

# Afstudeerverslag Ontwikkeling functionaliteit voor HARMONY WebAccess

Student: Alex Nap  
Studentnr: 98000158  
Periode: 2003-2-1

# Ontwikkeling functionaliteit voor HARMONY WebAccess

Product:	Afstudeerverslag
Datum:	januari 2004
Plaats:	Gouda
Door:	A.H. Nap
Studentnr:	98000158
Periode:	2003-2.1 (start 1 september 2003)
Instelling:	Haagse Hogeschool
Opleiding:	Informatica & Informatiekunde
Opleidingsvariant:	Management & Beheer van InformatieTechnologie
Examinatoren:	Dhr. L.C.M. van Koppen Dhr. A.J. Toet
Afstudeerbedrijf:	ORTEC Arbeidstijdmanagement bv
Bedrijfsmentoren:	Dhr. F. Bos Dhr. J. Ensing Dhr. P. van 't Hoff

## Referaat

Nap, A.H., afstudeerverslag, "Ontwikkeling functionaliteit voor HARMONY WebAccess", ORTEC Arbeidstijdmanagement bv, Gouda januari 2004.

Dit document bevat het afstudeerverslag van Alex Nap, student van de Haagse Hogeschool, opleiding Informatica & Informatiekunde en opleidingsvariant Management en Beheer van InformatieTechnologie(MBIT). Dit verslag bevat informatie over de afstudeerperiode bij het bedrijf ORTEC Arbeidstijdmanagement bv te Gouda. Deze afstudeerperiode liep van 1 september 2003 tot en met 9 januari 2004.

Descriptoren:

- Arbeidstijdmanagement
- Client-server
- DCOM
- Delphi
- DSDM
- Gouda
- HARMONY
- HARMONY WebAccess
- HTML
- ISAPI DLL
- ORTEC
- SDM
- Standaard ARTM
- UML
- WebSnap

## **Voorwoord**

Het afstuderen bij ORTEC Arbeidstijdmanagement (ARTM) bv, is een aangename periode geweest voor mij. Tijdens de afstudeerperiode heb ik veel steun en hulp gekregen van een aantal personen. Ik ben dankbaar voor alle hulp en steun die ik gekregen heb van de medewerkers van ARTM en dan in het bijzonder mijn naaste begeleiders dhr. P. van 't Hoff, dhr. F. Bos en dhr. J. Ensing voor de tijd, inzet en hulp bij de inhoudelijke kant van de opdracht en de verslaggeving.

Ook wil ik mijn examinatoren, dhr. L.C.M. van Koppen en dhr. A.J. Toet, bedanken voor hun commentaar op tussentijds ingeleverd werk, hun tips, aanbevelingen en aansturing vanuit de Haagse Hogeschool.

A.H. Nap

Gouda, januari 2004

# Inhoudsopgave

<b>1. INLEIDING .....</b>	<b>5</b>
<b>2. AFSTUDEEROPDRACHT .....</b>	<b>6</b>
2.1 AANLEIDING VAN DE OPDRACHT .....	6
2.2 PROBLEEMSTELLING.....	6
2.3 DOELSTELLING.....	6
2.4 OP TE LEVEREN PRODUCTEN .....	7
<b>3. PLAN VAN AANPAK .....</b>	<b>8</b>
3.1 OPDELING AFSTUDEERPROJECT .....	8
3.1.1 Definitiefase .....	8
3.1.2 Bouwfase .....	9
3.1.3 Implementatiefase .....	9
3.2 PLANNING .....	9
<b>4. METHODE &amp; TECHNIKEN .....</b>	<b>11</b>
4.1 STANDAARD VAN ARTM.....	11
4.1.1 Definitiefase .....	11
4.1.2 Bouwfase .....	12
4.1.3 Implementatiefase .....	14
4.2 VERGELIJKING ANDERE METHODEN.....	15
4.2.1 SDM .....	15
4.2.2 DSDM.....	17
4.2.3 Verschillen en overeenkomsten met standaard ARTM.....	19
<b>5. BEDRIJFSPROFIEL.....</b>	<b>21</b>
5.1 ORGANISATIE ORTEC BV.....	21
5.2 ORGANISATIE ORTEC ARBEIDSTIJDMANAGEMENT BV .....	22
5.3 FUNCTIONEREN VAN AFSTUDEERDER IN DE ORGANISATIE.....	22
<b>6. ANALYSE HARMONY .....</b>	<b>24</b>
6.1 FUNCTIONELE ANALYSE .....	24
6.2 TECHNISCHE ANALYSE .....	27
6.3 HARMONY WEBACCESS.....	29
6.3.1 Functionele analyse.....	29
6.3.2 Technische analyse .....	32
<b>7. FUNCTIONEEL ONTWERP.....</b>	<b>35</b>
7.1 GEWENSTE FUNCTIONALITEIT.....	35
7.2 FUNCTIONALITEIT UITGEWERKT .....	37
7.3 MENUSTRUCTUUR HARMONY WEBACCESS .....	39
7.4 CLASS DIAGRAM.....	41
<b>8. TECHNISCH ONTWERP.....</b>	<b>43</b>
8.1 TECHNISCHE SPECIFICATIES .....	43
8.2 CLASS DIAGRAM.....	46
<b>9. BOUW.....</b>	<b>49</b>
9.1 BOUW VAN DE FUNCTIONALITEIT .....	49
9.2 SYSTEEMTEST.....	54
9.3 IMPLEMENTATIEFASE .....	55
<b>10. EVALUATIE AFSTUDEERPROJECT .....</b>	<b>57</b>
10.1 PRODUCTEVALUATIE .....	57

10.1.1 Tussenproducten .....	57
10.1.2 Eindproduct .....	58
10.2 PROCESEVALUATIE .....	58
10.2.1 Standaard van ARTM .....	58
10.2.2 Gebruikte technieken .....	58
10.2.3 Planning .....	59
<b>11. AANBEVELINGEN VOOR DE TOEKOMST .....</b>	<b>60</b>
<b>LITERATUURLIJST .....</b>	<b>62</b>
<b>WOORDENLIJST .....</b>	<b>63</b>
<b>FIGURENLIJST .....</b>	<b>64</b>
<b>EXTERNE BIJLAGEN .....</b>	<b>65</b>
<b>EXTERNE BIJLAGE I OPDRACHTOMSCHRIJVING .....</b>	<b>66</b>
<b>EXTERNE BIJLAGE II WHITEPAPER HARMONY .....</b>	<b>70</b>
<b>EXTERNE BIJLAGE III PLAN VAN AANPAK .....</b>	<b>99</b>
<b>EXTERNE BIJLAGE IV CHECKLISTS STANDAARD ARTM .....</b>	<b>12</b>
<b>EXTERNE BIJLAGE V FUNCTIONEEL ONTWERP .....</b>	<b>16</b>
<b>EXTERNE BIJLAGE VI TECHNISCH ONTWERP .....</b>	<b>68</b>

## 1. Inleiding

Dit verslag beschrijft de werkzaamheden uitgevoerd tijdens de afstudeerperiode 2003-2.1 bij ORTEC Arbeidstijdmanagement bv (ARTM). Bij ARTM wordt het softwarepakket HARMONY ontwikkeld. HARMONY is een softwarepakket dat gericht is op het zo efficiënt mogelijk inzetten van personeel rekening houdend met onder andere persoonlijke wensen, CAO's en andere wetgeving. Een onderdeel van HARMONY is een webclient (HARMONY WebAccess) waarmee medewerkers de mogelijkheid hebben om bijvoorbeeld hun rooster te bekijken. De mogelijkheden van HARMONY WebAccess waren echter beperkt. Om de gebruikers meer mogelijkheden te bieden was de opdracht gegeven om de webclient verder te ontwikkelen. Hierbij ben ik begeleid door de bedrijfsmentoren dhr. P. van 't Hoff, dhr. F. Bos en dhr. J. Ensing. De ontwikkelingen die gewenst waren, staan beschreven in hoofdstuk 2. In dit hoofdstuk staat ook beschreven hoe de opdracht tot stand is gekomen, wat het doel van de opdracht was en de producten die ik wilde gaan opleveren. Voordat er met de opdracht begonnen kon worden heb ik een Plan van Aanpak opgesteld. Hoofdstuk 3 geeft een beeld van hoe ik het project heb aangepakt en mijn keuze voor deze aanpak. Hoofdstuk 4 beschrijft de methode die ik gebruik heb voor de aanpak van mijn afstudeeropdracht. De organisatie waar mijn afstudeeropdracht heeft plaats gevonden, is beschreven in hoofdstuk 5. Als eerste wordt een globale beschrijving van de hele organisatie en de afdeling ARTM gegeven, waarna wordt beschreven wat mijn plaats was binnen de organisatie. Binnen mijn opdracht heb ik de werking van de functies in HARMONY en HARMONY WebAccess geanalyseerd die van belang waren voor het ontwikkelen van de gewenste functionaliteit. De beschrijving van deze analyse staat in hoofdstuk 6. In hoofdstuk 7 wordt uiteengezet hoe het Functioneel Ontwerp tot stand is gekomen en van welke technieken en bronnen gebruik gemaakt is. Het Technisch Ontwerp is beschreven in hoofdstuk 8. Hoofdstuk 9 beschrijft hoe de ontwikkeling en het testen van HARMONY WebAccess is verlopen. In hoofdstuk 10, worden de opgeleverde producten en de procesgang geëvalueerd. Hier vindt een terugkoppeling plaats naar de beginsituatie en gekeken in hoeverre de doelstelling is gehaald. Als laatste hoofdstuk, hoofdstuk 11, wordt vooruitgekeken op de verdere ontwikkeling van HARMONY WebAccess, wat er wenselijk is en wat mijn aanbevelingen zijn. In het laatste hoofdstuk, Achter in het verslag heb ik de volgende bijlagen opgenomen:

- Literatuurlijst;
- Woordenlijst;
- Figurenlijst

Als externe bijlagen zijn de volgende producten opgenomen:

- Opdrachtoomschrijving;
- Whitepaper HARMONY;
- Plan van Aanpak;
- Checklists standaard ARTM;
- Functioneel Ontwerp;
- Technisch Ontwerp.

## **2. Afstudeeropdracht**

Dit hoofdstuk beschrijft wat de afstudeeropdracht was die ik heb uitgevoerd bij ARTM. Paragraaf 2.1 beschrijft wat de aanleiding was voor de afstudeeropdracht. Paragraaf 2.2 geeft een beeld van de problemen die opgelost moesten worden. Het doel van de afstudeeropdracht staat beschreven in paragraaf 2.3. In paragraaf 2.4 staat een overzicht van de producten waarmee ik het doel kon bereiken en hoe ik aan deze producten ben gekomen. Externe bijlage I toont de originele beschrijving van de afstudeeropdracht, zoals die is goedgekeurd door de afstudeercoördinator, de examinatoren van de Haagse Hogeschool en de afdeling ARTM.

### **2.1 Aanleiding van de opdracht**

HARMONY WebAccess was een prototype waar men bij ARTM graag mee verder wilde. Echter door gebrek aan tijd en kennis van web programmeren bij ARTM, kwam men er niet aan toe om het verder te ontwikkelen. HARMONY WebAccess is ontwikkeld met de bedoeling planners te ontlasten van een aantal tijdrovende taken. De gebruikers van HARMONY bestaan grotendeels uit planners. Het is de taak van de planner om ervoor te zorgen dat alle medewerkers weten wat voor werkzaamheden ze in een bepaalde periode moeten doen. Het resultaat hiervan wordt weergegeven in een rooster. Kenmerkend voor een rooster is zijn dynamische karakter. Door bijvoorbeeld ziekte en / of verlof kan het nodig zijn om vervanging te regelen, wat betekent dat het rooster gewijzigd moet worden. Het zou de planner veel tijd besparen als medewerkers zelf sommige acties in het systeem zouden kunnen doen. Bij aanvang van het afstudeerproject draaide er een prototype van HARMONY WebAccess in een testomgeving bij een klant. Deze klant had al aangegeven verder te willen met het prototype. Met het oog op de concurrentiepositie en het voldoen aan de wensen van de klant was het wenselijk zo snel mogelijk verder te gaan met de ontwikkeling van HARMONY WebAccess.

### **2.2 Probleemstelling**

Bij aanvang van het afstudeerproject waren de mogelijkheden van HARMONY WebAccess nogal beperkt. Een medewerker kon bijvoorbeeld wel zijn rooster bekijken, maar had niet de mogelijkheid om via HARMONY WebAccess aan te geven hoeveel uren hij had gewerkt in een bepaalde periode. Dit moest hij schriftelijk doorgeven aan de planner, waarna de planner alle urenregistraties invoerde in HARMONY. De wens van de opdrachtgever, de heer P. van 't Hoff, was om de medewerker meer interactie mogelijkheden te geven met HARMONY, zodat er tijd bespaard kon worden voor de planner. Het was mijn taak om uit te zoeken welke functionaliteit ontwikkeld moest worden om aan de wens van de opdrachtgever te kunnen voldoen. In hoofdstuk 3 wordt beschreven hoe ik dit heb aangepakt.

### **2.3 Doelstelling**

Het doel van de afstudeeropdracht was het verder ontwikkelen van HARMONY WebAccess. Aangezien dit geen meetbare doelstelling was, heb ik aan het begin van de afstudeeropdracht een brainstormsessie (zie



paragraaf 7.1) gehouden om de gewenste functionaliteit en de prioriteit van de gewenste functionaliteit te bepalen. De gewenste functionaliteit was om medewerkers met behulp van HARMONY WebAccess de mogelijkheid te geven om:

- Hun urenverantwoording in te kunnen voeren;
- Hun wensen te bekijken;
- Een overzicht van ruilingen te kunnen zien;
- Aan te kunnen geven dat ze hun rooster of mutatie(s) hebben gezien.

## **2.4 Op te leveren producten**

Om te bepalen wat voor producten ik moest opleveren, heb ik de methode geraadpleegd die ik voor dit project ging gebruiken. In paragraaf 4.1 wordt de methode toegelicht en uitgelegd waarom ik voor deze methode heb gekozen. Hoofdstuk 3 beschrijft waarvoor ik deze producten nodig had en hoe ik ze kon gebruiken. De producten waren:

- Plan van aanpak, beschrijft de aanpak van het afstudeerproject;
- Functioneel ontwerp, beschrijft de functionele eisen waaraan het prototype moet voldoen;
- Technisch ontwerp, beschrijft de technische specificaties van het prototype;
- Prototype van HARMONY WebAccess, oude versie van HARMONY WebAccess uitgebreid met de gewenste functionaliteit;
- Testplan, beschrijft de situaties die getest moeten worden;
- Gebruikershandleiding, geeft gebruiker uitleg over de werking van HARMONY WebAccess.

## 3. Plan van Aanpak

Dit hoofdstuk beschrijft hoe de afstudeeropdracht is aangepakt en waarom voor deze manier is gekozen. De aanpak is gebaseerd op de standaard zoals die bij ARTM wordt gebruikt (zie paragraaf 4.1). Paragraaf 3.1 beschrijft de indeling van het project, de activiteiten die ingepland waren en waarom deze activiteiten uitgevoerd moesten worden. Paragraaf 3.2 toont de planning die bij aanvang van het project is opgesteld.

### 3.1 Opdeling afstudeerproject

Het afstudeerproject was opgebouwd in drie fasen, conform de standaard van ARTM. De drie fasen waren de Definitiefase, Bouwfase en Implementatiefase. Per paragraaf is een fase beschreven.

#### 3.1.1 *Definitiefase*

Het doel van de Definitiefase was het achterhalen van de gewenste functionaliteit en de prioriteit van elke functionaliteit. Dit heb ik gedaan door middel van een brainstormsessie. Om mezelf voor te bereiden op de opdracht heb ik me georiënteerd op de organisatie, de ontwikkelomgeving, HARMONY en HARMONY WebAccess. Dit heb ik gedaan door het bestuderen van de beschikbare documentatie en het houden van een brainstormsessie. De documentatie die ik bestudeerd heb, zijn:

- Website en intranet van ORTEC;
- Help-bestanden van de gebruikte programmeeromgeving Delphi 6;
- Afstudeerverslag. Een verslag van een student die het eerste prototype van HARMONY WebAccess heeft ontwikkeld;
- Gebruikershandleiding HARMONY.

Voor dit project had ik een interne gebruikersgroep tot mijn beschikking. Deze groep bestond uit mensen van ARTM die veel contact hebben met klanten en zo de vraag van de klanten kennen. Met deze groep heb ik gebrainstormd over de gewenste functionaliteit. De reden dat ik voor een brainstormsessie heb gekozen en bijvoorbeeld niet voor een enquête of interviews, was omdat naar mijn idee met een brainstormsessie het snelst resultaat geboekt kon worden. Een enquête was geen optie omdat de geënquêteerde vragen krijgt waarbij hij een keuze kan maken uit een aantal antwoorden. De geënquêteerde heeft zodoende niet de mogelijkheid zijn eigen mening te geven en het was juist mijn bedoeling de mening te horen van de gebruikersgroep. De keuze moest dus gemaakt worden tussen het houden van een brainstormsessie of interviews. De voordelen van een brainstormsessie in vergelijking met interviews waren:

- Één afspraak nodig om alle betrokkenen te spreken;
- Één resultaat. Geen extra tijd nodig om de resultaten te vergelijken en geen extra afspraak nodig om de prioriteit van de gewenste functionaliteit vast te stellen.

Een nadeel had kunnen zijn dat een betrokkene zijn ideeën niet kwijt kon omdat anderen steeds aan het woord waren. Dit heb ik afgevangen door erop te letten dat iedereen aan het woord kwam. De resultaten van de brainstormsessie zijn verwerkt in het document Functioneel Ontwerp. Het

Functioneel Ontwerp wordt beschreven in hoofdstuk 7 en was de basis voor de Bouwfase.

### **3.1.2           Bouwfase**

De Bouwfase was bedoeld om de functionaliteit zoals beschreven in het Functioneel Ontwerp te ontwikkelen. De Bouwfase heb ik ingedeeld in incrementen. Deze indeling is volgens de standaard van ARTM, waarbij ik rekening heb gehouden met mijn kennis en ervaring van de ontwikkelomgeving Delphi en de source-code van HARMONY (WebAccess). Voor elk increment waren 15 dagen ingepland, zodoende was er plaats voor drie incrementen (zie Figuur 1). Bij de indeling van de incrementen is in overleg met de bedrijfsmentoren besloten om in het eerste increment een makkelijk onderdeel te ontwikkelen (*Wachtwoord wijzigen*) om zo kennis op te doen van de ontwikkelomgeving en de source-code van HARMONY (WebAccess). Daarnaast was ook het onderdeel *Uren verantwoorden* ingedeeld, omdat de bedrijfsmentoren dit het belangrijkste en moeilijkste onderdeel vonden. In het geval dat er meer dan 15 dagen nodig waren voor dit increment dan was dit snel bekend en kon de planning bijgesteld worden. Het gevolg zou zijn dat er minder tijd over zou blijven voor de andere incrementen, maar dat was geen probleem omdat ze een lagere prioriteit hadden. De indeling van de incrementen was:  
Increment 1: De functies *Wachtwoord wijzigen* en *Uren verantwoorden*.  
Increment 2: De functies *Overzicht van mutaties en gezien*.  
Increment 3: De functies *Rooster gezien* en *Overzicht wensen*.

Het resultaat van elk increment was een nieuwe functie in HARMONY WebAccess, zoals beschreven in het Functioneel Ontwerp (FO) en Technisch Ontwerp (TO). Elk increment bevatte de volgende stappen:

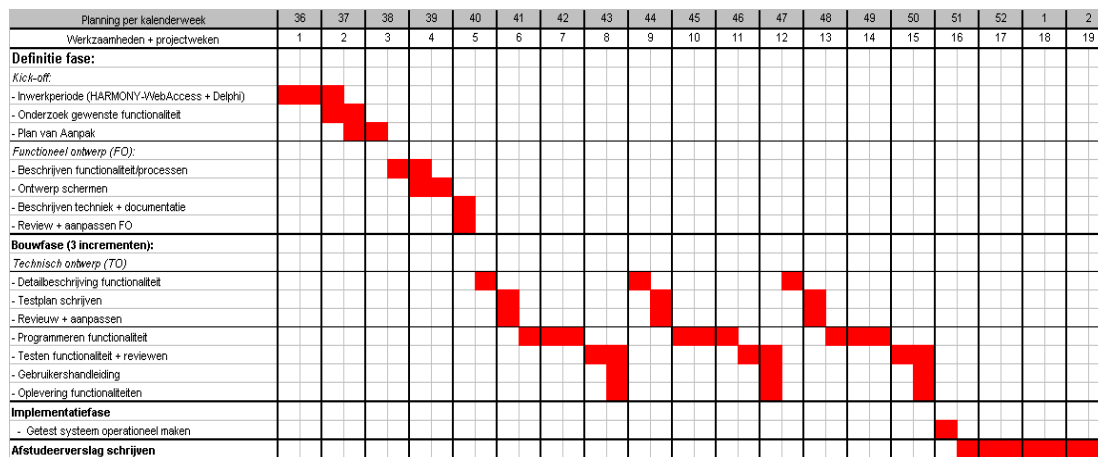
- Opstellen TO: het vaststellen van de technische eisen van de te bouwen functionaliteit;
- Opstellen Testplan. Het testplan is een onderdeel van het TO en bevat alle mogelijke acties van een gebruiker met het gewenste resultaat;
- Goedkeuring. Het TO en Testplan zijn goedgekeurd door een bedrijfsmentor voordat er geprogrammeerd werd;
- Programmeren. Het programmeren van de beschreven functionaliteit in FO en TO;
- Uitvoeren Systeemtest. Ontwikkelde functionaliteit laten testen door een collega met behulp van het Testplan;
- Aanpassingen doen. Als gevolg van de test aanpassen van het FO en / of TO en / of de source-code.

### **3.1.3           Implementatiefase**

Het doel van de Implementatiefase was het ontwikkelde systeem operationeel maken voor een klant. Bij aanvang van het project was bekend dat ik geen implementatie zou gaan doen bij een klant. Mijn doel was om ervoor te zorgen een overdraagbaar geheel op te leveren, dus zorgen dat de documentatie volledig was en dat de source-code getest en goedgekeurd was.

## **3.2           Planning**

Om een overzicht te krijgen van de activiteiten die uitgevoerd moesten worden is een planning gemaakt waarin wordt aangegeven hoeveel tijd beschikbaar was voor een bepaalde activiteit. Figuur 1 toont de planning die aan het begin van het afstudeerproject is gemaakt.



Figuur 1: Planning

De planning bestaat 19 weken, namelijk de periode van 1 september 2003 tot en met 9 januari 2004. Voor het uitvoeren van het afstudeerproject bleven effectief 18 weken over ten gevolge van vrije dagen, zoals kerst en oud en nieuw, waarvan 3 weken gereserveerd voor het schrijven van het afstudeerverslag. Dit betekende dat er 15 weken overbleven voor het uitvoeren van mijn afstudeeropdracht. De planning is gedurende het project een aantal malen aangepast. In Externe bijlage III wordt de laatste versie van de planning getoond.

## 4. Methode & technieken

Dit hoofdstuk beschrijft de door mij gebruikte methode en technieken en heb ik de gebruikte methode (standaard van ARTM) vergeleken met andere methoden. Paragraaf 4.1 beschrijft de standaard van ARTM en de punten waarop ik van deze standaard ben afgeweken. Mijn bedrijfsmentor heeft geadviseerd deze standaard te gebruiken, zodat mijn project volgens de regels van ARTM uitgevoerd zou worden. In paragraaf 4.2 wordt de organisatiegebonden standaard van ARTM vergeleken met twee systeemontwikkeling methoden en de conclusie die ik getrokken heb op basis van deze vergelijking.

### 4.1 Standaard van ARTM

De standaard van ARTM is gericht op het beheersen van projecten (projectmanagement). Binnen ARTM wordt namelijk gewerkt in projecten. Een project wordt opgezet wanneer het nodig is om aanpassingen te doen in HARMONY. De standaard is ontwikkeld door ARTM en is gebaseerd op ervaringen die opgedaan zijn tijdens projecten. De standaard is nog steeds in ontwikkeling. Bij de start van een project wordt een technisch projectleider aangewezen die de leiding heeft over een klein projectteam. De technisch projectleider is verantwoordelijk voor de voortgang van het project en de kwaliteit van de op te leveren producten. Binnen een project zijn drie fasen te onderscheiden:

- Definitiefase;
- Bouwfase;
- Implementatiefase.

#### 4.1.1 Definitiefase

De Definitiefase is bedoeld om helder te krijgen wat de gewenste aanpassingen en / of nieuwe functies zijn. Op basis van deze wensen wordt bepaald wat de moeilijkheidsgraad is en wat voor impact de ontwikkeling heeft op de bestaande source-code. Als dit bekend is wordt op basis van het kennisniveau van de projectleden een schatting gemaakt worden van de tijd die nodig is voor de ontwikkeling. Met behulp van deze gegevens wordt een projectplan opgesteld. In het projectplan wordt de gewenste functionaliteit verdeeld over een aantal incrementen. Het aantal functies dat binnen één increment ontwikkeld kan worden is afhankelijk van de moeilijkheidsgraad en impact die de functies hebben. De duur van een increment ligt rond de 10 dagen. Om ervoor te zorgen dat de projectleden weten wat ze moeten doen en dat het project binnen het opgestelde budget blijft, kan de technisch projectleider gebruik maken van een aantal checklists (zie Externe bijlage IV). Deze checklists zijn:

- Aansturing projectteam, hierin staan de taken en verantwoordelijkheden van de projectleden beschreven;
- Risicoanalyse, beschrijft de risico's die op kunnen treden tijdens het project en hoe hier op gereageerd moet worden;
- RAEW-matrix, geeft een overzicht van de verantwoordelijkheden van alle betrokkenen binnen het project;
- Samenwerking zusters, beschrijft een aantal afspraken die gemaakt moeten worden in het geval er met een zusterorganisatie van ORTEC samengewerkt wordt.

Als de technisch projectleider een projectplan heeft, kan begonnen worden met de Bouwfase.

De technieken die in deze fase gebruikt kunnen worden, zijn:

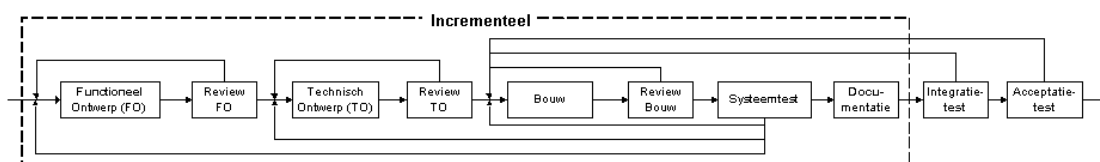
- Interviews. Voor het achterhalen van de wensen van de klant;
- Brainstormsessie. Kan gebruikt worden om met het projectteam afspraken te maken over de taken die gedaan moeten worden en wie verantwoordelijk is voor het uitvoeren van een bepaalde taak.

#### *Welke checklists en technieken gebruikt in mijn project*

Van deze standaard heb ik alleen de checklist Risicoanalyse gebruikt en opgenomen in mijn Plan van Aanpak (Externe bijlage III). Voor het achterhalen van de gewenste functionaliteit heb ik gebruik gemaakt van een brainstormsessie in plaats van interviews.

### **4.1.2 Bouwfase**

In de Bouwfase worden de incrementen ontwikkeld, zoals die beschreven staan in het projectplan. Het resultaat van elk increment is de (nieuwe) werkende functie in HARMONY (WebAccess). Figuur 2 toont de stappen die tijdens de Bouwfase worden verricht met daaronder een toelichting.



Figuur 2: Ontwikkelingschema Bouwfase

Bij elke stap wordt uitgelegd wat het doel is, wat er gedaan wordt en welke hulpmiddelen (zie Externe bijlage IV) gebruikt kunnen worden:

- Opstellen Functioneel Ontwerp (FO). Het FO beschrijft de gewenste functionaliteit die afgestemd zijn met de opdrachtgever en / of klant en dient als basis voor het Technisch Ontwerp (TO). Er kan gebruik gemaakt worden van de *Template Functioneel Ontwerp*;
- Review FO. Het FO moet eerst gecontroleerd worden op juistheid en volledigheid. Dit wordt gedaan door de reviewer<sup>1</sup> en is een andere persoon dan de diegene die het FO heeft opgesteld. Na de controle kan het FO verbeterd worden. Ná de goedkeuring van het FO, wordt het TO opgesteld;
- Opstellen TO. Met behulp van de checklist *Technisch Ontwerp* kunnen de technische eisen van de gewenste functionaliteit vastgelegd worden in het TO. Het TO is de basis voor het bouwen;
- Review TO. Het TO moet eerst gecontroleerd worden op juistheid en volledigheid. Dit wordt gedaan door de reviewer en is een andere persoon dan de schrijver van het TO. Na de controle kan het TO verbeterd worden. Ná de goedkeuring van het TO, kan begonnen worden met bouwen;
- Bouwen van functionaliteit en opstellen testplan. Programmeren van de gewenste functionaliteit zoals beschreven in het FO en TO. Voor informatie over het bouwen zelf kan de bouwer de checklist Bouwen

<sup>1</sup> Reviewer en bouwer zijn verschillende personen, omdat iemand anders beter in staat is om het werk te controleren.

gebruiken. Voor het opstellen van een testplan kan de checklist Testen gebruikt worden;

- Review Bouw. Is het controleren van de source-code en het testplan. De reviewer kan gebruik maken van de checklist Review Source-code waarin aangegeven wordt waar de source-code aan moet voldoen;
- Uitvoeren Systeemtest. De ontwikkelde functie(s) worden door de tester<sup>2</sup> getest met behulp van het Testplan;
- Mergen. Is het samenvoegen van verschillende versies van een bepaald source-bestand (een .pas-file). Aangezien het niet verstandig is zomaar verschillende versies samen te voegen, is een checklist Mergen opgesteld waarin uitgelegd wordt wat er gedaan moet worden;
- Opstellen documentatie, schrijven van help-bestanden. Beschrijven wat de ontwikkelde functie(s) kunnen en hoe het gebruikt kan worden.

Na de definitieve goedkeuring van een functie kan gestart worden met een volgend increment. Als alle incrementen gereed zijn kan begonnen worden met de Integratietest. Bij de Integratietest worden alle functies getest die anders zouden kunnen reageren door het toevoegen van de nieuwe functie. Als er fouten optreden dan moeten deze fouten opgelost worden door de bouwer, waarna de Integratietest opnieuw uitgevoerd moet worden tot het systeem foutloos is. In de Acceptatietest wordt hetzelfde gedaan als bij de Integratietest. Het verschil is nu dat de test wordt uitgevoerd in de testomgeving van de klant. Als er tijdens deze test fouten optreden wordt dit voorgelegd aan de bouwer en moet opnieuw de Systeemtest, Integratietest en Acceptatietest uitgevoerd worden. Wanneer de klant tevreden is met het resultaat, kan begonnen worden met de Implementatiefase.

De technieken die in deze fase gebruikt kunnen worden, zijn onderdeel van UML (Unified Modeling Language)<sup>3</sup>. Per techniek wordt beschreven wat het doel ervan is en in welk product het gebruikt kan worden:

- Use-cases. Bij het opstellen van het FO kan gebruik gemaakt worden van use-cases om voor de klant duidelijk te maken wat het systeem doet en wat door de gebruiker gedaan wordt;
- Class diagram. Om aan te geven wat voor relatie de (nieuwe) class heeft met andere classes, wordt aangeraden in het TO een class diagram te gebruiken;

Verder wordt in het TO aangeraden om een Collaboration diagram en / of een Activity diagram te maken als het om een ingewikkelde functie gaat.

- Collaboration diagram. Een collaboration diagram wordt gemaakt op basis van een use-case en toont alle gebeurtenissen (events) die tussen objecten van bepaalde classes plaatsvinden;
- Activity diagram. Een activity diagram wordt gemaakt op basis van een toestandsdiagram. Een toestandsdiagram toont de toestanden die een bepaald object kan hebben. Een activity diagram beschrijft een activiteit die plaatsvindt binnen de tijd dat een object zich in één toestand bevindt.

---

<sup>2</sup> Tester en bouwer zijn verschillende personen, omdat op deze manier het werk goed gecontroleerd kan worden.

<sup>3</sup> UML ondersteunt het objectgeoriënteerd ontwikkelen en dat is ook de manier waarop er bij ARTM wordt ontwikkeld.

### *Hoe gebruik gemaakt van de methode en de beschikbare middelen in mijn project*

In plaats van drie keer een FO maken (3 incrementen) heb ik de gewenste functies in één keer beschreven. Door deze keuze had ik een goed overzicht van datgene wat gedaan moest worden. Bovendien was aan het begin van het afstudeerproject voor de bedrijfsmentoren en de interne gebruikersgroep nog helder wat er besproken was tijdens de brainstormsessie en konden ze het FO goed beoordelen (reviewen). In het FO heb ik een eerste versie van een class diagram opgenomen. Mijn keuze voor het opnemen van een class diagram was, omdat met een class diagram de volgende onderdelen te zien zijn:

- de omvang en de complexiteit van het systeem;
- Objecten uit HARMONY WebAccess;
- Relaties tussen de objecten;
- Daadwerkelijk uit te voeren operaties in HARMONY WebAccess;
- Context van HARMONY WebAccess.

In het TO heb ik geen gebruik gemaakt van Activity diagrams of Collaboration diagrams. Deze diagrams zijn namelijk nodig om de toestand van een bepaald object of class verder uit te splitsen. Uit de class diagrams kon ik zien welke classes ik nodig had en wat de relaties tussen de classes waren. Het reviewen van de Technisch Ontwerpen en de source-code was de taak van dhr. F. Bos. Het Testplan heb ik gelijktijdig met het Technisch Ontwerp geschreven, zodat ik voor het bouwen al nagedacht had over wat de te bouwen functie moest kunnen en wat niet. Dit is een nieuwe ontwikkeling van de standaard en daarom nog niet doorgevoerd in de checklists van de standaard. Mergen heb ik niet hoeven doen, omdat ik als enige aan HARMONY WebAccess werkte waren er geen andere versies van de source-code in mijn project.

#### **4.1.3 Implementatiefase**

In de Implementatiefase wordt het systeem uit de testomgeving in de operationele omgeving geplaatst (geïmplementeerd). Als de implementatie succesvol verloopt kan het project afgerond worden. Tijdens en na de implementatie kan het nodig zijn de volgende stappen uit te voeren:

- Conversie. Het kan nodig zijn om gegevens uit andere systemen over te zetten in het nieuwe systeem (HARMONY);
- Opleidingen. De gebruikers leren werken met het nieuwe systeem;
- Ondersteuning. Mocht de opleiding niet voldoende zijn dan kan de klant nog ondersteuning krijgen in het gebruik van het systeem;
- Helpdesk en support. Als er problemen zijn kan de klant terecht bij de Helpdesk;
- Onderhoud. Na verloop van tijd kan het nodig zijn om onderhoud te plegen aan het systeem.

Van deze fase vond ik het vreemd dat de punten Helpdesk en support en Onderhoud opgenomen waren. De reden hiervoor is dat implementaties van HARMONY bij sommige klanten op meerdere locaties plaats kan vinden. Aangezien de implementatie niet op alle locaties tegelijk plaats vindt, kan het gebeuren dat een locatie tegen problemen is aangelopen en dus de Helpdesk nodig heeft. Als de Helpdesk het probleem niet kan verhelpen, wordt het probleem doorgegeven aan support en moet onderhoud gepleegd worden. Bij de afronding van het project komt het projectteam bij elkaar waarna een evaluatie plaats vindt over de gang van



zaken tijdens het project. Uit deze evaluatie kan vastgesteld worden welke zaken in een volgend project ook gedaan moeten worden en wat in het vervolg anders moet.

## **4.2      Vergelijking andere methoden**

Met het vergelijken van de standaard van ARTM met andere methode wil ik nagaan wat nu de overeenkomsten en de verschillen zijn met andere ontwikkelmethoden. Ik heb de ontwikkelmethoden System Development Methodology<sup>4</sup> (SDM) en Dynamic Systems Development Method<sup>5</sup> (DSDM) hiervoor uitgekozen. Naast deze methoden zijn er nog meer methoden. Ik heb voor deze twee methoden gekozen, omdat ik hier kennis van had. Tijdens mijn opleiding Informatica en Informatiekunde aan de Haagse Hogeschool heb ik kennis opgedaan met SDM en tijdens een vorige stageperiode heb ik gebruik gemaakt van DSDM. Paragraaf 4.2.1 beschrijft de methode SDM en DSDM staat beschreven in paragraaf 4.2.2. In paragraaf 4.2.3 worden de overeenkomsten en verschillen van de twee methoden in vergelijking met de standaard van ARTM beschreven. Op basis hiervan geef ik een conclusie.

### **4.2.1      SDM**

SDM maakt gebruik van het traditionele watervalmodel. De beschrijving van de systeemontwikkeling is in SDM "top down", dat wil zeggen dat steeds vanuit een groter geheel nadere detaillering plaatsvindt. Het ontwikkelen van een systeem wordt door SDM vanuit vier invalshoeken bekeken, namelijk:

- De projectbesturing. De besturing vindt vooral plaats door de fasering en door het uitvoeren van bepaalde activiteiten. De fasering is zodanig opgebouwd dat elke fase zo nodig uitgevoerd kan worden als een apart project, dit geldt ook voor het uitvoeren van bepaalde activiteiten. Elke fase begint daarom met een activiteit voor het maken van een plan van aanpak en kent één of meer mijlpaalproducten op basis waarvan beslissingen genomen kunnen worden;
- De systeemontwikkeling. Dit proces wordt bepaald door samenhangende activiteiten met bijbehorende producten, op basis waarvan een systeem kan worden ontwikkeld. Deze producten bevatten specificaties van zowel geautomatiseerde als handmatige handelingen;
- De verandering. Het invoeren van een nieuw systeem heeft voor een organisatie meestal vele gevolgen. Diverse activiteiten beschrijven de wijze waarop de (gewenste) verschillen tussen de huidige en de nieuwe situatie worden vastgesteld en welke maatregelen genomen moeten worden om dit veranderingsproces uit te voeren en te sturen;
- De validatie (kwaliteitsbewaking). In elke fase zijn één of meer specifieke activiteiten opgenomen, die ten doel hebben te zorgen dat de kwaliteit van de gemaakte producten aan de afgesproken normen

---

<sup>4</sup> Literatuur: Reader SO-04, Kopieën uit het boek: Systeemontwikkeling volgens SDM.

<sup>5</sup> Literatuur: Dynamic Systems Development Method van Jennifer Stapleton.

voldoet. Bij het uitvoeren van elke activiteit vindt bovendien verificatie plaats van de producten van die activiteit.

De verdeling van de ontwikkeling in fasen geeft de mogelijkheid het proces goed te besturen op basis van zogenaamde mijlpaalproducten. Per fase onderscheidt men één of meer beslissingspunten, waarvoor gerichte informatie moet worden verstrekt. De levenscyclus van een systeem wordt door SDM verdeeld in 7 fasen. Deze fasen zijn:

- Informatieplanning. Het doel van deze fase is het opstellen van een informatieplan dat als uitgangspunt zal dienen voor de informatievoorziening in de komende jaren;
- Definitiestudie. Het doel van deze fase is het vaststellen of het ontwikkelen en het invoeren van een nieuw systeem zinvol is en zo ja hoe dit moet plaatsvinden;
- Basisontwerp. In deze fase worden de benodigde gegevensverzamelingen en de benodigde faciliteiten van het systeem beschreven, daarnaast worden ook de functies van het systeem beschreven. Als het in één keer ontwikkelen van het systeem moeilijk te beheersen valt, kan besloten worden om het systeem in subsystemen in te delen. In dit geval moeten ook in detail de relaties tussen de subsystemen vastgelegd worden.
- Detailontwerp. In deze fase wordt zo gedetailleerd mogelijk de systeemeisen beschreven, evenals alle gegevens, die voor het functioneren van het systeem nodig zijn. De resultaten van deze fase zijn een functioneel ontwerp en een technisch ontwerp;
- Realisatie. Op basis van het technisch ontwerp wordt in deze fase het systeem gebouwd en gereed gemaakt voor invoering;
- Invoering, hier kunnen al voorbereidingen voor getroffen worden na het afsluiten van de definitiestudie. Dit is vaak ook wenselijk omdat in deze fase de volgende taken uitgevoerd moeten worden:
  - Voorlichting en opleidingen gegeven moeten worden;
  - Conversies van gegevens naar de eventuele nieuwe opslagstructuur;
  - Het gereed maken van de nieuwe werkomgeving en organisatie;
  - Het completeren en distribueren van de documentatie;
  - Het maken van de nodige afspraken over de (computer)verwerking.Als alles gereed is en alle gebruikers zijn in staat met het nieuwe systeem te werken en de acceptatietest met goed gevolg doorlopen is, vindt de echte invoering plaats en wordt overgegaan op normaal gebruik.
- Gebruik en beheer. Deze fase duurt zo lang het systeem in gebruik blijft. Het doel van deze fase is een goed beheer over het in gebruik zijnde systeem uit te voeren en wel zodanig dat het blijft voldoen aan de wensen en eisen van de gebruikers en ook blijft passen in de infrastructuur van de organisatie.

Na elke fase wordt een beslissing genomen of er doorgegaan wordt of dat er gestopt wordt met de ontwikkeling. Het kan dus gebeuren dat men bijvoorbeeld in de fase Detailontwerp zit en door het management of opdrachtgever besloten wordt dat het project wordt stop gezet. Het resultaat is dan een aantal documenten waar veel tijd in heeft gezeten, maar waar niemand iets mee kan. In het geval dat het hele traject wel goed verloopt, kan het zijn dat men erachter komt dat er één of meerdere fouten in de programmatuur zitten die opgelost moeten worden. Dit heeft

tot gevolg dat men opnieuw moet beginnen bij het Basisontwerp of Detailontwerp of Realisatie. In alle gevallen zal dit veel vertraging opleveren, omdat niet zomaar een fout hersteld kan worden. Vanwege de verwevenheid van de source-code kan een wijziging gevolgen hebben voor de gehele werking van het systeem.

#### **4.2.2 DSDM**

DSDM is gebaseerd op een iteratieve, incrementele opleverstrategie. Deze strategie maakt het mogelijk in kortere tijd op te leveren en daardoor sneller meerwaarde te genereren voor de gebruikersorganisatie. DSDM definieert een generiek, hoog niveau proces (= raamwerk) dat aangepast kan worden voor gebruik in een specifieke organisatie, afhankelijk van de bedrijfs- en technische mogelijkheden en beperkingen. In dit proces worden de volgende vijf fasen onderscheiden:

- Haalbaarheidsonderzoek. In deze fase wordt nagegaan of DSDM de juiste aanpak is voor het project;
- Bedrijfsanalyse. In deze fase worden de te automatiseren processen in kaart gebracht en wordt een beschrijving van de systeemarchitectuur gemaakt;
- Functioneel model iteratie. In deze fase worden de informatie-eisen uit de Bedrijfsanalyse verder uitgewerkt, het resultaat zijn zowel analysemodellen als softwarecomponenten;
- Ontwerp en bouw iteratie. Het systeem wordt in deze fase in zoverre ontwikkeld dat het veilig overgedragen kan worden aan de gebruikers;
- Implementatie. In deze fase wordt het systeem vanuit de ontwikkelomgeving overgezet naar de operationele omgeving.

Vanwege het continu managen van de op te leveren functionaliteit wordt bewerkstelligd dat die functionaliteit wordt opgeleverd waaraan de gebruikersorganisatie de meeste behoefte heeft. Voornamelijk in situaties waarin de oplevertermijn vaststaat, en de op te leveren functionaliteit nog niet volledig uitgespecificeerd is, is deze manier van opleveren succesvol. DSDM is gebaseerd op "best practice" ervaringen van organisaties met versnelde systeemontwikkeling. Het extract van al deze ervaringen heeft geleid tot de theorie. De basis wordt gelegd door 9 principes die in de hele theorie terug te vinden zijn. Hieronder staan de negen basisbeginselen met daaronder een toelichting:

- Actieve betrokkenheid van gebruikers is noodzakelijk;
- DSDM-teams moeten gemachtigd zijn besluiten te nemen;
- Frequente oplevering van producten is van wezenlijk belang;
- Geschiktheid voor bedrijfsdoeleinden is het essentiële criterium voor de acceptatie van producten;
- Iteratieve en incrementele ontwikkeling is noodzakelijk om te convergeren tot een juiste bedrijfsoplossing;
- Alle wijzigingen tijdens de ontwikkeling zijn terug te draaien;
- Eisen worden op hoog niveau vastgelegd;
- Testen is geïntegreerd in de levenscyclus;
- Een samenwerkende en coöperatieve houding van alle belanghebbenden is essentieel.

#### *Actieve betrokkenheid van gebruikers is noodzakelijk*

Door de gebruikers actief te betrekken bij de ontwikkeling kan de mening gepeild worden. Dit kan tot gevolg hebben dat tijdens de ontwikkeling de

opgestelde eisen wijzigen en meteen doorgevoerd worden in het te ontwikkelen systeem. Bij een andere methoden worden gebruikers aan het begin van het project betrokken om de eisen vast te stellen, daarna worden ze incidenteel opgeroepen voor het evalueren van producten en functies om tenslotte de acceptatietest uit te voeren.

Het ontwikkelteam wordt ondersteund door een paar kundige gebruikers, die de evaluatie van producten doen en zodoende snel reactie kunnen geven aan het ontwikkelteam.

*DSDM-teams moeten gemachtigd zijn besluiten te nemen*

Om de voortgang van het project niet te belemmeren moeten projectleden in staat zijn zelf beslissingen te nemen. DSDM-projecten zijn vaak gebonden aan strakke tijdschema's. Lange besluitvormingsprocedures kunnen tot gevolg hebben dat het systeem niet op tijd geleverd kan worden.

*Frequente oplevering van producten is van wezenlijk belang*

Het frequent opleveren van producten heeft als doel het in goede banen leiden van activiteiten en het effectief werken op korte termijn. Door frequent op te leveren kan het management in de gaten houden wat men aan het ontwikkelen is en hoe ver men daarin is gevorderd. Mocht het nodig zijn dan kunnen ze tijdig bijsturen. Door het plannen van een regelmatige oplevering (zeg wekelijks) van iets tastbaars en zichtbaars creëert men een vangnet voor het terughalen van verkeerde beslissingen.

*Geschiktheid voor bedrijfsdoeleinden is het essentiële criterium voor de acceptatie van producten*

Dit principe houdt in dat een ontwikkelaar niet op een bepaald punt moet blijven steken, omdat hij een perfecte oplossing wil hebben. Door de geschiktheid voor het bedrijfsdoel als uitgangspunt te nemen, kunnen bepaalde technische aspecten worden uitgesteld – mits er ook op korte termijn voldoende basis aanwezig is voor een veilig en betrouwbaar systeem.

*Iteratieve en incrementele ontwikkeling is noodzakelijk om te convergeren tot een juiste bedrijfsoplossing*

Als het team gebruikers bevat die vrijwel direct feedback geven op het werk van de ontwikkelaars, is het mogelijk om systeemontwikkeling stap voor stap uit te voeren in plaats van in één keer. Doordat systemen stukje bij beetje ontwikkeld worden, zorgt DSDM ervoor dat fouten vroeg ontdekt worden, waardoor men kostbare herstelprocedures achteraf kan voorkomen.

*Alle wijzigingen tijdens de ontwikkeling zijn terug te draaien*

In een iteratief en incrementeel proces moet het mogelijk zijn dat men erkent een verkeerde weg te zijn ingeslagen en dat men moet terugkeren naar een eerder veilig punt in het traject. Dit houdt in dat er spraken dient te zijn van een vlekkeloos beheer van alle softwarecomponenten en de bijbehorende documentatie.

*Eisen worden op hoog niveau vastgelegd*

De toepassing van dit principe houdt in dat de eisen die tijdens de Bedrijfsanalyse zijn verzameld, globaal de reikwijdte van het project bepalen. Als op dit punt globale eisen vastliggen, kunnen de meer

gedetailleerde eisen in de volgende iteraties eraan toegevoegd worden. Het kan voorkomen dat de uitgangspunten in een later stadium worden bijgesteld.

*Testen is geïntegreerd in de levenscyclus*

Bepaalde systeemonderdelen worden al vroeg in het project gemaakt en daarna geleidelijk verder ontwikkeld tot het uiteindelijke systeem. Daarom hanteert DSDM de filosofie: testen terwijl je bezig bent. De ontwikkelaar test de technische aspecten en de gebruikers in het team testen de functionele zaken. Op deze manier worden alle vormen van testen, inclusief integratietest, gedurende het traject van het project geleidelijk uitgevoerd.

*Een samenwerkende en coöperatieve houding van alle belanghebbenden is essentieel*

De ontwikkelaars weten zonder de hulp van de gebruikers niet wat nodig is, daarom is het belangrijk dat alle betrokkenen samenwerken en meewerken. Als bijvoorbeeld een bepaalde groep gebruikers bepaalde verwachtingen heeft van het te ontwikkelen systeem en deze blijken ongegrond, kan het hele proces fout aflopen.

De fasen van DSDM worden ingedeeld volgens het principe van timeboxing. Dit houdt in dat de doorlooptijd vast staat en dat er in deze doorlooptijd een variabele hoeveelheid werk uitgevoerd kan worden. Na afronding van de timebox zal een nieuwe timebox worden opgestart met opnieuw een vaste tijd en een nieuwe hoeveelheid werk. Werk dat is overgebleven uit een eerdere timebox zal opnieuw worden geprioriteerd en worden meegenomen naar de volgende timebox. In de timeboxen wordt door middel van de MoSCoW-methode gezorgd voor een prioritering van onderdelen. De MoSCoW-methode is een manier om bij een project de onderlinge prioriteiten van eisen te beheren. MoSCoW staat voor:

Prioriteit 1 Must-have. Voor eisen die essentieel zijn voor het systeem;

Prioriteit 2 Should-have. Voor belangrijke eisen die wellicht ook essentieel zijn beschouwd als de beschikbare tijd het had toegelaten. Zonodig kan het systeem er echter buiten;

Prioriteit 3 Could-have. Betreft eisen die wat eenvoudiger weggelaten kunnen worden;

Prioriteit 4 Want-to-have. Voor die waardevolle eisen die wel kunnen wachten totdat er verder wordt ontwikkeld.

Mits goed uitgevoerd is het resultaat een systeem dat voldoet aan de eisen van de gebruiker. Vanwege de betrokkenheid van de gebruikers hebben ze weinig of geen opleiding nodig. Fouten in de source-code worden eerder gevonden, dus de kans op fouten na oplevering is minder groot.

#### **4.2.3            Verschillen en overeenkomsten met standaard ARTM**

Deze paragraaf beschrijft de belangrijkste verschillen en overeenkomsten tussen enerzijds de standaard van ARTM en anderzijds SDM en DSDM. Op basis van deze vergelijking wordt een conclusie gegeven.

