbaar is over installatie en gebruik waardoor het niet gelukt is om deze werkend te krijgen.

3 Conclusies en aanbevelingen

Alle onderzochte Reporting & Charting Tools zijn uitgebreid genoeg om te voorzien in een breed scala aan rapportages. Qua grafieken is de uitzondering JasperReports die het minste aantal soorten grafieken ondersteund waardoor bepaalde rapportages niet begeleid kunnen worden met een verduidelijkende grafiek.

De keuze voor een Reporting & Charting Tool wordt daarom voornamelijk gebaseerd op beschikbare documentatie en hoe gebruikersvriendelijk een Reporting & Charting tool is. Op dit gebied is van de onderzochte Reporting & Charting tools BIRT de beste keuze door een combinatie van factoren.

Van de onderzochte Reporting & Charting Tools biedt BIRT een gebruikersvriendelijke interface voor het ontwerpen. Met deze interface kan een beginnende programmeur vrij snel standaard rapportages ontwikkelen en zijn zelfs gevorderde functionaliteiten vrij snel te vinden en makkelijk te gebruiken.

De documentatie is niet de beste van de onderzochte Reporting & Charting Tools maar komt zeer dicht in de buurt van de kwaliteit van JasperReports. Dit in combinatie met een levendige community op het internet is er veel informatie te vinden.

De integratie van BIRT in een applicatie is redelijk complex door de hoeveelheid configuratiewerk dat verzet moet worden. De enige uitzondering hierop is Crystal Reports die zeer eenvoudig te installeren en te gebruiken is binnen een applicatie (wel met enig inventief gebruik van de Crystal Reports plugin door ontbrekende documentatie). Eenmaal geïntegreerd is de documentatie van BIRT weer zeer goed en is het gebruik van de rapportages ook zeer eenvoudig.

Daarom ook mijn advies aan ISAAC om gebruik te maken van BIRT voor het toevoegen van uitgebreide rapportage mogelijkheden in hun producten.

Literatuurlijst

Notities van gesprekken met Dhr. M. Hufkens, Dhr S. Daemen, Dhr. P. Luedke, Dhr K. Peters en Mw. K. Schengenga

Document van Eisen PTM - 20071106 - v0.6, Collin Maessen

PvA PTM - 20071112 - v1.3, Collin Maessen

Aantekeningen van oriëntatie op onderwerp

BIRT website <http://www.eclipse.org/birt/phoenix/>

Crystal Reports website <http://www.businessobjects.com/>

JasperReports website
http://www.jasperforge.org/jaspersoft/opensource/business_intelligence/jasperreports/

Onderzoek Reporting Tools, Nick Martens, 2006

Gesprek met Mw. A van Gastel, VCD Automatisering, Crystal Reports Reseller.

Using the BIRT Report Viewer <http://www.eclipse.org/birt/phoenix/deploy/viewerUsage2.2.php>

Crystal Reports, Create a report with Plain Old Java Objects (POJO)
http://devlibrary.businessobjects.com/businessobjectsxir2/en/en/JRC_SDK/jrc_java_dg_doc/doc/jrcsdk_java_dg/StartHere16.html

JasperReports Tutorial <http://struts.apache.org/2.x/docs/jasperreports-tutorial.html>

Creating a Report with JasperReports
http://www.regdeveloper.co.uk/2006/10/24/jasperreports_tutorial/

JasperReports overzicht pagina http://www.jaspersoft.com/JasperSoft_JasperReports.html


Crystal Reports for Eclipse Release Notes <http://diamond.businessobjects.com/node/447>

BIRT 2.2 Project Plan http://www.eclipse.org/birt/phoenix/project/project_plan_R2_2_0.php

JasperReports Requirements
http://www.jasperforge.org/jaspersoft/opensource/business_intelligence/jasperreports/requirements.html

Bijlage 4:
Document van Eisen

Document van Eisen PTM

Auteur:	Collin Maessen	 ISAAC
Datum:	7 november 2007	
Versie / status:	1.0 / Definitief	
File:	Document van Eisen PTM - 20071107 - v1.0.doc	

Versiebeheer

Datum	Versie	Belangrijkste wijzigingen	Auteur
12-09-2007	0.1	Eerste opzet.	Collin Maessen
28-09-2007	0.2	Eerste concept versie.	Collin Maessen
08-10-2007	0.3	Feedback verwerkt	Collin Maessen
25-10-2007	0.4	Informatie gesprekken verwerkt	Collin Maessen
01-11-2007	0.5	Feedback bedrijfsbegeleider verwerkt	Collin Maessen
06-11-2007	0.6	Enkele Tekstuele fouten hersteld	Collin Maessen
07-11-2007	1.0	Definitieve versie	Collin Maessen

Inhoudsopgave

1	INLEIDING	114
2	USE CASES	115
2.1	CONFIGURATIE VAN PTM.....	116
2.1.1	BTW-tarief instellen	116
2.1.2	Bedrijfsgegevens instellen.....	116
2.1.3	Standaard uurtarief wijzigen.....	117
2.1.4	Invoeren van nationale feestdag.....	117
2.1.5	Bewerken nationale feestdag.....	118
2.1.6	Verwijderen nationale feestdag.....	119
2.1.7	Medewerker toevoegen aan verlofdagen aanvraag notificatie	119
2.1.8	Medewerker verwijderen van verlofdagen aanvraag notificatie	120
2.1.9	Standaard verlofdagen voor medewerkers bewerken.....	120
2.2	PTM BEVEILIGING	121
2.2.1	Limiteren aantal inlogpogingen.....	121
2.2.2	Limiteren aantal inlogpogingen uitzetten.....	121
2.2.3	Limiteren aantal inlogpogingen bewerken.....	122
2.2.4	Blokkering ip-adres opheffen	122
2.2.5	Blokkering gebruikersprofiel opheffen.....	123
2.3	RECHTEN	124
2.3.1	Rol aanmaken.....	124
2.3.2	Rol bewerken.....	124
2.3.3	Rol verwijderen	125
2.3.4	Rol toekennen aan medewerker.....	125
2.3.5	Rol van medewerker wijzigen.....	126
2.3.6	Inloggen	127
2.3.7	Uitloggen.....	127
2.4	MEDEWERKERS	128
2.4.1	Medewerker toevoegen.....	128
2.4.2	Medewerker bewerken.....	129
2.4.3	Medewerker verwijderen.....	129
2.4.4	Medewerker inactiveren.....	130
2.4.5	Medewerker activeren	130
2.4.6	Medewerker aan project toevoegen	130
2.4.7	Medewerker aan subproject toevoegen.....	131
2.5	PERIODES.....	132
2.5.1	Periode toevoegen.....	132
2.5.2	Periode verwijderen	132
2.5.3	Periode bewerken.....	133
2.5.4	Periode markeren als niet te factureren.....	134
2.5.5	Periode markeren als te factureren.....	134
2.6	KLANTEN.....	135
2.6.1	Klant toevoegen.....	135
2.6.2	Klant bewerken.....	136
2.6.3	Klant verwijderen.....	137
2.6.4	Klant inactiveren.....	137
2.6.5	Klant activeren	138
2.6.6	Aangepaste klantenlijst tonen.....	138
2.7	CONTACTPERSONEN	139
2.7.1	Contactpersoon aan klant toevoegen	139
2.7.2	Contactpersoon van klant verwijderen.....	139
2.7.3	Contactpersoon van klant bewerken	140
2.8	AFDELINGEN.....	141
2.8.1	Afdeling toevoegen aan klant	141
2.8.2	Afdeling van klant bewerken	141

2.8.3	Afdeling van klant verwijderen.....	142
2.9	ACCOUNTMANAGERS.....	143
2.9.1	Accountmanager toewijzen aan klant.....	143
2.9.2	Accountmanager van klant verwijderen.....	143
2.10	PROJECTEN	144
2.10.1	Project aanmaken.....	144
2.10.2	Project bewerken	145
2.10.3	Project verwijderen	146
2.10.4	Project inactiveren	146
2.10.5	Project activeren	147
2.10.6	Project markeren als gefactureerd.....	147
2.10.7	Project markeren als niet gefactureerd.....	148
2.10.8	Document toevoegen aan project	148
2.10.9	Document verwijderen uit project.	149
2.11	SUBPROJECT	150
2.11.1	Subproject aanmaken	150
2.11.2	Subproject bewerken	150
2.11.3	Subproject verwijderen.....	151
2.12	TAKEN	152
2.12.1	Taak aanmaken.....	152
2.12.2	Taak bewerken.....	152
2.12.3	Taak toekennen aan (sub)project	153
2.12.4	Taak verwijderen.....	153
2.13	FACTUREREN	154
2.13.1	Projectkosten dashboard.....	154
2.13.2	Factuur toevoegen.....	155
2.13.3	Factuur bewerken.....	155
2.13.4	Factuur verwijderen.....	156
2.13.5	Factuur markeren als betaald	156
2.13.6	Factuur markeren als niet betaald	157
2.14	EXTERNE KOSTEN	158
2.14.1	Externe kosten toevoegen aan (sub)project.....	158
2.14.2	Externe kosten bewerken	158
2.14.3	Externe kosten verwijderen	159
2.14.4	Externe kosten markeren als niet factureerbaar.....	159
2.14.5	Externe kosten markeren als factureerbaar	160
2.15	VERLOFDAGEN	161
2.15.1	Verlofdag indienen	161
2.15.2	Verlofdag verwijderen.....	161
2.15.3	Verlofdag bewerken.....	162
2.15.4	Verlofdag goedkeuren.	163
2.15.5	Opvragen overzichtverlofdagen	163
2.15.6	Opvragen overzichtverlofdagen voor alle medewerkers	164
2.15.7	Verlofdagen overzetten naar huidig kalenderjaar.....	164
2.15.8	Verlofdagen aanmaken voor kalenderjaar	165
2.16	ZIEKMELDING	166
2.16.1	Ziekmelding invoeren	166
2.16.2	Ziekmelding bewerken.....	167
2.16.3	Ziekmelding verwijderen	168
2.17	NOTIFICATIE	169
2.17.1	Deadline dreigt overschreden te worden.....	169
2.17.2	Urentoekenning dreigt overschreden te worden.....	170
3	RAPPORTAGE	171
4	NIET FUNCTIONELE EISEN	172
4.1	SOFTWARE EISEN	172
4.1.1	Toegankelijkheid	172
4.1.2	Beveiliging	172

4.2	KWALITEITEISEN	172
4.2.1	<i>Betrouwbaarheid</i>	172
4.2.2	<i>Onderhoudbaarheid</i>	173
4.2.3	<i>Uitbreidbaarheid</i>	173
4.2.4	<i>Gebruikersvriendelijkheid</i>	173

1 Inleiding

Het Document van Eisen geeft een beschrijving van de functionaliteit die in het systeem geïmplementeerd dienen te worden. Deze eisen zijn omschreven in use cases die werking van de functionaliteiten beschrijven.

Welke van de use cases geïmplementeerd worden en wanneer staat beschreven in het Plan van Aanpak. Deze beschrijving zal samen met de globale planning in het Plan van Aanpak een gedetailleerde planning opleveren met milestones en deadlines.

2 Use Cases

Use cases beschrijven de algemene stappen die doorlopen worden voor het gebruik van bepaalde functionaliteit binnen PTM.

Als actor worden alleen beheerder en medewerker genoemd. Beheerder geeft aan welke functionaliteiten toegankelijk zijn als beheerder van PTM. De actor medewerker geeft aan welke functionaliteiten beschikbaar zijn voor medewerkers van het systeem. Tot welke functionaliteiten een medewerker toegang heeft wordt bepaald door de rol die de gebruiker binnen het systeem heeft.

Use cases voor rapportages zijn niet uitgewerkt in het huidige document van eisen. Dit omdat de rapportage use cases beïnvloed kunnen worden door de resultaten die uit het onderzoek komen. Zie hoofdstuk 3 voor details.

Tevens zijn bepaalde use cases niet uitgewerkt. Het tonen van standaard lijsten die gebruikt worden in de GUI zijn hier een voorbeeld van. Dit is gedaan om de use cases in het document te beperken tot die waarover een beslissing genomen moet worden over of ze geïmplementeerd worden en hoe een use case dient te verlopen.

Er is ook op bepaalde punten afgeweken van de standaard notaties die gebruikt worden voor use cases. Zo zijn bijvoorbeeld zeer complexe use cases niet volledig uitgewerkt. In die use cases wordt bijvoorbeeld verwezen naar andere use cases of er wordt een uitgebreide uitleg gegeven in de notities veld van de use case. Dit om te leesbaarheid van dergelijke use cases te vergroten en de tijd die nodig is voor het schrijven van het Document van Eisen te beperken.

