

Human Resource Management System

Bachelorproject Technische Informatica
IN3700

Datum juli 2007
Versie 1.2

Auteurs Marco Krikke (1156004)
Mike Noordermeer (1178318)
Ruben Verhaaf (1153749)

Bedrijf Tam Tam B.V.

Commissie ir. B.A. Manuel (Tam Tam B.V., opdrachtgever)
dr. P.G. Kluit (Technische Universiteit Delft, begeleider)
ir. B.R. Sodoyer (Technische Universiteit Delft, coördinator)



Colofon

Document

Document type	Eindverslag
Titel	Human Resource Management System Eindverslag
Versie, datum	Versie 1.2, 26-7-2007
Project	Human Resource Management System
Bestandsnaam	hrmsystem_eindverslag.docx

Document Geschiedenis

Datum	Versie	Auteur(s)	Gebeurtenis
11 juli 2007	1.0	Marco Krikke, Mike Noordermeer, Ruben Verhaaf	Eerste major versie
16 juli 2007	1.1	Marco Krikke, Mike Noordermeer, Ruben Verhaaf	Ontwerp uitgebreid
26 juli 2007	1.2	Marco Krikke, Mike Noordermeer, Ruben Verhaaf	Ontwerp verder toegelicht en kleine wijzigingen

Document Eigenaar

Marco Krikke, Tam Tam B.V.
E-mail: marco.krikke@tamtam.nl

Voorwoord

Aan de Technische Universiteit Delft wordt de Bachelor opleiding van de studie Technische Informatica traditioneel afgesloten met een zogenaamd 'Bachelorproject'. Dit project bestaat uit het uitvoeren van een externe opdracht bij een bedrijf met een groep van circa vier studenten waar het gehele traject van Software Engineering wordt doorlopen. Het project is bedoeld om studenten ervaring op te laten doen in grotere projecten, waarbij zij hun opgedane kennis in de praktijk kunnen toepassen.

Dit eindverslag is het resultaat van een twaalf weken durend project bij Tam Tam B.V.; een full-service internetbureau gevestigd in Rijswijk. In dit verslag wordt onze opdracht en het hele proces daaromheen beschreven.

Graag willen wij Bart Manuel van Tam Tam bedanken voor zijn tijd, inzet en enthousiasme in de begeleiding van ons project. Ook willen wij Melanie van Wijngaarden en Jaqueline van Erve bedanken voor hun input en expertise op het gebied van *human resources* en hen veel plezier wensen met het eindproduct. Tevens willen wij Wouter Geurtzen, Stef van Hooijdonk, Ruben Ottink en Martin Bassie bedanken voor het delen van hun kennis en de technische ondersteuning van ons project. Tot slot willen wij alle medewerkers van Tam Tam bedanken voor de gezellige sfeer en de leuke tijd die wij hebben gehad.

Ook willen wij vanuit de Technische Universiteit Delft Peter Kluit bedanken voor de nuttige adviezen op het gebied van projectmanagement en coördinatie en zijn inzet voor de begeleiding van ons project.

Rijswijk, juli 2007

Marco Krikke
Mike Noordermeer
Ruben Verhaaf

Samenvatting

Het bachelorproject is uitgevoerd bij Tam Tam B.V. te Rijswijk. Tam Tam is sinds 1996 actief op het gebied van online marketing & communicatie en portals. De afgelopen jaren heeft Tam Tam een gestage groei doorgemaakt en op dit moment zijn er zo'n 80 medewerkers in dienst.

Gezien de voortdurende groei van Tam Tam bestaat er de wens een oplossing te creëren voor de Human Resource Management afdeling. Deze oplossing moet huidige oplossingen, bestaande uit een eenvoudige medewerkeradministratie en vele Excel sheets met persoonlijke informatie, vervangen en een volledige, portal-integrated oplossing bieden voor de HRM afdeling.

De opdracht is gestart met het opstellen van een duidelijke opdrachtschrijving waarna een plan van aanpak is gemaakt. Hierin zijn enkele constraints gedefinieerd en is een duidelijke planning opgenomen. Verder zijn de deliverables vastgelegd, waarna van start kon worden gegaan.

Tijdens de analyse fase is door middel van interviews met alle actoren een duidelijk beeld ontstaan van de requirements. Naast interviews met de afdeling HR zijn er ook interviews geweest met de afdeling financiën, de systeembeheerders en een unit manager. Daarnaast heeft de stagebegeleider, welke zelf een directielid van Tam Tam is, regelmatig zijn input geleverd op de requirements. Van alle interviews zijn duidelijke notulen gemaakt, welke zijn teruggekoppeld naar de actoren. Hierdoor hebben er geen misverstanden kunnen bestaan over de eisen en wensen van de actoren. Tot slot zijn de gedefinieerde requirements in groepen ingedeeld door middel van *cardsorting* en is aan iedere requirement een prioriteit gegeven.

De ontwerpfase is begonnen met het creëren van papieren prototypes van de gebruikersinterface. Deze zijn vervolgens met de eindgebruikers van het systeem doorgenomen, waarna ze zijn gedigitaliseerd en nogmaals zijn doorgesproken met HR. Ook zijn verschillende technieken onderzocht en is er veelvuldig gebruik gemaakt van prototypes waardoor ervaring met de techniek opgedaan kon worden. Uiteindelijk zijn een functioneel en technisch ontwerp gecreëerd, waarin ook het databasediagram en een eenvoudig klassediagram zijn opgenomen.

Vervolgens zijn de requirements geïmplementeerd, waarbij veelvuldig gebruik gemaakt is van virtual machines als testomgeving. Ook tijdens de implementatie is continu terugkoppeling naar de eindgebruikers geweest. Na de implementatie in de testomgeving is het geheel deployed op de productieomgeving, waarna er grondige tests zijn uitgevoerd. Na het slagen van de eigen tests en een acceptatietest met de opdrachtgever is de software in gebruik genomen.

Uiteindelijk is ongeveer de helft van de initiële requirements geïmplementeerd. Dit was vooraf voorzien, gezien de wensen van de opdrachtgever groter waren dan het tijdsbestek wat er voor het project was. De duidelijke prioritering van de requirements heeft hier erg bij geholpen. De planning is nauwgezet nageleefd, slechts op het einde is besloten de uiteindelijke tests pas na de deployment op de productieomgeving te doen. Dit zodat met de definitieve versie van het systeem de acceptatietest uitgevoerd kon worden.



Tam Tam

Inhoudsopgave

1	Introductie	7
	Tam Tam	7
	Opbouw document	7
2	Opdracht	8
	Huidige situatie	8
	Probleemstelling	8
	Opdrachtformulering	9
3	Aanpak en planning	10
	Aanpak	10
	Planning	12
4	Analyse	14
	Aanpak	14
	Afbakening	15
	Functionele requirements en prioritering	16
	Niet-functionele requirements	17
5	Ontwerp	18
	Beschikbare frameworks en applicaties	18
	Mogelijkheden en keuze	18
	Functioneel ontwerp	19
	Grafisch ontwerp	22
	Technisch ontwerp	23
6	Implementatie	29
	Aanpak	29
	Opbouw testportal	29
	Implementatie WebParts	30
	Rapportages	31
	Security	31
7	Testen	34
	Aanpak	34
	Testcases	34
	Issues en oplossingen	36
8	Deployment	38
	Aanpak	38
	Issues en oplossingen	39
9	Evaluatie resultaten	41
	Requirements	41
	Ontwerp	42
	Implementatie	42
	Testen	43
	Deployment	43
	Proces	44
10	Aanbevelingen	45



Tam Tam

	Aanbeveling Tam Tam	45
	Aanbeveling TU Delft	45
11	Literatuurlijst	46
12	Verklarende woordenlijst	47
A.	Huidige procedures	48
	Gegevens medewerkers	48
	In- en uitdiensttreding	48
	POP	48
	Relatie met medewerkers onderhouden	49
	Contact HR en Financiën	49
	Verzuim en vakanties	49
	Medewerkerdossier	49
	Salarisverwerking	49
B.	Uitgewerkte lijst van functionele requirements	50
C.	Beschikbare technologieën	59
	Microsoft Office SharePoint Server 2007	59
	Microsoft SQL Server 2005	59
	Microsoft .NET, C#, ASP.NET en ASP.NET AJAX	59
	Web Parts / Return of the SmartPart v1.2	59
	Microsoft Windows Workflow Foundation	60
	Microsoft InfoPath 2007 en Forms Server 2007	60
	Microsoft Excel Services	60
D.	SharePoint lists, workflows en WebParts	61
	SharePoint Lists	61
	Workflows	61
	WebParts	62
E.	Systeemarchitectuur en objectmodel	64
	Systeemarchitectuur	64
	Objectmodel	65
F.	Data dictionary	69
G.	Schermen	74

1 Introductie

Human Resource Management (HRM) is een onderdeel van het beleid van een onderneming dat direct betrekking heeft op het personeel van een onderneming. HRM houdt zich onder meer bezig met de instroom van personeel (werving en selectie), personeelsplanning, personeelsadministratie, functiemanagement, ontwikkeling en beoordeling van personeel (functioneringsgesprekken, competentie management), beloning, verzuimbeleid en uitstroom van personeel.

Voor de opslag, het beheersen en het onderhouden van HRM specifieke informatie wordt vaak een informatiesysteem ingezet waarvan zowel de HRM medewerkers als het personeel zelf gebruik maken. Voor medewerkers wordt dit systeem vooral gebruikt om persoonlijke gegevens in te zien en eventueel te wijzigen. Managers maken veelal gebruik van rapportagemogelijkheden binnen een dergelijk systeem om zo hun managementtaken correct uit te kunnen voeren.

Op het gebied van software engineering is het ontwikkelen van een HRM systeem een interessante opdracht. Het gaat hier om een veelgebruikt systeem met meerdere gebruikers met elk hun eigen wensen waarin een grote hoeveelheid aan vertrouwelijke informatie is opgeslagen. Bij de ontwikkeling van een dergelijk systeem is het belangrijk om met veel aspecten rekening te houden, waaronder de uitbreidbaarheid en veiligheid van een systeem.

Tam Tam

Tam Tam is een vooraanstaand full-service internetbureau, gevestigd in Rijswijk. Ze adviseren, ontwerpen en realiseren oplossingen op het gebied van online marketing & communicatie en (informatie-)portals. Tam Tam is al 11 jaar een succesvol bedrijf en werkt met talentvolle jonge adviseurs, ontwerpers en ontwikkelaars voor toonaangevende Nederlandse en internationale klanten als Oxxio, Royal Haskoning, ProRail en Hunter Douglas.

Opbouw document

De opbouw van dit document is chronologisch, waarbij de verschillende fasen van het project besproken worden. We beginnen met het uiteenzetten van onze opdracht waarna opeenvolgend de aanpak en planning, de analyse, het ontwerp, de implementatie, het testtraject en de uitrol besproken worden. Hierna zullen we de door ons behaalde resultaten evalueren waarna we afsluiten met onze aanbevelingen. In de verschillende appendices is aanvullende informatie te vinden, waar in dit document naar verwezen zal worden. Tot slot is er ook een verklarende woordenlijst aanwezig voor de uitleg van gebruikte terminologie.

2 Opdracht

In de afgelopen 11 jaar heeft Tam Tam een gestage groei meegemaakt; in het laatste jaar is het bedrijf met 50% gegroeid tot een aantal medewerkers van rond de 80. Met deze groei is ook de vraag naar een systeem om *human resource* specifieke gegevens te beheren toegenomen.

Huidige situatie

Tam Tam maakt intern gebruik van een intranet portal op basis van Microsoft Office SharePoint Server 2007 (MOSS2007 of kortweg SharePoint), waarin de belangrijkste bedrijfsprocessen worden beheerd en gemonitord. SharePoint heeft als voordeel dat het eenvoudig is om een portal op te zetten; het is een systeem waarin door veelal klikken en slepen de basis van een portal en verschillende applicaties die hierbinnen functioneren gelegd kan worden. Zo zijn er applicaties binnen de portal waarin declaraties voor reiskosten kunnen worden gemaakt, applicaties waarin uren geboekt kunnen worden op klanten en projecten, is er een lijst met medewerkers en een beheerpagina voor deze lijst. Het is ook eenvoudig om bijvoorbeeld een *document library*, *wiki* of een *blog* aan te maken op de portal.

De portal is ook de plaats waar door medewerkers zelf een profiel wordt bijgehouden waarin te zien is met welke projecten ze bezig zijn, wat hun *skills* zijn en met welke collega's ze nauw samenwerken. De pagina waarin dit wordt weergegeven, heet de My Site. Dit is een standaard beschikbare functionaliteit binnen een SharePoint portal.

Human Resource Management

Binnen Tam Tam worden verschillende applicaties gebruikt om het bedrijfsproces te stroomlijnen. Op dit moment wordt Human Resource Management gedaan middels de volgende systemen:

- Tammo-administratie: een applicatie binnen het Tam Tam intranet waarin alle adresgegevens en gerelateerde informatie van medewerkers worden bijgehouden, naast ook enkele gegevens met betrekking tot arbeidsvoorwaarden. De beveiliging van dit systeem is niet erg strikt; zo is het bijvoorbeeld mogelijk om van een willekeurige werknemer de adresgegevens te veranderen zonder hiervoor speciale rechten nodig te hebben;
- Excel-bestanden: het aantal beschikbare vakantie-uren van een medewerker wordt handmatig bijgehouden in een Excel-bestand. Elke maand wordt handmatig een nieuw overzicht van ieders vakantiedagen gemaakt en via mail naar alle medewerkers verstuurd.

Probleemstelling

Als snel groeiende organisatie heeft Tam Tam behoefte aan een structuur in proces en informatie voor Human Resource Management, welke op dit moment nog teveel ontbreekt. Ze denken hierbij aan *tracking* van werving- en selectie en registratie van primaire en secundaire arbeidsvoorwaarden (inclusief ontwikkeling en historie), opleidingen en competenties, verlof, ontwikkeling, etc. De ontwikkeling van een dergelijk systeem zal waarschijnlijk leiden tot een effectievere organisatie rondom personeelsmanagement, waardoor HR managers zich meer kan richten op hun primaire taken.

Opdrachtformulering

Er bestaat uiteraard standaard software voor HRM van bijvoorbeeld bedrijven als SAP of Oracle. Echter, Tam Tam heeft als internetbureau een reputatie waar te maken op het gebied van *Employee Self Service*, informatievoorziening in de vorm van *portal dashboards* en moderne procesautomatisering. Zij willen dan ook een dergelijke geïntegreerde oplossing van een HRM systeem laten onderzoeken, ontwerpen en ontwikkelen. Er zal hierbij een onderzoek moeten plaatsvinden naar de gewenste functionaliteit vanuit de organisatie waaruit een aantal requirements zullen worden geformuleerd. Vervolgens worden deze requirements vertaald in een ontwerp dat geïmplementeerd en getest zal worden. Uiteindelijk dient de implementatie, wanneer voldoende functioneel, uitgerold te worden in de organisatie van Tam Tam.

Randvoorwaarden

Er gelden een aantal randvoorwaarden bij de ontwikkeling van het HRM systeem, waarbij gedacht moet worden aan te gebruiken software en een aantal gebruiksvoorwaarden. Op het gebied van software en technieken dient er gebruik gemaakt te worden van:

- ASP.NET 2.0;
- Microsoft Office SharePoint Server 2007;
- ASP.NET AJAX waar bruikbaar en nuttig.

Tevens dient aan de volgende gebruiksvoorwaarden voldaan te worden:

- Web based, portal integrated oplossing;
- Role based security;
- Gebruiksvriendelijk en snel;
- Uitbreidbaar en onderhoudbaar in de toekomst;
- Goed gedocumenteerd.

Deliverables en rapportage

Als eindresultaat van dit project dienen een aantal producten opgeleverd te worden. Het betreft hier een aantal documenten die bij een standaard software engineering traject horen met daarnaast uiteraard ook het eindproduct en de bijbehorende documentatie. De volgende producten dienen opgeleverd te worden:

- Plan van Aanpak;
- Requirements;
- Ontwerp (functioneel en technisch);
- Werkende en geteste implementatie HRM systeem;
- Eindrapport;
- Presentatie.

Deze deliverables zijn het product van de verschillende fasen waaruit ons project bestaat. Een beschrijving van deze fasen is terug te vinden in het hoofdstuk Planning.

3 Aanpak en planning

Om een project tot een succesvol einde te laten komen is het van groot belang om voor de daadwerkelijke start van het project goed na te denken over de te volgen aanpak en de hierbij behorende planning. In dit hoofdstuk beschrijven wij de door ons gekozen manier van werken en de redenen voor deze keuzes.

Aanpak

Bij de aanpak van een dergelijk groot project dient er te worden nagedacht over de te gebruiken ontwerpmethodiek met zijn bijbehorende fasen en de kwaliteitsborging van deze verschillende fasen die uiteindelijk samenkomen tot een eindproduct. Het gaat hier niet alleen om een beste aanpak, voor zover deze al bestaat, maar een aanpak die toegespitst is op de specifieke situatie bij Tam Tam en waarmee wij als projectgroep het beste uit de voeten kunnen. Dit is de reden dat wij bij dit project gekozen hebben voor een ietwat afwijkende manier van werken, zoals we hieronder zullen toelichten.

Ontwerpmethodiek

Bij het opstellen van het plan van aanpak werd al snel duidelijk dat de keuze voor een ontwerpmethodiek niet triviaal zou zijn. We stonden hierbij voor de keuze tussen het klassieke watervalmodel en een meer *agile* of *spiraalmodel* benadering waarbij we in convergerende stappen tot een steeds completer eindproduct zouden komen.

Een *agile* benadering zou tot gevolg hebben dat elke iteratie een miniatuurproject op zich is, bestaande uit een losse planning, analyse, ontwerp, testtraject en schrijven van documentatie. Praktisch gezien leek dit ons om een aantal redenen geen juiste aanpak. Al snel werd duidelijk dat er op deze manier slechts een beperkt zicht op het gehele project zou zijn; veel onderdelen van de applicatie leken met elkaar samen te gaan hangen waardoor het nuttig leek om de requirements en het ontwerp van de gehele applicatie allereerst duidelijk te hebben voordat de daadwerkelijke implementatie van start ging. Vanuit Tam Tam leek dit ook de beste aanpak; gezien de drukke agenda van de betrokken medewerkers was het praktischer om de requirements in zijn geheel duidelijk te hebben. Op deze manier kon tevens duidelijk afgebakend worden welke ideeën voor een dergelijk systeem wel thuishoorden in de applicatie en welke zaken weliswaar nuttig leken, maar uiteindelijk niet thuishoren in een HRM systeem.

We hebben uiteindelijk voor het traditionele watervalmodel gekozen, maar met enkele belangrijke aanpassingen. Allereerst hebben we besloten de requirements zo duidelijk mogelijk te groeperen in subsystemen en daarbij de onderlinge afhankelijkheden aan te geven. Op deze manier kan er tijdens de implementatiefase toch *semi-agile* ontwikkeld worden, waarbij de lijst met subsystemen per onderdeel kan worden afgewerkt. Tevens heeft dit tot gevolg dat alle requirements duidelijk zijn, het hele ontwerp af is, maar het gezien de tijdslimiet van dit project niet per se nodig is om alle subsystemen te implementeren. Vanuit Tam Tam kan er dan altijd nog worden besloten om het project volledig af te maken op basis van onze requirements en ontwerp.

Omdat we te maken hadden met voor ons vrij onbekende technologieën als SharePoint, met daarbij veelal zeer recent ontwikkelde mogelijkheden in deze software die nog geen gemeengoed zijn bij ontwikkelaars, hebben we een tweede aanpassing in het watervalmodel aangebracht. Met deze onbekende technologieën in het achterhoofd hebben we gekozen om geen volledige scheiding aan te brengen tussen ontwerp en implementatie. Door middel van het maken van *proofs of concept* hebben we tijdens de ontwerpfase al

zoveel mogelijk technieken en oplossingen uitgewerkt in een conceptversie, om op deze manier duidelijk te krijgen in hoeverre onze ideeën over het ontwerp in de praktijk ook daadwerkelijk te ontwikkelen waren. Tevens was dit een goede oefening met voor ons genoemde onbekende technieken. Op deze manier werd ook redelijk afgevangen dat we niet voor onoverkomelijke verrassingen zouden komen te staan tijdens de implementatiefase. Deze manier van werken stelde ons in staat om een goede schatting te maken van de benodigde tijd voor de implementatiefase.

In het kort komt onze benadering neer op een project met daarin een vijftal fasen zoals deze hieronder genoemd zijn. Deze fasen staan redelijk vast, maar we hebben te allen tijde getracht zo efficiënt en slim mogelijk te werken binnen de tijd die we hebben. Een deel van de implementatie werd zoals gezegd al gedaan in de ontwerpfase, waarin door middel van *proofs of concept* een soort van prototypes werden ontwikkeld.



Figuur 1 - Projectfasen

Voor een verdere beschrijving van de aanpak tijdens de verschillende fasen van het project verwijzen we naar de eerste paragrafen van de hoofdstukken Analyse, Ontwerp, Implementatie, Testen en Deployment.

Prioritering requirements

Aangezien vanaf het begin duidelijk was dat project veel werk zou zijn en er een grote kans was dat we niet alle requirements konden kunnen opleveren, lag de uitdaging erin een duidelijke prioritering aan te geven in de requirements. We hebben hiervoor een variatie op de welbekende MoSCoW-analyse gebruikt met een aanpassing in het aantal verschillende niveaus, om zo tot een betere detaillering te kunnen komen. Ook zijn de requirements ingedeeld in hoofdgroepen die elk een subsysteem voorstellen of behelzen.

Kwaliteitsborging

Om de kwaliteit van de gemaakte software te waarborgen hebben we gebruik gemaakt van een aantal veel gebruikte technieken, waaronder het gebruik van *coding standards*, standaard design-patterns, reviews van code door groepsgenoten en *end-user tests*. Gezien het feit dat de door ons geschreven code veelal nauw gekoppeld is aan user-interfaces, had het weinig tot geen zin om gebruik te maken van unit-testing.

Tijdens de analysefase hebben we interviews gehouden met verschillende *stakeholders* waarvan notulen gemaakt zijn en ter goedkeuring voorgelegd aan de betrokkenen, om zo eventuele miscommunicatie te voorkomen. Ook werden wijzigingen aan productiesystemen gedocumenteerd en verspreid onder belanghebbenden. Meer informatie over kwaliteitsborging zal te vinden zijn bij de behandeling van elke fase van dit project.