#### **Overeenkomsten en verschillen in vergelijking met SDM**

De overeenkomsten hebben betrekking op het maken van een functioneel en technisch ontwerp, de zaken die geregeld moeten worden voor het invoeren en implementeren. Het verschil hierbij is dat bij SDM deze activiteiten parallel en / of na elkaar uitgevoerd worden. In de standaard van ARTM is het ook mogelijk om één of meerdere stappen terug te gaan en / of te herhalen om eventuele fouten te verbeteren.

### **Overeenkomsten en verschillen in vergelijking met DSDM**

De overeenkomsten met DSDM zijn:

- Incrementeel en iteratief ontwikkelen, zoals beschreven in paragraaf 4.1.2;
- Kleine ontwikkelteams met de mogelijkheid om zelf beslissingen te nemen. ARTM maak ook gebruik van kleine projectteams, waarbij de projectleden in overleg met de technisch projectleider bevoegd zijn beslissingen te nemen;
- Gebruik maken van timeboxing op basis van prioriteiten. ARTM maakt gebruik van incrementen die ook binnen een vastgestelde tijd gereed moet zijn, hierbij wordt gebruik onderscheid gemaakt door het stellen van prioriteiten;
- Een DSDM-project wordt afgesloten met een Project Evaluatie Document (PED). Bij ARTM houdt de projectgroep na afloop van een project ook een evaluatie om zo na te gaan wat er goed en / of fout ging. Door deze evaluatie kan besloten worden dat de standaard van ARTM gewijzigd moet worden.

De belangrijkste verschillen tussen DSDM en de standaard van ARTM zijn de betrokkenheid van de gebruikers bij de ontwikkeling en de prioritering van de te ontwikkelen functies.

#### *Betrokkenheid gebruikers*

De betrokkenheid van gebruikers is voor DSDM een absolute voorwaarde om een DSDM-project succesvol te laten zijn. Bij ARTM zijn hier geen regels voor opgesteld. De algemene opvatting is om een gebruiker op te bellen in het geval zekerheid wil hebben over de precieze bedoeling van een functie.

#### *Prioritering*

DSDM heeft een eenduidige manier voor het benoemen van de prioriteiten. Bij ARTM wordt gekeken naar de functies die gebouwd moeten worden, op basis van deze gegevens wordt een indeling gemaakt op de belangrijkheid van de functie. Het nadeel hiervan is dat er elke keer een andere prioritering plaats kan vinden en daardoor onduidelijkheid bestaat over het werkelijke belang van een bepaalde functie

### **Conclusie**

Op basis van de vergelijking, kan ik ARTM aanbevelen het principe van de MoSCoW-regels van DSDM over te nemen. Zodat er een eenduidige manier is om de prioriteit van een functie aan te geven.

## 5. Bedrijfsprofiel

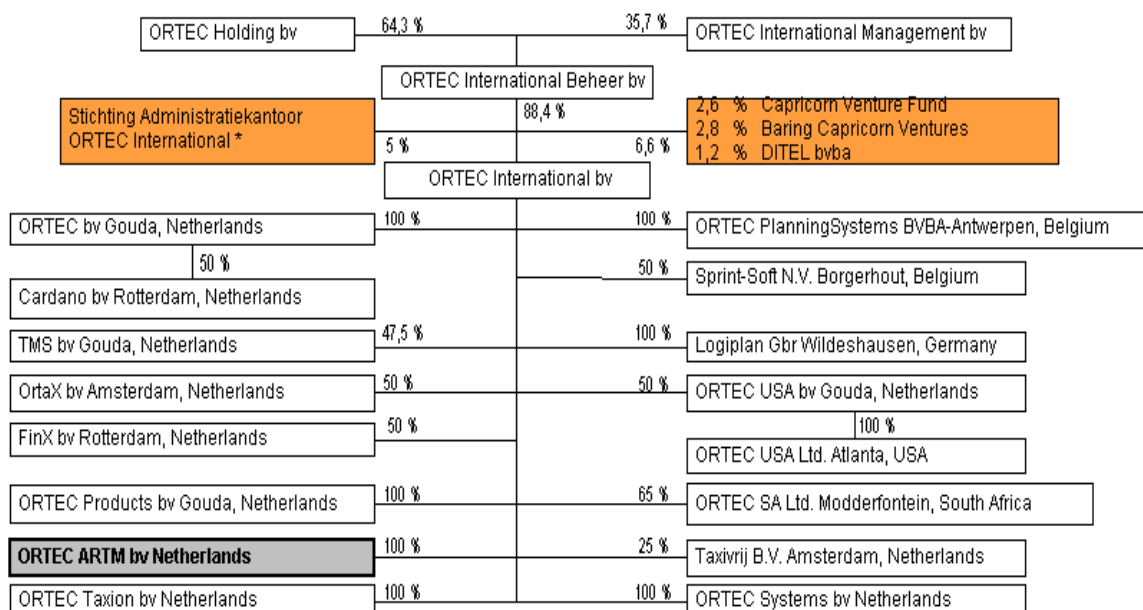
Dit hoofdstuk geeft een beeld van de organisatie waar ik mijn afstudeeropdracht heb uitgevoerd. In paragraaf 5.1 staat een algemene beschrijving van de organisatie. Paragraaf 5.2 beschrijft de afdeling waarbinnen de afstudeeropdracht heeft plaats gevonden en in paragraaf 5.3 wordt aangegeven wat mijn plaats binnen deze afdeling was.

### 5.1 Organisatie ORTEC bv

ORTEC (**O**perations **R**esearch **T**echnology) bv is een internationaal opererend bedrijf met meer dan 300 werknemers. ORTEC is gespecialiseerd in het op product- en projectbasis ontwikkelen en leveren van software ter ondersteuning van logistieke en financiële beslissingsprocessen. Naast het aandachtsgebied arbeidstijdmanagement is ORTEC sterk aanwezig op de volgende gebieden:

- Financiële zaken.
- Asset & Liability Management.
- Revenue Management.
- Logistiek.
- Transport & Distributie.
- Productieplanning Procesindustrie.

Kenmerkend voor alle systemen van ORTEC is dat zij één of meerdere optimalisatiemethoden bevatten die planningen genereren (advanced planning). In dit opzicht is ORTEC uniek in Nederland en internationaal toonaangevend. ORTEC is door nauwe banden met de internationale academische wereld, direct betrokken bij innovatief onderzoek op dit terrein. Diverse van haar medewerkers zijn parttime werkzaam als hoogleraar. De dienstverlening van ORTEC bestaat sinds haar oprichting in 1981 uit een combinatie van het geven van advies op het gebied van planning processen en het leveren van advanced planning software. Figuur 3 toont de organisatiestructuur van de ORTEC-groep. De afdeling ORTEC ARTM bv is 100% onderdeel van ORTEC International bv.

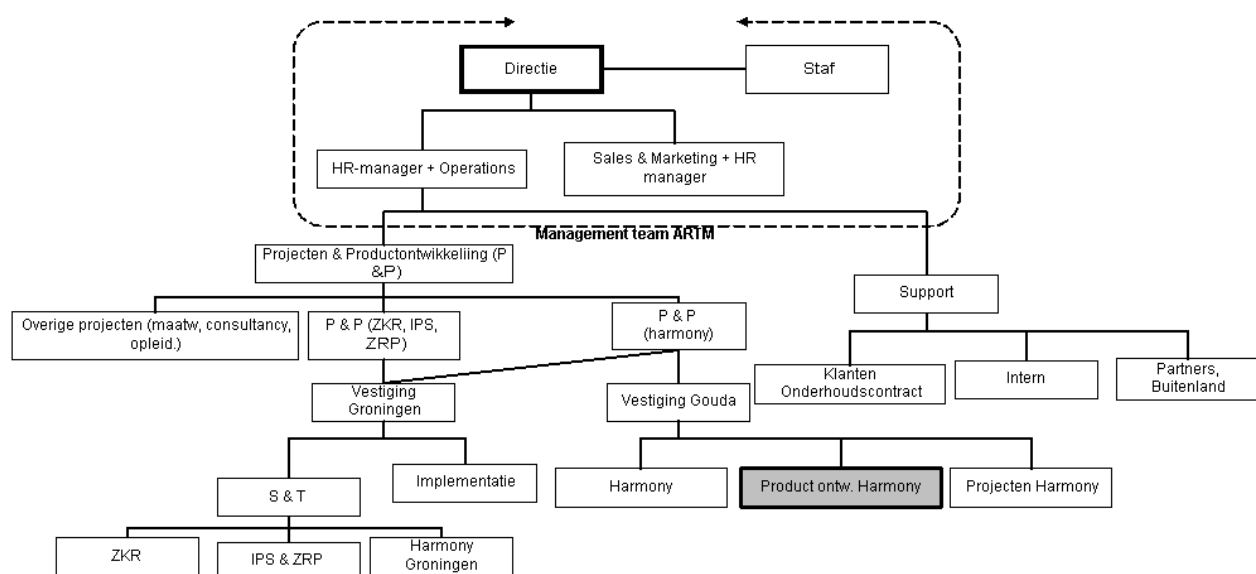


\* Van de stichting Administratiekantoor ORTEC International is 3,64% in handen van het overige personeel van ORTEC en 1,36% in handen van OF International Beheer bv

Figuur 3: Organigram ORTEC-groep

## 5.2 Organisatie ORTEC Arbeidstijdmanagement bv

Het aandachtsgebied arbeidstijdmanagement (inzetplanning van personeel) is een substantieel onderdeel van de activiteiten van ORTEC. De afdeling Arbeidstijdmanagement (ARTM) heeft circa 50 medewerkers. De medewerkers zijn onderverdeeld in marktgerichte groepen, clusters genaamd. Ieder cluster bevat medewerkers met verschillende specialismen. Figuur 4 toont de organisatiestructuur van ARTM.



Figuur 4: Organigram ORTEC Arbeidstijdmanagement bv

De ontwikkeling, bouw, implementatie en ondersteuning van het softwarepakket HARMONY omvatten het grootste deel van de inspanningen van de afdeling. Daarnaast houdt ARTM zich bezig met consultancy opdrachten op het gebied van arbeidstijdmanagement. Voorbeelden hiervan zijn marktspecifieke Arbeidstijdenwet- en CAO-trainingen.

De afdeling ARTM is verdeeld over een tweetal vestigingen, een vestiging in Groningen en een vestiging in Gouda. De vestiging in Gouda verzorgt de maatwerk- en consultancy opdrachten en de maatwerkonderdelen van HARMONY. De vestiging in Groningen is verantwoordelijk voor de standaard producten. Dit zijn de versies die zonder aanpassingen verkocht worden. Belangrijke klanten zijn bijvoorbeeld ambulancediensten en ziekenhuizen.

## 5.3 Functioneren van afstudeerder in de organisatie

Als afstudeerder bij de afdeling ARTM kon ik zelfstandig aan mijn afstudeeropdracht werken. Een reden hiervoor was onder andere het gebrek aan kennis van de scripttaal JavaScript<sup>6</sup> binnen de afdeling. De enige persoon binnen de afdeling ARTM die hier een beetje kennis van had, was de heer P. van 't Hoff. Binnen ARTM was wel literatuur over JavaScript beschikbaar en via internet was ook veel informatie te vinden. Voor vragen

<sup>6</sup> In het vervolg van dit verslag zal gesproken worden over de programmeertaal JavaScript.



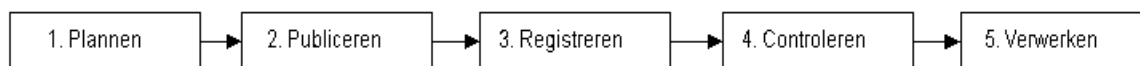
over de programmeertaal Delphi kon ik bij de bedrijfsmentoren, de heer P. van 't Hoff en de heer F. Bos, terecht of eventueel bij collega's, omdat HARMONY in deze programmeertaal wordt ontwikkeld. Mijn plaats binnen de organisatie was in het cluster Product ontwikkeling HARMONY (zie Figuur 4). Het hoofd van dit cluster was de heer P. van 't Hoff. Aan het eind van het project is de heer F. Bos, weggegaan bij ORTEC en heeft de heer J. Ensing zijn taken overgenomen.

## 6. Analyse HARMONY

Om een beeld te krijgen van de omgeving waarbinnen mijn opdracht uitgevoerd moest worden heb ik een analyse gemaakt van HARMONY en HARMONY WebAccess. In deze analyse heb ik gekeken naar datgene wat het systeem doet en wat voor technieken daarbij te pas komen. Eerst is HARMONY geanalyseerd, omdat dit de basis is van HARMONY WebAccess. De analyse is onderverdeeld in een functionele analyse (zie paragraaf 6.1) en een technische analyse (zie paragraaf 6.2). Paragraaf 6.3 geeft uitleg over de functionele en technische werking van HARMONY WebAccess. De functionele analyse staat beschreven in paragraaf 6.3.1 en paragraaf 6.3.2 beschrijft de technische analyse. In Externe bijlage I is een verdere beschrijving te vinden van HARMONY in de vorm van een whitepaper<sup>7</sup>.

### 6.1 Functionele analyse

De functionele analyse van HARMONY heb ik uitgevoerd, om zo te bepalen wat voor bestaande functies ik nodig had voor het uitvoeren van mijn opdracht. Een ander doel was ook het analyseren van de werking van deze functies. Hiervoor heb ik de gebruikershandleiding en de applicatie HARMONY geraadpleegd. HARMONY is een roosterapplicatie waarmee organisaties hun medewerkers kunnen inplannen. Dit kan door middel van een roostergenerator die op basis van criteria een rooster kan samenstellen. Voorbeelden van criteria zijn ATW (arbeidstijdenwet) en CAO-regelgeving. Daarnaast kan de gebruiker (planner) ook handmatig een rooster maken. Figuur 5 toont de processen die tezamen het



roosterproces vormen met daaronder een toelichting van de processen.

*Figuur 5: Overzicht roosterproces*

Om duidelijk te maken in welke fase van het roosterproces men zit, wordt een status bijgehouden die bij elke procesovergang wordt gewijzigd.

#### Stap 1: Plannen.

De planner maakt een rooster voor een bepaalde periode en de status van het rooster is *Planning*. De andere statussen zijn: *Gepubliceerd*, *Verantwoord*, *Geaccordeerd* en *Verwerkt*. De gegevens die nodig zijn voor het plannen zijn:

- Activiteiten. Dit zijn alle mogelijke activiteiten die een medewerker kan doen. Voorbeelden zijn: werk, pauze en verlof;
- Diensten. Een dienst is een verzameling van activiteiten en bevat minimaal één activiteit;
- Regelgeving. Hierin staan de regels waar een planner rekening mee moet houden, zoals wetgeving en wensen van medewerkers;
- Medewerkers. Dit zijn de mensen die de diensten uitvoeren en behoren tot een bepaalde roostergroep.

<sup>7</sup> In de whitepaper wordt gesproken over HARMONY-Online, dit is de oude naam van HARMONY WebAccess

- Roostergroep. Een roostergroep is een groep van medewerkers die tezamen verantwoordelijk zijn voor de uitvoering van de toegewezen diensten.

Als de status *Planning* is, betekent dit dat het rooster zonder problemen gewijzigd kan worden. Het is ook mogelijk om per dienst de status te veranderen. Als alle diensten en medewerkers zijn ingeroosterd krijgt het rooster de status *Gepubliceerd*. Het is de taak van de planner om het rooster bekend te maken aan de medewerkers / roostergroep voordat de desbetreffende periode is begonnen.

#### Stap 2: Publiceren.

Het rooster wordt verspreidt in de organisatie, het is echter mogelijk dat er aanpassingen gedaan moeten worden. Bij ARTM wordt er in plaats van aanpassingen gesproken over mutaties. Mutaties zijn onder te verdelen in twee soorten, namelijk Wijziging en Ruiling. Een wijziging wordt geïnitieerd vanuit de organisatie en een ruiling is een afspraak tussen twee medewerkers die een dienst met elkaar ruilen. Het onderscheid tussen deze soorten heeft te maken met het uitbetalen van de salarissen. Als een medewerker extra werkt door een wijziging dan krijgt hij deze uren uitbetaald, maar als het extra werk veroorzaakt wordt door een ruiling dan krijgt hij de extra uren niet uitbetaald.

#### Stap 3: Registreren.

Na het uitvoeren van een dienst, moeten de medewerkers hun urenbriefje inleveren. Het is de taak van de planner om de urenbriefjes over te zetten in HARMONY. Wanneer de planner dit gedaan heeft, kan hij per dienst of voor meerdere diensten tegelijk de status op *Verantwoord* zetten. Dit betekent dat deze diensten niet meer aangepast kunnen worden en gereed zijn voor goedkeuring door een leidinggevende.

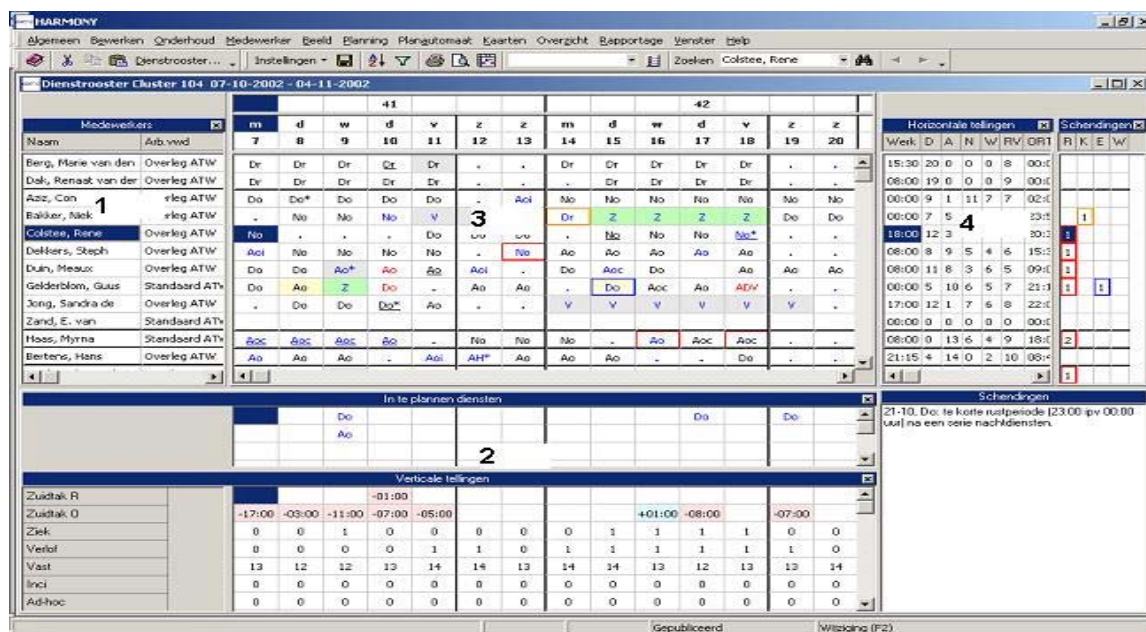
#### Stap 4: Controleren.

Het is de taak van een leidinggevende, bijvoorbeeld een afdelingshoofd, om de urenregistraties te controleren. Als de leidinggevende akkoord gaat verandert hij de status van de diensten in *Geaccordeerd*.

#### Stap 5: Verwerken.

De laatste stap in het proces wordt gedaan door de loonadministratie. Het is hun taak om de medewerkers hun salaris uit te betalen op basis van de verkregen informatie. Na het uitvoeren van deze stap wordt de status van de diensten op *Verwerkt* gezet en is het roosterproces beëindigd.

Voor mijn afstudeerproject waren vooral de stappen 1 tot en met 3 van belang, omdat deze stappen door de planner worden uitgevoerd en hier dus werk overgenomen moet kunnen worden door de medewerker. Hieronder wordt uitgelegd hoe deze drie stappen uitgevoerd worden. Het hulpmiddel dat de planner gebruikt is het planbord (zie Figuur 6). Het planbord is het belangrijkste onderdeel van HARMONY, omdat hierin het actuele rooster wordt getoond. Aanpassingen worden in dit scherm in het rooster doorgevoerd.



Figuur 6: Planbord

Het planbord is op te delen in vier velden. Hieronder wordt aangegeven wat er in deze velden staat en wat dit betekent.

#### Veld 1: Medewerkers

In dit veld worden alle in te roosteren medewerkers getoond.

#### Veld 2: In te plannen diensten en Verticale tellingen

In dit veld worden de in te plannen diensten en de verticale tellingen getoond. De in te plannen diensten zijn de diensten die nog niet aan een medewerker gekoppeld zijn. Met de verticale tellingen wordt informatie over de ingeplande diensten per dag bijgehouden. Een voorbeeld is het aantal keren dat een bepaald soort dienst is ingeroosterd.

#### Veld 3

Dit is de roostermatrix. De roostermatrix bestaat uit een groot aantal cellen. Voor elke medewerker (veld 1) is er op elke dag een aparte rooster cel. In deze cellen kunnen diensten (veld 2) geplaatst worden. Door middel van kleuren wordt aangetoond wanneer een dienst niet bij een medewerker geplaatst mag worden. Dit wordt in veld 4 getoond.

#### Veld 4: Horizontale tellingen en Schendingen

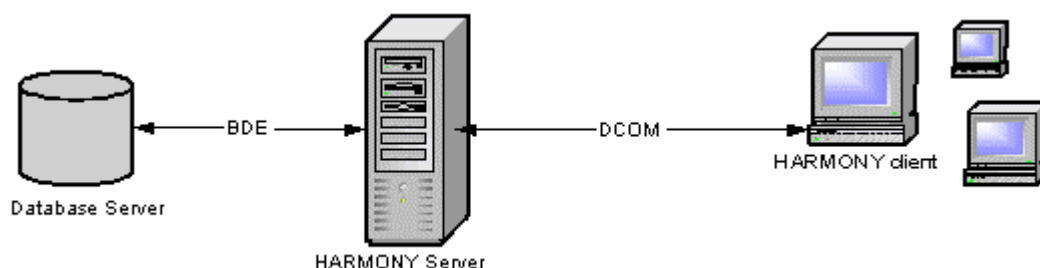
In dit veld worden de horizontale tellingen en schendingen van regels per medewerker bijgehouden, bijvoorbeeld voor hoeveel dagen hij ingeroosterd is in een bepaalde periode. Als dit aantal meer is dan wettelijk is toegestaan wordt dit aangegeven. Voorbeelden van mogelijke regels zijn CAO-regelingen, maar ook wensen van een medewerker.

Naast het inplannen van medewerkers is het ook mogelijk om per dienst aan te geven hoeveel uur er daadwerkelijk aan de ingeplande activiteiten is besteed en of alle ingeplande activiteiten zijn uitgevoerd. Door onder andere deze gegevens te registreren is het mogelijk om rapporten te genereren. Voorbeelden van deze rapporten zijn ziekerapporten,

presentielijsten, toeslagenrapporten en dienstrooster. Deze rapporten zijn bruikbaar voor verschillende zaken, zoals het verkrijgen van een overzicht van het ziekteverzuim in een bepaalde periode.

## 6.2 Technische analyse

In deze paragraaf staat de technische analyse van HARMONY. De analyse heb ik gedaan om te begrijpen hoe HARMONY werkt, voor zover dat nodig was voor mijn opdracht. HARMONY is een client/server-applicatie. Figuur 7 toont de architectuur met daaronder een toelichting van elk onderdeel.



Figuur 7: Architectuur HARMONY

### *HARMONY Client*

HARMONY Client verzorgt onder andere de interface naar gebruiker en communiceert met HARMONY Server.

### *HARMONY Server*

HARMONY Server regelt al het verkeer tussen de database en de HARMONY clients en voor het meeste rekenwerk uit.

### *Database server*

Op de database server wordt de data van HARMONY opgeslagen in een database.

### *BDE (Borland Database Engine)*

De BDE heeft tot doel de verbinding tot stand te brengen tussen een applicatie en een database. De database engine verzorgt alle database afhandelingen. Met de BDE is het mogelijk om gegevens in allerlei soorten databases op te slaan, zoals ORACLE, SQL Server en Interbase.

### *DCOM (Distributed Component Model)*

DCOM is het onderdeel van Microsoft's COM (Component Object Model), dat het mogelijk maakt om gedistribueerde applicaties te maken. DCOM is een verzameling van Microsoft concepten en interfaces waarin een client-object diensten kan aanvragen van server-objecten die op andere computers staan in een netwerk. Gedistribueerde applicaties zijn applicaties waarvan de componenten over meerdere computers verspreid zijn. Binnen gedistribueerde applicaties is het mogelijk dat een component die op de ene computer draait een component aanroept die op een andere computer draait. Als de aangeroepen component klaar is wordt het resultaat teruggestuurd naar de aanroepende component. DCOM maakt gebruik van de protocollen TCP/IP en HTTP en werkt daardoor op lokale netwerken (LAN's), maar ook over internet.

*Delphi 5*

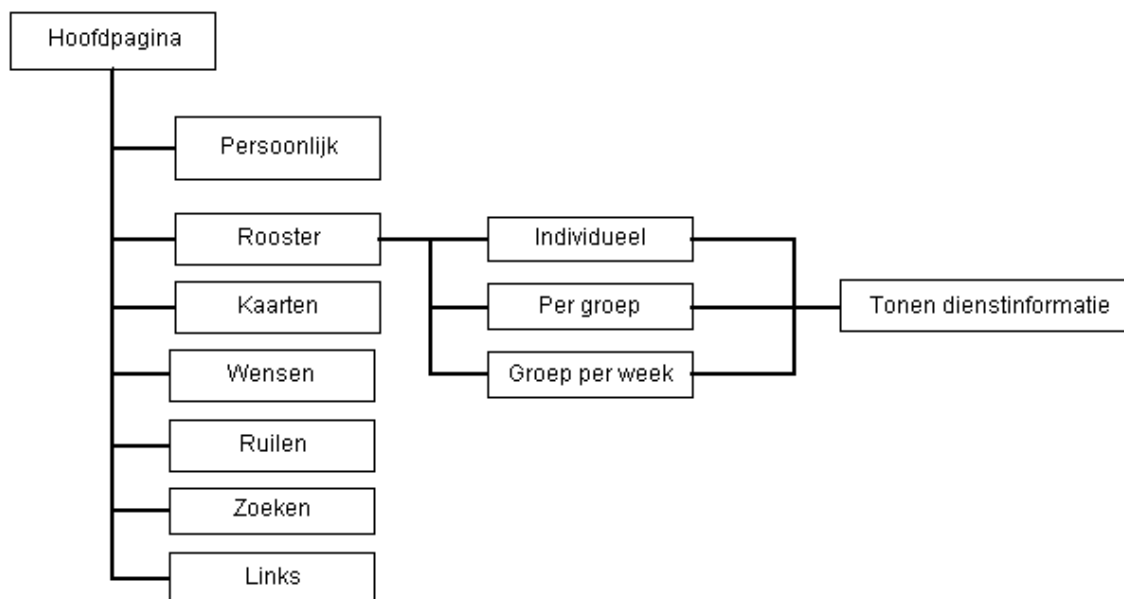
HARMONY Server wordt ontwikkeld in Delphi 5. Delphi (Object Pascal) ondersteunt het objectgeoriënteerd ontwikkelen. Delphi is een product van Borland en is ontstaan uit de programmeertaal Turbo Pascal.

## 6.3 HARMONY WebAccess

HARMONY WebAccess (WebAccess) is een uitbreiding van HARMONY die het mogelijk maakt via internet gebruik te maken van HARMONY. Waar HARMONY vooral gericht is op de planner is WebAccess bedoeld voor medewerkers met een rooster. Momenteel is nog niet alle functionaliteit uit HARMONY beschikbaar voor WebAccess. Paragraaf 6.3.1 beschrijft welke functionaliteit er wel aanwezig is en paragraaf 6.3.2 van welke technieken WebAccess gebruikt maakt.

### 6.3.1 Functionele analyse

De functionele analyse van HARMONY WebAccess heb ik uitgevoerd om te kunnen bepalen wat voor functies er al waren. Aan de hand van deze analyse wist ik welke functionaliteit HARMONY WebAccess al had en of ik bepaalde delen daarvan kon gebruiken voor mijn opdracht. Voor het analyseren van de functionele mogelijkheden van HARMONY WebAccess, heb ik gebruik gemaakt van het afstudeerverslag van de afstudeerder die het eerste prototype van HARMONY WebAccess heeft gemaakt en de applicatie HARMONY WebAccess. Figuur 8 toont de menustructuur van HARMONY WebAccess met daaronder een toelichting.



Figuur 8: Oude Menustructuur HARMONY WebAccess

#### Hoofdpagina / Persoonlijk

De hoofdpagina is de pagina waarop de gebruiker inlogt. Als het inloggen goed is gegaan, wordt de pagina Persoonlijk getoond. Op deze pagina staan de persoonlijke gegevens van de ingelogde medewerker en de ruilaanvragen (zie kopje Ruilen). Op de hoofdpagina wordt ook een menu getoond waarvandaan men naar een andere pagina kan gaan.

#### Rooster

Door op *Rooster* te klikken krijgt de medewerker de keuze uit drie verschillende weergaven van zijn rooster. De weergaven zijn:

- Individueel. Dit is het rooster van de ingelogde medewerker en geeft informatie over de dagen waarop hij dienst heeft.
- Per groep. Deze pagina toont het rooster van de roostergroep waartoe de ingelogde medewerker behoort. De ingelogde medewerker kan in deze pagina bekijken wanneer een collega dienst heeft. De roosters worden gegroepeerd per persoon.
- Groep per week. Deze pagina toont hetzelfde rooster als de pagina Per groep alleen nu wordt het rooster getoond gegroepeerd per week. In één overzicht kan dus bekeken worden wanneer de ingelogde medewerker dienst en op welke dagen zijn collega's.

Door op een dienst te klikken wordt de pagina *Tonen dienstinformatie* geopend. Hierin worden de activiteiten getoond die bij de aangeklikte dienst horen met daarbij de informatie over de werktijden.

#### *Kaarten*

In HARMONY wordt door middel van een kaartmechanisme bijgehouden hoeveel verlof-, ziekte- en vakantiedagen medewerkers hebben opgenomen en hoeveel overuren ze hebben gemaakt. Voor de medewerker die gebruik maakt van HARMONY WebAccess is het mogelijk om deze kaartinfo te bekijken.

#### *Wensen*

Op deze pagina heeft de medewerker de keuze uit drie vormen van wensen. De vormen van wensen betreffen:

- Wel / niet werken in een periode. Er wordt een venster getoond, waarin de medewerker door middel van een begin- en einddatum kan aangeven in welke periode hij wel of niet wenst te werken;
- Dienstwens opgeven. In dit venster kan de medewerker aangeven welke dienst hij op een bepaalde dag niet wil draaien;
- Verlofwens opgeven. In dit venster kan de medewerker aangeven in welke periode hij vrij wil hebben;

Op basis van de kaartinfo weet hij hoeveel verlofdagen hij nog heeft. Na het aangeven van de dag(en) waarop hij niet wil werken wordt de achtergrondkleur van de desbetreffende dag in het planbord in HARMONY veranderd, waardoor de planner kan zien dat er een wens is ingevoerd.

#### *Ruilen*

Op de pagina *Ruilen* heeft de medewerker de mogelijkheid een ruilingaanvraag te doen. Deze aanvraag gebeurt door het invullen van de datum van de dag waarop de medewerker wil ruilen. Vervolgens wordt een overzicht getoond van collega's waarmee eventueel geruild kan worden. Door een naam van een collega aan te klikken wordt een ruilaanvraag gedaan. Deze ruilaanvraag wordt getoond op de pagina *Persoonlijk*. Zodra de collega deze ruilaanvraag accepteert is de dienst geruild, zonder tussenkomst van de planner.

#### *Zoeken*

Op deze pagina kan de medewerker opzoeken op welke dag een bepaalde dienst wordt gedraaid en door wie dit wordt gedaan.

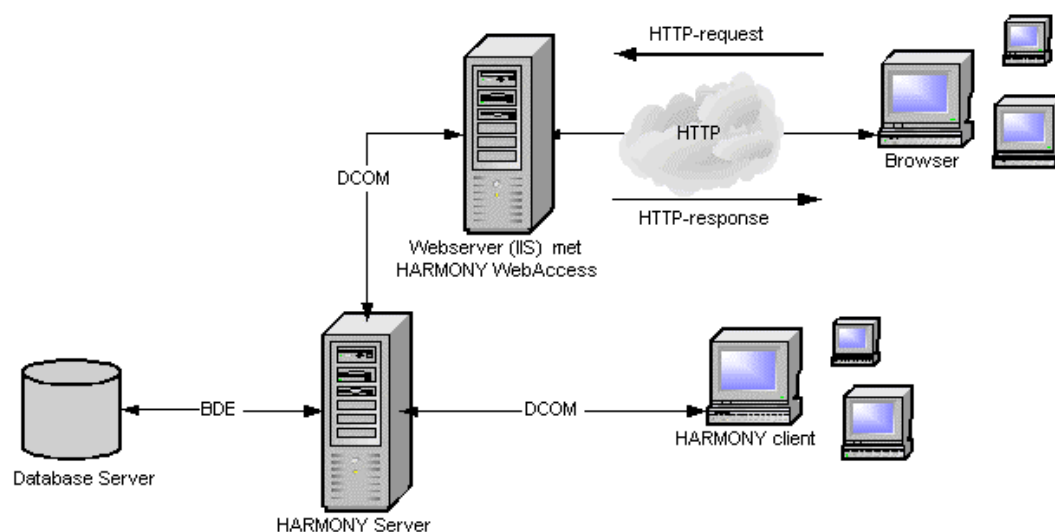
#### *Links*



Op deze pagina heeft de klant de mogelijkheid om verwijzingen (links) naar belangrijke informatie voor de medewerkers te zetten, bijvoorbeeld een verwijzing naar het personeelshandboek.

### 6.3.2 Technische analyse

Met het analyseren van de technische aspecten van HARMONY WebAccess kon ik een beeld krijgen van de werking van HARMONY WebAccess. Dit had ik nodig bij het uitvoeren van mijn opdracht, omdat er naast aanpassingen in HARMONY WebAccess ook aanpassingen in HARMONY Server gedaan moesten worden. Figuur 9 toont de architectuur van HARMONY met HARMONY WebAccess met daaronder een toelichting van de specifieke onderdelen van HARMONY WebAccess.



Figuur 9: Architectuur HARMONY met HARMONY WebAccess

Wat in dit figuur opvalt is dat HARMONY WebAccess (HW) dezelfde plaats heeft als de HARMONY Client (HC). Hier zit ook het verschil tussen HARMONY en HW. De gebruiker van een HC communiceert direct met HARMONY Server (HS). De gebruiker van HW communiceert via een browser met HW die weer communiceert met HS. Hieronder wordt toelichting gegeven op de verschillen.

#### IIS (Internet Information Services)

Om HW via internet te kunnen gebruiken is een webserver nodig. Een webserver maakt het mogelijk vanuit een browser toegang te krijgen tot bepaalde applicaties. De webserver waarop HW draait is IIS 5.0. IIS is een product van Microsoft en maakt het mogelijk om op een eenvoudige manier websites te hosten en te beheren.

#### ISAPI DLL

HW is een ISAPI<sup>8</sup> DLL<sup>9</sup>. Een DLL is een applicatie die na het opstarten in het geheugen blijft geladen. Het voordeel hiervan is dat gebruikers na het inloggen direct gebruik kunnen maken van de applicatie, omdat er geen opstarttijd is.

#### WebSnap

<sup>8</sup> ISAPI = Internet Server Application Programming Interface

<sup>9</sup> DLL = Dynamic Link Library

WebSnap is een componentenverzameling in de ontwikkelomgeving van Delphi 6 en hiermee is het mogelijk om web applicaties te maken. WebSnap ondersteunt onder andere sessiebeheer, wachtwoord-controle en een deel threading-beheersing. Met sessiebeheer en threading-beheersing zorgt WebSnap dat de requests (vragen) van de sessies het juiste response (antwoord) terug krijgen. De wachtwoord-controle is een component die de ingevoerde combinatie van gebruikersnaam en wachtwoord controleert met de login-gegevens in de database. Deze onderdelen zijn essentieel voor HW. WebSnap maakt bij het weergeven van een pagina gebruik van twee elementen. In het ene element staat de Delphi-code en in het andere element staat de HTML-code. Met behulp van de Delphi-code is het mogelijk om dynamische code te genereren. In de HTML-code kan de vormgeving gespecificeerd worden en eventueel gebruik gemaakt worden van scripting om bepaalde acties in de browser te kunnen uitvoeren.

#### *Sessie*

Een sessie is een opeenvolging van uitwisselingen van request-response boodschappen in een welbepaalde volgorde met dezelfde client. De sessie wordt gestart op het moment van inloggen en wordt afgesloten zodra de ingelogde persoon heeft uitgelogd of wanneer hij 20 minuten niets doet.

#### *DCOM*

DCOM wordt ook gebruikt voor de communicatie tussen HS en HW. Het verschil is dat alle requests die via HW binnen komen over één DCOM-connectie naar HS gaan. Er is dus sprake van multi-threading. Met behulp van een standaard variabele in Delphi 6, de variabele *Threadvar*, wordt geregeld dat sessies ook aan de kant van HS goed verlopen.

#### *Delphi 6*

HW wordt ontwikkeld in Delphi 6. ARTM heeft gekozen voor Delphi 6, omdat deze versie een extra componentenverzameling heeft waardoor het makkelijker is om web applicaties te bouwen. Deze componentenverzameling heet WebSnap. HS wordt nog ontwikkeld in Delphi 5, omdat het veel geld gaat kosten om over te stappen met daarbij de onnodige risico's die men loopt bij het overstappen. Aangezien alleen de persoon die zich bezig houdt met de ontwikkeling van HARMONY WebAccess er baat bij heeft om versie 6 te hebben, is ervoor gekozen om deze persoon gebruik te laten maken van twee versies.

#### *JavaScript*

Om vanuit de webbrowser te kunnen communiceren met HARMONY WebAccess was meer nodig dan alleen HTML. HTML is namelijk een taal om een webpagina op een bepaalde manier weer te geven. Om interactie te kunnen hebben met HARMONY heeft ARTM een programmeertaal gekozen waarmee de webpagina's dynamisch gemaakt konden worden. ARTM heeft gekozen voor de programmeertaal JavaScript. De reden waarom ARTM heeft gekozen voor JavaScript en bijvoorbeeld niet voor VBScript, heeft te maken met het browseronafhankelijk te zijn. JavaScript is ontwikkeld door Netscape en kan zodoende in een Netscape browser gebruikt worden, maar ook in Microsoft Internet Explorer. VBScript is een product van Microsoft dat niet door alle browsers wordt ondersteund. Met behulp van JavaScript kunnen HTTP-requests doorgegeven worden aan HARMONY WebAccess.

#### *Web App Debugger*

Met de Web App Debugger is het mogelijk om HARMONY WebAccess als een EXE applicatie te draaien. Web App Debugger is een tool die meegeleverd wordt in de ontwikkelomgeving van Delphi 6 en dient als alternatieve webserver. HARMONY WebAccess is een DLL, maar voor het project is een executabel gebruikt. Het is namelijk lastiger om een DLL te debuggen dan een executabel. Het nadeel van het debuggen van een DLL ten op zichte van een executabel, is dat elke keer de oude DLL verwijderd moet worden voordat een nieuwe DLL opgestart kan worden.

Het voordeel van HW is dat de gebruiker alleen een browser nodig heeft om toegang te krijgen tot HARMONY. Op dit moment kan HW vanwege problemen met security-settings alleen op dezelfde computer draaien als waar HS op draait.

## 7. Functioneel Ontwerp

Dit hoofdstuk geeft inzicht in het tot stand komen van het product Functioneel Ontwerp. Paragraaf 7.1 beschrijft de gewenste functionaliteit, de prioriteit van elke functie en waarom de desbetreffende functies ontwikkeld moesten worden. Paragraaf 7.2 geeft aan hoe de gewenste functionaliteit is uitgewerkt en waarom er voor deze manier is gekozen. In paragraaf 7.3 wordt weergegeven welke aanpassingen aan de functionele structuur van de webpagina's van HARMONY WebAccess gedaan moesten worden. Paragraaf 7.4 toont het class diagram waarin de benodigde classes worden weergegeven.

### 7.1 Gewenste functionaliteit

De gewenste functionaliteit heb ik bepaald door het houden van een brainstormsessie, zoals beschreven in hoofdstuk 3. Het doel van de brainstormsessie was het bepalen van de functionele eisen en de prioriteit van elke eis. Het stellen van prioriteiten was nodig om te bepalen in welk increment een bepaalde functionaliteit ontwikkeld moest worden. Bij ARTM is er geen standaard manier voor het aangeven van een prioriteit. Ik heb bij de prioritering daarom gebruik gemaakt van de prioritering zoals DSDM dat voorschrijft. Ter ondersteuning van de sessie had ik ervoor gezorgd dat de groep HARMONY WebAccess kon zien, zodat iedereen kon zien wat de mogelijkheden waren. Hieronder worden het resultaat van de brainstormsessie getoond met daaronder een toelichting van de functies:

- Uren verantwoorden – prioriteit 1;
- Overzicht van mutaties en gezien – prioriteit 1;
- Rooster gezien – prioriteit 2;
- Overzicht wensen – prioriteit 2;
- Wachtwoord wijzigen – prioriteit 3;
- Kleuren tonen – prioriteit 4;
- Print-optie – prioriteit 4.

De functies *Print-optie* en *Kleuren tonen* hebben prioriteit 4 gekregen, omdat met het ontwikkelen van deze functies geen wezenlijke bijdrage geleverd kon worden bij het halen van de doelstelling van mijn opdracht, namelijk het zoveel mogelijk ontlasten van de planners. Daarom zijn deze functies niet meegenomen in het verdere ontwikkelingstraject. Dit geldt eigenlijk ook voor het wijzigen van het wachtwoord. Deze functie heb ik echter meegenomen, omdat ik op deze manier ervaring kon opdoen met de ontwikkelomgeving.

#### *Uren verantwoorden*

Met de functionaliteit *Uren verantwoorden* kan de medewerker zelf zijn gewerkte uren in HARMONY invoeren. In de oude situatie was het voor de medewerker alleen mogelijk om te zien hoe zijn dienst er op een bepaalde dag eruit zag. Met het ontwikkelen van deze functionaliteit kan de medewerker zelf aangeven wat voor activiteiten hij tijdens een dienst heeft gedaan en hoe lang de desbetreffende activiteiten hebben geduurd. Dit betekent dat de medewerker activiteiten moet kunnen wijzigen, verwijderen en / of toevoegen. Als er geen wijzigingen zijn hoeft de medewerker de dienst alleen op te slaan.

#### *Overzicht van mutaties en gezien*

De bedoeling van deze functionaliteit is dat de medewerker bij het openen van de pagina *Overzicht mutaties* een overzicht krijgt te zien van de eventuele mutaties die de planner heeft gedaan nadat een rooster is gepubliceerd. Verder moet de medewerker aan kunnen geven dat hij de mutaties heeft gezien. De planner hoeft dan niet meer te bellen om door te geven dat er een wijziging in zijn rooster heeft plaats gevonden.

#### *Rooster gezien*

Met deze functionaliteit kan de medewerker aangeven dat hij zijn (nieuwe) rooster heeft gezien, zodat de planner niet het rooster hoeft op te sturen naar de medewerker.

#### *Overzicht wensen*

Met deze functionaliteit kan de medewerker zien welke wensen hij ingevoerd heeft. In de oude situatie was het alleen mogelijk om wensen in te voeren. De medewerker kon toen niet zien of de wens goed ingevuld was en wat de status (open, geaccepteerd of geweigerd) van de wens was. Om hier achter te komen moest er alsnog contact gezocht worden met de planner of de planner moest aan de medewerker doorgeven dat de wens geaccepteerd dan wel geweigerd was.

#### *Wachtwoord wijzigen*

Deze functie geeft de medewerker de mogelijkheid zelf zijn (toegewezen) wachtwoord te wijzigen.

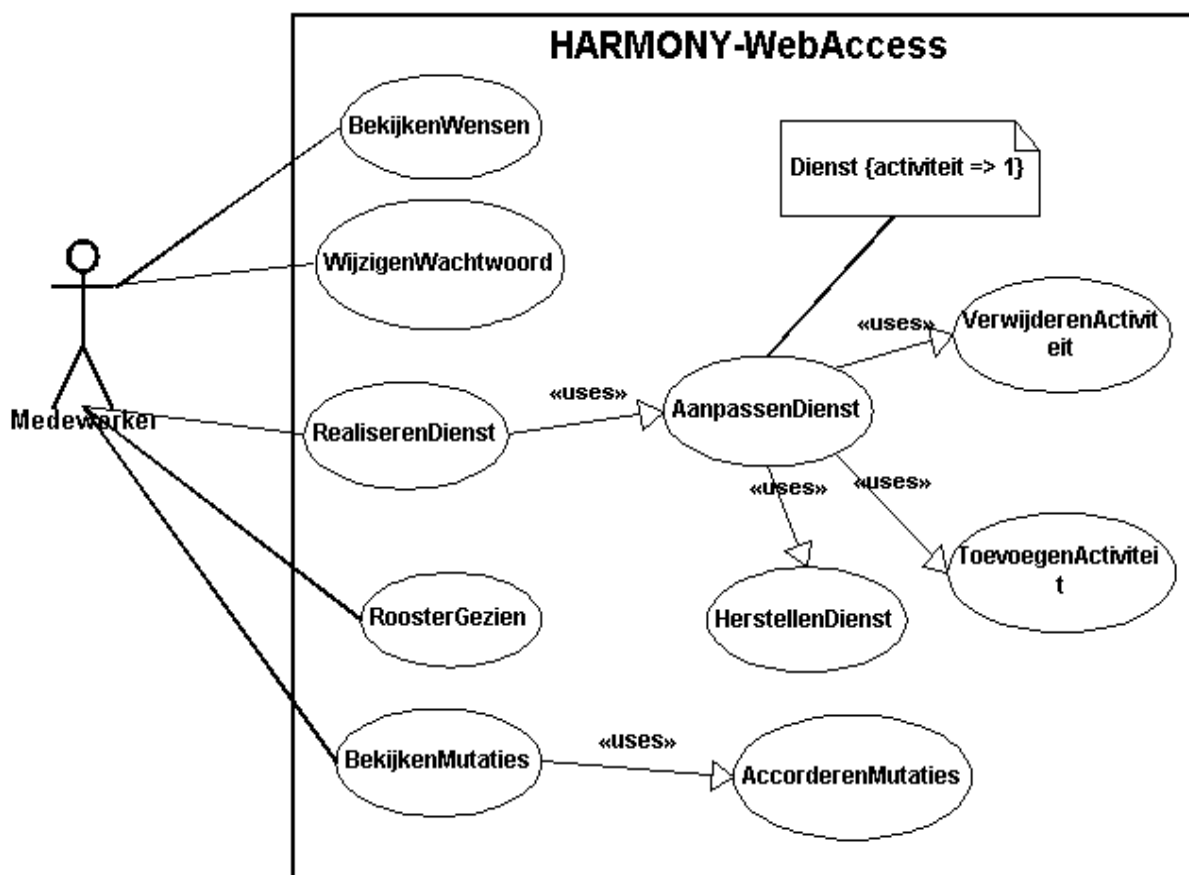
## 7.2 Functionaliteit uitgewerkt

Bij het uitwerken van de gewenste functionaliteit is gebruik gemaakt van use-cases. Een use-case beschrijft een reeks van interacties tussen één of meer gebruikers en het systeem. Deze beschrijving geeft aan wat het systeem moet doen bij een bepaalde actie van de gebruiker. Figuur 10 toont een voorbeeld van een use-case zoals die zijn terug te vinden in 0.

Naam	Realiseren dienst
Aannamen	Medewerker is ingelogd en zijn dienstrooster wordt getoond (link 'Rooster – Mijn rooster' in menu)
Beschrijving	(1) Medewerker klikt op een dienst. Als de datum en begintijd van de originele dienst groter is dan de huidige datum en tijd en / of status van de dienst is ongelijk aan 'Gepubliceerd' en / of dienst behoort niet bij de ingelogde persoon dan treedt uitzondering op. (2) Als de datum en tijd gelijk of kleiner is dan de huidige datum en tijd en status van de dienst is 'Gepubliceerd' en de dienst hoort bij de ingelogde persoon dan toont het systeem de gepubliceerde dienst (van elke activiteit ziet de medewerker de naam, het type, de werkplek, begintijd en eindtijd). (3) Medewerker kan nu de dienst realiseren of aanpassen. Als medewerker een dienst wil aanpassen wordt use-case 'Aanpassen dienst' uitgevoerd. Als hij wil stoppen treedt uitzondering op. (4) Medewerker geeft zijn akkoord door op de 'Opslaan'-button te klikken. (5) Voordat het akkoord definitief is krijgt de medewerker de melding "Weet u zeker dat u alles goed heeft ingevuld?". Zo nee dan treedt uitzondering op. Zo ja dan klikt de medewerker op de 'Ok'-button en keert terug naar zijn dienstrooster.
Uitzonderingen	[Datum en begintijd van ingeplande dienst is groter dan datum en tijd van huidige dag en / of status van de dienst is ongelijk aan 'Gepubliceerd' en / of dienst behoort niet bij de ingelogde persoon] De dienstgegevens worden getoond, maar alle buttons, icoontjes en invoervelden zijn onzichtbaar of gedisabled.  [Medewerker wil stoppen] Medewerker klikt op de 'Cancel'-button of op het kruisje en keert terug naar het dienstrooster in het scherm 'Rooster – Mijn rooster'.  [Medewerker is niet zeker dat hij alles goed heeft ingevuld] Medewerker klikt op de 'Cancel'-button en keert terug in het scherm van de geselecteerde dienst.
Resultaat	Medewerker heeft dienst gerealiseerd, in HARMONY is zichtbaar dat de dienst gerealiseerd is en HARMONY WebAccess toont het dienstrooster van de medewerker op het scherm.

Figuur 10: Use-case Realiseren dienst

Na het maken van de use-cases is een use-case diagram gemaakt (zie Figuur 11). Een use-case diagram geeft een overzicht van de mogelijkheden die het systeem (HARMONY WebAccess) aan de gebruiker geeft. De gebruiker is in dit geval de medewerker.



Figuur 11: Use-case diagram

Zoals in het use-case diagram is te zien, maakt de use-case *Realiserendienst* gebruik van meerdere use-cases, namelijk *AanpassenDienst*, *VerwijderenActiviteit*, *ToevoegenActiviteit* en *HerstellenDienst*. Deze use-cases beschrijven tezamen het proces *Uren verantwoorden*. Voor de technische aspecten heb ik overlegd met mijn bedrijfsmentoren. Een van de discussiepunten bij het maken van het Functioneel Ontwerp, was het wel of niet toevoegen van een status bij het een dienst. Zoals in paragraaf 6.1 staat beschreven kan een planning of een dienst een bepaalde status hebben. Mijn idee was om een status toe te voegen tussen de statussen *Gepubliceerd* en *Verantwoord*, zodat:

- De planner kon zien dat een medewerker zijn dienst(en) heeft aangepast en / of opgeslagen;
- Na wijziging van de status de medewerker niet meer de mogelijkheid zou hebben om aanpassingen te doen.