2.1 Configuratie van PTM

2.1.1 BTW-tarief instellen

Versie	1.1
Samenvatting	Hiermee stelt de gebruiker het BTW-tarief in dat binnen PTM gebruikt wordt tijdens het factureren van projecten.
Gebruiksfrequentie	Zeer zelden
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker wijzigt het BTW-tarief
Stappenplan	<ol style="list-style-type: none"> 1. Voer nieuw BTW-tarief in. 2. Systeem slaat het BTW-tarief op.
Alternatief stappenplan	-
Postcondities	BTW tarief is gewijzigd.
Uitzondering	-
Notities	-

2.1.2 Bedrijfsgegevens instellen

Versie	1.1
Samenvatting	Met het instellen van de bedrijfsgegevens wordt de naam van het bedrijf ingevoerd en indien gewenst de afdeling waar PTM gebruikt wordt (zie notitie voor reden).
Gebruiksfrequentie	Zeer zelden
Complexiteit	Laag
Prioriteit	Won't have
Actors	Beheerder
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker stelt de bedrijfsgegevens in
Stappenplan	<ol style="list-style-type: none"> 1. Voer bedrijfsnaam in. 2. Voer afdelingsnaam in.
Alternatief stappenplan	Bedrijfsnaam is leeg: <ol style="list-style-type: none"> 1. Systeem geeft de melding dat de bedrijfsnaam niet ingevuld is. 2. Ga naar stap 1 (stappenplan).
Postcondities	Bedrijfsgegevens zijn ingesteld.
Uitzondering	Bedrijfsnaam is leeg.
Notities	De use case bedrijfsgegevens instellen is toegevoegd om het mogelijk te maken om een rapportage programma te schrijven. Het rapportage programma gebruikt dan de gegevens van verschillende bedrijven en/of afdelingen die in de databases zijn opgeslagen. De bedrijfsgegevens worden dan gebruikt als identificatie van de bron van de gegevens. (op verzoek van Paul Luedke toegevoegd)

2.1.3 Standaard uurtarief wijzigen

Versie	1.1
Samenvatting	Standaard uurtarief wordt binnen PTM gebruikt voor de berekeningen van de uurloonposten op de facturen van klanten. Tevens wordt het gebruikt binnen PTM voor het genereren van rapportages. Met het wijzigen van het standaard uurtarief kan deze ingesteld worden.
Gebruiksfrequentie	Zeer zelden
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder
Precondities	Gebruiker is geautoriseerd.
Triggers	Standaard uurtarief wordt aangezet
Stappenplan	<ol style="list-style-type: none"> 1. Voer standaard uurtarief in. 2. Systeem slaat standaard uurtarief in. 3. Systeem slaat instelling op.
Alternatief stappenplan	-
Postcondities	Uurtarief is ingesteld.
Uitzondering	-
Notities	-

2.1.4 Invoeren van nationale feestdag

Versie	1.1
Samenvatting	De gebruiker voert een nationale feestdag in waarop alle werknemers automatisch verlof hebben. Een nationale feestdag gaat niet af van het aantal verlofdagen van een werknemer.
Gebruiksfrequentie	Zelden
Complexiteit	Gemiddeld
Prioriteit	Won't have
Actors	Beheerder
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker voegt een nieuwe nationale feestdag toe.
Stappenplan	<ol style="list-style-type: none"> 1. Voer naam in voor de nationale feestdag. 2. Kies de datum voor de nationale feestdag. 3. Voer omschrijving in. 4. Systeem slaat de nationale feestdag op. 5. Systeem verwijdert verlofdagen die samenvallen met de nationale feestdag.
Alternatief stappenplan	<p>Algemene dag met dezelfde naam bestaat al:</p> <ol style="list-style-type: none"> 2. Systeem geeft een melding dat de naam al in gebruik is. 3. Ga naar stap 1 (stappenplan). <p>Verlofdag van werknemer valt samen met de nationale feestdag.</p> <ol style="list-style-type: none"> 2. Systeem geeft de melding dat de nationale feestdag samenvalt met de verlofdag van <<aantal>> medewerkers. Deze zullen tijdens het opslaan van de nationale feestdag verwijdert worden. 3. Ga naar stap 3 (stappenplan).
Postcondities	Er is een nationale feestdag toegevoegd aan het systeem

Uitzondering	Een nationale feestdag met dezelfde naam bestaat al. Verlofdag van minimaal een medewerker valt samen met de nationale feestdag..
Notities	Nationale feestdagen zullen voor de komende 30 jaar beschikbaar zijn binnen het systeem. Conflicten tussen nationale feestdagen zullen door de gebruiker gecorrigeerd moeten worden zodra de overheid een beslissing heeft genomen over het onderwerp: http://www.sofim.info/index.php?s=-1&snr=311

2.1.5 Bewerken nationale feestdag

Versie	1.1
Samenvatting	Hiermee kan de nationale feestdag bewerkt worden.
Gebruiksfrequentie	Zeer zelden
Complexiteit	Gemiddeld
Prioriteit	Won't have
Actors	Beheerder
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker bewerkt een nationale feestdag
Stappenplan	<ol style="list-style-type: none"> 1. Voer naam in voor de nationale feestdag. 2. Systeem controleert of de naam beschikbaar is. 3. Kies de datum voor de nationale feestdag. 4. Systeem controleert of er toekomstige verlofdagen van werknemers samenvallen met deze datum. 5. Voer omschrijving in. 6. Systeem slaat de nationale feestdag op. 7. Systeem verwijdert verlofdagen die samenvallen met de nationale feestdag.
Alternatief stappenplan	<p>Algemene dag met dezelfde naam bestaat al:</p> <ol style="list-style-type: none"> 2. Systeem geeft een melding dat de naam al in gebruik is. 3. Ga naar stap 1 (stappenplan). <p>Verlofdag van werknemer valt samen met de nationale feestdag.</p> <ol style="list-style-type: none"> 4. Systeem geeft de melding dat de nationale feestdag samenvalt met de verlofdag van <aantal> medewerkers. Deze zullen tijdens het opslaan van de nationale feestdag verwijderd worden. 5. Ga naar stap 5 (stappenplan)
Postcondities	De nieuwe gegevens van de nationale feestdag zijn opgeslagen.
Uitzondering	Een nationale feestdag met eenzelfde naam bestaat al.
Notities	Bij verplaatsing worden al verwijderde verlofdagen niet teruggeplaatst.

2.1.6 Verwijderen nationale feestdag

Versie	1.1
Samenvatting	Hiermee kan een aangemaakte nationale feestdag verwijderd worden.
Gebruiksfrequentie	Zeer zelden
Complexiteit	Laag
Prioriteit	Won't have
Actors	Beheerder
Precondities	Gebruiker is geautoriseerd. Nationale feestdag bestaat.
Triggers	Gebruiker verwijdert nationale feestdag
Stappenplan	1. Systeem verwijdert nationale feestdag.
Alternatief stappenplan	-
Postcondities	Nationale feestdag is verwijderd
Uitzondering	-
Notities	Verwijderde verlofdagen worden niet teruggezet als de nationale feestdag verwijderd wordt.

2.1.7 Medewerker toevoegen aan verlofdagen aanvraag notificatie

Versie	1.1
Samenvatting	Door het toevoegen van een gebruiker aan de verlofdagen aanvraag notificatie krijgt de gebruiker de verlofdagen aanvraag notificaties en kan de verlofdagen op definitief zetten of afgekeurd.
Gebruiksfrequentie	Zeer zelden
Complexiteit	Laag
Prioriteit	Won't have
Actors	Beheerder
Precondities	Gebruiker is geautoriseerd. Gebruiker is nog niet toegevoegd aan de verlofdagen aanvraag notificatie.
Triggers	Gebruiker voegt een nieuwe medewerker toe aan verlofdagen aanvraag notificatie.
Stappenplan	1. Kies medewerker. 2. Systeem voegt medewerker toe aan verlofdagen aanvraag notificatie.
Alternatief stappenplan	-
Postcondities	Medewerker is toegevoegd aan de verlofdagen aanvraag notificatie.
Uitzondering	-
Notities	-

2.1.8 Medewerker verwijderen van verlofdagen aanvraag notificatie

Versie	1.1
Samenvatting	Door het verwijderen van een medewerker van de notificatielijst krijgt de medewerker geen notificaties meer en kan niet meer verlofdagen van andere medewerkers goedkeuren of afkeuren.
Gebruiksfrequentie	Zeer zelden
Complexiteit	Laag
Prioriteit	Won't have
Actors	Beheerder
Precondities	Gebruiker is geautoriseerd. Er zijn minimaal 2 medewerkers die de notificatie ontvangen.
Triggers	Gebruiker verwijdert medewerker uit de verlofdagen aanvraag notificatie
Stappenplan	1. Systeem verwijdert de medewerker uit de verlofdagen aanvraag notificatie.
Alternatief stappenplan	-
Postcondities	Medewerker ontvangt geen verlofdagen aanvraag notificaties meer. Gebruiker kan geen verlofdagen aanvraag notificaties goedkeuren en afkeuren.
Uitzondering	-
Notities	-

2.1.9 Standaard verlofdagen voor medewerkers bewerken

Versie	1.1
Samenvatting	Door het veranderen van de standaard verlofdagen krijgen de werknemers meer of minder verlofdagen als deze aangemaakt worden voor een jaar.
Gebruiksfrequentie	Zeer zelden
Complexiteit	Laag
Prioriteit	Won't have
Actors	Beheerder
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker wijzigt de standaard verlofdagen.
Stappenplan	Voer het standaard aantal verlofdagen in.
Alternatief stappenplan	-
Postcondities	Standaard aantal verlofdagen voor medewerkers is gewijzigd.
Uitzondering	-
Notities	Nieuwe instellingen hebben alleen invloed op nieuwe medewerkers en het aanmaken van verlofdagen voor een nieuw kalenderjaar.

2.2 PTM beveiliging

2.2.1 Limiteren aantal inlogpogingen

Versie	1.1
Samenvatting	De gebruiker zet de beveiligingsoptie limiteren van het aantal inlogpogingen aan en voert het gedrag van het systeem in als de limiet gepasseerd wordt.
Gebruiksfrequentie	Zelden
Complexiteit	Laag
Prioriteit	Could have
Actors	Beheerder
Precondities	Gebruiker is geautoriseerd
Triggers	Gebruiker zet de optie limiteren aantal inlogpogingen aan
Stappenplan	<ol style="list-style-type: none"> 1. Zet limiteren aantal inlogpogingen aan 2. Voer het maximum aantal inlogpogingen in. 3. Zet blokkering gebruikersprofielen en/of ip-adres aan. 4. Systeem slaat de instellingen op
Alternatief stappenplan	<p>Geen gebruikersblokkering en ip-adres blokkering ingesteld.</p> <ol style="list-style-type: none"> 4. Ga naar stap 3
Postcondities	Aantal inlogpogingen voor gebruikersprofiel en/of ip-adres zijn gelimiteerd.
Uitzondering	De opties blokkeren van gebruikersprofiel en blokkeren van ip-adres staan beiden op uit
Notities	Heeft gevolgen voor de complexiteit van de use case inloggen

2.2.2 Limiteren aantal inlogpogingen uitzetten

Versie	1.1
Samenvatting	De gebruiker zet het aantal inlogpogingen uit. Oude instellingen en blokkeringen worden bewaard in het systeem . Als het limiteren van inlogpogingen weer geactiveerd wordt gebruikt het systeem de oude instellingen.
Gebruiksfrequentie	Zelden
Complexiteit	Laag
Prioriteit	Could have
Actors	Beheerder
Precondities	Gebruiker is geautoriseerd
Triggers	Gebruiker zet de optie limiteren aantal inlogpogingen uit
Stappenplan	<ol style="list-style-type: none"> 1. Systeem slaat de instellingen op
Alternatief stappenplan	-
Postcondities	Aantal inlogpogingen zijn niet meer gelimiteerd.
Uitzondering	-
Notities	Heeft gevolgen voor de complexiteit van de use case inloggen.

2.2.3 Limiteren aantal inlogpogingen bewerken

Versie	1.1
Samenvatting	De gebruiker kan in deze use case het aantal maximum inlogpogingen veranderen en de blokkering van gebruikersprofiel en/of ip-adres veranderen.
Gebruiksfrequentie	Zelden
Complexiteit	Laag
Prioriteit	Could have
Actors	Medewerker, directie
Precondities	Gebruiker is geautoriseerd
Triggers	Gebruiker bewerkt de optie limiteren aantal inlogpogingen.
Stappenplan	<ol style="list-style-type: none"> 1. Voer het maximum aantal inlogpogingen in. 2. Zet blokkering gebruikerprofiel en/of ip-adres aan. 3. Systeem verifieert instellingen 4. Systeem slaat de instellingen op
Alternatief stappenplan	<p>Geen gebruikersblokkering en ip-adres blokkering ingesteld:</p> <ol style="list-style-type: none"> 1. Ga naar stap 2 <p>Er zijn oude instellingen beschikbaar:</p> <ol style="list-style-type: none"> 1. Systeem vult oude waardes automatisch in. 2. Ga naar stap 1 (stappenplan)
Postcondities	Aantal inlogpogingen voor gebruikersprofiel en/of ip-adres zijn gelimiteerd.
Uitzondering	De opties blokkeren van gebruikersprofiel en blokkeren van ip-adres staan beiden op uit Er zijn al oude instellingen beschikbaar.
Notities	-

2.2.4 Blokkering ip-adres opheffen

Versie	1.1
Samenvatting	De gebruiker heft de blokkering van het ip-adres op in het systeem. Dit zorgt ervoor dat gebruikers vanaf dat ip-adres weer kunnen inloggen.
Gebruiksfrequentie	Soms
Complexiteit	Laag
Prioriteit	Could have
Actors	Beheerder
Precondities	Gebruiker is geautoriseerd Er is een blokkering op het ip-adres
Triggers	Gebruiker heft de blokkering van het ip-adres op
Stappenplan	<ol style="list-style-type: none"> 1. Systeem verwijdert de blokkering
Alternatief stappenplan	-
Postcondities	Blokkering op het Ip-adres is verwijderd.
Uitzondering	-
Notities	Heeft gevolgen voor de complexiteit van de use case inloggen.

2.2.5 Blokkering gebruikersprofiel opheffen

Versie	1.1
Samenvatting	De gebruiker heft de blokkering voor de gebruikersprofiel op in het systeem. Dit zorgt ervoor dat de gebruiker weer kan inloggen.
Gebruiksfrequentie	Soms
Laag	Laag
Prioriteit	Could have
Actors	Beheerder
Precondities	Gebruiker is geautoriseerd Er is een blokkering op het gebruikersprofiel
Triggers	Gebruiker heft de blokkering van het gebruikersprofiel op
Stappenplan	1. Systeem verwijdert de blokkering
Alternatief stappenplan	-
Postcondities	Blokkering op de gebruikersprofiel is verwijderd.
Uitzondering	-
Notities	Heeft gevolgen voor de complexiteit van de use case inloggen.