Voor het versiebeheer van de code is gebruik gemaakt van Vault van SourceGear. Voor documenten is er gebruik gemaakt van het ingebouwde versiebeheer in de intranet portal van Tam Tam.

Begeleiding

Afgesproken met onze begeleider was om in het begin wekelijks te overleggen, naast de vele sessies die toen met andere *stakeholders* waren gepland. Als het project eenmaal op gang was zou er, wanneer nodig, tweewekelijks overleg plaatsvinden om de voortgang te bespreken. Ook waren er sessies gepland rondom onze *milestones*. Alle overleggen waren in principe van tevoren ingepland, maar er werd niet overbodig vergaderd wanneer dit niet nodig was. Aan de andere kant werden er wel extra sessies ingepland wanneer hier vraag naar was vanuit ons of de begeleider.

Planning

De eerder genoemde vijf fasen in ons project zijn in het plan van aanpak verwerkt in een planning, waarbij ook het oefenen met SharePoint en ASP.NET, het samenstellen van dit eindverslag en het geven van een presentatie zijn verwerkt. De originele planning zag er als volgt uit:

ID	Task Name	Start	Finish	Duration	apr 2007			mei 2007				jun 2007				jul 2007		
					15-4	22-4	29-4	6-5	13-5	20-5	27-5	3-6	10-6	17-6	24-6	1-7	8-7	
1	Plan van Aanpak	16-4-2007	18-4-2007	3d														
2	Inwerken / verdiepen in Sharepoint / ASP.NET	16-4-2007	27-4-2007	2w														
3	Analyse / requirements	18-4-2007	4-5-2007	2w 3d														
4	Interviews HR manager	18-4-2007	26-4-2007	1w 2d														
5	Interviews systeembeheer	19-4-2007	26-4-2007	1w 1d														
6	Opstellen requirements	19-4-2007	2-5-2007	2w														
7	Accordering requirements	3-5-2007	4-5-2007	2d														
8	Ontwerp	3-5-2007	23-5-2007	3w														
9	Ontwerp	3-5-2007	23-5-2007	3w														
10	Proof of Concepts	7-5-2007	23-5-2007	2w 3d														
11	Implementatie	24-5-2007	20-6-2007	4w														
12	Deployment	21-6-2007	4-7-2007	2w														
13	(Integratie)testen	21-6-2007	29-6-2007	1w 2d														
14	Deployment	2-7-2007	4-7-2007	3d														
15	Eindverslag samenvoegen / uitloop	25-6-2007	6-7-2007	2w														
16	Presentatie	9-7-2007	13-7-2007	1w														

Figuur 2 - Planning

De eerste drie dagen zijn gebruikt om wat meer duidelijkheid over de opdracht te krijgen en een plan van aanpak te schrijven. Hierna zijn we gestart met het interviewen van relevante medewerkers om zodoende in een ruime twee weken de requirements op te stellen. Tegelijkertijd hebben we ons verdiept in ASP.NET en SharePoint.

Nadat de requirements waren opgesteld en goedgekeurd zijn we begonnen met het ontwerp van de applicatie, waarbij de *proofs of concept* zijn gebruikt om de functionaliteit in het ontwerp te testen op implementeerbaarheid. Hiervoor waren drie weken ingepland, waarna er vier weken ingepland stonden voor

de daadwerkelijke implementatie van de applicatie. In deze vier weken hebben we tevens zoveel mogelijk losstaande code getest en hebben er user-tests plaatsgevonden.

Tot slot zijn er nog twee weken ingepland voor de deployment en het integratietesten van de applicatie in de productieomgeving. Tijdens deze periode is tevens tijd ingepland voor het samenstellen van dit eindverslag.

Elke fase van het project wordt gemarkeerd door *milestones* in het traject, welke hieronder in een tabel staan samengevat.

Datum	Milestone
4 mei 2007	Analyse en requirements af en goedgekeurd
23 mei 2007	Ontwerp af en goedgekeurd
20 juni 2007	Implementatie af, <i>code freeze</i>
4 juli 2007	Systeem uitgerold
6 juli 2007	Eindverslag opgeleverd

Tabel 1 - Milestones

4 Analyse

Tijdens de analysefase is bepaald wat de eisen van het HRM systeem moeten zijn. Resultaat van deze fase is een verzameling requirements met bijbehorende prioritering en indeling in groepen.

Allereerst wordt onze aanpak van deze fase beschreven, waarin onder andere een analyse wordt gemaakt van de actoren, het betreffende domein en de systeemomgeving. Hierna bakenen we de functionaliteit van ons systeem af, waarna we de functionele en niet-functionele requirements zullen uiteenzetten.

Aanpak

We hebben besloten om de requirements te verzamelen door middel van interviews met *stakeholders* en door te kijken naar de mogelijkheden van concurrerende pakketten. De requirements dienden zich toe te spitsen op functionaliteit die bruikbaar is binnen Tam Tam. Al snel stond hiermee vast dat het door ons te ontwikkelen systeem geen 'standaard' HRM systeem zou worden.

We zullen nu kort de actoren binnen het systeem en het domein waarin het systeem thuishoort beschrijven. Hierna worden de huidige procedures beschreven en wordt er gekeken naar concurrerende softwarepakketten.

Actoren

We hadden te maken met een zestal verschillende actoren binnen het systeem, die allemaal hun eigen eisen en functionaliteit hadden:

- HRM managers;
- Unit managers;
- Financieel medewerkers;
- Systeembeheerders;
- Directie;
- Overige medewerkers.

Om tot een goed beeld van de eisen aan het systeem te komen, hebben we met elke gebruikersgroep één of meerdere interviews georganiseerd. Na elk interview zijn de conclusies uitgewerkt en ter goedkeuring voorgelegd aan de betrokkenen. Op basis hiervan is een verzameling requirements samengesteld die in een ideaal geval tot een applicatie zouden leiden waar iedereen tevreden mee was. Er werd echter al snel duidelijk dat niet elke functionaliteit daadwerkelijk thuishoorde in een HRM applicatie, hoewel hier blijkbaar wel vraag naar was vanuit de organisatie. We zullen hier op terugkomen bij de afbakening van de functionaliteit van ons systeem.

Domeinanalyse

Het gebouwde HRM systeem wijkt af van een standaardoplossing doordat dit project specifiek is toegespitst op de organisatorische structuur en procedures binnen Tam Tam. Dit wil zeggen dat er gefocust is op de functionaliteit die het meest nodig en nuttig is; ofwel de functionaliteit die voor de organisatie het meest oplevert. De functionaliteit is onderverdeeld in de volgende HR-specifieke domeinen:

- Personeelsadministratie;
- Arbeidsvoorwaarden;
- Verzuim en vakanties;
- Uit- en indiensttreding;
- Beoordeling, loopbaan, opleiding en ontwikkeling.

Al deze domeinen hebben hun eigen administratie en bijbehorende procedures, actoren en rapportages. Op dit moment worden de meeste gegevens opgeslagen in Excel sheets, in de Tammo administratie en met behulp van papieren dossiers. Voor de financiële administratie wordt Exact software gebruikt.

De huidige procedures zijn nog niet altijd voldoende geformaliseerd en gedocumenteerd. Tevens worden de procedures vrijwel altijd met de hand uitgevoerd en is er veel controle nodig op het correcte verloop van de procedures. De belangrijkste bestaande procedures zijn opgesomd in Appendix A.

Concurrerende software

Er is veel concurrerende software aanwezig op de markt, waaronder:

- SAP ERP Human Capital Management;
- Exact e-Synergy;
- Unit 4 Personeel;
- Oracle PeopleSoft Enterprise Human Capital Management;
- People Partners Emplaza.

We hebben bovenstaande pakketten en tevens wat kleinere pakketten bestudeerd op hun functionaliteit, onder andere door middel van beschikbare demo's en white papers. Het viel op dat veel van de software met name bedoeld is voor zeer grote bedrijven; slechts de software van Unit4, Exact en People Partners leek binnen het bereik van Tam Tam te liggen.

Wat opviel was dat de onderzochte software niet direct binnen de SharePoint Portal van Tam Tam zou integreren. Tam Tam heeft als portal bureau een reputatie hoog te houden op het gebied van Employee Self Service, informatievoorziening in de vorm van *portal dashboards* en moderne procesautomatisering. Zij willen dan ook een dergelijke geïntegreerde oplossing laten onderzoeken, ontwerpen en ontwikkelen en niet direct kiezen voor een kant-en-klaar product.

Systeemomgeving

Tam Tam maakt gebruik van een intranet met daarop een web based portal. Deze portal functioneert via zowel het interne intranet als het internet op kantoor- en ontwikkelpc's met daarop Microsoft Internet Explorer. De portal draait op een server met Microsoft Windows 2003, Microsoft Office SharePoint Server 2007, Microsoft SQL Server 2005 en IIS 6.0.

Afbakening

Om tot een duidelijk overzicht van requirements te komen hebben we besloten om de functionaliteit van ons systeem af te bakenen. Na het afnemen van interviews met verschillende belanghebbenden en het

toepassen van *cardsorting* op alle ontdekte requirements is duidelijk geworden dat de volgende groepen van requirements te onderscheiden zijn:

1. Personeelsadministratie;
2. Arbeidsvoorwaarden;
3. Verzuim en vakanties;
4. Uit- en indiensttreding;
5. Beoordeling, loopbaan, ontwikkeling, opleiding;
6. Personeelsdossier.

We hebben in overleg met de opdrachtgever besloten om recruitment en een koppeling van de urenregistratie aan de declarabiliteit, verzuim en verlof van een medewerker buiten ons systeem te laten, aangezien ze minder sterk verbonden zijn met het kernproces. Tevens is de vraag vanuit de organisatie vooral gericht op bovenstaande punten.

Om op bovenstaande gebieden het proces te automatiseren, zijn we tot een aantal functionele requirements gekomen, die zijn ingedeeld in de zes eerder genoemde requirementsgroepen. De prioriteit van elke functionele requirement wordt aangegeven met een getal tussen de 1 en de 9, waarbij 1 de hoogste prioriteit heeft en 9 de laagste. Er is voor deze aanpak gekozen omdat de MoSCoW methode slechts een viertal prioriteiten kent, wat bij een dergelijk grote hoeveelheid requirements en een beperkte hoeveelheid beschikbare tijd als hier het geval is een accurate prioritering onmogelijk maakt.

We hebben in overleg met de opdrachtgever besloten dat voor het slagen van het project ten minste de requirements met de nummers 1 en 2 geïmplementeerd dienden te worden. Het zou daarnaast mooi zijn als er tijd genoeg was om meer requirements te implementeren. Een nadere afbakening hiervan kon gemaakt worden aan het begin van de implementatiefase.

Functionele requirements en prioritering

Na uitgebreide interviews zijn de requirements samengesteld en onderverdeeld in de bovenstaande zes groepen.

Bij personeelsadministratie gaat het om het opslaan en beheren van primaire gegevens, zoals personalia. Bij arbeidsvoorwaarden is het mogelijk om nieuwe sets arbeidsvoorwaarden aan te maken en de historie te bekijken. Verzuim en vakantie houdt het verzuim van medewerkers bij en geeft een overzicht van gebruikte vakantiedagen. Uit- en indiensttreding en beoordeling, loopbaan, ontwikkeling en opleiding beheersen de procedures rondom deze zaken en hebben ze mogelijkheid om hierover rapporten weer te geven. Ook vindt hier een synchronisatie met andere systemen plaats. In het digitaal dossier ten slotte worden alle relevante gegevens als contracten en POP formulieren opgeslagen.

Een uitgewerkt overzicht van alle functionele requirements is te vinden in Appendix B.

Niet-functionele requirements

Met de volgende niet-functionele requirements dient rekening gehouden te worden tijdens het ontwikkelen van het eindproduct:

- Iedere pagina moet in 95% van de gevallen openen binnen twee seconden en rapporten dienen binnen 30 seconden gegenereerd te kunnen worden;
- Het systeem moet minimaal 10 gebruikers tegelijk aankunnen zonder merkbare vertraging;
- De beschikbaarheid en betrouwbaarheid van het systeem moet dusdanig zijn dat het de bedrijfsprocessen van Tam Tam goed kan ondersteunen. Dit betekent dat de verwachte *uptime* boven de 99% moet liggen;
- Het systeem dient goed gedocumenteerd te zijn en voldoende onderhoudbaar te zijn om in de toekomst door andere softwaredevelopers te kunnen worden aangepast;
- Foutmeldingen dienen voldoende afgevangen te worden om verlies van data te voorkomen en dienen duidelijk voor de gebruiker te zijn.

De in hoofdstuk 2 genoemde randvoorwaarden zijn in het analyseproces ietwat uitgebreid en gewijzigd naar de volgende constraints:

- Er dient gebruik gemaakt te worden van Microsoft ASP.NET 2.0;
- Het product dient te integreren in de Microsoft Office SharePoint Portal van Tam Tam;
- ASP.NET AJAX dient gebruikt te worden waar bruikbaar en nuttig;
- Dataopslag dient te gebeuren binnen Microsoft Office SharePoint Server of Microsoft SQL Server 2005;
- Het geheel moet draaien op een Microsoft Windows Server 2003 omgeving;
- Het product dient te functioneren met Microsoft Internet Explorer 6 en 7 als client;
- Het project dient binnen 12 weken afgerond te worden.

5 Ontwerp

Tijdens de ontwerpfase hebben we gekeken naar verschillende mogelijkheden om het systeem vorm te geven. Uit deze verscheidene oplossingen hebben we een keuze gemaakt en vervolgens zowel een functioneel als technisch ontwerp gemaakt van de verschillende onderdelen van het systeem.

In dit hoofdstuk bespreken we kort de beschikbare frameworks en applicaties en de keuzes die we hadden bij het uitvoeren van onze opdracht. Hierna bespreken we opeenvolgend het functioneel ontwerp, grafisch ontwerp en het technisch ontwerp.

Beschikbare frameworks en applicaties

Bij het ontwerp is gekeken welke applicaties en *frameworks* beschikbaar zijn en van nut konden zijn voor onze applicatie. Aangezien het grote aanbod van al bestaande technologieën en gezien hergebruik van software het ontwikkelproces sterk kan versnellen leek het ons verstandig dat onze applicatie voor een groot deel zou leunen op deze al bestaande *frameworks* en applicaties. Tijdens het ontwerp hebben we een keuze gemaakt uit een aantal beschikbare systemen die voor onze applicatie nuttig konden zijn en welke we later in ons ontwerp konden gebruiken. Het gaat hierbij om de volgende technologieën:

1. Microsoft Office SharePoint Server 2007;
2. Microsoft SQL Server 2005;
3. Microsoft .NET, C#, ASP.NET en ASP.NET AJAX;
4. WebParts / Return of the SmartPart v1.2 beta;
5. Microsoft Windows Workflow Foundation;
6. Microsoft InfoPath 2007 en Forms Server 2007;
7. Microsoft Excel Services;
8. Microsoft Sequel Reporting Services.

Voor deze applicaties is gekozen gezien ze veelal al ingezet werden bij Tam Tam en vanwege de integratiemogelijkheden met elkaar.

Een uitgebreid overzicht van gebruikte technologieën is te vinden in Appendix C.

Mogelijkheden en keuze

In beginsel waren er drie mogelijkheden voor de implementatie van een HRM systeem in de organisatie. Men kon een kant-en-klare HRM applicatie aanschaffen, een custom ASP.NET applicatie bouwen en een applicatie bestaande uit losse WebParts integreren in de SharePoint portal.

HRM-applicatie aanschaffen

Alvorens aan het werk te gaan met het ontwikkelen van een applicatie hebben we gekeken of het aanschaffen van een HRM applicatie niet doeltreffender was dan het wiel opnieuw uitvinden. We hebben hiervoor enkele applicaties die op de markt zijn onder de loep genomen en onze eisen vergeleken met de specificaties van de desbetreffende pakketten.

Custom ASP.NET-applicatie

Een tweede mogelijkheid was het ontwikkelen van een custom ASP.NET applicatie. Voordelen van deze aanpak zouden zijn dat het ontwikkelen eenvoudiger zou zijn en dus waarschijnlijk sneller zou gaan. Een nadeel is echter de integratie in de portal, die niet eenvoudig is. Tevens zou het lastig zijn om standaard functionaliteit uit SharePoint te gebruiken.

SharePoint WebParts

De laatste mogelijkheid is een systeem van WebParts die binnen de SharePoint Portal kunnen worden toegevoegd aan pagina's en samen een applicatie vormen. Op deze manier ontwikkel je de applicatie per WebPart en is het erg eenvoudig om WebParts elders op de portal her te gebruiken. Dit geeft een grote mate van vrijheid en beperkt de ontwikkeling tot hapklare brokken functionaliteit. Tevens is standaard functionaliteit binnen SharePoint, zoals *Lists* en *alerts*, eenvoudig te gebruiken.

Na vergelijking van de specificaties met onze requirements bleek dat aanschaf van een applicatie niet tot de mogelijkheden behoorde, aangezien geen van de applicaties voldeed aan de gestelde eisen. Met name portal-integratie zou onmogelijk worden en de meeste pakketten waren ook veel te uitgebreid voor de functionaliteit die Tam Tam nodig had. Tevens ontbrak veelal specifieke functionaliteit die Tam Tam wel graag in het systeem terug wilde zien. Het uitbreiden en eventueel integreren van een bestaande applicatie in een SharePoint systeem werd door geen van de applicaties voldoende ondersteund.

Een volledige custom applicatie leek een winnende mogelijkheid, maar de moeizame integratie met SharePoint en het lastige gebruik van functionaliteit uit SharePoint zorgde dat ook deze methode al snel minder aantrekkelijk werd. De integratie in de portal zou op deze manier waarschijnlijk bestaan uit het in zijn geheel integreren van een .NET applicatie in een SharePoint pagina; iets wat technisch gezien wel mogelijk is, maar geen schoonheidsprijs verdient.

We hebben daardoor gekozen voor het ontwikkelen van losstaande WebParts waarmee met behulp van SharePoint de applicatie kon worden opgebouwd. Dit zorgde voor een optimale samenwerking met de functionaliteit van SharePoint en leidde tot een modulaire opbouw van de applicatie. Ook was hierdoor hergebruik van bepaalde WebParts op bijvoorbeeld de persoonlijke MySite van medewerkers eenvoudig.

Functioneel ontwerp

In het functioneel ontwerp zullen we eerst beschrijven hoe het systeem eruit ziet voor verschillende gebruikersgroepen. Vervolgens komt aan bod welke data moet worden opgeslagen en wat de samenhang hiertussen is.

Het systeem voor verschillende gebruikersgroepen

Om het systeem voor de verschillende gebruikersgroepen goed en overzichtelijk te laten werken, zullen zij allen hun eigen *view* op het systeem hebben. In dit gedeelte bespreken we kort hoe de verschillende gebruikersgroepen de applicatie zien.

HR managers en directie

Het centrale gedeelte van de site voor HR managers en de directie bestaat uit een *dashboard*, waarop de belangrijkste gegevens voor de HR manager zichtbaar zijn. Van hieruit kan er naar een overzicht van taken, medewerkers of rapportages worden geklikt. Alle taken zijn zichtbaar voor de HR manager, inclusief die van

andere actoren in het systeem. In de medewerkerlijst ziet men de basisgegevens van de medewerker, waarna men kan doorklikken naar de details. Vanuit de details heeft men toegang tot zaken als de vakantiedagen, verzuim, digitaal dossier, etc. Onder rapportage kan men overzichten van salarissen, bonussen, ziekte, verzuim, etc. bekijken. In principe heeft de directie dezelfde rechten als de HR managers.

Unit managers

De unit managers hebben dezelfde opbouw van de site als de HR managers, met als verschil dat zij alleen toegang hebben tot de gegevens van de betreffende medewerkers in hun unit. Ook kunnen zij geen rapportages over het gehele bedrijf bekijken, maar wederom slechts voor hun unit. De unit managers zien alleen de taken die voor hen of de medewerkers in hun unit bedoeld zijn.

Systeembeheerders

Systeembeheer heeft slechts toegang tot hun takenlijst en tot de lijst met hardware die de medewerkers in hun bezit hebben.

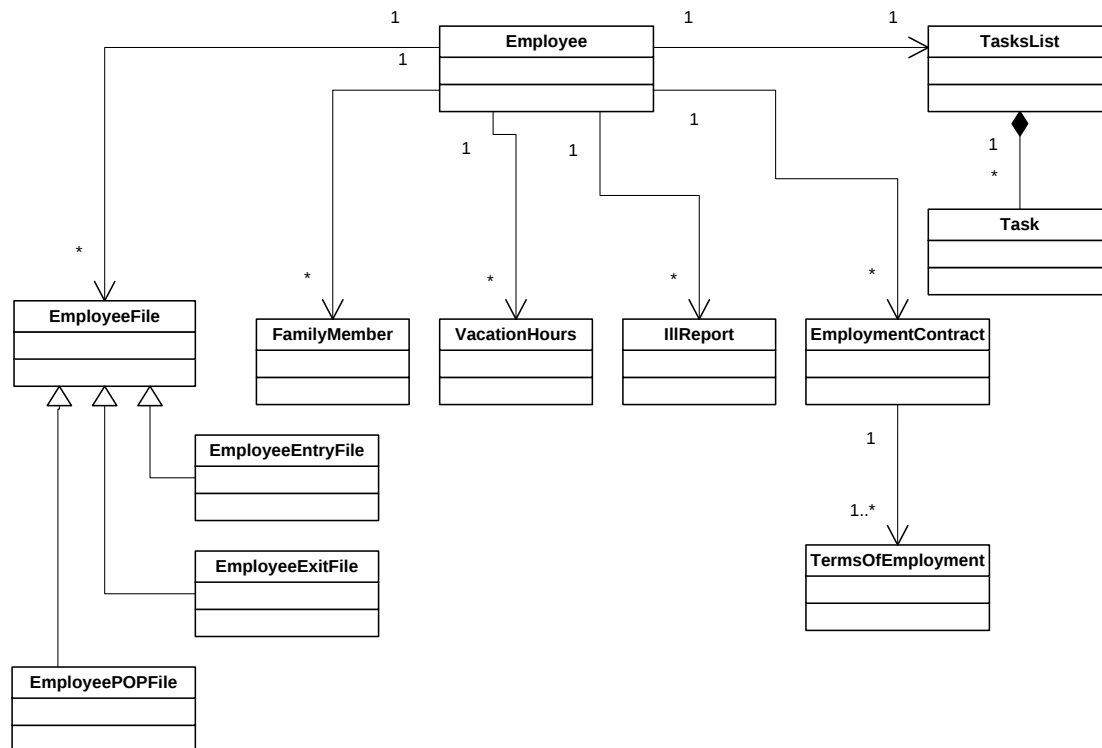
Medewerkers

Waar door de HR manager, unit managers en systeembeheer gebruik wordt gemaakt van een applicatie die rondom een dashboard is opgebouwd, kunnen medewerkers het systeem benaderen op een aparte, eigen pagina. Hier hebben zij toegang tot hun takenlijst, personalia, arbeidsvoorwaarden, vakantiedagen. Zij kunnen op deze wijze tevens hun adres- en bankgegevens wijzigen.