In overleg met mijn bedrijfsmentoren heb ik echter besloten geen status toe te voegen. Deze beslissing is genomen op basis van de volgende redenen:

- ARTM had besloten om de keuzelijst van mogelijke statussen in te korten, omdat er al te veel statussen waren;
- Een probleem dat op kan treden is dat een medewerker iets verkeerd heeft ingevoerd en daar achter komt nadat hij de dienst heeft opgeslagen. Om het te veranderen moet contact gezocht worden met de planner die dan de status terug moet zetten of zelf de dienst aanpast.

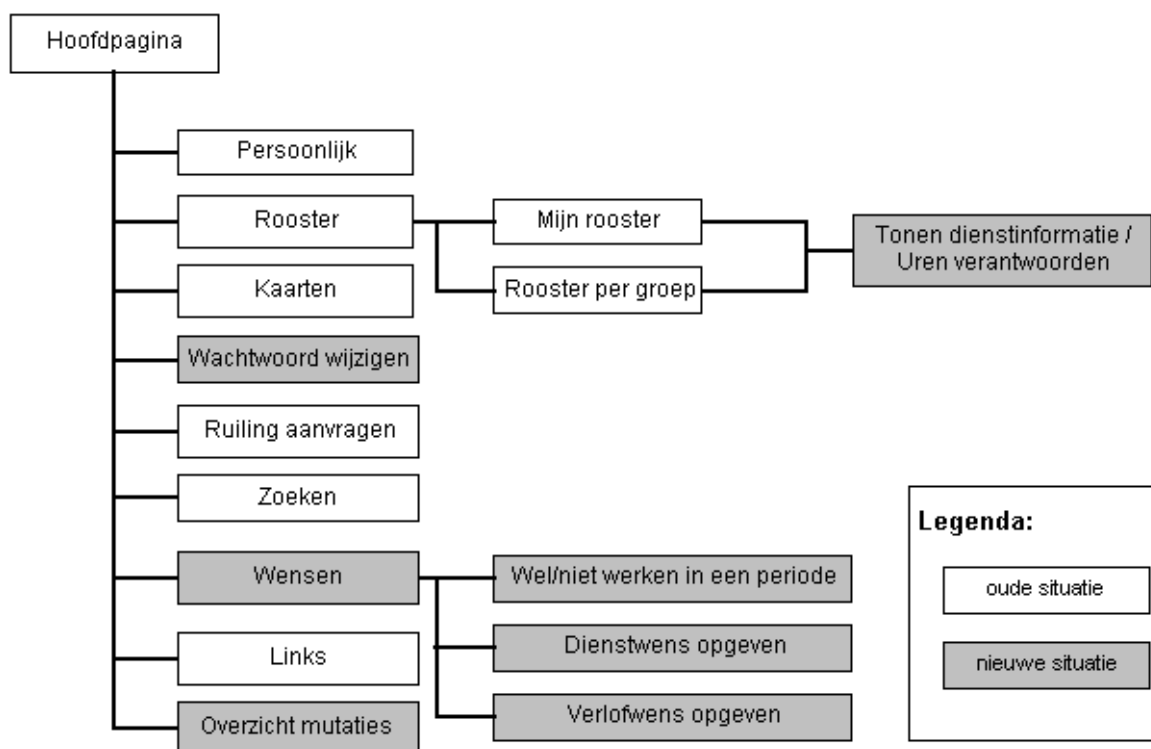


- De planner kan in het planbord zien dat een dienst is gewijzigd en / of opgeslagen aan het streepje onder de dienstnaam.

Bij het bedenken van een goede indeling van de schermen, heb ik regelmatig overlegd met een collega, mevrouw M. Jungman<sup>10</sup>, om haar mening te horen. Verder heb ik gebruik gemaakt van de gebruikershandleiding van HARMONY en het systeem HARMONY, omdat een niet-functionele eis was dat HARMONY WebAccess zoveel mogelijk de look-and-feel van HARMONY moest hebben.

### 7.3 Menustructuur HARMONY WebAccess

Met het toevoegen van nieuwe functies in HARMONY WebAccess is de structuur, zoals getoond in paragraaf 6.3.1, gewijzigd. In Figuur 12 wordt de gewijzigde menustructuur van HARMONY WebAccess weergegeven.



Figuur 12: Nieuwe menustructuur HARMONY WebAccess

De grijze vierkantjes tonen de gewijzigde of nieuwe pagina's, ook zijn er enkele naamswijzigingen aangebracht. Deze naamswijzigingen waren een wens van Mevr. M. Jungman. De nieuwe benamingen geven een beter beeld van welke informatie op de desbetreffende pagina is te vinden, bijvoorbeeld de pagina *Individueel* is gewijzigd in *Mijn rooster*. Een andere wijziging is het verwijderen van de pagina *Rooster per week*. Bij mijn analyse van HARMONY WebAccess heb ik me afgevraagd wat het nut van deze pagina was, omdat deze pagina dezelfde informatie gaf als de pagina *Rooster per groep*. In overleg met Mevr. M. Jungman heb ik besloten om

<sup>10</sup> Mevr. M. Jungman is verantwoordelijk voor de lay-out van schermen.

deze pagina weg te laten. Hieronder worden de gewijzigde en / of nieuwe pagina's toegelicht.

#### *Pagina Tonen dienstinformatie / Uren verantwoorden*

Deze pagina toonde de activiteiten die bij de door de medewerker aangeklikte dienst hoorden. Voor het proces Uren verantwoorden was het van belang dat de getoonde activiteiten gewijzigd en / of verwijderd konden worden. Ook moest het mogelijk zijn om activiteiten toe te voegen. Het scherm *Tonen dienstinformatie* kan vanuit twee pagina's opgevraagd worden, het was echter niet de bedoeling dat medewerkers de diensten van collega's aan konden passen. Daarom was de eis dat het tonen van dienstinformatie altijd mogelijk moest zijn, maar het wijzigen van de dienstinformatie (*Uren verantwoorden*) niet altijd. Andere eisen waren dat een dienst niet gewijzigd mocht worden als zijn status anders is dan *Gepubliceerd* en als de dienst nog niet is uitgevoerd.

#### *Pagina Wachtwoord wijzigen*

Deze pagina was nieuw. Voor het bepalen van wat er op deze pagina getoond moest worden, heb ik gekeken hoe dit in Windows 2000 (besturingssysteem op mijn computer) gedaan wordt. Ik heb gekozen voor vier velden en een knop (button) waarmee de wijziging bevestigd kon worden. De gebruikersnaam wordt getoond maar kan niet gewijzigd worden. Het oude wachtwoord moet ingevuld worden en het nieuwe wachtwoord twee keer. De eisen die aan het wachtwoord gesteld worden zijn:

- Niet korter dan 5 karakters, deze eis heb ik opgesteld omdat er anders hele korte wachtwoorden gekozen worden en die kunnen makkelijk gekraakt worden;
- Niet langer dan 19 karakters, anders kan het wachtwoord niet opgeslagen worden in de database.

#### *Pagina Wensen*

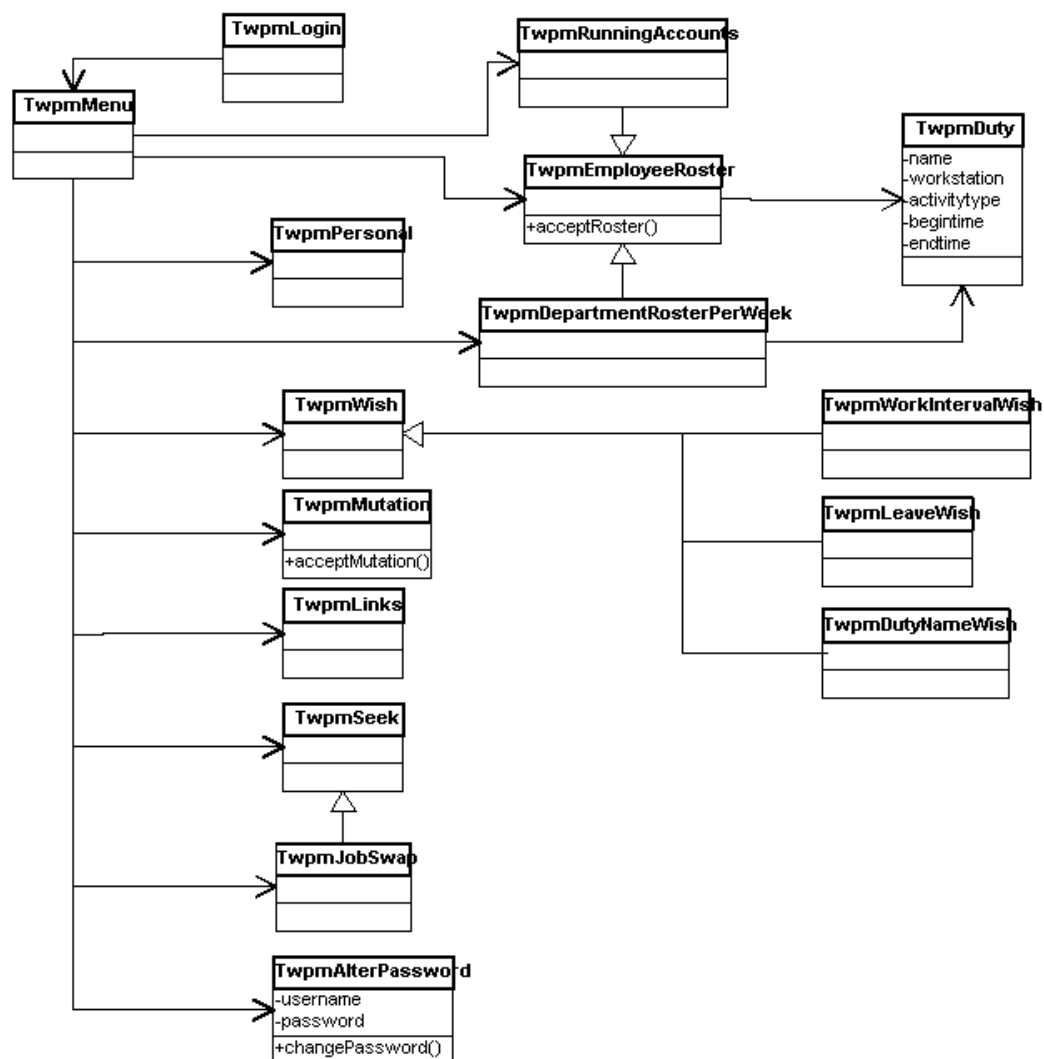
Op de oude pagina was het mogelijk een bepaalde wens in te voeren. Het was echter voor de medewerker niet zichtbaar of zijn ingevoerde wens ook daadwerkelijk ingevoerd was en welke wensen hij al had ingevoerd. De interne gebruikersgroep vond dat op deze pagina ook een overzicht moest komen van de ingevoerde wensen. Door het gebruik van vensters op deze pagina zou het erg onoverzichtelijk worden als er ook nog een overzicht bij zou komen. Daarom heb ik besloten om het invoeren van de wensen op aparte pagina's te doen. Dit is zijn de pagina's *Wel/niet werken in een periode*, *Dienstwens opgeven* en *Verlofwens opgeven*. Zodra de pagina wordt geopend is direct een overzicht te zien van de opstaande wensen en is voor de medewerker zichtbaar welke wensen gehonoreerd zijn en welke niet.

#### *Pagina Overzicht mutaties*

Wijzigingen die de planner in het rooster van de medewerker heeft gemaakt worden in deze pagina getoond. Op deze pagina kan de medewerker ook aangeven of hij een mutatie heeft gezien. De medewerker heeft de mogelijkheid om per mutatie aan te geven dat hij het heeft gezien, maar kan ook door een klik op de button *Accepteren Mutaties* aangeven dat hij alle mutaties heeft gezien.

## 7.4 Class diagram

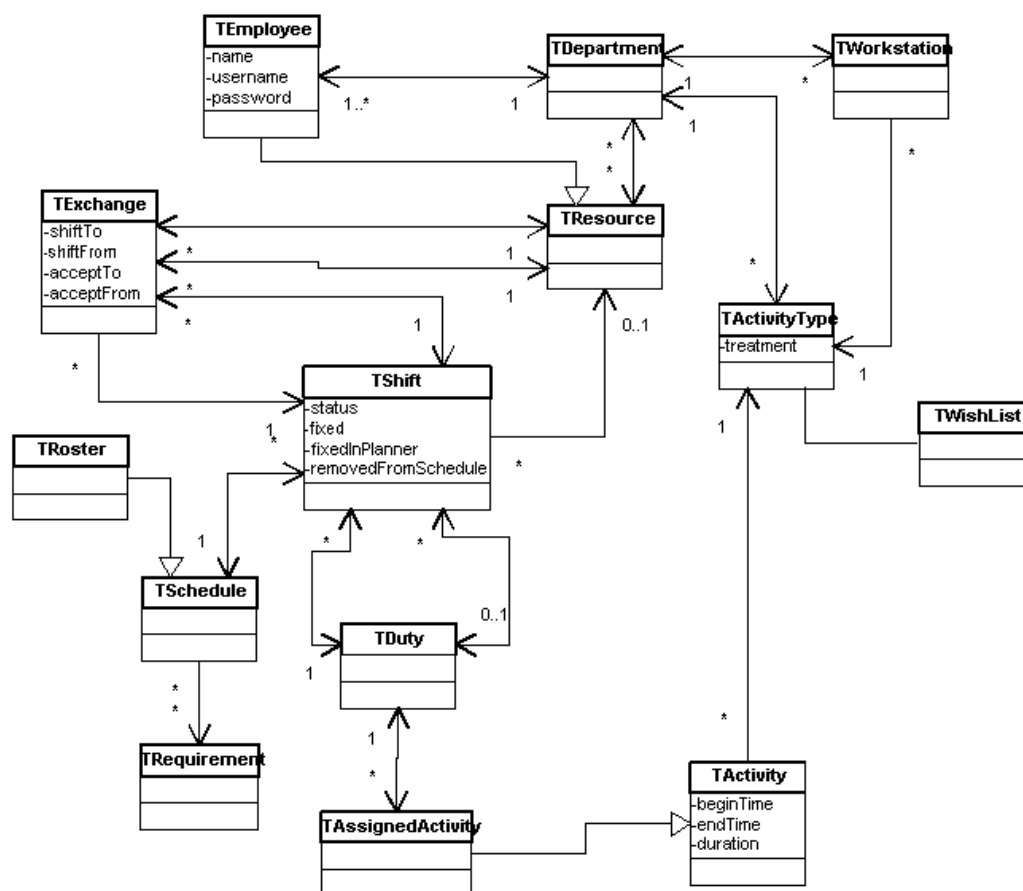
Om overzicht te krijgen van de classes die ik nodig had voor mijn opdracht heb een class diagram gemaakt van HARMONY WebAccess. Ik heb ook een vereenvoudigde versie van het bestaande class diagram van HARMONY gemaakt. Het class diagram van HARMONY WebAccess was eenvoudig, omdat er bijna geen relaties tussen de classes waren. Figuur 13 toont het class diagram van HARMONY WebAccess met daaronder een toelichting van het class diagram.



Figuur 13: Class diagram HARMONY WebAccess

De relaties van de class TwpmMenu zijn geen echte relaties, het geeft alleen aan dat vanuit deze class (het menu) deze classes opgevraagd kunnen worden. Met behulp van een "open" pijl heb ik de richting van de relatie weergegeven. De gesloten pijlen geven de overervingstructuur aan. Vanuit de classes van HARMONY WebAccess worden requests gestuurd naar HARMONY Server. HARMONY Server regelt dat de juiste response gestuurd wordt. Om overzicht te krijgen van de classes van HARMONY Server die ik nodig had voor mijn opdracht, heb ik een vereenvoudigde versie van het reeds bestaande en ingewikkelde class diagram van

HARMONY Server gemaakt. Figuur 14 toont het class diagram van HARMONY.



Figuur 14: Class diagram HARMONY

## 8. Technisch Ontwerp

Dit hoofdstuk beschrijft hoe ik het Technisch Ontwerp heb gemaakt. Paragraaf 8.1 beschrijft hoe ik de technische specificaties heb opgesteld, waarvan ik gebruik gemaakt heb en welke keuzes een rol speelden. Paragraaf 8.2 beschrijft het class diagram met daarin de relaties tussen de classes die ik nodig had.

### 8.1 Technische specificaties

Deze paragraaf geeft informatie over de inhoud en het opstellen van de technische specificaties van de gewenste functionaliteit. Voor de gewenste functionaliteit die ook in HARMONY aanwezig was, heb ik gebruik gemaakt van de beschikbare source-code van HARMONY. Een volledige uitwerking van de technische specificaties staat in Externe bijlage VI.

#### *WebSnap*

Bij het opstellen van de technische specificaties heb ik rekening gehouden met de mogelijkheden van WebSnap. Een WebSnap component bestaat zoals in paragraaf 6.3.2 is beschreven uit een Delphi-element (pas-file) en uit een HTML-element (html-file). In de checklists van de standaard van ARTM wordt wel verwezen naar tips voor het programmeren in Delphi, maar niets over het programmeren van web applicaties, zoals het gebruik van JavaScript-code en het gebruik van WebSnap. Het was mij tijdens de technische analyse van HARMONY WebAccess opgevallen dat er veel JavaScript-code werd gebruikt, terwijl veel van deze code ook aan de serverkant (Delphi) afgehandeld kon worden. De voordelen van het gebruik van Delphi-code zijn dat de code gecontroleerd kan worden door de Delphi-compiler en de code beter onderhoudbaar is, omdat dezelfde controles gebruikt kunnen worden als via de HARMONY client. De manier om zoveel mogelijk code in Delphi te houden is het gebruik maken van WebSnap-tags. In html-file kan een tag geplaatst worden en in de pas-file kan aan deze tag een bepaalde actie gehangen worden. Een tag ziet er als volgt uit:

*Ergens in de .HTML-file staat bijvoorbeeld deze tag:*

```
<#TOONTEKST>
```

*In de .pas-file kan vervolgens het volgende gedaan worden:*

```
If TagString = 'TOONTEKST' then  
  ReplaceText := 'Tekst op het scherm';
```

Voorbeelden van wat er met een tag gedaan kan worden, is:

- Het dynamisch genereren van veldnamen, zoals bij de pull-down menu's en checkboxen (zie volgende paragraaf);
- Het dynamisch maken van tekst. Afhankelijk van de ingestelde taal wordt de tekst getoond in die taal (hierbij worden de taal instellingen van HARMONY gebruikt);

Het dynamisch genereren van een tabel. Bijvoorbeeld bij het tonen van een dienst, moet er dynamisch activiteiten toegevoegd worden (er is van tevoren immers niet bekend hoeveel activiteiten een dienst heeft). Door in

de Delphi source-code de tabel samen te stellen kan de tabel in één keer naar de browser gestuurd worden.

#### *Wachtwoord wijzigen*

Om te kunnen bepalen hoe lang een wachtwoord maximaal mocht zijn, heb ik in de database gekeken hoe groot het dataveld van het wachtwoord was. Dit bleek 20 karakters te zijn, om er zeker van te zijn dat een ingevoerde wachtwoord kon worden opgeslagen heb ik een maximum van 19 karakters aangehouden.

#### *Uren verantwoorden*

Bij het Uren verantwoorden moest het mogelijk zijn om activiteiten van een dienst te wijzigen, toe te voegen of te verwijderen. In het Technisch Ontwerp heb ik een tabel opgenomen met daarin de velden die ge-edit kunnen worden (zie Figuur 15). Met behulp van deze tabel beschrijf ik mijn overwegingen bij het maken van de optie om het veld Type te disabelen<sup>11</sup> dan wel te enabelen en de opties voor het toevoegen en verwijderen van een activiteit.

<b>Naam:</b>	<b>Werkplek:</b>	<b>Type:</b>	<b>Begin:</b>	<b>Eind:</b>		
Activiteit1	Kantoor	Werk	08:30	12:30	<insert>	<delete>
Activiteit2	Kantine	Werk	12:30	14:45	<insert>	<delete>

*Figuur 15: Tabel in dienstenscherf*

#### *Disabelen of enabelen*

Bij het kiezen van een werkplek moest het veld Type afhankelijk van de keuze in het veld Werkplek gedisabled dan wel ge-enabled worden. Het disabelen dan wel het enabelen van het veld Type kon aan de serverkant of aan de clientkant (browser) gedaan worden. Ik heb ervoor gekozen om het aan de clientkant te doen met behulp van JavaScript, omdat er geen ingewikkelde controles nodig waren. Ik moest er alleen voor zorgen dat bij het openen van de pagina alle benodigde gegevens meegestuurd werden. Het voordeel is dat er zodoende geen communicatie met de server hoeft plaats te vinden om een veld te disabelen, wat de snelheid ten goede komt. Voor het toevoegen en verwijderen van een activiteit heb ik gekozen dit aan de serverkant te doen, omdat in de source-code van HARMONY al allerlei controles plaatsvonden, die ik anders in JavaScript had moeten maken. Het gevolg zou zijn dat er veel tijd besteedt moest worden aan het ontwikkelen van functies die al bestonden, alleen geschreven in een andere taal. Dit is niet verstandig met het oog op de onderhoudbaarheid van de source-code.

#### *Toevoegen of verwijderen*

Het toevoegen en verwijderen van activiteiten in HARMONY Client wordt gedaan door een bepaalde activiteit te selecteren en dan op een button *Toevoegen* of button *Verwijderen* te klikken. Met HTML en JavaScript is het niet mogelijk om dit op dezelfde manier te doen. Dit was echter geen beperking, want om op een willekeurige plaats een activiteit toe te voegen of te verwijderen, heb ik ervoor gekozen achter elke activiteit een icoontje te tonen. Hierdoor kan met één klik een activiteit verwijderd of toegevoegd worden. Het eerste icoontje toont een a4-tje met de hinttekst<sup>12</sup> *Activiteit*

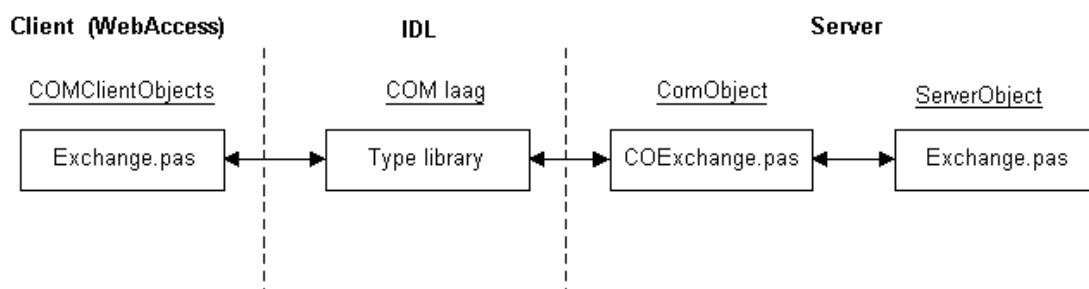
<sup>11</sup> Disabelen is het niet toegankelijk maken. Enabelen is iets toegankelijk maken.

<sup>12</sup> Een hinttekst is tekst die getoond wordt zodra de gebruiker met de cursor over het icoontje gaat.

toevoegen en het tweede icoontje toont een kruisje met de hinttekst *Activiteit verwijderen*. In de tabel wordt het nog aangegeven met <insert> en <delete>. Dit heb ik gedaan om de plaats van de icoontjes aan te geven. In plaats van icoontjes had ik ook kunnen kiezen voor het tonen van links of buttons, maar ik vond icoontjes beter staan. Bovendien kwam dit ten goede van de gebruikersinterface.

#### *Overzicht van mutaties en gezien*

Een mutatie kan betrekking hebben op twee medewerkers, omdat een dienst van de ene medewerker bij een andere medewerker geplaatst kan worden. Beiden moeten de mogelijkheid hebben om aan te geven dat ze de mutatie hebben gezien. De wijzigingen worden per mutatie bijgehouden, dus in de database. In de tabel Exchanges (waarin de mutaties worden opgeslagen) zijn twee velden toegevoegd (AcceptFrom en AcceptTo). Om deze velden uit kunnen lezen en te wijzigen moesten er aanpassingen gedaan worden in HARMONY Server. Figuur 16 toont waar de aanpassingen gedaan moesten worden. Onder figuur 16 wordt uitgelegd welke wijzigingen aangebracht zijn.



Figuur 16: Overzicht client-server architectuur HARMONY

#### *Client*

Aan de client-kant moest per veld een get- en een set-functie gemaakt worden. De get-functie is bedoeld om de waarde het veld AcceptTo of AcceptFrom uit de database te halen en de set-functie om de waarde van het veld AcceptTo of AcceptFrom weg te schrijven in de database.

#### *IDL-file*

IDL staat voor Interface Definition Language en zorgt ervoor dat de client en server met elkaar kunnen communiceren. In de IDL-file staan gegevens over de te gebruiken interface. De IDL kan de gegevens koppelen aan de juiste interface. Hierdoor is het mogelijk dat de Client en Server objecten in verschillende talen geprogrammeerd worden.

#### *Server*

Aan de serverkant zijn twee soorten objecten, namelijk ComObject en ServerObject. In het ComObject wordt de informatie van de IDL ontvangen en geschikt gemaakt voor het ServerObject. Het ServerObject communiceert met de database. Afhankelijk van de aanroep wordt er data uit de database gelezen of wijzigingen in de database aangebracht.

#### *Database*

In de tabel waarin de Exchanges worden opgeslagen zijn er twee velden toegevoegd. De toegevoegde datavelden zijn van het type boolean, omdat de vraag of de medewerker een mutatie heeft beantwoord kan worden met

Ja of Nee. In de database wordt de waarde J of N opgeslagen. Standaard is de waarde N (default-waarde).

Achter elke mutatie wordt een checkbox getoond. Bij deze keuze heb ik dezelfde afwegingen gemaakt als bij de keuze voor icoontjes bij *Uren verantwoorden*. De keuze voor het disabelen in plaats van weglaten heb ik gedaan, zodat de gebruiker kan zien dat zijn actie goed is doorgevoerd.

#### *Rooster gezien*

Per medewerker wordt bijgehouden wanneer hij voor het laatst zijn rooster heeft gezien. Ik heb daarom gekozen voor het opslaan van een datum, omdat je wilt weten wanneer de medewerker het rooster het laatst heeft gezien. Aangezien deze functie niet aanwezig is in HARMONY, moest er net als bij *Mutaties gezien* aanpassingen gedaan worden in de database en aan de serverkant (HARMONY Server).

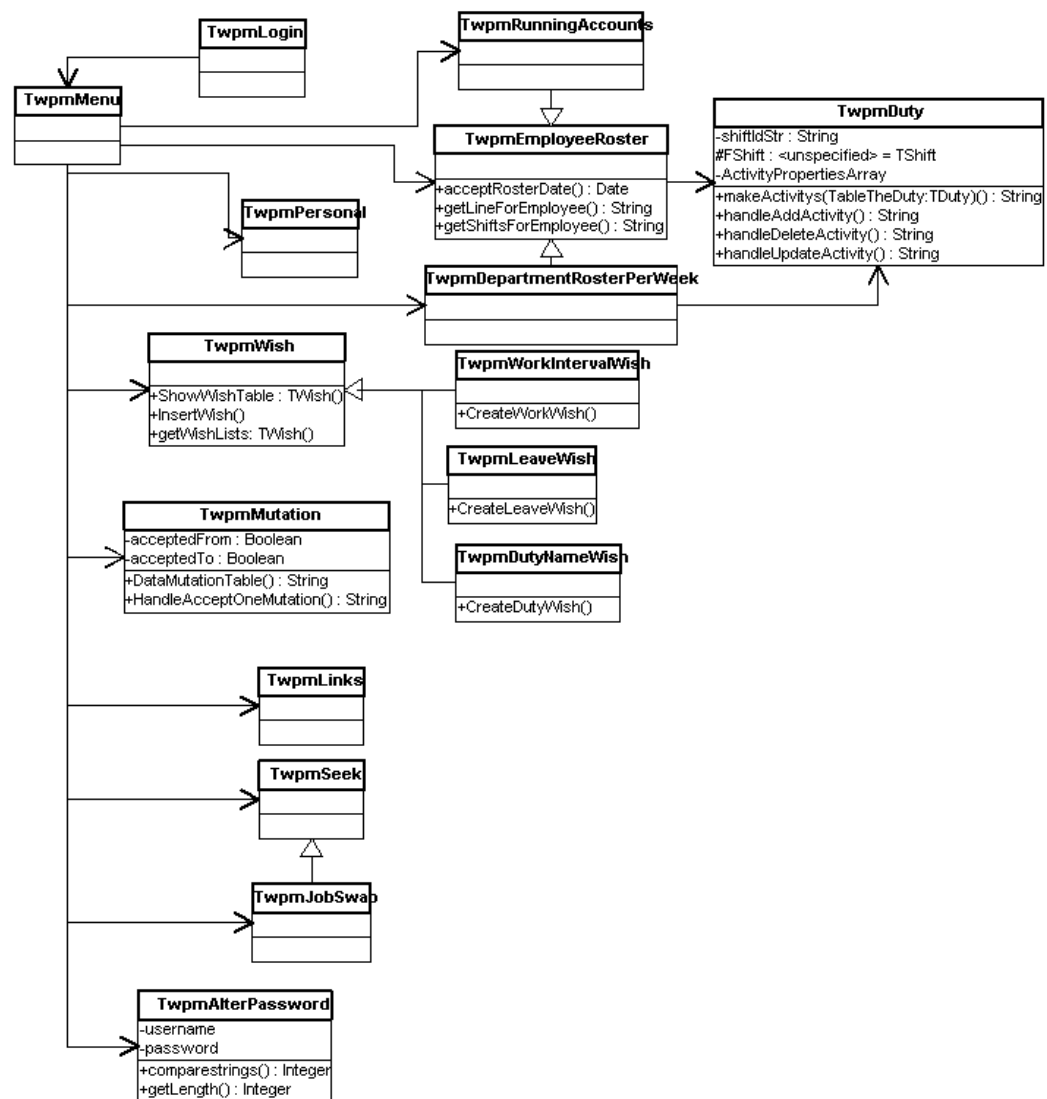
#### *Overzicht wensen*

Het overzicht van wensen is bedoeld om de medewerker te laten zien welke wensen hij heeft openstaan en of een door hem ingevulde wens goed is ingevoerd. Het is dus van belang dat standaard een overzicht wordt getoond van de wensen die betrekking hebben op de toekomst. Daarom heb ik ervoor gekozen om standaard de wensen vanaf de huidige dag of later te tonen. Met behulp van het selecteren van een datum, kunnen wensen uit het verleden bekeken worden.

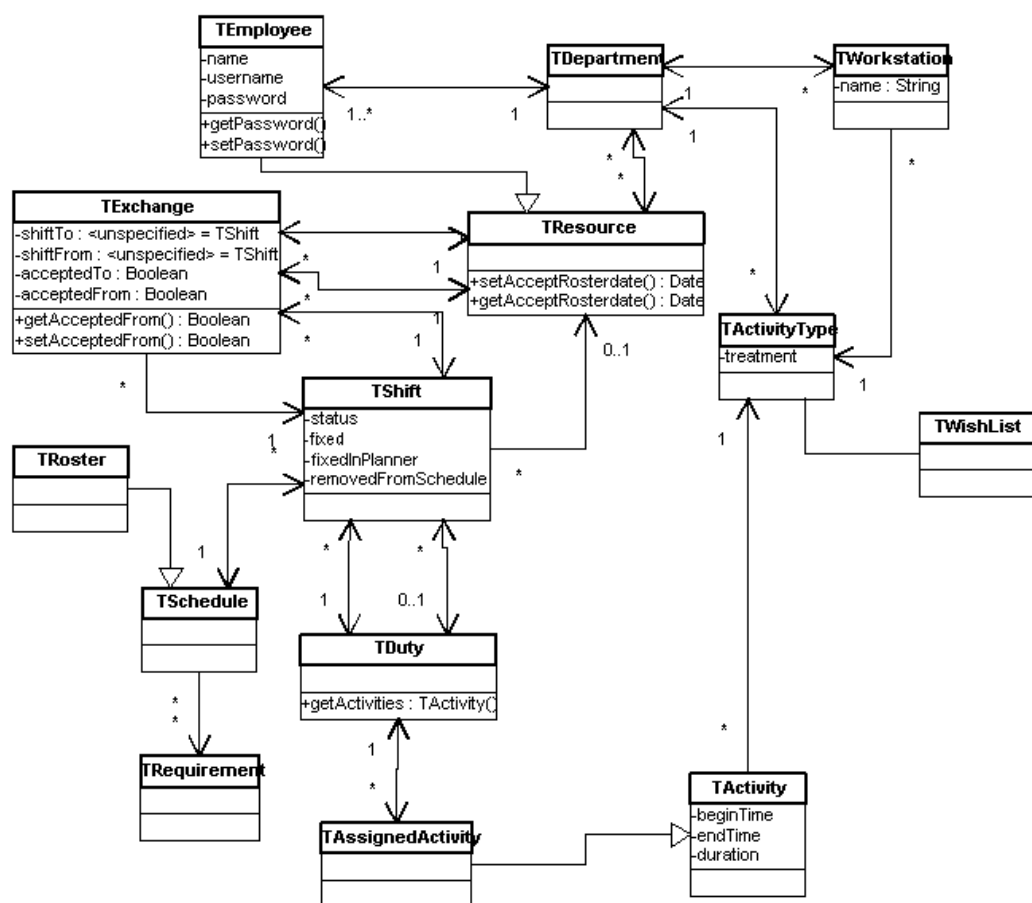
## **8.2 Class diagram**

Deze paragraaf werkt de class diagrams zoals die in het Functioneel Ontwerp zijn opgenomen, verder uit. De onderstaande class diagrams bevatten alleen de details die voor mijn opdracht van belang waren. Verdere uitwerking van de class diagrams is terug te vinden in Externe bijlage VI. Figuur 17 toont de classes van HARMONY WebAccess en Figuur 18 toont de classes van HARMONY.





Figuur 17: Class diagram HARMONY WebAccess



Figuur 18: Class diagram HARMONY

## 9. Bouw

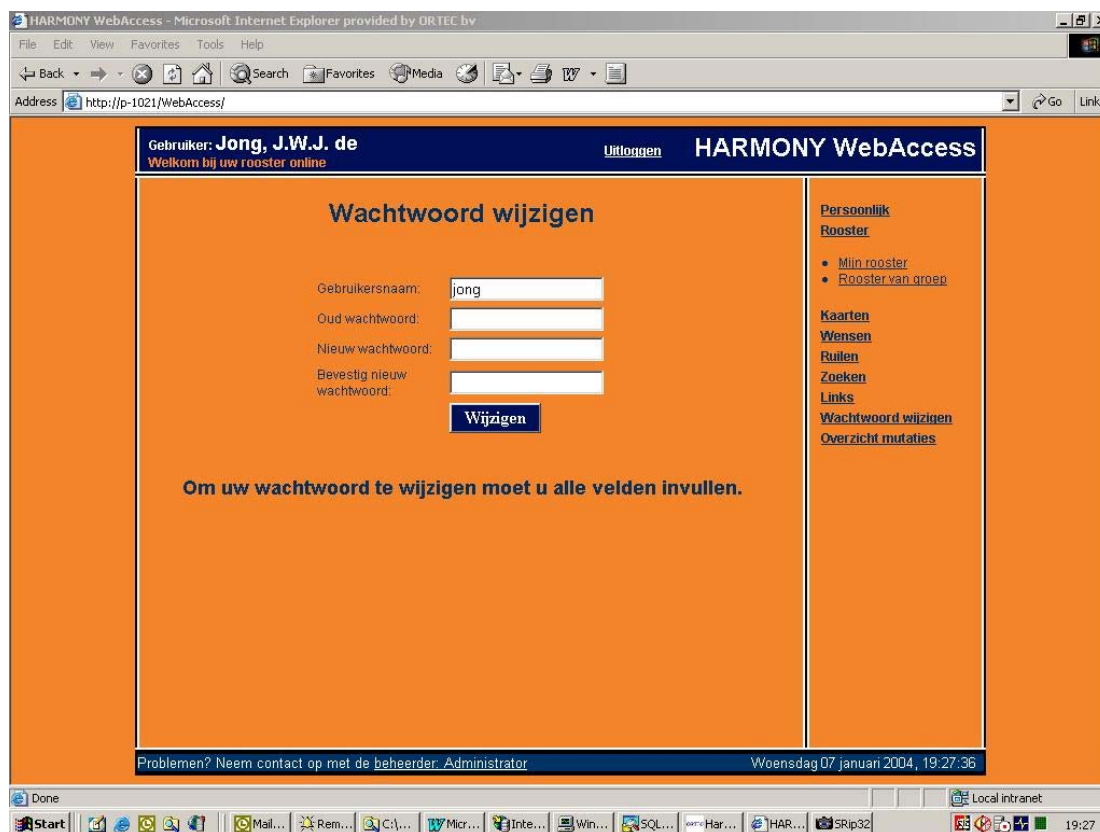
In dit hoofdstuk wordt een uitleg gegeven over de gang van zaken tijdens de bouw van de gewenste functionaliteit. In paragraaf 9.1 wordt toegelicht hoe de bouw heeft plaats gevonden. De ontwikkelde functies hebben een systeemtest ondergaan, hoe dit in zijn werk is gegaan en wat de resultaten waren, staat beschreven in paragraaf 9.2. In paragraaf 9.3 wordt beschreven wat ik in de Implementatiefase heb gedaan.

### 9.1 Bouw van de functionaliteit

Deze paragraaf beschrijft hoe de bouw van de gewenste functionaliteit is gegaan en van welk hulpmiddel ik gebruik gemaakt heb. Van de gewenste functionaliteit wordt aangegeven wat mij goed afging, waar ik wat meer moeite mee had en het resultaat van de functie.

#### *Wachtwoord wijzigen*

De functie Wachtwoord wijzigen was mijn eerste echte kennismaking met Delphi en WebSnap. In de Definitiefase had ik wel al naar de source-code van HARMONY WebAccess gekeken, maar nog niet geprogrammeerd. Aangezien de functie een eigen pagina kreeg in HARMONY WebAccess, moest er een pagina toegevoegd worden. Omdat nergens goed beschreven stond hoe dit gedaan moest worden en waar rekening mee gehouden moest worden, heb ik dit samen met dhr. P. van 't Hoff gedaan. De kennis die ik hiermee heb opgedaan, heb ik gebruikt bij het maken van de overige nieuwe functies. Verder ging het maken van deze functie gemakkelijk, het was een kwestie van de waarden uit de invoervelden op te vragen en deze waarden te controleren. Als het wachtwoord goed door alle controles heen komt dan kan het wachtwoord worden opgeslagen. Het opslaan wordt gedaan met behulp van de volgende methode: *SessionEmployee.Password := NewPassw*, waarbij *NewPassw* het nieuwe wachtwoord bevat. Het resultaat wordt getoond in Figuur 19.



Figuur 19: Wachtwoord wijzigen

### Uren verantwoorden

Bij het bouwen van de functie *Uren verantwoorden* heb ik moeite gehad met het ontwikkelen van de mogelijkheid om het pull-down menu "Type" te kunnen disabelen of enabelen. Ook het maken van de mogelijkheid om activiteiten toe te voegen en te verwijderen was lastig. Bij het nagaan van de mogelijkheden om een pull-down menu te disabelen, heb ik gebruik gemaakt van de aanwezige literatuur<sup>13</sup> en van de internetpagina <http://www.w3schools.com><sup>14</sup>. Op deze site vond ik een voorbeeld dat op de volgende pagina wordt getoond, voor het disabelen dan wel enabelen van een pull-down menu met behulp van buttons.

<sup>13</sup> Boek: JavaScript Second Edition

<sup>14</sup> Op deze pagina staat veel informatie over onder andere JavaScript en HTML.

```
<HTML>
<HEAD>
<SCRIPT type="text/javascript">
  function makeDisable()
  {
    var x=document.forms.myForm.mySelect
    x.disabled=true
  }

  function makeEnable()
  {
    var x=document.forms.myForm.mySelect
    x.disabled=false
  }
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="myForm">
  <SELECT NAME="mySelect">
    <OPTION>Apple</OPTION>
    <OPTION>Banana</OPTION>
    <OPTION>Orange</OPTION>
  </SELECT>
  <INPUT TYPE="button" ONCLICK="makeDisable()" VALUE="Disable
List">
  <INPUT TYPE="button" ONCLICK="makeEnable()" VALUE="Enable List">
</FORM>
</BODY>
</HTML>
```

Het disabelen kon dus door de waarde disabled van het pull-down menu op false dan wel true te zetten. Vervolgens heb ik onderzocht hoe vanuit het ene pull-down menu een andere pull-down menu kon disabelen. Verder moest ik voor elkaar krijgen dat ook het pull-down menu ook de juiste activiteit toonde. Dit heb ik opgelost door een JavaScript-functie aan te roepen, zodra er een andere werkplek wordt geselecteerd. In deze aanroep geef ik het id van de werkplek mee met de daarbijbehorende id van "Type". De JavaScript controleert vervolgens of het id van werkplek leeg is. Als dit zo is, is er geen werkplek geselecteerd en heeft de medewerker de mogelijkheid om een type te kiezen. Als er wel een id is meegegeven wordt met behulp van het id van type het pull-down menu "Type" gedisabled en wordt het juiste type geselecteerd.

In het Functioneel Ontwerp had ik beschreven dat een dienst pas opgeslagen wordt als op de *Opslaan* button wordt geklikt. In het geval van het toevoegen of verwijderen van een activiteit was het moeilijk om dit zo te ontwikkelen. Ik had er namelijk voor gekozen om de functies te gebruiken die in de source-code van HARMONY ontwikkeld waren. Dit betekende dus dat de activiteit toevoegen of verwijderen van een activiteit een request naar de server stuurde. Op de server wordt de actie meteen doorgevoerd in de database, waarna de gebruiker de meest recente gegevens van de dienst terug krijgt. Een voordeel van deze manier van afhandelen was dat de begin- en eindtijden van de activiteiten worden gewijzigd.

Figuur 20 toont het resultaat van de functie *Uren verantwoorden* wat de gebruikers vanuit zijn browser ziet.

**Dienst/Activiteit:**

D

Begintijd:	09:00	Werktijd:	07:00
Eindtijd:	17:00	Arbeidstijd:	07:00
Dag:	donderdag	Datum:	18-12-2003
Status:	Gepubliceerd		

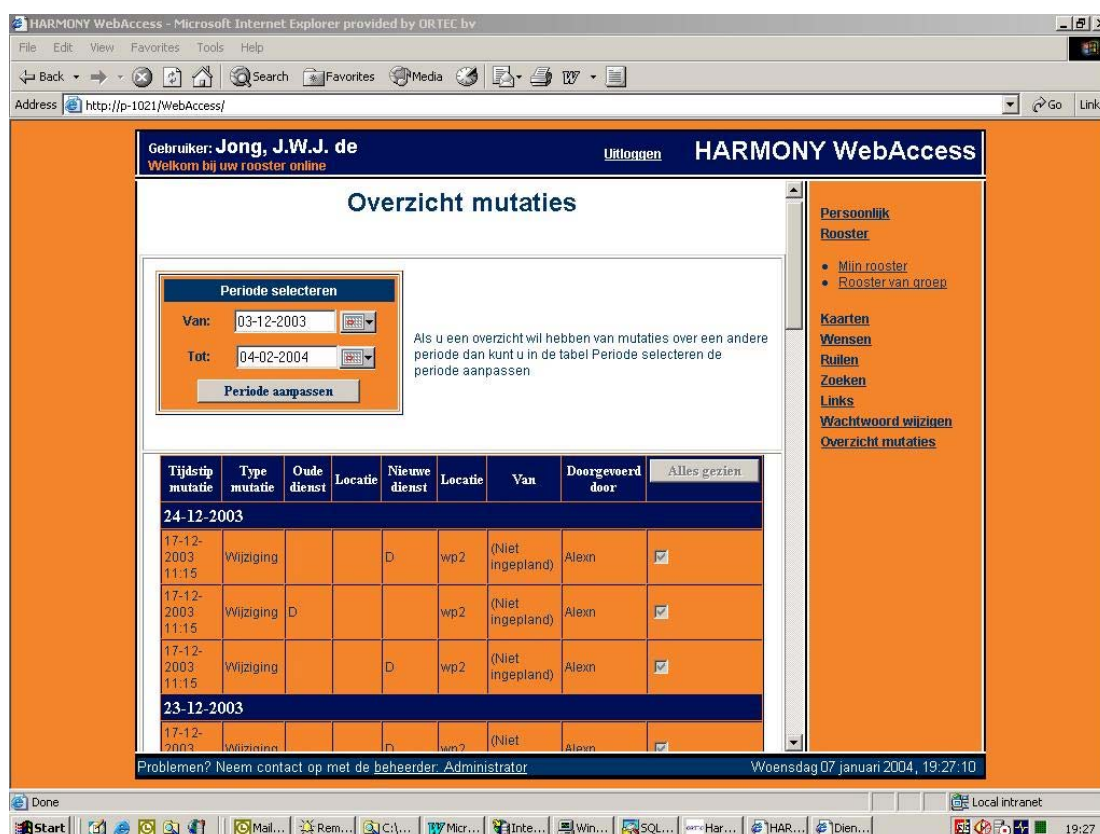
Naam	Werkplek	Type	Begin	Eind		
werken	wp2	Werk	09:00	12:00		
Lunch	Geen	Pauze	12:00	13:00		
werken	wp2	Werk	13:00	17:00		

Figuur 20: Overzicht dienstenscherm

#### Overzicht mutaties en gezien

Bij het ontwikkelen van de functie *Overzicht mutaties en gezien* moest per mutatie een checkbox aangevinkt worden of door een klik op een button allemaal tegelijk. Voor het disabelen kon ik gebruik maken van de kennis die ik in het eerste increment had opgedaan, maar het disabelen van checkboxen werkt anders dan bij de pull-down menu's. Ik heb dus eerst onderzocht wat de mogelijkheden waren van het disabelen van een checkbox. Bij een pull-down menu wordt op naam gedisabled. Bij checkboxen kan met de naam van de checkbox alle checkboxen gedisabled worden. Het disabelen van één checkbox kan gedaan worden door een waarde (value) mee te geven. Deze waarde wordt meegegeven bij het

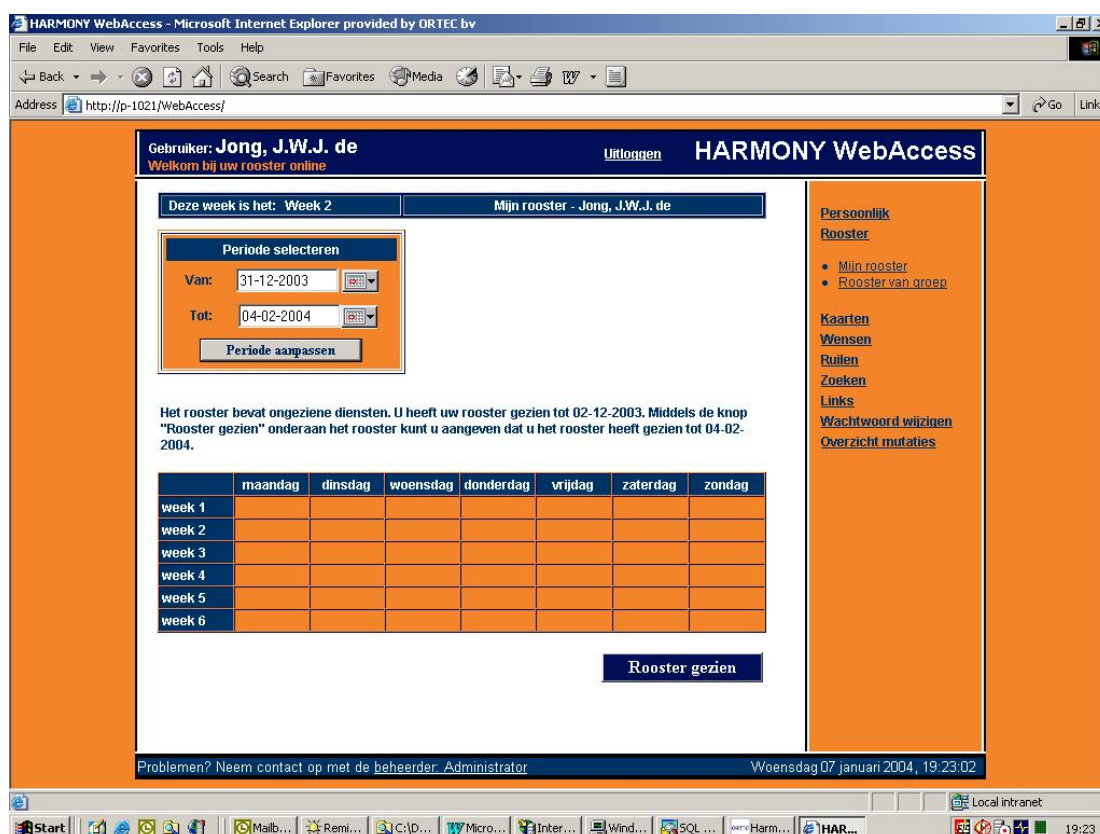
openen van de pagina. Als een bepaalde checkbox wordt aangeklikt, wordt de waarde van dit veld naar de server gestuurd. Op basis van de meegestuurde waarde is aan de serverkant bekend welke checkbox is aangeklikt, dus die kan gedisable worden. Figuur 21 toont het scherm *Overzicht mutaties*.



Figuur 21: Overzicht mutaties

### Functie Rooster gezien

Door op een button te klikken kan de medewerker aangeven dat hij zijn rooster tot en met de huidige dag heeft gezien. De datum van de huidige dag wordt opgeslagen in de database. Bij het bouwen van de functie *Overzicht mutaties en gezien* had ik kennis opgedaan met het aanmaken van de juiste properties en functie-aanroepen aan de client en server kant. Het resultaat van de functie *Rooster gezien* was als volgt.



Figuur 22: Rooster gezien

### Overzicht wensen

Deze functie heb ik niet afgekregen, omdat er een hogere prioriteit werd gesteld op het installeren van HARMONY WebAccess op de laptop van een collega van de Sales-afdeling. Bij deze installatie heb ik een soort van implementatiefase uitgevoerd. Hoe ik dit heb gedaan staat beschreven in paragraaf 9.3.

Voor het controleren van de JavaScript-code heb ik gebruik gemaakt van Microsoft Script Debugger (MSD). Met MSD is het mogelijk om JavaScript-code te debuggen. MSD is een tool van Microsoft en is onderdeel van het Windows 2000 besturingssysteem. De MSD wordt geopend, zodra in de browser een JavaScript-functie wordt aangeroepen waarin een fout zit. Zonder deze debugger toont de browser vaak alleen een niets zeggende melding, zoals "Object doesn't exist". De ontwikkelaar mag dan zelf uitzoeken waar de fout zit en wat er fout is. Met MSD wordt in de JavaScript-code de regel getoond waar de fout zich bevindt. Dit maakt het vinden van de fout al makkelijker. In vergelijking met het debuggen in Delphi, waarin precies de plaats van de fout wordt getoond met daarbij een melding van datgene wat fout is, is deze manier nog verre van ideaal, maar in ieder geval beter dan niets. De keuze voor MSD was vanzelfsprekend, omdat een zoektocht op internet geen debuggers toonde die scripts konden debuggen.