2.3 Rechten

2.3.1 Rol aanmaken

Versie	1.1
Samenvatting	Met het aanmaken van een rol kan de gebruiker rechten toekennen aan een rol zodat medewerkers met deze rol de toegekende functionaliteiten kunnen gebruiken..
Gebruiksfrequentie	Zeer zelden
Complexiteit	Hoog
Prioriteit	Must have
Actors	Beheerder
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker maakt een nieuwe rol aan
Stappenplan	<ol style="list-style-type: none"> 1. Voer naam rol in 2. Voer omschrijving rol in 3. Kies rolrechten. 4. Systeem slaat de rol op.
Alternatief stappenplan	Naam rol bestaat al: <ol style="list-style-type: none"> 1. Systeem geeft de melding dat de naam al in gebruik is. 2. Ga naar stap 1 (stappenplan)
Postcondities	Er is een nieuwe rol bekend in het systeem.
Uitzondering	Naam rol bestaat al
Notities	-

2.3.2 Rol bewerken

Versie	1.1
Samenvatting	Met het bewerken van een rol kan gewijzigd worden tot welke functionaliteiten een rol toegang tot heeft.
Gebruiksfrequentie	Zeer zelden
Complexiteit	Hoog
Prioriteit	Must have
Actors	Beheerder
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker bewerkt een rol
Stappenplan	<ol style="list-style-type: none"> 1. Voer naam rol in 2. Voer omschrijving rol in 3. Kies rolrechten. 4. Systeem slaat de nieuwe gegevens op.
Alternatief stappenplan	Naam rol bestaat al: <ol style="list-style-type: none"> 1. Systeem geeft de melding dat de naam al in gebruik is. 2. Ga naar stap 1 (stappenplan) Er zijn oude instellingen beschikbaar: <ol style="list-style-type: none"> 1. Systeem vult oude waardes automatisch in. 2. Ga naar stap 1 (stappenplan).
Postcondities	De nieuwe rechten zijn toegekend aan de rol
Uitzondering	Naam rol bestaat al. Er zijn oude instellingen beschikbaar.
Notities	-

2.3.3 Rol verwijderen

Versie	1.1
Samenvatting	Door het verwijderen van een rol is deze niet meer beschikbaar binnen het systeem.
Gebruiksfrequentie	Zeer zelden
Complexiteit	Hoog
Prioriteit	Must have
Actors	Beheerder
Precondities	Gebruiker is geautoriseerd. Rol is niet in gebruik
Triggers	Gebruiker verwijdert de rol
Stappenplan	1. Systeem verwijdert de rol.
Alternatief stappenplan	-
Postcondities	Rol is verwijderd.
Uitzondering	-
Notities	-

2.3.4 Rol toekennen aan medewerker

Versie	1.1
Samenvatting	Met het toekennen van een rol aan de medewerker kan de gebruiker inloggen op het systeem en gebruik maken van de functionaliteit waarop hij geautoriseerd is.
Gebruiksfrequentie	Soms
Complexiteit	Gemiddeld
Prioriteit	Must have
Actors	Beheerder
Precondities	Gebruiker is geautoriseerd. Medewerker bestaat. Medewerker is niet toegekend aan een rol
Triggers	Gebruiker maakt rechten aan voor de medewerker.
Stappenplan	1. Kies rol voor de gebruiker. 2. Systeem kent de rol toe aan de gebruiker.
Alternatief stappenplan	-
Postcondities	Rol is aan de medewerker toegekend
Uitzondering	-
Notities	-

2.3.5 Rol van medewerker wijzigen

Versie	1.1
Samenvatting	Met het wijzigen van de rol van de gebruiker in het systeem veranderen de functionaliteit waarop hij is geautoriseerd.
Gebruiksfrequentie	Soms
Complexiteit	Gemiddeld
Prioriteit	Must have
Actors	Beheerder
Precondities	Gebruiker is geautoriseerd. Medewerker bestaat. Rol bestaat.
Triggers	Gebruiker bewerkt rechten van de medewerker.
Stappenplan	<ol style="list-style-type: none"> 1. Kies de nieuwe rol voor de gebruiker. 2. Systeem kent de nieuwe rol toe aan de gebruiker.
Alternatief stappenplan	-
Postcondities	Nieuwe rol is toegekend aan de gebruiker.
Uitzondering	-
Notities	-

2.3.6 Inloggen

Versie	1.1
Samenvatting	Deze use-case beschrijft hoe gebruikers inloggen in PTM.
Gebruiksfrequentie	Vaak
Complexiteit	Hoog
Prioriteit	Must have
Actors	Gebruiker
Precondities	Ip-adres is niet geblokkeerd.
Triggers	Gebruiker is niet geauthentiseerd
Stappenplan	<ol style="list-style-type: none"> 1. Gebruiker start PTM 2. Gebruiker voert gebruikersnaam en wachtwoord in 3. Het systeem verifieert de gebruikersnaam en wachtwoord 4. Het systeem logt de gebruiker in.
Alternatief stappenplan	<p>Fout wachtwoord:</p> <ol style="list-style-type: none"> 1. Gebruiker voert wachtwoord in 2. Ga naar stap 3 (stappenplan) <p>Gebruikersprofiel bestaat niet:</p> <ol style="list-style-type: none"> 1. Ga naar stap 2 (stappenplan) <p>Over inlogpogingen limiet:</p> <ol style="list-style-type: none"> 1. Het systeem geeft dat de gebruikersprofiel en/of het ip-adres geblokkeerd is (instelling afhankelijk). 2. Ga naar stap 2 (stappenplan)
Postcondities	Gebruiker is geautoriseerd
Uitzondering	<p>Gebruiker voert een fout wachtwoord in voor de gebruikersprofiel.</p> <p>Gebruikerprofiel bestaat niet.</p> <p>De gebruiker gaat over het aantal toegestane inlogpogingen heen.</p>
Notities	Complexiteit voor deze use case is hoog in verband met dat de gebruiker hier geauthenticeerd en geautoriseerd wordt. Daarnaast moet tijdens implementatie ook rekening gehouden worden met technische beveiligingseisen.

2.3.7 Uitloggen

Versie	1.1
Samenvatting	Als de gebruiker uitlogt heeft hij geen toegang meer tot de functionaliteiten van het systeem.
Gebruiksfrequentie	Vaak
Complexiteit	Gemiddeld
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	<p>Gebruiker is geautoriseerd.</p> <p>Gebruiker is ingelogd</p>
Triggers	Gebruiker logt uit.
Stappenplan	<ol style="list-style-type: none"> 1. Systeem logt gebruiker uit.
Alternatief stappenplan	-
Postcondities	Gebruiker is uitgelogd
Uitzondering	-
Notities	-

2.4 Medewerkers

2.4.1 Medewerker toevoegen

Versie	1.2
Samenvatting	Met medewerker toevoegen wordt een nieuwe medewerker aan het systeem toegevoegd. Een medewerker wordt ook gelijk gebruikt als gebruikersprofiel voor de gebruiker.
Gebruiksfrequentie	Soms
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd
Triggers	Gebruiker voegt een nieuwe medewerker toe
Stappenplan	<ol style="list-style-type: none"> 1. Voer voornaam in 2. Voer wachtwoord in. 3. Voer voorletters in. 4. Voer tussenvoegsels in. 5. Voer achternaam in. 6. Voer geboortedatum. 7. Voer identiteitsbewijs, paspoort- of rijbewijsnummer in. 8. Voer verloopdatum in 9. Voeg fotokopie toe. 10. Voer adres in. 11. Voer postcode in. 12. Voer woonplaats in. 13. Voer telefoonnummer in. 14. Voer gsm nummer in. 15. Voer e-mail adres in. 16. Voer parttime percentage in. 17. Voer omschrijving in. 18. Systeem maakt de medewerker aan in het systeem 19. Systeem slaat de ingevoerde medewerkergegevens op.
Alternatief stappenplan	<p>Er is al een gebruikersprofiel met dezelfde naam:</p> <ol style="list-style-type: none"> 1. Systeem geeft een melding dat er al een gebruikersprofiel met eenzelfde naam bestaat. 2. Voer gebruikersnaam in 3. Ga naar stap 2 (stappenplan)
Postcondities	Er is een nieuwe medewerker toegevoegd aan het systeem
Uitzondering	Gebruikersnaam is al in gebruik.
Notities	<p>2 maanden voordat het paspoort of rijbewijs verloopt notificatie naar Max, Karlijn en de medewerker.</p> <p>Misschien voorkomen dat een medewerker kan inloggen als zijn identiteitsbewijs, paspoort- of rijbewijs is verlopen.</p>

2.4.2 Medewerker bewerken

Versie	1.1
Samenvatting	Met de use case medewerker bewerken kunnen oude gegevens worden gewijzigd en nieuwe gegevens toegevoegd worden.
Gebruiksfrequentie	Soms
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Medewerker bestaat in het systeem.
Triggers	Gebruiker bewerkt een medewerker.
Stappenplan	<ol style="list-style-type: none"> 1. Voer voornaam in 2. Voer voorletters in. 3. Voer tussenvoegsels in. 4. Voer achternaam in. 5. Voer adres in. 6. Voer postcode in. 7. Voer woonplaats in. 8. Voer telefoonnummer in. 9. Voer gsm nummer in. 10. Voer e-mail adres in. 11. Voer parttime percentage in. 12. Voer omschrijving in. 13. Systeem slaat de nieuwe medewerker gegevens op in het systeem.
Alternatief stappenplan	-
Postcondities	De nieuwe medewerkergegevens zijn opgeslagen
Uitzondering	-
Notities	-

2.4.3 Medewerker verwijderen

Versie	1.1
Samenvatting	Met de use case medewerker verwijderen wordt de medewerker als verwijderd aangemerkt en is niet meer beschikbaar binnen medewerker beheer. Tevens kan de gebruiker door deze actie niet meer inloggen op het systeem.
Gebruiksfrequentie	Zeer zelden
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Medewerker bestaat
Triggers	Gebruiker verwijdert een medewerker
Stappenplan	<ol style="list-style-type: none"> 1. Systeem inactieveert medewerker (zie use case medewerker inactiveren). 2. Systeem verwijdert de medewerker.
Alternatief stappenplan	-
Postcondities	Medewerker is uit het systeem verwijderd.
Uitzondering	-
Notities	Medewerkers worden niet daadwerkelijk verwijderd maar als zodanig gemerkt in de database. Dit voorkomt problemen met gegevens die medewerkers aanmaken in het systeem.

2.4.4 Medewerker inactiveren

Versie	1.1
Samenvatting	Door een medewerker te inactiveren wordt deze gemerkt als inactief en komt niet meer voor in de medewerkerlijsten. Tegelijkertijd zorgt het inactief maken van een medewerker ervoor dat deze medewerker niet meer kan inloggen op het systeem.
Gebruiksfrequentie	Zelden
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Medewerker bestaat.
Triggers	Gebruiker verwijdert een medewerker
Stappenplan	1. Systeem inactieveert de medewerker.
Alternatief stappenplan	-
Postcondities	Medewerker is geïnactiveerd.
Uitzondering	-
Notities	-

2.4.5 Medewerker activeren

Versie	1.1
Samenvatting	Door een medewerker te activeren wordt deze gemerkt als actief en komt daardoor weer voor in de medewerkerlijsten. Tegelijkertijd zorgt het actief maken van een medewerker ervoor dat deze medewerker weer kan inloggen op het systeem.
Gebruiksfrequentie	Zelden
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Medewerker bestaat.
Triggers	Gebruiker verwijdert een medewerker
Stappenplan	1. Systeem inactieveert de medewerker.
Alternatief stappenplan	-
Postcondities	Medewerker is geïnactiveerd.
Uitzondering	-
Notities	-

2.4.6 Medewerker aan project toevoegen

Versie	1.1
Samenvatting	Door het toevoegen van een medewerker aan een project kan de medewerker zijn gewerkte uren toevoegen aan het project.
Gebruiksfrequentie	Soms
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker wordt toegevoegd aan een project

Stappenplan	<ol style="list-style-type: none"> 1. Kies project. 2. Voer beschikbare uren in. 3. Systeem wijst project toe aan de medewerker.
Alternatief stappenplan	Medewerker is al toegewezen aan de project: <ol style="list-style-type: none"> 2. Systeem geeft de melding dat de medewerker al toegevoegd is aan de gekozen project. 3. Ga naar stap 1 (stappenplan)
Postcondities	Medewerker is toegevoegd aan het project
Uitzondering	Gebruiker is al toegevoegd aan de gekozen project
Notities	Beschikbare uren wordt gebruikt voor rapportages en notificaties.

2.4.7 Medewerker aan subproject toevoegen

Versie	1.1
Samenvatting	Door het toevoegen van een medewerker aan een subproject kan de medewerker zijn gewerkte uren toevoegen aan het subproject.
Gebruiksfrequentie	Soms
Complexiteit	Laag
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Project heeft subprojecten
Triggers	Gebruiker wordt toegevoegd aan een subproject
Stappenplan	<ol style="list-style-type: none"> 1. Kies project. 2. Kies subproject. 3. Voer beschikbare uren in. 4. Systeem wijst subproject toe aan de medewerker.
Alternatief stappenplan	Medewerker is al toegewezen aan de subproject: <ol style="list-style-type: none"> 2. Systeem geeft de melding dat de medewerker al toegevoegd is aan de gekozen subproject. 3. Ga naar stap 2 (stappenplan)
Postcondities	Medewerker is toegevoegd aan het subproject
Uitzondering	Gebruiker is al toegevoegd aan de gekozen subproject
Notities	-

2.5 Periodes

Afhankelijk van de rechten die een gebruiker heeft gekregen is het mogelijk om periodes van andere werknemers in te zien, te bewerken of toe te voegen.