Data en relaties

We zullen nu een kort overzicht geven van de data die opgeslagen dient te worden in het systeem en de relaties tussen de verschillende data. Allereerst zullen de relaties duidelijk gemaakt worden aan de hand van een vereenvoudigd objectdiagram, waarna door middel van een data dictionary de exacte eigenschappen worden besproken alsmede de rechten die de verschillende gebruikers op de data hebben.

Relaties



Figuur 3 - Relaties

Alle data in het systeem is geconcentreerd rond de medewerker (Employee). Alle bijgehouden data hoort dan ook bij een medewerker. We geven nu een korte beschrijving van de objecten, de getallen achter de beschrijving verwijzen naar de requirements in Appendix B.

1. Een medewerker heeft arbeidscontracten. Deze beschrijven één fysiek arbeidscontract (EmploymentContract) (2).
 - a. Binnen een arbeidscontract kunnen op meerdere momenten wijzigingen optreden in de arbeidsvoorwaarden. Zo kan het loon tussentijds gewijzigd worden. Hiertoe worden er meerdere versies van de arbeidsvoorwaarden per contract bijgehouden. (TermsOfEmployment)
2. De ziekmeldingen van een medewerker worden bijgehouden (IllReport) (3).
3. De vakantieuren van een medewerker worden bijgehouden in het systeem. Hiertoe ontvangt een medewerker bij of afschrijvingen op zijn saldo (VacationHours) (3).
4. De familieleden van een medewerker worden bijgehouden (FamilyMember) (1e/1f).
5. Iedere medewerker heeft een persoonlijke takenlijst (TasksList) (4a/4b/4c).
 - a. De lijst met taken bevat uiteraard taken (Task).
6. Van iedere medewerker wordt een digitaal dossier bijgehouden. Dit dossier bevat verschillende soorten bestanden. De inhoud van de bestanden is verder niet relevant voor het systeem. Wel is in

sommige gevallen het type bestand relevant voor het systeem. Om deze reden zijn er enkele subklassen (EmployeeFile) (4/5/6).

- a. Er is een specifiek bestandstype voor het indiensttreden van een medewerker. Hier zijn standaardlijsten voor beschikbaar welke het systeem moet opslaan (EmployeeEntryFile) (4).
- b. Er is een specifiek bestandstype voor het uitdiensttreden van een medewerker. Hier zijn standaardlijsten voor beschikbaar welke het systeem moet opslaan (EmployeeExitFile) (4).
- c. Er is een specifiek bestandstype voor het evalueren van een medewerker. Dit proces wordt ieder half jaar uitgevoerd door middel van een standaardformulier. Dit formulier moet digitaal in het systeem opgeslagen worden (EmployeePOPFile) (5).

Data dictionary

De verschillende eerder genoemde objecten worden in Appendix F uitgebreider beschreven. Hierbij wordt dieper ingegaan op welke data er precies moet worden opgeslagen en welke rechten de verschillende gebruikers hebben op de data. De tabellen geven per entiteit de eigenschappen met hun type (zoals te vinden in gebruikelijke programmeertalen), een beschrijving en de rechten van de verschillende gebruikers. De namen van de eigenschappen zijn in het Engels, zodat de overeenkomst met het latere datamodel, waar de namen ook Engels zijn, duidelijk is.

Grafisch ontwerp

Zoals eerder besproken zijn er twee verschillende varianten van de applicatie, één voor de HR medewerkers, directie, unit managers en systeembeheerders en één voor de overige medewerkers. De eerste applicatie is gebouwd in de SharePoint portal en heeft een *dashboard* centraal staan. De medewerkerapplicatie is gebouwd in de al bestaande My Site, die daar dient als centrale plaats, in plaats van het dashboard bij de eerste applicatie. We zullen nu kort beide aanpakken bespreken aan de hand van ruwe schetsen van de schermen welke te vinden zijn in Appendix G.

De navigatiestructuur van de pagina's is bewust eenvoudig gehouden. We hebben wederom *cardsorting* toegepast om een logische menustructuur op te bouwen vanuit de schermen die we ontwikkeld hadden. Het menu is aan de linkerkant te vinden, waar naar de onderliggende pagina's geklikt kan worden. Onder het hoofdmenu is er ruimte voor een aantal *quicklinks* naar dieper liggende en veelgebruikte locaties. Bovenaan elke pagina staan de zogenaamde *breadcrumbs* die de huidige locatie aangeven en de huidige medewerker die in het systeem geselecteerd is (mits van toepassing). Via deze *breadcrumbs* is het ook mogelijk om terug te gaan in de navigatieboom. Voor de lay-out is een standaard SharePoint *style* gebruikt.

Technisch ontwerp

In het technisch ontwerp zullen we de technische achtergrond van het systeem uit de doeken doen. We beginnen met een globale opzet van de applicatie, waarna het datamodel wordt toegelicht, inclusief de inbedding in de onderliggende architectuur van ASP.NET. Tot slot beschrijven we de verschillende WebParts, workflows en rapportages die het systeem kent.

Opzet applicatie

De primaire gebruikers van de applicatie gebruiken een aparte *site* binnen SharePoint. De centrale pagina van deze site is een *dashboard* waar een verzameling van de belangrijkste en meest gebruikte WebParts is te vinden. Rondom dit dashboard zijn de overige pagina's van de applicatie opgebouwd. Aan de andere kant wordt het *Employee Self Service* gedeelte waarin de medewerker gegevens kan bekijken en wijzigen opgebouwd op zijn of haar persoonlijke MySite. Op deze manier is er een duidelijke scheiding aanwezig, wat de gebruiksvriendelijkheid ten goede komt.

Datamodel

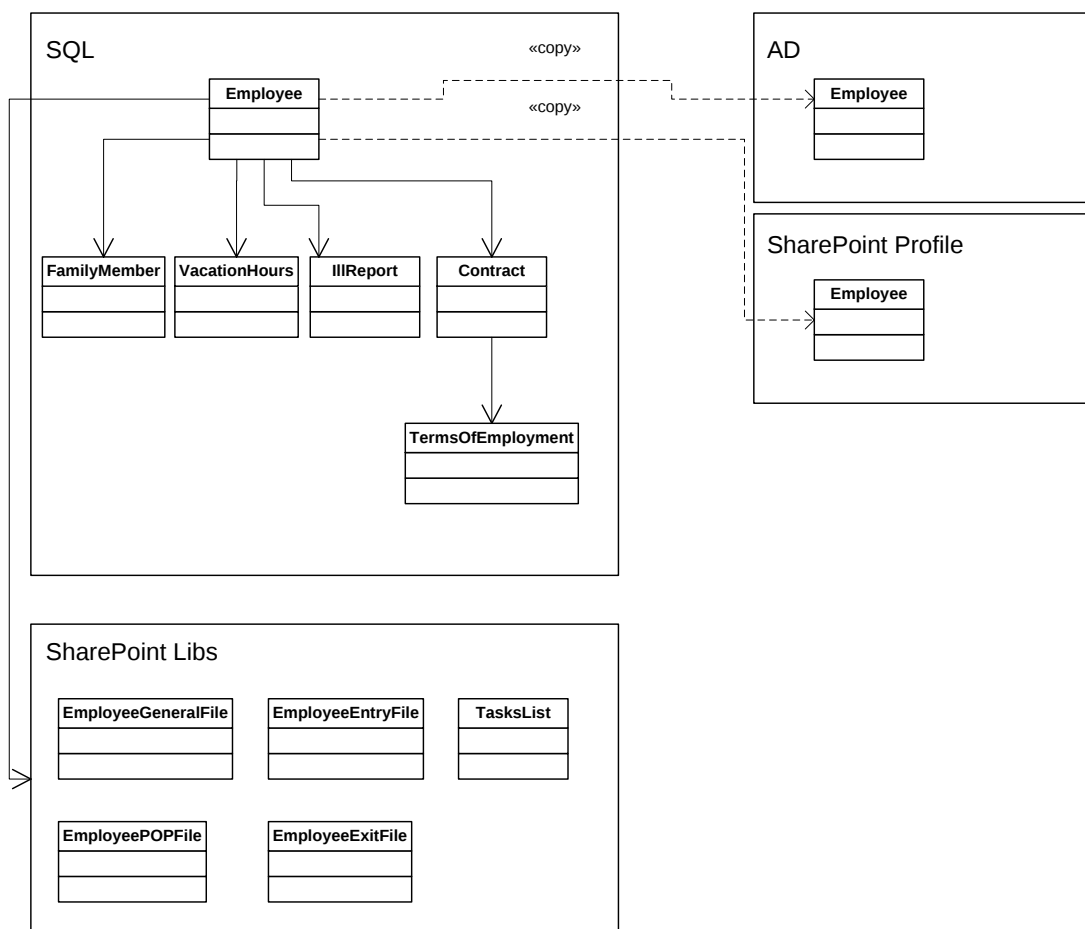
De opslag van data gebeurt binnen de applicatie op meerdere locaties. Dit omdat het soms handiger is om data in bijvoorbeeld SharePoint op te slaan om de integratie met de applicatie te vergroten, maar in andere gevallen is het handiger om data als relationele data in een database op te slaan. Tevens wordt sommige informatie gerepliceerd ten behoeve van backend systemen.

De relaties tussen de verschillende plaatsen van dataopslag zullen nu eerst globaal worden weergegeven, waarna dieper wordt ingegaan op enkele van de onderdelen. Tot slot worden de replicatie tussen verschillende databases en het beveiligingsmodel toegelicht.

Een beschrijving van de systeemarchitectuur en het objectmodel kan gevonden worden in Appendix E.

Globaal datamodel

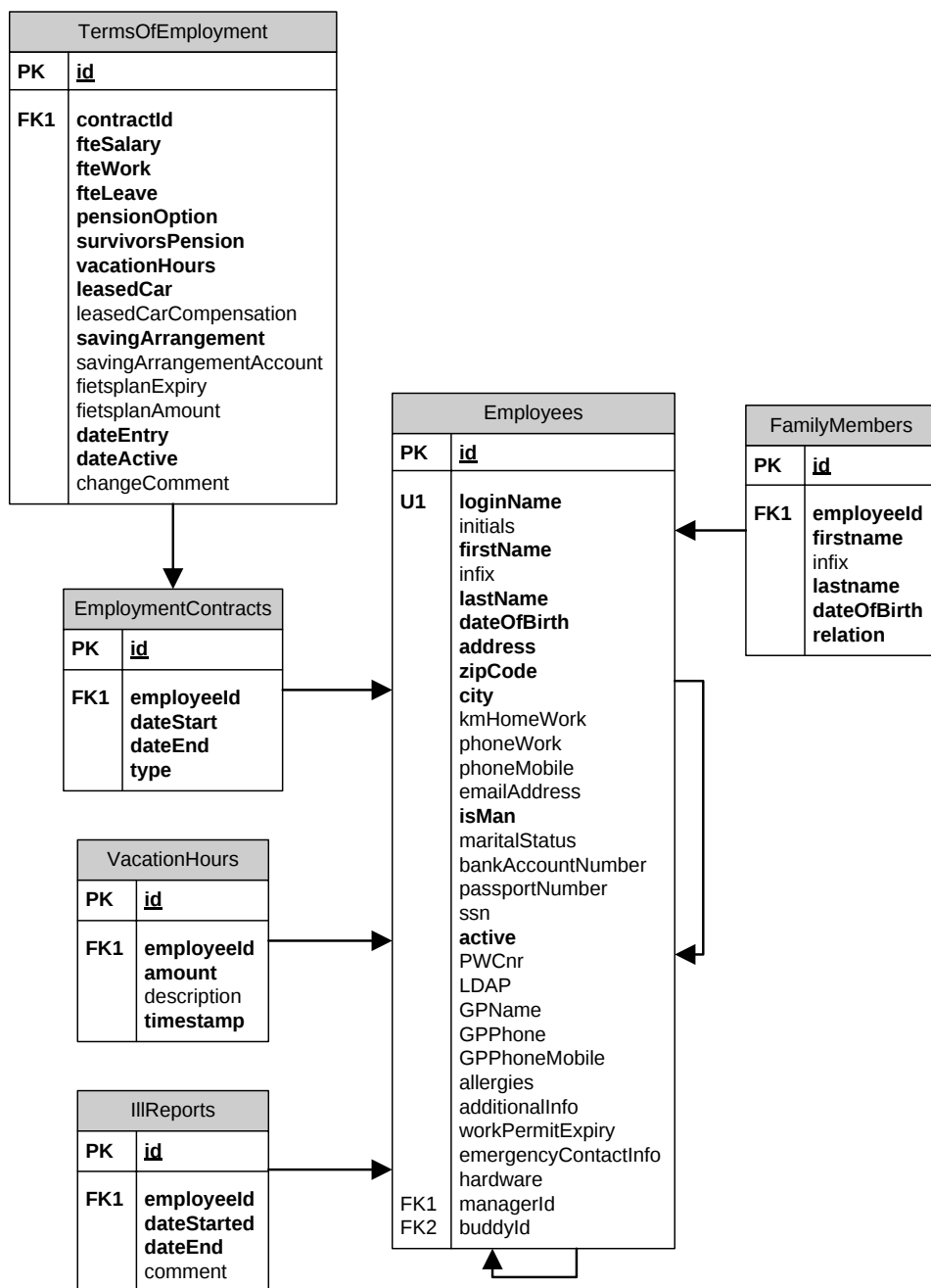
Onderstaand is een globale voorstelling van de dataopslag binnen de applicatie te vinden. Terug te zien is dat de basisdata behorende aan een Employee wordt opgeslagen in de SQL Server. Hieruit wordt vervolgens het één en ander gerepliceerd naar de Active Directory (voor de user accounts) en het SharePoint Profile van een user. Verder worden in de SQL Server de arbeidsvoorwaarden, vakantie-uren, ziektemeldingen en familieleden van werknemers opgeslagen. In SharePoint worden vervolgens de documenten (algemeen, in-dienst, uit-dienst en POP) alsmede de takenlijsten opgeslagen.



Figuur 4 - Globaal datamodel

SQL Server

Zoals al eerder aangegeven wordt een deel van de data in een MS SQL Server database opgeslagen. Hieronder is een weergave van de database te zien. De meeste velden spreken voor zich. De VacationHours tabel zal in dit geval alleen de correcties en toevoegingen aan de vakantie-uren bevatten, alle opgenomen vakantie-uren worden uit de bestaande urendatabase gehaald.



Figuur 5 - ERD diagram

SharePoint-componenten

Een *List* is een standaard onderdeel van SharePoint en is handig toe te passen vanwege de ingebouwde functionaliteit, waaronder bijvoorbeeld *alerts* die verstuurd worden bij wijzigingen in een lijst. Verder maakt het gebruik van SharePoint lists het gebruik van een eigen datamodel in deze gevallen overbodig.

De applicatie zal over vier document libraries beschikken die de formulieren voor de drie workflows bevatten en tevens de documenten van de medewerker opslaan. Om te zorgen dat een medewerker alleen

zijn eigen documenten ziet zullen er rechten op de documenten worden gezet en wordt er bij ieder document opgeslagen aan welke medewerker het document toebehoort.

Replicatie naar Active Directory en SharePoint profiles

Delen van de data uit de SQL database zullen gerepliceerd worden naar andere systemen. Deze replicatie kan gebeuren door middel van code in de verschillende WebParts of door middel van *triggers* die bij het updaten van data in de database de replicatie uitvoeren. Triggers genieten op dit moment de voorkeur gezien de databaseserver eigenlijk de plek is waar je de replicatie uit wilt voeren. Dit gezien de replicatie dan transparant voor de applicatie gebeurt en eventuele fouten in de replicatie ook direct tot het falen van de transactie zullen leiden, wat inconsistenties voorkomt.

De data die naar de Active Directory gerepliceerd moet worden zijn de inlog- en contactgegevens van medewerkers. De Active Directory wordt gebruikt door de verschillende Microsoft-producten om tegen te authenticeren. De meeste persoonlijke gegevens zullen naar de SharePoint profiles gerepliceerd moeten worden. Wat niet gerepliceerd moet worden betreft vertrouwelijke data zoals het sofinummer.

Authenticatie en autorisatie

Binnen het systeem is het van belang dat de toegang van de gebruikers beperkt wordt. Dit omdat bijvoorbeeld de salarissen worden opgeslagen in het systeem, maar ook omdat er persoonlijke informatie over de medewerkers, zoals de bankrekeningnummers, in het systeem staat.

De authenticatie zal plaatsvinden door de standaard SharePoint portal-mogelijkheden te gebruiken. De portal authenticceert op dit moment door middel van de in IIS ingebouwde HTTP authenticatie bij het TAMTAM domein. De gebruikers krijgen vervolgens door middel van “impersonation” van het webserverproces ook de lokale rechten waar zij over beschikken.

Hoe de autorisatie plaatsvindt is afhankelijk van de plaats waar de data in het systeem is opgeslagen. Een deel van de data wordt namelijk opgeslagen in een MS SQL database en een deel van de data wordt opgeslagen in standaard SharePoint libraries en lists.

De data die is opgeslagen in de SharePoint libraries en lists zal door middel van de standaard SharePoint permissiemogelijkheden worden beveiligd. Hierbij kan op itemniveau worden aangegeven welke rechten een user erop heeft, waarna de gebruikers zonder rechten op het item het item ook niet meer te zien krijgen (*security-trimmed*).

De in MS SQL Server opslagen data zal worden afgeschermd in de WebParts die toegang hebben tot de database. Dit omdat beveiliging op itemniveau in de database weliswaar mogelijk is, maar erg omslachtig zou werken en het gebruikelijker is om deze autorisatie in de applicatie te laten plaatsvinden. Ieder WebPart zal er zorg voor dragen dat een medewerker altijd alleen zijn eigen gegevens kan zien en dat unit managers alleen gegevens van hun eigen unit kunnen opvragen wanneer dit zinvol is.

WebParts

In het functioneel ontwerp zijn de verschillende WebParts al besproken. We zullen hier kort ingaan op de manier waarop deze geïmplementeerd worden en hoe de standaard WebParts van SharePoint gebruikt zullen worden.

De WebParts zullen geïmplementeerd worden als SmartPart. SmartPart is een webpart dat de mogelijkheid heeft om elk ASP.NET *user control* weer te geven en biedt tevens ondersteuning voor ASP.NET AJAX. Hierdoor kan er grafisch ontwikkeld en gedebugged worden, waarna de user control als WebPart te gebruiken is. Alle WebParts zullen ontwikkeld worden als ASP.NET User Control binnen een Web Site Project in Visual Studio 2005. Doorgeven van informatie tussen WebParts zal gebeuren door middel van de *querystring* in het geval van paginaoproepen en door middel van *postbacks* (bij voorkeur via ASP.NET AJAX) wanneer er data opgeslagen moet worden.

De standaard list-, task- en documentlibrary view WebParts zullen ook veelvuldig gebruikt worden binnen de applicatie. Deze worden gebruikt om een view te geven van de data welke aanwezig is binnen de SharePoint omgeving, zoals de POP formulieren of de takenlijsten. Door middel van custom views zorgen we ervoor dat alleen de informatie die voor de betreffende pagina relevant is aan de gebruiker getoond wordt. Kolommen van de gebruikte InfoPath formulieren zullen ook geëxporteerd worden binnen de views, zodat hier eenvoudig op te sorteren is de informatie ook eenvoudig doorzoekbaar is te maken.

De verschillende WebParts waaruit het systeem zal bestaan zijn in Appendix D opgesomd en kort toegelicht.

Workflows

Zoals reeds in het functioneel ontwerp besproken zullen er drie workflows, gebaseerd op Windows Workflow Foundation, gebruikt worden om het POP proces en de in- en uitdiensttreding te begeleiden. Binnen deze workflow zal van een *custom activity* gebruik gemaakt worden welke de gebruikersinformatie uit de SQL database haalt en vervolgens opslaat binnen de workflow, zodat de andere activiteiten hier eenvoudig over beschikken.

Taken welke aangemaakt worden binnen de workflows zullen in een globale tasklist worden gezet waardoor de gebruiker eenvoudig een overzicht heeft welke taken hij of zij nog te doen heeft. Deze taken zullen hangen aan *domain groups* en niet aan individuele gebruikers, waardoor het afhandelen van taken geen problemen oplevert wanneer bijvoorbeeld een medewerker ziek is. Een uitzondering hierop zijn de taken welke echt door een individuele medewerker gedaan moeten worden, zoals deze voorkomen bij indiensttreding van een medewerker.

Wanneer mogelijk zullen zoveel mogelijk taken tegelijk parallel open staan, zodat meerdere gebruikers aan een workflow kunnen werken en gebruikers niet onnodig op elkaar hoeven te wachten. Dit zal worden gerealiseerd door taken in de workflow direct achter elkaar te creëren en vervolgens te wachten totdat alle taken afgerond zijn voordat de workflow verder gaat.

De verschillende Workflows waaruit het systeem zal bestaan zijn in Appendix D opgesomd en kort toegelicht.

Rapportages

Voor de rapportages zou in eerste instantie gebruik worden gemaakt van Microsoft Excel Services. Dit omdat daardoor eenvoudig, zonder al te veel code, flexibele en krachtige rapportages te maken waren waar gebruikers eventueel zelf nog verder mee kunnen werken binnen Excel. Er zouden standaard Excel sheets ontworpen worden voor de verschillende rapportages, welke de beschikking hadden over een *trusted data source* om de benodigde data uit de SQL database te halen. Over deze data zouden enkele *pivot tables* gemaakt worden welke de benodigde rapportage geven.



