



Afstudeerverslag

**Ontwikkelen van een urenregistratiesysteem gekoppeld aan het
kaartleessysteem**

Bedrijf/opdrachtgever:	The Tritanium Company BV Kanaalpark 145 2321 JV Leiden
Bedrijfsmentor:	S. Presser
Begeleider:	A.M.J.J. Lousberg-Orbons
Expert:	W. van Vliet
Afstudeerder:	Maries Koole
Email afstudeerder:	I.J.Koole@student.hhs.nl
Studentnummer:	07019009
Plaats:	Leiden
Datum:	6 juni 2011
Versie:	1.0

Referaat

Afstudeer Rapport, ICT, Haagse Hogeschool, Den Haag, Maries Koole, Afstudeerstage, Tritanium, The Tritanium Company BV, SkillsXP, VB.NET, Unified Process, UP, SQL Server 2005, SQL Server 2008

In het kader van de opleiding Informatica aan de Academie van ICT en Media heeft Maries Koole een afstudeeropdracht uitgevoerd, betreffend: Het ontwikkelen van een urenregistratiesysteem gekoppeld aan een kaartleessysteem. De afstudeeropdracht is uitgevoerd bij The Tritanium Company BV in Leiden, gedurende de periode van 07 februari 2011 tot en met juni 2011.

Descriptoren:

- Afstudeer Rapport
- Afstudeerstage
- The Tritanium Company BV
- Tritanium
- SkillsXP
- UP
- Unified Process
- VB.NET
- Maries Koole

Voorwoord

Dit is het eindverslag van de afstudeeropdracht voor de Opleiding Informatica aan de Haagse Hogeschool, Academie voor ICT & Media. Tijdens de opdracht heb ik een uitbreiding gemaakt op het softwarepakket SkillsXP van het bedrijf The Tritanium Company BV.

Na vier jaar studeren is het afstuderen de laatste opdracht voor de studie. In deze vier jaar heb ik veel geleerd, waardoor het afstuderen mogelijk is geworden. Nu ik aan het einde ben van mijn afstuderen zijn er een aantal mensen die ik wil bedanken.

Allereerst wil ik alle docenten van de opleiding Informatica aan de Haagse Hogeschool bedanken voor de kennis die ze hebben overgedragen. Van de docenten wil ik Sarah Mimoun, Alwine Lousberg-Orbons en Willem van Vliet in het bijzonder bedanken voor alle begeleiding tijdens mijn studie.

Als tweede wil ik Sef Presser bedanken, mijn begeleider vanuit Tritanium. Hij investeerde tijd en enthousiasmeerde mij in de opdracht, waardoor mijn motivatie vergroot werd.

Het derde en laatste bedankje wil ik besteden aan Tritanium, die het voor mij mogelijk heeft gemaakt om af te studeren. Tijdens het project had ik alle benodigdheden tot mijn beschikking. Daarnaast wil ik ook mijn collega's bedanken, het samenwerken verliep positief en als ik vragen had namen ze altijd alle tijd om die vragen te beantwoorden.

Leiden, 6 Juni 2011
Maries Koole

Inhoudsopgave

HOOFDSTUK 1 – INLEIDING 1

HOOFDSTUK 2 – ORGANISATIE..... 2

2.1 BEDRIJFSHIËRARCHIE 4

HOOFDSTUK 3 – INITIËLE OPDRACHT 5

3.1 PROBLEEMSTELLING 5

3.2 DOELSTELLING VAN DE OPDRACHT 5

3.3 RESULTAAT 5

3.4 NIVEAU VAN DE OPDRACHT 6

Competenties (beroepstaken)..... 6

Organisatorische doelen 6

HOOFDSTUK 4 – VERLOOP VAN DE OPDRACHT ... 7

4.1 KAARTSCANNERSYSTEEM BIJ STRUKTON RAIL ... 7

4.2 URENREGISTRATIESYSTEEM EN

PLANNINGSYSTEEM 7

4.3 DOELSTELLING 9

4.4 GEKOZEN AANPAK 9

4.5 TWEE UP ITERATIES 9

4.6 PLANNING..... 10

4.7 MIJLPAALPRODUCTEN VOOR DE ITERATIES 11

HOOFDSTUK 5 – ITERATIE ÉÉN:

URENREGISTRATIESYSTEEM12

5.1 INCEPTION FASE 13

Projectopdracht 13

Eisen..... 14

Usecase model..... 16

Onderzoek dataopslag Werktijden 16

5.2 ELABORATION FASE 17

Huidige kaartscanner software..... 17

Databasemodel..... 18

Problemen tijdens de elaboration fase 20

5.3 CONSTRUCTION FASE 21

Problemen en werkzaamheden 21

Singleton 23

Testen 24

Query Latest Workingtime 25

5.4 TRANSITION FASE 26

HOOFDSTUK 6 – ITERATIE TWEE:

PLANNINGSYSTEEM27

6.1 INCEPTION FASE 27

Projectopdracht 27

Eisen..... 28

Functionaliteit..... 29

Betrouwbaarheid 29

Onderhoudbaarheid..... 29

Usecase Model..... 29

6.2 ELABORATION FASE 30

RMS..... 30

Request of Change 31

6.3 CONSTRUCTION FASE 34

Testen 36

6.4 TRANSITION FASE 37

HOOFDSTUK 7 – EVALUATIE 38

7.1 PROCES 38

Gehele afstudeerstage proces 38

Medewerker Tritanium 38

Twee UP fases..... 38

Transition fase Urenregistratiesysteem 38

7.2 PRODUCT..... 39

Urenregistratiesysteem 39

Uitbreiding op het planningsysteem..... 39

7.3 BEROEPSTAKEN 40

Uitvoeren analyse door definitie van

requirements..... 40

Ontwerpen en opstellen gegevensmodel voor

database 40

Ontwerpen systeemarchitectuur 41

Ontwerpen systeemdeel 42

Bouwen applicatie 42

AFBEELDINGLIJST 43

LITERATUURLIJST 44

BIJLAGEN 45

Hoofdstuk 1 – Inleiding

Dit document is het schriftelijke resultaat van de afstudeeropdracht “Ontwikkelen van een urenregistratiesysteem gekoppeld aan het kaartleessysteem”. De opdracht is door P & O Ferries gegeven aan The Tritanium Company B.V. Dit verslag beschrijft hoe het project is verlopen en wat er gebeurd is tijdens het project.

Dit verslag is gericht aan de keurende docenten van de Haagse Hogeschool: Alwine Lousberg-Orbons en Willem van Vliet. Daarnaast is dit verslag ook gericht aan de externe gecommitteerde.

De inhoud van dit document is opgedeeld in hoofdstukken. Het eerste hoofdstuk is de inleiding. Het tweede hoofdstuk betreft mijn afstudeerbedrijf The Tritanium Company B.V. Het derde hoofdstuk bevat een globale omschrijving van de afstudeeropdracht. Het vierde hoofdstuk gaat over het verloop van de opdracht. Het vijfde hoofdstuk gaat over ontwikkeliteratie één: Urenregistratiesysteem. Het zesde hoofdstuk gaat over ontwikkeliteratie twee: Planningsysteem. Hoofdstuk zeven bevat mijn evaluatie van het algehele proces, de afgeleverde producten en mijn beroepstaken. En als laatste zal hoofdstuk acht mijn geraadpleegde literatuur bevatten.

Hoofdstuk 2 – Organisatie

One Solution

TOTAL CONTROL

In gevaarlijke industrieën (zoals olie, gas en nucleaire industrie) kan veiligheidstraining het verschil betekenen tussen leven en dood. Om de spreadsheet van alle behaalde certificaten van elke medewerker bij te houden is een gigantisch karwei. SkillsXP is de oplossing!



The Tritanium Company B.V. is begonnen als de afdeling ICT van Nutech, die later haar bedrijfsnaam veranderde in 'Falck Nutech'. Falck Nutech is een wereldwijd netwerk van Maritime Training Centers.

Falck Nutech wenste op een gegeven moment een systeem voor training en certificeringmanagement van de afdeling. En zo is SkillsXP ontstaan.

Gedreven door de verandering in de markt van de olie en gas, voegde Tritanium het beheren van de competenties van personeel toe aan de applicatie. Deze nieuwe functie maakte de applicatie ook handig en bruikbaar voor andere bedrijven en instellingen.

De applicatie bleef zich verder ontwikkelen, en op een gegeven moment was SkillsXP al geliefd bij grote, internationale bedrijven, zoals Pride International en Fugro.

Sinds 2000 was Tritanium niet langer een afdeling, maar een zeer hooggewaardeerd lid van de Global Competence B.V.

De Global Competence B.V. biedt allerlei oplossingen aan bedrijven die te maken hebben met de bovenstaande gebieden, zoals olieplatformen, of andere risicovolle omgevingen.

Op dit moment is Tritanium nog steeds bezig om de applicatie SkillsXP succesvoller te maken. SkillsXP wordt van steeds meer gemakken voorzien en wordt ook steeds van nieuwe functionaliteit voorzien zodat de gebruikers SkillsXP voor nog meer doeleinden kunnen gaan gebruiken.

Soms is het nodig, dat klanten van Tritanium speciale training nodig hebben om erachter te komen hoe de applicatie SkillsXP in elkaar zit. Natuurlijk zijn er al vele handleidingen geschreven, maar het komt nog vaak voor, dat klanten of de handleidingen niet begrijpen, of ze kunnen niet de benodigde informatie eruit halen. Tritanium doet haar uiterste best om de handleidingen van SkillsXP zo goed mogelijk te schrijven, en de trainingen en presentaties zo goed mogelijk te geven.

Meestal vinden die speciale trainingen en/of presentaties in het buitenland plaats, omdat Tritanium over veel grote, internationale bedrijven als klanten beschikt.

De klanten van Tritanium die SkillsXP hebben aangeschaft, hebben zowel Onshore als Offshore locaties. Op de Offshore locaties wordt er flink met mechanica gewerkt om naar olie te boren of gas op te slaan. Dat is zware en gevaarlijke arbeid waar de nodige certificaten vereist voor zijn. SkillsXP wordt gebruikt door de Onshore locaties, oftewel: de kantoorvestigingen van de klant. SkillsXP houdt dus onder andere bij, welke certificaten voor welke rol/functie vereist zijn, en welke werknemer op de Offshore locatie al over dat ene certificaat beschikt, of niet.

SkillsXP wordt op dit moment ook gebruikt voor de volgende handelingen:

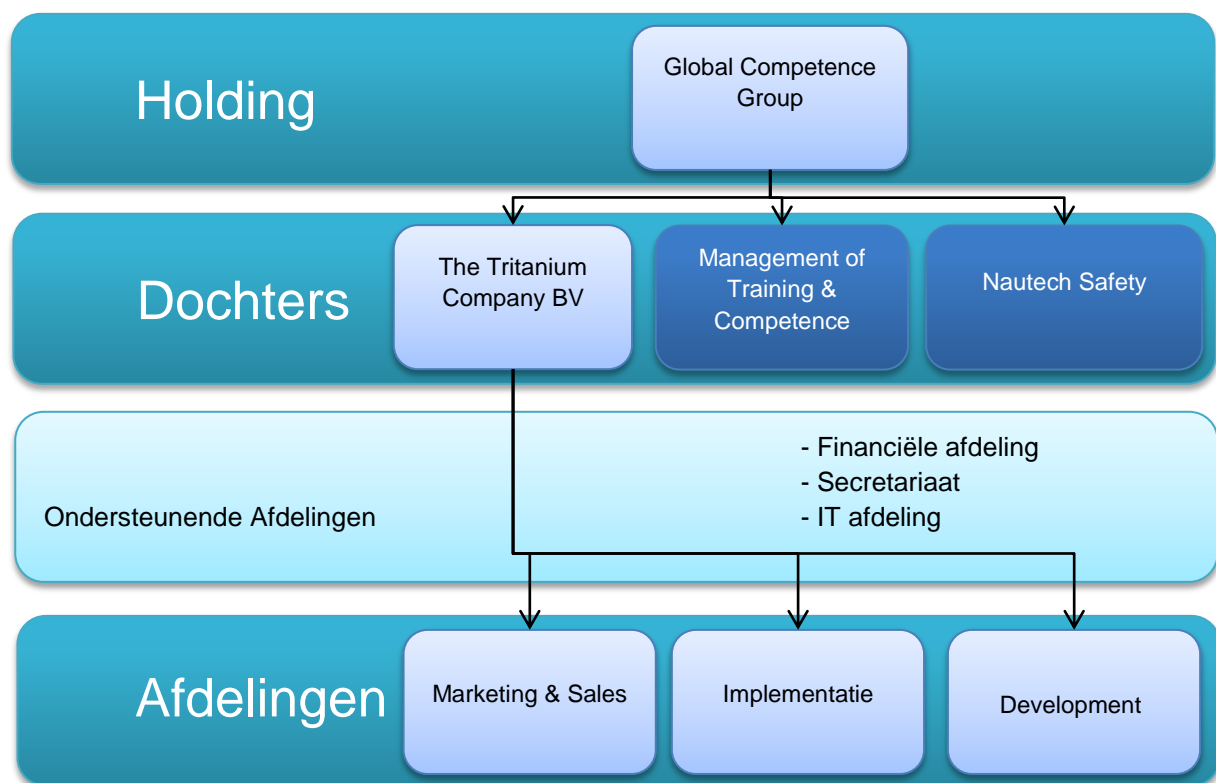
- Registreren en bijhouden van competenties en certificeringen (toetsen) van personeel;
- De planning van projecten, project locaties en werknemers wordt bijgehouden;
- Assessments uitvoeren op personeel om een controle te doen op de huidige functie;
- Het genereren van rapporten met de gewenste data.

2.1 Bedrijfshiërarchie

The Tritanium Company is een dochteronderneming van de Global Competence Group Holding. Deze holding heeft drie onderdelen, namelijk:

- The Tritanium Company B.V.
- Nautech Safety Equipment
- Management of Training & Competence BV (M.T.C.)

In onderstaand schema is goed te zien dat de andere 2 dochterondernemingen niet nader belicht zullen worden. Schematisch gezien ziet de bedrijfshiërarchie van The Tritanium Company er als volgt uit:



Figuur 1 - Organogram Tritanium – Ik maak deel uit van de 'Development' afdeling

Doordat de directeur (Edi Gittenberger) vaak weg is, is de sturing van bovenaf klein. Daarnaast is the Tritanium Company een klein bedrijf. Dit bij elkaar opgeteld vraagt een grote zelfstandigheid en verantwoordelijkheid van de medewerkers. Ook zonder controle en directe aansturing moeten zij zelfstandig keuzes kunnen maken en proactief werken. Door het opzetten van aparte afdelingen zijn de taken wel opgesplitst. In bovenstaand organigram ziet u dat the Tritanium Company bestaat uit drie hoofdafdelingen. Daarnaast opereren er drie kleinere, ondersteunende afdelingen.

Hoofdstuk 3 – Initiële Opdracht

In dit hoofdstuk zal de initiële opdracht worden behandeld. De initiële opdracht is de opdrachtsomschrijving die opgesteld is voor mijn afstudeerperiode.

3.1 *Probleemstelling*

Voor Tritanium is hun competentie management software erg belangrijk. Er vinden dan ook continue vernieuwingen en wijzigingen plaats. Een van de vernieuwingen is de doorontwikkeling van het planningsgedeelte waarbij het mogelijk moet worden om personen op projecten te plannen, de projecten aan een afdeling te koppelen en de afdeling weer aan een klant. Aan de GUI kant moet de planning zichtbaar worden gemaakt dmv een grafische GANTT weergave, waarbij gebruikersvriendelijkheid extra aandacht verdient. Daarnaast moet de software ook verschillende rapporten kunnen weergeven, gekoppeld aan een kaartscannersysteem.