2.5.1 Periode toevoegen

Versie	1.1
Samenvatting	Door het toevoegen van een periode aan een project worden de ingevoerde uren gebruikt voor facturatie en rapportages binnen PTM.
Gebruik frequentie	Vaak
Complexiteit	Gemiddeld
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. (sub)project is actief.
Triggers	Gebruiker maakt een nieuwe periode aan.
Stappenplan	<ol style="list-style-type: none"> 1. Kies taak. 2. Voer datum in. 3. Voer starttijd in. 4. Voer eindtijd in. 5. Voer omschrijving in. 6. Systeem voegt periode toe aan project.
Alternatief stappenplan	<p>Periode valt op een nationale feestdag:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat de periode op een nationale feestdag valt. 2. Ga naar stap 2 (stappenplan). <p>Omschrijving is leeg:</p> <ol style="list-style-type: none"> 4. Systeem geeft de melding dat omschrijving verplicht is om in te vullen. 5. Ga naar stap 4 (stappenplan). <p>Project is gefactureerd:</p> <ol style="list-style-type: none"> 5. Systeem markeert de periode als niet factureerbaar.
Postcondities	Periode is toegevoegd aan het project.
Uitzondering	De periode valt op een nationale feestdag. Omschrijving is leeg. (sub)project is gefactureerd.
Notities	-

2.5.2 Periode verwijderen

Versie	1.1
Samenvatting	Door het verwijderen van een periode wordt deze niet meer gebruikt voor facturatie en rapportages.
Gebruiksfrequentie	Soms
Complexiteit	Gemiddeld
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. (sub)project is actief.
Triggers	Gebruiker verwijdert de periode.
Stappenplan	<ol style="list-style-type: none"> 1. Systeem haalt gegevens op 2. Systeem verwijdert de periode.

Alternatief stappenplan	<p>Project is gefactureerd:</p> <ol style="list-style-type: none"> 1. Systeem controleert of de periode gemarkeerd is als niet factureerbaar. 2. Ga naar stap 2 (stappenplan) <p>Project is gemarkeerd als factureerbaar:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat de periode gemarkeerd is als gefactureerd aan de klant en daardoor niet verwijderd kan worden. 2. Systeem annuleert de actie.
Postcondities	Periode is verwijderd.
Uitzondering	<p>Project is gefactureerd.</p> <p>Periode is gemarkeerd als factureerbaar.</p>
Notities	-

2.5.3 Periode bewerken

Versie	1.1
Samenvatting	Door het bewerken van een periode kunnen er nieuwe gegevens worden toegevoegd en oude gegevens gewijzigd.
Gebruiksfrequentie	Vaak
Complexiteit	Gemiddeld
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebbruiker is geautoriseerd. (sub)project is actief.
Triggers	Gebbruiker bewerkt periode.
Stappenplan	<ol style="list-style-type: none"> 1. Systeem haalt gegevens op. 2. Kies taak. 3. Voer datum in. 4. Voer starttijd in. 5. Voer eindtijd in. 6. Voer omschrijving in. 7. Systeem voegt periode toe aan project.
Alternatief stappenplan	<p>Periode valt op een nationale feestdag:</p> <ol style="list-style-type: none"> 2. Systeem geeft de melding dat de periode op een nationale feestdag valt. 3. Ga naar stap 2 (stappenplan). <p>Omschrijving is leeg:</p> <ol style="list-style-type: none"> 5. Systeem geeft de melding dat omschrijving verplicht is om in te vullen. 6. Ga naar stap 4 (stappenplan). <p>Project is gefactureerd:</p> <ol style="list-style-type: none"> 1. Systeem controleert of de periode gemarkeerd is als niet factureerbaar. 2. Ga naar stap 2 (stappenplan) <p>Project is gemarkeerd als factureerbaar:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat de periode gemarkeerd is gefactureerd aan de klant en kan daardoor niet bewerkt worden. 2. Systeem annuleert de actie.

Postcondities	Nieuwe gegevens van de periode zijn opgeslagen.
Uitzondering	Omschrijving is leeg. Project is gefactureerd. Periode is gemarkeerd als factureerbaar.
Notities	-

2.5.4 Periode markeren als niet te factureren.

Versie	1.1
Samenvatting	Door het markeren van een periode als niet te factureren wordt een periode niet gefactureerd aan een klant.
Gebruiksfrequentie	Vaak
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Periode is factureerbaar. Project is niet gefactureerd.
Triggers	Gebruiker markeert periode als niet te factureren
Stappenplan	1. Systeem markeert de periode als niet te factureren.
Alternatief stappenplan	-
Postcondities	Periode is gemarkeerd als niet te factureren.
Uitzondering	-
Notities	-

2.5.5 Periode markeren als te factureren.

Versie	1.1
Samenvatting	Door een periode te markeren als te factureren wordt deze weer gebruikt voor facturatie.
Gebruiksfrequentie	Vaak
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Periode is gemarkeerd als niet factureerbaar. Project is niet gefactureerd.
Triggers	Gebruiker markeert de periode als te factureren
Stappenplan	1. Systeem markeert de periode als te factureren.
Alternatief stappenplan	-
Postcondities	Periode is markeert als te factureren.
Uitzondering	-
Notities	-

2.6 Klanten

2.6.1 Klant toevoegen

Versie	1.2
Samenvatting	Met het toevoegen van een klant worden de basisgegevens geregistreerd binnen het systeem en wordt het mogelijk om andere klantgegevens en projecten voor de klant te beheren.
Gebruiksfrequentie	Soms
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker voegt een nieuwe klant toe.
Stappenplan	<ol style="list-style-type: none"> 1. Voer klantnummer in. 2. Voer klantnaam in. 3. Voer adres in. 4. Voer postcode in. 5. Voer woonplaats in. 6. Voer telefoonnummer in. 7. Voer faxnummer in. 8. Voer e-mailadres in. 9. Voer internetadres in. 10. Voer KvK nummer in (verplicht). 11. Voer BTW-nummer in (verplicht). 12. Voer IBAN nummer in. 13. Voer BIC nummer in. 14. Systeem voegt de klant toe.
Alternatief stappenplan	<p>Geen klantnummer ingevoerd</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat er geen klantnummer is ingevoerd. 2. Ga naar stap 1 (stappenplan). <p>Klantnummer bestaat al</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat de klantnummer al bestaat. 2. Ga naar stap 1 (stappenplan). <p>Klantnaam bestaat al:</p> <ol style="list-style-type: none"> 2. Systeem geeft een waarschuwing dat een bedrijf onder dezelfde naam al voorkomt in de database. 3. Ga naar stap 3 (stappenplan). <p>Geen KvK nummer ingevoerd:</p> <ol style="list-style-type: none"> 10. Systeem geeft een waarschuwing dat een KvK nummer verplicht is. 11. Ga naar stap 11 (stappenplan). <p>Geen BTW-nummer ingevoerd:</p> <ol style="list-style-type: none"> 11. Systeem geeft een waarschuwing dat een BTW-nummer verplicht is. 12. Ga naar stap 11 (stappenplan).
Postcondities	Er is een nieuwe klant toegevoegd aan het systeem.
Uitzondering	<p>Er is geen klantnummer ingevoerd.</p> <p>Klantnummer bestaat al.</p> <p>Klantnaam komt al voor in de database.</p>

	Geen KvK nummer ingevoerd Geen BTW-nummer ingevoerd
Notities	-

2.6.2 Klant bewerken

Versie	1.2
Samenvatting	Met het bewerken van de klant is het mogelijk om oude basis gegevens aan te passen en/of nieuwe gegevens toe te voegen aan de klant.
Gebruiksfrequentie	Soms
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker bewerkt een klant
Stappenplan	<ol style="list-style-type: none"> 1. Voer klantnummer in. 2. Voer klantnaam in. 3. Voer adres in. 4. Voer postcode in. 5. Voer woonplaats in. 6. Voer telefoonnummer in. 7. Voer faxnummer in. 8. Voer e-mailadres in. 9. Voer internetadres in. 10. Voer KvK nummer in (verplicht). 11. Voer BTW nummer in (verplicht). 12. Systeem voegt de klant toe.
Alternatief stappenplan	<p>Geen klantnummer ingevoerd</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat er geen klantnummer is ingevoerd. 2. Ga naar stap 1 (stappenplan). <p>Klantnummer bestaat al</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat de klantnummer al bestaat. 2. Ga naar stap 1 (stappenplan). <p>Klantnaam bestaat al:</p> <ol style="list-style-type: none"> 2. Systeem geeft een waarschuwing dat een bedrijf onder dezelfde naam al voorkomt in de database. 3. Ga naar stap 3 (stappenplan)
Postcondities	De nieuwe klantgegevens zijn toegevoegd aan de klant.
Uitzondering	Er is geen klantnummer ingevoerd. Klantnummer bestaat al. Klantnaam komt al voor in de database.
Notities	-

2.6.3 Klant verwijderen

Versie	1.1
Samenvatting	Tijdens het verwijderen van een klant worden ook automatisch eventueel aanwezige contactpersonen, afdelingen en de accountmanager van de klant verwijderd.
Gebruiksfrequentie	Zelden
Complexiteit	Gemiddeld
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Klant mag geen projecten hebben.
Triggers	Gebruiker verwijdert een klant.
Stappenplan	1. Systeem verwijdert de klant.
Alternatief stappenplan	Klant heeft contactpersonen: 1. Systeem verwijdert contactpersonen. 2. Ga naar stap 1 (stappenplan). Klant heeft afdelingen: 1. Systeem verwijdert afdelingen. 2. Ga naar stap 1 (stappenplan). Klant heeft een accountmanager: 1. Systeem verwijdert de accountmanager. 2. Ga naar stap 1 (stappenplan).
Postcondities	Klant is verwijderd
Uitzondering	Klant heeft contactpersonen. Klant heeft afdelingen. Klant heeft een accountmanager.
Notities	-

2.6.4 Klant inactiveren

Versie	1.1
Samenvatting	Door het inactiveren van een klant komt deze niet meer standaard voor in de overzichtslijsten. Ook kunnen er geen nieuwe projecten aan een klant toegevoegd worden.
Gebruiksfrequentie	Soms
Complexiteit	Gemiddeld
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Klant heeft geen actieve projecten. Klant heeft geen ongefactureerde projecten.
Triggers	Klant wordt geïnactiveerd.
Stappenplan	1. Systeem inactieveert klant.
Alternatief stappenplan	-
Postcondities	Klant is geïnactiveerd
Uitzondering	-
Notities	-

2.6.5 Klant activeren

Versie	1.1
Samenvatting	Door het activeren van een klant komt deze weer standaard voor in de overzichtslijsten. Ook kunnen er weer nieuwe projecten aan een klant toegevoegd worden.
Gebruiksfrequentie	Soms
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Klant wordt geactiveerd.
Stappenplan	1. Systeem activeert klant.
Alternatief stappenplan	-
Postcondities	Klant is geactiveerd.
Uitzondering	-
Notities	-

2.6.6 Aangepaste klantenlijst tonen

Versie	1.1
Samenvatting	Met behulp van het maken van een aangepaste klantenlijst kunnen niet standaard klanten overzichten genereert worden.
Gebruiksfrequentie	Soms
Complexiteit	Gemiddeld
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Er zijn klanten.
Triggers	Aangepaste klantenlijst wordt opgevraagd.
Stappenplan	<ol style="list-style-type: none"> 1. Kies welke klantgegevens weergegeven moeten worden. 2. Kies het klantgegeven waarop de lijst gesorteerd moet worden. 3. Kies de sorteervolgorde. 4. Kies of alleen actieve, alleen inactieve of zowel actieve als inactieve klanten getoond moeten worden. 5. Kies of per klant detailgegevens van de contactpersoon getoond moeten worden. 6. Systeem genereert aangepaste klantenlijst. 7. Systeem geeft aangepaste klantenlijst.
Alternatief stappenplan	-
Postcondities	Aangepaste klantenlijst is gegenereerd.
Uitzondering	-
Notities	-

2.7 Contactpersonen

2.7.1 Contactpersoon aan klant toevoegen

Versie	1.1
Samenvatting	Met het toevoegen van contactpersonen aan een klant worden extra contactgegevens voor klanten ingevoerd.
Gebruiksfrequentie	Soms
Complexiteit	Laag
Prioriteit	Should have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker voegt een contactpersoon toe aan de klant.
Stappenplan	<ol style="list-style-type: none"> 1. Voer voorvoegsel in. 2. Voer voornaam in. 3. Voer voorletters in. 4. Voer tussenvoegsel in. 5. Voer achternaam in. 6. Kies afdeling. 7. Voer telefoonnummer toe. 8. Voer mobiel nummer toe. 9. Voeg fax nummer toe. 10. Voeg omschrijving toe.
Alternatief stappenplan	<p>Contactpersoon heeft geen achternaam:</p> <ol style="list-style-type: none"> 5. Systeem geeft de melding dat er geen achternaam is ingevuld voor de contactpersoon. 6. Ga naar stap 5 (stappenplan). <p>Klant heeft geen afdelingen:</p> <ol style="list-style-type: none"> 6. Ga naar stap 7 (stappenplan)
Postcondities	Contactpersoon is toegevoegd aan klant.
Uitzondering	Contactpersoon heeft geen achternaam. Klant heeft geen afdelingen.
Notities	-

2.7.2 Contactpersoon van klant verwijderen

Versie	1.1
Samenvatting	Met het verwijderen van een contactpersoon wordt de contactpersoon en de bijbehorende contactgegevens verwijderd.
Gebruiksfrequentie	Zelden
Complexiteit	Laag
Prioriteit	Should have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Contactpersoon bestaat.
Triggers	Gebruiker verwijdert contactpersoon
Stappenplan	<ol style="list-style-type: none"> 1. Systeem verwijdert contactpersoon
Alternatief stappenplan	-
Postcondities	Contactpersoon van klant is verwijderd
Uitzondering	-
Notities	-

2.7.3 Contactpersoon van klant bewerken

Versie	1.1
Samenvatting	Tijdens het bewerken van een contactpersoon kunnen oude gegevens bewerkt en/of nieuwe gegevens toegevoegd worden.
Gebruiksfrequentie	Zelden
Complexiteit	Laag
Prioriteit	Should have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker bewerkt een contactpersoon
Stappenplan	<ol style="list-style-type: none"> 1. Voer voorvoegsel in. 2. Voer voornaam in. 3. Voer voorletters in. 4. Voer tussenvoegsel in. 5. Voer achternaam in. 6. Kies afdeling 7. Voer telefoonnummer toe. 8. Voer mobiel nummer toe. 9. Voeg fax nummer toe. 10. Voeg omschrijving toe.
Alternatief stappenplan	<p>Contactpersoon heeft geen achternaam:</p> <ol style="list-style-type: none"> 5. Systeem geeft de melding dat er geen achternaam is ingevuld voor de contactpersoon. 6. Ga naar stap 5 (stappenplan). <p>Klant heeft geen afdeling:</p> <ol style="list-style-type: none"> 6. Systeem voert automatisch n.v.t. in. 7. Ga naar stap 7 (stappenplan).
Postcondities	Contactpersoon is toegevoegd aan klant.
Uitzondering	Contactpersoon heeft geen achternaam. Klant heeft geen afdeling
Notities	Bij kies afdeling is er de mogelijkheid om n.v.t. te kiezen.