## 9.2 Systeemtest



Deze paragraaf geeft een beeld hoe de systeemtests zijn uitgevoerd en wat de resultaten van de tests waren. Na elke increment heb ik de ontwikkelde functie(s) laten testen door een collega, zoals dat in de standaard van ARTM staat beschreven. Deze tests heb ik laten uitvoeren door Mevr. M. Jungman met behulp van de testplannen die ik in elk increment had opgesteld. Ze kon bij het testen vanaf eigen werkplek inloggen op HARMONY WebAccess. Alleen het resultaat van de systeemtest van het eerste increment had een belangrijke wijziging tot gevolg. Tijdens de test bleek dat het niet mogelijk was om een reeds opgeslagen dienst in één keer te herstellen. Hiermee werd bedoeld dat door één actie alle wijzigingen ongedaan gemaakt konden worden en kon door de aangepaste dienst te verwijderen. Van elke dienst waarin wijzigingen worden aangebracht wordt namelijk een kopie van het origineel gemaakt. De kopie wordt bewerkt en het origineel blijft ongewijzigd. Door de kopie te verwijderen, zal bij het opnieuw wijzigen van de dienst een nieuwe kopie gemaakt worden. Aangezien ik bij het onderzoeken van de werking van het uren verantwoord dit niet was tegengekomen, had ik het niet meegenomen in het Functioneel en het Technisch Ontwerp. Hieruit konden de volgende conclusies getrokken worden:

- Mijn onderzoek was niet goed genoeg en / of
- De personen die het FO hebben gerieviewed, hebben het niet goed gedaan.

Ik vond dat deze mogelijkheid een nuttige toevoeging was. Als een medewerker al zijn wijzigingen ongedaan wil maken, hoeft hij alleen maar op de button *Herstellen* (zie Figuur 20) te klikken om de originele dienst terug te krijgen. Dit betekende dat naast de source-code ook het Functioneel en Technisch Ontwerp aangepast moesten worden.

### **9.3 Implementatiefase**

Deze paragraaf beschrijft de implementatie van HARMONY WebAccess die ik heb uitgevoerd op de laptop van een collega die demo's van HARMONY verzorgt. In het Plan van Aanpak had ik aangegeven geen daadwerkelijke implementatie te gaan uitvoeren. Een collega van de Sales-afdeling wilde HARMONY WebAccess op zijn laptop hebben, zodat hij dit tijdens demo's kon laten zien. Hierbij heb ik een keuze moeten maken tussen of het afronden van de laatste functie *Overzicht wensen* of het installeren van HARMONY WebAccess. Om deze beslissing te nemen heb ik met dhr. P. van 't Hoff overlegd. Hierbij heb ik aangegeven dat het ontwikkelen van de laatste functie een kwestie was ongeveer twee dagen. Deze tijd zou ik niet meer hebben als ik de installatie zou gaan uitvoeren, vanwege de tijd die ik nodig had om me voor te bereiden op de installatie en het uitvoeren van de installatie. De redenen waarom we voor de installatie hebben gekozen, was omdat de functies met de hoogste prioriteit al afgerond waren en het maken van een oplevering met installatie extra kennis zou opleveren. Tijdens deze installatie heb ik geleerd dat het belangrijk is om te weten of het te installeren systeem in een zelfde omgeving komt te draaien als tijdens de ontwikkeling of niet. In dit geval bleek dat niet zo te zijn. Ik heb tijdens de ontwikkeling gebruik gemaakt van een ORACLE database en op de laptop werd gebruik gemaakt van SQL Server. Aangezien er gebruik gemaakt werd van een BDE die platform onafhankelijk werkt, moest dit geen probleem zijn. Dit bleek echter wel een probleem te zijn. SQL Server

en ORACLE hebben niet precies dezelfde manier voor het opslaan van gegevens. Een verschil zit onder andere in de datum velden. Hier kan het systeem op vastlopen. Ook was het verschil met de versies te groot: ik ontwikkelde in een oude versie en op de laptop stond de nieuwste versie van HARMONY. In de nieuwste versie waren een aantal wijzigingen doorgevoerd wat betreft de beveiliging van het systeem en daardoor kon HARMONY WebAccess niet communiceren met HARMONY Server. Aangezien de collega er maar één dag was, moest het binnen één dag opgelost zijn en dat ging niet lukken. Om toch gebruik te kunnen maken van HARMONY WebAccess heb ik mijn ORACLE database en versie van HARMONY Server overgezet naar de laptop. Voor mijn collega was dit niet de meest ideale oplossing, maar hij kon in ieder geval de applicatie HARMONY WebAccess laten zien.

## 10. Evaluatie afstudeerproject

In dit hoofdstuk wordt het project geëvalueerd. In paragraaf 10.1 worden de op te leveren producten geëvalueerd en in paragraaf 10.2 wordt de gehele procesgang geëvalueerd.

### 10.1 Productevaluatie

In deze paragraaf worden alle op te leveren producten uit het Plan van Aanpak geëvalueerd.

#### 10.1.1 *Tussenproducten*

##### Plan van Aanpak

Het doel van het Plan van Aanpak is het vastleggen van de activiteiten en werkzaamheden die nodig zijn voor de uit te voeren opdracht. Een belangrijk onderdeel van dit product was de planning. Aan de hand van de planning kon ik de voortgang van het project in de gaten houden. Ondanks dat het project niet volledig is afgerond, ben ik tevreden over het Plan van Aanpak.

##### Functioneel Ontwerp

Het Functioneel Ontwerp was een Definitiestudie en Functioneel Ontwerp ineen. De standaard van ARTM heeft geen richtlijnen voor het opstellen van een Definitiestudie. De resultaten van de Definitiefase zijn daarom opgenomen in het Functioneel Ontwerp. Ook zijn de functionele eisen van de gewenste functionaliteit en de lay-out van de schermen vastgelegd. Tijdens het project is gebleken dat het Functioneel Ontwerp goed heeft gefunctioneerd, want ik heb weinig aanpassingen hoeven te doen. Afgezien van het toevoegen van de functie *Herstellen dienst* bij het *Uren verantwoorden*, was het product volledig genoeg voor het ontwikkelen van de gewenste functionaliteit.

##### Technisch Ontwerp

Met het vastleggen van de technische eisen wordt een basis gelegd voor het daadwerkelijk ontwikkelen van de functie. Voor mij was het document goed genoeg voor het ontwikkelen. Bij het ontwikkelen van de JavaScript-functies moest ik vaak nog onderzoeken hoe de precieze invulling moest worden, maar die tijd was ik ook kwijt geweest bij het maken van het Technisch Ontwerp. Voor mij was het belangrijk dat bekend was wat voor gegevens er getoond dan wel aangepast moesten worden en of er gebruik gemaakt moest worden van Delphi-code (serverkant) of van JavaScript-code. Om overzicht te hebben van de classes die ik nodig had, heb ik naast het class diagram van HARMONY WebAccess ook een class diagram van HARMONY gemaakt. Met deze class diagrams had ik een goed overzicht van de relaties tussen de classes, waardoor snel te zien was welke classes ik nodig had. Bovendien heb ik in het Technisch Ontwerp ook de testcases gemaakt. Dit was verstandig, omdat ik met behulp van de testcases kon controleren of de functie functioneerde, zoals het moest functioneren. Het Technisch Ontwerp was voor mij een goede basis voor het bouwen.

##### Gebruikershandleiding

Dit product heb ik niet hoeven schrijven, omdat bij ARTM mensen in dienst zijn die de taak hebben om handleidingen te schrijven. Mijn taak was om in het Functioneel Ontwerp zo nauwkeurig mogelijk te beschrijven wat de functies kunnen, zodat ze aan de hand hiervan de handleiding kunnen schrijven.

### **10.1.2      *Eindproduct***

Het doel van het project was dat medewerkers de mogelijkheid moeten hebben om:

- Hun urenverantwoording in te kunnen voeren;
- Hun wensen te bekijken;
- Een overzicht van ruilingen te kunnen zien;
- Aan te kunnen geven dat ze hun rooster of mutatie(s) hebben gezien

Van deze mogelijkheden is alleen de mogelijkheid om de wensen te bekijken nog niet gereed. Aangezien deze functie een lage prioriteit had, is dit niet zo heel erg. Daarom kan ik tevreden zijn met het resultaat dat ik heb behaald.

## **10.2      *Procesevaluatie***

In de procesevaluatie wordt nagegaan worden hoe het project doorlopen is en wat de invloed van de gebruikte methode en de planning was.

### **10.2.1      *Standaard van ARTM***

De standaard van ARTM is door ARTM zelf ontwikkeld en was daarom nieuw voor mij. Dit bleek geen probleem vanwege de overeenkomsten van de standaard met DSDM. Een ander voordeel was dat het voor mijn bedrijfsmentoren mogelijk was om mij bij te sturen waar dat nodig was, omdat zij wisten wat ik nodig had voor mijn project. Aangezien ik het project in mijn eentje heb uitgevoerd, heb ik geen gebruik gemaakt van de projectmanagement zaken van de standaard. Verder bestond de standaard uit goede richtlijnen voor het uitvoeren van een project, zoals het indelen van het project in fasen en het incrementeel en iteratief ontwikkelen. Het voordeel van het incrementeel en iteratief ontwikkelen was dat er snel zichtbare resultaten opgeleverd werden. Voor mij had dit een stimulerende werking.

### **10.2.2      *Gebruikte technieken***

#### *Brainstormsessie*

Bij aanvang van het project heb ik een brainstormsessie gehouden om te kunnen bepalen wat de gewenste functionaliteit was. Een probleem was echter dat de gebruikersgroep naar de lay-out van HARMONY WebAccess keek of ging discussieren over de technische mogelijkheden van een gewenste functionaliteit. De belangrijkste oorzaak was volgens mij dat er geen centraal punt was waarop iedereen de genoemde suggesties kon zien. Als zichtbaar is wat de genoemde suggesties zijn, kan op basis van die punten verder gebrainstormd worden. De reden waarom ik dit niet gedaan

heb, was vanwege ruimtegebrek. Ik had een deel van een muur nodig om HARMONY WebAccess te kunnen laten zien en daardoor was er geen ruimte meer voor bijvoorbeeld een flip-over. Voor de volgende keer moet ik dus zorgen voor een grotere ruimte of een andere manier van presenteren, bijvoorbeeld door screen-dumps te maken. In deze gevallen is er wel ruimte voor een centraal punt waar de suggesties getoond kunnen worden.

#### *Delphi 5 en 6*

In dit project is het me meegevallen hoe snel ik met Delphi aan de slag kon. Alhoewel Delphi veel op de programmeertaal Pascal leek, was ik er niet echt overtuigd dat ik er zo snel mee zou kunnen werken. Dit kwam omdat ik drie jaar geleden voor het laatst in Pascal had geprogrammeerd en deze kennis dus minimaal was. Een andere reden was het feit dat Delphi een object georiënteerde programmeertaal is en Pascal een procedurele programmeertaal (imperatief). Imperatief houdt in dat het programma opgebouwd is uit commando's. Dit bleek geen probleem, omdat het declareren van variabelen en het maken van functies in beide talen op dezelfde manier plaatsvindt.

#### *JavaScript*

De programmeertaal JavaScript was voor mij niet onbekend, omdat ik daar veel mee gewerkt had tijdens een vorige stageperiode. In dit project is het voor mij echter wel duidelijk geworden dat er een groot verschil is tussen een programmeertaal kennen en het beheersen ervan. Bovendien is het niet altijd even gemakkelijk om tegelijkertijd met twee verschillende programmeertalen te werken. Wat betreft de Delphi source-code was het niet zo groot probleem omdat de compiler deze fouten wel eruit haalde, maar de Microsoft Script Debugger (MSD) die ik voor mijn JavaScript gebruikte gaf niet precies aan waar de fouten zaten, waardoor ik onnodig tijd verloor met het zoeken van de fouten.

### **10.2.3 Planning**

De planning, zoals ik die had opgesteld bij aanvang van het project heb ik tot en met het eerste increment kunnen gebruiken. Het eerste increment ging zelfs sneller dan ik had gedacht. Het gevolg was dat ik 5 dagen eerder kon beginnen aan het tweede increment. Dit kwam omdat ik niet alle tijd nodig had die ingepland was voor het Testen, Gebruikershandleiding schrijven en het Opleveren van een functionaliteit. Het testen deed ik zelf namelijk voortdurend tijdens het ontwikkelen en de systeemtest heb ik door een collega laten doen en die kon dat vanaf haar eigen werkplek doen. Het schrijven van de handleiding hoefde ik niet te doen, omdat dat de taak van andere mensen is binnen ARTM en het opleveren betekende dat ik alle source-code wat bij de desbetreffende functionaliteit hoorde, opgeslagen moest hebben in het versiebeheersysteem MKS. Door deze meevallers kon ik de duur van de incrementen met een paar dagen inkorten. Hierdoor had ik ruim een week extra tijd. Deze tijd heb ik gebruikt voor het schrijven van mijn afstudeerverslag en voor het aanpassen van de lay-out van HARMONY WebAccess, ten gevolge van de veranderde huisstijl van ORTEC. In Externe bijlage III wordt de laatste versie van de planning getoond.

## 11. Aanbevelingen voor de toekomst

Om HARMONY WebAccess in gebruik te nemen moeten er nog een aantal essentiële zaken afgehandeld worden. In dit hoofdstuk wil ik aan de hand van een aantal punten toelichten wat ter verbetering gedaan kan of moet worden. De punten die ik belangrijk vind, zijn:

- De mogelijkheid om HARMONY WebAccess op een andere server te installeren dan waar HARMONY Server op draait;
- De beveiliging van de HTTP-data;
- De uitbreiding van de functionaliteit;
- Het vervangen van HARMONY Client door HARMONY WebAccess.

### *De mogelijkheid om HARMONY WebAccess op een andere server te installeren dan waar HARMONY Server op draait*

Momenteel kan HARMONY WebAccess alleen op dezelfde computer draaien als waar HARMONY Server op staat. Zoals in paragraaf 6.3.2 staat beschreven heeft dit probleem met security-settings te maken. Het is van belang dit probleem op te lossen voordat HARMONY WebAccess daadwerkelijk in gebruik genomen wordt door de klant. Met het oog op onderhoudbaarheid is het namelijk beter om de webserver te scheiden van de applicatieserver. Bovendien hebben klanten vaak al een webserver draaien en is het niet logisch om daar HARMONY Server daarbij te installeren.

### *De beveiliging van de HTTP-data*

Aan de beveiliging van de HTTP-data is nog niets gedaan. Aangezien HARMONY een bedrijfskritische applicatie is, is het belangrijk dat dit goed is afgeschermd. Om de HTTP-data te beveiligen kan gebruik gemaakt worden van SSL<sup>15</sup> (Secure Socket Layer). HARMONY WebAccess draait op Microsoft Information Services (IIS) en binnen IIS kan gebruik gemaakt worden van het beveiligingsprotocol SSL. SSL is een beveiligingsprotocol waarmee persoonlijke gegevens of creditcardgegevens veilig via het Internet kunnen worden verstuurd terwijl de privacy gegarandeerd blijft. SSL biedt gegevensprivacy, gegevensverificatie en briefgeheim.

### *De uitbreiding van de functionaliteit*

In de aanleiding van mijn afstudeeropdracht in paragraaf 2.1 wordt al aangegeven dat de uitbreiding van HARMONY WebAccess wenselijk is met het oog op de concurrentiepositie en de wens van de klanten. Het kan wenselijk zijn om de medewerkers meer mogelijkheden te geven. HARMONY WebAccess zou ook uitgebreid kunnen door ook functionaliteit voor de planner te ontwikkelen. Hiermee kom ik bij mijn laatste punt, het vervangen van HARMONY Client door HARMONY WebAccess.

### *Het vervangen van HARMONY Client door HARMONY WebAccess*

Als zowel de medewerkers als de planners al hun zaken via HARMONY WebAccess kunnen regelen dan is de HARMONY Client feitelijk overbodig. Dit zou de meest ideale situatie zijn, omdat:

- Er geen clients meer geïnstalleerd en ingericht moeten worden (denk aan registry-settings en het installeren van clients op de computers);

---

<sup>15</sup> Informatie op: <http://www.microsoft.com/windows2000/server/evaluation/features/web.asp>

- Het een flinke kostenbesparing oplevert voor de klant, want in plaats van allemaal HARMONY Clients hoeft alleen HARMONY WebAccess geïnstalleerd te worden. De gebruikers hebben vervolgens alleen een browser nodig om met HARMONY te kunnen werken;
- Het systeem makkelijker onderhouden kan worden, omdat het systeem op één is geïnstalleerd.

## Literatuurlijst

### Boeken:

Eilers, H.B. (1997), *Reader SO-04: Kopieën uit het boek: Systeemontwikkeling volgens SDM*, Den Haag: Haagse Hogeschool

HoneyCutt, J., Morgan, M., Reynolds, M. C. & Woolridge, A. (1997), *Special Edition Using JavaScript*, Indianapolis: Que Corporation

Jong, J. de (2002), *Ontwikkeling van webbased toegang tot HARMONY*, Gouda: ORTEC Arbeidstijdmanagement bv

Kleppe, A. & Warmer, J. (1999), *Praktisch UML, objectgeoriënteerde systeemontwikkeling*, Amsterdam: Addison Wesley Longman

ORTEC ARTM (2003), *Gebruikershandleiding HARMONY*, Gouda: ORTEC Arbeidstijdmanagement bv

Stapleton, J. (2002), *DSDM, De methode in de praktijk*, Schoonhoven: Academic Service

### Internetpagina's:

W3Schools:  
[www.w3schools.com](http://www.w3schools.com)

Informatie over (D)COM:  
<http://www.sei.cmu.edu/str/descriptions/com.html>

Informatie over (D)COM:  
[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndcom/html/msdn\\_dcomarch.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndcom/html/msdn_dcomarch.asp)

Informatie over WebSnap:  
<http://www.lemanix.com/lemanix/lemanixisapi.dll/WebSnap>

SDGN = Software Developers Group Netherlands:  
<http://www.sdgn.nl>

Website van Borland:  
<http://community.borland.com>

Website met informatie over Delphi:  
<http://www.nldelphi.com>

Informatie over SSL:  
<http://www.microsoft.com/windows2000/server/evaluation/features/web.asp>



## Woordenlijst

API	= Applications Programming Interface. Een interface die bestaat uit een set van instructies of software-routines, die de gebruiker/programmeur de mogelijkheid geeft om applicaties te bouwen voor bepaalde software of hardware
ARTM	= Arbeidstijdmanagement
BDE	= Borland Database Engine
DCOM	= Distributed Component Model
Debuggen	= Controleren van source-code op fouten
DLL	= Dynamic Link Library
IDL	= Interface Definition Language
IIS	= Internet Information Services (onderdeel van Windows)
ISAPI	= Internet Server API
Multi-threading	= Het opdelen van een applicatie in verschillende onafhankelijk uitvoerbare onderdelen waarbij deze onderdelen ook tegelijk kunnen uitgevoerd worden.
UML	= Unified Modeling Language

## Figurenlijst

Figuur 1: Planning .....	10
Figuur 2: Ontwikkel-schema Bouwfase.....	12
Figuur 3: Organigram ORTEC-groep .....	22
Figuur 4: Organigram ORTEC Arbeidstijdmanagement bv.....	22
Figuur 5: Overzicht roosterproces .....	24
Figuur 6: Planbord .....	26
Figuur 7: Architectuur HARMONY .....	27
Figuur 8: Oude Menustructuur HARMONY WebAccess .....	29
Figuur 9: Architectuur HARMONY met HARMONY WebAccess .....	32
Figuur 10: Use-case Realiseren dienst .....	37
Figuur 11: Use-case diagram .....	38
Figuur 12: Nieuwe menustructuur HARMONY WebAccess .....	39
Figuur 13: Class diagram HARMONY WebAccess .....	41
Figuur 14: Class diagram HARMONY.....	42
Figuur 15: Tabel in dienstenschermb.....	44
Figuur 16: Overzicht client-server architectuur HARMONY .....	45
Figuur 17: Class diagram HARMONY WebAccess .....	47
Figuur 18: Class diagram HARMONY.....	48
Figuur 19: Wachtwoord wijzigen .....	50
Figuur 20: Overzicht dienstenschermb .....	52
Figuur 21: Overzicht mutaties .....	53
Figuur 22: Rooster gezien .....	54

## **Externe bijlagen**

## Externe bijlage I      Opdrachtschrijving

Uitbreiden van functionaliteit van HARMONY-WebAccess bij ORTEC Arbeidstijdmanagement

ORTEC (Operations Research Technology) B.V., internationaal opererend met meer dan 250 werknemers, is gespecialiseerd in het op product- en projectbasis ontwikkelen en leveren van software ter ondersteuning van logistieke en financiële beslissingsprocessen.

Naast het aandachtsgebied arbeidstijdmanagement is ORTEC sterk aanwezig op de gebieden:

- Financiële zaken.
- Asset & Liability Management.
- Revenue Management.
- Logistiek.
- Transport & Distributie.
- Productieplanning Procesindustrie.

Kenmerkend voor alle systemen van ORTEC is dat zij een of meerdere optimalisatiemethoden bevatten die planningen genereren (advanced planning). In dit opzicht is ORTEC uniek in Nederland en internationaal toonaangevend. ORTEC is door middel van nauwe banden met de internationale academische wereld direct betrokken bij innovatief onderzoek op dit terrein. Diverse van haar medewerkers zijn parttime werkzaam als hoogleraar.

De dienstverlening van ORTEC bestaat sinds haar oprichting in 1981 uit een combinatie van het geven van advies op het gebied van planning processen en het leveren van advanced planning software.

De afdeling Arbeidstijdmanagement (inzetplanning personeel) is met circa 50 medewerkers een substantieel onderdeel van de activiteiten van ORTEC. Deze afdeling is verdeeld over een tweetal vestigingen. Een vestiging in Groningen en een vestiging in Gouda.

Arbeidstijdmanagement is ontstaan nadat ORTEC HRM (de vestiging in Gouda) een van de concurrenten, namelijk IKS Producten (vestiging Groningen), in het begin van 2001 had overgenomen. De twee vestigingen zijn blijven bestaan maar dragen nu de naam Arbeidstijdmanagement.

Arbeidstijdmanagement voert verschillende soorten werkzaamheden uit. De ontwikkeling, bouw, implementatie en ondersteuning van het softwarepakket HARMONY omvatten het grootste deel van de inspanningen van de afdeling. Daarnaast houdt Arbeidstijdmanagement zich ook nog bezig met consultancy opdrachten op het gebied van

arbeidstijdmanagement. Voorbeelden hiervan zijn marktspecifieke ATW- en CAO-trainingen.

HARMONY is een roosterapplicatie waarmee organisaties hun medewerkers kunnen inplannen. Dit kan door middel van een roostergenerator waar de applicatie op basis van criteria een rooster kan samenstellen. Er zijn verschillende criteria die kunnen worden ingevuld. Voorbeelden hiervan zijn verlof, ziekte, ATW (arbeidstijdenwet), CAO-regelgeving en persoonskwalificaties.

Daarnaast kan de gebruiker ook handmatig een planning maken. Daarbij wordt in het planbord, het scherm waarin de planning gemaakt wordt, aangegeven wat de criteria zijn waaraan de planning zou moeten voldoen.

HARMONY is een client/server-applicatie die is ontwikkeld in Borland Delphi. Er is een begin gemaakt met een web-client (HARMONY-WebAccess), zodat de server via het Internet is te benaderen. HARMONY-WebAccess is echter nog niet gelijkwaardig aan de gewone client. De planningsapplicatie kan zijn gegevens opslaan in verschillende soorten databases waaronder ORACLE, SQL Server en Interbase. Het opslaan van gegevens vindt plaats via de HARMONY-server. Deze server regelt al het verkeer tussen de data en de verschillende clients. De server zorgt er op deze manier voor dat een wijziging in een scherm bij de ene medewerker, direct wordt doorgevoerd bij de andere medewerkers. Dit heeft als voordeel dat een gebruiker altijd de meest actuele informatie tot zijn beschikking heeft.

Met de productlijn HARMONY bedient Arbeidstijdmanagement en diversiteit aan klanten met hoogwaardige dienstroostersoftware.

HARMONY-WebAccess zoals die er nu is bevat slechts enkele triviale basisfunctionaliteiten van de gewone client. Dit betekent dat niet alle functionaliteiten, zoals die in de gewone client zitten beschikbaar zijn via het Internet. De gegevens die nu beschikbaar zijn via het Internet zijn over het algemeen passief. Dit wil zeggen dat een persoon wel kan zien voor wanneer hij ingeroosterd is, maar niets kan wijzigen. Zo kan hij bijvoorbeeld niet ingeven wanneer hij vakantie wil. Dit moet hij via de gewone client of via de planner doorgeven. Dit wordt gezien als een belemmering, omdat het de planner ontlast als personen zelf in kunnen vullen wanneer ze beschikbaar zijn. Een taak van de planner is nu dat hij medewerkers een planning toestuurdt of moet doorbellen, hier gaat ook veel tijd in zitten als er veel medewerkers zijn.

Het doel van de afstudeeropdracht is het ontwikkelen van uitbreidingen op HARMONY-WebAccess, zodat medewerkers in HARMONY-WebAccess hun urenverantwoording kunnen invoeren (geplande activiteiten kunnen aanpassen of verwijderen en activiteiten kunnen toevoegen), bekijken van hun wensen, overzicht van mutaties kunnen bekijken en aan kunnen geven dat ze hun rooster of ruiling(en) hebben gezien.

De volgende software zal gebruikt worden:

- Borland Delphi 6.
- Microsoft Internet Information Server.
- Microsoft Office 97.
- ORACLE.
- MKS Integrity.

De volgende rapporten zijn beschikbaar:

- Gebruikershandleiding HARMONY (-WebAccess).
- Technische systeemdokumentatie.
- Afstudeerverslag van dhr J. de Jong.

In het kader van de afstudeeropdracht zullen de volgende activiteiten verricht worden:

Plan van aanpak opstellen.

Kick-off:

Onderzoek werking programmeertaal Delphi.

Inwerken in Harmony-WebAccess source.

Onderzoek naar gewenste functionaliteiten.

Opstellen lijst van de gewenste functionaliteiten.

Functioneel ontwerp:

Beschrijven van de processen "Uren verantwoorden", "Overzicht mutaties + bevestigen gezien", "Bevestigen rooster gezien" en "Wensen bekijken" met behulp van use-cases.

Ontwerp van schermen.

Beschrijven techniek die nodig is om de te bouwen functies te kunnen gebruiken en documentatie, zoals help-files. Review en aanpassen van het FO.

Technisch ontwerp per increment:

Uitwerken procesbeschrijving "Uren verantwoorden" met behulp van class diagram.

Review en aanpassen TO.

Realisatie per increment:

Programmeren van functionaliteiten zoals beschreven in het TO.

Testplan schrijven.

Testen van functionaliteiten en reviewen (source controleren).

Gebruikershandleiding schrijven.

Oplevering functionaliteiten.

Getest systeem operationeel maken.

De ontwikkeling is onderverdeeld in de volgende incrementen:

1e increment: proces "Uren verantwoorden".

2e increment: proces "Overzicht mutaties + bevestigen mutatie(s) gezien".

3e increment: proces "Bevestigen rooster gezien" en proces "Wensen bekijken".

Bij de uitvoering van de opdracht zal de standaard zoals bij Arbeidstijdmanagement wordt gebruikt gehanteerd worden. De volgende technieken zullen gebruikt worden:

- Brainstormsessie.
- UML (use-cases, class diagram).

De volgende producten zullen opgeleverd worden:

- Plan van aanpak.
- Functioneel ontwerp.
- Technisch ontwerpen.
- HARMONY-WebAccess uitgebreid met functionaliteiten "Uren verantwoorden", "Overzicht mutaties + bevestigen mutatie(s) gezien", "Bevestigen rooster gezien" en "Wensen bekijken".
- Testplan.

Gebruikershandleiding.

## **Externe bijlage II    Whitepaper HARMONY**



ORTEC bv  
Postbus 490  
2800 AL Gouda  
  
tel. 0182 540 500  
  
[www.ortec.nl](http://www.ortec.nl)

**1. Inhoud**

**INLEIDING..... 74**

**ORGANISATIE..... 75**

ROOSTERGROEP .....	75
INDELING IN ROULEERGROEPEN OF TEAMS .....	75
EN VERDER.....	75
<b>MEDEWERKERS .....</b>	<b>77</b>
PERSOONSGEGEVENS .....	77
INDELING IN ROOSTERGROEPEN .....	77
FUNCTIE EN KWALIFICATIES.....	77
SAMENWERKING .....	77
INDIVIDUELE WENSEN .....	78
EN VERDER.....	78
<b>ROOSTERBEHEER.....</b>	<b>79</b>
DIENTEN EN DIENSTOPBOUW .....	79
BEZETTINGSEISEN.....	79
VERTICALE NORMEN.....	79
STATUS .....	79
EN VERDER.....	80
<b>ROOSTERMETHODIEKEN .....</b>	<b>81</b>
FLEXIBEL ROOSTER.....	81
WENSENROOSTERS.....	81
DIENTSTREEKSEN.....	81
CYCLISCH BASISROOSTER.....	81
PLOEGENROOSTER.....	81
<b>INTELLIGENTE PLANBORDEN.....</b>	<b>82</b>
INZICHT & OVERZICHT .....	82
TELLINGEN .....	83
ACTIEVE ONDERSTEUNING .....	83
EEN ANDERE KIJK OP DE WERELD.....	84
EEN GROTER DETAILNIVEAU .....	84
EN VERDER.....	84
<b>WERKPLEKPLANNING .....</b>	<b>86</b>
ONDER- EN OVERBEZETTING.....	86
EN VERDER.....	86
<b>PLANAUTOMATEN .....</b>	<b>87</b>
DIENTSTROOSTER GENEREREN .....	87
ROOSTERVOORSTEL .....	87
GENERATOR WERKPLEKPLANNING .....	87
<b>WET- EN REGELGEVING .....</b>	<b>88</b>
ARBEIDSTIJDENWET, ARBEIDSTIJDENBESLUIT EN CAO .....	88
PER MEDEWERKER INSTELBAAR .....	88
EN VERDER.....	88
<b>SCENARIO ANALYSE .....</b>	<b>89</b>
MEERDERE ROOSTERS VOOR DEZELFDE PERIODE.....	89
<b>TIJDREGISTRATIE &amp; URENVERANTWOORDING .....</b>	<b>90</b>
KLOKTIJDEN.....	90
EN VERDER.....	90

<b>VOORBEREIDING SALARISVERWERKING.....</b>	<b>91</b>
SALDOKAARTEN.....	91
VERLONING OP BASIS VAN DE REGELS UIT DE CAO .....	91
DIRECT TOEGANKELIJK .....	91
CORRECTIES DOORVOEREN.....	91
EN VERDER.....	92
<b>TOEGANG VIA INTERNET: HARMONY-ONLINE.....</b>	<b>93</b>
DIENSTROOSTER .....	93
RUILINGEN .....	93
EN VERDER.. .....	93
<b>RAPPORTAGES &amp; OVERZICHTEN .....</b>	<b>94</b>
MANAGEMENTINFORMATIE.....	94
RAPPORTGENERATOR.....	94
EN VERDER.....	94
<b>INTEGRATIE: INTERFACING &amp; KOPPELING .....</b>	<b>95</b>
PERSONEELSINFORMATIESYSTEEM .....	95
PRIKKLOK & TIJDREGISTRATIESYSTEEM.....	95
SALARISSYSTEEM.....	95
EN VERDER.....	95
<b>TOEGANGSBEVEILIGING.....</b>	<b>96</b>
<b>CONFIGURATIEMOGELIJKHEDEN .....</b>	<b>97</b>
<b>TECHNISCHE KARAKTERISTIEKEN .....</b>	<b>98</b>

## 2. Inleiding

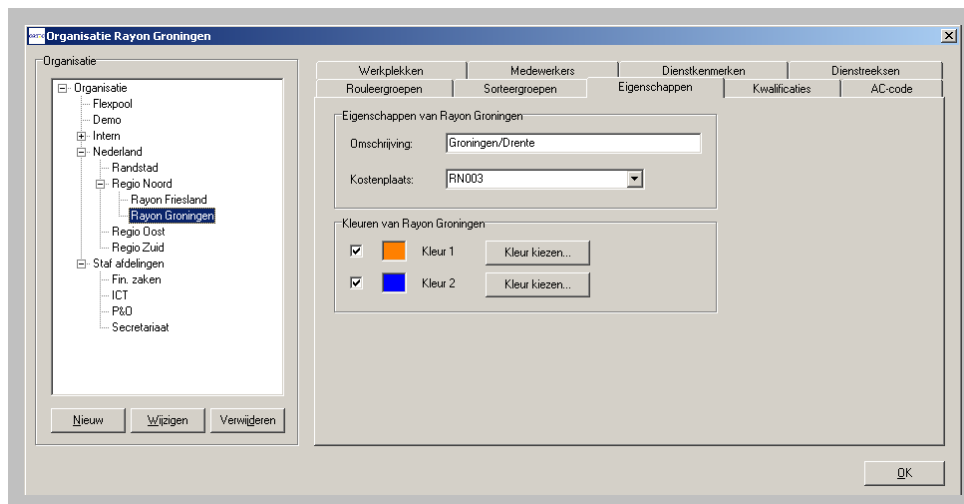
Personeel is zowel zeer kostbaar als ook heel waardevol. Een goede inzetplanning van uw personeel is dan ook niet alleen gericht op efficiëntie en effectiviteit, maar ook op arbeidsmotivatie en ergonomie.

ORTEC HARMONY biedt de ondersteuning die nodig is om aan deze doelstellingen tegemoet te komen. Door overzichtelijke planborden, door *realtime* controles op regelgeving (uit de Arbeidstijdenwet, het Arbeidstijdenbesluit en de CAO), door rekening te houden met ergonomische criteria en individuele wensen, door handige tellingen en controles op vastgestelde normen en door geavanceerde planautomaten. HARMONY zorgt dat u snel kunt inspelen op ad hoc verstoringen in de planning, bijvoorbeeld als gevolg van ziekte of onvoorziene werkzaamheden, en voor een nauwkeurige registratie en verwerking van gewerkte uren, ziekte en verlof.

In de volgende hoofdstukken presenteren wij u de belangrijkste functionaliteit van HARMONY.

### 3. Organisatie

Een organisatie is een min of meer logische indeling van mensen en werk. In HARMONY is het mogelijk de volledige organisatiestructuur op te nemen. Het aantal niveaus is hierbij onbeperkt.



#### 3.1 Roostergroep

Het laagste niveau in de organisatie wordt gevormd door een groep medewerkers die gezamenlijk verantwoordelijk is voor een hoeveelheid uit te voeren taken, activiteiten of diensten. In HARMONY heet dit een roostergroep. De planner maakt een rooster per roostergroep op basis van de bezettingseisen die gelden voor die groep.

##### 3.1.1 Dynamisch

In de praktijk blijkt dat de indeling van mensen en werk niet statisch is. Allereerst kunnen medewerkers in meerdere groepen werkzaam zijn. Daarnaast kan het natuurlijk altijd gebeuren dat de ene afdeling mensen tekort heeft, terwijl er elders mensen over zijn. In deze situaties biedt HARMONY de in- en uitleenfunctionaliteit, waarmee medewerkers tijdelijk in een andere roostergroep worden geplaatst. Ook het uitbesteden van een dienst aan een andere roostergroep is mogelijk. Deze functionaliteit is met name praktisch voor organisaties die veel gebruik maken van poolkrachten of (interne) uitzendbureaus.

#### 3.2 Indeling in rouleergroepen of teams

Binnen een roostergroep is een nadere indeling in rouleergroepen en teams mogelijk. Een rouleergroep is een groep medewerkers die gezamenlijk een cyclisch rooster hebben. Teams zijn noodzakelijk voor het opstellen van ploegenroosters. Beide indelingen zijn ook te gebruiken om de medewerkers in het planbord te ordenen.

#### 3.3 En verder...

- Elke roostergroep heeft als kenmerk een kleur. In het dienstrooster van een **flexpool** is daarmee in één oogopslag te zien waar de poolkrachten werkzaam zijn.

*De organisatiestructuur mag geen belemmering zijn bij het efficiënt benutten van beschikbare capaciteit.*

*HARMONY staat het toe dat medewerkers tegelijkertijd in meerdere roostergroepen (of afdelingen) werken. Controles en tellingen lopen hierbij door over beide roosters. Daarnaast ondersteunt HARMONY het in- en uitleenen van medewerkers en diensten.*

- Aan elke roostergroep is een **kostenplaats** gekoppeld. Standaard geldt voor alle diensten binnen de roostergroep dat de kosten op deze kostenplaats drukken. Per dienst of activiteit is hier van af te wijken.
- Per roostergroep is het mogelijk extra **kenmerken** voor de diensten te definiëren. Op deze manier is het bijvoorbeeld mogelijk aan te geven dat bijzondere kleding is vereist voor de dienst. In het planbord kunnen tellingen over deze kenmerken gedefinieerd worden.
- **Functies** en **kwalificaties** zijn op elk niveau van de organisatie te definiëren. Op deze manier is bijvoorbeeld de kwalificatie 'bedrijfshulpverlening' organisatiebreed te gebruiken, terwijl de kwalificatie 'teamleider' per afdeling een andere inhoud kan hebben.

## 4. Medewerkers

### 4.1 Persoonsgegevens

Over het algemeen zijn de persoonsgegevens van een medewerker vastgelegd in een personeelsinformatiesysteem. HARMONY neemt deze gegevens, voor zover zij relevant zijn voor de planning, over, zodat invoer en beheer op één plaats binnen de organisatie geconcentreerd zijn en blijven.

*dikwijls dat een planner al weet dat hij een nieuwe medewerker in zijn rooster kan opnemen, terwijl de relevante gegevens nog niet (door de afdeling personeelszaken) in het personeelsinformatiesysteem zijn verwerkt.*

*HARMONY maakt het mogelijk medewerkers tijdelijk in te voeren en in te plannen en de exacte gegevens over te nemen op het moment dat deze ook in het personeelsinformatiesysteem bekend zijn.*

#### 4.1.1 Zelf kenmerken toevoegen

Binnen HARMONY is het mogelijk zelf extra informatie over medewerkers vast te leggen die specifiek noodzakelijk is voor de planning. Voorbeelden hiervan zijn het telefoonnummer waarop de medewerker bereikbaar is tijdens het weekend of de afstand woon-werkverkeer ten behoeve van de reiskostenvergoeding.

### 4.2 Indeling in roostergroepen

HARMONY kent een flexibele indeling in roostergroepen. Zo kan een medewerker

op elk moment toetreden tot een roostergroep en is hij niet langer inzetbaar na de einddatum die bij de indeling wordt aangegeven. Zodra de medewerker is toegekend aan een groep, is hij zichtbaar in het rooster van die groep. Daarbij toont HARMONY ook de diensten die hij in een andere groep draait op het moment dat hij gelijktijdig in meerdere groepen werkzaam mag zijn.

Incidentele inzet in een andere groep is mogelijk door de medewerker in of uit te lenen.

### 4.3 Functie en kwalificaties

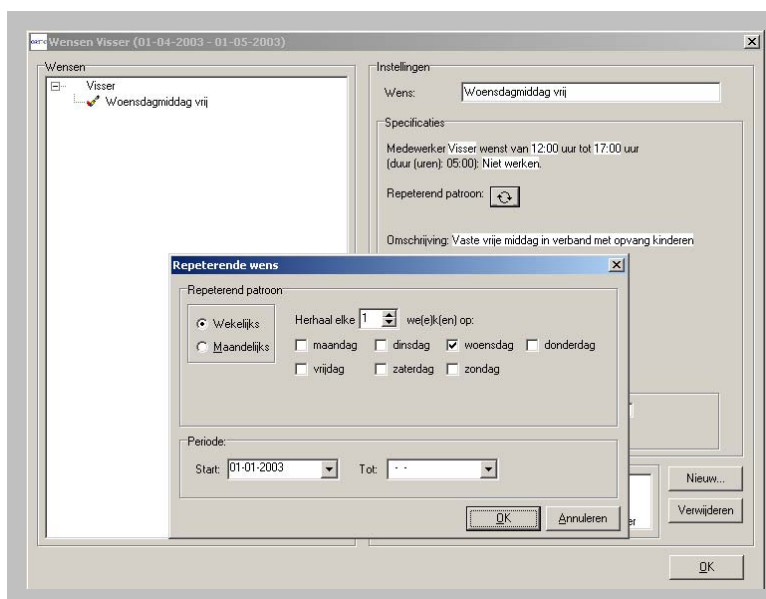
Middels functies en kwalificaties is vast te leggen welke activiteiten of diensten een medewerker mag uitvoeren. Bij het inplannen van een activiteit of dienst bij een medewerker controleert HARMONY of de medewerker bevoegd of geschikt is om de dienst uit te voeren.

#### 4.3.1 Verlopen van kwalificaties

Om bepaalde handelingen te mogen verrichten is het soms noodzakelijk dat de medewerker voldoende uren maakt zodat kennis en routine niet verloren gaan. Andere kwalificaties verlopen op het moment dat een medewerker geen opfrustraining volgt. Zo is het een vereiste voor een piloot om voldoende vlieguren op een bepaald type vliegtuig te maken en moet een politieagent regelmatig schietinstructie volgen. HARMONY signaleert de planner op het moment dat een kwalificatie verloren dreigt te gaan en geeft een schending aan op het moment dat een medewerker wordt ingezet terwijl zijn kwalificatie niet langer geldig is.

### 4.4 Samenwerking

HARMONY ondersteunt verschillende samenwerkingsverbanden. Zo kan een leerling door een vaste of door meerdere begeleiders begeleid worden, waarbij een minimum overlap in diensten is aan te geven. Tijdens het plannen controleert HARMONY of medewerkers waarvoor een samenwerking geldt voldoende met elkaar samenwerken. Ook is het mogelijk een ongewenst samenwerkingsverband in te voeren als geldt dat medewerkers juist niet met elkaar samen kunnen of mogen werken.



## 4.5 Individuele wensen

Vaak hebben medewerkers wensen met betrekking tot hun inzet. Parttimers maken afspraken over vaste vrije dagen, ouders hebben vaste dagen ouderschapverlof, in het rooster moet rekening worden gehouden met zorgtaken en sociale activiteiten buiten de werksfeer.

In HARMONY zijn zowel eenmalige als

terugkerende wensen vast te leggen. Daarnaast is het mogelijk aan te geven dat een medewerker een voorkeur voor een bepaalde dienst heeft.

## 4.6 En verder..

- Binnen een roostergroep kunnen medewerkers werkzaam zijn met ieder verschillende **arbeidsvoorwaarden**. Vandaar dat het in HARMONY mogelijk is per individu aan te geven onder welke CAO hij valt.
- Als een medewerker ergens niet mag werken (bijvoorbeeld omdat hij een **toegangsverbod** heeft) dan kan dit voor een locatie (bijvoorbeeld een heel gebouw) of een specifieke werkplek (alleen de portiersloge) worden vastgelegd. Het planbord toont een schending als de medewerker toch op een dergelijke locatie werkzaam is.
- Met behulp van een **voorkeurslocatie** is het mogelijk aan te geven waar een medewerker bij voorkeur werkzaam is en voor welk percentage van zijn werktijd.
- Via HARMONY of via HARMONY-Online kan elke medewerker zijn **persoonlijk rooster** raadplegen en afdrukken.
- Een medewerker kan ook zijn eigen **urenverantwoording** invoeren. De planner hoeft deze alleen nog te accorderen.
- Een medewerkers kan ook zelf zijn wensen invoeren via **HARMONY-Online** (zie ook pagina 93). Het enige dat de planner doet, is de wensen accorderen.
- De planner kan een wens voorzien van een **belangrijkheid**, die bepaalt in hoeverre de planautomaat tracht de wens te respecteren.
- Per medewerker kunnen **individuele werkafspraken** worden vastgelegd. Voor de planautomaat gelden deze afspraken als beperkingen bij het maken van het rooster. Op deze manier is aan te geven dat bepaalde medewerkers liever korte of juiste lange series diensten hebben of meer rust nodig hebben na een nachtdienst.



## 5. Roosterbeheer

Het roosterbeheer dient ervoor om op gestructureerde wijze een rooster op te stellen, te publiceren en af te ronden tot aan de salarisverwerking toe. Het scherm roosterbeheer geeft direct inzicht in de statussen van de verschillende roosters en biedt de mogelijkheid om bezettingseisen per periode te definiëren.

### 5.1 Diensten en dienststopbouw

Onderdeel van het roosterbeheer is het onderhouden en beheren van de afzonderlijke diensten. Belangrijk punt hierbij is dat de diensten zijn opgebouwd uit één of meerdere activiteiten, waaronder bijvoorbeeld een pauze. Per activiteit is aan te geven welke kwalificaties noodzakelijk zijn en op welke locatie of werkplek de activiteit uitgevoerd moet worden. HARMONY toont direct de totale werktijd en de totale arbeidstijd van de dienst. Per dienst kan de planner twee kleuren definiëren die naar wens in het planbord zichtbaar zijn.

### 5.2 Bezettingseisen

De bezettingseis (ook wel roostersleutel) geeft aan hoeveel diensten de roostergroep per dag moet opbrengen. Dit aantal hoeft niet altijd gelijk te zijn. Daarom is de bezettingseis voor elke willekeurige periode te definiëren. Op deze manier kan bijvoorbeeld met een zomer- en winterbezetting worden gepland.

*Het aantal medewerkers dat nodig is op een dag kan verschillen per seizoen. De bezettingseisen in HARMONY zijn daarom per periode instelbaar.*

Voor feestdagen bestaat de optie voor een "bezetting als op". Dit is met name handig als bijvoorbeeld op Tweede Paasdag de bezetting als op zondag geldt.

### 5.3 Verticale normen

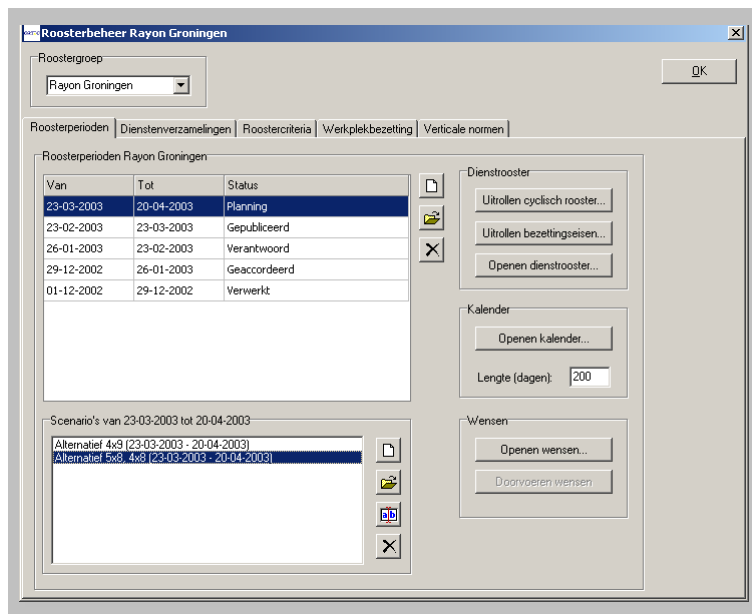
Behalve een bezettingseis in de vorm van diensten, is het mogelijk normen te definiëren die over alle diensten geëvalueerd worden. Een voorbeeld hiervan is de

aanwezigheid van een persoon met een BHV-diploma. In het planbord telt HARMONY het aantal medewerkers dat op enig moment aanwezig is met de gestelde kwalificatie. Onderbezetting of overbezetting is vervolgens zichtbaar als schending.

### 5.4 Status

De status van een rooster bepaalt wie toegang heeft tot het rooster, wie bepaalde aanpassingen mag verrichten en wat de gevolgen van wijzigingen zijn.

Een roosterperiode waarvoor de status 'Planning' geldt, biedt de planner de vrije hand om te puzzelen, zonder dat de wijzigingen die hij aanbrengt enige gevolgen met zich meebrengen. Zodra hij het rooster bekendmaakt aan de medewerkers ('publiceert'), plaatst HARMONY alle ruilingen en wijzigingen in een logboek. Enerzijds zodat de planner weet wat er allemaal veranderd is, zodat hij de betrokkenen op de hoogte kan brengen. Anderzijds vanwege de arbeidsvoorwaardelijke gevolgen die een dergelijke aanpassing kan hebben. Op het moment dat een rooster gerealiseerd is, kunnen medewerkers hun urenverantwoording invoeren. Na het accorderen bereikt het rooster uiteindelijk de status 'Verwerkt' op het moment dat de salarissen op basis van het rooster zijn verwerkt.



## 5.5 En verder..

- Voor bijzondere data is een **afwijkende bezetting** aan te geven. Deze optie gebruikt men bijvoorbeeld in de detailhandel voor koopzondagen en extra koopavonden en bij veerdiensten voor de dagen dat er een grote toeloop van toeristen wordt verwacht.
- Het rooster kan voor **elke willekeurige periode** worden bekeken. Zo is het mogelijk voldoende historie in beeld te hebben om een rooster voor de toekomst op te stellen.
- De **werkplekbezetting** geeft aan hoeveel personen op enig moment op een bepaalde werkplek aanwezig moeten zijn. Deze bezettingseis is de basis van de werkplekplanning. In het planbord werkplekplanning is voor ieder willekeurig tijdinterval te zien of sprake is van over- of onderbezetting.
- Met behulp van een (**kamer**)**openingsurenplan** is in één keer de bezetting in te voeren voor werkplekken van verschillende roostergroepen. Zo volgt bijvoorbeeld voor een operatiekamer uit dit plan de bezettingseis voor zowel de chirurgie als de operatieassistenten en anesthesie.
- Het plannen van verlof, cursussen en iedere andere vorm van afwezigheid die ver van tevoren bekend is, gebeurt met behulp van de **kalender**. Deze is beschikbaar vanuit het roosterbeheer. Alvorens een nieuw rooster wordt gemaakt, worden de gegevens uit de kalender in het rooster overgenomen.
- De kalender biedt ook de mogelijkheid een **verlof-/vakantieplanning** te maken, waarbij een minimaal benodigde bezetting gewaarborgd blijft.

## 6. Roostermethodieken

De wijze van plannen verschilt niet alleen per organisatie, maar vaak zelfs per afdeling. Soms wordt de keuze voor een bepaalde methodiek bepaald door de bedrijfscultuur, meestal is deze werkgerelateerd. Maar hoe u ook plant: HARMONY ondersteunt uw werkwijze.