Op dit moment is het alleen mogelijk personen op projecten te zetten, maar niet deze projecten aan klanten en aan afdelingen (boten of olieboorplatforms) te koppelen. Dit laatste is echter belangrijk omdat de software wel in staat moet zijn om via rapportages te bepalen of iemand geschikt is voor een bepaald project en of hij voldoet aan de projecteisen, de klanteisen en de afdelingseisen.

De competentie management software is uitgerust met een kaartscannersysteem. Bij elk bedrijf waar de competentie management software is geïmplementeerd lopen de werknemers met een kaart/pasje op zak. Op dit moment kan via een kaartlezer een werknemer aflezen welke competenties, certificeringen en toetsen hij heeft afgerond of nog moet behalen en eventueel de vervaldatum ervan.

In de toekomst moet het kaartscannersysteem worden gebruikt voor het in- en uitchecken van werknemers. De werknemers moeten straks aan het begin van hun dag inchecken en aan het eind weer uitchecken.

3.2 *Doelstelling van de opdracht*

De doelstelling houdt in dat er twee systemen worden gebouwd/uitgebreid, namelijk:

- Bouw van een in- en uitcheck module in de huidige kaartscanner software
- Uitbreiding van het planningsysteem

3.3 *Resultaat*

Als de afstudeeropdracht is uitgevoerd zal er een urenregistratiesysteem zijn in de vorm van het in- en uitchecken van de werknemers met hun kaart/pasje. Er zal een uitbreiding/nieuwe module komen in het planningsysteem. Hiervoor zullen eerst de nodige requirements verzameld moeten worden, een ontwerp moeten worden gemaakt en een testplan opgesteld worden.

3.4 Niveau van de opdracht

Tijdens de afstudeeropdracht zal ik een aantal beroepstaken moeten gaan gebruiken om de opdracht tot een goed einde te brengen. Een aantal van deze beroepstaken heb ik gekozen om op beoordeeld te worden tijdens mijn gehele afstudeerstage. Een lijst van deze gekozen beroepstaken is hieronder te vinden.

Competenties (beroepstaken)

Analyseren en adviseren	Niveau
1.4 uitvoeren analyse door definitie van requirements	3
Databaseontwerp en -ontwikkeling	
2.1 ontwerpen en opstellen gegevensmodel voor database	3
Softwareontwerp en -ontwikkeling	
3.1 Ontwerpen softwarearchitectuur	3
3.2 ontwerpen systeemdeel	3
3.3 bouwen applicatie	4

Organisatorische doelen

Tevens heb ik een aantal organisatorische doelen voor ogen.

- Bewijzen goed te kunnen overleggen met collega's en mijn afstudeermentor voordat ik belangrijke keuzes ga maken.
- Bewijzen dat ik alle ontwikkelfasen in één project kan doorlopen.
- Bewijzen dat ik alle eisen helder en gedetailleerd op papier krijgen.

Hoofdstuk 4 – Verloop van de opdracht

Ik dit hoofdstuk beschrijf ik het verloop van de opdracht. Dit houdt onder andere in welke veranderingen er waren ten opzichte van de initiële opdracht, hoe ik te werk ben gegaan, welke keuzes ik gemaakt heb en mijn argumenten voor het maken van een aantal keuzes.

4.1 Kaartscannersysteem bij Strukton Rail

Al vroeg tijdens het proces begreep ik van mijn afstudeerbegeleider, S. Presser, dat er binnenkort een pilot project gepland stond bij de klant Strukton Rail B.V. Omdat ik tijdens mijn afstuderen de kaartscanner software ging uitbreiden leek mijn begeleider het slim om mij te betrekken bij het opzetten en uitvoeren van het pilot project van Strukton.



Strukton Rail B.V. was tot januari 2011 een dochteronderneming van de NS en is sindsdien een zelfstandige B.V.. Strukton is verantwoordelijk voor het spooronderhoud in heel Nederland en is daarnaast betrokken bij een aantal technische projecten door

Europa heen. In de spoorbeheer branche zijn strenge eisen wat betreft het bijhouden van competenties en certificeringen van werknemers. Het gebeurt daarom regelmatig dat bedrijven zoals Strukton een audit krijgen.

Tijdens een audit worden alle werknemers gecontroleerd of ze beschikken over de juiste certificeringen voor de functie die ze uitvoeren. Momenteel loopt iedere werknemer met een zogenaamd 'groene paspoort' op zak. In dit boekje staan alle behaalde certificaten met bijbehorende trainingscentrum, behaal- en vervaldatum. Tijdens een audit worden alle groene paspoorten gecontroleerd en vervolgens nog een tweede controleslag op kantoor om te verifiëren of alle certificaten echt behaald zijn. Strukton is op zoek naar een manier om dit gemakkelijker te doen, en is zo bij Tritanium terecht gekomen.

Tijdens een gesprek met de medewerkers van Strukton is gebleken dat ze een aantal requirements hadden wat betreft de kaartscanner software. Strukton wil dat de indeling van het certificaten scherm meer ging lijken op de indeling van hun groene paspoort zodat de medewerkers het scherm makkelijker konden herkennen en aflezen. Daarnaast was er een wens voor een aantal filter opties die bepalen welke certificaten zichtbaar worden.

4.2 Urenregistratiesysteem en Planningsysteem



Het urenregistratiesysteem en de uitbreidingen op het planningsysteem worden voor een klant van Tritanium gemaakt, namelijk voor P & O Ferries. P & O heeft een deal

gesloten met Tritanium over nieuwe functionaliteit in SkillsXP. Zodra de nieuwe functionaliteit gebouwd en getest is gaat P & O SkillsXP in gebruik nemen.

P & O biedt een aantal diensten en services aan om passagiers van A naar B te vervoeren. Naast passagiers worden er ook containers getransporteerd. P & O is voornamelijk actief in de landen Groot-Brittannië, Frankrijk en Nederland.

P & O wil op een gemakkelijkere wijze haar vaartplanning, personeelplanning, loonadministratie, aanwezigheid personeel, personeelsgegevens en de functies en certificaten van het personeel bijhouden. Kortom P & O gaat SkillsXP gebruiken.

Urenregistratiesysteem

Het urenregistratiesysteem houdt in dat de kaartscanner software uitgebreid moet gaan worden met in- en uitcheck functionaliteit. Deze functionaliteit zal het onderdeel 'aanwezigheid personeel' mogelijk maken.

Planningsysteem

Het huidige planningsgedeelte van SkillsXP heeft een aantal beperkingen waar P & O tegenaan loopt. Op dit moment is het bijvoorbeeld zo dat in het planningssysteem een project, een persoon op een project en de persoonlijke planning alleen met een begin- en einddatum kunnen worden gepland. Dit gaat uitgebreid worden naar datum en tijd zodat er meerdere projecten per dag opgeslagen kunnen worden. Omdat een groot deel van de klanten hier geen behoefte aan heeft, moet de gebruiker zelf kunnen kiezen of de weergave van de planning items datum of datum plus tijd moet worden.

In de vorige ontwikkeliteratie heb ik een urenregistratiesysteem gebouwd. Het urenregistratiesysteem wordt gebruikt om de in- en uitcheck tijden van werknemers op te slaan. Omdat het fysiek in- en uitchecken foutgevoelig is moet er een manier komen om de in- en uitcheck tijden te kunnen beheren. Met beheren wordt bedoeld het zelf toevoegen, wijzigen en verwijderen van werktijden.

De data die het urenregistratiesysteem opslaat moet weergegeven kunnen worden in een aantal rapporten binnen SkillsXP. Een gebruiker moet rapporten kunnen genereren met de gewenste informatie. Deze rapporten zullen bijvoorbeeld gebruikt worden voor de loonadministratie.

Er is ook behoefte aan een scherm waarop per ingepland project te zien is wie van de ingeplande werknemers aanwezig is en wie nog niet. Deze informatie zal worden gehaald uit het urenregistratiesysteem.

Er is ook behoefte om meer informatie achter de projectlocaties op te gaan slaan. Er moeten één of meer scenario's en accommodaties aan project locaties gehangen kunnen worden. In een scenario moet worden opgeslagen welke accommodaties actief zijn tijdens dat scenario en welke type medewerkers er gepland kunnen worden tijdens het actieve scenario met het bijbehorende minimum en maximum.

4.3 Doelstelling

De doelstelling houdt in dat SkillsXP dusdanig wordt uitgebreid zodat het aan de eisen van P & O voldoet. Hiervoor zal een eisenanalyse nodig zijn. In de analyse kan elke eis van P & O op één van de volgende manieren afgehandeld worden:

- Toevoeging van functionaliteit in SkillsXP;
- Verandering van functionaliteit in SkillsXP;
- Is al reeds aanwezig in SkillsXP;

Vervolgens ga ik uitzoeken hoeveel tijd alles bij elkaar gaat kosten. Mocht ik niet genoeg tijd hebben om alle onderdelen te kunnen maken, zal ik in overleg een aantal onderdelen overdragen naar andere programmeurs.

Het tweede deel van de doelstelling is de uitbreiding van de kaartscanner software. Voor Strukton moeten er een aantal requirements worden geïmplementeerd en voor P & O moet er checkin en –uit functionaliteit worden geïmplementeerd in de kaartscanner software. Omdat Strukton en andere toekomstige gebruikers wellicht niet het in- en uitchecken willen gebruiken zal dat functionaliteit dus generiek moeten zijn.

Daarnaast wordt er tijdens mijn afstudeerperiode een pilot project gedraaid bij Strukton met mijn herziene kaartscanner software.

4.4 Gekozen aanpak

Ik heb gekozen om het softwareontwikkelproces structuur te geven met de methode Unified Process (UP). Uit ervaringen met deze softwareontwikkelmethode blijkt dat het project gestructureerd en overzichtelijk blijft.

UP werkt met vier fases:

- De inception fase, in deze fase wordt het project globaal gepland en wordt er begrepen wat er gebouwd moet worden;
- De elaboration fase, in deze fase wordt uitgezocht hoe er gebouwd gaat worden, denk aan ontwerpen en modellen die gemaakt worden;
- De construction fase, in deze fase ligt het accent op het bouwproces van het project;
- De transition fase, in deze fase wordt het project overhandigd aan de opdrachtgever en wordt er gekeken hoe het systeem geïmplementeerd kan worden bij een klant.

4.5 Twee UP iteraties

Na een globaal idee te hebben van de requirements van beide systemen (Zoals beschreven in paragraaf 4.2) en daarnaast de startdatum van de pilot van Strukton heb ik de keuze gemaakt om twee keer het gehele UP proces te doorlopen. De eerste UP iteratie zal de uitbreiding van de kaartscanner software zijn en de tweede iteratie zal alle uitbreidingen bevatten voor het planningsgedeelte in SkillsXP.

Omdat de kaartscanner software los van de hoofdapplicatie SkillsXP staat hoefde ik me pas te verdiepen in SkillsXP nadat de eerste UP iteratie (uitbreiding van de kaartscanner software) was afgerond. Dit leverde tijdwinst op wat de startdatum van de pilot van Strukton haalbaar maakte.

Wat de exacte architectuur is omtrent SkillsXP en de kaartscanner software zal ik uitleggen in Hoofdstuk 5 – Iteratie één: Urenregistratiesysteem.

4.6 Planning

Kalenderweek →	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Afstudeerperiode week →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Activiteit ↓																		
Fase: Inception																		
Algemeen plan van aanpak	1																	
Plan van aanpak	1						.5											
Inception Rapport	2						1											
Requirements opstellen door contact opdrachtgever/klant						1		1										
Inwerken in het huidige systeem	1	5	3					2	1									
Fase: Elaboration																		
Elaboration Rapport			2						2									
Technisch rapport database structuur				1														
Testplan				1					1									
Fase: Construction																		
Construction Rapport				2							1			1				
Bouw van de urenregistratie software				1	5	3												
Bouw/uitbreiden van het planningsysteem									1	5	4	5	5	2				
Testen van de software						1								2	1			
Fase: Transition																		
Transition Rapport							.5								2			
Systeemdocumentatie															1			
Afstudeerdossier							3	2							1	5	5	1

Als de transition fase van het urenregistratiesysteem begint zal ik ondertussen ook al beginnen met het verzamelen van de requirements van het planningsysteem. Dit omdat de kans sterk aanwezig is dat verzamelen van de gedetailleerde requirements van het planningsysteem tijdrovend gaat zijn. Als er onduidelijkheden ontstaan over requirements moet ik contact gaan zoeken met mensen die niet meteen een antwoord paraat hebben of die gewoonweg niet beschikbaar zijn. Omdat dit veel tijd kan gaan kosten begin ik al eerder met het verzamelen van de requirements.

4.7 Mijlpaalproducten voor de iteraties

De mijlproducten zijn samengesteld door de richtlijnen van de UP producten. Daarnaast is er een product te zien wat niet direct bij een UP product hoort: Een databasestructuur onderzoeksrapport voor de data van het in- en uitchecken.

Omdat het opslaan van werktijden van personeel in grote organisaties binnen een paar jaar erg veel data kan worden moet er onderzocht worden of dit de performance van de database kan beïnvloeden. Dit zal worden onderzocht tijdens ontwikkeliteratie één.

De volgende producten (met daarbij passende omschrijving) zullen per ontwikkeliteratie opgeleverd worden:

- Een *inceptionrapport*
 - ➔ Het *inceptionrapport* zal een rapport zijn over de eerste fase van het project. Hierin worden de volgende punten meegenomen:
 - Het Plan van Aanpak;
 - De vernieuwde Use Cases;
 - De nieuwe functionele eisen, opgesteld in een MoSCoW principe;
- Een databasestructuur onderzoeksrapport voor de data van het in- en uitchecken (**Alleen 1^e iteratie**)
 - ➔ Hierin worden de bevindingen en consequenties vastgelegd van de opslaglocatie van de data van het kaartleessysteem.
- Het *elaborationrapport*
 - ➔ Het *elaborationrapport* zal een rapport zijn over de tweede fase van het project. Hierin worden de volgende punten meegenomen:
 - Het klassendiagram;
 - Een statediagram;
 - De uiteindelijke usecase diagrammen + beschrijvingen;
- Het testplan
 - ➔ Hierin staat het plan vermeld voor het uitvoeren van de tests op het urenregistratiesysteem.
- De source code
 - ➔ De uiteindelijke software.
- Het *constructionrapport*
 - ➔ Het constructionrapport zal een rapport zijn over de derde fase van het project. Hierin worden de volgende punten meegenomen:
 - Final versie klassendiagram;
 - Final versie Usecasediagram + beschrijvingen;
 - Final versie Statediagrammen;
 - Het testrapport met daarin de uitkomst van het testplan.
- Het *transitionrapport*
 - ➔ Het transitionrapport zal een rapport zijn over de vierde fase van het project. Hierin worden de volgende punten meegenomen:
 - Het installeren van de kaartscanner software; (**Alleen 1^e iteratie**)
 - Een manual over hoe de nieuwe functionaliteit bediend kan worden.