2.8 Afdelingen

2.8.1 Afdeling toevoegen aan klant

Versie	1.1
Samenvatting	Aan een klant kunnen afdelingen toegevoegd worden. Aan deze afdelingen kunnen net als bij een klant projecten toegevoegd worden.
Gebruiksfrequentie	Zelden
Complexiteit	Laag
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Klant voegt aan de klant een afdeling toe.
Stappenplan	<ol style="list-style-type: none"> 1. Voer afdelingnaam in. 2. Voer adres in. 3. Voer postcode in. 4. Voer woonplaats in. 5. Voer telefoonnummer in. 6. Voer faxnummer in. 7. Voer e-mailadres in. 8. Systeem voegt de afdeling toe.
Alternatief stappenplan	Afdelingnaam bestaat al: <ol style="list-style-type: none"> 1. Systeem geeft de melding dat er al een afdeling bestaat met dezelfde naam. 2. Ga naar stap 2 (stappenplan).
Postcondities	Er is een afdeling toegevoegd aan de klant.
Uitzondering	Klant heeft al een afdeling met dezelfde naam
Notities	Het toevoegen van een afdeling zelf is laag qua complexiteit. Echter als voor een afdeling van een bedrijf ook een project gestart moet kunnen worden veroorzaakt dit een complexiteit van hoog.

2.8.2 Afdeling van klant bewerken

Versie	1.1
Samenvatting	Met het bewerken van een klant kunnen nieuwe gegevens worden toegevoegd en/of oude gegevens gewijzigd worden.
Gebruiksfrequentie	Zelden
Complexiteit	Laag
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker wijzigt de afdeling.
Stappenplan	<ol style="list-style-type: none"> 1. Voer afdelingnaam in. 2. Voer adres in. 3. Voer postcode in. 4. Voer woonplaats in. 5. Voer telefoonnummer in. 6. Voer faxnummer in. 7. Voer e-mailadres in. 8. Systeem slaat de nieuwe afdelinggegevens op.

Alternatief stappenplan	Afdelingnaam bestaat al: 1. Systeem geeft de melding dat er al een afdeling bestaat met dezelfde naam. 2. Ga naar stap 2 (stappenplan).
Postcondities	Er is een afdeling toegevoegd aan de klant.
Uitzondering	Klant heeft al een afdeling met dezelfde naam
Notities	-

2.8.3 Afdeling van klant verwijderen

Versie	1.1
Samenvatting	Met deze use case wordt een afdeling uit het systeem verwijderd. Een afdeling kan echter niet verwijderd worden als deze voor projecten gebruikt wordt of voor contactpersonen.
Gebruiksfrequentie	Zelden
Complexiteit	Gemiddeld
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker verwijdert de afdeling.
Stappenplan	1. Systeem verwijdert afdeling.
Alternatief stappenplan	Afdeling wordt gebruikt voor een of meerdere projecten. 1. Systeem geeft de melding dat de afdeling gebruikt wordt door een of meerdere projecten en geeft de naam weer van de projecten. 2. Systeem annuleert verwijdering van afdeling. Afdeling heeft een of meerdere contactpersonen. 1. Systeem geeft de melding dat de afdeling een of meerdere contactpersonen bevat en geeft de namen van de contactpersonen. 2. Systeem annuleert verwijdering van afdeling.
Postcondities	Afdeling is verwijderd.
Uitzondering	De afdeling wordt gebruikt voor een of meerdere projecten. Afdeling heeft een of meerdere contactpersonen.
Notities	Het gebruikersvriendelijk afvangen van het verwijderen van een klant wordt afgevangen in de GUI.

2.9 Accountmanagers

2.9.1 Accountmanager toewijzen aan klant

Versie	1.2
Samenvatting	Met het toewijzen van een accountmanager wordt een medewerker gekoppeld aan een klant. Deze koppeling wordt gebruikt voor het versturen van notificaties.
Gebruiksfrequentie	Zelden
Complexiteit	Laag
Prioriteit	Should have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Er zijn medewerkers. Klant bestaat.
Triggers	Gebruiker voegt een nieuwe accountmanager toe aan de klant.
Stappenplan	<ol style="list-style-type: none"> 1. Kies medewerker die als accountmanager dient voor de klant. 2. Systeem voegt de medewerker toe als accountmanager aan de klant.
Alternatief stappenplan	-
Postcondities	Er is een medewerker als accountmanager toegevoegd aan de klant.
Uitzondering	-
Notities	Er kunnen meerdere accountmanagers toegevoegd worden aan een klant.

2.9.2 Accountmanager van klant verwijderen

Versie	1.1
Samenvatting	De accountmanager koppeling tussen project en medewerker wordt verwijderd, de medewerker zelf blijft bestaan binnen het systeem.
Gebruiksfrequentie	Zelden
Complexiteit	Laag
Prioriteit	Should have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker verwijdert de accountmanager
Stappenplan	<ol style="list-style-type: none"> 1. Systeem verwijdert de accountmanager.
Alternatief stappenplan	-
Postcondities	Accountmanager is verwijderd.
Uitzondering	-
Notities	Idem aan de notitie van accountmanager toevoegen.

2.10 Projecten

2.10.1 Project aanmaken

Versie	1.1
Samenvatting	Met het aanmaken van een project kunnen uren en kosten bijgehouden worden in het systeem voor de opgegeven klant.
Gebruiksfrequentie	Soms
Complexiteit	Hoog
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker maakt een nieuw project aan
Stappenplan	<ol style="list-style-type: none"> 1. Voer Job nr. in. 2. Voer naam van project in. 3. Voer omschrijving in. 4. Kies klant. 5. Kies afdeling. 6. Kies projectmanager. 7. Voer startdatum in. 8. Voer deadline in. 9. Voer algemene geschatte externe kosten in. 10. Voer taken in (zie use cases taken). 11. Voeg subproject(en) toe (ga naar subproject toekennen aan project, stap 1 stappenplan). 12. Systeem berekend offertebedrag 13. Systeem maakt nieuw project aan.
Alternatief stappenplan	<p>Job nr. is leeg:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat er geen Job nr. is ingevuld. 2. Ga naar stap 1 (stappenplan). <p>Job nr. bestaat al:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat de Job nr. al bestaat. 2. Ga naar stap 1 (stappenplan). <p>Klant heeft geen afdeling:</p> <ol style="list-style-type: none"> 5. Ga naar stap 6 (stappenplan).
Postcondities	Project is aangemaakt.
Uitzondering	<p>Job nr. is leeg</p> <p>Job nr. bestaat al</p> <p>Geen startdatum ingevoerd maar wel een deadline.</p> <p>Klant heeft afdelingen.</p>
Notities	<p>Autogeneratie van job nr. is op het moment niet mogelijk omdat Software Solutions en Web Solutions andere standaarden gebruiken voor nummering. Interne projecten en algemene taken worden ook op een andere manier genummerd.</p> <p>Volgende autogeneratie van job nr. Is voorgesteld: yy/klantnr_projectnr</p>

2.10.2 Project bewerken

Versie	1.1
Samenvatting	Met het bewerken van een project kunnen bestaande gegevens gewijzigd worden en nieuwe gegevens toegevoegd worden. De nieuwe gegevens worden gelijk gebruikt voor berekeningen en notificaties.
Gebruiksfrequentie	Soms
Complexiteit	Hoog
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Project is niet gefactureerd.
Triggers	Gebruiker bewerkt een project
Stappenplan	<ol style="list-style-type: none"> 1. Voer Job nr. in. 2. Voer naam van project in. 3. Voer omschrijving in. 4. Kies klant. 5. Kies afdeling. 6. Kies projectmanager. 7. Voer startdatum in 8. Voer deadline in. 9. Voer algemene geschatte uren in. 10. Voer taken in (zie use cases taken). 11. Voer algemene geschatte externe kosten in. 12. Voeg subproject(en) toe (ga naar subproject toekennen aan project, stap 1 stappenplan). 13. Systeem berekend offertebedrag 14. Systeem slaat de nieuwe projectgegevens op.
Alternatief stappenplan	<p>Job nr. is leeg:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat er geen Job nr. is ingevuld. 2. Ga naar stap 1 (stappenplan). <p>Job nr. bestaat al:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat de Job nr. al bestaat. 2. Ga naar stap 1 (stappenplan). <p>Klant heeft geen afdeling:</p> <ol style="list-style-type: none"> 5. Ga naar stap 6 (stappenplan).
Postcondities	Nieuwe projectgegevens zijn opgeslagen.
Uitzondering	Job nr. is leeg Job nr. bestaat al Geen startdatum ingevoerd maar wel een deadline. Klant heeft afdelingen.
Notities	

2.10.3 Project verwijderen

Versie	1.1
Samenvatting	Door het verwijderen van een project komt deze niet meer voor in PTM. Projecten kunnen alleen verwijderd worden als deze geen gegevens bevatten.
Gebruiksfrequentie	Zelden
Complexiteit	Gemiddeld
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker verwijdert project.
Stappenplan	1. Systeem verwijdert project.
Alternatief stappenplan	Project is gefactureerd. 1. Systeem geeft de melding dat het project niet verwijderd kan worden omdat deze gefactureerd is. Project heeft periodes. 1. Systeem geeft de melding dat de volgende medewerkers uren ingevoerd hebben voor het project.
Postcondities	Project is verwijderd
Uitzondering	Project mag niet gefactureerd zijn. Project mag niet gebruikt worden voor werknemers periodes.
Notities	-

2.10.4 Project inactiveren

Versie	1.1
Samenvatting	Door het inactiveren van een project komt het project niet meer voor in de lijst met actieve projecten. Gebruikers kunnen ook geen nieuwe externe kosten en periodes toevoegen aan het project.
Gebruiksfrequentie	Soms
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Project is actief
Triggers	Gebruiker inactieveert project.
Stappenplan	1. Systeem inactieveert project.
Alternatief stappenplan	-
Postcondities	Systeem is geïnactiveerd.
Uitzondering	-
Notities	-

2.10.5 Project activeren

Versie	1.1
Samenvatting	Door het activeren van een project komt deze weer beschikbaar binnen het systeem en kunnen er weer periodes en externe kosten toegevoegd worden aan het project.
Gebruiksfrequentie	Zelden
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Project is inactief
Triggers	Gebruiker activeert project.
Stappenplan	1. Systeem activeert project.
Alternatief stappenplan	-
Postcondities	Systeem is geïnactiveerd.
Uitzondering	-
Notities	-

2.10.6 Project markeren als gefactureerd

Versie	1.1
Samenvatting	Door het markeren van een project als gefactureerd wordt in PTM aangegeven dat het huidige project volledig gefactureerd is aan de klant.
Gebruiksfrequentie	Soms
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Project is niet gefactureerd.
Triggers	Gebruiker factureert project.
Stappenplan	1. Systeem markeert het project als gefactureerd.
Alternatief stappenplan	-
Postcondities	Project is gemarkeerd als gefactureerd.
Uitzondering	-
Notities	Project wordt niet gemarkeerd als inactief. Als een project gefactureerd is worden periodes die door werknemers worden toegevoegd als niet factureerbaar gemarkeerd. Systeem geeft een melding als de deadline van het project nog niet verstreken is en het project als gefactureerd wordt aangemerkt.

2.10.7 Project markeren als niet gefactureerd

Versie	1.1
Samenvatting	Door het markeren van een project als niet gefactureerd wordt in PTM aangegeven dat het huidige project niet of niet volledig gefactureerd is aan de klant.
Gebruiksfrequentie	Soms
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Project is gefactureerd.
Triggers	Gebruiker factureert project.
Stappenplan	1. Systeem markeert het project als niet gefactureerd.
Alternatief stappenplan	-
Postcondities	Project is gemarkeerd als niet gefactureerd.
Uitzondering	-
Notities	Project wordt gemarkeerd als actief. Als een project niet meer gefactureerd is worden periodes die door werknemers worden toegevoegd als factureerbaar gemarkeerd.

2.10.8 Document toevoegen aan project

Versie	1.1
Samenvatting	Aan een project kunnen documenten toegevoegd worden.
Gebruiksfrequentie	Zelden
Complexiteit	Hoog
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker voegt een document toe aan het project Systeem voegt een document toe aan het project (facturen)
Stappenplan	1. Systeem slaat het document op.
Alternatief stappenplan	-
Postcondities	Document is opgeslagen in het systeem.
Uitzondering	-
Notities	Standaard IO exceptions zijn niet uitgewerkt. Timestamp informatie over creatie van het document moet beschikbaar zijn. Een document moet op meerdere plaatsen te gebruiken zijn en men moet de mogelijkheid hebben om documenten in te delen in een soort van verkennen view.