De gemaakte Excel sheets zouden vervolgens via Excel Services gepubliceerd worden in de HRM applicatie, waardoor een gebruiker deze kon opvragen. Bij het opvragen zouden de Excel Calculation Services er zorg voor dragen dat de data uit de database werd gehaald waarna deze aan de gebruiker kon worden getoond via de webinterface. Op verzoek kon de gebruiker een Excel sheet downloaden om hier verder mee te werken.

Tijdens de implementatie bleek echter dat Excel Services niet de juiste capaciteiten had voor ons systeem. Zoals we tijdens de bespreking van de implementatie in het volgende hoofdstuk zullen zien hebben we daarom gekozen voor een andere aanpak.

6 Implementatie

Tijdens de vier weken durende implementatiefase zijn de eerder vastgestelde requirements ontwikkeld in code. We hebben aan het begin van deze fase een nadere afbakening gemaakt van welke requirements daadwerkelijk geïmplementeerd zouden worden. Het gaat hier om de requirementsgroepen 1 tot en met 3, aangezien de verwachting was dat we voldoende tijd hadden om deze requirements binnen de planning te implementeren. Het resultaat van deze fase is een werkend systeem dat op de testomgeving draait.

We beschrijven in dit hoofdstuk eerst onze aanpak van de implementatiefase. Vervolgens beschrijven we het opbouwen van de testportal. Daarna bespreken we de implementatie van respectievelijk de WebParts waaruit het systeem bestaat, de rapportages en de security. In deze gedeelten noemen we de uitdagingen die we tegenkwamen, de problemen die opdoken en de manier waarom wij deze problemen hebben opgelost.

Aanpak

Het ontwikkelen van software gebeurt bij Tam Tam vrijwel altijd in een *virtual machine* op de lokale pc onder Microsoft Virtual PC. Er is een standaard image beschikbaar met daarop een complete ontwikkelomgeving onder Windows 2003, met daarop uiteraard ook SharePoint geïnstalleerd. Ook wij hebben deze omgeving gebruikt voor de bouwen van onze software. Naast lokale virtual machines hadden we de beschikking over een *virtual server*, waarop wij onze test-deployment konden doen. Op deze server was SharePoint geïnstalleerd en hebben wij onze volledige testportal opgebouwd. WebParts werden allereerst op de lokale virtual machine ontwikkeld en getest alvorens ze werden gedeployed naar de testserver.

Voor versiebeheer en als centrale *repository* van onze code is SourceGear Vault gebruikt, een product dat volledig integreert in Visual Studio 2005 en standaard binnen Tam Tam gebruikt wordt voor versiebeheer. Door middel van Vault was het mogelijk om altijd de beschikking te hebben over de meest recente code om in te ontwikkelen en om eenvoudig bestanden te *mergen*, met als gevolg dat groepswerk weer een stukje makkelijker werd.

Opbouw testportal

Allereerst is begonnen met de implementatie van het grafisch ontwerp in de testportal door alle op papier ontwikkelde pagina's als dummy te implementeren. Hier werden ook alle WebParts als dummy opgebouwd, zodat ook HR enig zicht had op de opbouw van de applicatie in verschillende schermen. Hiermee is ook de eerste gebruikerstest uitgevoerd waarin we gekeken hebben of alle velden in de formulieren aanwezig waren en of de opbouw van de applicatie logisch was. Hier werden wat kleine foutjes gevonden en kwamen we tot de conclusie dat enkele veldjes misten; iets wat snel opgelost kon worden.

Custom menu

Bij de navigatie kwamen we het probleem tegen dat het onmogelijk is om elementen in een SharePoint menu dieper dan één niveau te nesten. Dit probleem was op te lossen door volledige stukken code in SharePoint te *overriden*, of door in de *masterpage* van de HRM site het menu aan te passen en te vervangen door een SmartPart met custom menu erin. We hebben voor het laatste gekozen, aangezien het eerste veel werk zou kosten en de tweede oplossing de meeste flexibiliteit bood.

Het door ons gemaakte menu bestaat uit platte HTML en JavaScript. Tevens is in de *masterpage* wat JavaScript gebruikt om onder de *breadcrumbs* aan te geven welke *employee* op dat moment geselecteerd is (mits van toepassing). Deze oplossing is simpel maar doeltreffend en kost weinig tijd.

Implementatie WebParts

Toen de testportal eenmaal draaide en was opgebouwd, zijn we begonnen met de implementatie van de verschillende WebParts. De problemen die we hierbij tegenkwamen worden hier kort besproken, alsmede de manier waarop we deze problemen hebben opgelost.

Contracten

In eerste instantie wilden we met contracten op de volgende manier omgaan: een verzameling arbeidsvoorwaarden zou tot een nieuw contract gaan behoren indien het type arbeidsrelatie veranderde (bijvoorbeeld van stagiair naar een jaarcontract). Gaandeweg werd echter duidelijk dat er behoefte was aan een meer expliciete manier van omgaan met contracten. We hebben daarom besloten om het WebPart waarin de arbeidsvoorwaarden worden bijgehouden uit te breiden met een stukje contractbeheer. Dit heeft geleid tot het maken van onderscheid tussen arbeidsvoorwaarden wijzigen en arbeidsvoorwaarden toevoegen als een nieuw contract.

Beheer-WebParts

Voor enkele velden in het systeem geldt dat de waarde een selectie uit verschillende mogelijke waarden is. Voor deze lijsten met mogelijke waarden was in het oorspronkelijke ontwerp nog geen beheermogelijkheid toegevoegd. Om het gebruikersgemak te maximaliseren en er tevens voor te zorgen de drempel voor het up-to-date houden van alle waarden, hebben we besloten hier toch nog enkele WebParts voor te maken. We hebben dit gedaan voor:

- Units en unit managers
- Baarda-rollen
- Baarda-leerwegen
- Pension contribution-opties

Takenlijsten

De *views* van de gebruikte takenlijsten zijn aangepast zodat alleen relevante en nog openstaande taken worden getoond. Tevens is het datumformaat in deze *views* aangepast aan het door ons gebruikte datumformaat. Dit was alleen mogelijk door de view aan te passen in SharePoint designer en deze te converteren naar een XSLT view. De beveiliging van de takenlijsten is zo ingesteld dat synchronisatie met Outlook en het toevoegen van *alerts* wel mogelijk is, maar bijvoorbeeld niet het toevoegen van taken.

Ondersteunende klassen

Zoals eerder genoemd bestaat de applicatie uit een aantal UserControls, wat op zichzelf staande componenten zijn. Om dubbele code te voorkomen zijn er een aantal algemene klassen aangemaakt, welke in de App_Code directory van het project zijn geplaatst. Dit is een directory met code die vervolgens door het gehele project beschikbaar is. Er zijn hier drie klassen aangemaakt, welke we nu kort zullen beschrijven.

- HRMUserControl: Deze klasse is de superklasse van alle UserControls. In de constructor van deze klasse wordt een *init eventhandler* toegevoegd welke enkele variabelen op een uniforme manier

door de gehele applicatie instelt. Deze code kon overigens niet direct in de constructor omdat tijdens het construeren van de verschillende objecten bepaalde informatie (zoals het Request object) nog niet beschikbaar is.

- **HRMUtils:** Na een tijd programmeren hadden we behoefte aan een klasse om algemene, generieke functies in te zetten, welke verder geen object state vereisen. Hiervoor hebben we de static class HRMUtils in het leven geroepen, welke enkele static methodes bevat die door de gehele applicatie direct aangeroepen kunnen worden. Hierbij moet worden gedacht aan methodes om informatie uit de database op te vragen of om een email te sturen.
- **HRMSecurity:** Deze klasse wordt door de HRMUserControl geïnstantieerd en biedt de verschillende klassen de mogelijkheid om te controleren over welke rechten de ingelogde medewerker beschikt. Zo kan de klasse kijken of de medewerker van de HR afdeling is of dat het de unit manager is van de persoon welke hij probeert te bekijken. Meer hierover onder het kopje Security.

Rapportages

Zoals in de requirements naar voren gekomen moest de applicatie de mogelijkheid bieden tot het creëren van rapportages. Voor het genereren van deze rapportages hadden we twee opties: Excel of Reporting Services. De uiteindelijk uitwerking van deze keuze is pas tijdens de implementatie gemaakt, vandaar dat dit hier besproken wordt.

Excel is het bekende spreadsheet programma van Microsoft. Veel van de huidige rapportages zijn in dit formaat, gezien het door middel van draaitabellen eenvoudig is eigen rapportages in elkaar te zetten. Met behulp van Excel Web Access en Excel Calculation Services is het zelfs mogelijk om een Excel sheet direct in een SharePoint site weer te geven, waarbij de cliënt geen Excel nodig heeft om de sheet te kunnen bekijken. Een nadeel van Excel sheets is dat ze slecht te beveiligen zijn. Wanneer data uit een database wordt gehaald valt informatie zoals de *connection string* niet te beveiligen, wat zou betekenen dat alle data die op de één of andere manier via een Excel rapportage beschikbaar moest zijn voor iedereen toegankelijk zou zijn. Wanneer je gegevens op itemniveau wilt beveiligen blijkt Excel om deze reden geen optie.

Toen Excel in de proof of concept fase geen oplossing bleek te bieden zijn we verder gaan zoeken. We zijn toen uitgekomen op Microsoft SQL Server Reporting Services. Sinds het laatste Service Pack blijkt deze minstens net zo goed te integreren in SharePoint als Excel, maar biedt het veel meer mogelijkheden. De security valt binnen Reporting Services ook prima te regelen, gezien het mogelijk is eigen programmacode te gebruiken in je rapportages. Een nadeel van Reporting Services is wel dat het aanmaken van rapportages wat lastiger is dan bij Excel. Ondanks deze reden waren de voordelen op security gebied dusdanig dat we toch voor Reporting Services hebben gekozen.

Security

Beveiliging speelt binnen een HRM applicatie een grote rol; wanneer persoonlijke of financiële gegevens van de medewerkers van een bedrijf op straat komen te liggen is dit een groot probleem. Om deze reden is reeds tijdens de requirements fase met de HR afdeling overlegd over de inrichting van de beveiliging van het systeem. Hieruit werd duidelijk welke gebruikers welke rechten in het systeem dienden te hebben. Vervolgens is hier een bijpassend security model bij bedacht, wat in detail is uitgewerkt tijdens deze implementatiefase.

Een probleem bij het implementeren waren de verschillende applicaties en servers waar de HRM applicatie op moest komen te draaien. Naast de SharePoint server waarbinnen pagina's werden aangemaakt, waren er nog de UserControls welke binnen de pagina draaiden, de Reporting Server en de SQL Server. Op alle niveaus is beveiliging noodzakelijk, gezien één onbeveiligd niveau zou betekenen dat een ongeautoriseerde gebruiker via dat niveau toegang zou kunnen krijgen tot het systeem en dus tot persoonlijke gegevens. We zullen nu één voor één bespreken welke problemen we tegenkwamen tijdens het inrichten van de beveiliging op deze verschillende niveaus.

SharePoint

SharePoint kent een vrij goed rechtensysteem waarbij op lijsten per item de beveiliging is in te stellen. Gezien pagina's ook gewoon items in een list zijn (in een Pages library om precies te zijn) viel hiermee eenvoudig in te stellen dat alleen bepaalde groepen de pagina's van het HRM systeem mochten zien. De groepen welke vervolgens rechten kregen waren de superusers, HRM, finance, system administrators en unit managers groep, waarbij de eerste vier direct verwezen naar bijbehorende groepen in de Active Directory en de laatste groep dynamisch werd aangepast aan de hand van de huidige unit managers. Dit laatste zodat de HR afdeling zelf de unit managers kon beheren, zonder toegang tot de Active Directory nodig te hebben.

UserControls

Met alleen een beveiliging van de SharePoint pagina's ben je er echter nog niet. Gezien we hadden besloten gebruik te gaan maken van UserControls welke als SmartPart op de SharePoint portal werden geplaatst, werden we met het probleem geconfronteerd dat iedereen altijd nog zelf een bepaalde UserControl op zijn pagina kon zetten. Verder moesten de rechten naast van de ingelogde medewerker, ook van de te bekijken medewerker afhangen. Om deze redenen moest ook op het niveau van de UserControls beveiliging toegepast worden.

Voor de beveiliging van de UserControls hebben we allereerst gekeken naar de standaard ASP.NET security. Deze beveiliging voorziet in gebruikers, groepen en rollen. Vervolgens is het mogelijk bepaalde rollen slechts toegang te geven tot bepaalde pagina's. Deze rechten kunnen nog verder doorgevoerd worden, waarna het mogelijk is om bepaalde rollen slechts toegang te geven tot bepaalde methoden in de klassen van je applicatie. Dit was echter niet genoeg voor onze wensen, tenslotte moesten wij rechten kunnen toekennen van gebruikers op andere gebruikers. Zo mag een unit manager alleen zijn eigen medewerkers bekijken.

Gezien de ASP.NET security afviel zijn we verder gaan zoeken naar een andere oplossing. We hebben toen twee oplossingen bedacht. De ene oplossing was het creëren van een eigen security framework, waarin we rollen en rechten zouden definiëren, en vervolgens de mogelijkheid zouden bieden om rollen en groepen slechts rechten te geven op bepaalde medewerkers. Dit zou een zeer flexibel, maar complex systeem opleveren. Gezien de beperkte tijd en de security niet zozeer flexibel, maar wel goed getest moest zijn, hebben we gekozen voor een eenvoudigere oplossing.

Voor de beveiliging van de UserControls is uiteindelijk gekozen voor het maken van een extra klasse welke enkele eenvoudige security checks zou uitvoeren; de klasse HRMSecurity. Deze klasse controleert of de gebruiker tot één van de volgende groepen behoort: superusers, HRM, Finance, system administrators, unit manager van de opgevraagde medewerker of dat de user de opgevraagde medewerker zelf is. De losse

UserControls roepen vervolgens deze klasse aan en aan de hand van de uitkomst worden de mogelijkheden op de pagina's aangepast.

Reporting Server

De beveiliging van de Reporting Server was een uitdaging op zich. Gezien de Reporting Server overzichten moet kunnen genereren waarin meerdere medewerkers terugkomen en direct SQL uitvoert op de SQL Server, lag het voor de hand om de SQL query aan te passen aan de hand van de ingelogde gebruiker. Uiteindelijk is er voor een oplossing gekozen waarbij een aparte DLL in de Reporting Server wordt geladen, welke een INNER JOIN genereert die aan de query toegevoegd kan worden. Afhankelijk van de ingelogde gebruiker (welke door middel van integrated Windows security wordt doorgegeven), is de toe te voegen query leeg (voor superusers, finance en HR), of bevat hij slechts de medewerkers welke gezien mogen worden (voor unit managers).

SQL Server

Een probleem met de beveiliging van de toegang tot de SQL Server was het feit dat binnen Tam Tam veel medewerkers toegang hebben tot de verschillende databaseservers. Gezien individuele rechten op databaseniveau binnen de huidige administratie database nog weinig wordt toegepast en veel mensen volledige toegang tot deze database hebben, is er in overleg met systeembeheer voor gekozen om een aparte databaseserver op te zetten. Deze server draait als aparte MS SQL *instance* en is zo tamelijk afgeschermd van de overige gebruikers. De in de administratiedatabase benodigde data wordt via een *linked server connectie* en een *view* uit de HRM database gehaald. Dit gebeurt met een beperkt useraccount, zodat ook via dit useraccount geen volledige rechten op de HRM database verkregen kunnen worden. Tot slot is ervoor gekozen de *connection strings*, die de connectiegegevens naar de database bevatten, versleuteld op te slaan. Hiertoe is gebruik gemaakt van de ingebouwde ASP.NET ondersteuning voor DPAPI (Data Protection API).

7 Testen

In dit hoofdstuk zullen we het testtraject behorende bij dit project bespreken. We beginnen met het uiteenzetten van onze aanpak waarna de verschillende testcases besproken worden. Tot slot bespreken we de gevonden kwesties en de manier waarop we deze opgelost hebben.

Aanpak

Door het gehele traject heen is er veelvuldig getest, wat voor ons erg belangrijk was voor de kwaliteitsbewaking. Tijdens de ontwerpfase zijn *proofs of concept* gebruikt om te testen of de door ons gekozen aanpak ook daadwerkelijk een goede was. Ook is hier het grafisch ontwerp getest door middel van een *papieren prototype* waar de lay-out van de verschillende schermen al door de eindgebruiker werd beoordeeld.

Tijdens de implementatie is alle losstaande code getest en is het grafisch ontwerp teruggekoppeld door middel van een *computer prototype test*. Bij de laatste test waren de schermen al opgebouwd en 'klikbaar', maar hing er nog geen functionaliteit aan. Input vanuit de eindgebruiker op dit prototype was ook hier erg van belang.

Na de implementatie is er een integratietest gedaan waarbij onderstaande testcases zijn uitgevoerd. Hierna is een acceptatietest uitgevoerd waarbij de knoop werd doorgelicht over het al dan niet in productie nemen van het systeem.

Kwaliteitsbewaking

Door tijdens het gehele traject veelvuldig te testen hebben we getracht om de kwaliteit van het geleverde werk zo hoog mogelijk te houden. Een continue terugkoppeling vanuit zowel eindgebruikers als groepsgenoten in het ontwikkelproces werd als zeer prettig ervaren door alle belanghebbenden en zorgde zowel voor een nuttige bijdrage aan het systeem alsmede voor een toename van *awareness* bij de eindgebruikers.

Testcases

De uiteindelijke integratietest bestond uit een groot scala aan niet-geautomatiseerde tests die zijn losgelaten op het systeem in productieomgeving, waar echter nog niet definitief was overgeschakeld naar het live systeem. We maakten hier gebruik van een database met een set gegevens, overgenomen uit de productieomgeving, zodat we konden testen met semi-live data. We hebben de verschillende testcases verdeeld in een zevental groepen die we nu zullen behandelen.

Basis features

In dit onderdeel is de werking van de formulieren getest. Er is getest of de formulieren inderdaad hun gegevens correct in de database opslaan en of de validatie vooraf en de foutmeldingen naar de gebruiker consistent en duidelijk zijn. De los gebouwde WebParts bestaan voornamelijk uit eenvoudige formulieren, welke vervolgens wel koppelingen hebben met andere (sub)systemen. Een voorbeeld hiervan is de authenticatie welke direct gekoppeld is aan de Active Directory door middel van *integrated Windows authentication*.

Rapportages

Binnen het systeem kunnen verschillende rapportages worden gegenereerd. Deze rapportages zijn gecontroleerd op hun correcte werking en de juiste uitvoer. Deze tests zijn uitgevoerd in de development omgeving, aangezien het voor de uiteindelijke integratietest niet mogelijk was om SQL Reporting Services werkende te krijgen en hier de hulp van systeembeheer vereist was.

WebPart connections

Er bestaan verschillende koppelingen tussen WebParts waarbij het testen van WebParts op zichzelf niet erg zinvol was. De basale werking van ieder Web Part is namelijk triviaal (een formulier welke naar een database post), de lastige punten zitten hem met name in de integratie met de verschillende systemen. Deze connecties tussen WebParts zijn bij de vakantiedagenberekening en de weergave van de vakantiedagen getest, waarbij de WebParts dus onderling moeten communiceren om tot een juiste weergave van data te komen.

In- en uitdiensttreding scenario's

De basale werking van de WebParts die verantwoordelijk zijn voor in- en uitdiensttreding is reeds getest bij de *basis features*, maar de verschillende scenario's zijn hier nogmaals uitvoerig getest. Het gaat hier bijvoorbeeld om standaardscenario's waar medewerkers in dienst komen, een nieuwe set arbeidsvoorwaarden krijgen of worden ontslagen, maar ook om scenario's waar medewerkers na een eerder ontslag weer in dienst komen of nieuwe arbeidsvoorwaarden krijgen waarbij er in de toekomst al nieuwe arbeidsvoorwaarden zijn ingegeven. We hebben hier vooral de consistentie van de date in ogenschouw genomen.

Berekening vakantiedagen

De berekening van vakantiedagen is afhankelijk van vele variabelen en is derhalve uitgebreid getest. Tijdens de integratietests is ook nog ingegaan op scenario's waarbij een medewerker halverwege een maand in dienst komt, de maand december pas mag worden uitgerekend nadat de overige maanden in het jaar uitgerekend zijn en de beveiliging tegen het tweemaal uitrekenen van vakantiedagen.

Notificaties en tasks

Bij deze tests zijn zowel de geautomatiseerde als actieafhankelijke notificaties getest. Voor de geautomatiseerde notificaties zijn twee testscenario's gemaakt om de grenswaarden van de invoer te testen. Hierna hebben we het script uitgevoerd dat de notificaties genereert en gekeken of de uitvoer correct was. Voor de actieafhankelijke notificaties zijn de bijpassende acties uitgevoerd waarna gekeken werd of de notificatie gestuurd werd. Bij alle tests is ook gecontroleerd of de juiste persoon of groep personen de notificatie krijgt. Ook is de koppeling van takenlijsten met Outlook getest.

Beveiliging

Voor de beveiliging van het systeem was het ondermeer nodig om de units en bijbehorende unitmanagers bij te houden in het systeem, waarbij dit zowel in SharePoint als in de database gebeurde. Hiervoor zijn een aantal triviale testcases uitgevoerd, waarbij onder andere units werden aangemaakt en unitmanagers werden toegevoegd en verwijderd. Hierbij werd goed gekeken of de wijzigingen zowel in de database en SharePoint werden doorgevoerd. Ook is per groep gebruikers getest of de rechten op de WebParts en de pagina's in de portal goed waren ingesteld.

Issues en oplossingen

Door tijdens de loop van het project regelmatig te testen kwamen bugs al vrij snel aan het licht. We zullen hieronder kort de resultaten van de testfasen beschrijven en eventuele issues die hierbij opdoken bespreken.

Interfacetest

De vele uitgevoerde tests met de eindgebruikers zijn altijd constructief gelopen waarbij men al snel erg enthousiast werd over het systeem. De issues die naar boven kwamen tijdens de tests bleken nooit onoverkomelijk en zorgden meer en meer voor duidelijkheid over het te bouwen systeem.