Hoofdstuk 5 – Iteratie één: Urenregistratiesysteem

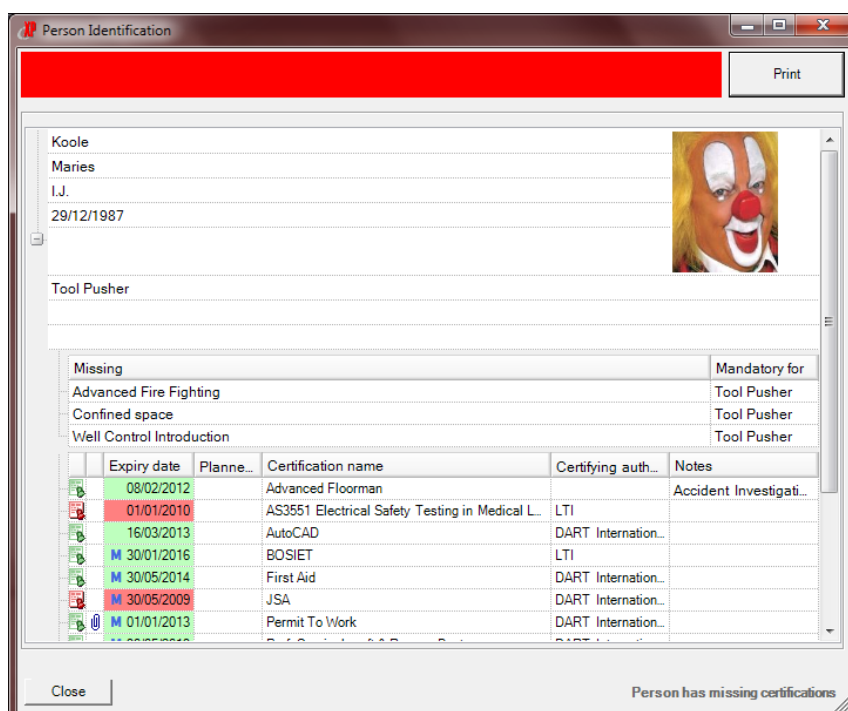
De ontwikkeling van het urenregistratiesysteem houdt in: het uitbreiden van de huidige kaartscanner software van Tritanium. De huidige versie van de kaartscanner software kan werknemers koppelen aan pasjes en een controle doen op een werknemer. Wat deze controle inhoudt staat in de onderstaande alinea.

Achtergrond

Een aantal jaar geleden is er voor de klant BIS Industrial Services kaartscanner software ontwikkeld. Met die software en een kaartscanner kan op een werkplaats terplekke worden gecontroleerd of een werknemer de juiste certificaten in zijn/haar bezit heeft om de werkplaats te mogen betreden. Het controleren gaat met een persoonsgebonden werknemerspas en een kaartscanner die data ophaalde van de SkillsXP database en dat naar het scherm bracht.



Figuur 3 - Kaartscanner met pasjes



Figuur 2 - Origineel certificaten scherm

5.1 Inception Fase

In de inception fase wordt het urenregistratieproject globaal gepland en wordt er begrepen wat er gebouwd moet worden.

Projectopdracht

De projectopdracht voor deze iteratie is al grotendeels in het vorige hoofdstuk “verloop van de opdracht” besproken. Ik zal hier een korte samenvatting van de projectopdracht geven.

Strukton Rail gaat binnenkort een pilot project met Tritanium starten. Daarbij gaat de kaartscanner software gebruikt worden zodat de competentiegraad van haar werknemers makkelijker gecontroleerd kan worden. De kaartscanner software gaat het huidige ‘groene paspoort’ systeem vervangen. Het ‘groene paspoort’ is een papieren boekje dat elke werknemer altijd bij zich moet hebben. In dit boekje staan alle certificaten met bijbehorend trainingscentrum, behaal- en vervaldatum. Voordat de pilot kan starten wil Strukton een paar veranderingen aan de kaartscanner software. De indeling van het certificaten scherm moet lijken op de indeling van het groene paspoort. En een aantal filter opties zijn gewenst die bepalen welke soorten certificaten zichtbaar zijn.

P & O wil dat haar werknemers via de kaartscanner software gaan in- en uitchecken (Denk aan de OV-chipkaart). Deze gegevens zullen vervolgens gebruikt worden voor de aanwezigheid van werknemers en voor de loonadministratie.

De uitbreidingen op de kaartscanner software moeten op een generieke manier worden gebouwd omdat niet alle gebruikers de kaartscanner software als urenregistratiesysteem willen gebruiken.

Eisen

Functionele eisen

De eisen voor het urenregistratiesysteem heb ik opgemaakt uit het dossier van P & O. Vooraf aan mijn afstudeerstage is er al veel overleg geweest tussen P & O Ferries en Tritanium. Al deze documenten en mails worden altijd uitgeprint en in het klantdossier geplaatst. Ik heb het dossier doorgenomen en het tweetal collega's die het meest met P & O hebben overlegd geïnterviewd waardoor ik alle eisen voor het urenregistratiesysteem op een rijtje had.

In het P & O dossier stonden weinig (technische) details die met het urenregistratiesysteem te maken hadden. P & O heeft wel benadrukt dat het systeem zeer gebruikersvriendelijk moet zijn. De twee eisen die staan onder 'Should have' heb ik uit mijn eigen initiatief toegevoegd om het systeem gebruikersvriendelijker te maken. De twee eisen staan op een lagere prioriteit als de 'Must have' eisen omdat ze niet tot de 'core functionaliteit' van het urenregistratiesysteem behoren.

De eisen van Strukton heb ik in overleg met de medewerkers van Strukton vastgesteld.

Must have

- **[P & O]** Het systeem moet het tijdstip van het swipen van de werknemerspas opslaan als inchecktijd als de pas gescand wordt bij een incheck scanner of als uitchecktijd als bij een uitcheck scanner.
- **[P & O]** De locatie van de scanner moet ook worden opgeslagen. Dit kan uiteindelijk een scanner nummer of een locatie nummer worden.
- **[P & O]** De tijdzone tijdens het in- en uit checken moet achterhaald en opgeslagen worden via het Windows systeem waar de kaartscanner software op draait.
- **[Strukton]** De indeling van het certificaten scherm moet dezelfde indeling krijgen als het 'groene paspoort' dat bij Strukton wordt gehanteerd.
- **[Strukton]** Certificaten die binnen een vooraf ingestelde aantal maanden verlopen, moeten een oranje achtergrond krijgen in het certificaten scherm.
- **[Strukton]** Er moeten een aantal filter opties komen die bepalen welke certificaten zichtbaar worden in het certificaten scherm. Namelijk de volgende:
 - 'Alleen hoofdrol certificaten' en/of 'Alleen Bijrol certificaten';
 - 'Alleen rol certificaten' of 'Alle certificaten';
 - 'Mandatory' en/of 'Recommended' en/of 'Not required'.

Should have

- **[P & O]** Binnen 60 seconden na een tijdsregistratie kan geen tweede tijdsregistratie plaats vinden
- **[P & O]** Feedback naar de werknemer, dat het in- of uitchecken is geslaagd in de vorm van tekst op een beeldscherm, audio of visueel

Niet-functionele eisen

De niet-functionele eisen heb ik samengesteld in overleg met mijn collega's van development. De niet-functionele eisen hebben immers met de ontwerpkeuzes van de software te maken en niet met de eisen van de klant. De volgende niet-functionele eisen zijn waargemaakt:

Functionaliteit

De applicatie moet geschikt zijn voor alle geïnteresseerde en toekomstige klanten van Tritanium. Dat houdt in dat een groot deel van de kaartlezer software zo generiek moet zijn dat het systeem naar de wens van de klant kan worden ingesteld. Denk hierbij aan de volgende combinaties:

- Certificaten controle(simpel), de certificaten worden gecontroleerd op vervaldatum en er wordt gecontroleerd of de werknemer alle verplichte certificaten in bezit heeft voor zijn huidige functie.
- Certificaten controle(geavanceerd) Deze optie is hetzelfde als Certificaten controle(simpel) plus een aantal filter opties voor de certificaten die zichtbaar worden.
- Certificaten controle + Urenregistratiesysteem. De in- of uitchecktijd wordt per swipe van de werknemers opgeslagen; afhankelijk of de persoon bij de incheck of de uitcheck zijn pas swiped. Bij het swipen van de werknemerspas kan de werknemer ook zien of hij verlopen certificaten heeft alsmede alle verplichte certificaten in bezit heeft voor zijn huidige functie.
- Urenregistratiesysteem. Bij deze optie worden geen certificaten opgehaald. Puur de swipe tijden worden opgeslagen en worden afgedrukt op het scherm.

Als het urenregistratiesysteem actief is kan de gebruiker ook instellen welke vorm van feedback er gegeven moet worden. Dit kan één of meerdere van de volgende opties zijn:

- Geluid
- Lampje met een bepaalde kleur
- Tekst op een scherm

Beveiliging

De applicatie moet goed beveiligd zijn omdat het urenregistratiesysteem een makkelijk doelwit zou kunnen zijn voor hackers. Een hacker zou kunnen invoeren dat hij afgelopen week X aantal uur heeft gewerkt.

De klanten die met SkillsXP werken krijgen van Tritanium de keuze of ze gebruik willen maken van Tritanium's database hosting tegenover hun eigen database hosting. Mocht de klant kiezen voor een eigen database omgeving geven wij een aantal minimale eisen om deze twee redenen:

- Zodat wij onze software niet aan oude versies van database software hoeven aan te passen.
- Omdat de nieuwste database software mee wordt genomen met de nieuwe antivirus definities, zijn de nieuwere systeem vaak minder kwetsbaar dan verouderde database systemen.

Betrouwbaarheid

Het urenregistratiesysteem dient getest te worden.

Onderhoudbaarheid

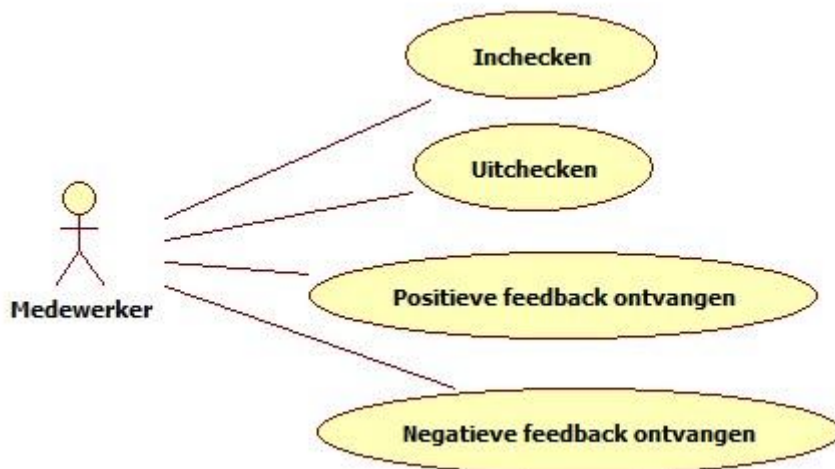
Het systeem moet goed te analyseren zijn, dus er wordt gebruikgemaakt van systeemdokumentatie en extra commentaar waar nodig. De applicatie moet ook goed aan te passen zijn omdat er eventuele uitbreidingen bij kunnen komen.

Portabiliteit

Het urenregistratiesysteem moet gemakkelijk te installeren en te gebruiken zijn door de ontwikkelaars van Tritanium. Tevens moet er rekening worden gehouden met de portabiliteit van het gehele systeem, de database verbinding moet zo generiek mogelijk zijn. In sommige gevallen staat de database lokaal en in andere gevallen extern.

Usecase model

Het usecase model van het urenregistratiesysteem.



Figuur 4 - Usecase model

Naam	Inchecken
Samenvatting	De begintijd van de werknemer is geregistreerd in het systeem aan het begin van zijn/haar werkdag
Actoren	Medewerker van een bedrijf
Aannamen	Het kaartleessysteem draait en kan de werknemerspas scannen. De werknemerspas is onbeschadigd.
Beschrijving	1.) De werknemer haalt zijn pas langs een scanner 2.) De code van de werknemerspas is gescand. 3.) Het werknemersnummer dat bij de pas hoort wordt opgehaald en wordt samen met de tijd en locatie naar de database waar de urenregistratiedata wordt opgeslagen geschreven.
Uitzonderingen	Geen
Resultaat	De begintijd is in de database opgeslagen

Figuur 5 - Usecase beschrijving Inchecken

Onderzoek dataopslag Werktijden

Tijdens deze inception fase van het urenregistratiesysteem heb ik een onderzoek uitgevoerd naar de dataopslaglocatie van de werktijden. Het aantal records in een werktijden tabel kan immers hard oplopen bij een bedrijf met veel werknemers.

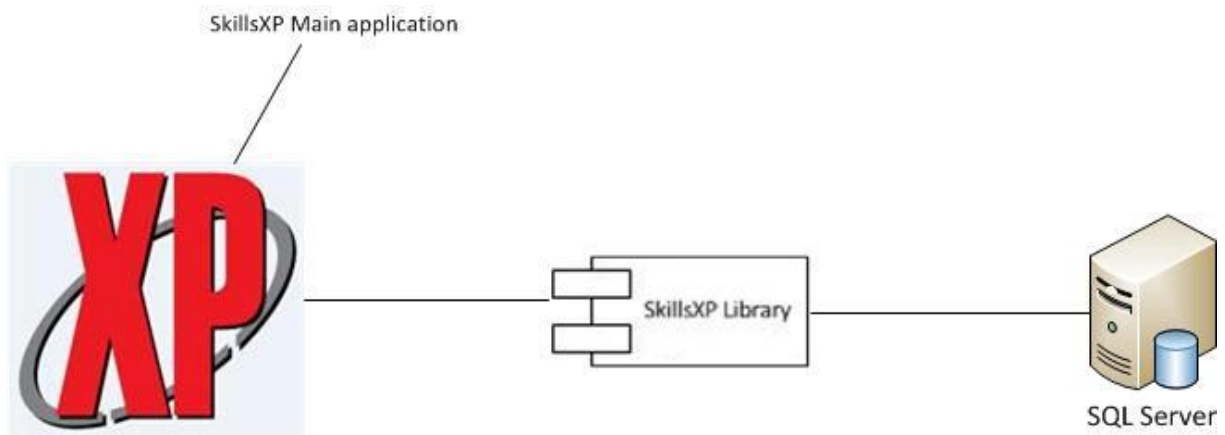
Voor het onderzoek ben ik uitgegaan van 500 actieve personeelsleden die elk 250 keer per jaar in en uitchecken. Na een gebruik van 5 jaar zouden er 625.000 records in de tabel staan. Zelfs toen er 625.000 records in de tabel stonden bleven de laadtijden van de gewenste data erg snel. Ik heb uiteraard een index op het personeels nummergezet (WotiPersGuid) omdat dit veld ongetwijfels het meeste word aangevraagd. Dit kwam de performance erg ter goede. De conclusie was uiteindelijk dat mijn werktijden gewoon in de hoofddatabase van SkillsXP mochten komen omdat dit geen gevaar voor de performance kon worden.

5.2 Elaboration Fase

Tijdens de elaboration fase van het urenregistratiesysteem heb ik gekeken waar de problemen gingen liggen tijdens de ontwikkeling. Daarnaast is er in die fase een eerste versie van het databasemodel en klassediagram gemaakt. Als eerste heb ik me verdiept in de huidige kaartscanner software van Tritanium. Daarna heb ik me verdiept in de database waar de werktijden in opgeslagen gaan worden. Vervolgens heb ik een klassendiagram en databasemodel gemaakt.