2.10.9 Document verwijderen uit project.

Versie	1.1
Samenvatting	In deze use case wordt beschreven hoe documenten verwijderd worden die toegevoegd zijn aan een project.
Gebruiksfrequentie	Zelden
Complexiteit	Hoog
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker verwijdert een document.
Stappenplan	1. Systeem verwijdert het document
Alternatief stappenplan	-
Postcondities	Document is verwijderd uit het systeem.
Uitzondering	-
Notities	Standaard IO exceptions zijn niet uitgewerkt.

2.11 Subproject

2.11.1 Subproject aanmaken

Versie	1.1
Samenvatting	Met het toekennen van een subproject aan een project wordt het subproject gebruikt binnen het project.
Gebruiksfrequentie	Soms
Complexiteit	Gemiddeld
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Er is een project opgegeven. Project is niet gefactureerd.
Triggers	Gebruiker voegt een subproject toe aan het project.
Stappenplan	<ol style="list-style-type: none"> 1. Voer Job nr. in. 2. Voer naam van het subproject in. 3. Voer omschrijving in. 4. Voer deadline in. 5. Voer geschatte uren in. 6. Voer taken in (zie use cases taken). 7. Voer geschatte externe kosten in 8. Systeem kent subproject toe aan project.
Alternatief stappenplan	Deadline ligt voorbij deadline project: <ol style="list-style-type: none"> 4. Systeem geeft de melding dat de deadline van het subproject voorbij de deadline van het project ligt. 5. Ga naar stap 4 (stappenplan).
Postcondities	Subproject is toegevoegd aan project
Uitzondering	Deadline ligt voorbij deadline project.
Notities	-

2.11.2 Subproject bewerken

Versie	1.1
Samenvatting	Met het bewerken van een subproject kunnen bestaande gegevens gewijzigd worden en nieuwe gegevens toegevoegd worden.
Gebruiksfrequentie	Soms
Complexiteit	Won't have
Prioriteit	Could have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Project is niet gefactureerd.
Triggers	Gebruiker bewerkt subproject.
Stappenplan	<ol style="list-style-type: none"> 1. Voer Job nr. in. 2. Voer naam van het subproject in. 3. Voer omschrijving in. 4. Voer deadline in. 5. Voer geschatte uren in. 6. Voer taken in (zie use cases taken). 7. Voer geschatte externe kosten in. 8. Systeem kent subproject toe aan project.

Alternatief stappenplan	Deadline ligt voorbij deadline project: 4. Systeem geeft de melding dat de deadline van het subproject voorbij de deadline van het project ligt. 5. Ga naar stap 4 (stappenplan).
Postcondities	De nieuwe gegevens zijn opgeslagen.
Uitzondering	Deadline ligt voorbij deadline project.
Notities	-

2.11.3 Subproject verwijderen

Versie	1.1
Samenvatting	Door het verwijderen van een subproject komt deze niet meer voor in PTM en wordt niet meer gebruikt voor berekeningen. Subprojecten kunnen alleen verwijderd worden als deze geen gegevens bevatten.
Gebruiksfrequentie	Zelden
Complexiteit	Gemiddeld
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Project is niet gefactureerd.
Triggers	Gebruiker verwijdert subproject.
Stappenplan	1. Systeem verwijdert subproject.
Alternatief stappenplan	-
Postcondities	Subproject is verwijderd.
Uitzondering	-
Notities	-

2.12 Taken

Een taak wordt gebruikt voor het oormerken en schatten van uren die gemaakt worden in een project. Deze gegevens worden gebruikt voor het inzichtelijk maken hoeveel tijd voor bepaalde taken beschikbaar wordt gesteld en hoeveel daadwerkelijk wordt besteed. Hierdoor kunnen projecten op den duur beter ingeschat en gepland worden.

Taken worden niet gebruikt voor deadline waarschuwingen en het overschrijden van de toegekende uren veroorzaakt geen notificatie. Deze keuze is gemaakt omdat het toevoegen van dergelijke functionaliteit zeer grote gevolgen heeft voor de complexiteit van de software.

2.12.1 Taak aanmaken

Versie	1.1
Samenvatting	Door het aanmaken van een taak kan deze toegekend worden aan een project. Deze taak kan dan gebruikt worden voor het oormerken van uren voor het genereren van rapporten.
Gebruiksfrequentie	Soms
Complexiteit	Laag
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker maakt een taak aan
Stappenplan	<ol style="list-style-type: none"> 1. Voer naam in van taak 2. Voer omschrijving van taak in
Alternatief stappenplan	Naam van taak is niet uniek: <ol style="list-style-type: none"> 1. Systeem geeft de melding dat er al een taak bestaat met dezelfde naam. 2. Ga naar stap 1 (stappenplan).
Postcondities	Er is een nieuwe taak beschikbaar binnen het systeem
Uitzondering	Naam van taak is niet uniek
Notities	-

2.12.2 Taak bewerken

Versie	1.1
Samenvatting	Door het bewerken van een taak veranderen de naam en omschrijving van de taak binnen de hele software. Ook voor taken die toegekend zijn aan projecten.
Gebruiksfrequentie	Zelden
Complexiteit	Laag
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker wijzigt een taak.
Stappenplan	<ol style="list-style-type: none"> 1. Voer naam in van taak 2. Voer omschrijving van taak in
Alternatief stappenplan	Naam van taak is niet uniek: <ol style="list-style-type: none"> 1. Systeem geeft de melding dat er al een taak bestaat met dezelfde naam. 2. Ga naar stap 1 (stappenplan).
Postcondities	Er is een nieuwe taak beschikbaar binnen het systeem
Uitzondering	Naam van taak is niet uniek
Notities	-

2.12.3 Taak toekennen aan (sub)project

Versie	1.1
Samenvatting	Met het toekennen van een taak aan een (sub)project kunnen de beschikbare uren voor de taak voor dat (sub)project ingesteld, en kunnen medewerkers de taak gebruiken voor het invoeren van periodes.
Gebruiksfrequentie	Soms
Complexiteit	Gemiddeld
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Project is actief.
Triggers	Gebruiker voegt taak aan (sub)project toe.
Stappenplan	1. Kies project. 2. Voer beschikbare uren voor taak in.
Alternatief stappenplan	-
Postcondities	Taak is toegekend aan een project
Uitzondering	-
Notities	Een taak kan aan verschillende projecten en subprojecten toegevoegd worden. Voor elk project kan een apart aantal beschikbare uren ingevoerd worden.

2.12.4 Taak verwijderen

Versie	1.1
Samenvatting	Met het verwijderen van een taak komt deze niet meer voor in het systeem.
Gebruiksfrequentie	Zelden
Complexiteit	Gemiddeld
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Taak wordt niet gebruikt voor een project
Triggers	Gebruiker verwijdert de taak.
Stappenplan	1. Systeem verwijdert de taak
Alternatief stappenplan	-
Postcondities	Taak is verwijdert.
Uitzondering	-
Notities	-

2.13 Factureren

2.13.1 Projectkosten dashboard

Versie	1.2
Samenvatting	In de projectkosten dashboard kan voor een project aangegeven worden welke uren en kosten factureerbaar zijn. En kunnen kortingen en algemene niet te factureren kosten en uren ingevoerd worden.
Gebruiksfrequentie	Soms
Complexiteit	Zeer hoog
Prioriteit	Could have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Project is niet gefactureerd.
Triggers	Gebruiker vraagt de projectkosten dashboard op
Stappenplan	<ol style="list-style-type: none"> 1. Systeem haalt benodigde gegevens op 2. Berekent totalen 3. Systeem geeft overzicht van totalen. 4. Bewerk de externe kosten (zie de use cases voor externe kosten voor details). 5. Voer niet te factureren kosten in. 6. Systeem herberekent de totalen. 7. Bewerk de gewerkte uren (zie de use cases voor periodes voor details). 8. Voer niet te factureren uren in. 9. Systeem berekent de totalen. 10. Voer kortingspercentage in. 11. Systeem berekent de totalen. 12. Voeg facturen toe (zie use cases factuur toevoegen, bewerken en verwijderen). 13. Systeem berekend de totalen. 14. Systeem slaat de nieuwe gegevens op.
Alternatief stappenplan	-
Postcondities	De projectkosten dashboard is bewerkt.
Uitzondering	-
Notities	<p>Complexiteit is hoog voor de use case omdat tijdens bewerken de volledige functionaliteiten van externe kosten en de volledige functionaliteit van de periode use cases beschikbaar moeten zijn. De gebruiker moet wel geautoriseerd zijn voor de in de use cases genoemde functionaliteiten om er gebruik van te kunnen maken.</p> <p>Het gebruik van de use cases voor het bewerken van externe kosten en periodes is niet volledig uitgewerkt in deze use case om de complexiteit van de use case te beperken en de leesbaarheid te verhogen.</p> <p>Het invoeren van niet te factureren kosten en uren is mogelijk voor een algemene correctie op gewerkte uren. Dit voor het geval het niet mogelijk is om aan te geven welke gewerkte uren niet factureerbaar zijn.</p> <p>Kortingspercentage is voor het mogelijk maken van acties.</p> <p>BTW berekenen moet instelbaar zijn. Dit is nodig voor buitenlandse bedrijven.</p>

2.13.2 Factuur toevoegen

Versie	1.1
Samenvatting	Met het toevoegen van een factuur verandert het uitstaande bedrag op het projectkosten dashboard.
Gebruiksfrequentie	Soms
Complexiteit	Laag
Prioriteit	Could have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker voegt een factuur toe.
Stappenplan	<ol style="list-style-type: none"> 1. Voer factuurdatum in. 2. Voer factuurnummer in. 3. Voer bedrag in (ex btw).
Alternatief stappenplan	<p>Factuurdatum is niet ingevoerd:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat het invoeren van een factuurdatum verplicht is 2. Ga naar stap 1 (stappenplan). <p>Factuurnummer is niet ingevoerd:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat het invoeren van een factuurnummer verplicht is. 2. Ga naar stap 2 (stappenplan). <p>Bedrag is niet ingevoerd:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat het invoeren van een factuurbedrag verplicht is. 2. Ga naar stap 3 (stappenplan).
Postcondities	Er is een factuur toegevoegd aan de projectkosten dashboard.
Uitzondering	Factuurdatum is niet ingevoerd. Factuurnummer is niet ingevoerd. Bedrag is niet ingevoerd.
Notities	Een factuur mag negatieve bedragen bevatten zodat het invoeren van creditfacturen mogelijk is.

2.13.3 Factuur bewerken

Versie	1.1
Samenvatting	Met het bewerken van een factuur is het mogelijk oude gegevens te wijzigen.
Gebruiksfrequentie	Soms
Complexiteit	Laag
Prioriteit	Could have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker bewerkt een factuur.
Stappenplan	<ol style="list-style-type: none"> 1. Voer factuurdatum in. 2. Voer factuurnummer in. 3. Voer bedrag in (ex btw).
Alternatief stappenplan	<p>Factuurdatum is niet ingevoerd:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat het invoeren van een factuurdatum verplicht is 2. Ga naar stap 1 (stappenplan). <p>Factuurnummer is niet ingevoerd:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat het invoeren van een factuurnummer verplicht is.

	<p>2. Ga naar stap 2 (stappenplan).</p> <p>Bedrag is niet ingevoerd:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat het invoeren van een factuurbedrag verplicht is. 2. Ga naar stap 3 (stappenplan).
Postcondities	De nieuwe gegevens van de factuur zijn opgeslagen.
Uitzondering	<p>Factuurdatum is niet ingevoerd.</p> <p>Factuurnummer is niet ingevoerd.</p> <p>Bedrag is niet ingevoerd.</p>
Notities	Een factuur mag negatieve bedragen bevatten zodat het invoeren van creditfacturen mogelijk is.

2.13.4 Factuur verwijderen

Versie	1.1
Samenvatting	Door het verwijderen van een factuur komt deze niet meer in het systeem voor en wordt niet meer gebruikt voor het berekenen van het uitstaande bedrag in de projectkosten dashboard.
Gebruiksfrequentie	Soms
Complexiteit	Laag
Prioriteit	Could have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker verwijdert de factuur
Stappenplan	Systeem verwijdert de factuur.
Alternatief stappenplan	-
Postcondities	Factuur is verwijderd.
Uitzondering	-
Notities	-

2.13.5 Factuur markeren als betaald

Versie	1.2
Samenvatting	Door een factuur te markeren als betaald verandert het behaalde resultaat van het project.
Gebruiksfrequentie	Soms
Complexiteit	Laag
Prioriteit	Could have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker markeert de factuur als betaald.
Stappenplan	1. Systeem markeert de factuur als betaald
Alternatief stappenplan	-
Postcondities	Factuur is gemarkeerd als betaald.
Uitzondering	-
Notities	-

2.13.6 Factuur markeren als niet betaald

Versie	1.2
Samenvatting	Door een factuur te markeren als niet betaald verandert het behaalde resultaat van het project.
Gebruiksfrequentie	Soms
Complexiteit	Laag
Prioriteit	Could have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker markeert de factuur als niet betaald.
Stappenplan	1. Systeem markeert de factuur als niet betaald
Alternatief stappenplan	-
Postcondities	Factuur is gemarkeerd als niet betaald.
Uitzondering	-
Notities	-

2.14 Externe kosten

De use case externe kosten vervangen de huidige functionaliteit van leverancier en materiaal.