Door in een vroegtijdig stadium al met papieren prototypes te beginnen kwamen problemen al snel bovendien. Hier en daar bleek de lay-out niet consistent, was de interface niet handig te werken of bleken er veldjes te ontbreken. Door een actieve houding van de eindgebruikers in deze tests was het mogelijk om deze problemen vroegtijdig af te vangen. Tijdens de tests zijn hier dan ook geen noemenswaardige issues opgedoken.

Integratietest

Tijdens de integratietest kwamen nog vele kleine bugs naar voren, welke vrijwel allemaal snel opgelost konden worden. Enkele grote issues zullen we nu bespreken.

Het grootste probleem dat we tegenkwamen was het goed instellen van de rechten. De meeste gebruikersgroepen bleken op sommige pagina's te weinig of juist teveel rechten te hebben, waardoor het systeem onjuist functioneerde voor deze gebruikers. Een onvoorziene *restart* van IIS bleek nog wat roet in het eten te gooien waardoor alle rechten opnieuw ingesteld dienden te worden. Ook bleken de rechten hier en daar in de WebParts nog niet goed te staan, waardoor er knoppen ontbraken of acties mogelijk waren waar men geen rechten toe had. Deze instellingen zijn gecorrigeerd en nogmaals uitvoerig getest is.

Een mogelijkheid van SharePoint is om *alerts* aan te zetten bij takenlijstjes waardoor je automatisch een melding krijgt per email als een taak is toegevoegd, gewijzigd of verwijderd. Deze functionaliteit werkte prima, maar helaas bleek het formaat waarin de alerts werden gestuurd niet door Exchange 2003 begrepen te worden. In de mailbox was hierdoor alleen het onderwerp van de alert te zien, maar niet de tekst in de email. De oplossing voor dit probleem is het installeren van een *hotfix* voor Exchange 2003 of een migratie naar Exchange 2007, waarin het probleem waarschijnlijk wel is opgelost. Daar de voorbereidingen voor een migratie al in gang zijn gezet verwachten we dat dit probleem binnenkort is opgelost. Tot die tijd is er een verzoek ingediend bij de systeembeheerders om de *hotfix* te installeren, waardoor de alerts wel gaan werken.

Acceptatietest

De acceptatietest bestond uit een korte demonstratie van het systeem met semi-live data en een korte doorloop. Ook hier werden nog een aantal kleine bugs gevonden welke snel verholpen konden worden. Het enige grote probleem bleek de administratie van oproepkrachten te zijn. Dit contracttype was na de vele gesprekken, tests en doorlopen van het systeem nog niet naar voren gekomen en was zodoende ook niet in het ontwerp meegenomen. Een oproepkracht is anders dan andere medewerkers in het opzicht dat hij geen vast salaris en vakantiedagen krijgt, maar per uur betaald wordt.



Na afloop van de test is besloten om dit probleem later op te lossen. Ook is toen besloten om het systeem in gebruik te nemen na het oplossen van de gevonden kleine bugs.

8 Deployment

Na weken werken was het eindelijk tijd voor de uitrol van onze oplossing in de organisatie. In dit hoofdstuk bespreken we de door ons gekozen aanpak en de problemen die we tegenkwamen tijdens de uitrol.

Aanpak

Het uitrollen van de HRM applicatie naar de live portal kwam neer op het volledig nabouwen van de testomgeving in de live omgeving. Er zijn manieren om SharePoint sites in het geheel over te zetten, maar handmatig overzetten zou waarschijnlijk sneller zijn. Tevens hadden we op deze manier de controle en konden we het systeem gelijk 'netjes' maken.

Besloten is om al relatief vroeg een eigen *site* aan te (laten) maken op de portal waarin de uitrol kon plaatsvinden. Aangezien we gebruik maakten van een losstaande database was het mogelijk om de portal rustig op te bouwen en te testen. Hiervoor dienden echter wel SmartPart v1.2, ASP.NET AJAX en de ASP.AJAX toolkit geïnstalleerd te worden op de live portal. Ook diende ondersteuning voor SQL Server 2005 Reporting Services geïnstalleerd te worden.

Nadat de hele portal was opgebouwd zijn de integratie- en acceptatietests uitgevoerd. Ook zijn voorbereidingen getroffen om over te schakelen naar een live situatie. Hiervoor moest de huidige administratiedatabase voor een deel vervangen worden door onze data. Hiervoor is besloten om een *view* te maken van onze medewerkers-tabel die er hetzelfde uitziet als de oude medewerkers-tabel. Op deze manier zouden applicaties die deze tabel ook gebruiken moeten blijven werken, gezien zij nu nog alle data kunnen uitlezen.

Hierna is er overgeschakeld naar de live situatie waarna onze applicatie en de verschillende systemen die gebruik maken van de *view* zijn getest op hun werking.

Mogelijke problemen

Bij de overschakeling naar de live situatie hebben we rekening gehouden met mogelijke problemen die zouden kunnen optreden. Het belangrijkste waren eventuele afwijkingen in de configuratie en werking van de live portal, waardoor onze applicatie niet correct zou werken. Ook de conversie van de huidige data kon nog problemen opleveren, waarbij onze applicatie zich zou verslikken in de geconverteerde data. Nog belangrijker was de correcte werking van andere systemen binnen Tam Tam die gebruik zouden gaan maken van onze *view* op de medewerkergegevens; iets waarvan vrijwel zeker was dat het niet geheel meer zou functioneren. Op voorhand was al duidelijk dat het oplossen van problemen in andere applicaties nog het meeste werk zou zijn, gezien slechts weinig mensen goed zicht hebben op de werking van het huidige systeem.

Kwaliteitsbewaking

Om zoveel mogelijk problemen te voorkomen hebben we voor de overgang naar het live systeem erg goed nagedacht over problemen die eventueel zouden kunnen optreden en de manier waarop we deze zouden oplossen. Tevens hebben we gezorgd voor correcte back-ups van het huidige systeem, voor het geval alles echt mis mocht gaan. Om een en ander te coördineren hebben we al in een vroeg stadium een planning gemaakt voor de uitrol, waarbij ook rekening gehouden is met afhankelijkheden, zoals de installatie van software door systeembeheerders.

Om de overgang zo soepel mogelijk te laten verlopen en om niemand tot last te zijn, zijn de kritieke fases van de uitrol 's avonds uitgevoerd. Ook hebben we van tevoren aan alle medewerkers een email gestuurd met de aankondiging dat er onderhoud zou gaan plaatsvinden. Achteraf hebben we naar alle kritieke gebruikers van systemen (bijvoorbeeld de Servicedesk) een email gestuurd met de mededeling dat er wijzigingen doorgevoerd waren. Op deze manier was iedereen op de hoogte en konden wij snel handelen op het moment er toch iets fout ging in applicaties.

Procedure

Alvorens met de uitrol te kunnen beginnen dienden er een aantal applicaties geïnstalleerd te worden op de portal. Het betrof hier ASP.NET AJAX, SmartPart v1.2 bèta, de AJAX Control Toolkit en een SharePoint add-in voor SQL Server Reporting Services 2005 (SP2). Hierbij werden ook een aantal wijzigingen in de configuratie van de portal doorgevoerd.

Nadat deze voorbereidingen waren getroffen was de echte uitrol mogelijk. We hebben overdag alle pagina's opgebouwd, de *masterpage* aangepast voor het *custom* menu en alle WebParts geplaatst. Tevens is alle *security* toegepast op de verschillende SharePoint pagina's. Ook het conversiescript voor de oude data is geschreven en getest, zodat wij ook de overige tests met semi-live data konden uitvoeren.

Nadat het hele systeem draaide hebben de uiteindelijke integratie- en acceptatietests plaatsgevonden. Hierna is wederom op een avond de uiteindelijke conversie van data gestart en is de tot dan toe gebruikte medewerkerstabel vervangen door een *view* op onze data.

Door deze uitrol enkele dagen voor het einde van het project uit te voeren bleven er nog enkele dagen over om kleine bugs te herstellen en te controleren of andere applicaties nog correct werken.

Issues en oplossingen

Het eerste grote probleem dat we tegenkwamen was de installatie van SmartPart v1.2 bèta op de portal. Hier bleek dat er al een oude versie van SmartPart op de portal aanwezig was, die echter niet compatible was met AJAX en waarvan de vraag was of alle functionaliteit nog werkte bij een update. Besloten is daarom om beide versies naar elkaar te installeren op de portal; een procedure die wat tijd kostte maar uiteindelijk wel werkbaar bleek.

Tijdens de deployment van Reporting Services kregen we de integratie tussen SharePoint en Reporting Services niet goed werkend. Waar het geheel in de development omgeving prima draaide, geeft de productie omgeving onverklaarbare authenticatie fouten. Eén en ander heeft waarschijnlijk te maken met het feit dat in de productie omgeving de SharePoint en Reporting server op twee verschillende servers draaien. Hier moet op dit moment nog een oplossing voor worden gezocht, tot die tijd zijn de rapportages via de development omgeving op te vragen.

De uitrol van de employee self service functionaliteit naar de MySite is niet uitgevoerd; de functionaliteit staat nu als een aparte site op de portal. Op dit moment is deze functionaliteit nog niet uitgebreid aangekondigd bij de medewerkers om te voorkomen dat de Servicedesk overspoeld wordt met *calls* van medewerkers als er iets niet goed blijkt te werken. In de toekomst zal deze site officieel in gebruik worden genomen door alle medewerkers te berichten van de employee self service site. In de toekomst is het

eventueel mogelijk om de WebParts die nu op deze aparte site staan in de MySite template op te nemen, maar tot nu toe is dit nog niet het geval.

Na de uiteindelijke uitrol leek alles probleemloos te werken, de systemen die van de migratie afhankelijk waren inclusief. Na de deployment zelf werden er nog een drietal niet noemenswaardige bugs gevonden. De volgende dag waren er echter enkele problemen opgetreden. Allereerst werkte de urenboekapplicatie niet correct bij alle medewerkers, wat kwam door een *cast* in een andere applicatie naar een integer van data die in principe nog niet bekend was. In onze tabel stond deze dan ook als NULL aangegeven, wat een *exception* tot gevolg had. Dit probleem was met een simpele *cast* in onze view snel opgelost.

Ook bleek het boeken van uren en reiskosten niet meer goed te werken. Schijnbaar at random leek er een *exception* te verschijnen, waarbij de kans groot leek dat dit aan onze wijzigingen in de database lag. Echter bleek het probleem in de aanpassing van een configuratiebestand te zitten wat door een andere medewerker gedaan was. Het ongedaan maken van deze aanpassing bleek het probleem op te lossen. De standaard weergave van je rol en vervoersmiddel voor het declareren van uren en reiskosten bleek ook niet meer te werken. Dit hebben we aangepast door onze applicatie iets uit te breiden waardoor dit soort gegeven ook weer werd opgeslagen.

Al met al is de overgang naar het nieuwe systeem soepel verlopen. De paar fouten die optraden lagen vooral in andere applicaties en waren snel opgelost. In de komende weken blijft het spannend of er nog bugs binnenkomen. Uiteraard hopen we dat dit aantal miniem blijft en dat de applicatie snel en stabiel blijft draaien.

9 Evaluatie resultaten

Aan het eind van het project gekomen, nemen we de resultaten in beschouwing en evalueren we ons bachelorproject. In dit hoofdstuk bespreken we het resultaat per fase van het project. Allereerst bespreken we de requirements en kijken we of het opgeleverde systeem aan de gestelde eisen voldoet. Vervolgens evalueren we de ontwerpfase. In het stuk over de implementatiefase kijken we terug op onze aanpak hiervan. Tevens nemen we in dit gedeelte de gebruikte ontwikkelomgeving en het softwareplatform in beschouwing. Daarna komt de testfase aan bod. In het gedeelte over deployment evalueren we het uitrollen van ons systeem binnen Tam Tam. Het hoofdstuk wordt afgesloten met een evaluatie van het proces waarin de aanpak en de planning besproken worden.

Requirements

In dit stuk evalueren we de analysefase en de hieruit voortvloeiende requirements. We evalueren de aanpak, de kwaliteit van de requirements en de beperking van te implementeren requirements. Vervolgens controleren we de implementatie van afzonderlijke requirements.

Aanpak

Het gebruik van interviews met de verschillende gebruikersgroepen van het systeem is ons goed bevallen. We hebben gemerkt dat het goed werkt om de conclusies van deze interviews op schrift te stellen en ter goedkeuring voor te leggen aan de betrokkenen.

Kwaliteit requirements

De requirements zijn redelijk aan het begin van het project opgesteld. Tijdens de fasen erna zijn we weinig zaken tegengekomen die niet in de requirements waren opgenomen. Hieruit kunnen we concluderen dat de opgestelde requirements een goed pakket van eisen vormden en dat de analysefase geslaagd is.

Beperking

De keuze om de implementatie van het systeem tot de requirements met prioriteiten van 1 tot 3 te beperken is een goed besluit geweest, aangezien het op deze manier is gelukt om een goed werkend systeem op te leveren binnen de gestelde tijd.

Controle

Benevens te voldoen aan de minimale eisen van het implementeren van tenminste de requirementsgroepen 1 en 2, hebben we ook de requirements met prioriteit 3 geïmplementeerd, met de volgende bijzonderheden:

- Requirement 1g (Toevoegen/wijzigen hardware) is wel geïmplementeerd, maar zal in eerste instantie nog niet gebruikt worden aangezien de gebruikers bij nader inzien toch hogere eisen aan deze functionaliteit stellen. Wij hebben het geïmplementeerd als een enkel tekstveld waarin kan worden bijgehouden welke hardware een medewerker in zijn bezit heeft. Er is echter behoefte aan een uitgebreider invoermogelijkheid met verschillende velden waarbij bepaalde informatie verplicht ingevoerd moet worden. Wij hadden geen gelegenheid meer om deze aangepaste requirement te implementeren.

- Requirement 1h (Invoeren/wijzigen schulden werknemer) is ondanks een prioriteit van 6 toch geïmplementeerd. Dit is omdat het weinig werk kostte om deze requirement mee te nemen en hij behoort tot de requirementsgroep waartoe de meeste requirements met prioriteit 1 behoren.
- Requirement 3f (Notificatie HR bij twee weken ziekte) is ondanks een prioriteit van 9 toch geïmplementeerd, met dezelfde redenen als het voorgaande.

We hebben vooraf ingeschat dat er voldoende tijd zou zijn om de eerste drie requirementsgroepen te implementeren, in tegenstelling tot de eerste twee groepen, wat afgesproken was met de opdrachtgever. Achteraf gezien was dit een juiste inschatting.

Ontwerp

In dit gedeelte evalueren we kort enkele aspecten van de ontwerpfase. We kijken eerst terug naar de gehanteerde aanpak en behandelen daarna de kwaliteit van het ontwerp.

Aanpak

Het ontwikkelen van het grafisch ontwerp van de HRM-applicatie op papier is ons goed bevallen. We kregen op die manier snel zicht op de benodigde onderdelen van het systeem die in het functioneel ontwerp terug moesten komen. Tevens was het op deze manier mogelijk om al vroeg in het proces onze ideeën over het systeem voor te leggen aan de eindgebruikers.

Kwaliteit

Het is tijdens de implementatiefase niet nodig geweest om veel veranderingen aan te brengen in het ontwerp. Het initiële ontwerp bleek een goede basis voor ontwikkeling van het systeem. We kunnen hier dan ook tevreden over zijn.

Implementatie

In dit gedeelte evalueren we om te beginnen de aanpak van de implementatie. Vervolgens kijken we naar het implementeren van de WebParts. Daarna beschouwen we kort de voor- en nadelen van de gebruikte softwareomgevingen.

Aanpak

Door voor het begin van de daadwerkelijke implementatiefase al aan de slag te gaan met het maken van *proofs of concept*, hebben we de kans op verrassingen zo klein mogelijk gemaakt. Achteraf denken we dat het een goede keus is geweest om al vroeg te kijken naar de daadwerkelijke implementatie van het systeem.

Het veelvuldig gebruik van virtual machines als testomgeving voor ons systeem heeft twee kanten. Aan de ene kant heeft het ons meer tijd gekost omdat virtual machines iets trager werken, en met name omdat we de code steeds moesten verplaatsen naar de testomgeving voor hij op fatsoenlijke wijze uitgevoerd kon worden. Aan de andere kant was deze simulatie van de uiteindelijke productieomgeving een prima manier om op veilige wijze te kunnen ontwikkelen.

Implementatie WebParts

Het implementeren van de WebParts kostte vrij veel tijd, omdat tijdens het schrijven van de code veel kleine probleempjes opdoken. We hadden hier echter rekening mee gehouden in de planning, zodat het toch op tijd af was.

Softwareplatform en technologieën

Bij het softwareplatform hebben we gemengde gevoelens. Binnen SharePoint zijn sommige dingen eenvoudig te maken en is het mogelijk om met weinig werk erg nette componenten te maken. Ook bij het ontwikkelen van WebParts in .NET is ons opgevallen hoe snel sommige oplossingen zijn te maken door middel van de componenten van het framework. Voor zowel SharePoint als .NET geldt echter dat het vaak veel werk kost om iets te maken dat niet goed past binnen de standaardcomponenten van het framework.

Een ander probleem met de huidige opstelling welke voornamelijk veroorzaakt wordt door SharePoint, is de spreiding over veel verschillende onderdelen van je applicatie en de configuratie. Er is niet één *sourcetree* (gezien er aparte rapportages zijn) en het systeem bestaat uit meerdere servers met allen hun eigen security instellingen. Dit maakt de opbouw van de applicatie minder doorzichtig en bemoeilijkt de deployment. Langzamerhand komen hier wel oplossingen voor beschikbaar (in de vorm van “Solutions”, welke een applicatie in één keer kunnen uitrollen), maar je merkt zo toch dat het SharePoint platform nog sterk in ontwikkeling is en hier en daar nog wat ruwe kanten kent.

De gebruikte programmeertaal, C#, is een taal die in veel opzichten lijkt op Java. Sommige zaken zijn echter nog helderder dan binnen Java. Het werken met C# was dan ook erg prettig.

Testen

De verschillende manieren van testen die we hebben gehanteerd, hebben een goede bijdrage aan het slagen van het project geleverd.

De schermentest met op papier uitgewerkte user interfaces leverde nuttig commentaar op om de usability te verbeteren. Door de gebruikerstest met zowel de papieren versie als de eerste digitale conceptversie kwamen daarnaast enkele kleine verbeterpunten boven wat betreft de verschillende velden van het systeem.

De integratietest, waarin we volgens een testplan verschillende scenario's op het systeem loslieten, bracht een aantal bugs naar voren die vrij snel opgelost konden worden.

Deployment

Omdat ons systeem geïntegreerd moest worden in een systeem dat dagelijks gebruikt wordt en waarvan veel mensen afhankelijk zijn, was het belangrijk dat deze stap zorgvuldig plaatsvond. Ondanks het feit dat de bestaande systemen op vele onzichtbare wijzen afhingen van de gegevens in de database waarvoor ons systeem een vervanging was, is het gelukt om ons systeem zonder noemenswaardige problemen uit te rollen.

Wel hadden we voor het uitrollen van het systeem graag wat meer tijd gehad. Omdat het invoeren van de arbeidsvoorwaarden vanuit Excel sheets wat tijd kost en het goed is om tijdens dit proces ook gelegenheid te hebben om bij te springen mochten er onverhoopt problemen optreden, hadden we graag nog een week de tijd gehad om het systeem volledig in actie te zien met het gedeelte voor arbeidsvoorwaarden ook in gebruik.

Proces

In dit gedeelte bespreken we het proces van het bachelorproject. We evalueren eerst onze aanpak van het project en kijken vervolgens naar de planning.

Aanpak

De gehanteerde ontwerpmethodiek is een klassiek watervalmodel geweest, waarin echter de aanpassing is gedaan van het gebruik van *proofs of concept* om vroeg kennis te maken met de praktijk van het ontwikkelen van oplossingen in .NET en SharePoint. Deze aanpak is een vruchtbare gebleken, aangezien het ons gelukt is om binnen de gestelde tijd een werkend systeem te ontwikkelen. We merkten dat het erg veel hielp om vroeg een stukje implementatie te doen zodat we een realistische tijdsinschatting konden maken en niet voor verrassingen kwamen te staan. Tevens beviel het ons goed om de verschillende fasen van het project enigszins gescheiden te houden, om zo duidelijke momenten te hebben waarop we tot een afronding van een fase moesten komen.

De requirements zijn ingedeeld op prioriteit volgens een uitgebreidere variant van de MoSCoW-analyse. Achteraf bleek dit een prettige manier van werken, aangezien we flexibel konden zijn in hoeveel prioriteitsniveaus we implementeerden en hoeveel we lieten liggen om na afloop van het project eventueel geïmplementeerd te worden.

De communicatie en evaluatie van deelresultaten binnen Tam Tam heeft naar tevredenheid plaatsgevonden. De regelmatige controle door eindgebruikers van de eisen aan en de werking van het systeem hielp ons om gericht te kunnen ontwikkelen.

Het regelmatige contact met onze stagebegeleider zorgde voor een goede stok achter de deur om doelgericht te blijven en om onze prioriteiten goed te stellen, zodat we in de beperkte tijd die we hadden een goed systeem konden ontwikkelen. De frequentie van ontmoeten (aan het begin wekelijks, later iets minder vaak en een meer vraaggestuurde aanpak) werd door ons beiden als prettig ervaren.

Planning

De planning die we in de eerste week hebben opgesteld is een goede richtlijn geweest voor ons project. Achteraf kunnen we concluderen dat we ons vrij strak aan de planning hebben gehouden. De enige afwijkingen zijn dat we de code-freeze niet geheel hebben voltrokken op het geplande moment, waar tegenover staat dat we iets eerder zijn begonnen met deployment van ons systeem.

Wel hadden zoals vermeld we achteraf graag wat meer tijd gehad voor deployment van het systeem. We hadden beter iets meer tijd hiervoor kunnen inruimen in de planning.

10 Aanbevelingen

In dit hoofdstuk willen we kort enkele aanbevelingen doen. Eerst een aanbeveling aan Tam Tam, en vervolgens een aanbeveling aan de TU Delft.