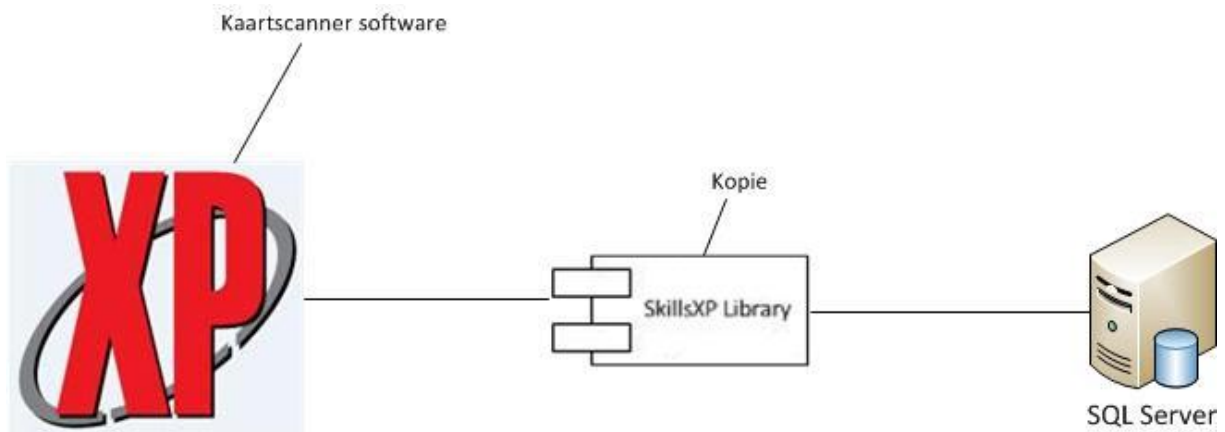
Huidige kaartscanner software

Als eerste heb ik me verdiept op architectuur niveau. Om de architectuur van de kaartscanner software te begrijpen zal ik eerst de globale architectuur van SkillsXP zelf uitleggen. SkillsXP gebruikt een aantal library's om te praten met de database(SQL Server). Tritanium heeft een versiebeheer voor de database. Dit is om eventuele compatibiliteitsproblemen te voorkomen mochten gebruikers gaan hun SkillsXP gaan upgraden.



Figuur 6 - SkillsXP globale architectuur

In de kaartscanner software is gekozen om de library's die SkillsXP gebruikt om met de database te communiceren, te kopiëren. Hierdoor hoeft SkillsXP niet geïnstalleerd te zijn als een pc met de kaartscanner software moet werken. In de library's wordt het databaseversie nummer opgeslagen. Daardoor was ik in staat een controle in te bouwen die de versie van de database controleert. Mocht de databaseversie niet meer overeen komen met de database versie die de library's verwachten, zal er een foutmelding verschijnen die meldt dat kaartscanner software een update moet krijgen.



Figuur 7 - Kaartscanner software globale architectuur

Database model

Voor het urenregistratiesysteem heb ik een tabel toegevoegd aan de SkillsXP database. Het ontwerp van de Workingtimes tabel was niet zo makkelijk als het leek. Want hoe sla je het beste checkin en checkuit tijden op? Het betreft in dit geval vaarten die soms meerdere dagen kunnen duren maar ook vaarten die maar een aantal uur duren. Dat betekent dat ik niet van een standaard werkdag uit kan gaan waarbij de eerste kaartregistratie van de dag de checkin tijd is en de tweede de checkuit tijd. Ik moest er achter komen per kaartregistratie of de persoon wilt in- of uitchecken; dit kon op twee manieren:

- De persoon geeft aan te willen in of uitchecken.
- Het systeem bepaald afhankelijk van de vorige kaartregistratie van de persoon of de persoon nu in- of uitcheckt.

De tweede manier vond ik niet gebruikersvriendelijk genoeg want; stel de werknemer checkt in op dag één, vergeet vervolgens uit te checken. De werknemer is vervolgens weer aanwezig op dag twee en doet een kaartregistratie; het systeem zegt dan dat de persoon uitgecheckt is. De persoon wil dan meteen weer een kaartregistratie doen om daadwerkelijk in te checken. Echter omdat het systeem verbiedt dat dezelfde persoon twee keer een kaartregistratie doet binnen 60 seconden zal het systeem een foutmelding geven. De werknemer moet de 60 seconden afwachten voordat hij/zij in kan checken.

Na een gesprek met de lead programmer en de hoofd tester van Tritanium wist ik dat het veel slimmer was om twee scanners te gaan gebruiken, één voor het inchecken en één voor het uitchecken. Omdat een scanner immers maar rond de 60 euro kost terwijl een software oplossing in de duizenden euro's kan lopen, is het beter om voor de meest gebruikersvriendelijke oplossing te kiezen. Met twee scanners was het makkelijk om een gebruikersvriendelijke oplossing te bedenken; één scanner voor inchecken en één voor het uitchecken.

Vervolgens stond ik voor de keuze in welke vorm ik de in- en checkuit tijden wou gaan opslaan.

PersonID	Datum + tijd	Checkin/Checkuit
12345	01-01-2011 09:30:16	IN
12345	01-01-2011 17:06:50	UIT

PersonID	Incheck datum + tijd	Uitcheck datum + tijd
12345	01-01-2011 09:30:16	01-01-2011 17:06:50
12345	01-02-2011 09:26:46	01-02-2011 16:55:02

Ik heb gekozen voor de tweede bovenstaande oplossing omdat de eerste oplossing het erg lastig maakt om te bepalen welke twee kaartregistraties bij elkaar horen. Uiteindelijk wil je toch weten hoe lang een werknemer heeft gewerkt, in dat geval blijft oplossing twee het beste want dan wordt het meteen goed in de database gezet.

Dit is het Relationeel Representatiemodel (RRM) van de Workingtimes tabel:

Workingtimes(WotiGuid, WotiPersGuid, WotiCheckintime, WotiCheckintimezoneGuid, WotiCheckouttime, WotiCheckouttimezoneGuid, WotiPrloGuid, WotiScannerNumber, WotiPaPrGuid, WotiBreak, WotiOverruled)

WotiPersGuid is vreemde sleutel, verwijst naar PersGuid in Persons, Null toegestaan
WotiCheckintimezoneGuid is vreemde sleutel, verwijst naar TimeGuid in Timezones, Null toegestaan
WotiCheckouttimezoneGuid is vreemde sleutel, verwijst naar TimeGuid in Timezones, Null toegestaan
WotiPrloGuid is vreemde sleutel, verwijst naar PrloGuid in ProjectLocations, Null toegestaan
WotiPaprGuid is vreemde sleutel, verwijst naar PaprGuid in ParticipatedInProjects, Null toegestaan

Figuur 8 - RRM Workingtimes

```
<relation table="WorkingTimes" key="WotiPersGuid"
target="Persons" action="RActionCascadeDelete" />

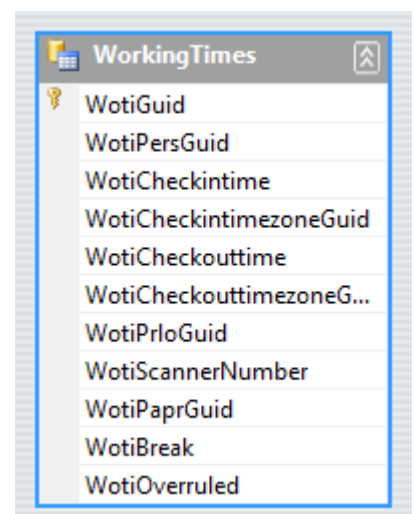
<relation table="WorkingTimes" key="WotiCheckintimezoneGuid"
target="Timezones" action="RActionBlock" />

<relation table="WorkingTimes" key="WotiCheckouttimezoneGuid"
target="Timezones" action="RActionBlock" />

<relation table="WorkingTimes" key="WotiPrloGuid"
target="ProjectLocations" action="RActionBlock" />

<relation table="WorkingTimes" key="WotiPaprGuid"
target="ParticipatedInProjects" action="RActionSetNull" />
```

Figuur 10 - Foreign Keys SkillsXP database



WorkingTimes
WotiGuid
WotiPersGuid
WotiCheckintime
WotiCheckintimezoneGuid
WotiCheckouttime
WotiCheckouttimezoneG...
WotiPrloGuid
WotiScannerNumber
WotiPaprGuid
WotiBreak
WotiOverruled

Figuur 9 - Workingtimes tabel

De foreign keys worden niet op database niveau maar op programmeer-niveau gedefinieerd. In bovenstaande figuur 'Foreign Keys SkillsXP database' laat ik zien hoe de foreign keys van Workingtimes gedefinieerd worden in een XML bestand.

In **WotiPaprGuid** wordt het ID van het planningsrecord opgeslagen die bij de desbetreffende persoon hoort en bij het tijdstip dat de persoon de kaartregistratie doet. Aan dit veld kan je afleiden of de persoon ingepland is of niet, dit kan in de toekomst een manier worden om niet-gewenste werknemers te detecteren.

WotiBreak is een Bit (1 of 0). 1 Betekend dat de rij pauzetijden bevat en 0 betekend dat de rij werktijden bevat.

WotiOverruled is een Bit (1 of 0). 1 Betekend dat de werktijd via SkillsXP is aangepast of toegevoegd en 0 betekend dat de werktijd niet aangepast is. Dit kan in de toekomst gebruikt worden om te bepalen welke werknemers vaak vergeten om in- of uit te checken

WotiCheckintimezoneGuid en **WotiCheckouttimezoneGuid** zijn opgenomen om aan de requirement 'tijdzones' te voldoen. Omdat P & O vaarten vaart die door verschillende tijdzones kunnen gaan heb ik rekening gehouden dat het mogelijk is dat werknemers in een andere tijdzone uitchecken als dat ze zijn ingecheckt. Via een verwijzing naar de tijdzones tabel kan altijd achterhaald worden hoeveel uur iemand daadwerkelijk gewerkt heeft.

WotiPrloGuid en **WotiScannerNumber** zijn opgenomen om te bepalen waar de werknemer in- of uitcheckt.

Problemen tijdens de elaboration fase

Scanner(s)

In bovenstaande paragrafen heb ik uitgelegd hoe ik er toe ben gekomen met twee kaartscanners te gaan werken.

Met twee scanners kwamen er alleen wel weer andere problemen bij; wat als iemand twee keer achter elkaar incheckt? Daar heb ik ook een gebruikersvriendelijke oplossing voor bedacht. Als een persoon een keer vergeet in of uit te checken en daardoor twee keer achter elkaar dezelfde handeling doet wordt dit netjes in beeld vermeldt. In de database wordt een nieuw record aangemaakt en het oude record blijft staan met een leeg veld. Tijdens de feedback naar de werknemer toe was het daardoor makkelijk te realiseren het aantal gemiste checkin/checkuit tijden te tellen. De controle voor gemiste kaartregistraties gebeurt na het opslaan, vandaar:

- Na het inchecken mag maximaal één record zijn zonder uitcheck tijd en geen records zonder incheck tijd.
- Na het uitchecken mogen er geen records zijn met missende in- of uitcheck tijden.

Dit is een voorbeeld van de feedback die een medewerker te zien krijgt:

Check in succesfull - You forgot to check OUT on 12 day(s) - You forgot to check IN on 1 day(s)

Figuur 11 - Vergeten in- en uitcheck tijden

Generiek

Ik moest de uitbreidingen aan de kaartscanner software zo generiek mogelijk maken zodat huidige gebruikers gewoon dezelfde handelingen kunnen doen als voor de kaartscanner update. Dat houdt bijvoorbeeld in dat BIS, de klant waar de oorspronkelijke kaartscanner software voor gemaakt is, gewoon haar eigen certificaten scherm kon behouden en Strukon ook haar eigen versie van het certificaten scherm kon gaan gebruiken. Daarnaast moet P & O met dezelfde software haar werknemers kunnen in- en uitchecken.

Ik heb de uitbreidingen op de kaartscanner software update zo gemaakt dat gebruikers via een settings bestand de software zo kunnen instellen zodat het aan hun wensen voldoet. Zo kunnen de gebruikers kiezen of ze het certificaten controle scherm met of zonder filter opties willen hebben en daarnaast of de checkin/checkuit functionaliteit aan of uit moet staan.

5.3 Construction Fase

In de construction fase legt het accent op het bouwen en programmeren van de uitbreidingen binnen de kaartscanner software. Daarnaast heb ik in deze fase een testplan opgesteld en uitgevoerd om de kwaliteit van de kaartscanner software te waarborgen.

Problemen en werkzaamheden

Filter opties

De filter opties van het certificaten scherm bepalen welke certificaten zichtbaar worden. In SkillsXP kan een certificaat verplicht(mandatory) of aangeraden(recommended) worden gezet voor een functie. De requirement van Strukton zag er zo uit:

- **[Strukton]** Er moeten een aantal filter opties komen die bepalen welke certificaten zichtbaar worden in het certificaten scherm. Namelijk de volgende:
 - 'Alleen hoofdrol certificaten' en/of 'Alleen Bijrol certificaten';
 - 'Alleen rol certificaten' of 'Alle certificaten';
 - 'Mandatory' en/of 'Recommended' en/of 'Not required'.

De functionaliteit die ik nodig had om de juiste certificaten zichtbaar te maken was al gebouwd in de library's van SkillsXP. De filter opties waren daardoor gemakkelijk te implementeren.

Een voorbeeld van het uiteindelijke certificaten scherm met filter opties:

Persoon Identificatie

Koole
I.J.
Dutch
12/29/1987
zuideinde 42
2445AW
aarlanderveen
Car
D1234567

Hoofd functie: Tool Pusher
Overige Bevoegdheden: Forklift Operator, Electrician, Fire F

Certificaten
☒ Rol certificaten
☐ Verplicht (M)
☒ Optioneel (R)
☒ Alle certificaten
☐ Niet Benodigd

Missende Certificaten		Verplicht voor	
BOSIET		Fire Fighting team member	
Fall Protection		Electrician	
Forklift training		Forklift Operator	
Network cabling		Electrician	
Stop Training Supervisor		Tool Pusher	
Well Control Introduction		Tool Pusher	

Verloopdatum	Certificaat
R 1/1/2010	AS3551 Electrical Safety Testing in Medical Locations

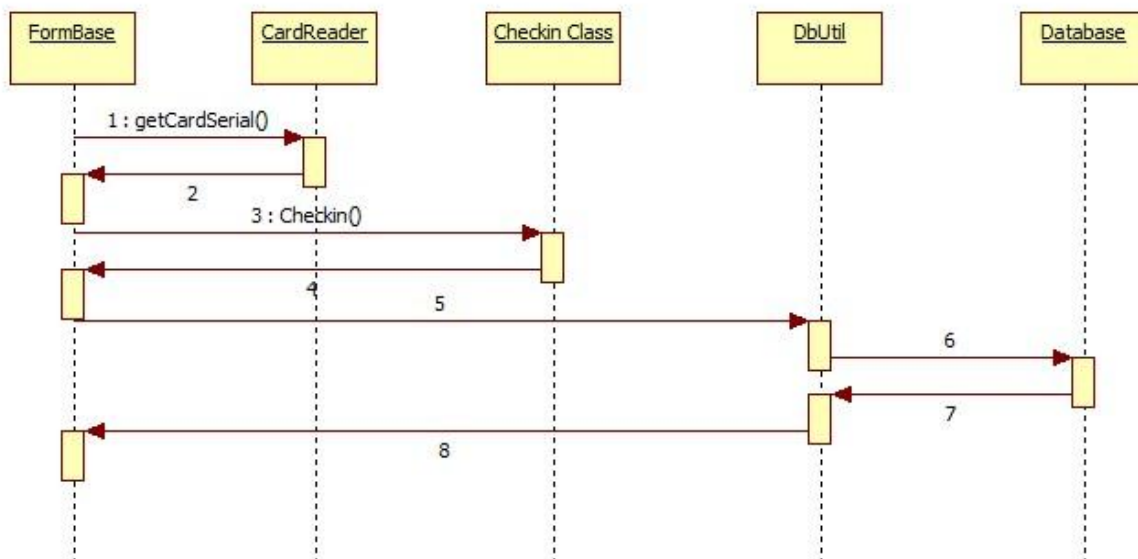
Checkin Time	Timez.	Checkout Time	Timez.	Pr. Lo. Name
05-30-2011 01:00:00 AM	UTC -10:00	05-30-2011 03:00:00 AM	UTC -10:00	
05-28-2011 12:00:00 AM	UTC -07:00	05-28-2011 07:00:00 AM	UTC -07:00	
05-27-2011 05:30:00 PM	UTC -09:00	05-27-2011 07:30:00 PM	UTC -09:00	Houston, United States
05-27-2011 03:00:00 PM	UTC -09:00	05-27-2011 05:00:00 PM	UTC -09:00	
05-20-2011 05:30:00 PM	UTC -09:00	05-20-2011 07:30:00 PM	UTC -09:00	Houston, United States
05-12-2011 12:00:00 AM	UTC -09:30	05-12-2011 06:00:00 AM	UTC -09:30	Cape Town, South Afr...
05-11-2011 11:00:00 AM	UTC +01:00	05-11-2011 11:45:00 PM	UTC +01:00	North Sea