*De invoering van een advanced planning systeem moet er niet toe leiden dat u verplicht bent uw werkwijze aan te passen als daar op zich geen aanleiding toe is.*

### 6.1 Flexibel rooster

Een flexibel rooster wordt per roosterperiode opgesteld op basis van de bezettingseisen. Hierbij houdt de planner met name rekening met de afwezigheid van medewerkers in verband met verlof of ziekte. Daarnaast let hij uiteraard op de restricties die de Arbeidstijdenwet en de CAO hem opleggen. Voor zover mogelijk komt de planner tegemoet aan de wensen van de medewerkers. HARMONY ondersteunt deze wijze van roosteren, zowel 'handmatig' als middels de planautomaat.

### 6.2 Wensenroosters

In geval van een wensenrooster geven de medewerkers zelf aan welke dienst zij op een bepaalde dag willen hebben. Ook voor dit type rooster biedt HARMONY diverse vormen van ondersteuning.

### 6.3 Dienstreeksen

Met behulp van dienstreeksen is het mogelijk vooraf reeksen van diensten op te geven die de medewerkers als prettig ervaren. Een voorbeeld van een dergelijke reeks is het werken van het volledige weekend, inclusief de vrijdag ervoor en de maandag erna. Binnen HARMONY zijn de dienstreeksen zowel op groepsniveau als individueel te definiëren. De planautomaat houdt zoveel als mogelijk rekening met de gewenste dienstreeksen. Bovendien zijn volledige reeksen middels een enkele code in het rooster in te voeren.

### 6.4 Cyclisch basisrooster

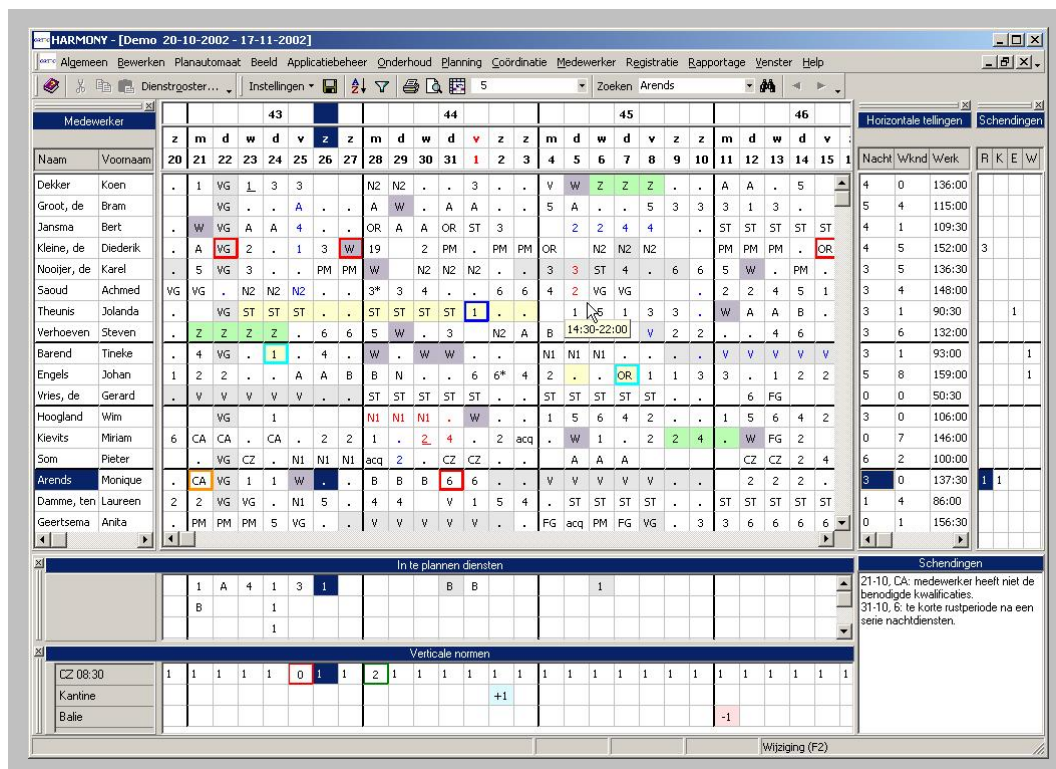
Cyclische roosters kunnen zowel voor een groep gelden als individueel bepaald zijn. HARMONY controleert hierbij of het rooster (op het moment dat het zou worden uitgerold) voldoet aan de Arbeidstijdenwet en het Arbeidstijdenbesluit. Door meerdere medewerkers aan dezelfde regel van het rooster te koppelen is het mogelijk duo-banen in het basisrooster op te nemen. Doordat de basisroosters voor alle rouleergroepen binnen de roostergroep onder elkaar getoond worden, is het mogelijk de bezettingseisen met meerdere groepen af te dekken.

### 6.5 Ploegenrooster

Het ploegenrooster maakt het mogelijk om cyclische roosters in te voeren voor teams. De teams bestaan uit medewerkers met verschillende functies. Bij het 'uitrollen' van het rooster kent HARMONY de juiste diensten toe aan de medewerker met de overeenkomstige functie. Indien noodzakelijk is een verdere specificatie van de activiteiten van de diverse medewerkers op de dag mogelijk middels de module werkplekplanning. Het scherm waarin het ploegenrooster wordt gedefinieerd biedt dezelfde functionaliteit als het dienstroosterplanbord. Hierdoor is al tijdens het opstellen van het rooster duidelijk of het rooster voldoet aan de geldende regels uit de Arbeidstijdenwet en het Arbeidstijdenbesluit.

## 7. Intelligentie Planborden

Het planbord is het belangrijkste instrument van de planner. Met behulp van dit scherm is hij dagelijks in de weer om het rooster voor de toekomst op te stellen of het actuele rooster aan te passen aan (onverwachte) gebeurtenissen. De belangrijkste handelingen kan de planner direct vanuit het planbord uitvoeren.



### 7.1 Inzicht & Overzicht

De belangrijkste toegevoegde waarde van een planbord is het bieden van inzicht en overzicht. De inhoud van deze begrippen verschilt echter en is afhankelijk van de situatie of de gebruiker. Binnen de planborden van HARMONY bepaalt de gebruiker grotendeels zelf hoe hij de informatie gepresenteerd wil hebben.

#### 7.1.1 Persoonlijke beeldinstellingen

De persoonlijke beeldinstellingen maken het mogelijk het planbord aan te passen aan de voorkeuren van de planner of aan de situatie waarin het scherm gebruikt wordt.

Zo bepaalt de gebruiker onder meer:

- welke persoonsgegevens van de medewerkers hij in het planbord wil zien,
- hoe hij de dienst gepresenteerd wil hebben (tijden of codes, kleuren),
- welke tellingen hij wil tonen (zowel horizontaal als verticaal), waarbij hij snel kan switchen tussen verschillende verzamelingen tellingen,

#### 7.1.2 Filters & sortering

Door het instellen van filters is het mogelijk in te zoomen op een deelverzameling van de medewerkers. Om de volgorde van de getoonde medewerkers te bepalen, biedt HARMONY een scala aan sorteermogelijkheden. Zo kan de planner een eigen sortering instellen, is hij vrij om de medewerkers in sortegroepen te verdelen en kan hij

sorteren op de waarde van een telling, zodat bijvoorbeeld de medewerkers met de minste overuren bovenaan komen te staan.

### **7.1.3 Markeren op basis van telling**

Door te markeren op basis van een telling is het mogelijk snel te zien welke diensten een bijdrage leveren aan een telling. Zo *highlight* HARMONY na het markeren op basis van het saldo overuren alle diensten die overuren bevatten.

*Niet iedere gebruiker vindt dezelfde informatie even belangrijk. Ook het moment waarop een planning wordt geraadpleegd kan bepalend zijn voor de relevantie van bijvoorbeeld de getoonde tellingen.*

## **7.2 Tellingen**

Zowel de horizontale tellingen (over de diensten per persoon) als de verticale tellingen (over de diensten per dag) zijn per gebruiker in te stellen. In het planbord kan snel gewisseld worden tussen verschillende verzamelingen tellingen.

Doordat de saldokaarten (zie pagina 91) ook als telling kunnen worden getoond, zijn arbeidsvoorwaardelijke gevolgen van de planning (bijvoorbeeld het saldo van de verlofkaart, de onregelmatigheidstoelage of het aantal keren woon-werkverkeer) direct zichtbaar te maken.

## **7.3 Actieve ondersteuning**

HARMONY biedt actieve ondersteuning bij het opstellen en beheren van het dienstrooster.

### **7.3.1 Directe terugkoppeling en controles**

Alle planborden binnen HARMONY bevatten online controles met betrekking tot de regels uit de Arbeidstijdenwet, het Arbeidstijdenbesluit, de CAO en andere (collectieve) werktijdenregelingen. Zo is direct zichtbaar of een planningshandeling leidt tot een schending ten aanzien van de regels.

*Een volledig rooster achteraf controleren maakt het vrijwel onmogelijk eenmaal ontstane schendingen te corrigeren.*

*Een melding bij iedere schending leidt tot 'wegklikgedrag', waardoor juist de belangrijke meldingen worden gemist.*

*HARMONY toont schendingen direct en blijvend in het planbord.*

Ook de tellingen in het planbord zijn continu up-to-date. Zelfs als een collega via een ander werkstation een verandering in het rooster aanbrengt.

### **7.3.2 Vervangingsvoorstel**

Als het rooster (bijna) af is, is het lastig om te reageren op ad hoc verstoringen. Het roostervoorstel geeft verschillende alternatieven om een opgevallende dienst in te plannen. Bijvoorbeeld om een

vervanger te vinden voor een zieke werknemer. Ook op het moment dat er een schending ontstaat in het dienstrooster biedt de optie roostervoorstel uitkomst.

## 7.4 Een andere kijk op de wereld

Over het algemeen heeft het de voorkeur om vanuit de medewerker naar het dienstrooster te kijken. Dit met name omdat de ingewikkelde regels uit

de Arbeidstijdenwet betrekking hebben op het rooster van een individuele medewerker. Zo af en toe is het echter handig om de inzet per dienst te zien. Bijvoorbeeld om te controleren of de teambezetting op orde is of te zien of er voldoende continuïteit in de bezetting is. De 'dienstenview' toont het rooster vanuit een andere invalshoek.

## 7.5 Een groter detailniveau

Voor bepaalde organisaties is het noodzakelijk op een hoger detailniveau te plannen. Binnen HARMONY kan de planner vanuit het dienstroosterplanbord direct inzoomen op de werkplekplanning. Dit planbord is qua opbouw identiek, maar toont in een *ganttchart*-representatie op welke werkplekken de medewerkers ingezet zijn en hoe dit zich verhoudt ten opzichte van de bezettingseisen die voor deze werkplekken gelden. (zie ook pagina 86)

## 7.6 En verder...

- **Ziekmeldingen** zijn direct in het planbord te verwerken. HARMONY plant de oorspronkelijke dienst(en) van de zieke medewerker uit, waarna de planner middels de optie 'roostervoorstel' een vervanger kan zoeken.
- Vanaf het moment dat een rooster de status 'Gepubliceerd' heeft, logt HARMONY alle **wijzigingen** die in het rooster worden aangebracht. Niet alleen om te registreren welke medewerkers eventueel geïnformeerd moeten worden, maar ook omdat een (late) wijziging arbeidsvoorwaardelijke gevolgen kan hebben. Hierbij maakt Harmony onderscheid tussen mutaties door de medewerkers zijn geïnitieerd ('ruilingen') en wijzigingen die voor de organisatie van belang zijn ('wijzigingen').
- Vanuit het planbord kan een medewerker direct worden **ingeleend** van of **uitgeleend** aan een andere roostergroep. Hetzelfde geldt voor diensten. Deze kunnen met één handeling worden doorgegeven aan bijvoorbeeld de roosteraar van de flexpool.

- Als een medewerker een dienst heeft in een **andere roostergroep** is deze ook zichtbaar in het planbord. De tellingen en controles lopen uiteraard gewoon door.
- Vanzelfsprekend is het mogelijk **meerdere diensten op één dag** aan een medewerker toe te kennen.
- Het **dienstenpalet** geeft snel inzicht in alle diensten die binnen de roostergroep gedefinieerd zijn. Indien noodzakelijk zijn afwijkende diensten in te voeren.
- Voor het plannen zijn alle standaard Windows functies beschikbaar, zoals knippen en plakken, **drag&drop**, herhalen van de invoer en multiple select.
- Diensten die niet aan een andere medewerker mogen worden gekoppeld, kunnen worden **vastgezet** (geblokkeerd).
- Bij een individuele dienst kan de planner een **opmerking** invoeren, zoals bijvoorbeeld het telefoonnummer waarop iemand bereikbaar is tijdens een cursus.

The screenshot displays the HARMONY software interface, which is used for project planning and resource management. The main window shows a Gantt chart with a timeline from 08:00 to 23:00. The chart is divided into columns representing 30-minute intervals. The left side of the chart lists resources, and the right side shows the duration of activities for each resource.

**Resources and Activities:**

- Bobbiaan:** No activities shown.
- Boogaard, E. de:** Activity "Werkplek 11" from 08:00 to 17:00.
- Boogerd, E.:** Activity "Werkplek 11" from 17:00 to 23:00.
- Boogman:** No activities shown.
- Dijkgraaf, L. van:** Activity "Werkplek 10" from 08:00 to 17:00.
- Eiburg:** No activities shown.
- Ether, R.:** Activity "Ruimte 3" from 17:00 to 23:00.
- Fredrikson, P.:** Activity "Ruimte 3" from 17:00 to 23:00.
- Gorrer, D.:** Activity "Algemeen" from 08:00 to 17:00.
- Gorter, A.:** Activity "Algemeen" from 08:00 to 17:00.
- Grondslag, S. van:** Activity "10:00 - 17:00" from 10:00 to 17:00.
- Hakkansson, H.:** Activity "10:00 - 17:00" from 10:00 to 17:00.
- Havermout, Y.:** No activities shown.

**Summary Tables:**

**Tellingen (Counts):**

Nacht	Werk	Rv	Atr
0 0	00:00	0 00:	
0 0	09:00	1 09:	
1 0	08:30	0 08:	
0 0	00:00	0 00:	
0 0	07:00	1 07:	
0 0	00:00	0 00:	
1 0	08:30	0 08:	
1 0	08:30	0 08:	
0 0	07:00	1 07:	
0 0	07:00	1 07:	
0 0	00:00	1 00:	
0 0	00:00	1 00:	

**Schendingen (Violations):**

R	K	E	W
1			
1	1		
1	1		
1	1		

**Summary Table:**

Resource	Activity	Start	End	Duration
Algemeen		08:00	17:00	+1
Gorter, A.		08:00	17:00	-1
Gorrer, D.		08:00	17:00	-2
Stevanovic, L.		08:00	17:00	-2
Kantoor		08:00	17:00	+1
Bhagelow, R.		08:00	17:00	-1
Azzis, S.		08:00	17:00	-1
Ruimte 2		08:00	17:00	-1
Ruimte 3		08:00	17:00	-1
Spierg, E.		08:00	17:00	-1
Werkplek 10		08:00	17:00	+1
Ether, R.		08:00	17:00	-1

## 8.1 Onder- en overbezetting

## 8.2 En verder...

- 86



## **9. Planautomaten**

ORTEC heeft een lange historie ten aanzien van de ontwikkeling en implementatie van software die automatisch planningen genereert. Internationaal gezien is ORTEC toonaangevend in het bedenken en toepassen van wiskundige en econometrische modelleringen en methodieken in genererende plansoftware.

### **9.1 Dienstrooster genereren**

De planautomaat die een volledig dienstrooster kan genereren, houdt rekening met de randvoorwaarden die volgen uit de Arbeidstijdenwet, het Arbeidstijdenbesluit en de CAO. Ook de individuele wensen van en werkafspraken gemaakt met een medewerker spelen een rol. De planautomaat zal diensten niet toewijzen aan een medewerker die niet over de vereiste kwalificaties beschikt. Door aan diensten een prioriteit toe te kennen, is mogelijk eerst te roosteren op minimale bezetting en vervolgens op gewenste bezetting.

#### **9.1.1 Ergonomische roostercriteria**

Met behulp van de ergonomische criteria is het resultaat van de planautomaat te sturen. Voorbeelden van criteria zijn de voorkeur voor een voorwaarts roterend rooster, de voorkeur voor het werken van hele weekenden en de voorkeur minimaal 3 en maximaal 6 diensten achter elkaar te werken.

De planner bepaalt zelf het gewicht van de verschillende criteria. De scores ten aanzien van de ingestelde criteria zijn als telling weer te geven in het planbord en kunnen op gedetailleerd niveau gerapporteerd worden, zodat de gebruiker inzicht heeft in de afwegingen die de planautomaat heeft moeten maken.

### **9.2 Roostervoorstel**

Met behulp van het roostervoorstel kan de planner knelpunten in het rooster oplossen. Het roostervoorstel is beschikbaar in het dienstroosterplanbord, de dienstenview en in het planbord voor de werkplekplanning.

### **9.3 Generator werkplekplanning**

De generator voor de werkplekplanning deelt de aanwezige medewerkers in op een werkplek. Als uitgangspunt hierbij gelden de werkplekbezettingseisen, waarbij die werkplekken waarvoor de bezetting een hoge prioriteit heeft eerst worden gepland. De generator zorgt ervoor dat een medewerker alleen ingezet wordt op een werkplek waarvoor hij gekwalificeerd is. De gebruiker kan sturen op afwisseling in activiteiten of continuïteit in activiteiten.

## 10. Wet- en Regelgeving

In Nederland kennen we een stringente regelgeving ten aanzien van de arbeids- en rusttijden. Veel van de regels zijn handmatig nauwelijks te controleren, bijvoorbeeld doordat zij betrekking hebben op een lange periode of ingewikkelde uitzonderingen bevatten.

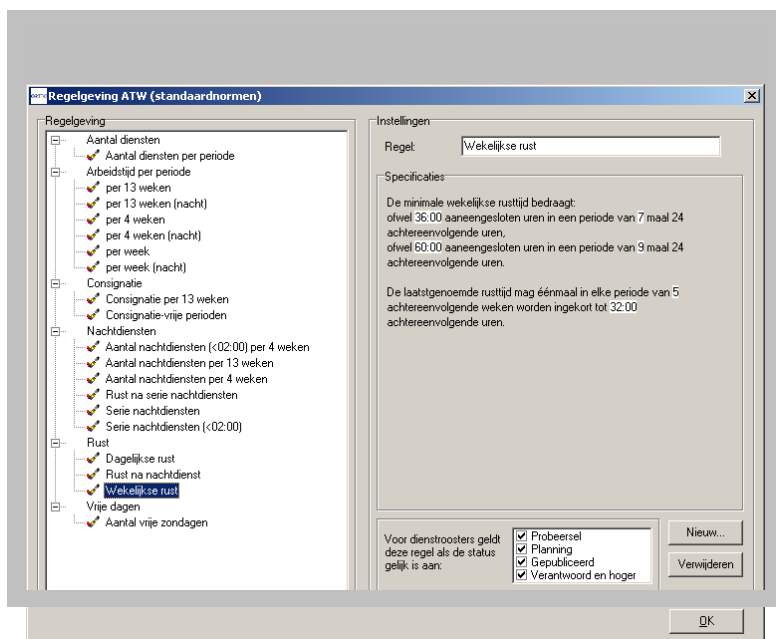
### 10.1 Arbeidstijdenwet, Arbeidstijdenbesluit en CAO

HARMONY is in staat alle regels uit de Arbeidstijdenwet, het

Arbeidstijdenbesluit, maar ook uit de CAO minutieus te controleren.

#### 10.1.1 Geparametriseerd

Alle regels zijn geparametriseerd in HARMONY opgenomen. Kleine wijzigingen in de CAO kunnen daardoor eenvoudig door een daartoe bevoegde medewerker worden doorgevoerd.



## 10.2

### 10.3 Per medewerker instelbaar

Binnen HARMONY is het mogelijk meerdere CAO's naast elkaar te voeren of medewerkers waarvoor andere normen uit de Arbeidstijdenwet gelden binnen één rooster te plannen.

### 10.4 En verder...

- Het mogelijk om tijdens de planningsfase strenger te controleren dan strikt noodzakelijk, waarmee ruimte wordt gecreëerd om ad hoc verstoringen op te lossen binnen de grenzen die de wet stelt.
- Schendingen van de regels worden in het planbord getoond zodra deze ontstaan.
- De planautomaten houden rekening met de regels en genereren schendingvrije roosters.

*De planner is en blijft uiteindelijk verantwoordelijk voor de planning. Hij kan als enige bepalen of een overtreding van de regels uit ATW of CAO noodzakelijk is.*

*In harmony kan de planner een schending van de regels negeren. De schending blijft echter wel zichtbaar in het planbord. Ter verantwoording kan de planner registreren waarom de overtreding in zijn ogen nodig is.*

## 11. Scenario Analyse

Zo af en toe is het nodig te experimenteren met het rooster, bijvoorbeeld op het moment dat belangrijke veranderingen voor de deur staan. Wat is het gevolg van een lagere zomerbezetting? Wat gebeurt er als we meer rekening houden met individuele wensen of een andere roostermethodiek introduceren?

### 11.1 Meerdere roosters voor dezelfde periode

Vanuit roosterbeheer is het mogelijk meerdere roosters voor dezelfde periode op te stellen. Voor elk rooster zijn de randvoorwaarden in te stellen en te veranderen. Verschillende alternatieven zijn op deze manier door te rekenen. Op het moment dat de planner tevreden is over één van de roosters kan hij deze als definitief bestempelen.

The screenshot shows a 'Scenario' dialog box with the following details:

- Scenario section:**
  - Naam: Alternatief 4x9
  - Van: 23-03-2003
  - Tot: 20-04-2003
- Onderdelen section:**
  - Dienstverzameling: Rayon Groningen
  - Bezettingseisen: Zomerbezetting
  - Cyclisch rooster: geen
  - Roostercriteria: Criteria 4x9
  - Verticale normen: Normen 2003
- Rooster section:**
  - Buttons: 'Uitrollen cyclisch rooster', 'Openen dienstrooster', 'Uitrollen bezettingseisen', 'Definitief maken'
- Bottom:** 'OK' button

## 12. Tijdregistratie & Urenverantwoording

De daadwerkelijk gewerkte uren zijn van belang voor de evaluatie van de planning, een juiste salarisbetaling en hebben bovendien invloed op de planning voor de toekomst. Zo kan het nodig zijn overuren te compenseren of moet het rooster worden aangepast om een overtreding van de Arbeidstijdenwet te voorkomen.

### 12.1 Kloktijden

Binnen HARMONY is het mogelijk kloktijden, afkomstig uit een tijdregistratiesysteem of van een prikklok te interpretern. Op deze wijze kan de leidinggevende bepalen of extra gewerkte uren als overuren gelden en hoe afwezigheid moet worden verrekend.

### 12.2 En verder...

- Als een medewerker zijn eigen uren verantwoordt, kunnen deze pas kunnen worden verwerkt als zij in het systeem zijn **geaccordeerd** door een bevoegd persoon.

## 13. Voorbereiding salarisverwerking

Veel salarissystemen zijn niet in staat de emolumenten, toeslagen, verrekeningen en compensaties die voortvloeien uit het rooster te berekenen.

### 13.1 Saldokaarten

Met het kaartmechanisme biedt HARMONY een unieke manier van berekening en weergave van alle arbeidsvoorwaardelijke gevolgen van het rooster. Het mechanisme is afgeleid van de ouderwetse (papieren) verlofkaart, waarop de afdeling personeelszaken aan het begin van het jaar het verloftegoed noteerde en waarop gedurende het jaar de genoten verlofuren in mindering werden gebracht.

### 13.2 Verloning op basis van de regels uit de CAO

De saldokaarten staan het toe berekeningen los te laten op het dienstrooster van een medewerker. De

berekeningen volgen uit bezoldigingsregels, die zijn opgenomen in de CAO of het contract dat geldt voor de medewerker. Op deze manier is HARMONY in staat het overwerk af te leiden op basis van complexe regelgeving, waarin ondermeer rekening wordt gehouden met dienstverschuiving of het moment waarop een dienst in de planning is gewijzigd of geruild. Precies zoals dat in de geldende CAO is vastgelegd.

Doordat de saldi van de verschillende kaarten en rubrieken daarop als tellingen in het planbord te tonen zijn, kan de planner al tijdens het plannen rekening houden met de arbeidsvoorwaardelijke gevolgen van zijn handelen.

### 13.3 Direct toegankelijk

Net als de tellingen in het planbord, worden de saldi op de saldokaarten direct bijgewerkt op basis van de veranderingen in het rooster. De kaarten zijn dus altijd up-to-date.

*Een medewerker wil direct inzicht in zijn verlofsaldo. Een manager wil op elk moment kunnen zien hoeveel overwerktoeslag er moet worden uitgekeerd of hoe het ziekteverzuim zich ontwikkelt. En een planner wil er wellicht naar streven de onregelmatigheidstoeslag gelijkmatig te verdelen.*

### 13.4 Correcties doorvoeren

Zo af en toe is het nodig een correctie door te voeren op een kaart. Bijvoorbeeld als een medewerker is vergeten door te geven dat hij een aantal maanden geleden heeft overgewerkt. Of als een medewerker vakantiedagen heeft ingeleverd in het kader van het PC privé-project. In het

eerste geval is de verloning over betreffende periode al uitgevoerd, in het tweede geval is het van belang de verlofregistratie te corrigeren. HARMONY biedt de mogelijkheid de saldi op de kaarten handmatig te corrigeren. De correcties moeten door een bevoegd persoon worden geaccordeerd.

### 13.5 En verder...

- Het kaartmechanisme geeft ook snel inzicht in bijvoorbeeld het totale verloftegoed van alle medewerkers in een roostergroep of het totale overwerk op een dag. Door de boekingen op de kaarten te **sorteren** en te **filteren** kan het juiste inzicht worden gekregen in de inzet van een bepaalde groep of deelgroep.
- Alle saldi van kaarten zijn te rapporteren en te exporteren naar bijvoorbeeld Microsoft Excel. Hiermee levert het kaartmechanisme de basis voor een schat aan **management informatie**.
- De saldi of boekingen van kaarten zijn te exporteren naar een **salarissysteem**.

## 14. Toegang via internet: Harmony-Online

Een medewerker komt zaterdag terug van een welverdiende vakantie. Daar zijn nieuwe dienstrooster nog niet bekend was op het moment dat hij naar het zonnige zuiden vertrok, weet hij niet hoe laat hij maandag weer door de baas verwacht wordt. Zal hij zijn planner storen tijdens zijn vrije weekend of niet?

### 14.1 Dienstrooster

Middels HARMONY-Online is het dienstrooster ook online beschikbaar. Zo kan een medewerker direct informatie over zijn dienstrooster opvragen via internet of intranet. Op elk moment van de dag en vanaf waar ook ter wereld. De informatie is altijd actueel.

### 14.2 Ruilingen

Afhankelijk van de vrijheid die u als organisatie aan uw medewerkers wilt geven, is het mogelijk om via HARMONY-Online ruilaanvragen in te dienen. Op het moment dat de tegenpartij akkoord gaat met het voorstel te ruilen kan deze ook via HARMONY-Online de ruiling accepteren, waarna deze wordt doorgevoerd. Uiteraard zorgt HARMONY ervoor dat hierbij geen schendingen ontstaan.

### 14.3 En verder..

- Een medewerker kan de actuele saldi van de verschillende kaarten opvragen. Zo kan hij vanuit huis zien hoeveel **verlofuren** hij nog heeft.
- Via HARMONY-Online kan de medewerker zijn **wensen** ten aanzien van het rooster kenbaar maken.

## 15. Rapportages & Overzichten

HARMONY is uitgerust met verschillende rapportagemogelijkheden. Vrijwel alle standaardrapporten zijn aanwezig. De ervaring leert echter dat de gewenste rapportage per organisatie verschilt. In aanvulling op de standaardrapporten definiëren we samen met de klant de exacte definities en lay-out van bedrijfsspecifieke rapporten.

### 15.1 Managementinformatie

Alle tellingen binnen HARMONY zijn direct te verwerken in rapportages en overzichten. Hetzelfde geldt voor de afzonderlijke boekingen, rubrieken en saldi van de saldokaarten. Doordat tellingen en saldi zijn te bewaren in verzamelingen is het eenvoudig de gewenste managementrapportages vanuit HARMONY te genereren.

### 15.2 Rapportgenerator

Door de open databasestructuur is het altijd mogelijk de gegevens uit HARMONY te benaderen en te rapporteren middels een query-tool of een rapportgenerator. Bovendien biedt de XML-interface de mogelijkheid direct te koppelen met een management informatiesysteem.

### 15.3 En verder...

- Alle rapporten en overzichten zijn direct vanuit HARMONY **via e-mail** te versturen naar de relevante medewerkers.
- Alle rapporten en overzichten zijn te exporteren in **Excel-formaat**, zodat verdere verwerking mogelijk is.



## 16. Integratie: Interfacing & Koppeling

HARMONY kan als stand-alone systeem worden ingezet, maar het is ook zeer goed mogelijk HARMONY volledig te integreren in uw administratieve processen.

### 16.1 Personeelsinformatiesysteem

Aan de 'voorkant' van het systeem is HARMONY te koppelen met vrijwel elk personeelsinformatiesysteem. Met diverse bekende systemen zijn standaardkoppelingen voorhanden. De gegevens uit het personeelsinformatiesysteem kunnen binnen HARMONY worden aangevuld. Daarnaast biedt HARMONY de mogelijkheid om medewerkers reeds te introduceren ten behoeve van de planning, waarbij in een later stadium de relevante gegevens worden aangevuld.

### 16.2 Prikklok & Tijdregistratiesysteem

Ten behoeve van de registratie van gewerkte uren is HARMONY zowel te koppelen met een prikklok als met een tijdregistratiesysteem. In het eerste geval doet HARMONY eveneens dienst als tijdregistratiesysteem, waarbij geldt dat HARMONY in staat is de juiste gegevens met betrekking tot het gerealiseerde rooster aan de salarisadministratie aan te leveren. Indien reeds een tijdregistratiesysteem aanwezig is, zorgt HARMONY dat de gevolgen van afwijkingen tussen realisatie en planning in het rooster zichtbaar zijn, zodat hiermee in de toekomst rekening wordt gehouden.

### 16.3 Salarissysteem

HARMONY beschikt over standaardkoppelingen met een groot aantal salarisverwerkende systemen. Afhankelijk van de behoefte en de mogelijkheden levert HARMONY de gewerkte tijden of de volledige berekening van toeslagen en vergoedingen ten behoeve van de verloning.

### 16.4 En verder...

- Door de **XML-interface** van HARMONY is koppeling met vrijwel elk systeem mogelijk.
- Voor diverse relaties is voorzien in integratie met **bedrijfsspecifieke toepassingen**. Voorbeelden hiervan zijn systemen met vluchtgegevens, de dienstregeling van treinen en looproutes van postbodes.
- ORTEC realiseert tevens de integratie met bijvoorbeeld een **ziekteregistratiesysteem**, een **facturatiesysteem** of een management informatiesysteem.
- HARMONY is onderdeel van de **ORTEC Advanced Planning Software Suite**. Deze biedt toegang tot een veelheid aan functionaliteit op het gebied van ondermeer projectplanning, multi-resourceplanning, taakplanning en routeplanning.
- ORTEC heeft HARMONY al diverse keren met succes geïntegreerd in **ERP-systemen**.

## **17. Toegangsbeveiliging**

De toegangsbeveiliging van HARMONY is in te stellen middels HARMONY SECURITY. Deze applicatie geeft de applicatiebeheerder of systeembeheerder de mogelijkheid om medewerkers toegang tot HARMONY te verlenen op basis van zelf te definiëren rollen. Een rol bepaalt de functionaliteit waarover de medewerker beschikt.

Afhankelijk van de autorisaties per rol krijgt de gebruiker rechten om gegevens te raadplegen of om aan te passen. De toegang kan verder worden beperkt tot één of meerdere roostergroepen.

Indien gewenst gebeurt identificatie van de gebruiker op basis van de aanlog via Windows. Hierdoor is het niet nodig dat de gebruiker zich apart aanmeldt bij HARMONY.

## 18. Configuratiemogelijkheden

HARMONY kent uitgebreide configuratiemogelijkheden, waardoor het systeem als het ware op maat van de organisatie wordt gesneden. Middels instellingen die van buiten het systeem worden gemaakt, is het mogelijk functionaliteit toe te voegen of weg te laten.

*configuratie-instellingen kunnen de consultants van ORTEC het systeem perfect afstemmen op uw organisatie en werkwijze.*

Onze maatwerkhistorie brengt het met zich mee dat we altijd bereid zijn **functionaliteit op maat** te ontwikkelen. Hierbij kunnen we garanderen dat de functionaliteit in toekomstige releases behouden blijft.

## 19. Technische Karakteristieken

HARMONY werkt op bekende besturingssystemen als Microsoft Windows 95/98, NT, 2000 en XP. In het ontwerp is uitgegaan van een zogenaamde 3-tier (3-lagen) componentenarchitectuur, met verschillende lagen voor:

- gegevensopslag,
- communicatie/integratie en business logic,
- user interface.

Voor de uitwisseling van informatie wordt gebruik gemaakt van de protocollen:

- DCOM als integratiemechanisme (een open standaard voor gedistribueerde systemen), en
- ODBC (open database connectivity).
- XML (extensible markup language).

Gegevensopslag gebeurt, afhankelijk van de systeemeisen, met behulp van een ODBC-benaderbare database server zoals bijvoorbeeld Oracle of MS SQL server. Door de open structuur zijn koppelingen met

andere systemen eenvoudig te realiseren.

De grafische user interfaces zijn ontwikkeld conform Windows-standaarden. Bediening van het systeem is hierdoor intuïtief.

HARMONY kan als stand-alone applicatie ingericht worden, maar is tevens volledig uitgerust voor concurrent multi-user en multi-site gebruik. Dit betekent dat verschillende planners gelijktijdig op verschillende locaties op dezelfde data kunnen werken.

Door het gebruik van DCOM kan HARMONY onder verschillende netwerkprotocollen opereren, waaronder TCP/IP, IPX, NetBUI en SPX. HARMONY is eveneens te gebruiken via Terminal Server of in een Citrix omgeving.

## **Externe bijlage III   Plan van Aanpak**

# **Plan van Aanpak HARMONY-WebAccess**

Versie 1.0

Alex Nap

# Tekenpagina

Wijzigingshistorie:

Versie	Datum	Auteur(s)	Wijzigingen	Goedgekeurd door		
				Datum	Naam	Handtekening

# 1. Inhoudsopgave

## 1. INLEIDING 1

## 2. OPDRACHTOMSCHRIJVING 2

- 2.1 PROBLEEMSTELLING 2
- 2.2 DOELSTELLING 2

## 3. OP TE LEVEREN PRODUCTEN 3

- 3.1 PRODUCTEN 3
- 3.2 OMSCHRIJVING PER PRODUCT 3

## 4. METHODEN EN TECHNIEKEN 4

- 4.1 METHODE 4
- 4.2 TECHNIEKEN 4

## 5. PROJECTRISICO'S 4

## 6. PROJECTKADER 7

- 6.1 DEELNEMERS 7
  - 6.1.1 Alex Nap 7
  - 6.1.2 Ferry Bos 7
  - 6.1.3 Peter van 't Hoff 7
  - 6.1.4 Interne gebruikersgroep 7
- 6.2 SPLITSING PROJECT IN FASES 7
- 6.3 SAMENHANG 8
- 6.4 GLOBALE PLANNING 8
- 6.5 SCHEMA GLOBALE PLANNING 10

## 7. ONTWIKKELOMGEVING 11

- 7.1 PROGRAMMEERTAAL 11
  - 7.2 HARDWARE EN SOFTWARE 11
-

## **2. Inleiding**

Dit Plan van Aanpak beschrijft de probleemstelling en de globale oplossingsmethode. Tevens kadert het document de grenzen van het project af en specificeert het de rollen van de deelnemers aan het project.

In de verschillende hoofdstukken zullen de zaken worden behandeld die dienen als uitgangspunt om dit project met succes te kunnen afronden.

In hoofdstuk 2 zal de probleemstelling en de doelstelling van dit project worden beschreven. Dit is tevens de doelstelling van de afstudeeropdracht die de afstudeerder gaat vervullen.

Hoofdstuk 3 behandelt de producten, tussenproducten en eindproducten, die gedurende de projectperiode zullen worden opgeleverd. Deze lijst is een duidelijke checklist voor het proces van opleveren. Als verduidelijking wordt per product ook nog aangegeven wat de inhoud zal zijn. Zo kan hier geen onduidelijkheid over ontstaan.

In hoofdstuk 4 zullen de te gebruiken methoden en technieken worden beschreven.

Bij elk project moeten projectrisico's onderkend worden, zodat adequaat gehandeld kan worden in het geval een risico werkelijkheid wordt. De risico's die voor dit project onderkend worden staan beschreven in hoofdstuk 5.

Hoofdstuk 6 geeft een beeld van het projectkader. Het projectkader beschrijft de personen die deelnemen aan het product. De afstudeerder staat in dit geval het meest centraal, de andere personen hebben voornamelijk een controlerende en ondersteunende rol.

Als laatste zal in hoofdstuk 7, ontwikkelomgeving, worden beschreven welke programmeertaal gebruikt zal worden en hoe de ontwikkelomgeving eruit zal zien.



### **3. Opdrachtschrijving**

In dit hoofdstuk worden de probleem- en doelstelling van het project beschreven.

#### **3.1 Probleemstelling**

HARMONY-WebAccess zoals die er nu is bevat slechts enkele triviale basis-functionaliteiten van de gewone client. Dit betekent dat niet alle functionaliteiten, zoals die in de gewone client zitten beschikbaar zijn via het internet.

De gegevens die nu beschikbaar zijn via het internet zijn over het algemeen passief. Dit wil zeggen dat een medewerker wel kan zien voor wanneer hij ingeroosterd is, maar niets kan wijzigen. Zo kan hij bijvoorbeeld niet ingeven wanneer hij vakantie wil. Dit kan hij via de gewone client of via de planner doorgeven. Dit wordt gezien als een belemmering, omdat het de planner ontlast als medewerkers zelf in kunnen vullen wanneer ze beschikbaar zijn. Een taak van de planner is nu dat hij medewerkers een planning toestuuft of moet doorbellen, hier gaat ook veel tijd in zitten als er veel medewerkers zijn.

#### **3.2 Doelstelling**

Het doel van de opdracht is het ontwikkelen van uitbreidingen op HARMONY-WebAccess, zodat medewerkers in HARMONY-WebAccess hun urenverantwoording kunnen invoeren (geplande activiteiten kunnen aanpassen of verwijderen en activiteiten kunnen toevoegen), bekijken van hun wensen, overzicht van ruilingen kunnen bekijken en aan kunnen geven dat ze hun rooster of ruiling(en) hebben gezien.

## **4. Op te leveren producten**

Door het project heen dient de afstudeerder een aantal (tussen)producten op te leveren. Sommige zijn geschikt voor extern gebruik sommige zijn alleen geschikt voor intern gebruik. In de opsomming hieronder staan de producten die opgeleverd gaan worden.

### **4.1 Producten**

#### **Kick-off:**

- Plan van aanpak
- Beschrijving van te ontwikkelen functionaliteiten

#### **Functioneel ontwerp**

#### **Technisch ontwerpen**

#### **Bouwfase:**

- Ontwikkelde functionaliteiten
- Testplan documenten
- Gebruikersdocumentatie

#### **Implementatiefase:**

- Getest systeem

#### **Afstudeerverslag**

### **4.2 Omschrijving per product**

#### **Plan van aanpak**

Het plan van aanpak beschrijft de probleemstelling en de globale oplossingsmethode. Tevens kadert het document de grenzen van het project af en specificeert het de rollen van de deelnemers aan het project.

#### **Beschrijving van te ontwikkelen functionaliteiten**

In deze beschrijving staan de functionaliteiten die in HARMONY-WebAccess toegevoegd moeten worden. Per functionaliteit wordt beschreven aan welke eisen de functionaliteit moet voldoen en wat de prioriteit van ontwikkeling is.

#### **Functioneel ontwerp**

In het functioneel ontwerp zullen de functionaliteiten zoals die beschreven zijn in het document Beschrijving van te ontwikkelen functionaliteiten, verder uitwerkt worden. Per functionaliteit zal een ontwerp gemaakt worden met alle punten die daarvoor van belang zijn.

#### **Technisch ontwerpen**

In de technisch ontwerpen zullen de technische specificaties en de technische eisen van de in het functioneel ontwerp beschreven functionaliteiten beschreven worden. Er zullen meerdere ontwerpen zijn, omdat er 3 incrementen zijn en per increment wordt een technisch ontwerp gemaakt.

#### **Ontwikkelde functionaliteiten**

Dit zijn de ontwikkelde functionaliteiten zoals beschreven in de doelstelling van de opdracht. Dit product zal uiteindelijk in HARMONY-WebAccess geïmplementeerd worden. Het product is een combinatie van grafische en technische elementen.

#### **Testplan documenten**

Elk ontwikkelde functionaliteit moet getest worden. Dit zal geschieden met behulp van een testplan. Voor elk ontwikkelde functionaliteit wordt een testplan geschreven en zal als leidraad dienen voor de persoon die de functionaliteit gaat testen. De tests van de

functionaliteiten zullen geschieden door zowel de ontwikkelaar als door een persoon die niet direct betrokken is geweest bij de ontwikkeling.

### **Gebruikersdocumentatie**

Van de opgeleverde functionaliteiten zal ook gebruikersdocumentatie gemaakt worden. Deze documentatie beschrijft de werking van de functionaliteiten. De gebruikers kunnen uit deze documentatie hun informatie halen wanneer zij hulp nodig hebben op een bepaald punt.

### **Getest systeem**

Het geteste systeem bevat de ontwikkelde functionaliteiten die zijn ontworpen aan gebruikerstesten. Hierbij wordt gekeken naar de functionele kant. De vraag is hier dan ook: "Treden er fouten op?". Andersoortige problemen als ontbrekende functionaliteiten worden hier niet behandeld. Dit zal gebeuren bij het product "Functioneel ontwerp".

### **Afstudeerverslag**

De afstudeerperiode van de afstudeerder zal door twee examinatoren en een gecommitteerde beoordeeld worden. Deze personen zullen door school aangesteld worden. De beoordeling zal gedaan worden aan de hand van een afstudeerverslag waarin het proces en het eindproduct zullen worden beschreven. Belangrijke elementen in dit verslag zullen onder andere de keuzemomenten van de afstudeerder binnen het project zijn. Daarnaast moeten alle keuzes goed onderbouwd zijn.

## **5. Methoden en technieken**

Dit hoofdstuk beschrijft de methoden en technieken die de afstudeerder gaat toepassen tijdens het project.

### **5.1 Methode**

Binnen de afdeling Arbeidstijdmanagement is een standaard opgesteld die gebruikt wordt bij de ontwikkeling / uitbreiding van systemen. De afstudeerder zal deze standaard gebruiken.

### **5.2 Technieken**

De verschillende tussenproducten zullen op een bepaalde manier afgerond worden. Hiervoor zullen de verschillende technieken gebruikt worden.

Tijdens de kick-off zal er een brainstormsessie gehouden worden met de begeleider, de opdrachtgever en een aantal mensen uit de gebruikersgroep van ARTM medewerkers. Met het houden van deze brainstormsessie moet naar voren komen wat voor functionele eisen er zijn met betrekking tot het systeem. Tevens zal hier ook bepaald worden welke functionaliteit de hoogste prioriteit heeft. Voor het houden van de brainstormsessie zal gekozen worden voor de klassieke brainstorm methode.

Voor het maken van het functioneel ontwerp, de technisch ontwerpen en de testplannen zal gebruik gemaakt worden van de templates zoals die opgesteld zijn door de afdeling Arbeidstijdmanagement.

Voor het beschrijven van de processen en functionaliteiten zal gebruik gemaakt worden van UML (Unified Modeling Language). UML is eigenlijk geen techniek, maar een modelleringstaal.

## **6. Projectrisico's**

Aan het begin van dit project zijn een aantal risico's onderkend. Per risico wordt uitgelegd wat de inhoud van het risico is, wanneer het op kan treden en wat voor maatregelen er genomen kunnen / moeten worden om het risico te kunnen verhelpen.

### **Risico 1: Onduidelijkheid over de inhoud van de gewenste functionaliteiten.**

Omschrijving	De functionaliteiten zoals genoemd tijdens brainstormsessie zijn verkeerd geïnterpreteerd door de afstudeerder.
--------------	---

Moment van mogelijk optreden	Tijdens maken van Functioneel Ontwerp
Signaal waaraan risico is te onderkennen	Tijdens het reviewen door bedrijfsmentor(en).
Consequenties voor project	Aanpassing Functioneel ontwerp

**Detectiemaatregelen:**

Tijdens review van het functioneel ontwerp blijkt dat de functionaliteiten niet zijn uitgewerkt zoals de gebruikersgroep (zie H7.1.4) dat voor ogen heeft.

**Preventieve maatregelen:**

Voldoende tijd besteden aan het achterhalen van daadwerkelijke inhoud van elke functionaliteit en vroegtijdig afstemmen met de gebruikersgroep.

**Correctieve maatregelen:**

Na review het functioneel ontwerp aanpassen of gebruikersgroep duidelijk maken waarom de functionaliteit op deze manier is uitgewerkt (bijvoorbeeld om technische redenen). Na eventuele aanpassingen het functioneel ontwerp weer laten reviewen.

**Risico 2: Onervarenheid met programmeeromgeving Delphi.**

Omschrijving	De programmeeromgeving van Delphi is nieuw voor de afstudeerder. In de planning is hier rekening mee gehouden, maar het kan zijn dat de afstudeerder het minder snel zal leren dan gedacht.
Moment van mogelijk optreden	Tijdens het maken van het TO en het programmeren van de beschreven functionaliteiten.
Signaal waaraan risico is te onderkennen	Als het programmeren langer duurt dan gepland.
Consequenties voor project	Uitloop van het project

**Detectiemaatregelen:**

De planning goed in de gaten houden, zodat snel inzichtelijk is aan de hand van de nog beschikbare tijd en de nog te ontwikkelen functionaliteiten, of de planning gehaald kan worden.

**Preventieve maatregelen:**

In de planning rekening houden met de ervaring van de afstudeerder. Begeleiding bij het maken van het TO en het programmeren door de bedrijfsmentor.

**Correctieve maatregelen:**

Alleen de echt benodigde functionaliteiten ontwikkelen en de minder belangrijke zaken in een volgend project oppakken.

**Risico 3: Langdurige afwezigheid bedrijfsmentor**

Omschrijving	Bedrijfsmentor is langdurig (meer dan 5 dagen) afwezig,
Moment van mogelijk optreden	Gedurende het hele project.
Signaal waaraan risico is te onderkennen	Als bedrijfsmentor niet aanwezig door bijvoorbeeld ziekte of verlof.
Consequenties voor project	Minder begeleiding dan wenselijk is.

**Detectiemaatregelen:**

Zodra bedrijfsmentor afwezig is, kijken of dat ingepland is in zijn agenda. Als dit niet het geval is bij afdelingschef navragen waarom de bedrijfsmentor niet aanwezig is.

**Preventieve maatregelen:**

Er voor zorgen dat de bedrijfsmentor preventief een vervanger aanstelt en deze vervanger op de hoogte houden van de voortgang van het project zodat hij bekend is met het project.

**Correctieve maatregelen:**

De vervanger op de hoogte stellen van de laatste ontwikkelingen van het project, zodat hij de begeleiding kan geven die gewenst is.

**Risico 4: Wijziging van wensen / eisen tijdens bouw**

Omschrijving	Tijdens bouw wordt er opeens door de gebruikersgroep geconstateerd dat het wel handig zou zijn als een bepaalde functionaliteit ook ontwikkeld zou worden (ad-hoc wensen).
Moment van mogelijk optreden	Tijdens bouwen van de ingeplande functionaliteiten.
Signaal waaraan risico is te onderkennen	Als een van de leden van de gebruikersgroep aan de afstudeerder gaat vragen of een bepaalde functionaliteit niet eerst of erbij ontwikkeld kan worden.
Consequenties voor project	Uitloop van het project, waardoor de ingeplande functionaliteiten niet ontwikkeld kunnen worden.

**Detectiemaatregelen:**

Iemand van de gebruikersgroep komt met nieuwe wensen of ideeën.

**Preventieve maatregelen:**

De gebruikersgroep intensief betrekken bij het functioneel ontwerp en uiteindelijk laten accorderen door de gebruikersgroep, zodat er overeenstemming over de te ontwikkelen functionaliteit is.

Zonder akkoord wordt er niet verder ontwikkeld.

**Correctieve maatregelen:**

Direct afstemmen met de bedrijfsmentor in hoeverre de nieuwe wensen ingepland moeten worden.

## **7. Projectkader**

Dit hoofdstuk zal verder ingaan op de verdeling van de verantwoordelijkheden over verschillende mensen die naast de afstudeerder betrokken zijn bij het project.

### **7.1 Deelnemers**

Om het project tot een succes te maken zijn er meerdere personen nodig. De volgende personen, samen met hun rollen, staan hieronder benoemd.

#### **7.1.1 Alex Nap**

Alex is als afstudeerder de hoofdsverantwoordelijke voor dit project. Hij zal alle onderdelen van dit project uitvoeren. Zijn bijdrage omvat 36 uur per week. De vier uur die per week overblijven zijn gereserveerd voor algemene zaken zoals het verwerken van e-mail, vergaderingen en andere algemene zaken.

#### **7.1.2 Ferry Bos**

Ferry is de eerste begeleider van Alex en zal daarom als eerste alle documenten beoordelen. Tevens is hij verantwoordelijk voor het bijspringen op het technische vlak van het project. Hierbij kan gedacht worden aan de manier voor het aanroepen van de benodigde objecten. Voor deze taken is vier tot acht uur per week ingepland. Dit is afhankelijk van de hoeveelheid ondersteuning die benodigd is.

#### **7.1.3 Peter van 't Hoff**

Peter is de interne opdrachtgever. Hij is verantwoordelijk voor het stellen van de voorwaarden waaraan het project moet voldoen. Alex kan voorstellen doen in de vorm van documenten. Deze zullen echter uiteindelijk altijd door Peter goedgekeurd moeten worden. Peter is de persoon die de definitieve beslissing(en) zal nemen. Voor deze taak is twee uur per week ingepland.

#### **7.1.4 Interne gebruikersgroep**

De interne gebruikersgroep zal bestaan uit personen die veel ervaring hebben met klanten die nu gebruik maken van HARMONY(-WebAccess) en zo ook een goed beeld hebben van de wensen van deze klanten met betrekking tot HARMONY-WebAccess. Deze groep zal bestaan uit een drietal personen, te weten: Wouter Luijten, Marjolein Jungman en Leen Westerhof. Van deze groep personen wordt verwacht dat zij deelnemen aan een brainstormsessie om zo de functionele eisen voor het systeem te bepalen. Daarnaast wordt van deze mensen verwacht dat zij mee doen aan het evalueren van het functioneel ontwerp. Dit zal maximaal twee keer voorkomen en zal gedaan worden door middel van een brainstormsessie van één uur.

### **7.2 Splitsing project in fases**

Om overzicht te houden en controle uit te kunnen oefenen is het project opgedeeld in fases. Deze fases kunnen over het algemeen niet parallel uitgevoerd worden, omdat ze onderling van elkaar afhankelijk zijn.