Aanbeveling Tam Tam

We hebben gemerkt dat Tam Tam door de grote groei die het bedrijf in de afgelopen jaren heeft doorgemaakt een ware wildgroei aan verschillende applicaties voor intern gebruik heeft, waardoor de samenhang tussen deze systemen mist. Daarnaast zijn er gebieden waarop verdere automatisering winst in efficiëntie zou opleveren. We willen Tam Tam dan ook aanbevelen om meer tijd te stoppen in interne ontwikkeling van een goed en samenhangend systeem om zodoende de bedrijfsprocessen op een slimme manier te ondersteunen.

Aanbeveling TU Delft

Onze presentatie vindt iets later plaats dan verwacht. De voornaamste reden hiervoor is dat het niet mogelijk was om de presentatie vroeg te plannen. Regel is om de presentatie twee weken na ontvangst van het verslag te plannen. We willen de TU Delft aanbevelen om voor het bachelorproject aan het begin een datum te plannen voor de presentatie en duidelijk af te spreken wanneer het verslag binnen moet zijn. Mogelijk is natuurlijk dat een project anders loopt dan gepland, maar aan de andere kant kan van universitaire studenten verwacht worden dat ze de verantwoordelijkheid voor hun eigen planning nemen en zorgen dat ze zich aan de vooraf gestelde deadlines houden.

11 Literatuurlijst

Alan Dennis, Barbara Haley Wixom, David Tegarden, *System Analysis & Design, An object-oriented approach with UML*, Wiley, 2002

David S. Platt, *Introducing Microsoft .NET*, Microsoft Press, third edition, 2003

Dino Esposito, *Introducing Microsoft ASP.NET 2.0*, Microsoft Press, 2005

John Sharp, *Microsoft Visual C# 2005, step by step*, Microsoft Press, 2006

Microsoft Developer Network, <http://msdn2.microsoft.com/en-us/default.aspx>

Scot Hillier, *Microsoft SharePoint, building Office 2007 solutions in C# 2005*, Apress, 2007

Timothy C. Lethbridge and Robert Laganière, *Object-oriented software engineering*, McGraw-Hill, 2001

12 Verklarende woordenlijst

BSN	Burger Service Nummer, opvolger van het sofinummer
Business unit	Afdeling in het primaire bedrijfsproces
CRM	Customer Relationship Management, bijhouden van klantrelaties in een geautomatiseerd systeem
Document library	Bibliotheek met bestanden in SharePoint
ESS	Zie Employee Self Service
Employee Self Service	Service voor medewerkers om toegang te krijgen tot hun persoonlijke gegevens en de mogelijkheid deze in sommige gevallen te wijzigen
Fte	full-time-equivalent, één fte is gelijk aan een volledige werkweek
HRM	Zie Human Resource Management
Human Resource Management	Afdeling en medewerkers die over personeelszaken gaat
HTTP	Hyper Text Transfer Protocol
IIS	Internet Information Server, webserver van Microsoft
Impersonation	De identiteit van een gebruiker aannemen
MOSS 2007	Microsoft Office SharePoint Server 2007
MySite	Persoonlijke pagina op een SharePoint portal
POP	Persoonlijk OntwikkelingsPlan, in een POP geeft een medewerker aan wat zijn ontwikkelingswensen/doelen zijn voor een bepaalde periode. Bij Tam Tam wordt dit tevens gebruikt in het beoordelingsproces
POP-gesprek	Gesprek met leidinggevende over het POP van een medewerker
Postback	Terugsturen van data naar een webserver
QueryString	Deel van een URL dat na het vraagteken staat, gebruikt op parameters mee te geven
SDK	Software Development Kit
Security-trimmed	Alleen objecten waar de gebruiker toegang toe heeft weergeven
Service desk	Afdeling verantwoordelijk voor de afhandeling van service calls
Tam Tam model	Beoordeling- en ontwikkelingsmodel voor medewerkers op basis van rollen binnen een organisatie, ontwikkeld door Bureau Baarda
Trigger	Procedurele code die wordt uitgevoerd na aanleiding van een actie in een database
Unit manager	Leidinggevende binnen een business unit
UserControl	Een stuk ASP pagina dat kan worden hergebruikt op andere pagina's
Workflow	Organisatorische procedure met daarin taken, wie deze uitvoert, de onderlinge afhankelijkheden en bijbehorende informatiestromen

A. Huidige procedures

Gegevens medewerkers

HR is in de huidige situatie verantwoordelijk voor het onderhouden van gegevens van medewerkers, waaronder personalia, arbeidsvoorwaarden, etc. De meeste personalia worden bijgehouden in een systeem genaamd 'Tammo-administratie'. Gegevens over secundaire arbeidsvoorwaarden worden voornamelijk bijgehouden met behulp van Excel-sheets en in Word-documenten. Daar er vrijwel geen automatisering is moet informatie vaak op verschillende plaatsen aangepast worden, waarbij ook andere afdelingen (zoals office management of systeembeheer) en instanties (zoals de Belastingdienst) ingelicht moeten worden.

In- en uitdiensttreding

Bij indiensttreding dienen verschillende partijen taken uit te voeren zodat de nieuwe medewerker aan de slag kan binnen de organisatie. HR dient onder andere een aantal formulieren te verzamelen voor de Belastingdienst en het GAK. Office management dient voor een werkplek te zorgen die ingericht wordt door office management zelf en systeembeheer, waarbij laatstgenoemde tevens een account en eventuele hardware verzorgt. Bij uitdiensttreding dient er ook een aantal formaliteiten uitgevoerd te worden, waaronder wederom het verzenden van allerlei formulieren aan officiële instanties en het afnemen van een *exitinterview*.

POP

De Persoonlijk Ontwikkelings Plan procedure dient voor het evalueren van de prestaties van de medewerkers, het vaststellen van doelstellingen en het bepalen van het salaris en de bonus voor medewerkers. Voor de POP procedure zijn er drie formulieren beschikbaar – terugkijken, vooruitkijken en salaris – welke deels door de unit manager en deels door de medewerker worden ingevuld. De POP procedure wordt twee keer per jaar uitgevoerd: in januari en in juli, waarbij er in januari ook een eventuele salarisverhoging plaatsvindt. De POP procedure verloopt als volgt:

1. Financiën zorgt voor de sheets met de salarisontwikkeling, projecten, uren en *declarabiliteit* (verhouding tussen het aantal uur wat daadwerkelijk aan projecten is gewerkt en het aantal contracturen minus vakantie-uren) van de medewerkers en stuurt deze naar de unit managers;
2. Zowel de medewerker als de unit manager vullen de voorbereidingsvelden op de POP formulieren in waardoor de verschillende partijen weten wat ze van het gesprek kunnen verwachten;
3. De unit managers bereiden in overleg met het overige management een salarisvoorstel voor;
4. Het POP gesprek vindt plaats, waarin de doelstellingen over het afgelopen half jaar worden bekeken en geëvalueerd en de doelstellingen voor het aankomende half jaar worden vastgesteld. Verder wordt de bonus en de eventuele salarisverhoging besproken. Hiervan wordt ook een verslag bijgehouden op het POP formulier;
5. Eventuele aanvullingen op het verslag worden uitgezocht en het POP formulier wordt waar nodig aangevuld. Eventueel vindt er nog een extra gesprek plaats met de medewerker om de uiteindelijke uitkomst van de POP-procedure te bespreken;

6. De salarisverhoging en bonus wordt vastgesteld en de POP formulieren worden geprint en ondertekend. Dit geheel gaat in de personeelsmap van de medewerker en tevens in het digitale dossier;
7. De unit managers leveren Financiën en HR een overzicht aan van de salarismwijzigingen en de bonussen, waarna deze worden verwerkt in de administratie. Tevens stuurt HR een bevestigingsbrief naar alle medewerkers.

Relatie met medewerkers onderhouden

Aan verjaardagen van medewerkers wordt door HR altijd wat aandacht besteed. Op dit moment zetten zij alle verjaardagen handmatig in een Outlookagenda. Ook voor aflopende contracten, werkvergunningen die bepaalde tijd geldig zijn, einde-proeftijd-gesprekken en het plannen van een gesprek met de unit manager worden reminders in een Outlookagenda geplaatst.

Contact HR en Financiën

Wijzigingen in de personeelsadministratie worden nu handmatig doorgegeven aan Financiën, die het op zijn beurt doorspeelt aan belanghebbende instanties. Financiën geeft de gegevens door aan PriceWaterhouseCoopers en HR aan Van den Ende Adviesgroep ten behoeve van de pensioenen.

Verzuim en vakanties

Verzuim en vakanties worden bijgehouden in het al bestaande urenregistratiesysteem. Excel sheets halen deze informatie uit een database en verwerken deze semi-geautomatiseerd. Het aantal beschikbare vakantiedagen wordt handmatig bijgewerkt (aan het begin van het jaar opgehoogd en ook handmatig herberekend als iemand parttime gaat werken). Elke maand wordt deze informatie aan alle medewerkers beschikbaar gesteld.

Bij ziekte dient men in de gaten te houden dat na enige tijd afwezigheid actie ondernomen moet worden met verschillende instanties. Tevens dient Financiën wederom handmatig op de hoogte gesteld te worden bij ziekte van een personeelslid.

Medewerkerdossier

Het huidige medewerkerdossier bestaat uit een hardcopy-dossier in een map met daarin alle relevante door de medewerker ondertekende formulieren. Tevens bestaat er digitaal een dossier bestaande uit relevante Word-documenten.

Salarisverwerking

Financiën doet nu alle salarisverwerking met behulp van Excel sheets en Exact software. Maandelijks wordt een overzicht van alle af te handelen betalingsmutaties gemaakt en naar PriceWaterhouseCoopers gestuurd.

B. Uitgewerkte lijst van functionele requirements

1. Personeelsadministratie

a) Bekijken medewerker (1)

Medewerkers en hun unit managers kunnen alle voor hen geregistreerde gegevens bekijken. Welke gegevens in het systeem worden bijgehouden is te zien in de tabel in appendix F.

b) Invoeren medewerker (1)

HR kan een nieuwe medewerker invoeren. Welke gegevens hierbij minstens moeten worden ingevoerd staat in de tabel in appendix F. De gegevens van de medewerker moeten eventueel uit de CRM database met sollicitanten gehaald kunnen worden.

Na het invoeren van een medewerker wordt een traject gestart van acties die bij indiensttreding van een werknemer moeten gebeuren. Zie hiervoor punt 4.

Opmerking: naast persoonsgegevens moeten bij het invoeren ook arbeidsvoorwaarden worden ingevuld. Zie hiervoor punt 2.

c) Wijzigen medewerker (1)

Medewerkers kunnen enkele van hun eigen gegevens wijzigen via de My Site in de portal. Welke gegevens zij kunnen wijzigen staat gespecificeerd in de tabel in appendix F.

De HR-medewerker kan alle gegevens van een medewerker wijzigen.

De unit manager kan voor de medewerkers in zijn unit enkele gegevens wijzigen die te maken hebben met hun rol(len) binnen Tam Tam.

d) Verwijderen medewerker (1)

De HR-medewerker kan een werknemer uit het systeem verwijderen. In de praktijk worden de gegevens niet daadwerkelijk verwijderd, maar vallen ze vanaf het moment van verwijderen 'buiten het systeem'.

e) Bekijken partners of kinderen (1)

De HR-medewerker en de medewerker zelf kunnen gegevens over diens partner en kinderen bekijken.

f) Aanpassen partners of kinderen (1)

Medewerkers kunnen hun eigen partner en kinderen in het systeem invoeren, wijzigen en verwijderen. De HR-medewerker kan dit voor alle medewerkers.

g) Toevoegen/wijzigen hardware (1)

De hardware welke een medewerker in zijn bezit heeft moet kunnen worden bijgehouden. Dit moet gebeuren door systeembeheer. Dit zodat bij uitdiensttreding van een medewerker ook gegarandeerd kan worden dat alle hardware weer terug komt.

h) Invoeren/wijzigen schulden werknemer (6)

Per medewerker moet de afdeling HR eventuele schulden kunnen bijhouden. De meeste van deze schulden zullen automatisch gegenereerd worden aan de hand van de opleidingen welke een medewerker volgt, maar soms kan het voorkomen dat een schuld handmatig ingevoerd moet worden. Hierbij kan ook gedacht worden aan bijvoorbeeld een renteloze lening voor een pc welke over 36 maanden wordt afgelost.

i) Notificatie HR bij verjaardag / jubileum medewerker (1)

Bij een verjaardag van een medewerker krijgt de HR-medewerker een in te stellen hoeveelheid tijd van tevoren een notificatie. Ook bij een jubileum van medewerkers wordt de HR-medewerker op de hoogte gesteld.

j) Notificatie HR bij verjaardag partner of kind (1)

Bij een verjaardag van partner of kind van een medewerker krijgt de HR-medewerker een in te stellen hoeveelheid tijd van tevoren een notificatie.

k) Notificatie HR bij aflopen werkvergunning (1)

Voor medewerkers met een werkvergunning die beperkt geldig is, is het belangrijk dat HR tijdig op de hoogte wordt gesteld van het aflopen van deze vergunning. Er kan door de HR-medewerker ingesteld worden hoeveel tijd van tevoren een notificatie wordt gegeven.

l) Notificatie HR bij einde proeftijd en einde contract (1)

Voor de proeftijd of een contract van een medewerker afloopt krijgt HR een herinnering. Er kan ingesteld worden hoeveel tijd van tevoren deze notificatie wordt gegeven.

m) Notificatie Financiën bij wijziging adres- of bankgegevens (1)

Bij een wijziging in adres- of bankgegevens, wordt de afdeling Financiën middels een notificatie op de hoogte gesteld, zodat deze dit kan doorgeven aan PWC.

n) Rapportage overzicht medewerkers (1)

Voor een HR-medewerker is het mogelijk om een overzicht te krijgen van de gegevens van alle medewerkers. Een unit manager kan ditzelfde overzicht krijgen voor alle medewerkers in zijn unit.

2. Arbeidsvoorwaarden

a) Bekijken arbeidsvoorwaarden (1)

Zowel HR, Financiën, als de unit manager van een medewerker moeten de arbeidsvoorwaarden van de medewerker kunnen bekijken. Welke verschillende onderdelen dit precies betreft is terug te vinden in de tabel in appendix F. Ook kan een medewerker zijn eigen arbeidsvoorwaarden via zijn My Site bekijken.

b) Invoeren arbeidsvoorwaarden (1)

Nadat de gegevens van een medewerker zijn ingevoerd (punt 1a) moeten de basisarbeidsvoorwaarden van de medewerker worden ingevuld. Welke dit zijn is in de tabel in appendix F terug te vinden. Dit gebeurt door HR. Na dit invoeren gaat een notificatie naar de Financiën afdeling (zie ook 4) waardoor deze op de hoogte zijn van de arbeidsvoorwaarden en voor verdere verwerking kunnen zorgen.

c) Wijzigen arbeidsvoorwaarden (zie ook 5.) (1)

Arbeidsvoorwaarden kunnen in de loop der tijd wijzigen. Deze wijzigingen worden gerapporteerd door de unit manager en in het systeem ingevoerd door HR. Het is hierbij erg belangrijk dat de exacte ingangsdatum van de wijziging wordt bijgehouden, zodat voor Financiën duidelijk is vanaf welke datum eventuele betalingen in moeten gaan (zie ook e). De arbeidsvoorwaarden kunnen ook door het systeem vanuit het POP proces gewijzigd worden; dat is de gebruikelijke manier.

d) Historie arbeidsvoorwaarden (1)

Unit managers, Financiën en HR moeten de historie van de arbeidsvoorwaarden van een medewerker kunnen bekijken. Deze historie moet wanneer mogelijk ook voor de medewerker zichtbaar zijn via zijn My Site. Duidelijk moet te zien zijn welke arbeidsvoorwaarden op welk moment gewijzigd zijn. Het formaat waarin dit gerepresenteerd wordt kan bijvoorbeeld een grafiek of een tabel zijn.

e) Notificatie Financiën bij wijziging arbeidsvoorwaarden (1)

Het is voor Financiën belangrijk op de hoogte te zijn van wijzigingen in de arbeidsvoorwaarden van een medewerker. Hiertoe dienen zij een notificatie te ontvangen van de wijziging van de arbeidsvoorwaarden van een medewerker, zodat dit op de juiste manier in de administratie verwerkt kan worden. Naast een wijziging van het salaris is bijvoorbeeld ook een wijziging van het parttimepercentage belangrijk om te melden.

f) Rapportage salaris, bonus en ontwikkeling per medewerker (1)

Er dient door HR, Financiën en het management een standaard rapport uitgedraaid te kunnen worden waarin duidelijk de ontwikkeling van het salaris en de bonus van een medewerker weergegeven wordt. Ook dienen hierin belangrijke secundaire arbeidsvoorwaarden (leaseauto met name) een plaats te krijgen.

g) Rapportage overzicht salarissen, bonussen etc. (1)

Unit manager, HR en Financiën dienen een overzicht van alle salarissen en bonussen te kunnen uitdraaien, samen met totalen per unit of functie. De unit managers kunnen dit alleen van hun eigen afdeling genereren, HR en Financiën kunnen dit ook van het hele bedrijf. Tevens dienen de verloopcijfers per unit en voor het hele bedrijf inzichtelijk gemaakt te worden, dient er een overzicht van het aantal parttime medewerkers te zijn en dient de totale hoeveelheid *fte* bekend te zijn.

h) Rapportage t.b.v. salarisadministratie (5)

Ten behoeve van de salarisadministratie wordt er een specifiek formaat document gebruikt wat naar de salarisadministrateur gaat. Financiën moet dit document per maand kunnen verkrijgen uit het systeem om verder te verwerken en naar de salarisadministrateur te sturen.

3. Verzuim en vakanties

a) Overzicht vakantiedagengebruik en resterend per werknemer (2)

Alle actoren moeten een overzicht van het vakantiedagengebruik van een medewerker op kunnen vragen. Een medewerker kan dit alleen voor zichzelf, een unit manager alleen voor zijn unit en HR en Finance kunnen dit voor het gehele bedrijf. Het overzicht laat zien hoeveel vakantiedagen de medewerker dit jaar al heeft ontvangen, hoeveel er dit jaar nog te ontvangen zijn (ervan uitgaande dat er geen wijziging in secundaire arbeidsvoorwaarden optreedt) en hoeveel er reeds gebruikt zijn. Ook wordt een overzicht getoond van wanneer de vakantiedagen gebruikt zijn. Tevens worden de dagen ook in uren getoond, aangezien Financiën en HR hiermee werken. De informatie voor het vakantiedagengebruik wordt uit de uren database verkregen. Een medewerker kan een negatief aantal vakantieuren hebben, dit kan hij dan in de rest van het jaar compenseren.

b) Invoeren ziek en betermeldingen (3)

HR wordt door de medewerker gecontacteerd wanneer deze ziek wordt. Deze meldingen moeten bijgehouden kunnen worden samen met het tijdstip van ziekmelding.

c) Bekijken ziek en betermeldingen per medewerker (3)

Van de meldingen zoals besproken bij 3b) moet per medewerker een overzicht getoond kunnen worden aan de medewerker, HR of de unit manager. Dit overzicht bevat tijdstip ziekmelding, reden, tijdstip betermelding en duur. Ook worden er enkele totalen gegeven van de hoeveelheid verzuim dit jaar en afgelopen jaar. Tevens komt er op de voorpagina van de applicatie een klein overzicht van zieken en ziekte duur.

d) Invoeren correctie vakantiedagen (2)

Af en toe zal het systeem niet volledig automatisch om kunnen gaan met de vakantiedagen, bijvoorbeeld omdat een medewerker pas op de helft van de maand in dienst treedt. Ook willen medewerkers soms nog wel eens fouten maken tijdens het invoeren van hun vakantie-uren. Het systeem dient hierom een mogelijkheid aan HR te bieden om een correctie in te voeren van het aantal vakantie-uren.

e) Automatische aanpassing vakantiedagen a.d.h.v. arbeidsvoorwaarden (2)

Iedere maand dient de medewerker nieuwe vakantie-uren te krijgen aan de hand van de arbeidsvoorwaarden en het parttimepercentage. Deze dienen automatisch door het systeem toegevoegd te worden waarna ze beschikbaar zijn voor de medewerker.

f) Notificatie HR na twee weken ziekte (9)

Na twee weken moeten er bepaalde processen opgestart worden bij HR ten behoeve van de re-integratie van de zieke medewerker. Het is daarom handig als HR een notificatie krijgt wanneer de medewerker twee weken ziek is om op dit moment een dossier te starten.

g) Notificatie Financiën bij ziekte (3)

Bij ziekte wordt er na twee weken nog 70% van het loon doorbetaald. Financiën moet daarom een notificatie krijgen wanneer een medewerker ziek wordt zodat hiermee rekening gehouden kan worden bij uitbetaling van het salaris. Eventueel dient dit ook op het overzicht bij 2h) opgenomen te worden.

h) Rapportage verzuim per medewerker en unit (5)

Het systeem dient in staat te zijn een rapportage te genereren per medewerker en per unit waarop het verzuim wordt weergegeven. In het geval van het verzuim per medewerker worden hierop de gegevens getoond zoals bij a), in het geval van verzuim per unit wordt getoond wie wanneer ziek was en worden de totalen getoond. Dit moet kunnen gebeuren over een arbitraire periode en moet mogelijk zijn voor de unit managers en HR.

4. Uit- en indiensttreding

Bij uit- en indiensttreding van medewerkers moeten verschillende partijen een aantal handelingen verrichten. Bij het toevoegen van een medewerker in het systeem worden automatisch voor alle betrokken partijen lijstjes gemaakt met *todo* items.

a) Bewerken standaard *todo* items (4)

Voor zowel de HR-medewerker, de unit manager, de Financiënmedewerker als de System Administrator is het mogelijk om hun eigen lijst van *todo* items – die standaard worden aangemaakt bij toevoeging van een nieuwe medewerker – te wijzigen door items toe te voegen en te verwijderen.

b) Afvinken *todo* item met eventuele notitie (4)

Na voltooiing van een *todo* item voor een partij kan deze het item afvinken en daarbij een notitie toevoegen, die door hemzelf of door een unit manager of een HR-medewerker gelezen kan worden.