Check in succesfull - You forgot to check OUT on 12 day(s)
Persoon heeft missende certificaten

Figuur 12 - Het certificaten scherm met filter opties en de tabellen: missende certificaten, behaalde certificaten en werktijden

Sequence diagram Inchecken

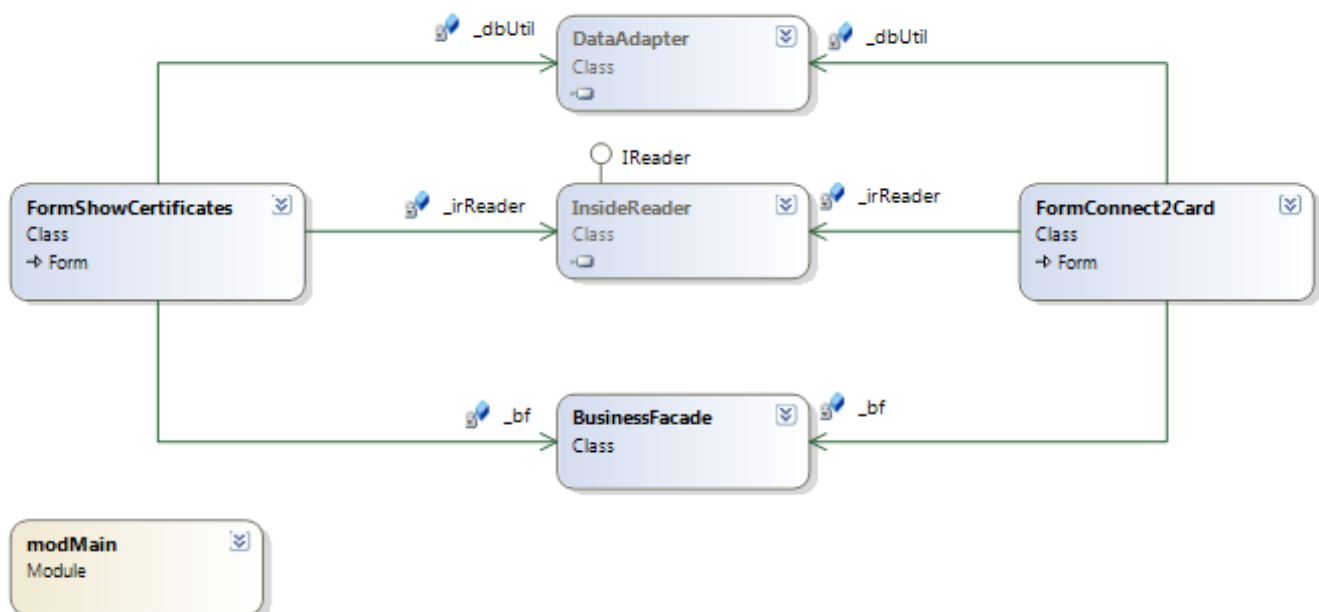
Deze sequence diagram heb ik gemaakt om de volgorde van de aanroeping van verschillende klassen overzichtelijk te maken bij een incheck scenario.



Figuur 13 - Sequence Diagram Inchecken

Oorspronkelijke klassendiagram kaartscanner software

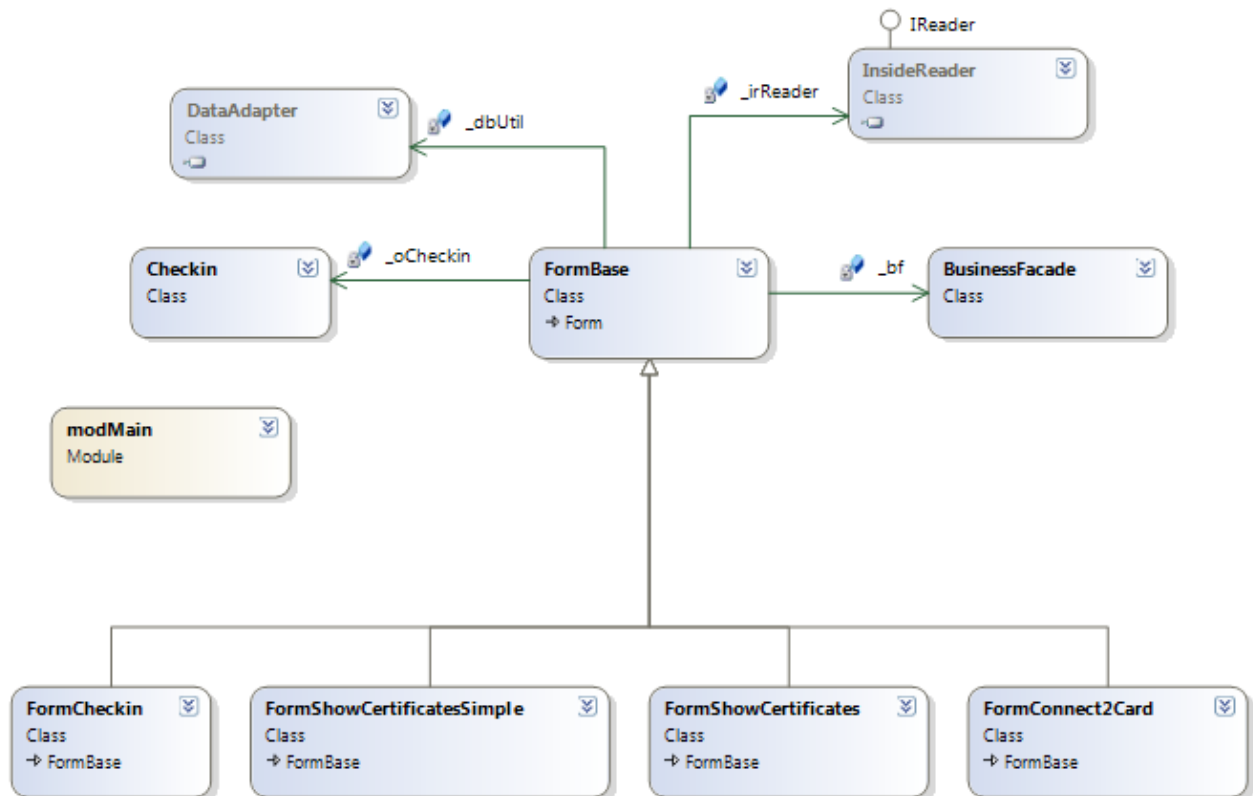
Onderstaand zie je het klassendiagram van de oorspronkelijke kaartscanner software. In de oorspronkelijke situatie was redelijk wat code duplicatie aanwezig, dit heb ik verholpen door een base Class in te bouwen die alle code duplicatie onnodig maakt. Meer uitleg hierover geef ik op de volgende pagina.



Figuur 14 - Klassendiagram Oorspronkelijk kaartscanner software

Klassendiagram Kaartscanner software

Onderstaande klassendiagram laat zien hoe de uiteindelijke kaartscanner software er uit ziet. De Windows formulieren erven alle publieke functies van de base Class (FormBase) over. In deze functies staat de code om met de kaartscanner te praten, data op te vragen van de database, data weg te schrijven naar de database. In de BusinessFacade staat alle code om personen, certificaten en medewerker functies op te vragen uit de database.



Figuur 15 - Klassendiagram Nieuwe Kaartscanner Software

Singleton

Tijdens het inwerken in de kaartscanner software merkte ik op dat het singleton principe niet werd toegepast op de instantie van het kaartscanner object. Dat betekent dat er meerdere instanties aangemaakt kunnen worden van het kaartscanner object wat niet de bedoeling is. Dit zou bijvoorbeeld kunnen gebeuren als een gebruiker de kaartscanner software twee keer opstart of als er meerdere threads met de kaartscanner software kunnen praten. Om mogelijke conflicten te voorkomen heb ik tijdens het uitbreiden van de kaartscanner software het Singleton principe toegepast. Kort samengevat zorgt het Singleton principe dat er altijd maar één instantie van het kaartscanner object aangemaakt kan worden. Het Singleton principe controleert of er al een instantie is aangemaakt. Zo ja geeft het Singleton principe de bestaande instantie terug, zo niet wordt er een nieuwe instantie aangemaakt van de kaartscanner software.

Onderstaand staat een voorbeeld van broncode van een Singleton die double-check locked gebruikt. Als een thread vraagt om een instantie van het object zal er gecontroleerd worden of dit object al een keer eerder is aangemaakt, mocht dit niet het geval zijn zal er een lock plaatsvinden. Een lock houdt alle threads tegen die op dat punt aankomen. Er kan dus maar één thread in het lock gedeelte zitten. Als de eerste thread klaar is met het aanmaken van het object wordt de lock opgeheven. De wachtende threads gaan nu weer verder en komen aan bij de

tweede controle, op dat punt is het object reeds aangemaakt door thread één en mogen ze geen nieuw object aanmaken.

```
Public Class ContactlessSystem
    Private Shared _sysInstance As ContactlessSystem
    Private Shared ReadOnly lock As New Object()

    Public Shared Function GetInstance() As ContactlessSystem

        If _sysInstance Is Nothing Then
            SyncLock (lock)

                If _sysInstance Is Nothing Then
                    _sysInstance = New ContactlessSystem
                End If

            End SyncLock
        End If

        Return _sysInstance

    End Function
End Class
```

Testen

Tijdens en na de construction fase ben ik bezig geweest met het opstellen en uitvoeren van tests. Tijdens het uitvoeren van de tests heb ik rekening gehouden met extremen situaties. Bijvoorbeeld een persoon met 0 werktijden op zijn naam en een persoon met 100.000 werktijden op zijn naam. Alle tests zijn uitgevoerd op twee verschillende computer met verschillende besturingssystemen.

Wordt de incheck tijd opgeslagen?

Titel	Wordt de incheck tijd opgeslagen?	Verwachte waarde/resultaat	Resultaat PC 1	Resultaat PC 2
Samenvatting	Het tijdstip dat de medewerker zijn pas langs de scanner haalt is opgeslagen als incheck tijd.	←	Ja	Ja
Aannamen	De cardreader is actief, de cardreader software is actief, de werknemerspas is gelinkt aan een medewerker	-	-	-
Handelingen	<ol style="list-style-type: none"> 1. De medewerker haalt zijn pas langs een scanner 2. Het tijdstip dat de pas langs de scanner gaat word opgeslagen in de database 3. De tester haalt de tijd op aan de hand van een query met het personeels ID 	<p>Incheck Feedback (Zie Feedback)</p> <p>-</p> <p>De juiste incheck tijd</p>	Ja	Ja

Uitzonderingen	De vorige keer dat de medewerker zijn pas langs een scanner heeft gehaald is de medewerker al ingecheckt. Hij zal dus deze keer uitchecken.	Uitcheck Feedback (Zie Feedback)	Ja	Ja
----------------	---	-------------------------------------	----	----

Query Latest Workingtime

Tijdens het ontwikkelen van het urenregistratiesysteem heb ik een complexe query moeten gebruiken om te bepalen wat de laatste checkin- of checkuittijd is van een medewerker. Deze query wordt gebruikt als de medewerker uitcheckt. Het systeem moet dan bepalen of de persoon de vorige keer netjes in ingecheckt zodat de uitchecktijd er kan worden bijgeschreven, of dat de persoon is vergeten in te checken en dat er een nieuw record aan moet worden gemaakt met alleen een uitchecktijd.

Kortom deze query is complex omdat er drie soorten records in de database kunnen staan:

Checkintijd	Checkuittijd	Project locatie
01-01-2011 13:00:56	01-01-2011 17:31:01	Noordzee
NULL	01-01-2011 17:31:01	Noordzee
01-01-2011 13:00:56	NULL	Noordzee

Afhankelijk van of de checkin- en/of checkuittijd zijn gevuld bepaald mijn urenregistratiesysteem of de uitchecktijd naar een bestaand record of naar een nieuw record wordt toegeschreven.

```

SELECT TOP 1
    WotiGuid, WotiCheckintime, WotiCheckouttime
FROM
    WorkingTimes
WHERE
    WotiPersGuid = '@PersonGuid'
ORDER BY
    CASE WHEN (select MAX(WotiCheckintime) from WorkingTimes WHERE
        WotiPersGuid = '@PersonGuid' ) < (SELECT MAX(WotiCheckouttime) FROM
        WorkingTimes WHERE WotiPersGuid = '@PersonGuid' )
        THEN WotiCheckouttime
    END
DESC,
    CASE WHEN (select MAX(woticheckintime) from WorkingTimes WHERE
        WotiPersGuid = '@PersonGuid' ) > (SELECT MAX(WotiCheckouttime) FROM
        WorkingTimes WHERE WotiPersGuid = '@PersonGuid' )
        THEN WotiCheckintime
    END
DESC

```

Figuur 16 - Query Latest Workingtime

5.4 Transition Fase

In de Transition fase wordt het project overhandigd aan Tritanium en wordt er gekeken hoe het systeem geïmplementeerd kan worden bij een klant.

In deze fase heb ik twee handleidingen geschreven:

- Een uitleg handleiding voor de eindgebruiker, welk knopje wat doet enz.
- Een implementatie handleiding voor de persoon die het urenregistratiesysteem ooit gaat implementeren bij een klant.

In deze fase ben ik bij Strukton langs geweest om de kaartscanner software te implementeren. Omdat bij de eerste poging de software te implementeren de beveiliging te hoog was, was een connectie met onze database server in Leiden niet mogelijk. Na kort overleg met de IT desk van Strukton was dit verholpen. De IT desk had de poorten waar SQL server gebruik van maakt open gezet voor ons IP adres van de server.

Tijdens mijn implementatie werd er erg enthousiast gereageerd op de kaartscanner software bij Strukton. Ik kreeg ook de kans om met een aantal eindgebruikers te praten over de kaartscanner software. Deze eindgebruikers reageerden ook zeer positief. Ik ga ervan uit dat het pilot project met de kaartscanner software positief gaat verlopen en dat Strukton het straks door heel Nederland gaat gebruiken.

Hoofdstuk 6 – Iteratie twee: Planningsysteem

Tijdens de tweede iteratie van mijn afstudeerstage heb ik gewerkt aan een uitbreiding binnen het pakket SkillsXP. Dit hield in dat ik de gehele opdracht van P & O Ferries voor mijn rekening kreeg. Ik heb de requirements van P & O Ferries vastgesteld, vervolgens bepaald welke te ontwikkelen onderdelen er binnen mijn afstudeerstage gingen passen en uiteindelijk die onderdelen daadwerkelijk ontwikkeld.