De fases:

- Algemene fase (Kick-off en functioneel ontwerp)
- Bouwfase (drie incrementen)
  - Implementatiefase

### **7.3 Samenhang**

Bij ORTEC wordt gebruik gemaakt van iteratieve ontwikkeling. In dit project zal alleen in de Bouwfase spraken zijn van iteratieve ontwikkeling. Er zijn drie incrementen in de planning opgenomen.

Voor elk increment staat een bepaalde tijd. Het kan gebeuren dat een increment niet binnen deze tijd gereed zal zijn. Mocht dit het geval zijn dan zal worden bekeken of de niet gereed gekomen werkzaamheden meegenomen zullen worden in het volgende increment of dat hier een nieuw increment van gemaakt zal worden die dan na het project uitgevoerd zal worden.

### **7.4 Globale Planning**

In dit Plan van Aanpak is een globale planning opgezet. Per increment zal een aparte planning gemaakt worden met de te verrichten taken. Het kan namelijk gebeuren dat niet alle taken van een increment binnen de vastgestelde tijd gereed komen. In dit geval kan besloten worden om deze taken door te schuiven naar het volgende increment. Daarom zal voor de aanvang van elk increment een planning gemaakt worden.

#### **Week 1**

Deze week zal dienen als inwerkperiode in het project en het configureren van de software en hardware. Daarnaast zal er onderzoek gedaan worden naar de wensen met betrekking tot de uitbreiding van HARMONY-WebAccess (HW).

#### **Week 2**

In week 2 zullen door middel van een brainstormsessie de wensen met betrekking tot de uitbreiding van HW duidelijk worden. Aan de hand van deze wensen en de prioriteit die aan een elke wens is gekoppeld, kan bepaald worden wat voor werkzaamheden verricht moeten worden en in welke volgorde. Aan de hand van deze gegevens kan een planning opgesteld worden. Deze planning zal opgenomen worden in dit Plan van Aanpak. Het Plan van Aanpak zal tot doel hebben om het komende project te beschrijven en het kader waarbinnen het project valt vast te stellen.

#### **Week 3**

In deze week zal begonnen worden met het maken van het Functioneel Ontwerp (FO). In dit ontwerp zullen alle gewenste functionaliteiten, die in dit project ontwikkeld zullen worden, beschreven worden.

#### **Week 4**

Verder gaan met het beschrijven van de functionaliteiten en schermontwerpen maken.

#### **Week 5**

Afronden van het FO. Het ontwerp zal gecontroleerd worden door de bedrijfsmentoren. Na controle zullen er eventuele aanpassingen gedaan worden. Na de controle en eventuele aanpassingen zal gestart worden met het 1e increment. Per increment zal een Technisch Ontwerp (TO) gemaakt worden. In het 1e increment zullen dus alleen de functionaliteiten 'Wachtwoord wijzigen' en 'Uren verantwoorden' in detail uitgewerkt worden.

#### **Week 6**

Controleren van het TO door de bedrijfsmentoren en eventueel aanpassingen doen in het TO. Na goedkeuring van het TO zal begonnen worden met het programmeren van de functionaliteiten.

#### **Week 7**

Verder gaan met het programmeren van de functionaliteiten.

#### **Week 8**

Testplan schrijven om de ontwikkelde functionaliteiten te kunnen testen. Ontwikkelde functionaliteiten laten testen door de bedrijfsmentor(en) en eventueel aanpassingen doen

in de programmatuur. Beginnen met schrijven Gebruikershandleiding en opleveren van functionaliteiten.

#### **Week 9**

Start 2e increment. In het TO zullen de functionaliteiten 'Overzicht ruilingen' en 'Bevestigen ruilingen gezien' in detail uitgewerkt worden. Tevens zal er weer een controle plaatsvinden door de bedrijfsmentoren en eventueel aanpassingen gedaan worden. Na goedkeuring van het TO zal begonnen worden met het programmeren van de functionaliteiten.

#### **Week 10**

Verder gaan met het programmeren van de functionaliteiten.

#### **Week 11**

Afronden van het programmeren en schrijven testplan. Aan de hand van testplan de ontwikkelde functionaliteiten testen. Deze test zal ook door de bedrijfsmentor(en) gedaan worden.

#### **Week 12**

Afronden testen functionaliteiten en eventueel laatste aanpassingen doen in de programmatuur. De Gebruikershandleiding uit het 1e increment uitbreiden en opleveren van de functionaliteiten. Tevens zal begonnen worden met het 3e increment. In het TO zullen de functionaliteiten 'Bevestigen rooster gezien' en 'Wensen bekijken' in detail uitgewerkt worden.

#### **Week 13**

Controleren van het TO door de bedrijfsmentoren en eventueel aanpassingen doen in het TO. Na goedkeuring van het TO zal begonnen worden met het programmeren van de functionaliteiten.

#### **Week 14**

Verder gaan met het programmeren van de functionaliteiten.

#### **Week 15**

Schrijven Testplan voor het testen van de ontwikkelde functionaliteiten. Tevens zullen de functionaliteiten getest worden en eventueel aangepast worden. Het document Gebruikershandleiding zal verder uitgebreid worden en de functionaliteiten zullen opgeleverd worden.

#### **Week 16**

In deze week zal het geteste systeem operationeel gemaakt worden. Hierna zal begonnen worden met het maken van het afstudeerverslag. Daarnaast zal er ook ruimte zijn voor het afhandelen van allerhande zaken.

#### **Week 17**

Verder gaan met het maken van het afstudeerverslag en afhandelen overige zaken.

#### **Week 18**

Verder gaan met het maken van het afstudeerverslag en afhandelen overige zaken.

#### **Week 19**

Verder gaan met het maken van het afstudeerverslag en afhandelen overige zaken.



## 7.5 Schema globale planning

[illegible]

## **8. Ontwikkelomgeving**

In dit hoofdstuk wordt beschreven welke programmeertaal gebruikt zal worden en welke hard- en software de afstudeerder tot zijn beschikking heeft.

### **8.1 Programmeertaal**

Het ontwikkelen van de functionaliteiten zal gedaan worden in de ontwikkelomgeving Borland Delphi 6. Om de ontwikkelde code in webpagina's te kunnen tonen wordt gebruik gemaakt van WebSnap. WebSnap is onderdeel van de ontwikkelomgeving van Borland Delphi. Voor het afhandelen van functies aan de client-kant zal gebruik gemaakt worden van JavaScript.

### **8.2 Hardware en software**

De afstudeerder heeft voor de uitvoering van het project de volgende zaken tot zijn beschikking:

- PC met Microsoft Windows 2000
- Printer
- Netwerkverbinding
- Borland Delphi 6
- Microsoft Internet Information Server
- Microsoft Office 97
- Microsoft Visio 2000
- Oracle 8
- MKS Source Integrity

## **Externe bijlage IV    Checklists standaard ARTM**

### **Projectaansturing en samenwerking zusters**

#### **Checklist aansturing projectteam**

Bij het aansturen van een projectteam

⇒ gebruik je deze [checklist](#)

#### **Checklist risicoanalyse**

Maak voor de start van een nieuw project een risicoanalyse

⇒ klik [hier](#) voor een voorbeeld

#### **Checklist RAEW-matrix**

Maak voor de start van een nieuw project een matrix met verantwoordelijkheden van de verschillende partijen

⇒ klik [hier](#) voor een voorbeeld

#### **Checklist samenwerking zusters**

Als je binnen een project of acquisitietraject met een zuster samenwerkt

⇒ gebruik je deze [checklist](#)

## Checklist aansturing projectteam

### Doelen

- Samenwerking in het project verbeteren
- Verbeteren van de planning van het project (voor interne en externe communicatie)
- Project uitvoeren binnen budget & deadlines
- Het leveren van gewenste kwaliteit
- Variatie in werkzaamheden

### Samenwerking

- Voordat het project start moet het voor alle teamleden duidelijk zijn wie welke rol heeft en welke taken / bevoegdheden daarbij horen. Plan hiervoor altijd een kick-off meeting.
- Ga bij voorkeur met het hele projectteam op een kamer zitten. Dit bevordert de teamspirit en communicatie.
- Zorg voor een duidelijke overdracht van taken als een van de teamleden afwezig is (inclusief jezelf).
- Zorg dat algemeen projectleider en technisch projectleider zoveel mogelijk samen naar bijeenkomsten met de klant gaan.

### Planning

#### Bevindingendatabase

Zorg dat de nieuwe (bouw)items allemaal bekend zijn in de bevindingendatabase. Het is verstandig om dit al te doen bij oplevering van het Functioneel Ontwerp.

#### TRS

Maak voor een nieuw project altijd een nieuwe fase aan voor de betreffende klant. In onderstaande tabel is aangegeven welke onderdelen een project tenminste moet bevatten. De onderdelen worden voorafgegaan om de gewenste sortering op de diverse overzichten uit TRS te kunnen bepalen.

Onderdeel	Taken
010 Bouw	Alle items die gebouwd moeten worden. Het werkt het handigst als deze items als bevinding zijn ingevoerd in de bevindingendatabase. De betreffende nummers kunnen dan in de naam van de taak worden opgenomen.
020 Review	Idem
030 Systeemtest	Idem
040 Oplossen testbevindingen	Idem
050 Integratietest 1	<ul style="list-style-type: none"> <li>– Voorbereiden integratietest</li> <li>– Uitvoeren integratietest</li> <li>– Invoeren bevindingen</li> </ul>
051 Oplossen nav int. test 1	<ul style="list-style-type: none"> <li>– Bugs oplossen (algemeen)</li> <li>– Opgeloste bugs reviewen</li> <li>– Opgeloste bugs hertesten</li> <li>– Als het bugs betreft in de nieuwe functionaliteit moet er voor dit item een nieuwe taak worden aangemaakt.</li> </ul>
052 Integratietest ...	
053 Oplossen nav int.test ...	
060 Overig	<ul style="list-style-type: none"> <li>– Kick-off meeting: plan deze meeting altijd in (per increment) om de te bouwen functionaliteit met de bouwers door te nemen zodat de bouw met meer klant focus kan plaatsvinden en om eventuele gaps in de</li> </ul>

	beschrijving in een vroeg stadium te detecteren. – Teamoverleg – Overleg met klant – Onvoorzien – ...
070 Installatie	– Voorbereiden oplevering – Installatie op locatie of Versturen nieuwe versie – Nazorg installatie
080 Acceptatietest	– Begeleiden van de klant – Invoeren van bevindingen – Analyseren van bevindingen – Bugs oplossen, reviewen en hertesten
090 Projectmanagement	– Intern project management & overleg – Extern project management

### **Begroting**

Vul voor alle items die in TRS zijn aangemaakt de begroting in. Gebruik het Bestedingsoverzicht om te controleren of het project dezelfde omvang heeft als de aan de klant doorgegeven omvang. Gebruik de [checklist begroten](#) voor het maken van de begroting.

### **Doorlooptijdplanning**

Om het mogelijk te maken om de klant danwel de algemeen projectleider te informeren over de doorlooptijd is het van belang dat alle items die in de oorspronkelijke begroting zijn genoemd worden opgenomen in MS Project. Houd hierbij rekening met de beschikbare capaciteit volgens de capaciteitsplanning. Gebruik de [checklist doorlooptijdplanning](#). Zorg ook dat je op de hoogte bent van de planning van andere projecten die mogelijk invloed hebben op jouw project (het 'pijlendiagram'). Dit kunnen ook interne projecten zijn, zoals het omzetten van requirements naar de nieuwe structuur.

### **Korte termijn planning**

Aangezien de planning altijd onderhevig is aan verstoringen is het verstandig om wekelijks de planning voor de komende week te maken. Deze planning wordt genoteerd in een weeklijst.

Een weeklijst moet de volgende informatie bevatten :

- Alle teamleden
- Per teamlid:
  - Nummer van de toegewezen taak (indien mogelijk)
  - Onderwerp van de taak
  - Actie (bijv. TO, Bouw, Review, Test, Oplossen in geval van items uit de bevindingendatabase)
  - Aantal uren dat is begroot
  - Deadline (houd hierbij rekening met de geplande vergaderingen en afwezigheden) – hierin is geen onvoorzien opgenomen, deze deadline ligt altijd voor de deadline die met de klant is afgesproken.

Leg deze weeklijst voor aan de teamleden en vraag hen te controleren of ze alle informatie hebben. Het gaat hierbij met name om de beschikbare FO's en TO's. Vraag tevens om een inschatting voor de toegewezen taken. Bespreek de afgegeven schatting als er een duidelijke afwijking is ten opzichte van de oorspronkelijke begroting en stel eventueel de begroting bij (in TRS door Onvoorzien op te nemen dan wel te reserveren).

Niet onbelangrijk: wijs de teamleden op afhankelijkheden in de planning. Het gaat hierbij met name om de items die aan elkaar moeten worden 'doorgegeven' en om items die van elkaar afhankelijk zijn.

Opmerkingen:

- Het aantal taken dat per week kan worden ingepland is afhankelijk van de beschikbare capaciteit volgens de capaciteitsplanning. Indien de medewerker voltijds is ingepland voor een project plan dan geen 40 uur in maar houd de vrijdagmiddag vrij. Dit in verband met telefoontjes van klanten en binnenwandellende collega's waardoor de deadlines nauwelijks haalbaar zullen zijn. Geef overigens wel aan wat de taken voor de maandag en dinsdag van de daaropvolgende week zijn, zodat de medewerker wel gewoon door kan werken.

### **Realisatie van de planning**

Gebruik het bestedingsoverzicht uit TRS om de planning te bewaken, maar vraag medewerkers ook zelf om dit zo spoedig mogelijk te melden zodra ze denken te gaan uitlopen. Vraag het betreffende teamlid altijd naar de oorzaak van uitloop. Doe dit niet eens per week maar continu om de kans op uitloop te minimaliseren en de uitloop zelf binnen de perken te houden door bijvoorbeeld iemand mee te laten kijken naar wat er gedaan is en wat er nog moet gebeuren. Vergeet niet om inloop te 'belonen'!

### **Kwaliteit**

- Monitor of de projectteamleden de afgesproken richtlijnen en checklists ook daadwerkelijk volgen, zie er o.a. op toe dat gebouwde onderdelen ook gereviewed en getest worden.
- Laat een medewerker pas beginnen aan een volgende taak als de vorige af is.
- Zorg bij tests door klanten dat ze in een aparte ruimte kunnen zitten. Laat indien mogelijk de programmeur van het testen onderdeel bij de test aanwezig zijn. Zorg dat klanten echt focussen op de onderdelen die ze moeten testen en laat ze zichtbaar maken wat ze testen en hoe ver ze ermee zijn (bijv. lijst afvinken).
- Zorg voor een specifiek e-mail adres (bijv. [harmonyesso@ortec.nl](mailto:harmonyesso@ortec.nl)) waarnaar de klant vragen en/of bevindingen kan mailen. De mail komt terecht in een public folder in Outlook, zodat ieder teamlid deze kan bekijken c.q. beantwoorden. Zet de verwerkte mails in een aparte map.

### **Inhoud van het werk**

Algemene opmerkingen:

- Voor elke medewerker geldt dat het prettig is om ook eens iets anders in een project te doen. Houd hier rekening mee bij zowel de doorlooptijdplanning als het opstellen van de weeklijsten.
- Laat indien mogelijk de bouwer zelf het technisch ontwerp maken. Dit biedt meer ruimte voor eigen inbreng en creativiteit en dwingt de bouwer goed na te denken over het te bouwen onderdeel.
- Probeer bij het toewijzen van de taken rekening te houden met het ontwikkeltraject van de medewerker. Voorbeelden: iemand die in de toekomst technisch projectleider wordt, moet vaker worden ingezet voor het maken van TO's en iemand die meer 'richting klant' gaat, kan bijvoorbeeld de klant helpen bij configuratiewerkzaamheden danwel het geven van trainingen.

## **Checklist Begroten en plannen**

### **Checklist begroten**

Als je een begroting gaat maken, dan:

- ⇒ gebruik je deze [checklist](#) en deze [template](#)
- ⇒ laat je deze **reviewen** door Arjen of Marjon

### **Checklist doorlooptijdplanning**

Als je een doorlooptijdplanning gaat maken, dan:

- ⇒ gebruik je deze [checklist](#)
- ⇒ laat je deze **reviewen** door Arjen of Marjon

## Checklist begroten

Denk bij het opstellen van een begroting in ieder geval aan de volgende onderdelen.

**NB 1:** de percentages zijn slechts richtlijnen, denk altijd zelf na of er in je project bijzonderheden zijn waardoor afwijkingen kunnen optreden.

**NB 2:** projectmanagement komt in alle fases voor en staat als apart kopje genoemd.

### Inventarisatie:

- Kick-off (intern/extern)
- Opstellen projectplan: 3 dagen
- Interviews: 1 à 1,5 dag per interview
- Doornemen relevante documentatie (bijv. CAO)
- Klant kennis laten maken met HARMONY (introductiecursus?)
- Verwerken van reacties op documenten

### Bouwtraject:

- Opstellen functioneel ontwerp + overleg (intern/extern) hierover: 15% v.d. bouwtijd
- Opstellen technisch ontwerp + overleg hierover: 10% v.d. bouwtijd
- Afhankelijk van detailniveau ontwerp: verder uitwerken ontwerp
- Inrichten ontwikkelomgeving: Delphi, MKS, Database
- Bouw (inclusief opstellen testplan, programmeurstest en oplossen bevindingen uit programmeurstest, systeemtest en review). Globale verdeling: 75% van de tijd reserveren voor bouw + testplan, 25% van de tijd voor oplossen bevindingen. De totale tijd (100%) geldt als basis voor de opslagpercentages die hierna gebruikt worden.
- Code review: 10% v.d. bouwtijd (TPG en GHA/BBA: 10%, ESSO: 6%)
- Systeemtest: 25% v.d. bouwtijd (TPG: 33%, GHA/BBA: 20%, ESSO: 12%)
- Tussentijdse tests door de klant? (incl. voorbereiding, begeleiding, nazorg). Plan deze liefst met minimaal 5 à 6 weken tussenruimte.
- Opstellen documentatie / help: 13% v.d. bouwtijd (TPG: 13%)
- Maken van oplevering(en): 1 dag per oplevering
- Integratietest (hele systeem: voorbereiding, uitvoering, administratieve afhandeling bevindingen): benodigde tijd afhankelijk van de impact van de nieuwe/gewijzigde functionaliteit op bestaande functionaliteit
- Oplossen bevindingen uit integratietest (incl. review en hertest)
- Mergen van functionaliteit naar de hoofdtak
- Onvoorzien (afhankelijk van detailniveau van het FO 20 – 50% bouwtijd). Benoem dit richting de klant niet als expliciete post maar gebruik het als opslagpercentage.
- Niet functionele eisen
- Technische documentatie
- Performance
- Netwerkconfiguratie
- Security eisen
- Safety en recovery eisen
- Eisen aan logging etc.

### Acceptatietest: 25% van de bouwtijd (TPG 23%, ESSO 15%)

- Maken van opleveringen
- Installatie bij de klant
- Eventuele conversies uit vorige versies of andere systemen
- Opleiding van de klant (wellicht niet nodig als er tussentijdse tests zijn geweest)
- Ondersteuning van de klant tijdens het testen
- Analyseren bevindingen van de klant
- Oplossen bevindingen (incl. review/hertest/documentatie)

### Implementatie:

- Maken van opleveringen
- Installatie bij de klant



- Eventuele conversies uit andere systemen
- Opleidingen
- Ondersteuning van de klant tijdens pilot
- Helpdesk en support
- Analyseren bevindingen van de klant
- Oplossen bevindingen (incl. review/hertest/documentatie)
- Onderhoud

**Projectmanagement: 25% v.h. totaal**

- Intern projectmanagement: o.a. inrichten TRS, bevindingendatabase, projman, overleg, aansturing, voortgangsmonitoring: 15% van de *totaal* te besteden tijd (TPG 15%, ESSO 17%)
- Extern projectmanagement: o.a. overleg, beantwoorden van vragen, voortgangsrapportage, vergaderingen bij de klant inclusief voorbereiding en verwerking (houd er rekening mee dat algemeen en technisch projectleider zoveel mogelijk beide aanwezig zijn): 10% van de *totaal* te besteden tijd (TPG 7%, ESSO 8%). Kijk ook naar het aantal te houden vergaderingen, wellicht leidt dit tot een ander bedrag aan uren.

**Algemene reminders:**

- Is duidelijk genoeg wat er precies gebouwd moet worden?
- Kijk naar historische gegevens! (uit TRS/projman)
- Overleg met experts op bepaalde functionele onderdelen.
- Zijn alle punten uit het functioneel ontwerp ook inderdaad begroot?
- Moet er aan de planautomaat gewerkt worden? Zo ja, reserveer dan meer tijd voor zowel de systeemtest als de acceptatietest (ongeveer 2x zoveel als normaal)
- Is er tijd ingeruimd voor kwaliteitsaspecten (bijvoorbeeld algemene opzet van functionaliteit, reduceren van overbodige code)?
- Is het helder welke punten wel en niet in de bouwtijd meegenomen zijn en zijn deze indien niet meegenomen apart begroot (bijv. programmeurstest)?
- Zijn er heldere afspraken over wat binnen projectmanagement valt?
- Is er een heldere met de klant afgesproken meerwerk procedure?
- Is er rekening gehouden met wie het project gaat uitvoeren? Is er bijvoorbeeld extra tijd nodig mensen in te werken?
- Is het FO gereviewed en goedgekeurd (ook door de klant)?
- Is de begroting gereviewed en goedgekeurd?
- Is de checklist voor het maken van een [doorlooptijdplanning](#) gevolgd?

## **Checklist Doorlooptijdplanning**

### ***Aanleiding***

Er is vaak tijdgebrek in een project. Ten gevolge hiervan worden onderdelen van het software-ontwikkelingsproces niet of slecht uitgevoerd wat de kwaliteit en beheersbaarheid van de software niet ten goede komt.

### ***Uitgangspunten***

- In TRS wordt van elk project in behoorlijk detail bijgehouden hoeveel uren er besteed zijn aan een taak ten opzichte van de planning en door wie. Ook taken die niet waren ingepland komen wel in TRS voor met het aantal gerealiseerde uren.
- Er is een realistische tijdsschatting gemaakt voor alle uit te voeren taken voor een gemiddelde ervaren persoon op basis van historische gegevens. Zie hiervoor de [checklist begroten](#).

## ***Het maken van een doorlooptijdplanning***

### ***Bepalen beschikbaarheid medewerkers***

- Zet alle vakanties en cursussen van de medewerkers in de planning. Ook als deze nog niet vaststaan maar redelijkerwijs wel kunnen gaan gebeuren. Plan ze dan maar ergens in. Later kun je het precies maken. Houd ook rekening met feestdagen en schoolvakanties.
- Bepaal voor elke medewerker hoeveel dagen gemiddeld per week deze persoon daadwerkelijk aan het project kan werken. Denk hierbij aan overhead, beschikbaarheid voor andere projecten, vraagbaak voor anderen (intern en extern). Neem hierbij geen incidenten mee, alleen terugkerende activiteiten.
- Bepaal voor elke medewerker hoeveel dagen per maand kunnen uitvallen door onvoorziene omstandigheden. Denk hierbij aan een extra vrije dag, ziekte, verlof, onvoorzien werk zoals demo's geven. Dit zijn incidenten.

### ***Het bepalen van het totaal aantal dagen werk***

Check of er bij het maken van de tijdsschatting aan alle taken is gedacht in het software-ontwikkelingsproces. Zie hiervoor de [Checklist begroten.doc](#)

Verdeel het werk in logische brokken die je los kunt opleveren en een mijlpaal zijn in je planning. Dit zijn grote brokken. Doe voor elk van deze fases het volgende: op basis van historische gegevens:

- Check of er een percentage tijd is ingeruimd voor kwaliteit: reviewen van alle onderdelen van het proces en het aanbrengen van correcties naar aanleiding van de review, het uitvoeren van alle taken volgens afgesproken standaards.
- Check of er een percentage onvoorzien is opgenomen voor uitloop van taken.
- Ruim een percentage tijd in voor het managen van het proces: interne en externe vergaderingen inclusief voorbereidingen en uitvoeren van acties, reistijd, projectadministratie, onderling overleg intern en extern. Realiseer je dat medewerkers ook dit soort tijd nodig hebben.
- Neem een percentage onvoorzien op voor taken die je nu niet voorziet maar ongetwijfeld wel zullen moeten gebeuren.

- Je weet hoeveel dagen werk er moet worden uitgevoerd. Je weet welke medewerkers in je team zitten dus je kent het ervaringsniveau van je team. Pas het aantal dagen werk aan aan het ervaringsniveau van je team. (Stel 100 dagen werk, waarvan er 50 door een minder ervaren persoon worden uitgevoerd die twee keer zo langzaam is dan een gemiddelde persoon, dan moet je dus 150 dagen werk inplannen).

## **Het maken van een grove planning**

Zet de fases globaal in de planning en wijs het werk toe aan medewerkers, rekening houdend met hun beschikbaarheid. Plan medewerkers in voor het percentage dat ze beschikbaar zijn voor het project (dus 100% +/- de tijd nodig voor andere zaken als vergaderingen, overhead, andere projecten, vakantie, ziekte, etc.). Het gaat er hierbij alleen om dat je zeker weet dat het werk in de doorlooptijd die eruit komt, kan worden uitgevoerd. Dus: heb ik op het juiste moment de juiste medewerkers voor het werk? Zet onvoorzien per fase achteraan. Dan hebben we per fase een deadline voor de interne planning (start onvoorzien) en een deadline die we aan de klant kunnen communiceren. Houdt bij kleine projecten minimaal 1 week ruimte tussen het einde van de integratietest en de oplevering aan de klant. Bij grote projecten (meer dan 100 dagen bouw) dient dit minimaal 3 weken te zijn. Indien de klant tussentijdse tests uitvoert, zorg dan dat deze minimaal 5 à 6 weken uit elkaar liggen. Honoreer aanvragen voor extra ad-hoc tests tussendoor niet.

## ***Het maken van een gedetailleerde planning.***

Maak pas een gedetailleerde planning vlak voor je aan een fase gaat beginnen. Verdeel de fase in logische deelfases die je tussendoor kunt opleveren. Zo'n deelfase is ordegrrootte 10 dagen. Doe hetzelfde voor de deelfases als beschreven bij het maken van een grove planning. Zorg dat je onvoorzien netjes verdeelt over de deelfases. Dan weet je wanneer elke deelfase af kan en moet zijn. Maak een gedetailleerde planning voor een deelfase vlak voor de deelfase begint. Let bij het inplannen van een deelfase op het volgende:

- Zet de activiteiten in de volgorde zoals ze ook daadwerkelijk worden uitgevoerd. Doe je dat niet, dan is het moeilijk bewaken of je nog volgens planning werkt.
- Houd rekening met afhankelijkheden tussen de verschillende activiteiten. Je kunt bijvoorbeeld niet gaan testen voordat het onderdeel gebouwd is. Ook tussen verschillende onderdelen die gebouwd moeten worden kunnen volgorde-relaties liggen (bijvoorbeeld: een bepaald rapport kan pas gebouwd worden als een bezoldigingsregel af is). Ook afhankelijkheden tussen projecten onderling kunnen een rol spelen (zoals tussen de vertaalmodule en het bouwen van nieuwe regels).
- Zet elke activiteit waarvan je expliciet wilt bewaken dat die gebeurt, apart in de planning. Dan kan het niet vergeten worden en is er ook tijd voor ingeruimd. Bijvoorbeeld voor de bouw van een onderdeel:
  - detail ontwerp, review ontwerp, aanpassen ontwerp
  - technisch ontwerp, review TO, aanpassen TO
  - bouw, programmeurstest
  - moduletest, review sourcecode, oplossen bugs & verwerken aanmerkingen
- Zet onvoorzien voor deze deelfase als een losse taak aan het einde van de deelfase. De startdatum van deze taak is de deadline van de interne planning voor de deelfase. In dit onvoorzien is rekening gehouden met uitloop op taken en vergeten taken. Was er van een vorige deelfase onvoorzien over, neem deze bij dit

onvoorzien op. Was er in de vorige deelfase teveel onvoorzien gebruikt, haal dit af van dit onvoorzien.

- Plan mensen in volgens het gemiddelde aantal dagen per week. Voor elke medewerker is daarnaast een aantal dagen per maand niet beschikbaar gereserveerd voor het project. Voor zover bekend is wanneer die dagen vallen, plan je ze in waar ze vallen. Voor zover het niet bekend is, laat je ze aan het einde van de maand vallen.

### ***Het bewaken van de doorlooptijdplanning***

- Zorg dat elke medewerker zich heel goed realiseert hoeveel hij/zij beschikbaar is voor het project. Dat wil zeggen:
  - De medewerker kan alleen op vakantie of op cursus als dat in de planning staat
  - De medewerker beseft dat hij echt zeg 4 dagen per week aan het project moet werken en niet minder.
  - De medewerker beseft dat hij voor zeg 5 dagen per maand "Ja" kan zeggen tegen een klus tussendoor of onwerkbaar tijd kan hebben en dan echt "Nee" moet zeggen.
  - De medewerker geeft zijn akkoord aan de beschikbaarheid.
- Zorg dat de medewerker aan het begin van elke week precies weet wat hij moet opleveren in die week (en wanneer als dat relevant is) en hoeveel tijd hij daarvoor heeft.
- Monitor bij onervaren medewerkers elke dag hoe het gaat en zorg dat de medewerker zijn deadlines haalt door te zorgen voor goede instructie, begeleiding, review enz.
- Monitor bij ervaren medewerkers wekelijks hoe het gaat.
- Zorg ervoor dat medewerkers zich aan de afgesproken inspanning houden en dat ze melden dat er een kink in de kabel dreigt te komen. Als er in "onvoorzien" rekening is gehouden met de kink in de kabel, dan hoeft alleen de detailplanning te worden aangepast. Als dat niet zo is, dan moet aan het management worden duidelijk gemaakt dat het project gaat uitlopen als de medewerker van het project wordt gehaald. Als het management daarvoor kiest, pas dan de totale planning aan. Hetzelfde proces dient te worden gevolgd als er extra werk ontstaat wat niet wordt gedekt door "onvoorzien". In alle gevallen moet met de klant worden gecommuniceerd.
- Pas wekelijks de detailplanning van de deelfase aan op basis van de binnengekomen realisaties van medewerkers. Als de planning goed is opgezet heeft een aanpassing van de planning van een deelfase geen invloed op de fase als geheel en zeker geen invloed op andere fasen.

## **Checklists Ontwerpen**

### **Checklist functioneel ontwerp**

Als je een functioneel ontwerp gaat maken, dan:

- ⇒ gebruik je deze [template](#) (klik [hier](#) voor een ingevuld voorbeeld)

- ⇒ laat je dit reviewen door Marjolein of Wessel. Mail een link naar het FO naar [harmonyarchitect@ortec.nl](mailto:harmonyarchitect@ortec.nl) en vermeld in het mailtje om welke klant het gaat en wanneer het FO uiterlijk gereviewed moet zijn.

⇒

## **Checklist technisch ontwerp**

Als je een technisch ontwerp gaat maken, dan:

- ⇒ gebruik je deze [checklist](#)
- ⇒ laat je dit reviewen door Wessel of Peter. Mail een link naar het TO naar [harmonyarchitect@ortec.nl](mailto:harmonyarchitect@ortec.nl) en vermeld in het mailtje om welke klant het gaat en wanneer het TO uiterlijk gereviewed moet zijn.

## **Template Functioneel Ontwerp**

**Titel (zie properties)**

Versie ?? (zie properties)

Datum (zie properties)

Alexn

The logo for ORTEC, featuring the word "ORTEC" in a bold, stylized serif font. The "O" and "R" are dark, while the "T" and "E" are lighter, and the "C" is dark. The letters are slightly shadowed, giving it a 3D appearance.

**Tekenpagina**

**Wijzigingshistorie:**

Versie	Datum	Auteur(s)	Wijzigingen	Goedgekeurd door		
				Datum	Naam	Handtekening
0.1						

## Inhoudsopgave

<b>1</b>	<b>INLEIDING</b>	<b>26</b>
<b>2</b>	<b>REFERENTIES</b>	<b>26</b>
<b>3</b>	<b>PROCESBESCHRIJVING</b>	<b>26</b>
<b>4</b>	<b>FUNCTIES</b>	<b>26</b>
4.1	ALGEMEEN	26
4.2	SCHERMEN	26
4.3	RAPPORTEN	26
4.4	REKENMODULES	27
4.5	INTERFACES MET ANDERE SYSTEMEN	27
4.6	NIET OPGENOMEN PUNTEN	27
<b>5</b>	<b>TECHNIEK</b>	<b>27</b>
5.1	HARDWARE	27
5.2	SOFTWARE	27
5.3	PERFORMANCE	27
5.4	INSTALLATIE	27
<b>6</b>	<b>DOCUMENTATIE</b>	<b>27</b>
<b>7</b>	<b>DEFINITIES</b>	<b>28</b>



## 1 Inleiding

Beschrijf hier de doelen van het project en de projectorganisatie (verwijs naar het projectplan) en geef een aantal tips voor het lezen van het document.

## 2 Referenties

Geef hier een lijst met referenties naar alle gebruikte documenten.

- ☐ Projectplan
- ☐ Requirements analyse

## 3 Procesbeschrijving

Beschrijf hierin het proces waarin de applicatie gebruikt zal gaan worden. Geef daarbij aan welke delen van het proces automatisch ondersteund worden en welke delen van het proces nog handmatig moeten gebeuren (systeemproces -organisatieproces). Gebruik eventueel een aantal 'use-cases' om dit voor de klant duidelijk te maken.

- ☐ Wordt van ieder stap in het proces duidelijk aangegeven of deze in het systeem gedaan wordt?
- ☐ Is van alle punten die in besprekingen naar voren zijn gekomen duidelijk aangegeven of ze wel of niet in het systeem zitten? Loop hiervoor notulen na. Geef ook duidelijk aan wat je niet doet.
- ☐ Overdekt de procesbeschrijving het hele programma van eisen en is van alle eisen aangegeven of het systeem dit wel of niet doet?

Geef aan wat de hoeveelheden data zijn en hoe vaak functies gebruikt worden.

- ☐ Is aangegeven of en hoe multi-user en/of multi-site gebruik mogelijk is?
- ☐ Is de project/afdelingsspecifieke checklist gecontroleerd?
- ☐ Is er gekeken naar functionaliteit in huidige systemen?
- ☐ Is er aangegeven hoe de requirements in dit proces afgedekt zijn.

## 4 Functies

### 4.1 Algemeen

Beschrijf hier de algemene eigenschappen van de applicatie zoals: het aanwezig zijn van menu's en toolbars, gebruik van GUI-standaarden, etc.

### 4.2 Schermen

Beschrijf hier alle schermen die in het systeem komen. Door alle schermen te noemen is het voor de klant duidelijk dat hij schermen die er niet in staan niet krijgt. Geef zo veel mogelijk plaatjes van schermen. Dit geeft de klant een didelijker beeld van wat hij krijgt en hoe hij ermee kan werken. Voorkom dat je te veel in (onnodige details vervalt (in het veld opmerking kan de gebruiker een opmerking invoeren). Neem wel alle controls die niet direct duidelijk zijn.

Let op hoe je omgaat met de geldigheid van data

- ☐ Is het duidelijk hoe foutmeldingen getoond worden?

### 4.3 Rapporten

Geef hier in ieder geval een voorbeeld van de lay-out van het rapport en een omschrijving welke gegevens getoond worden. Geef ook aan hoe de gebruiker kan kiezen wat hij wil zien (parameterschermen)

#### **4.4 Rekenmodules**

Describe of each module what input is needed, what the general functionality of the module is and what output is produced. Describe all limitations and restrictions on the input.

- ☐ Zijn alle eisen en criteria duidelijk?
- ☐ Zijn de randvoorwaarden waaronder een goede planning gemaakt wordt duidelijk?

Neem ook hier performance eisen op

- ☐ Is duidelijk welke gegevens de algoritmiek gebruikt?
- ☐ Geef eisen voor de kwaliteit van de output van het algoritme op.

#### **4.5 Interfaces met andere systemen**

Beschrijf hier de data-interfaces met andere systemen. Beschrijf in ieder geval ook hoe omgegaan wordt met foutafhandeling.

- ☐ In hoeverre zijn er handmatige controles/aanpassingen nodig voor het gebruik van de interface?
- ☐ Is dit niet te technisch? Zet technische details in een bijlage.

Checklist voor heel hoofdstuk 4:

- ☐ Zijn alle fases in het proces die automatisch uitgevoerd moeten worden afgedekt?
- ☐ Kunnen alle entiteiten aangemaakt, gewijzigd verwijderd worden en worden ze ergens gebruikt? (CRUD-Matrix)
- ☐ Staan er geen controls op de schermen waarvan het niet duidelijk is wat ze doen?

#### **4.6 Niet opgenomen punten**

Beschrijf hier alle punten waarover gesproken is maar waarvan besloten is om ze niet op te nemen.

### **5 Techniek**

#### **5.1 Hardware**

Beschrijf hier de benodigde hardware om het systeem te kunnen gebruiken.

#### **5.2 Software**

Beschrijf hier de benodigde third-party software om het systeem te kunnen gebruiken.

#### **5.3 Performance**

Geef hier de eisen die gesteld worden aan performance. Let hierbij op locaties en hardwareeisen. De lijst hoeft niet alle functies te bevatten. Je kunt voor minder gebruikte functies wat algemener blijven. Bijvoorbeeld: Alle functies binnen 2 seconden alleen slepen in planbord in 1 seconden en de planautomaat 15 minuten voor 50 medewerkers voor 4 weken inplannen.

- ☐ Heb je rekening gehouden met de grootte van de data zoals beschreven in de procesbeschrijving?
- ☐ Heb je gekeken welke functies vaak gebruikt worden?

#### **5.4 Installatie**

Beschrijf hier of je installatieprocedures (bijvoorbeeld InstallShield maakt) of dat de verantwoordelijkheid voor installatie bij de klant ligt.

### **6 Documentatie**

Beschrijf de op te leveren documentatie  
- on-line help

- user guide
- installation guide
- technical documentation

Beschrijf ook wat de inhoud van de documenten zal zijn.

Dit overlapt voor een groot deel met het projectplan.

## **7 Definities**

Geef hier een lijst met alle termen die in het FO opgenomen zijn waarvan niet voor iedere lezer (programmeur of klant) duidelijk is wat die term inhoudt.

## **Checklist Technisch Ontwerp**

### **Doel**

- Het is voor de bouwer duidelijk hoe hij functionaliteit moet maken
- Het TO kan later (bijgewerkt) als documentatie dienen

### **Procedure**

- Check of het FO duidelijk en eenduidig is.
- Bekijk wat er al aan technische ontwerpen of documentatie is op het onderdeel waar je mee bezig gaat.
- Schrijf het TO. Gebruik hiervoor de UML module van Visio.
  - Bedenk hoe het moet gaan werken.
  - Werk het proces dmv in Use Cases of in tekst uit.
  - Maak een mapping van entiteiten naar classes. Bijvoorbeeld: een rooster is een TSchedule en een cyclisch rooster een TRotation.
  - Geef de benodigde classes met hun relaties weer in een Class diagram. Gebruik hiervoor UML-Static-Structure in Visio.
  - Werk eventueel moeilijke details uit in een Activity diagram of een Collaboration diagram.
- Check of alle functionaliteit in het FO afgedekt is.
- Sla het TO op in ...

### **Hints**

- Pas het detail van het TO aan aan de ervaring van de bouwer. Een ervaren bouwer heeft minder details nodig.
- Classes in een TO hoeven niet alle classes in Delphi af te dekken. Tijdens de bouw kunnen eventueel details in meerdere classes uitgewerkt worden. Zorg dat je in ieder geval wel de classes tekent die entiteiten in de werkelijkheid weergeven.
- Een tekening van een scherm kan vaak enorm helpen.
- Begin met potlood op een A3 vel. Dit tekent veel makkelijker.
- Over het gebruik van de diverse diagrammen kun je meer informatie vinden in de bij Visio meegeleverde .pdf files. Er zijn ook een aantal goede boeken over.

## Checklists Bouwen

### Bouw, database, review, test, bugs oplossen en mergen Checklist bouwen

Als je een nieuw onderdeel of scherm bouwt, dan:

- ⇒ maak je een **ontwerp** (of gebruik je een door een ander gemaakt ontwerp)
- ⇒ bespreek je dit ontwerp met iemand anders (of met de ontwerper als je het niet zelf gemaakt hebt) Check sowieso of het ontwerp volledig is.
- ⇒ maak je zelf een **inschatting** van de benodigde bouw- en doorlooptijd en communiceer je dit met de projectleider (met name als dit afwijkt van de begroting). Reserveer ongeveer 75% voor het bouwen en het maken van een testplan en de rest voor het oplossen van bevindingen. Werk zo naar een deadline toe. Houd ook tijdens het bouwen in de gaten of er afwijkingen ontstaan en meld dit z.s.m.
- ⇒ gebruik je deze [handleiding](#) voor het werken in **zijtakken** en **mergen**
- ⇒ schrijf je code volgens deze **coding standaard** en onze eigen [uitbreidingen](#) daarop
- ⇒ gebruik je dit [template](#) voor het invoeren van **commentaar**
- ⇒ gebruik je voor een **scherm** deze [checklist](#) (denk aan [locking](#) en autorisaties!)
- ⇒ laat je het scherm (GUI) **reviewen** door Marjolein
- ⇒ gebruik je voor een **serverobject** deze [checklist](#)
- ⇒ gebruik je voor het toevoegen van **events** deze [checklist](#)
- ⇒ gebruik je voor een **requirement** deze [checklist](#)
- ⇒ maak je een eerste aanzet voor het **testplan** en zet dit op [\\Gdvl1\Artem\Projects\Harmony\Documentatie\Testing\Testplannen](#)
- ⇒ doorloop je dit testplan (gebruik een realistische dataset)
- ⇒ zorg je dat je code **gereviewed** wordt
- ⇒ zorg je dat je code **getest** wordt (met behulp van het testplan)

### Checklist database

Als je een verandering aanbrengt aan de database, dan:

- ⇒ gebruik je deze [checklist](#) voor de **scripts**
- ⇒ zet je de scripts op [\\Gdvl1\Artem\Projects\scripts\Update\Test](#) (dit geldt voor elk database platform) om ze door Erwin te laten reviewen en te laten vertalen naar andere database systemen (Oracle/SQL Server/Interbase/Solid, ...)
- ⇒ gebruik je deze [handleiding](#) voor de verwerking in **MKS**
- ⇒ klik [hier](#) voor een beschrijving van het **importeren** en **exporteren** van een Oracle user
- ⇒ klik [hier](#) voor een beschrijving van het **importeren** en **exporteren** van een SQL Server backup
- ⇒ klik [hier](#) voor een tip over het programmeren van **Select queries** in Delphi
- ⇒ klik [hier](#) voor een beschrijving van het **vergroten** van een Oracle **datafile**
- ⇒ klik [hier](#) voor een beschrijving van het **bijwerken** van een Oracle **sequence**
- ⇒ klik [hier](#) voor een beschrijving van een **conversie** van **Oracle** naar **SQL Server**
- ⇒ klik [hier](#) voor een beschrijving van een **conversie** van **Oracle** naar **Interbase 6**

### Checklist reviewen

Als je iets gaat reviewen, dan:

- ⇒ gebruik je deze [checklist](#)
- ⇒ controleer je of er een **testplan** is gemaakt en review je dit
- ⇒ **koppel** je je commentaar **terug** naar de bouwer
- ⇒ **controleer** je achteraf of je commentaar is doorgevoerd

## Checklist testen

Als je iets gaat testen, dan:

- ⇒ controleer je of er een **testplan** is
- ⇒ breid je dit testplan verder uit en gebruik je dit (gebruik een realistische dataset!)
- ⇒ gebruik je deze [checklist](#)
- ⇒ als je een test uitvoert gerelateerd aan **bezoldigingsregels** en de resultaten daarvan (bijvoorbeeld **overwerk**), gebruik daarnaast dan ook deze [checklist](#)
- ⇒ als je reviewfouten tegenkomt, **koppel** dit dan **terug** naar de reviewer

## Checklist bugs oplossen

Als je een bug oplost

- ⇒ gebruik je deze [checklist](#)
- ⇒ zorg je dat de bug **gereviewed** wordt
- ⇒ zorg je dat de bug **geheretest** wordt (bij voorkeur door de melder)

## Checklist mergen

Als je iets gaat mergen

- gebruik je deze [handleiding](#) voor het werken in **zijtakken**
- zorg je dat de aangepaste bestanden **gereviewed** worden
- zorg je dat de relevante onderdelen **geheretest** worden

## Checklist Testen

### Doelen

- Effectief testen: veel fouten vinden
- Efficiënt testen: geen tijd verspillen
- Herhaalbaarheid: een ander kan de test snel nog een keer uitvoeren

### Procedure

- Bepaal wat je gaat testen: Zorg dat dit goed afgebakend is, en check dat dit nog niet door een ander gedaan is. Test onderdelen één voor één. Als er 3 nieuwe verlofregels zijn loop je deze procedure 3 keer door.
- Bepaal hoe goed het getest moet worden: functionaliteit die niet veranderd is kan je even snel doorlopen; een lastige nieuwe regel zal je uitgebreid moeten testen.
- Schrijf de hoofdonderdelen van je test op. Dit kunnen onderdelen van een scherm zijn of stukken van een regel. Bijvoorbeeld: voor het tabblad roostergegevens in het medewerker scherm 'Indeling teams' en 'Indeling categorieën'. Voor de eis dagelijkse rust de onderdelen 'Gewone rust', 'Gebruik uitzonderingen' en 'Omgaan met tussenliggende activiteiten die niet meetellen'.
- Schrijf per onderdeel de gebruiksmogelijkheden op. Doe dit kort, maar probeer wel zo veel mogelijk gevallen te onderscheiden. Probeer hier alles op te nemen wat de gebruiker kan doen. Bijvoorbeeld: maak een nieuwe regel aan; verander de begintijd; verander de eindtijd; gebruikersvriendelijkheid.
- Schrijf nu per gebruiksmogelijkheid de testgevallen op. Probeer hier zo veel mogelijk moeilijke situaties te vinden. Bijvoorbeeld: verander de begintijd tot voor de huidige; verander de begintijd tot na de huidige; verander de begintijd tot na de eindtijd, etc.
- Bepaal voor gevallen waarvoor het niet evident is wat de uitkomst moet zijn de gewenste uitkomst.
- Voer één voor één de testgevallen uit.
- Loop hierna ook een keer het proces door. Let daarbij vooral op gebruikersvriendelijkheid en of alle onderdelen uit het FO afgedekt worden.
- Noteer de gevonden fouten in de bugdatabase. Check hierbij wel eerst of het echt fouten zijn en geen fouten in de uitgevoerde test. Probeer fouten te reproduceren en schrijf er bij hoe je dat kan doen.
- Zorg zo veel mogelijk dat je testomgeving bewaard blijft, zodat de test makkelijk herhaald kan worden.
- Sla je testplan wat je zo geschreven hebt onder k:\componentendoos\documentatie\testing\Testplannen in de juiste directory op.

### Eindtest

- Doe voordat een nieuwe oplevering gedaan wordt, altijd een multi-user test en een stress-test. Hierover is een apart document beschikbaar (of verschijnt binnenkort).

## Hints

- Laat je niet te veel door het proces leiden. Fouten zitten in het algemeen niet in het goed doorlopen van het proces, maar treden op als een gebruiker iets verkeerd doet. Dit moet of onmogelijk zijn, of goed afgevangen worden.
- Los geen fouten op tijdens het testen. Leg bevindingen zo veel mogelijk terug bij de programmeur.

## Doelen

- Effectief testen: veel fouten vinden
- Efficiënt testen: geen tijd verspillen
- Herhaalbaarheid: een ander kan de test snel nog een keer uitvoeren

## Procedure

- Bepaal wat je gaat testen: Zorg dat dit goed afgebakend is, en check dat dit nog niet door een ander gedaan is. Test onderdelen één voor één. Als er 3 nieuwe verlofregels zijn loop je deze procedure 3 keer door.
- Bepaal hoe goed het getest moet worden: functionaliteit die niet veranderd is kan je even snel doorlopen; een lastige nieuwe regel zal je uitgebreid moeten testen.
- Schrijf de hoofdonderdelen van je test op. Dit kunnen onderdelen van een scherm zijn of stukken van een regel. Bijvoorbeeld: voor het tabblad roostergegevens in het medewerker scherm 'Indeling teams' en 'Indeling categorieën'. Voor de eis dagelijkse rust de onderdelen 'Gewone rust', 'Gebruik uitzonderingen' en 'Omgaan met tussenliggende activiteiten die niet meetellen'.
- Schrijf per onderdeel de gebruiksmogelijkheden op. Doe dit kort, maar probeer wel zo veel mogelijk gevallen te onderscheiden. Probeer hier alles op te nemen wat de gebruiker kan doen. Bijvoorbeeld: maak een nieuwe regel aan; verander de begintijd; verander de eindtijd; gebruikersvriendelijkheid.
- Schrijf nu per gebruiksmogelijkheid de testgevallen op. Probeer hier zo veel mogelijk moeilijke situaties te vinden. Bijvoorbeeld: verander de begintijd tot voor de huidige; verander de begintijd tot na de huidige; verander de begintijd tot na de eindtijd, etc.
- Bepaal voor gevallen waarvoor het niet evident is wat de uitkomst moet zijn de gewenste uitkomst.
- Voer één voor één de testgevallen uit.
- Loop hierna ook een keer het proces door. Let daarbij vooral op gebruiksvriendelijkheid en of alle onderdelen uit het FO afgedekt worden.
- Noteer de gevonden fouten in de bugdatabase. Check hierbij wel eerst of het echt fouten zijn en geen fouten in de uitgevoerde test. Probeer fouten te reproduceren en schrijf er bij hoe je dat kan doen.
- Zorg zo veel mogelijk dat je testomgeving bewaard blijft, zodat de test makkelijk herhaald kan worden.
- Sla je testplan wat je zo geschreven hebt onder  
k:\componentendoos\documentatie\testing\Testplannen in de juiste directory op.



## **Eindtest**

- Doe voordat een nieuwe oplevering gedaan wordt, altijd een multi-user test en een stress-test. Hierover is een apart document beschikbaar (of verschijnt binnenkort).

## **Hints**

- Laat je niet te veel door het proces leiden. Fouten zitten in het algemeen niet in het goed doorlopen van het proces, maar treden op als een gebruiker iets verkeerd doet. Dit moet of onmogelijk zijn, of goed afgevangen worden.
- Los geen fouten op tijdens het testen. Leg bevindingen zo veel mogelijk terug bij de programmeur.