2.14.1 Externe kosten toevoegen aan (sub)project

Versie	1.1
Samenvatting	Door het toevoegen van een externe kostenpost aan een project worden deze kosten tijdens facturatie doorberekend aan de klant.
Gebruiksfrequentie	Vaak
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Project is actief.
Triggers	Gebruiker voegt externe kosten toe.
Stappenplan	<ol style="list-style-type: none"> 1. Voer omschrijving in. 2. Voer prijs in. 3. Systeem voegt externe kosten toe aan (sub)project.
Alternatief stappenplan	Project is gefactureerd: <ol style="list-style-type: none"> 3. Systeem markeert de externe kosten als niet factureerbaar.
Postcondities	Er is een externe kostenpost toegevoegd aan het (sub)project.
Uitzondering	Project is gefactureerd.
Notities	Functionaliteit voor het toevoegen van meerdere documenten aan een externe kostenpost is gewenst.

2.14.2 Externe kosten bewerken

Versie	1.1
Samenvatting	Tijdens het bewerken van een externe kostenpost kunnen nieuwe gegevens toegevoegd worden en oude gegevens gewijzigd.
Gebruiksfrequentie	Vaak
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Project is actief
Triggers	Gebruiker bewerkt de externe kostenpost
Stappenplan	<ol style="list-style-type: none"> 1. Systeem haalt gegevens op. 2. Voer omschrijving in. 3. Voer prijs in. 4. Systeem voegt externe kosten toe aan (sub)project.
Alternatief stappenplan	Project is gefactureerd: <ol style="list-style-type: none"> 3. Systeem controleert of de externe kostenpost gemarkeerd is als niet factureerbaar. 4. Ga naar stap 2 (stappenplan) Project is gemarkeerd als factureerbaar: <ol style="list-style-type: none"> 3. Systeem geeft de melding dat de externe kostenpost gemarkeerd is als gefactureerd aan de klant en daardoor niet bewerkt kan worden. 3. Systeem annuleert de actie.
Postcondities	De nieuwe gegevens zijn opgeslagen.
Uitzondering	Project is gefactureerd. Externe kostenpost is gemarkeerd als factureerbaar.
Notities	

2.14.3 Externe kosten verwijderen

Versie	1.1
Samenvatting	Door het verwijderen van een externe kostenpost word deze niet meer doorberekend aan de klant.
Gebruiksfrequentie	Zelden
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Project is niet gefactureerd.
Triggers	Gebruiker verwijdert externe kosten.
Stappenplan	1. Systeem verwijdert externe kosten.
Alternatief stappenplan	-
Postcondities	Externe post is verwijderd.
Uitzondering	-
Notities	-

2.14.4 Externe kosten markeren als niet factureerbaar

Versie	1.1
Samenvatting	Door een externe kostenpost als niet factureerbaar te markeren wordt deze niet gebruikt op de factuur van de klant. Deze gegevens worden wel gebruikt voor het genereren van rapportages.
Gebruiksfrequentie	Zelden
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Project is niet gefactureerd
Triggers	Gebruiker markeert de externe kostenpost als niet factureerbaar.
Stappenplan	1. Systeem markeert de externe kostenpost als niet factureerbaar.
Alternatief stappenplan	-
Postcondities	Externe kostenpost is gemarkeerd als niet factureerbaar.
Uitzondering	-
Notities	-

2.14.5 Externe kosten markeren als factureerbaar

Versie	1.1
Samenvatting	Door het markeren van een externe kostenpost als factureerbaar wordt deze tijdens facturatie doorberekend aan de klant.
Gebruiksfrequentie	Zelden
Complexiteit	Laag
Prioriteit	Must have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Project is niet gefactureerd.
Triggers	Gebruiker markeert de externe kostenpost als factureerbaar.
Stappenplan	1. Systeem markeert de externe kostenpost als factureerbaar.
Alternatief stappenplan	-
Postcondities	Externe kostenpost is gemarkeerd als factureerbaar.
Uitzondering	-
Notities	-

2.15 Verlofdagen

Verlofdagen hebben een complexiteit van hoog omdat een verlofdag door verschillende gebruikers tegelijk bewerkt kan worden.

2.15.1 Verlofdag indienen

Versie	1.1
Samenvatting	Door het indienen van een verlofdag wordt de dag voor de werknemer gemarkeerd als voorlopige verlofdag en een notificatie wordt verstuurd voor toestemming.
Gebruiksfrequentie	Soms
Complexiteit	Hoog
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker vraagt een verlofdag aan.
Stappenplan	<ol style="list-style-type: none"> 1. Voer datum in. 2. Voer starttijd in. 3. Voer eindtijd in. 4. Voer omschrijving in. 5. Systeem slaat verlofdag op. 6. Systeem verstuurt notificatie.
Alternatief stappenplan	<p>Periode valt op een nationale feestdag:</p> <ol style="list-style-type: none"> 3. Systeem geeft de melding dat de periode op een nationale feestdag valt. 4. Ga naar stap 1 (stappenplan). <p>Omschrijving is leeg:</p> <ol style="list-style-type: none"> 4. Systeem geeft de melding dat omschrijving verplicht is om in te vullen. 5. Ga naar stap 4 (stappenplan).
Postcondities	Verlofdag is ingediend.
Uitzondering	De periode valt op een nationale feestdag Omschrijving is leeg
Notities	Als de verlofdag niet binnen 48 uur goedgekeurd of afgekeurd is wordt een nieuwe notificatie verstuurd.

2.15.2 Verlofdag verwijderen

Versie	1.1
Samenvatting	Door het verwijderen van de verlofdag komt de dag weer beschikbaar voor de werknemer als normale werkdag en wordt de verlofdag weer bij zijn beschikbare dagen gezet.
Gebruiksfrequentie	Zelden
Complexiteit	Hoog
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker verwijdert de verlofdag
Stappenplan	<ol style="list-style-type: none"> 1. Systeem controleert status van verlofdag. 2. Systeem verwijdert de verlofdag.

Alternatief stappenplan	Verlofdag is nog niet goedgekeurd: 1. Systeem trekt de notificatie in. 2. Ga naar stap 2 (stappenplan).
Postcondities	Verlofdag is verwijderd
Uitzondering	Verlofdag is nog niet goedgekeurd.
Notities	-

2.15.3 Verlofdag bewerken

Versie	1.1
Samenvatting	Door het bewerken van een verlofdag kan de verlofdag verplaatst worden naar een andere datum, nieuwe gegevens toegevoegd worden of oude gegevens aangepast worden. Het systeem stuurt een notificatie als de uren en de datum veranderen van de verlofdag.
Gebruiksfrequentie	Soms
Complexiteit	Hoog
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker bewerkt de verlofdag.
Stappenplan	1. Voer datum in. 2. Voer starttijd in. 3. Voer eindtijd in. 4. Voer omschrijving in. 5. Systeem slaat verlofdag op. 6. Systeem stuurt notificatie.
Alternatief stappenplan	Periode valt op een nationale feestdag: 1. Systeem geeft de melding dat de periode op een nationale feestdag valt. 2. Ga naar stap 1 (stappenplan). Omschrijving is leeg: 4. Systeem geeft de melding dat omschrijving verplicht is om in te vullen. 5. Ga naar stap 4 (stappenplan). Verlofdag is nog niet goedgekeurd: 6. Systeem haalt verzonden notificatie op. 7. Systeem werkt notificatie bij.
Postcondities	De nieuwe gegevens zijn opgeslagen.
Uitzondering	De periode valt op een nationale feestdag Omschrijving is leeg Verlofdag is nog niet goedgekeurd.
Notities	

2.15.4 Verlofdag goedkeuren.

Versie	1.1
Samenvatting	Door het goedkeuren van een verlofdag wordt deze gemarkeerd als definitief en de medewerker die de verlofdag heeft aangevraagd krijgt een notificatie dat de verlofdag is goedgekeurd.
Gebruiksfrequentie	Soms
Complexiteit	Hoog
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker ontvangt een verlofdag aanvraag
Stappenplan	<ol style="list-style-type: none"> 1. Systeem vraagt de verlofdag aanvraag op. 2. Gebruiker keurt de verlofdag. 3. Systeem zet de status voorlopig uit van de verlofdag. 4. Systeem stuurt een notificatie naar de medewerker met de melding dat de verlofdag is goedgekeurd.
Alternatief stappenplan	<p>Verlofdag wordt afgekeurd:</p> <ol style="list-style-type: none"> 2. Voer reden voor afkeuring in. 3. Systeem verwijdert de verlofdag. 4. Systeem stuurt een notificatie naar de medewerker met de melding dat de verlofdag is afgekeurd met vermelding van de ingevoerd reden.
Postcondities	Verlofdag is goedgekeurd
Uitzondering	Verlofdag wordt afgekeurd.
Notities	-

2.15.5 Opvragen overzichtverlofdagen

Versie	1.1
Samenvatting	Een werknemer kan op het overzicht van zijn verlofdagen zien hoeveel verlofdagen hij beschikbaar heeft, wanneer hij verlofdagen heeft gepland en welke status deze verlofdagen hebben.
Gebruiksfrequentie	Soms
Complexiteit	Gemiddeld
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Medewerker bestaat.
Triggers	Gebruiker vraagt verlofdagen overzicht op. Er zijn verlofdagen ingevoerd. Werknemer bestaat.
Stappenplan	<ol style="list-style-type: none"> 1. Systeem genereert een overzicht met de opgenomen, geplande en de nog beschikbare verlofdagen 2. Systeem geeft het overzicht.
Alternatief stappenplan	-
Postcondities	Systeem geeft een overzicht van opgenomen, geplande en de nog beschikbare verlofdagen.
Uitzondering	-
Notities	-

2.15.6 Opvragen overzichtverlofdagen voor alle medewerkers

Versie	1.1
Samenvatting	In het overzicht verlofdagen kan de gebruiker zien wanneer werknemers verlofdagen gepland hebben en de status hiervan.
Gebruiksfrequentie	Soms
Complexiteit	Hoog
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker vraagt overzicht voor alle medewerkers op.
Stappenplan	<ol style="list-style-type: none"> 1. Systeem genereert een overzicht met de opgenomen, geplande en de nog beschikbare verlofdagen 2. Systeem geeft het overzicht.
Alternatief stappenplan	-
Postcondities	Systeem geeft een overzicht van alle werknemers wanneer ze verlofdagen hebben gepland en de status hiervan.
Uitzondering	-
Notities	-

2.15.7 Verlofdagen overzetten naar huidig kalenderjaar.

Versie	1.1
Samenvatting	Door het overzetten van verlofdagen van een voorgaand kalenderjaar naar het huidige kalenderjaar komen deze verlofdagen beschikbaar voor de medewerker in het huidig jaar.
Gebruiksfrequentie	Zelden
Complexiteit	Gemiddeld
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd. Medewerker heeft verlofdagen over van voorgaand kalenderjaar.
Triggers	Gebruiker zet verlofdagen over naar nieuw kalenderjaar
Stappenplan	<ol style="list-style-type: none"> 1. Kies aantal over te zetten verlofdagen. 2. Systeem zet verlofdagen over.
Alternatief stappenplan	-
Postcondities	Verlofdagen van vorig kalenderjaar zijn toegevoegd aan de verlofdagen van het huidig kalenderjaar.
Uitzondering	-
Notities	-

2.15.8 Verlofdagen aanmaken voor kalenderjaar

Versie	1.1
Samenvatting	Door het aanmaken van verlofdagen voor een kalenderjaar kan een in dat kalenderjaar verlofdagen plannen.
Gebruiksfrequentie	Zelden
Complexiteit	Laag
Prioriteit	Won't have
Actors	Beheerder, medewerker.
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker maakt verlofdagen aan voor huidig kalenderjaar.
Stappenplan	<ol style="list-style-type: none">1. Systeem berekent verlofdagen voor de medewerker.2. Voer extra verlofdagen in.3. Systeem voegt verlofdagen toe aan de medewerker.
Alternatief stappenplan	-
Postcondities	Werknemer heeft verlofdagen voor huidig kalenderjaar.
Uitzondering	-
Notities	-

2.16 Ziekmelding

Een ziekmelding is te zien tussen de periodes, net zoals verlofdagen.