6.1 *Inception Fase*

In deze fase zijn de requirements vastgesteld, de planning en het gehele plan van aanpak van het planningsysteem gemaakt.

Projectopdracht

De projectopdracht, waar de globale requirements in staan, heb ik al eerder behandeld in Hoofdstuk 4.2. Ik zal ze hier nog een keer kort samenvatten.

Alle planningitems (projecten, personen op projecten en persoonlijke planning) moeten opgeslagen kunnen worden als datum plus tijd. Dit maakt het mogelijk meerdere projecten op een dag te plannen.

Omdat het fysiek in- en uitchecken foutgevoelig is, is een beheergedeelte voor de werktijden zeer gewenst. Onder een beheren wordt verstaan het zelf toevoegen, wijzigen en verwijderen van werktijden.

Er is ook behoefte aan een scherm waarop per ingepland project te zien is wie van de ingeplande werknemers aanwezig is en wie nog niet. Deze informatie zal worden gehaald uit het urenregistratiesysteem.

Er is ook behoefte om meer informatie achter de projectlocaties op te gaan slaan. Er moeten één of meer scenario's en accommodaties aan project locaties gekoppeld kunnen worden. In een scenario moet worden opgeslagen welke accommodaties actief zijn tijdens dat scenario en welke type medewerkers er gepland kunnen worden tijdens het actieve scenario met het bijbehorende minimum en maximum.

Door de wijziging van de planningsitems die naar datum + tijd gaan zullen er een aantal rapporten binnen SkillsXP veranderen. Er zullen ook een aantal rapporten gemaakt moeten worden voor de werktijden van het personeel.

Eisen

De specifieke gedetailleerde eisen van P & O moest ik gaan opmaken uit het klantendossier dat Tritanium van P & O bijhoudt. In het dossier trof ik een Excel sheet aan met alle onderwerpen waar P & O SkillsXP voor wilt gaan gebruiken. Al deze onderwerpen kunnen op één van de volgende manier afgehandeld worden:

- Toevoeging van functionaliteit in SkillsXP;
- Verandering van functionaliteit in SkillsXP;
- Is al reeds aanwezig in SkillsXP.

Door het inwerken in SkillsXP dat ik al reeds gedaan had kon ik een groot gedeelte beantwoorden met 'is al reeds aanwezig in SkillsXP'. Op het onderwerp 'Meerdere projecten op een dag plannen' moest ik 'Verandering van functionaliteit in SkillsXP' antwoorden. De onderwerpen die over bleven werden allemaal 'Toevoeging van functionaliteit in SkillsXP'. Deze onderwerpen heb ik in de volgende alinea uitgewerkt.

Functionele eisen

De functionele eisen zijn opgesteld in een MoSCoW benadering. MoSCoW staat voor Must have, Should have, Could have, Won't have. Door deze indeling ontstaat er een duidelijk beeld van wat er moet gebeuren en wat er mag gebeuren.

Net als bij het urenregistratiesysteem heeft P & O hier ook om hoge gebruikersvriendelijkheid gevraagd. Het beheren van de werktijden was geen eis die P & O gesteld heeft maar een eis die ik zelf heb toegevoegd onder het mom van gebruikersvriendelijkheid. Om deze reden staat de eis 'Manage workingtimes' dan ook onder 'Should', een prioriteit lager als 'Must'.

Must have

- Planning items in hours
 - o Alle planning items binnen het planningsysteem moeten datum plus tijd gaan ondersteunen in plaats van alleen datum.
- Crew list production
 - o Rapport/overzicht waarbij de gebruiker kan aangeven van welke locatie en van welk ingepland project hij/zij de verwachte en daadwerkelijk aanwezige personeel te zien krijgt. Dit hangt samen met de data van het urenregistratiesysteem.
- Workingtimes overview
 - o Er moet een overzicht gegenereerd kunnen worden waar de werktijden per medewerker te zien zijn. Dit gaat met name gebruikt worden voor de loonadministratie.
- Project planning
 - o Bij het plannen van een project moet nu een scenario worden gekozen. Alle medewerkers die vervolgens op een project worden gepland moeten aan de regels van het scenario voldoen.
- Project Location
 - o Een project locatie bestaat uit accommodaties en bevat daarnaast verschillende scenario's. Bij een scenario kunnen de opgegeven accommodaties op actief en inactief worden gezet en daarnaast moet je per scenario een aantal verplichte type medewerkers (rollen) instellen. Bij het instellen van een rol moet je een minimum en een maximum aan geven.
- Accommodations planning
 - o Werknemers kunnen op een (actieve, afhankelijk van het scenario) accommodatie worden gepland.

Should have

- Manage workingtimes
 - o Een overzicht van werktijden per werknemer. De gebruiker kan hierin zelf werktijden toevoegen, bestaande werktijden aanpassen, bestaande werktijden opsplitsen in meerdere werktijden of verwijderen.

Niet-functionele eisen

Functionaliteit

De uitbreidingen in het planningsysteem moeten voor alle huidige en toekomstige klanten bruikbaar zijn. Met nieuwe functionaliteit doet Tritanium altijd haar best om te zorgen dat huidige klanten hun bedrijfsprocessen, waarin SkillsXP gebruikt wordt, niet aan te hoeven passen. Ik zal als voorbeeld de eis 'Planning items in Hours' nemen. Een aantal schermen in SkillsXP laten visueel een kalender zien van de planning items op dag niveau. Voor deze schermen zal een checkbox moeten komen met de functie 'Show items in hours'. Met andere woorden deze uitbreiding moet generiek zijn.

Betrouwbaarheid

De uitbreidingen binnen het planningsysteem dienen getest te worden.

Onderhoudbaarheid

Het systeem moet goed te analyseren zijn, dus er wordt gebruikgemaakt van systeemdokumentatie en extra commentaar waar nodig. De applicatie moet ook goed aan te passen zijn omdat er eventuele uitbreidingen bij kunnen komen.

Usecase Model

In dit usecase model staan alle handelingen die een SkillsXP gebruiker straks moet kunnen doen na de uitbreidingen.



6.2 Elaboration Fase

Tijdens deze fase is gebleken dat het Roster Management System (RMS) dat zeer recent voor de klant BHP Billiton door Tritanium ontwikkeld is, veel overlap heeft met de eisen van P & O. RMS is tijdens mijn afstudeerperiode ontwikkeld en tijdens het schrijven van dit document draait de eerste SkillsXP Bèta versie met de nieuwe RMS functionaliteit bij BHP. Zo gauw het Bèta label komt te vervallen zal er een update worden uitgerold met de nieuwe functionaliteit naar alle andere SkillsXP gebruikers.

In de Elaboration fase heb ik onderzocht wat die overlappen met de eisen van P & O Ferries precies zijn, welke functionaliteit zal moeten veranderen en welke functionaliteit nog gebouwd zal moeten worden om aan de eisen van P & O te voldoen.

Omdat de kans zeer groot is dat er functionaliteit zal moeten gaan veranderen om aan de voorwaarden van P & O te voldoen, zal ik een 'Request of Change' gaan schrijven. De functie van deze request is dat collega's overzichtelijk kunnen zien welke onderdelen er binnen SkillsXP zullen moeten veranderen om aan de eisen van P & O te voldoen. Voordat ik de Request of Change heb kunnen schrijven heb ik eerst de overeenkomsten tussen de requirements van het RMS en de requirements van P & O onderzocht.

RMS

Ik had de beschikking om met twee collega's te praten/interviewen die nauw zijn betrokken bij de klant BHP en het RMS project. Eén van deze collega's was de directeur van Tritanium. Omdat dhr. Gittenberger het vaak druk heeft en maar weinig tijd heeft voor extra bezigheden had ik mijn gesprek met hem goed voorbereidt. Ik had goede vragen opgesteld zodat mij in korte tijd een hoop duidelijk werd van de RMS requirements. Door de gesprekken met mijn collega's heb ik veel geleerd over het RMS en de gelijkenissen met P & O.

RMS staat voor een nieuwe functionaliteit in het planningsgedeelte van SkillsXP. RMS maakt het mogelijk om scenario's, accommodaties, lifeboats en muster points (verzamelpunten) te definiëren en koppelen aan project locaties. Een scenario houdt in dat accommodaties actief of inactief kunnen worden gezet. Accommodaties kunnen gekoppeld worden aan lifeboats en muster points en er kan worden ingesteld hoeveel van elk type werknemer ingepland mag worden. Lifeboat en musterpoint moeten worden gekoppeld aan accommodations voor het geval van nood. Een accommodatie wordt dus niet alleen als verblijf aan een werknemer toegewezen, het is ook voor de veiligheid.

RMS heeft erg veel overlap met de eisen van P & O. Ik ben mij gaan verdiepen in het Requirements document van RMS en ben mij gaan voorbereiden op een vergadering met de hoofd programmeur en de project manager om te discussiëren over hoe de eisen van P & O verwerkt zullen gaan worden. Deze vergadering had voor mij als hoofddoel hun visie te horen over de requirements van P & O en een tijdsschatting te krijgen per eis. Ik heb de vergadering daarom ook opgedeeld per opgestelde eis van P & O die ik in de Inception fase heb vastgesteld.

Dit is de samenvatting van dat gesprek:

- Een soortgelijk systeem als werktijden van werknemers en een beheer module daarvan komen niet voor in RMS, deze moeten nog gebouwd worden.
- Alle planningsitems worden nog steeds met alleen start- en einddatum gepland. Deze eis van P & O is eigenlijk een stuk groter en complexer geworden omdat veel functionaliteit in RMS werkt met hele dagen. Bijvoorbeeld het inplannen van werknemers op accommodaties werkt op hele dagen.
- Alle functionaliteit achter de project locaties is grotendeels al aanwezig en komt overeen met de eisen van P & O. Echter is in RMS de relatie tussen projecten en scenario's anders gedefinieerd dan P & O het zou willen. De reden hiervan is dat BHP Billiton een boormaatschappij dat werkt met olieboorplatform en P & O een vaartmaatschappij. Op een olieboorplatform lopen een aantal projecten naast elkaar die soms weken duren, BHP wilde een overkoepelend scenario kunnen definiëren zoals 'onderhoud' met een start- en einddatum. Alle projecten die binnen het scenario 'onderhoud' vallen moeten aan andere voorwaarden voldoen. P & O wil per vaart een scenario kunnen kiezen zoals 'Full Capacity Passengers' of 'Freight Only'. Afhankelijk van het gekozen scenario zijn accommodations beschikbaar en mogen er maar een bepaald aantal werknemers gepland worden.
- Mijn werkzaamheden vanaf nu zullen zijn:
 - o **(Onveranderd)** Een overzicht van werktijden per werknemer. De gebruiker kan hierin zelf werktijden toevoegen, bestaande werktijden aanpassen, bestaande werktijden opsplitsen in meerdere werktijden of verwijderen.
 - o **(Onveranderd)** Rapport/overzicht waarbij de gebruiker kan aangeven van welke locatie en van welk ingepland project hij/zij de verwachte en daadwerkelijk aanwezig personeel te zien krijgt. Dit hangt samen met de data van het urenregistratiesysteem.
 - o **(Onveranderd)** Er moet een overzicht gegenereerd kunnen worden waar de werktijden per medewerker te zien zijn. Dit gaat met name gebruikt worden voor de loonadministratie.
 - o Een Request of Change indienen voor alle veranderingen die plaats moeten vinden voor de eisen van P & O.

De eis 'Alle planningsitems ombouwen van datum naar datum + tijd heb ik moeten overdragen aan mijn collega's. Beide de hoofd programmeur en de project manager zeiden dat het teveel werk was en niet ging passen binnen mijn afstudeerperiode.

Request of Change

Deze request of Change is geschreven voor mijn collega's die na of tijdens mijn afstudeerperiode functionaliteit moeten aanpassen/toevoegen in SkillsXP. Deze request bevat twee grote onderwerpen waarvan ik de databaseveranderingen van het eerste onderwerp heb opgenomen in dit verslag. De twee grote onderwerpen zijn:

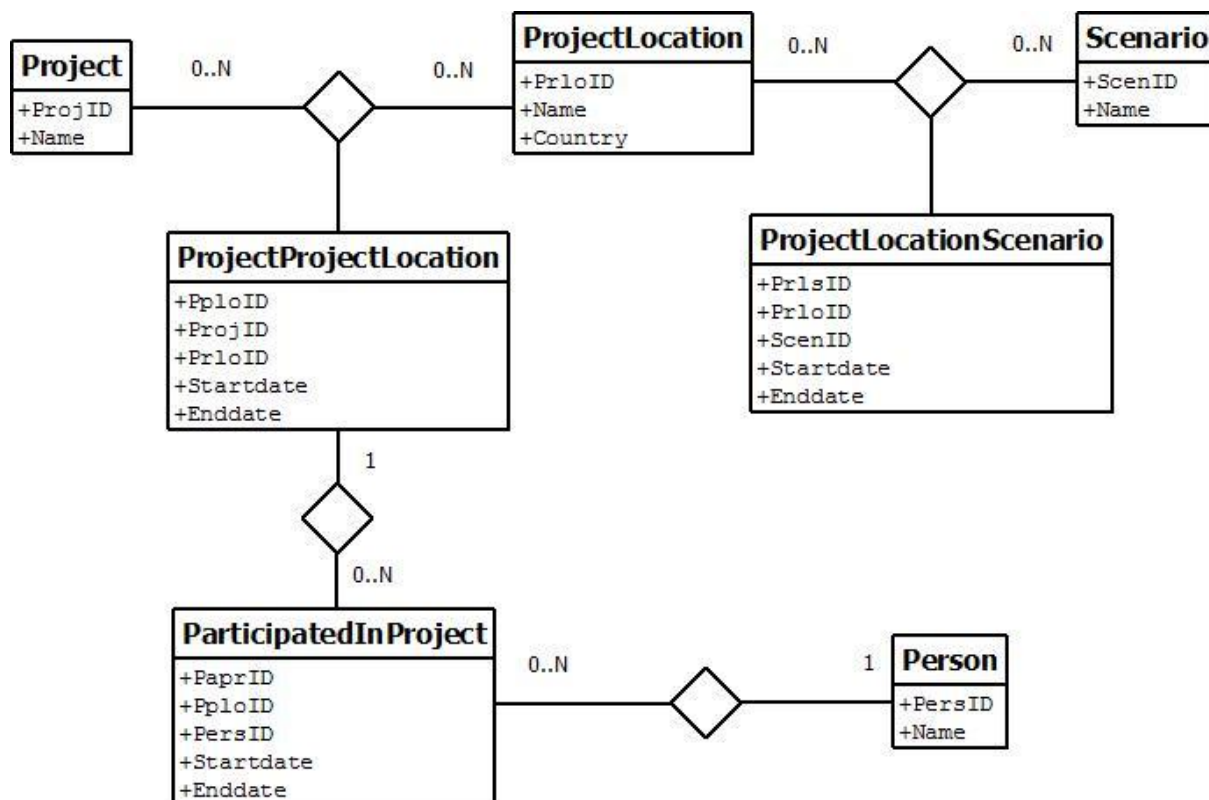
- De relatie tussen projecten en scenario's.
- Planningsitems van datum naar datum + tijd

Relatie tussen projecten en scenario's

Scenario's zijn gekoppeld aan een project locaties. In de tabel ProjectLocationScenario (volgende pagina) wordt opgeslagen op welke projectlocatie wanneer welk scenario actief is. De data uit de tabel ProjectLocationScenario wordt geraadpleegd wanneer een nieuw record wordt aangemaakt in ParticipatedInProject. Tijdens het aanmaken wordt gecontroleerd of het maximaal aantal

ingeplande werknemers wordt overschreden. Dit maximale aantal werknemers wordt opgeslagen in de tabel ScenarioRoles. Deze tabel is niet afgebeeld in dit model om het geheel overzichtelijk te behouden. Dit geldt ook voor Accommodation, Lifeboat Musterpoint en Role. Met het onderstaande ontwerp kan je op de volgende wijze scenario's definiëren.