## Checklist Opleveren

### Checklist releasebuild

Als je een releaseversie van de software maakt

- ⇒ gebruik je deze [checklist](#) (en [deze](#) voor de LNVL)
- ⇒

### Checklist oplevering/installatie

Als je een oplevering en/of installatie doet voor een klant

- ⇒ gebruik je voor de **administratieve afhandeling** van de oplevering deze [checklist](#)
- ⇒ gebruik je voor de **installatie** deze [handleiding](#) en deze [checklist](#) (inclusief installatieverslag)
- ⇒ **test** je altijd de op te leveren versie met de database van de klant
- ⇒ klik [hier](#) om veelvoorkomende **problemen** bij het **installeren** van HARMONY op te lossen

klik [hier](#) voor een beschrijving van het **automatisch stoppen en (her)starten** van een **service** (bijvoorbeeld voor het maken van een back-up)

## **Externe bijlage V      Functioneel Ontwerp**

# **Functioneel Ontwerp HARMONY-WebAccess**

Versie 1.0

Alex Nap

## Tekenpagina

### Wijzigingshistorie:

Versi e	Datum	Auteur(s)	Wijzigingen	Goedgekeurd door		
				Datum	Naam	Handteken ing

# Inhoudsopgave

## 1. INLEIDING 40

## 2. REFERENTIES 42

## 3. RESULTAAT BRAINSTORMSESSIE 42

- 3.1 GEWENSTE FUNCTIES 42
- 3.2 FUNCTIONELE EISEN 42
- 3.3 NIET-FUNCTIONELE EISEN 43

## 4. BESCHRIJVING HUIDIG HARMONY-WEBACCESS 44

- 4.1 ALGEMENE FUNCTIES 44
  - 4.1.1 *Uitloggen* 44
  - 4.1.2 *Administrator e-mailen* 44
  - 4.1.3 *Menu* 44
- 4.2 INLOGGEN 44
- 4.3 PERSOONLIJK 45
- 4.4 ROOSTER 46
  - 4.4.1 *Menu Rooster – Mijn rooster* 46
  - 4.4.2 *Menu Rooster – Per groep* 47
  - 4.4.3 *Menu Rooster – Groep per week* 47
- 4.5 KAARTEN 47
- 4.6 WENSEN 48
- 4.7 RUILING AANVRAGEN 48
- 4.8 ZOEKEN 50
- 4.9 LINKS 51

## 5. SCHEMATISCH OVERZICHT HUIDIG HARMONY-WEBACCESS 52

## 6. BESCHRIJVING TOEKOMSTIG HARMONY-WEBACCESS 53

- 6.1 FUNCTIONALITEIT IN DETAIL 53
  - 6.1.1 *Wachtwoord wijzigen* 53
  - 6.1.2 *Uren verantwoorden* 54
  - 6.1.3 *Overzicht mutaties + gezien* 56
  - 6.1.4 *Rooster gezien* 56
  - 6.1.5 *Wensen + verlofaanvragen bekijken* 57
- 6.2 ACTIES GEBRUIKER MET HET SYSTEEM 57
- 6.3 LAY-OUT SCHERMEN 58
  - 6.3.1 *Scherm Wachtwoord wijzigen* 58
  - 6.3.2 *Scherm Uren verantwoorden* 58
  - 6.3.3 *Scherm Overzicht mutaties* 59
  - 6.3.4 *Scherm Rooster gezien* 60
  - 6.3.5 *Scherm Wensen* 61
- 6.4 SCHEMATISCH OVERZICHT TOEKOMSTIG HARMONY-WEBACCESS 63
- 6.5 NIET OPGENOMEN PUNTEN 64

## 7. CLASS DIAGRAM 65

## 8. TECHNIEK 67

- 8.1 HARDWARE 67
- 8.2 SOFTWARE 67
- 8.3 PERFORMANCE 67

8.4 INSTALLATIE 67

**9. DOCUMENTATIE 67**

## 1. Inleiding

Dit document (Functioneel Ontwerp) dient als basis voor de ontwikkeling van de functionaliteit voor HARMONY-WebAccess. In dit project zullen alleen de functies ontwikkeld worden die in dit document worden beschreven. Mocht er gedurende de ontwikkeling besloten worden om toch andere functies eraan toe te voegen dan moeten hier grondige redenen voor zijn en moet men goed beseffen wat de gevolgen hiervan zijn voor de voortgang van het project. De gevolgen kunnen zijn dat ingeplande activiteiten niet uitgevoerd worden en daardoor uitloop van het project veroorzaken.

De referenties (Hoofdstuk 2) die gebruikt zijn voor het opstellen van dit document zijn: het Plan van Aanpak, de resultaten van de brainstormsessie, het Functioneel Ontwerp van J. de Jong en de Gebruikersdocumentatie van HARMONY. De resultaten van de brainstormsessie worden

beschreven in hoofdstuk 3. In dit hoofdstuk staan de gewenste functies met de prioriteit van ontwikkeling vermeld. Verder staat er een overzicht van de niet-functionele eisen.

Om een beeld te krijgen van hoe het huidige HARMONY-WebAccess eruitziet, heb ik van alle schermen die in de huidige HARMONY-WebAccess staan beschreven wat de functie ervan is. De informatie hiervoor komt voor een deel uit de informatie van J. de Jong en is voor een deel naar eigen inzicht ingevuld (zie hoofdstuk 4).

In hoofdstuk 5 wordt van de in hoofdstuk 4 beschreven schermen een schematisch overzicht gegeven.

In hoofdstuk 6 worden de functies, zoals vermeld in hoofdstuk 3, in detail uitgewerkt. Per functies wordt beschreven wat de functies moet doen en hoe deze functies op het scherm getoond worden. Het overzicht dat getoond wordt in hoofdstuk 5 is in dit hoofdstuk weer te zien, alleen zijn daar de nieuwe functies aan toegevoegd. Verder worden in dit hoofdstuk ook de punten beschreven die tijdens de brainstormsessie naar voren zijn gekomen, maar die niet in dit project meegenomen zullen worden.

In hoofdstuk 8 staat een beschrijving van de techniek die nodig is om HARMONY-WebAccess te gebruiken. De punten die aan de orde komen zijn de benodigde hardware en software, de performance eisen en hoe de installatie moet geschieden.

In hoofdstuk **Fout! Verwijzingsbron niet gevonden.** staat de uitleg van moeilijke / onbekende termen die in dit document gebruikt worden.

### Leestips:

#### Medewerker (hij / zij)

Er wordt regelmatig over de medewerker in de mannelijke vorm (hij) gesproken. Dit is ook van toepassing op de vrouwelijke vorm (zij).

### Menu

---

Als er gesproken wordt over het menu dan wordt daar mee bedoeld het menu in HARMONY-WebAccess dat aan de rechterkant van het scherm getoond wordt.

---



## 2. Referenties

Hieronder staan de gebruikte documenten:

- Plan van Aanpak.
- Resultaat brainstormsessie.
- Functioneel Ontwerp van J. de Jong.
- Gebruikersdocumentatie HARMONY.

## 3. Resultaat brainstormsessie

Om te kunnen bepalen welke functies toegevoegd moeten worden aan HARMONY-WebAccess is er voor gekozen om een brainstormsessie te houden. In dit hoofdstuk zullen de resultaten van deze brainstormsessie getoond worden.

### 3.1 Gewenste functies

In de onderstaande tabel staan de functies die toegevoegd moeten worden aan HARMONY-WebAccess.

	Functionaliteit	Prioriteit
A	Urenverantwoording invoeren	1
B	Overzicht van ruilingen / mutaties en aangeven of hij ze gezien heeft	1
C	Medewerker moet aan kunnen geven dat hij zijn planning gezien heeft (zodat planner medewerker niet hoeft te bellen om te vragen of medewerker zijn planning heeft gezien)	2
D	Bekijken van wensen + verlofaanvragen	2
E	Wachtwoord wijzigen	3

#### Legenda:

prioriteit 1 = Must-have;  
prioriteit 2 = Should-have;  
prioriteit 3 = Could-have;  
prioriteit 4 = Want-to-have.

### 3.2 Functionele eisen

In deze paragraaf worden de functionele eisen beschreven. De functionele eisen staan op volgorde van prioriteit.

#### Prioriteit 1:

Medewerker moet:

- A1: in kunnen voeren wat zijn daadwerkelijke uren zijn.
- A2: activiteiten die hij niet heeft uitgevoerd kunnen verwijderen.
- A3: activiteiten kunnen toevoegen.
- A4: activiteiten kunnen wijzigen.
- A5: dienst kunnen goedkeuren, nadat medewerker deze gedaan heeft (daarbij verschil tussen ongewijzigde diensten en gewijzigde diensten)
- B1: de mutaties / ruiling(en) die voor hem van toepassing zijn kunnen zien.
- B2: aan kunnen geven mutaties / ruiling(en) gezien.

#### Prioriteit 2:

Medewerker kan:

- C: aangeven dat hij zijn rooster gezien heeft.
- D1: zien welke wensen hij heeft ingevoerd;
- D2: zien wat voor verlofaanvragen hij heeft ingediend.

#### Prioriteit 3:

- E: Medewerker zou zijn wachtwoord moeten kunnen wijzigen.
-

### **3.3 Niet-functionele eisen**

In deze paragraaf worden de randvoorwaarden waaraan HARMONY-WebAccess moet voldoen beschreven. De niet-functionele eisen zijn:

De applicatie moet...

- zelfde autorisatiegegevens gebruiken als die in de HARMONY-client;
  - integreerbaar zijn in intranetten / extranetten;
  - te beveiligen zijn op het gebied van rechten en beveiliging van datastromen;
  - geschikt zijn voor de doelgroep medewerkers;
  - aangepast kunnen worden aan de look-and-feel van de organisatie;
  - de look-and-feel van huidige HARMONY-client hebben;
  - een browserformaat van 800x600 ondersteunen;
  - functioneren op Microsoft Internet Explorer 5+;
  - robuust zijn, dwz: mag geen andere processen in gevaar brengen;
  - onderhouden kunnen worden door de afdeling.
-

## 4. Beschrijving huidig HARMONY-WebAccess

Om een beeld te krijgen van de huidige functies wordt hieronder globaal beschreven welke functies er zijn, wat de functies doen en voor wie deze functies bestemd zijn.

### 4.1 Algemene functies

Met de algemene functies wordt bedoeld dat ze altijd te gebruiken zijn ongeacht welke pagina de medewerker heeft geopend. Het gaat om de functionaliteit 'Uitloggen' en contact opnemen met de administrator.

#### 4.1.1 Uitloggen

Boven aan elk scherm is het volgende te zien:



Bovenbalk

Aan de linkerkant staat de achternaam van de ingelogde medewerker. In het midden van de pagina is de link 'Logout' te vinden. Door hier op te klikken kom je terug in het inlogscherf.

#### 4.1.2 Administrator e-mailen

Onder elk scherm is het volgende te zien:

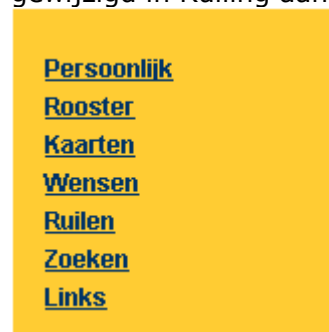


Onderbalk

Aan de linkerkant staat wat je kunt doen als je problemen hebt met de applicatie. Door op het woord 'Administrator' te klikken heb je de mogelijkheid om een e-mail bericht te versturen.

#### 4.1.3 Menu

Hieronder staat het menu zoals dat getoond wordt aan de rechterkant van elk scherm. Vanuit dit menu kan de gebruiker (=medewerker) zijn keuze maken. De link Ruilen is gewijzigd in Ruiling aanvragen.



Menu

## 4.2 Inloggen

Bij het openen van de hoofdpagina van de applicatie zal de medewerker eerst moeten inloggen. Dit kan gebeuren met behulp van een gebruikersnaam en een wachtwoord. Deze gebruikersnaam en dit wachtwoord kan de medewerker verkrijgen bij de planner. Wanneer de medewerker een onjuiste combinatie van gebruikersnaam en wachtwoord invoert, krijgt deze daar een foutmelding over. Daarna kan door de medewerker wederom getracht worden om in te loggen. Wanneer de medewerker wel succesvol inlogt komt hij op de hoofdpagina. De hoofdpagina is gelijk aan de pagina zoals getoond onder de link 'Persoonlijk'. Voor uitleg over deze pagina zie volgende paragraaf Persoonlijk.

## 4.3 Persoonlijk

Gebruiker: **Elfring**  
Welkom bij uw rooster online

[Logout](#)

**HARMONY-Online**

**Personalia:**

Voorvoegsel:  Geboortedatum:   
Naam: **Elfring** Personeelsnummer:   
Voorletters:  Locker:

**Adres:**

**Bereikbaarheid:**

Straat:  Telefoon:   
Postcode:  Mobiel:   
Woonplaats:  E-mail:

**Mededelingen:**

**Bij u aangevraagde ruilingen:**

Aanvrager	Dienst	Uw dienst	Datum dienst	Status	Actie

**Door u aangevraagde ruilingen:**

Medewerker	Uw dienst	Dienst	Datum dienst	Status	Actie
Jong, J.W.L.de	ALL	AT	10.05.2003	Geaccepteerd	

Problemen? Neem contact op met de beheerder: Administrator

Vrijdag 12 september 2003, 14:13:21

Hoofdscherm / Scherm Persoonlijk

In dit scherm worden de persoonlijke gegevens van de ingelogde persoon getoond. Verder wordt een overzicht gegeven van ruilingen die een collega met de ingelogde persoon wil doen en de ruilingen die door de ingelogde persoon zijn aangevraagd met een bepaalde collega. Het is de bedoeling dat het overzicht van ruilaanvragen in de toekomstige situatie onder de link 'Ruiling aanvragen' zal worden getoond.

In het overzicht: "Bij u aangevraagde ruilingen:" worden de volgende velden getoond:

- Aanvrager. Naam van de collega;
- Dienst. De naam van de dienst waar het om gaat;
- Uw dienst. De naam van de dienst die de ingelogde persoon dan heeft;
- Datum dienst. De datum waarop de dienst gepland staat;
- Status. Is 'Open', 'Geaccepteerd' of 'Geweigerd';
- Actie. Is 'Accepteer' of 'Weiger'.

In het overzicht: "Door u aangevraagde ruilingen:" worden de volgende velden getoond:

- Medewerker. Naam van de ingelogde persoon;
- Uw dienst. De naam van de dienst waar het om gaat;
- Dienst. De naam van de dienst waarvoor de collega voor ingepland is;
- Datum dienst. De datum waarop de dienst gepland staat;
- Status. Is 'Open', 'Geaccepteerd' of 'Geweigerd';
- Actie. Is 'Verwijderen'.

### Ruiling accepteren

Als de ingelogde persoon op 'Accepteer' klikt, wordt de dienst toegevoegd in zijn rooster en wordt deze dienst verwijderd uit het rooster van de aanvrager. De status wordt van 'Open' gewijzigd in 'Geaccepteerd'.

## Ruiling weigeren

Als de ingelogde persoon op 'Weiger' klikt, wordt er niets veranderd in de dienstroosters van de betrokken personen. De status wordt van 'Open' gewijzigd in 'Geweigerd'.

## Ruiling verwijderen

De ingelogde persoon heeft de mogelijkheid om een ruiling die hij heeft aangevraagd te verwijderen. Dit verwijderen kan alleen als de status van de aanvraag 'Open' is. Dit komt omdat zolang de status open is er geen wijzigingen gedaan worden in de roosters.

### 4.4 Rooster

Als op de link 'Rooster' geklikt wordt zijn er de volgende keuze-opties te zien: Mijn rooster, Per groep en Groep per week. Per optie zal uitgelegd worden wat de mogelijkheden ervan zijn.

#### 4.4.1 Menu Rooster – Mijn rooster

Bij het selecteren van 'Rooster – Mijn rooster' krijgt de medewerker het eigen rooster te zien voor de komende weken. Door in het onderdeel "Periode selecteren" een andere periode te selecteren en deze door te voeren door op "Periode aanpassen" te klikken, zal een andere roosterperiode worden weergegeven.

Boven aan de pagina staat het huidige weeknummer, de functie van de pagina en de naam van de medewerker.

Wanneer geen periode wordt ingevoerd, of wanneer de medewerker op de pagina komt zal de periode vanaf de huidige datum worden getoond. Wanneer de medewerker een periode selecteert die groter is dan één jaar zal de periode worden beperkt tot dit ene jaar.

De, op basis van de geselecteerde begin- en einddatum, weergegeven periode toont de diensten per week. Wanneer de medewerker de muis over een dienst heen beweegt, en op deze dienst blijft staan, zal een hinttekst verschijnen. Deze hinttekst bevat de begintijd en eindtijd van een dienst, eventueel de dienst waarmee geruild is en eventueel een opmerking bij de dienst. Door met de muis over een weeknummer heen te bewegen, en op deze week te blijven staan, zal er ook een hinttekst verschijnen. Deze hinttekst geeft de begin- en einddatum van de specifieke week (van maandag tot en met zondag).

Deze week is het: Week 38		Persoonlijk rooster - Elfring, C					
<div><b>Periode selecteren</b> Van: 22-09-2003  Tot: 15-10-2003  <b>Periode aanpassen</b></div>							
	maandag	dinsdag	woensdag	donderdag	vrijdag	zaterdag	zondag
week 39							
week 40							
week 41							
week 42							

Mijn rooster

#### **4.4.2      Menu Rooster – Per groep**

Boven aan de pagina staat het huidige weeknummer, de functie van de pagina, de naam van de roostergroep en de naam van de ingelogde persoon.

Bij het selecteren van 'Rooster – Per groep' krijgt de medewerker een overzicht van alle werknemers binnen zijn eigen roostergroep te zien. Door op de naam van een werknemer te klikken krijgt de medewerker het rooster van de desbetreffende werknemer te zien. Door in het onderdeel "Periode selecteren" een andere periode te selecteren en deze door te voeren door op "Periode aanpassen" te klikken, zal een andere roosterperiode worden weergegeven.

Wanneer geen periode wordt ingevoerd, of wanneer de medewerker op de pagina komt zal de periode vanaf de huidige datum worden getoond. Wanneer de medewerker een periode selecteert die groter is dan één jaar zal de periode worden beperkt tot dit ene jaar.

De, op basis van de geselecteerde begin- en einddatum, weergegeven periode toont de diensten per week. Wanneer de medewerker de muis over een dienst heen beweegt, en op deze dienst blijft staan, zal een hinttekst verschijnen. Deze hinttekst bevat de begintijd en eindtijd van een dienst, eventueel de dienst waarmee geruild is en eventueel een opmerking bij de dienst. Door met de muis over een weeknummer heen te bewegen, en op deze week te blijven staan, zal er ook een hinttekst verschijnen. Deze hinttekst geeft de begin- en einddatum van de specifieke week (van maandag tot en met zondag).

Met deze gegevens is het voor de medewerker mogelijk om te zien met welke collega hij eventueel zou kunnen ruilen.

#### **4.4.3      Menu Rooster – Groep per week**

Boven aan de pagina staat het huidige weeknummer, de functie van de pagina en de naam van de ingelogde persoon.

Bij het selecteren van 'Rooster – Groep per week' krijgt de medewerker per week de planning van alle werknemers binnen zijn roostergroep te zien. Door in het onderdeel "Periode selecteren" een andere periode te selecteren en deze door te voeren door op "Periode aanpassen" te klikken, zal een andere roosterperiode worden weergegeven.

Met deze gegevens is het voor de medewerker mogelijk om te zien met welke collega hij eventueel zou kunnen ruilen.

#### **4.5      Kaarten**

Boven aan de pagina staat het huidige weeknummer, de soort pagina en de naam van de medewerker.

---

Deze week is het: <b>Week 38</b>				Vakantie - Elfring, C		
----------------------------------	--	--	--	-----------------------	--	--

Periode selecteren

**Van:**

**Tot:**

[Periode aanpassen](#)

Rubriek	Boeking	Referentie	Uren	%	Waarde	Opmerkingen
Saldo	01-01-2003	(Geen)	1337:45	8,8392	118:15	
Saldo	21-04-2003	V	09:00	-100	-09:00	
<b>Totaal</b>			<b>1346:45</b>		<b>109:15</b>	

Kaart Vakantie

Bij het selecteren van het onderdeel kaarten wordt, afhankelijk van de geselecteerde kaart, de kaartinformatie getoond. De medewerker kan een overzicht krijgen van alle op een kaart geboekte onderdelen. Per rubriek wordt aangegeven hoeveel toeslagen de medewerker heeft verdiend. Door gebruik te maken van de "Periode Selectie" kan de medewerker zelf de periode selecteren waarvan hij het overzicht wilt hebben. Zo kunnen bijvoorbeeld alle toeslagen die een medewerker in de afgelopen maand heeft gehad bekeken worden.

## 4.6 Wensen

Onder deze link heeft de medewerker de mogelijkheid om zijn wensen te kennen te geven. Het scherm ziet er als volgt uit:

**Wensen:**

**Op deze pagina kunt u uw wensen opgeven:**

Indien u er zeker van wilt zijn dat uw wens gehonoreerd wordt, raden we u aan contact op te nemen met uw roosteraar.

**Kies hieronder het type wens dat u in wilt voeren.**

[Wel/niet werken in een periode](#)

[Een dienstwens opgeven](#)

[Verlofwens](#)

Huidig scherm Wensen

## 4.7 Ruiling aanvragen

In dit scherm kan de medewerker aangeven dat hij een dienst wil ruilen.

Boven aan de pagina staat het huidige weeknummer, de functie van de pagina en de naam van de medewerker.

---

Deze week is het: Week 37

Ruilen van een dienst: - Elfring

Een dienst ruilen op:

datum: 07-04-2003 

Zoek de dienst

Geen dienst

U heeft op deze dag geen dienst

Scherf Geen dienst

Door op het kalender-icoontje te klikken kan de medewerker een datum selecteren. Door op de button 'Zoek de dienst' te klikken worden de diensten op de desbetreffende dag getoond. De meldingen die kunnen verschijnen zijn:

- Geen dienst (U heeft op deze dag geen dienst);
- Vastgezet (Kan niet ruilen, want dienst is vastgezet. kan namelijk alleen bij status 'Gepubliceerd').

Of er wordt een lijst getoond van medewerkers waarmee de medewerker kan ruilen.

Hieronder is een voorbeeld van een ruiling te zien. Er is een datum geselecteerd waarop medewerker Elfring dienst hoort te hebben. In het scherm worden de namen getoond van collega's waarmee geruild kan worden.

Door op de naam van de desbetreffende medewerker te klikken, wordt de aanvraag ingediend.

Als het om een geldige aanvraag gaat, wordt de volgende melding getoond: "Uw ruilaanvraag voor dienst *dienstnaam* met *naam collega* is ingevoerd."

De planner ziet in zijn planbord door middel van kleuren of een dienst gewijzigd of geruild is. Als de status gewijzigd is, is de naam van de dienst blauw gekleurd. Als de dienst geruild is dan is de kleur rood. Door de dienst aan te klikken kan de planner zien met wie de persoon een dienst heeft geruild.

---



Deze week is het: Week 39

Ruilen van een dienst: - Elfring, C

Een dienst ruilen op:

datum: 18-09-2003

Zoek de dienst

U kunt hieronder een ruiling aanvragen voor dienst Dienst1 op 18-09-2003.

Hieronder ziet u een lijst met medewerkers die voor uw ruilaanvraag in aanmerking kunnen komen. Een ruiling met personen in deze lijst zal geen CAO of aan andere regelgeving schendingen veroorzaken.

Click op een van deze medewerkers om uw ruilaanvraag in te voeren.

Naam:	Telefoon:	Dienstnaam:
<a href="#">Jong, J.W.J. de</a>		
<a href="#">Gabriels</a>		

Voorbeeld van ruilaanvraag

## 4.8 Zoeken

In dit menu is het mogelijk om te zoeken op een bepaalde dienst per dag. Je krijgt dan een overzicht van alle personen die voor de geselecteerde dienst op de geselecteerde datum dienst hebben.

Boven aan de pagina staat het huidige weeknummer, de functie van de pagina en de naam van de medewerker.

Deze week is het: Week 37

Zoeken - Elfring

Zoek naar dienst op bepaalde dag:

Dienst: AT

Datum: 21-04-2003

Zoek de persoon

Naam:	Dienstnaam
Jong, J.W.J. de	AT

## Scherm Zoeken

In het veld 'Dienst:' kan je een dienst selecteren.

In het veld 'Datum:' kan je door het kalender-icoontje te klikken de datum selecteren van wanneer je de desbetreffende dienst wilt zien.

Als er diensten zijn behorend bij de selectie dan worden de namen van de personen die ingeroosterd zijn getoond met daarachter de dienstnaam.

Als er geen resultaten zijn dan wordt de melding "Geen medewerkers gevonden" getoond.

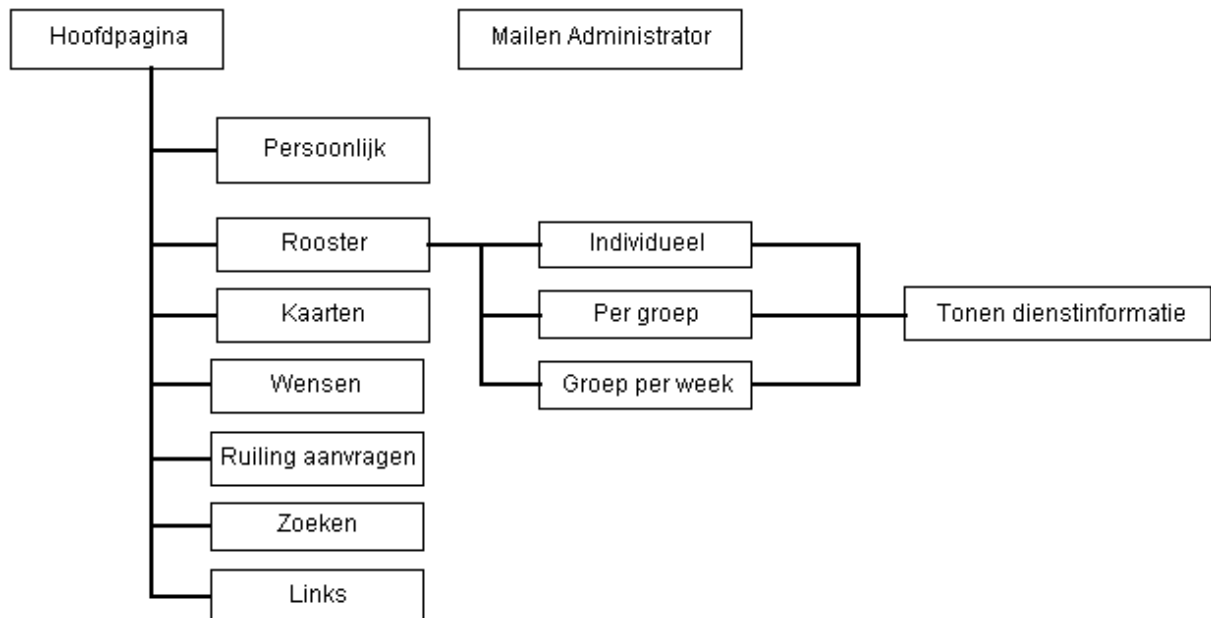
## **4.9 Links**

In dit scherm worden links getoond. Op deze pagina kunnen links gezet worden die waardevol kunnen zijn voor de medewerker. Bijvoorbeeld meer informatie over ORTEC b.v. of over HARMONY zelf.

---

## 5. Schematisch overzicht huidig HARMONY-WebAccess

Het schema dat in dit hoofdstuk wordt getoond, geeft de structuur aan van HARMONY-WebAccess zoals dat er nu uit ziet.



Schema Huidig HARMONY-WebAccess

## 6. Beschrijving toekomstig HARMONY-WebAccess

In dit hoofdstuk worden de nieuwe functies beschreven. Dit zal gedaan worden door eerst in detail aan te geven wat de functies doen, een weergave van hoe de schermen eruit komen te zien

### 6.1 Functionaliteit in detail

De functies die in deze paragraaf in detail uitgewerkt zullen worden zijn: 'Wachtwoord wijzigen', 'Uren verantwoorden', 'Overzicht mutaties + gezien', 'Rooster gezien' en 'Wensen + verlofaanvragen bekijken'. Sommige uitwerkingen zijn gedaan met behulp van use cases en andere tekstueel. De reden voor het gebruik van use cases is gedaan, omdat dit direct de structuur toont van hoe het systeem moet reageren op de acties van de medewerker.

#### 6.1.1 Wachtwoord wijzigen

In het menu 'Wachtwoord wijzigen' kan de medewerker zijn wachtwoord wijzigen.

Naam	Wijzigen wachtwoord
Aannamen	Medewerker is ingelogd en heeft het scherm 'Wachtwoord wijzigen' voor zich. (link 'Wachtwoord wijzigen' in menu)
Beschrijving	(1) Medewerker vult de velden 'Oud wachtwoord', 'Nieuw wachtwoord' en 'Bevestig nieuw wachtwoord' in en klikt op de 'Wijzigen'-button. (2) Het systeem controleert of de velden geldige waarden bevatten. Is het antwoord 'Nee' dan treedt uitzondering op. (3) Is het antwoord 'Ja' dan wordt onder de invoervelden de melding 'Uw wachtwoord is gewijzigd.' getoond.
Uitzonderingen	<p>[Niet alle velden ingevuld] Melding die getoond wordt onder de invoervelden = "U heeft niets ingevuld. Om uw wachtwoord te wijzigen moet u alle velden invullen."</p> <p>[Niets ingevuld in het veld Oud wachtwoord] Melding die getoond wordt onder de invoervelden = "Wachtwoord kan niet worden gewijzigd, want u heeft niets ingevuld in het veld Oud wachtwoord."</p> <p>[Niets ingevuld in het veld Nieuw wachtwoord] Melding die getoond wordt onder de invoervelden = "Wachtwoord kan niet worden gewijzigd, want u heeft niets ingevuld in het veld Nieuw wachtwoord."</p> <p>[Niets ingevuld in het veld Bevestig nieuw wachtwoord] Melding die getoond wordt onder de invoervelden = "Wachtwoord kan niet worden gewijzigd, want u heeft niets ingevuld in het veld Bevestig nieuw wachtwoord."</p> <p>[Oud wachtwoord onjuist] Melding die getoond wordt onder de invoervelden = "Het oude wachtwoord is onjuist. Vul wachtwoord opnieuw in."</p> <p>[Nieuw wachtwoord minder dan 5 karakters] Melding die getoond wordt onder de invoervelden = "Het wachtwoord moet minimaal 5 karakters hebben. Voer nieuw wachtwoord in."</p> <p>[Nieuw wachtwoord meer dan 19 karakters] Melding die getoond wordt onder de invoervelden = "Kan het wachtwoord niet wijzigen, want het wachtwoord mag maximaal 19 karakters hebben. Vul nieuw wachtwoord in."</p> <p>[Nieuw wachtwoord en Bevestig nieuw wachtwoord niet gelijk aan elkaar] Melding die getoond wordt onder de invoervelden = "De invoer</p>

	van de velden Nieuw wachtwoord en Bevestig nieuw wachtwoord zijn niet gelijk aan elkaar. Vul nieuwe waarden in.”.
Resultaat	Medewerker heeft zijn wachtwoord gewijzigd en systeem kent nieuw wachtwoord.

## 6.1.2 Uren verantwoord

In het menu wordt een link 'Rooster – Mijn rooster' getoond. Door hier op te klikken wordt het dienstrooster getoond van de ingelogde persoon.

Naam	Realiseren dienst
Aannamen	Medewerker is ingelogd en zijn dienstrooster wordt getoond (link 'Rooster – Mijn rooster' in menu)
Beschrijving	(1) Medewerker klikt op een dienst. Als de datum en begintijd van de originele dienst groter is dan de huidige datum en tijd en / of status van de dienst is ongelijk aan 'Gepubliceerd' en / of dienst behoort niet bij de ingelogde persoon dan treedt uitzondering op. (2) Als de datum en tijd gelijk of kleiner is dan de huidige datum en tijd en status van de dienst is 'Gepubliceerd' en de dienst hoort bij de ingelogde persoon dan toont het systeem de gepubliceerde dienst (van elke activiteit ziet de medewerker de naam, het type, de werkplek, begintijd en eindtijd). (3) Medewerker kan nu de dienst realiseren of aanpassen. Als medewerker een dienst wil aanpassen wordt use-case 'Aanpassen dienst' uitgevoerd. Als hij wil stoppen treedt uitzondering op. (4) Medewerker geeft zijn akkoord door op de 'Opslaan'-button te klikken. (5) Voordat het akkoord definitief is krijgt de medewerker de melding "Weet u zeker dat u alles goed heeft ingevuld?". Zo nee dan treedt uitzondering op. Zo ja dan klikt de medewerker op de 'Ok'-button en keert terug naar zijn dienstrooster.
Uitzonderingen	[Datum en begintijd van ingeplande dienst is groter dan datum en tijd van huidige dag en / of status van de dienst is ongelijk aan 'Gepubliceerd' en / of dienst behoort niet bij de ingelogde persoon] De dienstgegevens worden getoond, maar alle buttons, icoontjes en invoervelden zijn onzichtbaar of ge-disabled.  [Medewerker wil stoppen] Medewerker klikt op de 'Cancel'-button of op het kruisje en keert terug naar het dienstrooster in het scherm 'Rooster – Mijn rooster'.  [Medewerker is niet zeker dat hij alles goed heeft ingevuld] Medewerker klikt op de 'Cancel'-button en keert terug in het scherm van de geselecteerde dienst.
Resultaat	Medewerker heeft dienst gerealiseerd, in de gewone client is zichtbaar dat de dienst gerealiseerd is en de webclient toont het dienstrooster van de medewerker op het scherm.

Naam	Aanpassen dienst
Aannamen	Het systeem toont de geselecteerde dienst en de datum en begintijd van de dienst is kleiner of gelijk aan huidige datum en tijdstip en de status van de dienst is 'gepubliceerd' en de dienst behoort bij de ingelogde persoon.
Beschrijving	(1) Medewerker kan activiteiten toevoegen, wijzigen of verwijderen. Als medewerker op 'Activiteit toevoegen' klikt wordt use-case 'Toevoegen activiteit' uitgevoerd. Als medewerker op 'Activiteit verwijderen' klikt wordt use-case 'Verwijderen activiteit' uitgevoerd. Als medewerker een activiteit wil wijzigen dan kan hij de volgende velden behorend bij de desbetreffende activiteit aanpassen. Deze velden zijn: Naam, Werkplek,

	Type en Eind (=eindtijd). Het veld 'Begin' (=begintijd) kan alleen aangepast worden als het de eerste activiteit van de dienst betreft. Als medewerker zijn wijzigingen ongedaan wil maken wordt use case Dienst herstellen uitgevoerd. (2) Medewerker slaat de gegevens op door op de 'Opslaan'-button te klikken.
Uitzonderingen	[Geen]
Resultaat	Medewerker heeft dienst aangepast en gerealiseerd, het systeem toont het dienstrooster van de medewerker.

Naam	Toevoegen activiteit
Aannamen	Het systeem toont de door de medewerker toegevoegde dienst en de medewerker heeft op het icoontje 'Activiteit toevoegen' geklikt. De begintijd is gelijk aan de eindtijd van de voorgaande activiteit en de eindtijd is gelijk aan begintijd + één uur.
Beschrijving	(1) Het systeem toont de activiteit en medewerker kan de volgende velden aanpassen: Naam, Werkplek, Type en Eind (=eindtijd). (2) Als medewerker akkoord is met het invoeren van de activiteit klikt hij op de 'Opslaan'-button. Als medewerker niet akkoord gaat treedt uitzondering op. (3) Systeem slaat wijziging op en keert terug naar het individuele rooster.
Uitzonderingen	[Toevoegen ongedaan maken] Voer use case Verwijderen activiteit uit.
Resultaat	Het systeem toont het individuele rooster (link 'Rooster – Mijn rooster') en er is een activiteit toegevoegd aan de desbetreffende dienst.

Naam	Verwijderen activiteit
Aannamen	Medewerker heeft op het icoontje activiteit verwijderen geklikt.
Beschrijving	(1) Activiteit wordt verwijderd.
Uitzonderingen	[Geen]
Resultaat	Het systeem refresht het dienstenscherm en de aangeklikte activiteit is verwijderd.

Naam	Herstellen dienst
Aannamen	Medewerker heeft op de button Herstellen geklikt.
Beschrijving	(1) De realisatie van de dienst wordt verwijderd en het scherm wordt gesloten. (Als medewerker weer op de desbetreffende dienst klikt, wordt de originele dienst getoond).
Uitzonderingen	[Geen]
Resultaat	Realisatie is verwijderd en het scherm dienstrooster wordt getoond.

### 6.1.3 Overzicht mutaties + gezien

In het menu wordt een link 'Overzicht mutaties' getoond. Door op deze link te klikken krijgt de medewerker standaard een overzicht van alle wijzigingen (mutaties) die hij nog niet heeft gezien.

De medewerker heeft ook de mogelijkheid om alle mutaties binnen een bepaalde periode op te vragen, dus ook de mutaties die hij al heeft gezien.

De medewerker moet per mutatie kunnen aangeven dat hij de mutatie heeft gezien en het moet mogelijk zijn om alle mutaties in één keer te accorderen (AccorderenMutaties).

Deze functie is bedoeld om de planner te ontlasten. Het kan zijn dat door bijv. ziekte van een medewerker een dienst moet worden opgevuld door een andere medewerker. De planner moet dan het rooster aanpassen. De medewerker die de dienst moet opvullen heeft dus een wijziging in zijn dienstrooster. Door middel van deze functies (AccorderenMutaties) kan de medewerker aangeven dat hij de wijziging heeft gezien.

Deze functies is nieuw in HARMONY. Dit betekent dat als deze functies ontwikkeld is, dit niet is terug te vinden in de gewone client. De planner die met de gewone client werkt zal dus niet kunnen zien of een medewerker de mutaties heeft gezien. In de gewone client zal dus ook een aanpassing gedaan moeten worden, dit valt echter buiten het project.

### 6.1.4 Rooster gezien

In het menu 'Rooster' → 'Mijn rooster' kan de medewerker aangeven dat hij zijn rooster heeft gezien.

In het individuele rooster wordt standaard een periode van zes (6) weken getoond. Deze periode is als volgt opgebouwd. De week voor de huidige week, de huidige week en vier weken na de huidige week.

Zodra een week in het scherm getoond wordt die een medewerker niet heeft gezien, wordt de button 'Rooster gezien' getoond. Zodra de medewerker hierop klikt wordt de tot-datum uit de date-time-picker opgeslagen in de tabel medewerker. De date-time-picker wordt in de huidige situatie ook al getoond om een periode te kunnen selecteren die de medewerker wil zien. Deze hoeft dus niet apart ontwikkeld te worden.

Als de geselecteerde einddatum groter is dan de opgeslagen datum wordt tussen de Periode selectie tabel en het rooster de tekst "Het rooster bevat ongeziene diensten. U heeft uw rooster gezien tot *opgeslagen\_datum*. Middels de button onderaan het rooster kunt u aangeven dat u het rooster heeft gezien tot *geselecteerde\_einddatum*." getoond.

Zodra het verschil tussen de opgeslagen datum en de geselecteerde einddatum groter is dan 7 (zeven) dagen, wordt de melding getoond "U heeft aangegeven tot *geselecteerde\_datum* uw rooster gezien te hebben. Wilt u deze datum opslaan?".

Klikt de medewerker op:

- Ok dan wordt de geselecteerde einddatum opgeslagen;
- Cancel dan wordt er geen datum opgeslagen.

Deze functies is geheel nieuw in het HARMONY-pakket. Dit betekent dat er in het huidige HARMONY nog niet bijgehouden wordt of een medewerker zijn rooster heeft gezien. De reden hiervan is dat in de huidige situatie de medewerkers via de telefoon of via de post (meestal rooster per maand) te horen / zien krijgen wanneer ze dienst hebben. De planner ging er dan vanuit dat alle medewerkers wisten wat hun rooster was.

De reden voor het ontwikkelen van deze functies is dat de kostenpost flink kan oplopen bij het periodiek (meestal maandelijks) via de post versturen van de dienstroosters. Het is voor grote bedrijven goedkoper als medewerkers via internet toegang hebben tot hun rooster. Dit scheelt in kosten wat betreft het versturen van post alswel in tijd omdat de

---

briefen ook nog uitgeprint en in enveloppen gedaan moeten worden. Dit is niet nodig als medewerkers via internet hun rooster kunnen bekijken.

## **6.1.5 Wensen + verlofaanvragen bekijken**

In het menu 'Wensen' van de huidige versie van HARMONY-WebAccess heeft de medewerker de volgende mogelijkheden:

Hij kan:

- aangeven wanneer hij wel / niet wil werken;
- een dienstwens opgeven;
- een verlofwens opgeven.

In de header waarin de soort wens staat, zal een tekst-hyperlink getoond worden "Nieuwe wens invoeren".

Als een medewerker een wens wil invoeren klikt hij de gewenste hyperlink en vult zijn wens in. Als hij het ermee eens is, klikt hij op de button 'Opslaan'. Als hij wil stoppen klikt hij op de button 'Annuleren'. Gebruiker keert terug in het menu Overzicht wensen.

Naam	Bekijken wensen
Aannamen	Medewerker is ingelogd en heeft op de link 'Wensen' geklikt.
Beschrijving	(1) Het systeem toont de alle wensen vanaf de huidige van de medewerker in het scherm 'Overzicht wensen'. De wensen worden gegroepeerd per soort getoond. Als er geen wensen zijn treedt uitzondering op.
Uitzonderingen	[Geen wensen] Als er geen wensen zijn wordt de tekst "Geen wensen" getoond.
Resultaat	Medewerker ziet zijn wensen

Mogelijke soorten wensen:

- eenmalig;
- repeterend;
- dienstgebonden.

De repeterend wensen zullen één keer op het scherm getoond worden met daarbij de vermelding om welke periode het gaat (wekelijks, maandelijks, etc.).

Mogelijke statussen:

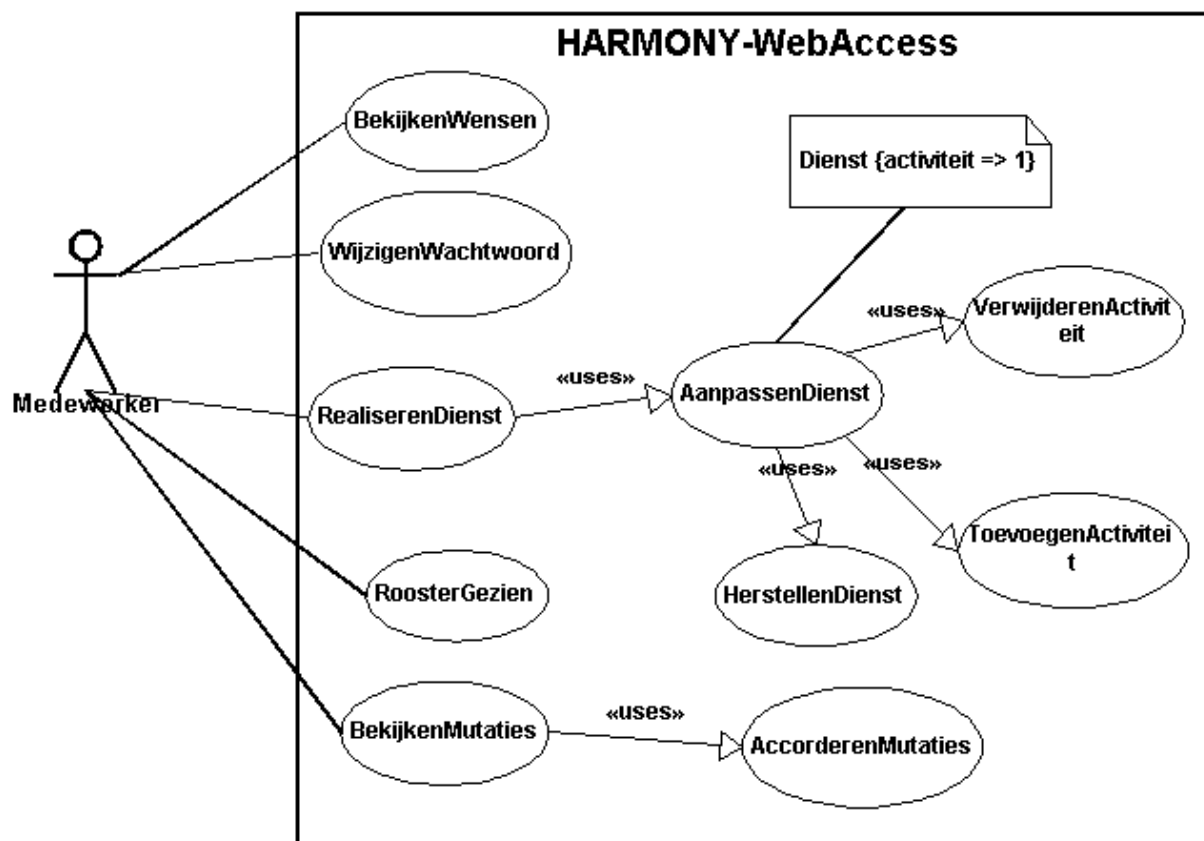
- Eis (er zal altijd aan de wens voldaan worden);
- Anders 0 t/m 100. Deze getallen corresponderen met de mate waarin de planautomaat rekening zal houden met de honorering.

## **6.2 Acties gebruiker met het systeem**

Door middel van het onderstaande schema wordt aangegeven binnen welke de context (=HARMONY-WebAccess) het project zal plaatsvinden en welke gebruikers (=Medewerker) communiceren met deze context.

---





Use-case diagram

## 6.3 Lay-out schermen

In deze paragraaf zal van de functies 'Wachtwoord wijzigen', 'Uren verantwoorden', 'Overzicht mutaties + gezien', 'Rooster gezien' en 'Wensen + verlofaanvragen bekijken' beschreven worden wat voor gegevens op de pagina zijn te zien en wat voor buttons er getoond worden.

### 6.3.1 Scherm Wachtwoord wijzigen

In het menu wordt een link 'Wachtwoord wijzigen' opgenomen.

Als een gebruiker deze link aanklikt komt hij in het scherm 'Wachtwoord wijzigen'.

In het scherm staan vier velden en één button.

De namen van de vier velden zijn:

- Gebruikersnaam (wordt automatisch gevuld en kan niet gewijzigd worden);
- Oud wachtwoord (hier moet het huidige wachtwoord ingevuld worden);
- Nieuw wachtwoord (het nieuwe wachtwoord);
- Bevestig wachtwoord (nogmaals het nieuwe wachtwoord).

De naam van de button is wijzigen.

Boven aan het scherm komt Wachtwoord wijzigen te staan.

### 6.3.2 Scherm Uren verantwoorden

Als medewerker op de link 'Rooster – Mijn rooster' klikt, krijgt hij zijn rooster te zien. In dit rooster worden de diensten getoond die de medewerker heeft gedaan of die hij nog moet doen.

Als hij op een dienst klikt wordt een nieuw scherm geopend waarin alle activiteiten van deze dienst worden getoond.

De opmaak van dit scherm is als volgt:

Van de dienst wordt getoond:

- De naam van de dienst;
- De begin- en eindtijd van de dienst, de werktijd, arbeidstijd, dag + datum en de status;
- Naam, werkplek, type, begin en eind.

Achter elke activiteit wordt een icoontje 'Activiteit toevoegen' en 'Activiteit verwijderen' getoond.

Als op activiteit toevoegen wordt geklikt wordt achter de desbetreffende activiteit een activiteit toegevoegd.

Als medewerker bij een bepaalde activiteit op 'Activiteit verwijderen' klikt, wordt deze activiteit verwijderd.

Een toevoeging of verwijdering is meteen definitief.

Om terug te gaan naar de oude situatie kan de medewerker op de Herstellen-button klikken.

Onderaan dit scherm zijn twee buttons te zien:

1. Opslaan;
2. Herstellen;
3. Annuleren.

Het scherm ziet er als volgt uit:

**Dienst/Activiteit:**

**D**

Begintijd:	09:00	Werktijd:	08:00
Eindtijd:	17:00	Arbeidstijd:	08:00
Dag:	vrijdag	Datum:	31-10-2003
Status:	Verantwoord		

Naam	Werkplek	Type	Begin	Eind
Activiteit	wp2	Werk	09:00	16:00
Acgfh	wp2	Werk	16:00	17:00

**Opslaan** **Herstellen** **Annuleren**

### 6.3.3 Scherm Overzicht mutaties

Dit scherm wordt geopend door op de link 'Overzicht mutaties' te klikken. In het scherm worden alle mutaties getoond die van toepassing zijn op de medewerker en die hij nog niet heeft gezien. Boven aan het scherm komt Overzicht mutaties te staan.

Om aan te geven dat de medewerker een mutatie heeft gezien kan hij deze mutatie aanvinken. Achter elke mutatie staat namelijk een vakje waarin je een vinkje kan zetten.

Als de medewerker wil aangeven dat hij alle mutaties heeft gezien klikt hij op de 'Alles gezien'-button. Achter alle mutaties komt een vinkje te staan. De 'Alles gezien'-button wordt boven aan het overzicht van de mutaties getoond. Onder het overzicht komt een button 'Opslaan'. Als de medewerker hier op klikt worden alle wijzigingen opgeslagen.

De informatie waaruit de mutatie bestaat ziet er als volgt uit:

- Datum = de datum van de dienst;
- Tijdstip van mutatie = tijdstip waarop mutatie is doorgevoerd;
- Type mutatie = gaat het om een wijziging of een ruiling;
- Oude dienst = dienst die op die dag stond gepland;
- Roostergroep = de naam van de roostergroep waartoe de oude dienst behoort;
- Nieuwe dienst = de naam van de dienst die geruild is;
- Roostergroep = de naam van de roostergroep waartoe de nieuwe dienst behoort;
- Van = De naam van de persoon waarmee geruild is;
- Doorgevoerd door = de naam van de persoon die de mutatie heeft ingevoerd.

Als de medewerker alle mutaties wil zien, dus ook de mutaties die hij al heeft gezien dan kan hij aan de hand van een begindatum en einddatum een periode geselecteerd worden.

#### 6.3.4 Scherm Rooster gezien

In het menu 'Rooster – Mijn rooster' wordt het persoonlijke dienstrooster van de ingelogde medewerker getoond.

Zodra een periode op het scherm getoond wordt die de medewerker niet heeft gezien wordt een button 'Rooster gezien' getoond, als de medewerker hierop klikt wordt de datum van vandaag opgeslagen.

Het scherm zal er als volgt uit gaan zien:

Deze week is het: Week 40

Persoonlijk rooster - Elfring, C

Periode selecteren

Van: 22-09-2003

Tot: 27-10-2003

Periode aanpassen

	maandag	dinsdag	woensdag	donderdag	vrijdag	zaterdag	zondag
week 39							
week 40			Dienst1				
week 41	Dienst1						
week 42		Dienst1					
week 43							
week 44							

Standaard periode roosteroverzicht van Persoonlijk rooster

Het scherm ziet er niet anders uit dan zoals die in de huidige situatie eruit ziet. Het verschil zit hem in de standaard-periode die getoond zal worden en de button die getoond zal worden zodra de medewerker het getoonde rooster in zijn geheel of gedeeltelijk niet heeft gezien.