2.16.1 Ziekmelding invoeren

Versie	1.1
Samenvatting	Een ziekmelding wordt gebruikt binnen het systeem voor het bijhouden van ziekteverzuim.
Gebruiksfrequentie	Soms
Complexiteit	Gemiddeld
Prioriteit	Won't have
Actors	Beheerder medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker voegt een ziekmelding toe.
Stappenplan	<ol style="list-style-type: none"> 1. Voer datum in. 2. Voer uren in. 3. Voer omschrijving in. 4. Systeem voegt de ziekmelding toe.
Alternatief stappenplan	<p>Ziekmelding valt op een nationale feestdag:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat de ziekmelding op een nationale feestdag valt. 2. Ga naar stap 1 (stappenplan). <p>Ziekmelding valt op een verlofdag:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat de ziekmelding op een verlofdag valt. 2. Ga naar stap 1 (stappenplan). <p>Ziekmelding valt in het weekend:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat de ziekmelding in het weekend valt. 2. Ga naar stap 1 (stappenplan). <p>Aantal ziek uren zijn nul:</p> <ol style="list-style-type: none"> 2. Systeem geeft de melding dat het invullen nul ziek uren niet mogelijk is. 3. Ga naar stap 1 (stappenplan). <p>Omschrijving is leeg:</p> <ol style="list-style-type: none"> 3. Systeem geeft de melding dat het invullen van een omschrijving verplicht is. 4. Ga naar stap 2 (stappenplan).
Postcondities	Er is een ziekmelding toegevoegd.
Uitzondering	<p>Ziekmelding valt op een nationale feestdag.</p> <p>Ziekmelding valt op een verlofdag.</p> <p>Ziekmelding valt in het weekend.</p> <p>Aantal ziekuren zijn nul.</p> <p>Omschrijving is leeg.</p>
Notities	-

2.16.2 Ziekmelding bewerken

Versie	1.1
Samenvatting	Met het bewerken van een ziekmelding kan de ziekmelding verplaatst worden naar een andere dag en gegevens gewijzigd worden
Gebruiksfrequentie	Soms
Complexiteit	Gemiddeld
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker bewerkt de ziekmelding.
Stappenplan	Gebruiker voegt een ziekmelding toe.
Alternatief stappenplan	<ol style="list-style-type: none"> 1. Voer datum in. 2. Voer uren in. 3. Voer omschrijving in. 4. Systeem voegt de ziekmelding toe.
Postcondities	<p>Ziekmelding valt op een nationale feestdag:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat de ziekmelding op een nationale feestdag valt. 2. Ga naar stap 1 (stappenplan). <p>Ziekmelding valt op een verlofdag:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat de ziekmelding op een verlofdag valt. 2. Ga naar stap 1 (stappenplan). <p>Ziekmelding valt in het weekend:</p> <ol style="list-style-type: none"> 1. Systeem geeft de melding dat de ziekmelding n het weekend valt. 2. Ga naar stap 1 (stappenplan). <p>Aantal ziek uren zijn nul:</p> <ol style="list-style-type: none"> 2. Systeem geeft de melding dat het invullen nul ziek uren niet mogelijk is. 3. Ga naar stap 1 (stappenplan). <p>Omschrijving is leeg:</p> <ol style="list-style-type: none"> 3. Systeem geeft de melding dat het invullen van een omschrijving verplicht is. 4. Ga naar stap 2 (stappenplan).
Uitzondering	Er is een ziekmelding toegevoegd.
Notities	-

2.16.3 Ziekmelding verwijderen

Versie	1.1
Samenvatting	Door het verwijderen van een ziekmelding komt deze niet meer voor in het systeem en wordt niet meer gebruikt voor het bijhouden van ziekteverzuim.
Gebruiksfrequentie	Zelden
Complexiteit	Laag
Prioriteit	Won't have
Actors	Beheerder, medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Gebruiker verwijdert de ziekmelding
Stappenplan	1. Systeem verwijdert de ziekmelding
Alternatief stappenplan	-
Postcondities	Ziekmelding is verwijderd.
Uitzondering	-
Notities	-

2.17 Notificatie

2.17.1 Deadline dreigt overschreden te worden

Versie	1.1
Samenvatting	Aan de hand van de gewerkte uren en de verstreken beschikbare werkdagen voor het project. Beslist het systeem of er een notificatie verstuurt moet worden naar de projectmanager en accountmanager.
Gebruiksfrequentie	Vaak
Complexiteit	Hoog
Prioriteit	Won't have
Actors	Medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Medewerker voegt periode toe aan een (sub)project.
Stappenplan	<ol style="list-style-type: none"> 1. Systeem berekent het totaal aan gewerkte uren voor het (sub)project. 2. Systeem vergelijkt de gewerkte uren met het totaal geschatte uren en berekent hoeveel tijd nog nodig is om de taak af te ronden. 3. Systeem stuurt een bericht naar de accountmanager of projectmanager met een toelichting van het probleem. 4. Systeem slaat de datum waarop de notificatie is verzonden op.
Alternatief stappenplan	Deadline dreigt niet overschreden te worden: 2. Systeem annuleert de berekening.
Postcondities	Er is een notificatie verstuurt naar de accountmanager en projectmanager.
Uitzondering	Deadline dreigt niet overschreden te worden.
Notities	<p>De huidige use case is een versimpelde weergave van wat het systeem precies doet, zie onderstaande notitie voor de details.</p> <p>Bijvoorbeeld: Het project start op 03-09-07 en eindigt op 28-9-07, dit levert 20 beschikbare werkdagen op. Voor het project is daadwerkelijk 10 werkdagen voorbij gegaan sinds de startdatum en zijn 40 uur gewerkt. Dit betekent dat in plaats van de verwachte minimale 5 uur per dag, 4 uur per dag gewerkt is. Dit levert een deadline die op 25 werkdagen ligt in plaats van 20. Het systeem verstuurt een notificatie. Het systeem stuurt geen notificatie binnen de eerst 10% van de beschikbare werkdagen. Dit betekent voor het huidige voorbeeld dat er de eerste 2 werkdagen geen notificatie verstuurt wordt (dagen worden naar boven afgerond). Systeem verstuurt alleen een nieuwe notificatie als er een nieuwe 20% van de beschikbare werkdagen voorbij is. Dit betekent voor het huidige voorbeeld dat als de geprojecteerde uren op werkdag 14 nog steeds de deadline overschrijd er een nieuwe notificatie verzonden wordt.</p>

2.17.2 Urentoekenning dreigt overschreden te worden

Versie	1.1
Samenvatting	Aan de hand van de gewerkte uren en de nog beschikbare uren berekend het systeem of de toegekende uren overschreden dreigen te worden. Als dit zo is stuurt het systeem een notificatie naar de projectmanager en de accountmanager.
Gebruiksfrequentie	Vaak
Complexiteit	Hoog
Prioriteit	Won't have
Actors	Medewerker
Precondities	Gebruiker is geautoriseerd.
Triggers	Medewerker voegt periode toe aan een (sub)project
Stappenplan	<ol style="list-style-type: none"> 1. Systeem berekent totaal aantal gewerkte uren voor het (sub)project. 2. Systeem projecteert het aantal nog te werken uren voor het project met behulp van de gewerkte uren. 3. Systeem stuurt een bericht naar de accountmanager of projectmanager met een toelichting van het probleem. 4. Systeem slaat op bij welk percentage er een waarschuwing is verzonden.
Alternatief stappenplan	<p>Geschatte uren dreigen niet overschreden te worden.</p> <ol style="list-style-type: none"> 2. Systeem stopt de berekening.
Postcondities	Notificatie is verstuurd naar accountmanager en projectmanager.
Uitzondering	Geschatte uren dreigen niet overschreden te worden.
Notities	<p>De huidige use case is een versimpelde weergave van wat het systeem precies doet, zie onderstaande notitie voor de details.</p> <p>Het aantal punten waarop het systeem een bericht verstuurt is instelbaar. Standaard worden 4 punten voorgesteld waarop een notificatie wordt verstuurd.</p> <p>Bijvoorbeeld: Een project van 100 uur zal opgedeeld worden in stukken van 25 uur waarop een notificatie verstuurd wordt.</p> <p>Het systeem bewaard de gegevens van de laatst verzonden notificatie om te voorkomen dat een notificatie niet of meerdere keren verzonden wordt.</p>

3 Rapportage

Door praktische overwegingen zijn de use cases voor genereren van rapportages niet uitgewerkt in het Document van Eisen.

Hiervoor is door de volgende redenen voor gekozen:

Inzicht krijgen in de gewenste rapportages

Er is al enig inzicht in de rapportages die men wenst binnen de vernieuwde PTM. Echter is dit nog niet een compleet beeld van de rapportages die gebruikt worden met behulp van gegevens uit PTM. Tijdens de opstart van het onderzoek zal een verdere verduidelijking komen van de rapportages die men wenst.

Mogelijkheden en onmogelijkheden voor het genereren van rapportages

Tijdens het onderzoek zal pas duidelijk worden wat voor soort rapportages mogelijk zijn. Dit heeft tot gevolg dat geschreven use cases zeer waarschijnlijk herschreven moeten worden of niet mogelijk blijken.

Agile ontwikkelen van PTM

PTM zal volgens de Agile ontwikkel methode geïmplementeerd worden. Dit betekent dat binnen een afgesproken periode een duidelijk omschreven blok werkende functionaliteit opgeleverd wordt. Belanghebbende hebben tijdens het evalueren van de opgeleverde functionaliteit invloed op hoe deze dient te werken. Dit heeft meestal geen grote gevolgen voor use cases en beschikbare gegevens. Wel op het detailniveau van de gegevens (over hoe de gegevens beschikbaar zijn) die verstrekende gevolgen kunnen hebben voor de moeilijkheid voor het genereren van een rapport of een rapport onmogelijk maken.

Verder zal tijdens het ontwikkelen van de rapportages zeer nauw samengewerkt worden met de uiteindelijke gebruikers van de rapportages om de rapportages zo goed mogelijk te laten aansluiten op de werkwijze en de wensen van de gebruikers.

Dit alles zorgt ervoor dat er geen betrouwbare inschatting geleverd kan worden over de complexiteit en daardoor ook geen goede schatting gemaakt kan worden over de tijd die nodig zou zijn voor het implementeren van de use case.

Desalniettemin is uit de feedback van gebruikers op te maken dat men onderstaande soort van rapportages wenst:

Project urenstaat

Een project urenstaat geeft een overzicht van de gewerkte uren, totaal aantal beschikbare (geschatte) uren voor het project, de te factureren uren, de niet te factureren uren en het rendement per werknemer.

Medewerkers rapportage

Toont een totaaloverzicht van de werknemers. In deze rapportage worden de gewerkte uren en de behaalde rendementen vergeleken voor alle werknemers.

Project rapportage

Vergelijking van het huidige project met geselecteerde andere projecten.

Denk aan, overzicht wat ingepland is en wat daadwerkelijk besteed is aan uren en externe kosten.

Urenoverzicht genereren

Deze rapportage geeft dezelfde informatie als de urenoverzicht rapportage in de huidige PTM versie.

Kostenoverzicht genereren

Deze rapportage geeft dezelfde informatie als de kostenrapportage in de huidige PTM versie.

4 Niet Functionele Eisen

4.1 Software Eisen

4.1.1 Toegankelijkheid

Van het systeem moeten minimaal de basisfunctionaliteiten die nodig zijn voor medewerkers om buiten het bedrijfsgebouw uren te registreren beschikbaar zijn via een webinterface. Deze webinterface dient zoveel mogelijk browser onafhankelijk te zijn en in de meest gebruikte browsers zoals Internet Explorer, Firefox en Opera minimaal leesbaar en te gebruiken zijn. Voor deze basis functionaliteiten mogen er ook geen afhankelijkheden zijn met niet in browsers standaard aanwezig functionaliteiten. Zoals bijvoorbeeld Flash of de Java Virtual Machine.

Extra functionaliteiten zoals rapportage en het weergeven van grafieken mogen wel gebruik maken van niet standaard in browsers aanwezige functionaliteit.

4.1.2 Beveiliging

Door de bedrijfskritische informatie die opgeslagen zal worden in PTM moet deze tegen ongewenste toegang en misbruik beschermd worden.

Om ongewenste toegang te voorkomen zal de daadwerkelijk functionaliteit van PTM in een secure omgeving draaien die alleen na authenticatie van een gebruiker beschikbaar is. Dit betekent dat het systeem zal werken met gebruikersprofielen en wachtwoorden. Tijdens de ontwikkeling zal geprobeerd worden om het verzenden van deze wachtwoorden en gebruikersnamen zo veilig mogelijk te maken. Dit om bijvoorbeeld het onderscheppen van inloggegevens te voorkomen.

De functionaliteiten zelf in de secure omgeving zullen alleen beschikbaar zijn voor gebruikers die geautoriseerd zijn om deze te gebruiken. Hiervoor zal een rechten en rollen systeem geïmplementeerd worden. Binnen de software zal een centraal punt ontwikkeld worden waar functies kunnen controleren of ze door de gebruiker uitgevoerd mogen worden. Met een dergelijke aanpak is het rechten systeem zeer eenvoudig te vervangen door een andere implementatie.

4.2 Kwaliteiteisen

4.2.1 Betrouwbaarheid

De software dient van een dergelijke betrouwbaarheid te zijn dat tijdens gebruik geen gegevens onbedoeld verloren gaan.

Dit zal onder andere geïmplementeerd worden door uitgebreid controleren van gegevens en pas als gegevens correct zijn bevonden deze toe te passen binnen het systeem. Mocht het dan alsnog niet lukken dit toe te passen moet het systeem op een dergelijke manier opgezet zijn zodat er geen gegevens verloren gaan.

4.2.2 Onderhoudbaarheid

De software dient te voldoen aan de coding standaard die gebruikt wordt binnen ISAAC. Verder moet de code gedocumenteerd zijn met behulp van JavaDoc.

Dit betekend onder andere dat voor de documentatie en code consistent gebruik wordt gemaakt van een gekozen taal, in het geval van PTM zal er gebruik gemaakt worden van Nederlands. En code dient op een dergelijke manier geschreven te zijn dat deze zo eenvoudig mogelijk te lezen is door andere programmeurs.

Verder zal er gebruik gemaakt worden van diverse design patterns, deze zorgen ervoor de code goed onderhoudbaar is en voor andere programmeurs eenvoudiger te begrijpen is.

4.2.3 Uitbreidbaarheid

De nieuwe versie van PTM zal in de toekomst uitgebreid worden om meer functionaliteit en gebruikersvriendelijkheid te bieden. Hier moet tijdens het ontwerpen en ontwikkelen van PTM rekening mee gehouden worden zodat dit mogelijk is.

Ook moet PTM zo opgezet zijn dat het uit te breiden is met extra 'modules'. Dit om zo toekomstige uitbreidingen mogelijk te maken en er voor te zorgen dat deze zo makkelijk mogelijk geïntegreerd kunnen worden.

Om dit mogelijk te maken zullen er diverse lagen van abstractie aangebracht worden in het programma.

4.2.4 Gebruikersvriendelijkheid

De huidige versie van PTM zal onderzocht worden en bekeken waar de bediening eenvoudiger of gestroomlijnd kan worden.

Een van de methodes die gebruikt zal worden om het gebruik te vereenvoudigen en te stroomlijnen is om het aantal laadmomenten te beperken. Onder andere zal hiervoor zoveel mogelijk functionaliteit die bij elkaar horen of vaak samen gebruikt worden gegroepeerd worden op een scherm. Om er voor te zorgen dat deze schermen werkbaar en gebruikersvriendelijk blijven zal alleen functionaliteit getoond worden die de gebruiker nodig heeft.

Als tweede onderdeel van het project wordt een grafisch weergave module toegevoegd zodat het management de informatie binnen PTM beter kunnen gebruiken.