Bij het afvinken kunnen System Administrators en Financiënmedewerkers alleen hun eigen *todo* items zien. Unit managers kunnen alle *todo* items van een medewerker zien, maar alleen hun eigen *todo* items afvinken. HR-medewerkers kunnen alle items afvinken.

c) Koppeling *todo* items aan automatisch acties (8)

Het is mogelijk om aan een *todo* item een automatische actie te koppelen. Dit is vooral interessant voor de System Administrators, aangezien die veelal goed automatiseerbare handelingen moeten verrichten.

d) Automatische actie aanmaken account AD (9)

De System Administrator kan automatisch een Active Directory-account aanmaken. De gegevens die hierin komen worden automatisch ingevuld vanuit de personeelsgegevens. De gebruikersnaam wordt door de System Administrator ingevuld.

e) Automatische actie aanmaken mailbox (9)

De System Administrator kan automatisch een mailbox aanmaken. De gegevens die nodig zijn voor het aanmaken worden automatisch ingevuld; voor het e-mailadres wordt een voorstel gedaan. De System Administrator kan hier nog wel wijzigingen in aanbrengen voor de mailbox wordt aangemaakt.

f) Automatische actie aanmaken userfolder (9)

De System Administrator kan automatisch een map op de netwerkschijf aanmaken voor een nieuwe gebruiker.

g) Automatische actie aanmaken SIP account Communicator (9)

De System Administrator kan automatisch een SIP account voor medewerkers aanmaken voor gebruik van Office Communicator.

h) Automatische deactivatie account bij uitdiensttreding (9)

Bij uitdiensttreding van een medewerker dient zijn account gedeactiveerd te worden in de Active Directory. Verder dient het account uit de groepen en distributielijsten gehaald te worden, zodat deze schoon en overzichtelijk blijven.

i) Notificatie relevante personen van nieuw todo item (4)

Wanneer een nieuwe medewerker wordt aangemaakt, wordt automatisch een nieuwe todo-list gemaakt. De partijen waarvan hier items op staan, krijgen hiervan een notificatie.

j) Notificatie HR bij lang openstaande todo items (9)

Wanneer een todo item geruime tijd niet wordt afgevinkt, wordt er een nieuwe notificatie gegeven. Deze gaat naar de betrokken partij en naar de HR-medewerker. Na hoe lang deze herinnering wordt gegeven, is in te stellen door de HR-medewerker.

k) Rapportage verschillende gegevens t.b.v. uitdiensttreding (6)

Bij de uitdiensttreding is het handig als de HR medewerker een overzicht kan krijgen van verschillende gegevens welke van belang zijn. Hierbij gaat het bijvoorbeeld om de hardware welke de medewerker nog in zijn bezit heeft en de openstaande schulden.

5. Beoordeling, loopbaan, ontwikkeling, opleiding

a) Invoeren/wijzigen POP voorbereiding door werknemer (7)

Ten behoeve van de POP procedure dienen enkele voorbereidingsvelden door de werknemer ingevuld te worden. Dit moet mogelijk zijn voor de werknemer via zijn My Site. Hierin geeft hij aan om welk POP gesprek het gaat en vult hij zijn voorbereiding in.

- b) Invoeren/wijzigen POP voorbereiding door UM (7)

Hetzelfde als bij a) geldt ook voor de unit manager. Ook deze moet een voorbereiding op de POP gesprekken in kunnen vullen. Het enige verschil is dat de unit manager dit voor meerdere medewerkers moet doen.
- c) Invoeren/wijzigen voorstel salaris en bonus (7)

De unit manager moet vooraf aan het POP gesprek met een medewerker een salaris- en bonusvoorstel in kunnen geven op eenzelfde manier als dit nu in de POP formulieren gebeurt.
- d) Overnemen doelstellingen uit vorige POP verslag (9)

In het POP formulier vooruitkijken worden doelstellingen voor het komende half jaar opgenomen. Het POP formulier terugkijken moet deze doelstellingen uit het vorige vooruitkijken-formulier kunnen kopiëren.
- e) Invoeren/wijzigen doelstellingen (7)

Op het POP formulier vooruitkijken worden doelstellingen ingevoerd voor de komende periode. Dit dient op een eenvoudige manier mogelijk te zijn en de doelstellingen dienen ook wijzigbaar te zijn.
- f) Invoeren/wijzigen gespreksverslag en overige onderdelen POP (7)

Tijdens en na het gesprek dient het mogelijk te zijn nog verscheidene andere velden in het POP formulier in te vullen. Het gaat hierbij om een gespreksverslag en de overige huidige onderdelen uit de POP formulieren.
- g) Accorderen POP verslag door werknemer (7)

Na het gesprek dient de werknemer het POP verslag te accorderen. Mogelijk kan dit door middel van een digitale handtekening, de mogelijkheden hiervoor dienen nog verder te worden onderzocht.
- h) Accorderen POP verslag en voorstel salaris en bonus door UM (7)

Nadat de medewerker het POP verslag heeft geaccordeerd dient de unit manager het verslag, inclusief het salarisvoorstel, goed te keuren. Hierna kan het salaris en bonus voorstel uit het verslag daadwerkelijk uitgevoerd worden.
- i) Automatisch aanpassen arbeidsvoorwaarden (8)

Nadat het verslag is goedgekeurd dient het salaris en bonus voorstel automatisch overgenomen te worden in de administratie. Hierbij dienen de notificaties en andere regels zoals bij 2) gedefinieerd te worden gehanteerd.
- j) Invoeren/wijzigen verslagen eventuele tussentijdse gesprekken (9)

Soms zijn er tussentijdse gesprekken met een medewerker. Het zou erg praktisch zijn als hiervan ook een verslagje wordt opgeslagen, zodat later eventuele gemaakte afspraken nog teruggezocht kunnen worden.

k) Invoeren/wijzigen opleidingshistorie werknemer (6)

De opleidingshistorie van een medewerker kan worden bijgehouden. Hierbij kunnen nieuwe opleidingen van een medewerker worden ingevoerd en worden gewijzigd. Per opleiding wordt de naam van de opleiding, naam instituut, begin en einddatum en diploma behaald of niet bijgehouden.

l) Rapportage openstaande leningen en schulden n.a.v. opleiding (9)

Het is mogelijk om de openstaande leningen per werknemer te bekijken. Ook een overzicht van alle openstaande schulden is te bekijken. De actoren die dit mogen doen zijn de HR-medewerker en de Financiënmedewerker.

m) Rapportage gewerkte uren per project per werknemer (5)

Er moet een rapportage uitgedraaid kunnen worden door de unit manager en door de medewerker zelf welke weergeeft hoeveel uur een werknemer aan welke projecten heeft besteed.

n) Rapportage beoordelingsindicatoren per werknemer (5)

Er moet een rapportage uitgedraaid kunnen worden door de unit manager en door de medewerker zelf welke enkele beoordelingsindicatoren weergeeft. Deze komen van pas tijdens de POP gesprekken. De exacte indicatoren zijn terug te vinden in het gespreksverslag van het gesprek met de unit manager.

o) Rapportage overzicht beoordelingsindicatoren (5)

Er moet door de unit manager een overzicht van de beoordelingsindicatoren over alle medewerkers uitgedraaid kunnen worden. Dit zijn dezelfde indicatoren als bij het vorige punt, maar dan overzichtelijk in een lijst per werknemer weergegeven. Ook worden hierbij enkele totalen en gemiddelden verstrekt.

p) Rapportage overzicht doelstellingen unit (9)

Er moet een overzicht uitgedraaid kunnen worden door zowel de unit manager als HR welke de doelstellingen uit alle POP verslagen van een bepaalde unit weergeeft. Dit kan ervoor zorgen dat overeenkomsten in doelstellingen gevonden worden waardoor mensen met elkaar kunnen samenwerken om doelstellingen te bereiken.

q) Rapportage Curriculum Vitae per werknemer (6)

Met de informatie die beschikbaar is binnen de HR database moet een standaard CV gegenereerd kunnen worden voor een werknemer. Hierin worden enkele personalia van de medewerker opgenomen, zijn opleidingshistorie en zijn projecthistorie. Vervolgens wordt dit als Word document aangeleverd waarna de medewerker of unit manager deze zelf verder kan bewerken.

r) Rapportage totale effect van wijzigingen arbeidsvoorwaarden (7)

Zowel per unit als voor het gehele bedrijf dient inzichtelijk te worden gemaakt wat de (financiële) gevolgen van de POP gesprekken en bijbehorende wijzigingen in de arbeidsvoorwaarden zijn.

6. Digitaal dossier

a) Documentcategorieën toevoegen, wijzigen en verwijderen (6)

De HR-medewerker kan categorieën toevoegen, wijzigen en verwijderen. Voorbeelden van categorieën zijn contracten en belastingformulieren. Niet alle categorieën zijn te verwijderen of wijzigen; enkele categorieën zitten vast in het systeem in verband met de koppeling met de rest van het systeem.

b) Documenten op kunnen slaan, eventueel digitaal ondertekend en gewatermerkt (6)

De HR-medewerker en de unit manager kunnen voor een medewerker documenten opslaan (bijvoorbeeld een contract of een belastingformulier). Deze documenten zijn eventueel digitaal ondertekend en voorzien van een watermerk. Tevens zijn deze documenten ingedeeld in een categorie.

c) Koppeling tussen documenten en relevante overige onderdelen applicatie (9)

Er worden relevante koppelingen gemaakt met de rest van het systeem. Denk bijvoorbeeld aan een link naar opgeslagen contracten vanuit het stuk van het programma dat te maken heeft met POP-gesprekken.

d) E-mails direct naar dossier kunnen sturen om op te slaan (9)

Het is mogelijk om e-mails naar het systeem te sturen die automatisch in het dossier worden opgeslagen om later weer te kunnen bekijken. Dit kan gedaan worden door de unit manager en door de HR-medewerker.

C. Beschikbare technologieën

Microsoft Office SharePoint Server 2007

Het gehele intranet van Tam Tam is op dit moment opgebouwd rondom Microsoft Office SharePoint Server 2007. Deze applicatie biedt geïntegreerd documentmanagement en vele mogelijkheden tot samenwerken tussen verschillende personen en groepen. Eén van de requirements is dat het HRM systeem geïntegreerd moet worden in SharePoint, vandaar dat SharePoint een groot deel van de basis van het systeem vormt. Zo worden voor een deel van de dataopslag de verschillende libraries welke SharePoint standaard biedt gebruikt en worden de takenlijsten uit SharePoint gebruikt om mensen te laten samenwerken.

Microsoft SQL Server 2005

Voor de opslag van de data is het niet altijd het beste om te kiezen voor opslag binnen SharePoint Server. Soms is het handiger om data in een relationele database op te slaan. Voor deze gevallen is gekozen voor MS SQL Server 2005. Tevens staan de huidige gegevens over medewerkers en uren al in een MS SQL Server 2005 database; om de aansluiting hierop te vergemakkelijken is het wenselijk om voor MS SQL Server te kiezen. Verder biedt SQL Server door middel van Reporting Services een mogelijkheid om eenvoudige krachtige rapportages te maken.

Microsoft .NET, C#, ASP.NET en ASP.NET AJAX

SharePoint Server is volledig gebouwd op Microsoft .NET technologie. Het .NET framework van Microsoft is een relatief nieuw framework dat veel gebruikte code voor applicaties bevat, alsmede een *runtime* om de applicaties uit te voeren. Het uitvoeren van applicaties gebeurt in een managed omgeving, vergelijkbaar met de Java Virtual Machine. Tegelijk met het .NET framework is ook de taal C# ontwikkeld, een programmeertaal specifiek bedoeld voor het .NET framework. Binnen Tam Tam is het standaard dat applicaties in C# worden ontwikkeld, met Microsoft .NET als framework. De HRM applicatie zal hier dan ook mee ontwikkeld worden, zoals al naar voren kwam uit de voorwaarden bij de opdracht. Voor de integratie met SharePoint Server is het ook van belang om gebruik te maken van .NET, gezien via .NET SharePoint Server gemakkelijk is aan te spreken vanuit je eigen applicaties.

ASP.NET is een manier om server-side webpagina's (en onderdelen van webpagina's, WebParts) te genereren. Hierbij zal ook gebruik gemaakt worden van ASP.NET AJAX, een extra framework bovenop ASP.NET dat ondersteuning biedt voor Asynchronous Javascript And XML (AJAX). AJAX is een techniek om, terwijl een gebruiker bezig is op een webpagina, te communiceren met de server. Op deze manier is het bijvoorbeeld mogelijk een formulier te versturen zonder de gehele pagina te verversen, wat een betere beleving voor de eindgebruiker geeft.

Web Parts / Return of the SmartPart v1.2

Voor de uitbreiding van SharePoint wordt niet alleen gebruik gemaakt van de mogelijkheden welke SharePoint standaard biedt, maar zullen ook veel zogenaamde *custom* WebParts ontwikkeld worden. WebParts zijn kleine modulaire stukjes webpagina, welke door de gebruiker naar eigen inzicht op een bepaalde plaats gezet kunnen worden. Zo kan bijvoorbeeld het formulier om een medewerker toe te voegen gezien worden als één WebPart. Het ontwikkelen van deze WebParts gebeurt door middel van C# en ASP.NET code. Om het ontwikkelen van WebParts te vereenvoudigen zal er gebruik gemaakt worden van Smartpart. Hiermee is het mogelijk om een ASP.NET User Control (stukje ASP.NET pagina met bijbehorende code) om te

zetten naar een WebPart. De laatste versie (Return of the Smartpart 1.2 bèta) biedt tevens ondersteuning voor ASP.NET AJAX, vandaar dat deze versie gebruikt zal worden waar mogelijk.

Microsoft Windows Workflow Foundation

Bepaalde processen binnen de applicatie zullen duidelijk gericht zijn op samenwerking tussen personen en handelingen die verschillende personen moeten verrichten: een zogenaamde workflow. Voor het inrichten van deze workflows zal gebruik gemaakt worden van Windows Workflow Foundation (WF). WF is één van de nieuwe technieken in Microsoft .NET 3.0 en maakt het mogelijk om overzichtelijk workflows in te richten (in SharePoint designer of Visual Studio 2005) en hier vervolgens ook code achter te hangen om eigen acties uit te voeren. Vervolgens kan een workflow, wanneer deze een specifieke SharePoint workflow is, aan een list in SharePoint gehangen worden, waarna deze beschikbaar is voor de gebruikers.

Microsoft InfoPath 2007 en Forms Server 2007

Voor de diverse formulieren die ingevuld moeten worden zal gebruikt worden gemaakt van Microsoft InfoPath, een Microsoft Office applicatie die op dit moment ook al voor veel formulieren binnen Tam Tam gebruikt wordt. InfoPath maakt het mogelijk om op een eenvoudige manier XML gebaseerde formulieren te creëren welke op een goed integreren met SharePoint Server. Tevens zal gebruikt gemaakt worden van Microsoft Forms Server 2007 om de InfoPath formulieren in de browser te laten renderen, gezien het lastig en onnodig is om voor ieder formulier opnieuw InfoPath te moeten openen.

Microsoft Excel Services

Ten behoeve van de rapportagefunctie van de HRM-applicatie zal veelvuldig gebruikt gemaakt worden van Microsoft Excel Services. Deze services bieden de mogelijkheid om Excel-documenten binnen een browser te renderen. Gezien Excel veel ingebouwde mogelijkheden heeft voor rapportage (denk aan de nu al veel binnen Tam Tam gebruikte draaitabellen/pivot tables) is dit een ideale methode om rapportages te genereren.

D. SharePoint lists, workflows en WebParts

SharePoint Lists

De volgende SharePoint lists worden gebruikt:

TaskList

Deze standaard SharePoint tasklist zal door heel het HRM system gebruikt worden om de taken bij te houden van de verschillende actoren. Ook de workflows zullen hieraan gekoppeld zijn. Door middel van verschillende views zal het mogelijk zijn om op meerdere pagina's deze lijst opnieuw te gebruiken. Zo kan ervoor gekozen worden om een medewerker alleen zijn eigen taken te laten zien of om de HR medewerker alle taken te laten zien. Tevens is het mogelijk om *alerts* bij een lijst in te schakelen zodat een medewerker deze krijgt toegestuurd wanneer er een nieuwe taak wordt toegevoegd. Ook kan deze takenlijst eenvoudig gekoppeld worden aan Microsoft Outlook zodat de taken aan een persoonlijke agenda worden gekoppeld.

EmployeeFileGeneral

De EmployeeFileGeneral is een SharePoint *document library* waarin verschillende soorten documenten kunnen worden opgeslagen. Hierbij gaat het om alle documenten die niet direct gekoppeld zijn aan een workflow, zoals leaseovereenkomsten of formulieren voor de belastingdienst.

EmployeeFileEntrance

EmployeeFileEntrance is een documentlibrary met daaraan gekoppeld een InfoPath *form template*. Dit template zal gebruikt worden tijdens de indiensttreding van personeelsleden om te zorgen dat bijvoorbeeld een useraccount aangemaakt wordt en een bureau gereserveerd wordt. Hieraan zal tevens een workflow gekoppeld zijn om dit proces te begeleiden.

EmployeeFileExit

EmployeeFileExit is een documentlibrary met daaraan gekoppeld een InfoPath *form template*. Dit template zal gebruikt worden tijdens de uitdiensttreding van personeelsleden om te zorgen dat bijvoorbeeld de sleutels ingeleverd worden en het useraccount op non-actief wordt gezet. Hieraan zal tevens een workflow zijn gekoppeld om dit proces te begeleiden.

EmployeeFilePOP

EmployeeFilePOP is een documentlibrary met daaraan gekoppeld het bekende InfoPath *form template* voor het POP proces. Tijdens de halfjaarlijkse evaluatie van de medewerkers zal deze library zorgdragen voor het opslaan van de POP formulieren. Ook zal aan deze library weer een workflow gekoppeld zijn om het gehele POP proces te begeleiden.

Workflows

De volgende workflows worden gebruikt:

EmployeeEntry workflow

De EmployeeEntry workflow wordt gestart wanneer er een nieuwe medewerker wordt aangemaakt. Deze workflow wordt gestart door middel van het aanmaken van een nieuw formulier in de EmployeeFileEntry library. De workflow zet verschillende taken uit naar HR, de unit manager, systeembeheer en financiën, zoals beschreven in de requirements. Al deze taken worden tegelijk aangemaakt. Na het afronden van alle

taken is de workflow compleet. HR kan continue de status van alle taken monitoren; de overige deelnemers zien alleen de taken welke voor hen van belang zijn.

EmployeeExit workflow

De EmployeeExit workflow wordt gestart wanneer een medewerker uit dienst gaat. De workflow wordt gestart door middel van het aanmaken van een nieuw formulier in de EmployeeFileExit library, wat gebeurt na het drukken op een knop bij de arbeidsvoorwaarden van een werknemer. De workflow zet verschillende taken uit naar HR, de Unit Manager, systeembeheer en financiën, zoals beschreven in de requirements. Al deze taken worden tegelijk aangemaakt. Na het afronden van alle taken is de workflow compleet.

EmployeePOP workflow

De EmployeePOP workflow wordt gebruikt tijdens het halfjaarlijkse POP proces binnen Tam Tam. Aan het begin van de POP procedure wordt door HR door middel van een knop in de applicatie de procedure gestart voor alle werknemers (m.u.v. stagiaires). Dit zorgt ervoor dat er een reeks nieuwe POP formulieren wordt aangemaakt in de EmployeeFilePOP library; één formulier per werknemer. De workflow zorgt vervolgens voor de begeleiding van het POP proces zoals reeds beschreven in de requirements.

WebParts

De volgende WebParts worden gebruikt:

EmployeeList

Lijst met medewerkers waarin op naam gefilterd kan worden. Weergegeven velden zijn naam en contactgegevens. Tevens een link naar details, wanneer van toepassing (op basis van groepsrechten van de ingelogde gebruiker).

EmployeeView

Gegevens van de medewerker bestaande uit hoofdzakelijk personalia. Afhankelijk van groepsrechten mogelijk om op bewerken te klikken waardoor sommige (of alle) velden bewerkbaar worden.

EmployeeTerms

Arbeidsvoorwaarden van de medewerker. Voor HR bestaat hier de mogelijkheid om de arbeidsvoorwaarden te wijzigen. Hierbij wordt een datumveld weergegeven voor de ingangsdatum. Tevens zijn notities per wijziging mogelijk voor een eventuele toelichting. Door middel van een *dropdownlist* is het mogelijk om eerdere arbeidsvoorwaarden te bekijken. Deze kunnen uiteraard niet worden gewijzigd.

EmployeeTermsHistory

De historie van de arbeidsvoorwaarden voor een medewerker zijn hierin overzichtelijk weer te geven. Welke kolommen worden weergegeven wordt aangegeven middels een *kolom-selector*.

EmployeeVacationTotals

De huidige status en globale opbouw van de vakantiedagen van een medewerker worden weergegeven.

EmployeeVacationTaken

De opgenomen vakantiedagen van een werknemer worden weergegeven.

EmployeeVactionMutations

De wijzigingen in de aantallen vakantiedagen worden weergegeven. Dit zijn automatisch gegenereerde wijzigingen van bijtellingen aan het eind van elke maand, of handmatige wijzigingen met opgaaf van reden (bijvoorbeeld door indiensttreding halverwege de maand).

EmployeeIllReports

De ziekteperiodes van een werknemer worden hier weergegeven. Het is mogelijk om een werknemer ziek te melden of om hem beter te melden indien hij ziek is.

EmployeeAdd

Deze WebPart voegt een medewerker toe. Om een medewerker toe te voegen moeten personalia en initiële arbeidsvoorwaarden worden ingevoerd. Niet alle gegevens die kunnen worden ingevoerd zijn verplicht bij het toevoegen.

POPStart

Door middel van deze WebPart kan een POP-proces voor het hele bedrijf worden gestart.

IllReports

De ziekteperiodes van alle op dit moment zieke medewerkers worden weergegeven. Het is mogelijk deze vanuit de WebPart direct beter te melden. Ook kunnen ziekmeldingen hier worden toegevoegd.