- Scenario A: 1 Jan tot 31 Jan, 18 van de 20 accommodations actief, minimaal 10 maximaal 18 matrozen.
- Scenario B: 1 Feb tot 30 Apr, 6 van de 20 accommodaties actief, minimaal 4, maximaal 10 matrozen.

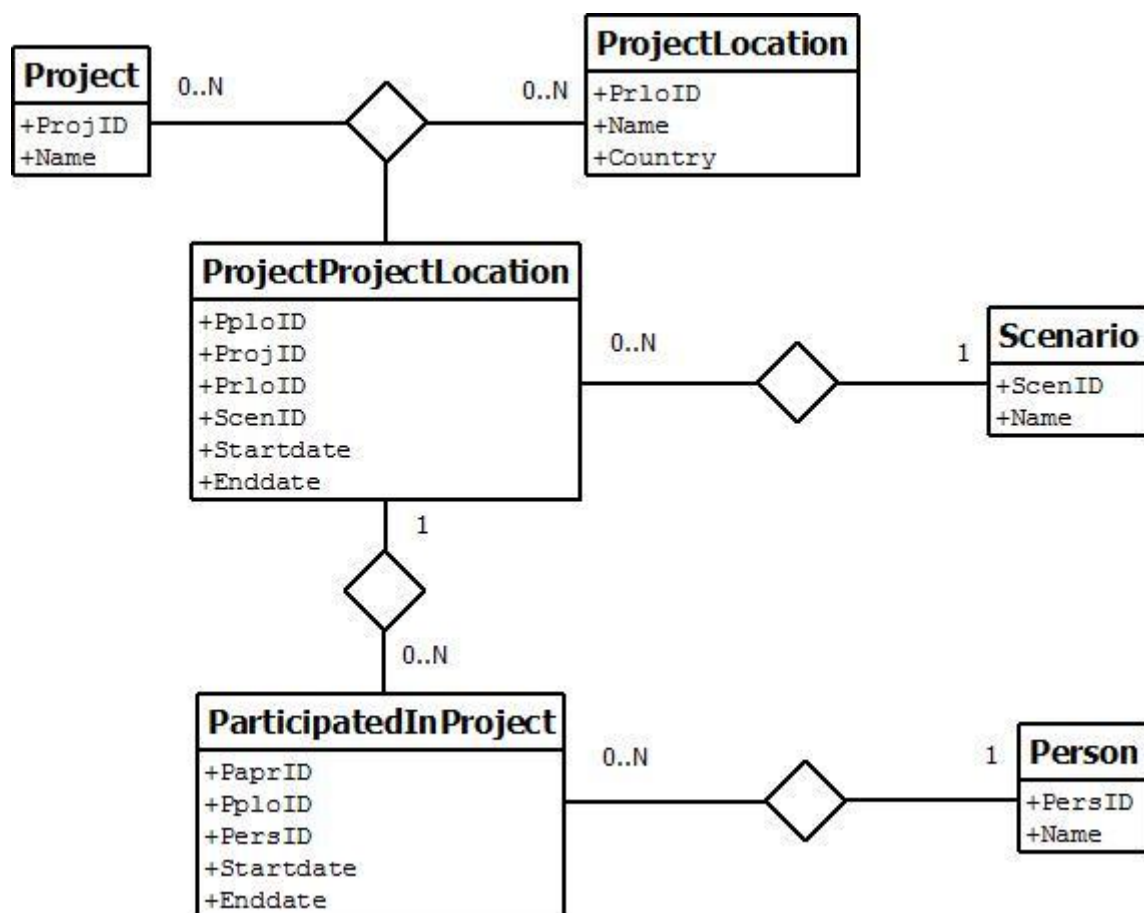


Figuur 17 - Database model Relatie Projecten - Scenario's RMS

P & O wil echter per project kunnen instellen wat voor scenario actief moet zijn. Enkele voorbeelden :

- Scenario 1: Full passenger capacity, alle accommodaties actief, minimaal 10 maximaal 20 matrozen aan boord enz.
- Scenario 2: Reduced passenger capacity, 60% van de accommodaties actief, minimaal 6 maximaal 12 matrozen aan boord enz.
- Scenario 3: Freight only, 5% van de accommodaties actief, minimaal 6 maximaal 12 matrozen aan boord enz.

Het database model zal in het onderstaande model moeten veranderen zodat P & O haar projecten op haar gewenste manier kan opslaan in de database.



Figuur 18 - Database model Projecten - Scenario's P & O

6.3 Construction Fase

In deze fase ligt het accent op het bouwen en programmeren van de nieuwe functionaliteit in SkillsXP. Het ontwikkelen gebeurde in de omgeving Microsoft Visual Studio 2010. Daarnaast werd er gebruik gemaakt van

Het bouwen van de nieuwe functionaliteit in SkillsXP ging erg voorspoedig. Als ik vragen had kon ik altijd bij ervaren collega's terecht die mij snel een antwoord konden geven.

Voor het klassendiagram die ik heb gemaakt over hoe de schermen in SkillsXP worden opgebouwd heb ik helaas geen toestemming gekregen om deze buiten het kantoor te tonen.

Voor en tijdens het ontwikkelen in SkillsXP heb ik erg weinig ontwerpkeuzes moeten maken. Uiteindelijk heb ik een WorkingTimes view aanmaken. In de view wordt gedefinieerd van welke tabellen je welke velden selecteert, welke relaties er gelegd moeten worden tussen tabellen en hoe de geselecteerde velden moeten heten. Daarnaast moest ik een zogenaamde system class van WorkingTimes aanmaken. In deze system class kwam alle logica te staan voor het selecteren, veranderen, toevoegen en verwijderen van data.

```
Dim ds As New Tritanium.DatabaseUtils.Clauses.DataSet

ds.AddCOLUMN("Persons", "PersEmployeeNum")
ds.AddCOLUMN("Persons", "PersFirstName")
ds.AddCOLUMN("Persons", "PersLastName")
ds.AddCOLUMN("WorkingTimes", "WotiCheckintime")
ds.AddCOLUMN("WorkingTimes", "WotiCheckouttime")
ds.AddCOLUMN("ProjectLocations", "PrloName")

ds.AddRELATION(msTableName, "WotiPrloGuid", "ProjectLocations", "PrloGuid",
DatabaseUtils Enumeration.RelationType.All_From_Main_Table)

ds.AddRELATION(msTableName, "WotiPersGuid", "Persons", "PersGuid",
DatabaseUtils Enumeration.RelationType.All_From_Main_Table)
```

Figuur 19 - Een stukje van de View van WorkingTimes uit SkillsXP

Een paar screenshots laten zien van de schermen die ik uiteindelijk heb gemaakt:

Employee number	First name	Last name	Checkin Time	Checkout Time	Pr. Lo. Name	Scanner Number	Break	Overruled
22222	Maries	Koole	11-May-11 00:00:00	11-May-11 01:00:00			<input type="checkbox"/>	<input checked="" type="checkbox"/>
22222	Maries	Koole	12-May-11 11:07:00	12-May-11 18:00:00	North Sea	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
22222	Maries	Koole	12-May-11 09:27:24	12-May-11 09:35:00	North Sea	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
22222	Maries	Koole	04-May-11 07:00:00	05-May-11 11:00:00	North Sea	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
22222	Maries	Koole	12-May-11 00:00:00	12-May-11 06:00:00	Cape Town, South Africa		<input type="checkbox"/>	<input checked="" type="checkbox"/>
22222	Maries	Koole	10-May-11 06:40:00	10-May-11 15:00:00	Rotterdam, The Netherlands		<input type="checkbox"/>	<input checked="" type="checkbox"/>
22222	Maries	Koole	12-May-11 09:52:00	12-May-11 09:54:00	North Sea	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
22222	Maries	Koole	11-May-11 11:00:00	11-May-11 23:45:00	North Sea		<input type="checkbox"/>	<input checked="" type="checkbox"/>
22222	Maries	Koole	12-May-11 09:51:29	12-May-11 09:51:00	North Sea	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Figuur 20 - Rapport Personeel Working Times

Checkin Time	Checkout Time	Pr. Lo. Name	Scanner Number	Break	Overruled
5/12/2011 9:38:05 AM		North Sea	1	<input type="checkbox"/>	<input type="checkbox"/>
5/16/2011 10:42:06 AM		North Sea	1	<input type="checkbox"/>	<input type="checkbox"/>
5/12/2011 9:55:23 AM		North Sea	1	<input type="checkbox"/>	<input type="checkbox"/>
5/16/2011 10:34:54 AM		North Sea	1	<input type="checkbox"/>	<input type="checkbox"/>
5/12/2011 11:07:00 AM	5/12/2011 6:00:00 PM	North Sea	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5/12/2011 9:27:24 AM	5/12/2011 9:35:00 AM	North Sea	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5/12/2011 12:00:00 AM	5/12/2011 6:00:00 AM	Cape Town, South Africa		<input type="checkbox"/>	<input checked="" type="checkbox"/>
5/12/2011 9:55:25 AM		North Sea	1	<input type="checkbox"/>	<input type="checkbox"/>
5/16/2011 10:44:45 AM		North Sea	1	<input type="checkbox"/>	<input type="checkbox"/>
5/12/2011 9:52:00 AM	5/12/2011 9:54:00 AM	North Sea	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5/16/2011 10:55:46 AM		North Sea	1	<input type="checkbox"/>	<input type="checkbox"/>
	5/20/2011 1:00:00 AM	India		<input type="checkbox"/>	<input checked="" type="checkbox"/>
5/16/2011 10:59:13 AM		North Sea	1	<input type="checkbox"/>	<input type="checkbox"/>
5/16/2011 10:54:44 AM		North Sea	1	<input type="checkbox"/>	<input type="checkbox"/>
5/12/2011 10:07:11 AM		North Sea	1	<input type="checkbox"/>	<input type="checkbox"/>
5/12/2011 9:55:24 AM		North Sea	1	<input type="checkbox"/>	<input type="checkbox"/>
5/12/2011 9:51:29 AM	5/12/2011 9:51:00 AM	North Sea	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Figuur 21 - Screenshot van het Working Times overzicht. Elke regel heeft een Edit, Copy en Delete button

Testen

Tijdens en na de construction fase ben ik bezig geweest met het opstellen en uitvoeren van tests. Tijdens het uitvoeren van de tests heb ik rekening gehouden met extremen situaties. Bijvoorbeeld een persoon met 0 werktijden op zijn naam en een persoon met 100.000 werktijden op zijn naam. Alle tests zijn uitgevoerd op twee verschillende computer met verschillende besturingsystemen. Als voorbeeld zal ik één testcase opnemen in dit verslag.

Pas bij 'Maries Koole' een werktijd aan en controleer of de wijzigingen opgeslagen worden.

Titel	Pas bij 'Maries Koole' een werktijd aan en controleer of de wijzigingen opgeslagen worden.	Verwachte waarde/resultaat	Resultaat PC 1	Resultaat PC 2
Samenvatting	Controleer of de aangepaste werktijd in de database word opgeslagen bij 'Maries Koole'	←	Ja	Ja
Aannamen	SkillsXP is opgestart	-	-	-
Handelingen	<ol style="list-style-type: none"> 1. Klik bovenin op Persons > Details > Navigeer naar Maries Koole en dubbelklik. Navigeer naar WorkingTimes en dubbelklik op een werktijd 2. Verander een aantal willekeurige data. 3. Controleer of de wijzigingen zijn opgeslagen in de DB 	<p>-</p> <p>-</p> <p>De zojuist ingevoerde wijzigingen staan in de DB opgeslagen</p>	<p>-</p> <p>-</p> <p>Ja</p>	<p>-</p> <p>-</p> <p>Ja</p>
Uitzonderingen	De ingevoerde werktijden conflicteren met andere werktijden.	Foutmelding	Ja	Ja

6.4 Transition Fase

In de Transition fase heb ik het planningsysteem project overgedragen aan mijn collega's bij Tritanium.

De transition fase van het planningsysteem is beperkt doorlopen omdat niet alle eisen van P & O tijdens mijn afstudeerstage zijn waargemaakt. Een implementatie bij P & O was dus helaas niet mogelijk. In De nieuwe of veranderende functionaliteit die ik moest overdragen aan collega's vanwege tijdgebrek konden helaas niet meteen door hun gemaakt worden vanwege andere prioriteiten. SkillsXP kan dus nog niet geïmplementeerd worden bij P & O. Wel heb ik een korte handleiding geschreven over het beheren van de werktijden die tijdens de toekomstige implementatie handig zal zijn.

Hoofdstuk 7 – Evaluatie

In dit hoofdstuk kijk ik terug op de gehele afstudeerperiode. Ik zal een evaluatie doen op het proces, de opgeleverde producten en als laatste de beroepstaken.

7.1 *Proces*

Tijdens de evaluatie op het gehele proces zal ik vooral kijken of ik naar mijn mening goed heb gefungeerd in het bedrijf The Tritanium Company B.V., wat achteraf gezien beter had gekund en wat goed ging naar mijn mening.

Gehele afstudeerstage proces

De afstudeerstage heb ik als erg leuk en erg leerzaam ervaren. Naast het ontwikkelen van software heb ik ook veel met klanten kunnen werken. Dit was vooral met de klant Strukon Rail waar ik ben een paar keer op bezoek ben geweest. Ik heb tijdens de bezoeken een medewerker kunnen interviewen voor de requirements, mijn kaartscanner software geïnstalleerd voor het pilot project. Daarnaast ben ik ook een keer naar Strukon geweest om samen met een college een training te geven aan een aantal medewerkers van Strukon. Tijdens deze training leerden wij de werknemers te leren werken met SkillsXP. Dit heb ik als erg leuk ervaren. Zelfs zo leuk dat ik aan het overwegen ben om daar in de toekomst iets mee te willen gaan doen.

De planning die ik van tevoren moest opgeven voor het afstudeerplan op school bleek verre van correct. Maar een correcte planning maken is en blijft altijd lastig, zelfs voor doorgewinterde projectleiders. Ik moest veel improviseren en inzet tonen om mijn opdracht tot een goed einde te brengen. Een voorbeeld hiervan is dat naast mijn afstudeeropdracht het RMS werd ontwikkeld wat veel overlap met de opdracht van P & O bleek te hebben. Daardoor verviel er een hoop programmeerwerk en moest ik een request of change schrijven om te bepalen welke functionaliteit er zal moeten veranderen zodat SkillsXP aan de eisen van P & O voldeed. Hier kom ik later nog op terug bij de product evaluatie.

Medewerker Tritanium

Persoonlijk vond ik de sfeer bij Tritanium erg positief en voelde mij er daardoor erg snel thuis en op m'n gemak. Omdat Tritanium wat kleiner is ontstaat er automatisch een minder formele sfeer wat zorgt dat de grens tussen bijvoorbeeld de directeur en het personeel wegvallen. Dit is naar mijn mening beter voor de werksfeer. Het nadeel is bij een klein bedrijf dat de grenzen tussen verschillende functies nogal snel kunnen vervagen. Het gebeurt soms wel eens dat ontwikkelaars bijvoorbeeld als supportdeskmedewerker moeten spelen als klanten met problemen komen. Dit omdat er geen vaste supportdeskmedewerkers zijn.

Twee UP fases

Al vrij snel tijdens de afstudeerstage werd duidelijk dat het project in twee delen en zelfs ook in twee verschillende UP ontwikkeltrajecten op te splitsen was. In eerste instantie was dit een vraag van Tritanium aan mij zodat het urenregistratiesysteem sneller klaar zou zijn wat weer nodig was voor het pilot project van Strukon (Hoofdstuk 8 – Overige werkzaamheden). Achteraf vond ik het persoonlijk beter dat alle UP fases twee keer doorlopen zijn. De onderwerpen van de twee onderdelen waren dusdanig verschillend dat het wel erg 'breed' zou worden in één UP iteratie.

Transition fase Urenregistratiesysteem

Tijdens schoolprojecten heb ik nooit een transition fase hoeven doorlopen. Ik heb wel gemerkt dat je tijdens een transition fase goed met de klant moet communiceren over de uiteindelijke situatie waarin de software komt te draaien. Daarnaast moeten de eindgebruikers leren om met jou software om te gaan. Het beste wat je kan doen, vind ik, is jezelf verplaatsen in de

eindgebruiker en dan bedenken 'wat wil ik dat de ontwikkelaar mij allemaal verteld over de nieuwe software'. Pas dan kan je nuttige handleidingen schrijven.

De transition fase van de uitbreiding van het planningsysteem heb ik niet meegemaakt omdat de bouw van de overige functionaliteit die niet in mijn afstudeerperiode pasten moest overdragen naar collega's. Helaas hadden mijn collega's de opdracht om RMS verder uit te breiden zodat de functionaliteit van P & O werd uitgesteld.

7.2 Product

Urenregistratiesysteem

Het woord urenregistratiesysteem dekt niet alle werkzaamheden af in mijn mening. Achteraf had ik de iteratie liever 'Uitbreidingen aan de kaartscanner software' genoemd. Dit vanwege het pilot project van Strukton en de eisen die Strukton had aan de kaartscanner software.

De kaartscanner software implementeren bij Strukton heb ik als erg leuk en leerzaam ervaren. Ik ben tot inzicht gekomen dat testen en het vaststellen van de requirements een zeer belangrijk onderdeel van het gehele proces zijn.

Uitbreiding op het planningsysteem

De overeenkomsten tussen P & O's requirements en RMS haar requirements achterhalen was lastiger dan ik had verwacht. De gedetailleerde requirements van het RMS stonden in een document van 80 paginas; wat het lastig maakte om de requirements helder op een rijtje te krijgen. Daarom was ik erg blij dat het mogelijk was om met collega's hierover te praten. Deze gesprekken had ik goed voorbereid zodat ik in korte tijd de juiste informatie kreeg die ik wilde hebben.

Hoewel de naam 'planningsysteem' niet helemaal toepasselijk meer is heb ik de naam onveranderd gelaten om verwarring te voorkomen. Oorspronkelijk had de naam correct geweest maar door het overlappende RMS waren de eisen van P & O al grotendeels waargemaakt. Nadat ik erachter kwam dat RMS een groot gedeelte van de eisen van P & O al waar had gemaakt was het mijn taak de overeenkomsten en verschillen vast te stellen en die te rapporten in een 'request of change' document. De request of change bestond uit twee grote onderdelen:

- Alle planningsitems van datum naar datum plus tijd
- De relatie tussen projecten en scenario's

Vooraf bij 'Alle planningsitems van datum naar datum plus tijd' moest ik erg veel ontwerpen en logisch nadenken. Dit omdat een hoop schermen en functies in SkillsXP zijn gebaseerd op hele dagen.

7.3 Beroepstaken

De volgende beroepstaken heb ik gekozen om over beoordeeld te worden tijdens mijn afstudeerperiode.

Analyseren en adviseren	Niveau
1.4 uitvoeren analyse door definitie van requirements	3
Databaseontwerp en -ontwikkeling	
2.1 ontwerpen en opstellen gegevensmodel voor database	3
Softwareontwerp en -ontwikkeling	
3.1 Ontwerpen softwarearchitectuur	3
3.2 ontwerpen systeemdeel	3
3.3 bouwen applicatie	4

Uitvoeren analyse door definitie van requirements

Deze beroepstaak heb ik veel moeten gebruiken. Allereerst heb ik alle requirements voor het urenregistratiesysteem vast moeten stellen. Vervolgens bij de tweede ontwikkeliteratie heb ik weer de requirements moeten vaststellen. Toen bleek echter dat naast mijn afstudeerstage het RMS werd ontwikkeld. Omdat het RMS veel overlap had met de eisen van P & O heb ik per eerder vastgestelde requirement moeten analyseren of deze requirement al (deels) was ingebouwd of niet. Alle wijzigingen voor SkillsXP heb ik gerapporteerd in een Request of Change.

Ontwerpen en opstellen gegevensmodel voor database

Voor deze beroepstaak moest ik mijn gewoontes aanpassen aan de eisen van Tritanium. Ik ben onder andere gewend om met SQL server 2008 te werken, dus automatisch gebruikte ik één van de nieuwste datatypes Datetimeoffset. Dit datatype is nieuw in SQL Server 2008 en is een uitbreiding op de huidige Datetime. Met Datetimeoffset kan naast de datum en tijd, ook de tijdzone opgeslagen worden.

Later begreep ik van collega's dat lang niet alle klanten met SQL server 2008 werken dus moest ik uiteindelijk een andere oplossing bedenken voor het opslaan van de tijdzone. Mijn oplossing werd uiteindelijk een gewoon datetime veld en een verwijzing naar de Timezones tabel. In de Timezones tabel worden de tijdzones als integers opgeslagen.

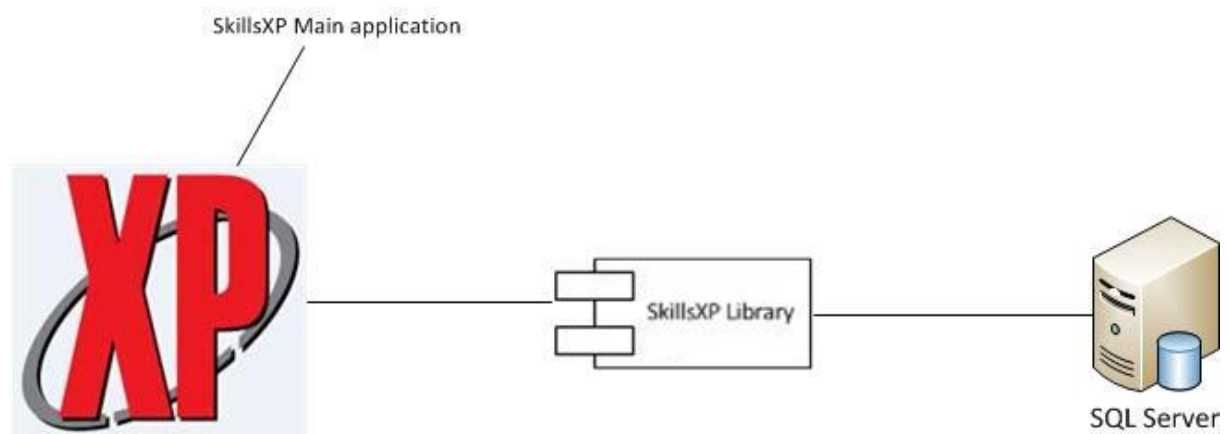
Tijdens de inception fase van het urenregistratiesysteem heb ik een onderzoek uitgevoerd naar de dataopslaglocatie van het urenregistratiesysteem. De conclusie was uiteindelijk dat zelfs een tabel met 625.000 records geen gevaar is voor de performance van een database. Ik had wel een index ingesteld op het veld 'WotiPersGuid' (PersonID) omdat dit veld veruit het meeste opgevraagd gaat worden.

Ontwerpen systeemarchitectuur

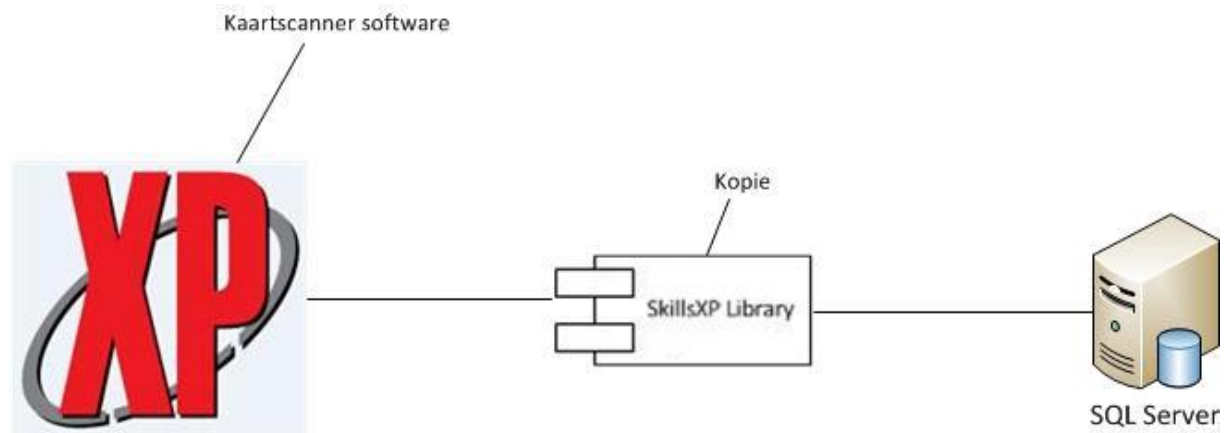
Tijdens de ontwikkeling van het urenregistratiesysteem moest ik ook een aantal architectuele keuzes maken. Omdat mijn urenregistratiesysteem complexe gegevensmodellen zoals persoon data en certificaten data moest ophalen uit de database heb ik gekozen om de SkillsXP library te kopiëren naar mijn urenregistratiesysteem. In deze library's staat de functionaliteit om de respectievelijke data op te vragen van de database. Omdat ik de SkillsXP library's heb gekopieerd hoeft er geen installatie van SkillsXP aanwezig te zijn om mijn kaartscanner software op een systeem te laten werken. Als ik de library's niet had gekopieerd had ik mijn eigen functie moeten schrijven om de juiste data uit de database te halen, als de database dan ooit zou veranderen zouden die veranderingen op twee plaatsen doorgevoerd moeten worden. Dit was natuurlijk geen optie.

Uiteindelijk heb ik dus de library's gekopieerd naar de kaartscanner software en een versie controle ingebouwd. Deze controle controleert of de database versie consistent is voor de versie van de kaartscanner software. Mocht de database versie niet meer overeen komen met de verwachte database versie wordt er een melding gegenereerd dat de kaartscanner software eerste een upgrade moet krijgen worden voordat deze zijn werk weer kan gaan doen.

Hieronder een visualisatie van de architectuur van SkillsXP en de kaartscanner software.



Figuur 22 - SkillsXP architectuur



Figuur 23 - Kaartscanner software architectuur

Ontwerpen systeemdeel

Urenregistratiesysteem

Tijdens het ontwerpen van het urenregistratiesysteem heb ik ervoor gekozen om een nieuwe Base Class toe te voegen voor alle Windows formulieren genaamd FormBase. In deze klasse staat alle functionaliteit die de forms nodig hebben. Daarnaast is FormBase de enige klasse die met de database mag communiceren. Deze oplossing scheelt een hoop code duplicatie.

Daarnaast heb ik nog een klasse toegevoegd genaamd Checkin. In de klasse Checkin staan de functies die het in en uit checken mogelijk maken. De klasse is ook verantwoordelijk voor de feedback die de gebruiker te zien krijgt.

Planningsysteem

Tijdens de bouw van de uitbreidingen in SkillsXP had ik geen ontwerpkeuzes die ik zelf moest maken. De gehele structuur van SkillsXP stond al vast. Het enige wat ik hoefde te doen was een view aanmaken. In de view wordt gedefinieerd van welke tabellen je welke velden selecteert, welke relaties er gelegd moeten worden tussen tabellen en hoe de geselecteerde velden moeten heten. Daarnaast moest ik een zogenaamde system class van WorkingTimes aanmaken. In deze system class kwam alle logica te staan voor het selecteren, veranderen, toevoegen en verwijderen van data.

Bouwen applicatie

Deze beroepstaak heb ik vervuld tijdens het ontwikkelen van de uitbreidingen van de kaartscanner software. Bij de kaartscanner software mocht ik zelf het gehele ontwerp bepalen en welke design patterns ik wilde toepassen. Ik heb uiteindelijk gekozen een Base klasse voor alle Windows formulieren toe te voegen die alle basis functionaliteit bevat. In de kaartscanner software heb ik het Singleton principe toegepast zodat er altijd maar één instantie van de kaartscanner aangemaakt kan worden. Als er meerdere instanties aangemaakt zouden kunnen worden zou dit conflicten kunnen opleveren in de database. Mijn singleton principe maakt gebruik van double-checked locking. Dit komt er in het kort op neer dat de toepassing thread-safe is, zelfs met meerdere threads kan er geen tweede instantie van de kaartscanner aan worden gemaakt.

Afbeeldinglijst

Figuur 1 - Organogram Tritanium – Ik maak deel uit van de ‘Development’ afdeling	4
Figuur 2 - Origineel certificaten scherm	12
Figuur 3 - Kaartscanner met pasjes	12
Figuur 4 - Usecase model	16
Figuur 5 - Usecase beschrijving Inchecken	16
Figuur 6 - SkillsXP globale architectuur	17
Figuur 7 - Kaartscanner software globale architectuur	17
Figuur 8 - RRM Workingtimes	19
Figuur 9 - Workingtimes tabel	19
Figuur 10 - Foreign Keys SkillsXP database	19
Figuur 11 - Vergeten in- en uitcheck tijden	20
Figuur 12 - Het certificaten scherm met filter opties en de tabellen: missende certificaten, behaalde certificaten en werktijden	21
Figuur 13 - Sequence Diagram Inchecken	22
Figuur 14 - Klassendiagram Oorspronkelijk kaartscanner software	22
Figuur 15 - Klassendiagram Nieuwe Kaartscanner Software	23
Figuur 16 - Query Latest Workingtime	25
Figuur 17 - Database model Relatie Projecten - Scenario's RMS	32
Figuur 18 - Database model Projecten - Scenario's P & O	33
Figuur 19 - Een stukje van de View van WorkingTimes uit SkillsXP	34
Figuur 20 - Rapport Personeel Working Times	35
Figuur 21 - Screenshot van het Working Times overzicht. Elke regel heeft een Edit, Copy en Delete button	35
Figuur 22 - SkillsXP architectuur	41
Figuur 23 - Kaartscanner software architectuur	41

Literatuurlijst

IBM (2011). *Proven best practices for software and systems delivery and implementation and for effective project management*. Geraadpleegd op februari/maart 2011 via <http://www-01.ibm.com/software/awdtools/rup/>

C# in Depth. (02-2011). *Implementing the Singleton Pattern in C#*.
Geraadpleegd op 10 April 2011 via <http://csharpindepth.com/Articles/General/Singleton.aspx>

Bijlagen

- A: Afstudeerplan Maries Koole Feb-2011**
- B: Algemeen plan van Aanpak**
- C: Inception Rapport Urenregistratiesysteem**
- D: Elaboration Rapport Urenregistratiesysteem**
- E: Construction Rapport Urenregistratiesysteem**
- F: Inception Rapport Planningsysteem**
- G: Elaboration Rapport Planningsysteem**
- H: Request of Change P & O Requirements**
- I: Construction Rapport Planningsysteem**