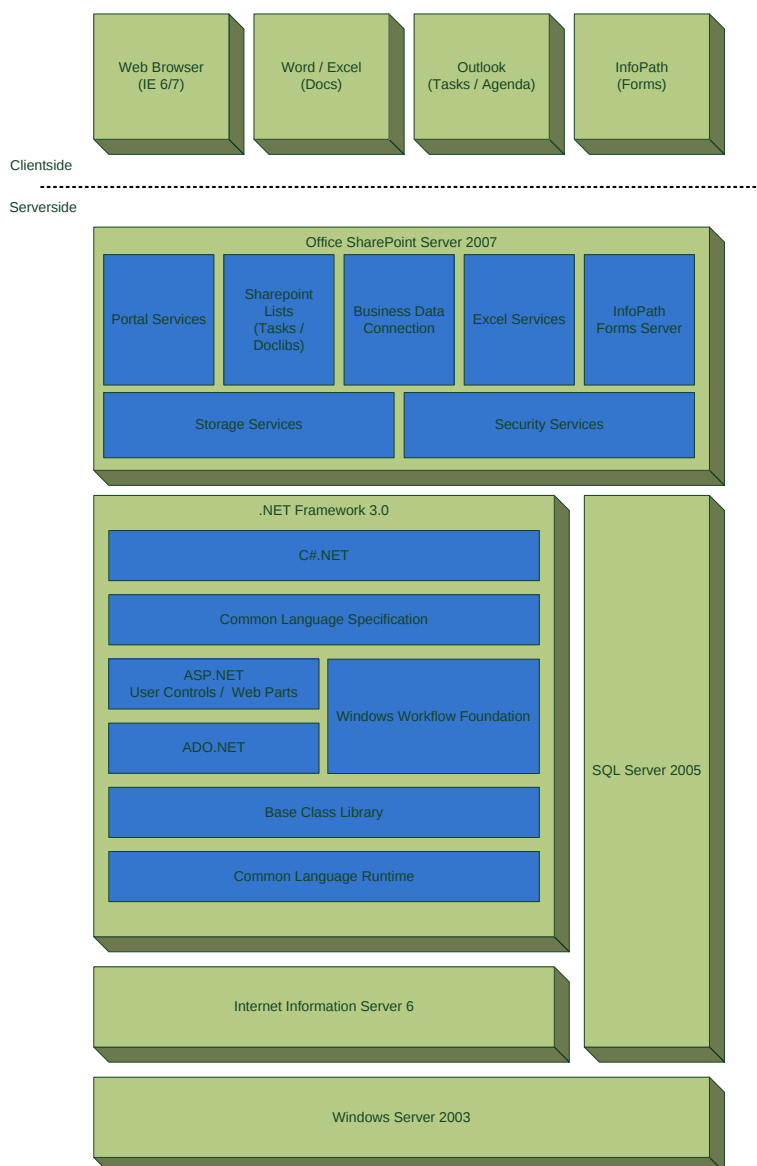
EmployeeHardware

Een tekstveld waarin bijgehouden wordt welke hardware in bezit is van een medewerker.

E. Systeemarchitectuur en objectmodel

Systeemarchitectuur

Er zal nu dieper worden ingegaan op de onderliggende architectuur. Gezien er gebruik wordt gemaakt van ASP.NET zijn veel componenten kant en klaar beschikbaar en wordt code vaak direct aan componenten gehangen in plaats van in aparte klassen gestopt. Dit heeft als voordeel dat er heel snel ontwikkeld kan worden. Wel kan dit bij verkeerd gebruik tot slechtere onderhoudbaarheid van de code leiden.



Figuur 6 - Systeemarchitectuur

De basis

Als basis voor het systeem dienen Windows Server 2003, IIS 6 en SQL Server 2005. Deze componenten verzorgen respectievelijk de functies van OS, webserver en database server.

.NET Framework 3.0

Het .NET Framework verzorgt de gehele programmeeromgeving. Van taal tot basis klassen: veel is hierin ondergebracht. Het framework begint met een zogenaamde Command Language Runtime (CLR). De CLR is verantwoordelijk voor het vertalen van de aanroepen uit de (managed) .NET omgeving naar het operating systeem. Hierboven komt de base class library, welke veel van de basisfunctionaliteit van .NET bevat. Hierin is onder andere de security welke .NET standaard bevat, zoals *code signing* en *trust*, terug te vinden. ADO.NET zorgt vervolgens voor de communicatie met de database en ASP.NET verzorgt de executie van dynamische webpagina's. Windows Workflow Foundation is een innovatie in .NET 3.0 welke de mogelijkheid biedt om workflows te definiëren, waardoor op een hoog niveau businessprocessen gemodelleerd kunnen worden. De Common Language Specification (CLS) zorgt er vervolgens voor dat een generieke taal als C# met dit framework kan samenwerken.

SharePoint

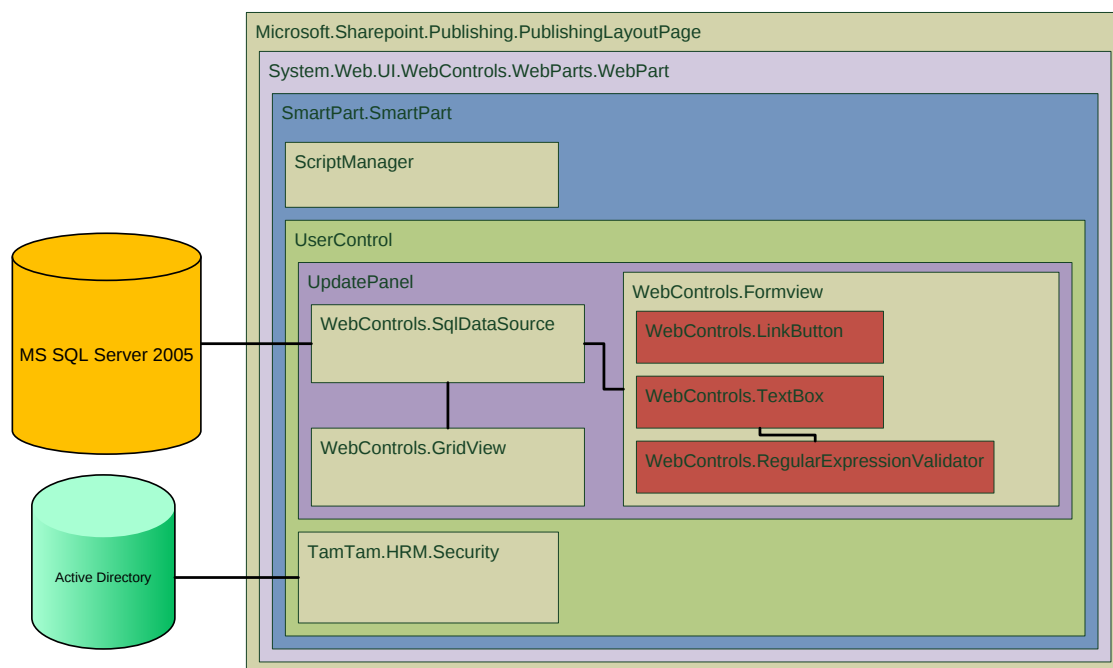
Bovenop het .NET framework is SharePoint gebouwd. Naast de portal functionaliteit welke deze standaard biedt en waarin het aanmaken van pagina's geregeld is, zullen ook de verschillende SharePoint dataopslag methoden gebruikt worden. Excel Services en InfoPath Forms Server zijn al eerder besproken. Business Data Connection dient om (read-only) verbindingen naar backend databases te maken en kan handig gebruikt worden voor het weergeven van overzichten van data welke niet binnen het SharePoint datamodel is opgeslagen.

Client-side

De kant van de gebruiker wordt gevormd door een webbrowser (bij voorkeur Microsoft Internet Explorer) en het Microsoft Office System. Voor documenten zullen Word en Excel gebruikt worden, Outlook wordt voor de takenlijsten en agenda gebruikt en InfoPath dient voor de formulieren welke binnen de applicatie worden gebruikt. Waar mogelijk worden deze formulieren echter via de InfoPath Forms Server aan de browser aangeboden.

Objectmodel

Een strak objectmodel valt erg lastig te beschrijven. Dit gezien er veel gebruik wordt gemaakt van standaard componenten die binnen .NET worden teruggevonden, welke vervolgens alleen nog "in elkaar geklikt" hoeven te worden. In het volgende diagram wordt geprobeerd een indicatie te geven van hoe de gemiddelde pagina in elkaar steekt.



Figuur 7 - Objectmodel

PublishingLayoutPage

Dit Page object is het basisobject binnen SharePoint. Het regelt het aanmaken van de verschillende objecten en verzorgt de communicatie met de browser.

WebPart

Binnen het Page object is een WebPart object te vinden. Dit is een generiek object welke een Web Part kan bevatten. Het regelt het opvragen, aanroepen en renderen van het Web Part.

SmartPart

Zoals eerder vermeld zal er binnen het systeem veelvuldig gebruik gemaakt worden van SmartPart. Dit betekent dat nagenoeg alle gebruikte WebParts simpelweg een SmartPart object zullen zijn. Dit object regelt vervolgens het aanroepen van een UserControl en plaatst desgewenst een ScriptManager op de pagina.

ScriptManager

Het ScriptManager object is benodigd voor ASP.NET AJAX. Dit object regelt het aanmaken van de benodigde Javascripts voor de ondersteuning van AJAX. Ook is dit het object welke AJAX postbacks verwerkt en de juiste methoden aanroept om data via Javascript naar de browser terug te sturen.

UserControl

Zoals vermeld zullen er User Controls worden ontwikkeld welke binnen SmartPart worden geladen. Dit object stelt één User Control voor en is vergelijkbaar met een Page object, maar dan voor een deel van de pagina.

UpdatePanel

Niet alleen een ScriptManager is nodig om ASP.NET AJAX te gebruiken, maar ook een UpdatePanel is vereist. Dit object zorgt ervoor dat de verschillende componenten welke hij bevat niet een echte postback zullen doen (waarbij de pagina refreshed) maar een AJAX postback zullen doen, waardoor de refresh onmerkbaar is voor de gebruiker. Bij een postback zal de gehele inhoud van het UpdatePanel worden ververs.

SqlDataSource

De SqlDataSource zorgt voor het maken van een verbinding met de database. Nadat de verbinding is gemaakt kunnen er 4 queries worden gedaan: SELECT, INSERT, UPDATE en DELETE. Deze queries zijn binnen de datasource te definiëren. Wanneer een SELECT query wordt uitgevoerd levert dit een DataView op welke direct te koppelen is aan een Gridview.

Gridview

De Gridview zorgt voor de weergave van de databaseresultaten uit de SqlDataSource. Het toont een overzicht van de verschillende rijen uit de database en biedt wanneer gewenst een mogelijkheid om rijen te bewerken of te verwijderen.

Formview

Een Formview is ook te koppelen aan een SqlDataSource. Een Formview biedt de mogelijkheid om één rij uit een tabel te bewerken of een nieuwe rij toe te voegen. Hoe het formulier eruit ziet is volledig zelf te definiëren.

LinkButton

LinkButtons zijn knoppen welke worden gebruikt om een actie uit te voeren.

TextBox

De TextBoxes worden door middel van een Bind() call gekoppeld aan een kolom uit een database welke vervolgens weergegeven wordt of te bewerken is.

RegularExpressionValidator

Binnen de applicatie zullen op verschillende plaatsen de ingevoerde waarden gecontroleerd worden. In het geval van dit voorbeeld wordt dit gedaan door middel van een RegularExpressionValidator. Dit component koppel je aan een TextBox, waarna deze aan bepaalde voorwaarden moet voldoen. Naast de RegularExpressionValidator zullen er nog andere Validators gebruikt worden. Deze zijn allemaal standaard binnen .NET aanwezig.

Security

Ten behoeve van de beveiliging van de applicatie zal een eigen klasse ontwikkeld worden welke door de gehele applicatie gebruikt zal worden voor de beveiliging. Deze klasse is bewust eenvoudig gehouden; zo gebeurt het daadwerkelijk afdwingen van de beveiliging in de code zelf. Weliswaar is deze methode minder overzichtelijk, maar door de opzet van de applicatie ontkomen we hier niet aan (zo zullen ook verschillende rechten vanuit SharePoint geregeld worden). Deze klasse zal de volgende methodes kennen:

HRMSecurity
+IsSuperuser() +IsHRM() +IsUnitManager() +IsUnitManager(int unitId) +IsSystemAdministrator() +IsEmployee(int employeeId) +getUnit()

Figuur 8 - Security object

F. Data dictionary

Voor de gebruikers zijn de volgende afkortingen gebruikt:

- HR: Human Resources
- F: Finance
- SA: Systeembeheerders
- UM: Unit Manager (van de betreffende medewerker)
- MW: De medewerker zelf

Voor de rechten zijn de volgende tekens gebruikt:

- X: Lees en schrijftoegang
- 0: Leestoegang

Employees			Rechten				
Naam	Type	Beschrijving	HR	F	SA	UM	MW
Login Name	String	De loginnaam van de medewerker op het Tam Tam netwerk	X	0	0	0	0
Initials	String	De initialen van de medewerker	X	0	0	0	0
First Name	String	De voornaam van de medewerker	X	0	0	0	0
Infix	String	De tussenvoegsels van de medewerker	X	0	0	0	0
Last Name	String	De achternaam van de medewerker	X	0	0	0	0
Date Of Birth	Date	De geboortedatum van de medewerker	X	0	0	0	0
Address	String	Het adres van de medewerker (straat)	X	0	0	0	X
Zip Code	String	De postcode van de medewerker	X	0	0	0	X
City	String	De woonplaats van de medewerker	X	0	0	0	X
Km Home-Work	Decimal	De afstand tussen het huis van de medewerker en Tam Tam (t.b.v. reiskostenvergoeding)	X	0	0	0	0
Phone Work	String	Het toestelnummer bij Tam Tam van de medewerker	X	0	0	0	X
Phone Mobile	String	Het mobiele nummer van de medewerker	X	0	0	0	X
Email Address	String	Het e-mailadres van de medewerker	X	0	0	0	X
Sex	Enum	Het geslacht van de medewerker	X	0	0	0	0
Marital Status	Enum	De burgerlijke staat van de medewerker	X	0	0	0	X
Bank Account Number	String	Het bankrekeningnummer van de medewerker (eventueel IBAN/BIC voor buitenlandse rekening)	X	0	0	0	X
Passport Number	String	Het paspoort / ID kaart nummer van de medewerker.	X	0	0	0	0

SSN	String	Het Sofinummer (of Burgerservicenummer) van de medewerker	X	0	0	0	0
Active	Bool	Een flag welke aangeeft of de medewerker nog actief is bij Tam Tam	X	0	0	0	0
PWCnr	String	Het referentienummer bij PWC van de medewerker t.b.v. de loonadministratie	X	0	0	0	0
LDAP	String	De entry van de medewerker in de Active Directory t.b.v. geautomatiseerde tools	X	0	0	0	0
GP Name	String	De naam van de huisarts van de medewerker	X	0	0	0	X
GP Phone	String	Het telefoonnummer van de huisarts van de medewerker	X	0	0	0	X
GP Phone Mobile	String	Het mobiele nummer van de huisarts van de medewerker	X	0	0	0	X
Allergies	String	Eventuele allergieën van de medewerker	X	0	0	0	X
Additional Info	String	Aanvullende informatie	X	0	0	0	0
Work Permit Expiry	Date	De verloopdatum van de werkvergunning van de medewerker	X	0	0	0	0
Emergency Contact Info	String	Personen waar contact mee moet worden opgenomen bij een noodgeval	X	0	0	0	X
Hardware	String	Een beschrijving van de hardware die de medewerker in bezit heeft.	X	0	X	0	0
Manager	Employee	De manager van de medewerker	X	0	0	0	0
Buddy	Employee	De buddy van de medewerker	X	0	0	0	0

Tabel 2 – Data en rechten Employees

FamilyMembers			Rechten				
Naam	Type	Beschrijving	HR	F	SA	UM	MW
First Name	String	De voornaam van het familielid	X	0	0	0	X
Infix	String	De tussenvoegsels van het familielid	X	0	0	0	X
Last Name	String	De achternaam van het familielid	X	0	0	0	X
Date Of Birth	Date	De geboortedatum van het familielid (t.b.v. notificatie HR om kaartje te sturen)	X	0	0	0	X
Relation	Enum	De relatie tussen de medewerker en het familielid (partner of kind)	X	0	0	0	X

Tabel 3 – Data en rechten FamilyMembers

<u>IllReports</u>			Rechten				
Naam	Type	Beschrijving	HR	F	SA	UM	MW
Date Started	Date	Startdatum van de ziekmelding	X	0		0	0
Date End	Date	Einddatum van de ziekmelding	X	0		0	0
Comment	String	Eventuele opmerking	X	0		0	0

Tabel 4 - Data en rechten IllReports

<u>VacationHours</u>			Rechten				
Naam	Type	Beschrijving	HR	F	SA	UM	MW
Amount	Decimal	Hoeveelheid uren behorende bij de vakantieuren boeking	X	0		0	0
Description	String	Optionele beschrijving bij de mutatie	X	0		0	0
Timestamp	DateTime	Het tijdstip waarop de boeking is gemaakt	X	0		0	0

Tabel 5 - Data en rechten VacationHours

<u>EmploymentContracts</u>			Rechten				
Naam	Type	Beschrijving	HR	F	SA	UM	MW
Date Start	Date	Datum waarop de arbeidsovereenkomst is ingegaan	X	0		0	0
Date End	Date	Datum waarop de arbeidsovereenkomst is beëindigd	X	0		0	0
Type	Enum	Het soort arbeidsovereenkomst (bepaalde tijd, onbepaalde tijd, stage, oproepkracht)	X	0		0	0

Tabel 6 - Data en rechten EmploymentContracts

<u>TermsOfEmployment</u>			Rechten				
Naam	Type	Beschrijving	HR	F	SA	UM	MW
FTE Salary	Decimal	Volgtijd salaris	X	0		0	0
FTE Work	Decimal	Aantal FTE voor loon	X	0		0	0
FTE Leave	Decimal	Aantal FTE voor verlofsparen	X	0		0	0
Pension Option	Enum	Pensioenkeuze (geen / Tam Tam bijdrage / Maximaal / Vast bedrag)	X	0		0	0
Survivors Pension	Bool	Deelname aan nabestaandenpensioen	X	0		0	0
Vacation Hours	Int	Aantal vakantieuren per maand	X	0		0	0
Leased Car	Bool	Leaseauto bij contract (ja/nee)	X	0		0	0
Leased Car Compensation	Decimal	Tam Tam compensatie voor leaseauto	X	0		0	0
Saving Arrangement	Enum	Spaarregeling van medewerker (geen, spaarloon of levensloop)	X	0		0	0
Saving Arrangement Account	Int	Bankrekening van spaarregeling	X	0		0	0

Fietsplan Expiry	Date	Datum waarop het Fietsplan contract afloopt	X	0		0	0
Fietsplan Amount	Decimal	Bedrag waarvoor Fietsplan fiets is gekocht	X	0		0	0
Date Entry	Date	Datum invoer arbeidsvoorwaarden	X	0		0	0
Date Active	Date	Datum van inwerkingtreding arbeidsvoorwaarden	X	0		0	0
Change Comment	String	Optionele opmerking bij de wijziging	X	0		0	0

Tabel 7 - Data en rechten TermsOfEmployment

<u>TasksList</u>			Rechten				
<u>Naam</u>	<u>Type</u>	<u>Beschrijving</u>	<u>HR</u>	<u>F</u>	<u>SA</u>	<u>UM</u>	<u>MW</u>
Tasks	Task	De taken welke de medewerker op dit moment nog open heeft staan t.b.v. de verschillende workflows	0				X

Tabel 8 - Data en rechten TasksList

<u>Task</u>			Rechten				
<u>Naam</u>	<u>Type</u>	<u>Beschrijving</u>	<u>HR</u>	<u>F</u>	<u>SA</u>	<u>UM</u>	<u>MW</u>
Description	String	Beschrijving van de taak	0				X
Due Date	Date	Datum waarop de taak af dient te zijn.	0				X
Status	Enum	Status van de taak (new / in progress / completed)	0				X

Tabel 9 - Data en rechten Task

<u>EmployeeFile</u>			Rechten				
<u>Naam</u>	<u>Type</u>	<u>Beschrijving</u>	<u>HR</u>	<u>F</u>	<u>SA</u>	<u>UM</u>	<u>MW</u>
Filename	String	Naam van het bestand	X	0		0	0
Content	File	Het bestand zelf, dit bestand kan vanalles zijn en kan door de gebruiker zelf geupload worden. Het kan bijvoorbeeld het arbeidscontract zijn, maar ook een loonbelastingverklaring.	X	0		0	0
Type	String	Het type bestand (.doc/.pdf/.txt/.xml)	X	0		0	0

Tabel 10 - Data en rechten EmployeeFile

De volgende datatypen bevatten bestanden zoals deze op dit moment al gebruikt worden binnen het bedrijf. Zie hiervoor ook Appendix A. De wens van de opdrachtgever was de bestaande bestanden te hergebruiken.

EmployeeEntryFile (overerft EmployeeFile)			Rechten				
Naam	Type	Beschrijving	HR	F	SA	UM	MW
Content (override)	File	De checklist zoals deze op dit moment wordt gebruikt binnen het bedrijf bij indiensttreding van een medewerker.	X	0	0	0	0
WorkflowState	Enum	Flag welke aangeeft wat de status van de bijbehorende workflow is, en daarmee aangeeft of de indiensttredingprocedure al is afgerond	X	0	0	0	0

Tabel 11 - Data en rechten EmployeeEntryFile

EmployeeExitFile (overerft EmployeeFile)			Rechten				
Naam	Type	Beschrijving	HR	F	SA	UM	MW
Content (override)	File	De checklist zoals deze op dit moment wordt gebruikt binnen het bedrijf bij uitdiensttreding van een medewerker.	X	0	0	0	0
WorkflowState	Enum	Flag welke aangeeft wat de status van de bijbehorende workflow is, en daarmee aangeeft of de uitdiensttredingprocedure al is afgerond	X	0	0	0	0

Tabel 12 - Data en rechten EmployeeExitFile

EmployeePOPFile (overerft EmployeeFile)			Rechten				
Naam	Type	Beschrijving	HR	F	SA	UM	MW
Content (override)	File	Het formulier zoals deze op dit moment wordt gebruikt binnen het bedrijf bij evaluatie van een medewerker.	X	0		0	0
WorkflowState	Enum	Flag welke aangeeft wat de status van de bijbehorende workflow is, en daarmee aangeeft of de POP procedure al is afgerond	X	0		0	0

Tabel 13 - Data en rechten EmployeePOPFile

G. Schermen

Op de volgende pagina's zijn onze schetsen van het grafisch ontwerp te vinden.