Zodra de medewerker een het rooster in zijn geheel of gedeeltelijk niet heeft gezien zal naast de tabel 'Periode selecteren' en tussen de bovenbalk en het getoonde rooster een button 'Rooster gezien' getoond worden.

### 6.3.5 Scherm Wensen

Als de medewerker op de link 'Wensen' klikt zal hij een overzicht krijgen van zijn eventueel ingevoerde wensen. Standaard zullen de wensen vanaf de huidige getoond worden.

De volgorde van tonen zal zijn:

- de wensen met betrekking tot het wel of niet willen werken in een bepaalde periode;
- de dienstwensen;
- de verlofwensen.

Van elke wens zullen de volgende velden getoond worden:

- Naam. De naam van de wens;
- Type. Type van de wens, bijv. Dienstwens, Verlofwens, Terugkerende wens, etc.;
- Omschrijving. Omschrijving van de wens, bijv. Sporten, Bruiloft, etc.;
- Van. Datum vanaf wens ingaat;
- Tot. Datum tot wanneer de wens gewenst is;
- Belangrijkheid. (zie beschrijving in H. 6.1.5)
- Gehonoreerd.

### Wel / niet werken in een periode

Onder de link 'Wel / niet werken in een periode' kan een medewerker aangeven wanneer hij wel of niet wil werken. Dit gaat op de volgende manier:



Scherf Wel/niet werken in een periode

In het veld 'Van:' kan medewerker begindatum + tijd kiezen.

In het veld 'Tot:' kan medewerker einddatum + tijd kiezen.

Aangeven wat hij in deze periode wil (werken / niet werken).

Voor akkoord moet hij op de 'Verzend'-button klikken.

### Een dienstwens opgeven

In het veld 'Wenst op:' kan medewerker datum invoeren waarop de wens betrekking heeft. In het veld 'Dienst:' kan hij aangeven om welke dienst het gaat. Vervolgens kan hij aanklikken of hij wel wil werken of juist niet wil werken. Voor akkoord moet hij op de 'Verzend'-button klikken.

---

Een dienstwens opgeven:

Medewerker: Elfring

Wenst op: 11-06-2003

Dienst: AT

☒ wel werken  
☐ niet werken

Verzend

Scherf Een dienstwens opgeven

## Verlofwens

In het veld 'Type verlof:' kan aangegeven worden wat voor soort verlof de medewerker wenst. In de velden 'Van:' en 'Tot:' kan aangegeven worden wanneer en hoe lang het verlof moet duren.

Voor akkoord moet hij op de 'Verzend'-button klikken.

Verlofwens

Medewerker: Elfring

Type verlof: Verlof

Van: 11-06-2003

Tot: 27-06-2003

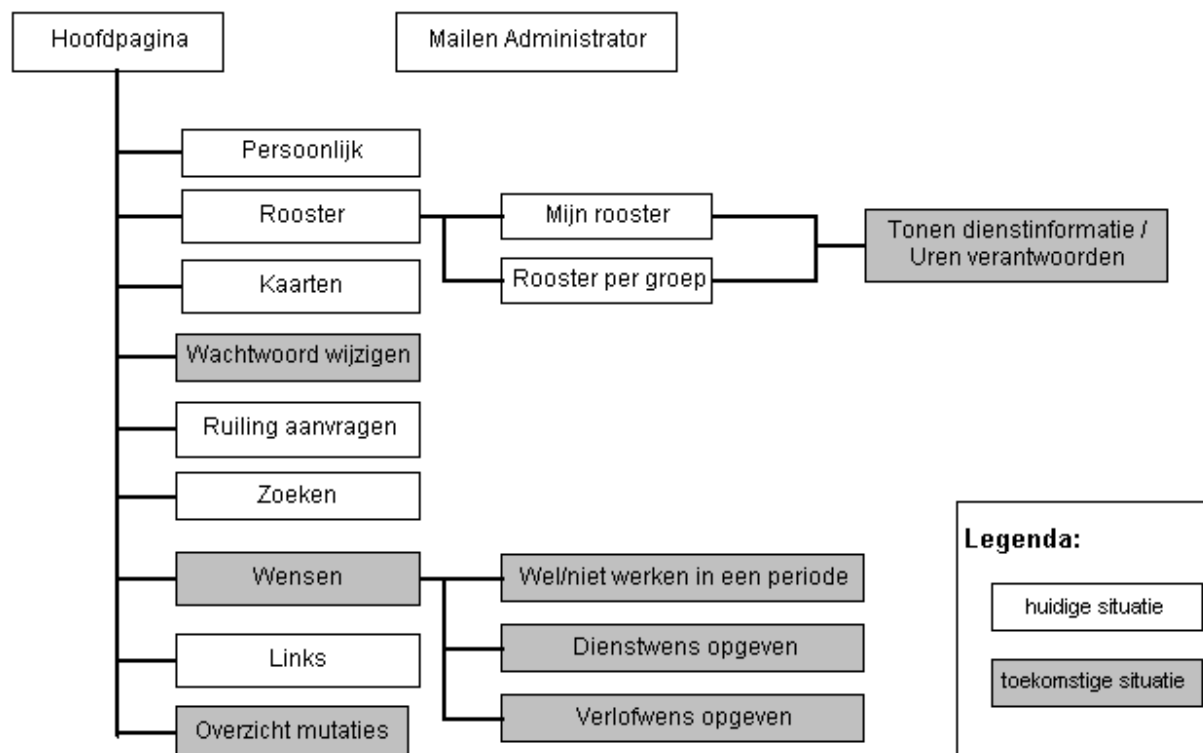
Verzend

Scherf Verlofwens

---

## 6.4 Schematisch overzicht toekomstig HARMONY-WebAccess

In het schema hieronder wordt een overzicht getoond van de structuur zoals HARMONY-WebAccess eruit zal zien na het ontwikkelen van de nieuwe functies.



**Schema Toekomstig HARMONY-WebAccess**

## **6.5 Niet opgenomen punten**

In deze paragraaf worden de punten beschreven die tijdens de brainstormsessie genoemd zijn, maar die niet meegenomen zijn in het functioneel ontwerp. Per punt zal uitgelegd worden wat de reden is voor het niet meenemen in dit project.

### **Print-optie**

Een punt dat tijdens de brainstorm-sessie naar voren was gekomen, was om een print-functies toe te voegen. De print-optie zou handig kunnen zijn voor de medewerker voor het uitprinten van zijn rooster.

Het blijkt echter dat de print-mogelijkheid die MS Internet Explorer biedt voorlopig voldoende is. In MS Internet Explorer is het mogelijk om een web-pagina af te drukken (zie menu File – Print). Om te kunnen zien wat er nu afgedrukt gaat worden kun je in het menu File – Print preview terecht. Je kunt kiezen voor de volgende opties:

- As laid out on screen = Het hele scherm. Wordt ongeveer  $\frac{3}{4}$  van het scherm afgedrukt, is niet zo mooi;
- Only the selected frame = Alleen het geselecteerde frame. Deze optie geeft het beste resultaat;
- All frames individually = Elk frame apart. Slechtste optie nu alle frames van de desbetreffende pagina getoond worden en je dus het juiste frame eerst moet opzoeken.

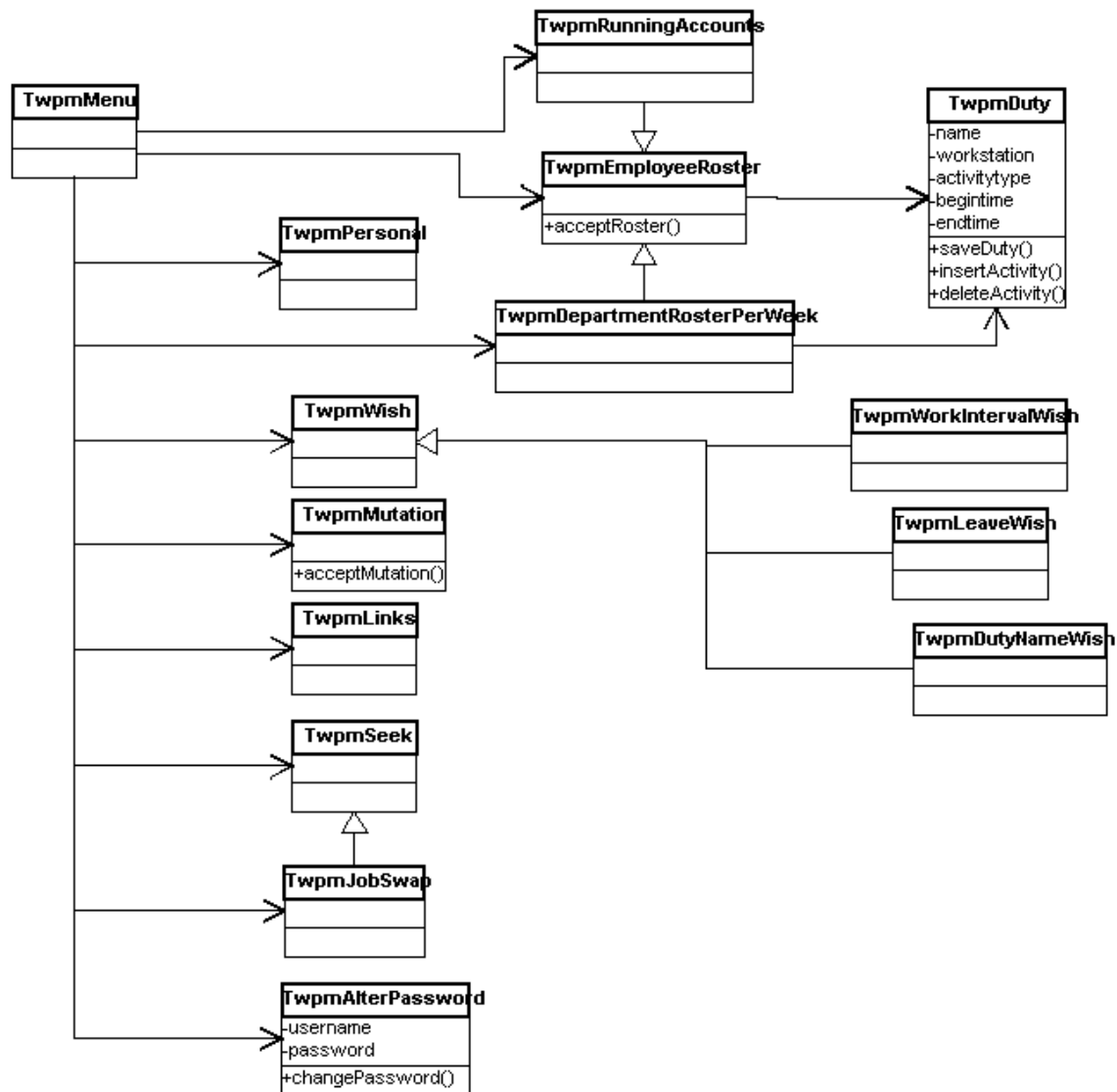
### **Kleuren tonen in planning**

Er werd voorgesteld om in de planning dezelfde kleuren te laten zien zoals die in het planbord van de client-versie is te zien. Deze mogelijkheid heeft echter geen hogere prioriteit dan de andere functies en aangezien het heel wat tijd zal kosten om deze mogelijkheid te ontwikkelen is besloten het niet mee te nemen in dit project.

---

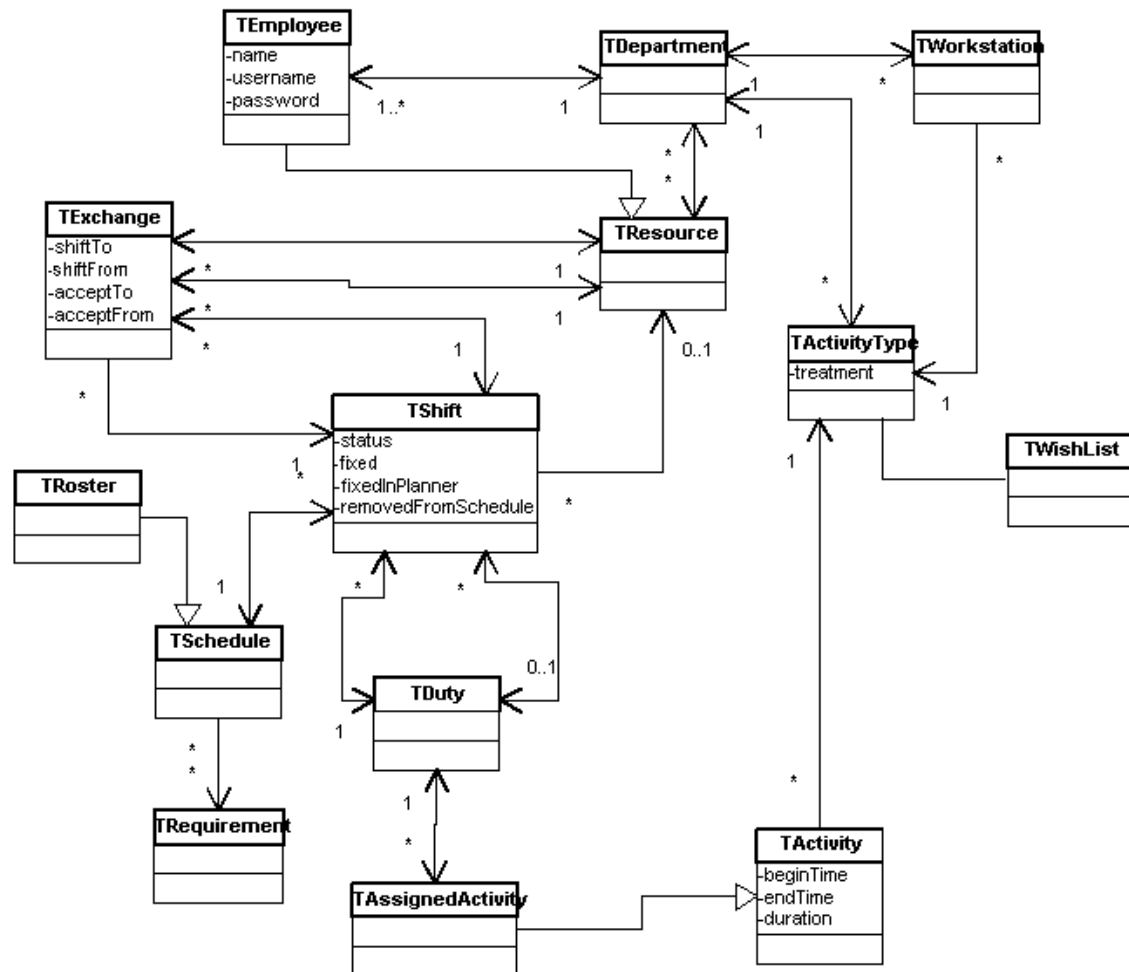
## 7. Class diagram

In dit hoofdstuk worden de relaties tussen de classes van HARMONY WebAccess getoond met behulp van een class diagram. Aangezien de afhandeling van de requests via HARMONY Server plaatsvindt is ook een class diagram van HARMONY gemaakt om overzicht te hebben van de classes die nodig zijn om de juiste response te kunnen geven.



Class diagram van Harmony WebAccess





Class diagram HARMONY

## 8. Techniek

Dit hoofdstuk geeft een beschrijving van de techniek die nodig is om de te ontwikkelen functies te kunnen gebruiken. De techniek bestaat uit de benodigde hardware, software, goede performance en de installatieprocedure.

### 8.1 Hardware

De hardware die nodig is om HARMONY-WebAccess te gebruiken bestaat uit een:

- Database-server;
- HARMONY-server;
- HARMONY-WebServer;
- PC.

### 8.2 Software

Als de klant alle genoemde hardware heeft met het benodigde HARMONY softwarepakket dan is voor het gebruik van HARMONY-WebAccess alleen een PC met webbrowser nodig met een internet-verbinding. De eis die aan de webbrowser wordt gesteld is dat de browser minimaal Microsoft Internet Explorer 5+ moet zijn.

### 8.3 Performance

De performance van HARMONY-WebAccess zal voornamelijk afhangen van de datasnelheid van de internet-verbinding van de gebruiker. Hoe sneller de verbinding hoe beter de performance.

Bij het versturen van data van de WebServer naar de browser zal alleen de data gestuurd worden waar de gebruiker om vraagt en bij het updaten van gegevens door de gebruiker zal alleen die data verstuurd worden die de WebServer nodig heeft om de gegevens door te sturen naar de HARMONY-server.

De afstudeerder verwacht dat de volgende functies veel gebruikt zullen worden:

- Bekijken Mijn rooster;
- Invoeren van uren verantwoorden;
- Bekijken mutaties;
- Bekijken + aanvragen ruilingen.

De functies voor het uren verantwoorden zal redelijk veel data gestuurd worden, aangezien het echter maar voor één persoon is zal dit geen nadelige gevolgen hebben op de performance.

### 8.4 Installatie

De installatie wordt beschreven in de Technische Systeemdokumentatie (versie 1.0) van Jeroen de Jong.

## 9. Documentatie

Er zal een gebruikershandleiding geschreven worden van de opgeleverde functies. Deze documentatie beschrijft de werking van de functies. De gebruikers kunnen uit deze documentatie hun informatie halen wanneer zij hulp nodig hebben op een bepaald punt.

---

## **Externe bijlage VI    Technisch Ontwerp**

### **Technisch Ontwerp HARMONY WebAccess**

Versie 1.0

Alex Nap

---

---

## Tekenpagina

### Wijzigingshistorie:

Versie	Datum	Auteur(s)	Wijzigingen	Goedgekeurd door		
				Datum	Naam	Handtekening

## Inhoudsopgave

<b>1. INLEIDING</b>	<b>1</b>
<b>2. REFERENTIES</b>	<b>1</b>
<b>3. TECHNISCHE DETAILS FUNCTIES</b>	<b>1</b>
3.1 EERSTE INCREMENT	1
3.1.1 Wachtwoord wijzigen (9303)	1
3.1.2 Uren verantwoorden (9439)	1
3.1.3 Testen functies 1e increment	4
3.2 TWEEDE INCREMENT	7
3.2.1 Overzicht mutaties (9754)	7
3.2.2 Testen Overzicht mutaties	8
3.3 DERDE INCREMENT	9
3.3.1 Rooster gezien (10087)	9
3.3.2 Overzicht wensen + verlofaanvragen (10088)	10
3.3.3 Testen Rooster gezien (10087)	11
3.3.4 Testen Overzicht wensen + verlofaanvragen (10088)	11
<b>4. CLASS DIAGRAM</b>	<b>13</b>
4.1 TOTAALOVERZICHT	13
4.2 OVERZICHT TOPLAAG	14
4.3 OVERZICHT LOGIN	15
4.4 OVERZICHT ONGEWIJZIGDE CLASSES (DATUM/TIJD, PERSOONLIJK, LINKS)	15
4.5 OVERZICHT ZOEKEN EN RUILING AANVRAGEN	16
4.6 OVERZICHT WACHTWOORD WIJZIGEN	16
4.7 OVERZICHT ROOSTER	17
4.8 OVERZICHT MUTATIES	18
4.9 OVERZICHT WENSEN	18
4.10 OVERZICHT HARMONY WEBACCESS	19
4.11 OVERZICHT HARMONY	20

---

## 1. Inleiding

In dit document worden de functies van de drie incrementen beschreven behorend bij het project "Ontwikkeling functionaliteit voor HARMONY WebAccess".

## 2. Referenties

Functioneel Ontwerp HARMONY-WebAccess van Alex Nap

## 3. Technische details functies

Dit project bestaat uit drie incrementen waarin één of meerdere functies ontwikkeld worden. Per increment zullen de technische details en de testcases van de functies beschreven worden.

### 3.1 Eerste increment

Het eerste increment bestaat uit de functies Wachtwoord wijzigen en Uren verantwoorden.

#### 3.1.1 Wachtwoord wijzigen (9303)

Het wachtwoord wijzigen bestaat uit vier velden en een button.

De velden die getoond worden zijn:

- Gebruikersnaam (een teksteditveld dat niet ge-edit kan worden);
- Oud wachtwoord, Nieuw wachtwoord en Bevestig nieuw wachtwoord. Allen teksteditvelden.

Een button 'Wijzigen', door op deze button te klikken worden een aantal controles uitgevoerd.

Als de controles geen negatief resultaat (dus Exit) opleveren dan wordt het wachtwoord opgeslagen op de volgende wijze:

```
// NewPassw bevat de waarde van het invoerveld NewPassword  
SessionEmployee.Password := NewPassw;
```

Controles die uitgevoerd worden, worden beschreven in de hieronder staande constraints.

Constraints:

- De velden Oud wachtwoord, Nieuw wachtwoord en Bevestig nieuw wachtwoord mogen niet leeg zijn;
- Het wachtwoord in de database van de ingelogde gebruiker moet overeen komen met het ingevulde wachtwoord in het veld Oud wachtwoord;
- De invoer van de velden Nieuw wachtwoord en Bevestig nieuw wachtwoord moet aan elkaar gelijk zijn;
- Wachtwoord moet minimaal 5 karakters lang zijn;
- Wachtwoord mag maximaal 19 karakters bevatten.

#### 3.1.2 Uren verantwoorden (9439)

De functie Uren verantwoorden is terug te vinden onder de link 'Rooster – Mijn rooster'. Door op een dienst te klikken worden de gegevens behorend bij die dienst getoond.

Een dienst (=Shift) bestaat uit een Duty, de datum van de dag waarop de Duty gepland is en een EmployeeId. Een Duty bestaat uit één of meerdere Activities. Een shift kan aan een medewerker gekoppeld worden.

Het moet mogelijk zijn om alle buttons, icoontjes en velden te disabelen of onzichtbaar maken, zodat een medewerker niet een dienst gaat wijzigen en / of realiseren die de status 'Verantwoord' heeft of de dienst nog moet doen.

Het scherm is alvolgt opgebouwd. Bovenaan staat de naam van de dienst en daaronder staan drie tabellen. Hieronder staat per tabel beschreven wat er in de tabel staat.

In de eerste tabel worden vier velden getoond, te weten: Begintijd, Eindtijd, Werktijd, Arbeidstijd, Dag, Datum en Status deze zeven velden staan los van de activiteiten en zullen niet bewerkt (ge-edit) kunnen worden.

In de tweede tabel staan de activiteiten, de volgende velden zijn te zien:

- Naam is texteditfield (style="width: 80" maxlength=15);
- Werkplek een pull-down menu (style="width: 60", size = 1);
- Type een pull-down menu (style="width: 60", size = 1);
- Begin en eindtijd texteditfields (style="width: 40", maxlength=5). Invoer = uu:mm.

Icoontjes die achter elke activiteit getoond zullen worden:

- Icoontje toevoegen met de hinttekst 'Activiteit toevoegen';
- Icoontje verwijderen met de hinttekst 'Activiteit verwijderen'.

De classes die bij de onderstaande methoden horen zijn:

```
ActivityPropertiesArray, UpdatedActivityPropertiesArray : class
TActivityPropertiesArray
TheDuty : class TDuty
```

### Activiteit toevoegen:

Alleen van de eerste activiteit mag de begintijd worden aangepast.

Bij alle activiteiten mag de eindtijd aangepast worden, waarbij erop gelet moet worden dat de begintijd van activiteit 2 gelijk is aan de eindtijd van activiteit 1. Zoals in het voorbeeld hieronder wordt getoond:

Naam:	Werkplek:	Type:	Begin:	Eind:		
Activiteit1	Kantoor	Werk	08:30	12:30	<insert >	<delet e>
Activiteit2	Kantine	Werk	12:30	14:45	<insert >	<delet e>

*Voorbeeld van dienstoverzicht*

Voor het toevoegen van een activiteit is de volgende functionaliteit nodig:  
JavaScript action = 'DoInsert' → submit → handleAddActivity(TheDuty: Tduty)

Voor het opslaan van de gegevens is deze methode nodig:

```
FillActivityPropertiesArray(ActivityPropertiesArray, TheDuty);
TheDuty.UpdateActivities(ActivityPropertiesArray,
UpdatedActivityPropertiesArray, Index + 1);
```

### Activiteit verwijderen:

Als de activiteit die verwijderd wordt tussen twee andere activiteiten staat dan wordt de begintijd van de onderste activiteit gelijk aan de begintijd van de bovenste activiteit. De eindtijd van de onderste activiteit blijft gelijk. Als de eerste of laatste activiteit uit een dienst wordt verwijderd blijven alle tijden gelijk.

Voor het verwijderen van een activiteit is de volgende functionaliteit nodig:  
JavaScript action = 'DoDelete' → submit → handleDeleteActivity(TheDuty: Tduty).

Om een activiteit te verwijderen wordt de volgende code gebruikt:

```
for J := I to High(ActivityPropertiesArray) - 1 do
  ActivityPropertiesArray[J] := ActivityPropertiesArray[J+1];
  SetLength(ActivityPropertiesArray, Length(ActivityPropertiesArray) - 1);
  Break;
```

Om deze wijziging op te slaan wordt gebruik gemaakt van de methode:  
 FillActivityPropertiesArray(ActivityPropertiesArray, Row);  
 TheDuty.UpdateActivities(ActivityPropertiesArray, UpdatedActivityPropertiesArray);

In de derde tabel staan de buttons:

- Opslaan (de dienst met eventuele wijzigingen wordt opgeslagen in de database en het scherm wordt gesloten);
- Herstellen (de gerealiseerde dienst wordt verwijderd en het scherm wordt gesloten);
- Annuleren (alle niet opgeslagen wijzigingen worden ongedaan gemaakt en scherm wordt gesloten).

### **Opslaan van wijzigingen**

Als op de button Opslaan wordt geklikt worden de gewijzigde gegevens in de database opgeslagen. Medewerker krijgt een pop-up scherm met de melding "U weet zeker dat u alles goed heeft ingevuld?". Zo nee, dan wordt er niets opgeslagen.

JavaScript action = 'Update' → submit → HandleUpdateDuty(TheDuty: TDuty)

Voor het vullen van de huidige dienst wordt gebruik gemaakt van de volgende functie:

```
FillActivityPropertiesArray(ActivityPropertiesArray);
```

Voor het wijzigen van de huidige dienst wordt de Updateactivities methode gebruikt:

```
TheDuty.UpdateActivities(ActivityPropertiesArray, UpdatedActivityPropertiesArray);
```

### **Herstellen**

Als op de button Herstellen wordt geklikt wordt de realisatie verwijderd en wordt de originele dienst getoond in het dienstrooster.

Voor het verwijderen van de gerealiseerde dienst wordt de procedure Duty.Delete gebruikt.

### **Annuleren**

Als op de button Annuleren wordt geklikt worden alle wijzigingen (die niet opgeslagen zijn) ongedaan gemaakt en wordt het scherm gesloten. Voor het ongedaan maken van wijzigingen kan zowel op Annuleren geklikt worden als op het kruisje in de rechterbovenhoek van het scherm.

Voor het sluiten van het scherm is de volgende JavaScript functie nodig:

```
window.close()
```

### **Veld Naam**

De naam van de activiteit moet altijd ingevuld zijn. Als de inhoud van dit veld leeg of alleen uit spatie(s) bestaat zal de dienst niet opgeslagen kunnen worden.

### **Veld Werkplek**

In het veld Werkplek worden alle werkplekken van de desbetreffende organisatie getoond. Als in het veld Werplek een werkplek is geselecteerd, zal in het veld Type het type getoond worden dat bij de geselecteerde werkplek hoort.

Als er geen werkplek is geselecteerd dan is het mogelijk om alle mogelijke typen te kiezen.



Om het werkplek en activiteit pull-down menu te kunnen vullen is de volgende methode nodig:

```
SomeDescriptions := TDepartment(TheDuty.Department).GetAllWorkstations;
```

Als een werkplek is geselecteerd mag het niet mogelijk zijn om type te wijzigen. Het veld Type zal daarom ge-disabled worden, zodra er een werkplek is geselecteerd. Dit zal gedaan worden met behulp van JavaScript.

Omdat het niet toegestaan is om een werkplek te kiezen die geen activiteitstype bevat. Zullen werkplekken zonder activiteitstype niet getoond worden in het veld Werkplek.

### **Veld Type**

Voor het vullen van het veld Type is de volgende methode nodig:

```
AllActivityTypes := TDepartment(TheDuty.Department).GetAllActivityTypes;
```

Constraints:

- Medewerker mag een dienst aanpassen / wijzigen, zolang de dienst de status 'Gepubliceerd' heeft.
- Medewerker kan geen dienst realiseren waar hij niet aan begonnen is (datum huidige dag < datum van de te accorderen dienst);
- Medewerker mag geen activiteit uit een dienst verwijderen als het de enige activiteit van die dienst betreft;
- De velden Begin en Eind moeten geldige waarden bevatten. Het veld is als volgt opgebouwd, uren ( uu ), een dubbele punt ( : ) en minuten ( mm ), dus uu:mm. De waarden van uu gaan van 00 t/m 24 en de waarden van mm gaan van 00 t/m 59. Deze controle wordt aan server-zijde gedaan. Bij een eventuele fout, zal er een foutmelding verschijnen.
- Medewerker mag geen diensten van collega's kunnen bewerken.

### **Controles:**

- Door middel van het Employee-id dat in Shift wordt bijgehouden kan gecontroleerd worden of de medewerker de desbetreffende dienst alleen mag bekijken of ook mag wijzigen;
- Als de datum van de dienst groter is dan de datum van de huidige dag dan mag de medewerker de dienst alleen bekijken en niet wijzigen;
- Als de status van een dienst ongelijk aan 'Gepubliceerd' is dan mag de medewerker de dienst niet wijzigen. Hij mag de dienst dan alleen bekijken.

### **3.1.3 Testen functies 1e increment**

#### **Test Wachtwoord wijzigen**

1. Alle velden leeg. Klik op de button 'wijzigen':

**Verwachte uitkomst:** "Om uw wachtwoord te wijzigen moet u alle velden invullen." + geen wijziging in database.

2. Veld Oud wachtwoord leeg laten en in overige velden wel iets invullen. Klik op de button 'wijzigen':

**Verwachte uitkomst:** "Wachtwoord kan niet worden gewijzigd, want u heeft niets ingevuld in het veld Oud wachtwoord." + geen wijziging in database.

3. In het veld Oud wachtwoord niet het juiste wachtwoord invullen. Klik op de button 'wijzigen':

**Verwachte uitkomst:** "Het oude wachtwoord is onjuist. Vul wachtwoord opnieuw in." + geen wijziging in database.

4. In het veld Oud wachtwoord het juiste wachtwoord invullen, in het veld Nieuw wachtwoord leeg laten. Klik dan op de button 'wijzigen':  
**Verwachte uitkomst:** "Wachtwoord kan niet worden gewijzigd, want u moet het veld Nieuw wachtwoord invullen." + geen wijziging in database.
5. In het veld Oud wachtwoord het juiste wachtwoord invullen, in het veld Nieuw wachtwoord een aantal waarden invullen en in het veld Bevestig nieuw wachtwoord niets invullen. Klik op de button 'wijzigen':  
**Verwachte uitkomst:** "Wachtwoord kan niet worden gewijzigd, want u heeft niets ingevuld in het veld Bevestig nieuw wachtwoord." + geen wijziging in database.
6. In het veld Oud wachtwoord het juiste wachtwoord invullen, in het veld Nieuw wachtwoord en in het veld Bevestig nieuw wachtwoord dezelfde combinatie van minder dan 5 (willekeurige) karakters invullen. Klik op de button 'wijzigen':  
**Verwachte uitkomst:** "Het wachtwoord moet minimaal 5 karakters hebben. Voer nieuw wachtwoord in." + geen wijziging in database.
7. In het veld Oud wachtwoord het juiste wachtwoord invullen, in het veld Nieuw wachtwoord en in het veld Bevestig nieuw wachtwoord niet dezelfde combinatie van invoer geven. Klik op de button 'wijzigen':  
**Verwachte uitkomst:** "De invoer van de velden Nieuw wachtwoord en Bevestig nieuw wachtwoord zijn niet gelijk aan elkaar. Vul nieuwe waarden in." + geen wijziging in database.
8. In het veld Oud wachtwoord het juiste wachtwoord invullen, in het veld Nieuw wachtwoord en in het veld Bevestig nieuw wachtwoord dezelfde combinatie van karakters invoeren. Klik op de button 'wijzigen':  
**Verwachte uitkomst:** "Uw wachtwoord is gewijzigd." + wijziging opgeslagen in database.

### Test Uren verantwoord

pre-condities:

- Een dienst heeft altijd minimaal één activiteit;
- Als een dienst exact één activiteit heeft is het niet mogelijk deze te verwijderen.
- Elke werkplek heeft precies één activiteitstype.

post-conditie:

Zodra de medewerker op annuleren klikt, verandert er niets aan de dienst. Eventuele wijzigingen worden niet opgeslagen.

Testcases:

1. Medewerker klikt op een dienst met datum groter dan de huidige dag.  
**Verwachte uitkomst:** Medewerker krijgt een overzicht van de dienst. Hij kan geen gegevens wijzigen. De Opslaan-button is gedisablend en om het scherm te sluiten kan hij op het kruisje of op de Annuleren-button klikken.
2. Medewerker klikt op een dienst met status ongelijk aan 'Gepubliceerd'.  
**Verwachte uitkomst:** Medewerker krijgt een overzicht van de dienst. Hij kan geen gegevens wijzigen. De Opslaan-button is gedisablend en om het scherm te sluiten kan hij op het kruisje of op de Annuleren-button klikken.
3. Medewerker klikt op een dienst van een collega.

- Verwachte uitkomst:** Medewerker krijgt een overzicht van de dienst. Hij kan geen gegevens wijzigen. De Opslaan-button is gedisablend en om het scherm te sluiten kan hij op het kruisje of op de Annuleren-button klikken.
4. Medewerker klikt op Annuleren of op het kruisje rechtsboven in het scherm.  
**Verwachte uitkomst:** Het scherm wordt gesloten en eventuele wijzigingen zullen ongedaan gemaakt worden. Toevoeg- en/of verwijderacties zijn definitief.
  5. Medewerker klikt op Opslaan en heeft pop-up scherm voor zich. Hij klikt vervolgens op 'Cancel' of op het kruisje.  
**Verwachte uitkomst:** Pop-up scherm verdwijnt en medewerker heeft de mogelijkheid om nog wijzigingen aan te brengen in de dienst.
  6. Medewerker klikt op Opslaan en heeft pop-up scherm voor zich. Hij klikt op 'Ok'.  
**Verwachte uitkomst:** Het pop-up scherm verdwijnt en alle wijzigingen worden opgeslagen en medewerker krijgt zijn rooster te zien.
  7. Medewerker heeft niets of alleen spatie(s) in het veld Naam ingevoerd.  
**Verwachte uitkomst:** Medewerker krijgt een melding dat de dienst niet opgeslagen kan worden omdat één of meerdere naam-velden geen geldige inhoud bevatten.
  8. Medewerker wil gegevens in het veld werkplek of in veld type aanpassen.  
**Verwachte uitkomst:** Niet mogelijk, want deze velden kunnen niet ge-edit worden.
  9. Medewerker selecteert een werkplek in het veld werkplek.  
**Verwachte uitkomst:** In het veld type wordt het type getoond dat behoort bij de geselecteerde werkplek en het veld type kan niet ge-edit worden.
  10. Medewerker heeft in het veld werkplek gekozen voor 'Geen'.  
**Verwachte uitkomst:** In het veld type kan de medewerker nu het type selecteren die hij wil.
  11. Medewerker vult in een tijdveld (begin en/of eind) een ander karakter in dan zou moeten.  
**Verwachte uitkomst:** Medewerker krijgt een melding in zijn scherm dat hij geen geldige tijd heeft ingevoerd. De dienst wordt niet opgeslagen en medewerker heeft de mogelijkheid om zijn fout ongedaan te maken.
  12. Medewerker klikt op een toevoegen-icoontje.  
**Verwachte uitkomst:** Er wordt een activiteit toegevoegd achter de regel van waar op het icoontje is geklikt. De begintijd van deze activiteit is gelijk aan de eindtijd van de voorgaande activiteit en de eindtijd = begintijd + 1 uur. Deze eindtijd is gelijk aan de begintijd van de opvolgende activiteit. De toevoeging is definitief.
  13. Medewerker klikt op een verwijderen-icoontje.  
**Verwachte uitkomst:** De activiteit waarvan het verwijderen-icoontje is aangeklikt wordt verwijderd. De verwijdering is definitief.
  14. Medewerker wil een ander scherm in de web-interface dan het dienstenscherm bewerken.  
**Verwachte uitkomst:** Niet mogelijk. Het is alleen mogelijk om het dienstenscherm te bewegen, zodat tekst achter het scherm zichtbaar wordt.

15. Medewerker vult bij minimaal één activiteit in het veld Naam niets in, dus veld Naam is helemaal leeg.

**Verwachte uitkomst:** Medewerker krijgt melding: "U heeft niets of alleen spaties ingevuld in één van de naamvelden. Er is niet gewijzigd".

16. Medewerker vult bij minimaal één activiteit in het veld Naam alleen spatie(s) in.

**Verwachte uitkomst:** Medewerker krijgt melding: "U heeft niets of alleen spaties ingevuld in één van de naamvelden. Er is niet gewijzigd".

### 3.2 Tweede increment

In het tweede increment worden de volgende functies beschreven: Overzicht mutaties en de medewerker heeft de mogelijkheid om aan te geven dat hij de mutaties heeft gezien.

#### 3.2.1 Overzicht mutaties (9754)

In het overzicht mutaties worden alle mutaties getoond die de planner heeft aangebracht in het gepubliceerde rooster van de ingelogde medewerker. De medewerker heeft de mogelijkheid om per activiteit aan te geven dat hij een mutatie heeft gezien. Om aan te geven dat hij alle wijzigingen heeft gezien kan hij op de 'Alles gezien'-button klikken. Dan worden alle mutaties aangevinkt.

Opbouw pagina:

Bovenaan de pagina komt een datetimesticker, zoals die getoond wordt in het rooster-menu. De medewerker kan hiermee een periode selecteren. Hij krijgt dan een overzicht van alle mutaties die in deze periode hebben plaatsgevonden.

Voor het opvragen van de datetimesticker is de volgende functie nodig:

GetIntervalSelector(ABeginTime, AnEndTime) uit BaseWebPage.pas

Onder deze datetimesticker worden de mutaties getoond.

Per dag waarop mutaties hebben plaatsgevonden zal een tabel gegenereerd worden.

De tabel wordt als volgt opgebouwd: een header en de data.

Voor het tonen van de header wordt de functie HeaderMutationTable gebruikt.

Voor het tonen van de data in de tabel wordt de functie DataMutationTable gebruikt.

Overzicht mutaties ziet er als volgt uit:

Boven elke dag zal de datum van de dienst en de dag getoond worden. In het overzicht worden de gegevens in de onderstaande volgorde beschreven:

Tijdstip mutatie, Type mutatie, Oude dienst, Dienst locatie, Nieuwe dienst, Dienst locatie, Van, Doorgevoerd door en de checkbox.

Al deze velden zullen niet aangepast kunnen worden door de medewerker.

Als je met je muis over een dienstnaam gaat krijg je door middel van een hinttekst te zien wat de dienstitijden zijn.

#### Één mutatie gezien

Achter elke mutatie komt een checkbox. Als een medewerker wil aangeven dat hij een mutatie heeft gezien kan hij de checkbox aanklikken. Zodra een checkbox wordt aangeklikt (onclick-event) wordt de JavaScript-functie AcceptMutation() aangeroepen en wordt de actie bevestigd / opgeslagen in de database.

#### Alle mutaties gezien

Als de medewerker wil aangeven dat hij alles heeft gezien kan hij op de Alles gezien-button klikken.

Zodra op deze button wordt geklikt (onclick-event) wordt de JavaScript-functie `AcceptAllMutations()` aangeroepen. Alle mutaties die nog niet aangevinkt zijn worden aangevinkt. Deze actie wordt opgeslagen in de database.

### **Opslaan van mutatie gezien**

Er is gekozen voor direct opslaan omdat zodra een mutatie is aangeklikt hij dit heeft gezien. Bovendien zou het voor de planner nog niet duidelijk zijn of een medewerker de mutaties heeft gezien.

Eén exchange kan van toepassing zijn voor twee medewerkers. In dit geval moeten beide medewerkers aan kunnen geven of ze de mutatie hebben gezien. Hier zijn nog geen velden voor beschikbaar. Daarom zullen er twee velden toegevoegd moeten worden aan de tabel `ROS_EXCHANGES`. De velden zijn van het type Boolean en heten `AcceptedFrom`, `AcceptedTo`.

Mutatiemogelijkheden zijn:

1. Een dienst wordt vanuit 'In te plannen diensten' toegevoegd aan het gepubliceerde rooster van een medewerker;
2. Een medewerker krijgt een dienst van een andere medewerker toegewezen op een dag dat er geen dienst ingepland is;
3. Een dienst is verwijderd van het dienstrooster van een medewerker;
4. Een dienst van een medewerker wordt geruild met een dienst van een andere medewerker door de planner (wijziging).

In situatie één krijgt veld `AcceptedFrom` automatisch de waarde Y en is de waarde `AcceptedTo` N.

In situatie twee heeft het veld `AcceptedFrom` de waarde Y en `AcceptedTo` automatisch de waarde N.

In situatie drie heeft het veld `AcceptedFrom` automatisch de waarde N en `AcceptedTo` de waarde Y.

In situatie vier hebben de velden `AcceptedFrom` en `AcceptedTo` automatisch de waarde N.

### **Ophalen van mutaties**

Voor het ophalen van de mutaties per dag is de volgende functie benodigd:

`function GetExchanges (AnBeginTime: TdateTime, AnEndTime: TdateTime):`

`TDescriptions;` uit `resource.pas`

#### **3.2.2 Testen Overzicht mutaties**

1. Medewerker klikt op de link Overzicht mutaties  
**Verwacht resultaat:** Medewerker krijgt een overzicht van alle mutaties vanaf de huidige dag + 14 dagen. Als er geen nieuwe mutaties zijn, wordt de melding "Er zijn geen mutaties." getoond.
2. Medewerker klikt in een checkbox bij een niet-aangevinkte mutatie  
**Verwacht resultaat:** Er wordt een vinkje in de checkbox gezet en wordt opgeslagen in de database. Checkbox wordt gedisable.
3. Medewerker klikt in een checkbox bij een aangevinkte mutatie  
**Verwacht resultaat:** Niet mogelijk, want checkbox is gedisable.
4. Medewerker klikt op de Alles gezien-button  
**Verwacht resultaat:** Alle mutaties die niet aangevinkt zijn, krijgen een vinkje. Gegevens worden opnieuw getoond, alle checkboxen zijn gedisable en Alles gezien-button is gedisable.
5. Medewerker selecteert een bepaalde periode

**Verwacht resultaat:** Medewerker krijgt een overzicht van alle mutaties die in de geselecteerde periode hebben plaatsgevonden.

6. Medewerker vult ongeldige waarden in, in de van:- en/of tot:-veld van de date-timepicker.

**Verwacht resultaat:** Medewerker krijgt de melding "De ingevoerde datum is ongeldig. Gebruik het volgende formaat: dd-mm-jjjj" op zijn scherm.

Vervolgens worden de van- en tot-datum aangepast.

7. Medewerker selecteert een periode langer dan één jaar.

**Verwacht resultaat:** Medewerker krijgt de melding "De door u geselecteerde duurt langer dan één jaar, selecteer een periode van maximaal één jaar." op zijn scherm. Vervolgens worden de van- en tot-datum aangepast.

8. Medewerker gaat met zijn muis over een dienstnaam.

**Verwacht resultaat:** In een hinttekst wordt de begin- en eindtijd van de dienst getoond.

### 3.3 Derde increment

In het derde increment worden de volgende functies beschreven: Rooster gezien en overzicht van de ingevoerde wensen + verlofaanvragen.

#### 3.3.1 Rooster gezien (10087)

De functionaliteit Rooster gezien bestaat uit een button die in de pagina "Mijn rooster" zal worden getoond zodra een (deel van een) rooster wordt getoond met een datum groter dan de datum die opgeslagen is in de database. De datum in de database is de datum van wanneer een medewerker voor het laatst zijn rooster heeft gezien.

Zodra een medewerker op de button klikt wordt de einddatum van zijn periode selectie opgeslagen.

Om de datum daadwerkelijk op te kunnen slaan moet er een nieuw veld van het type date/time in de tabel ROS\_RESOURCES (persoonsgegevens) toegevoegd worden. Naam van het veld is: ACCEPT\_ROSTERDATE van het type DATE.

In Resource.pas wordt de property AcceptedRosterDate en de functies GetAcceptedRosterDate en SetAcceptedRosterDate toegevoegd, zowel aan de server- als aan de client-kant.

Constraints:

1. Geselecteerde datum wordt opgeslagen als deze groter is dan de opgeslagen datum;
2. Als het verschil tussen de opgeslagen datum en de geselecteerde einddatum kleiner is dan 7 (zeven) dagen en aan constraint 1 is voldaan dan wordt de geselecteerde einddatum opgeslagen;
3. Als het verschil tussen de opgeslagen datum en de geselecteerde einddatum groter is dan 7 (zeven) dagen en aan constraint 1 is voldaan wordt een melding getoond.

Als de medewerker op OK klikt dan wordt de geselecteerde einddatum opgeslagen.

Als de medewerker op Cancel klikt dan wordt de geselecteerde einddatum opgeslagen;

In EmployeeRosterWebPage.pas worden de volgende TagStrings toegevoegd:

- ACCEPT\_ROSTERBTN = button met tekst "Rooster gezien";
- CONFIRMDATE = bericht in JavaScript-functie SaveDate(). Als de waarde true is wordt de de einddatum opgeslagen in ACCEPT\_ROSTERDATE.

Als op de button "Rooster gezien" wordt geklikt wordt in de JavaScript-functie SaveDate() gecontroleerd of het verschil tussen de geselecteerde einddatum en ACCEPT\_ROSTERDATE groter is dan 7.

Zo ja, dan wordt het bericht CONFIRMDATE getoond.

Zo nee, dan wordt gecontroleerd of het verschil kleiner of gelijk aan 7 is. Als dit zo is dan krijgt ACCEPT\_ROSTERDATE de waarde van de geselecteerde einddatum.

Het opslaan van de geselecteerde einddatum (AnEndTime) wordt door de onderstaande methode gedaan:

```
// AnEndTime is de einddatum uit de Periode selectie tabel  
SessionEmployee.AcceptedRosterDate := AnEndTime;
```

Positie van de TagStrings in EmployeeRosterWebPage.html:

- ACCEPT\_ROSTERBTN komt in een aparte tabel onder EMPLOYEEEROSTERTABLE te staan en wordt alleen getoond als aan constraint 1 is voldaan.

Als op ACCEPT\_ROSTERBTN wordt geklikt wordt de JavaScript-functie SaveDate() aangeroepen.

### **3.3.2      Overzicht wensen + verlofaanvragen (10088)**

Deze functie is nodig om de gebruiker te laten zien dat een door hem ingevoerde wens of verlofaanvraag daadwerkelijk is verzonden, zodra hij op de verzend-button drukt.

Door op de tekst-hyperlink "Nieuwe wens invoeren" te klikken wordt de Javascript-functie InsertWish(FormNr) aangeroepen.

FormNr = 1, 2 of 3.

Voor het opvragen van de wel / niet werken in een periode is de volgende RequirementType / id nodig: WorkIntervalWish = FormNr 1

Voor het opvragen van de dienstwensen is de volgende RequirementType / id nodig: DutyNameWish = FormNr 2

Voor het opvragen van de verlofwensen is de volgende RequirementType / id nodig: LeaveWish = FormNr 3

Om de wensen van de ingelogde persoon op te halen wordt de onderstaande methode gebruikt:

AnEmployee.WishList

De wensen worden opgevraagd met de functie GetRequirementList.

Met de property Count wordt het aantal Requirements (wensen) van de ingelogde persoon opgevraagd.

Met de property Items[..] worden de Requirements opgevraagd.

Met behulp van de functie RequirementType kan bekeken worden om wat voor soort Requirement het gaat, soort Requirement kan zijn: WorkIntervalWish, DutyNameWish of LeaveWish.

De Requirements worden gegroepeerd per soort op het scherm getoond.

Om de inhoud van een Requirement te bekijken wordt de functie Params gebruikt.

In WishWebPage.html tussen de body-tags wordt de TagString WISHTABLE gezet.

In WishWebPage.pas wordt de TagString WISHTABLE gezet. Als TagString de waarde WISHTABLE heeft wordt de functie ShowWishTable aangeroepen en

wordt een tabel gemaakt van de eventuele wensen van de medewerker. Voor het soort waarvan geen wensen zijn, wordt onder de header de tekst "Geen wensen" getoond.

### **Nieuwe %wens invoeren**

Zodra op één van hyperlinks geklikt wordt zal de pop-up scherm "InsertWish?Wish=FormNr" getoond worden. In dit scherm heeft de medewerker de mogelijkheid om aan te geven wat zijn wens is. Het verschil met de oude situatie zal zijn dat de naam "Verzend" gewijzigd wordt in "Opslaan" en er een button "Annuleren" toegevoegd wordt.

Als het om de wens wel / niet werken in een periode gaat, is: FormNr 1.

Als het om een dienstwens gaat, is: FormNr 2.

Als het om een verlofwens gaat, is: FormNr 3.

### **3.3.3 Testen Rooster gezien (10087)**

Een aantal testcases om de functionaliteit Rooster gezien te testen of het aan de gewenste eisen voldoet.

1. Medewerker klikt op de link "Mijn rooster" en einddatum is kleiner dan de datum opgeslagen in de database.  
**Verwacht resultaat:** Medewerker krijgt overzicht van zijn rooster en button "Alles gezien" is ge-disabled.
2. Medewerker klikt op de link "Mijn rooster" en einddatum is gelijk of groter dan de datum opgeslagen in de database.  
**Verwacht resultaat:** Medewerker krijgt overzicht van zijn rooster en button "Alles gezien" is enabled, dus de mogelijkheid om op de button te klikken.
3. Medewerker klikt op de "Alles gezien" button (= enabled) en geselecteerde periode is korter of gelijk aan 5 (vijf) weken.  
**Verwacht resultaat:** De einddatum uit de periode selectie tabel wordt opgeslagen in de database.
4. Medewerker klikt op de "Alles gezien" button (= enabled) en geselecteerde periode is langer dan 5 (vijf) weken.  
**Verwacht resultaat:** Er wordt een melding op het scherm getoond met de tekst "Periode is langer dan een maand. Weet u zeker dat u over deze periode uw rooster gezien wilt hebben?"  
Klikt de medewerker op "Ok" dan wordt de einddatum uit de periode selectie tabel opgeslagen in de database en de "Alles gezien" button wordt gedisablend. Klikt de medewerker op "Cancel" dan wordt de einddatum niet opgeslagen en blijft de "Alles gezien" button enabled.
5. Medewerker klikt op de "Alles gezien" button en de button is gedisablend.  
**Verwacht resultaat:** Er gebeurt niets.

### **3.3.4 Testen Overzicht wesen + verlofaanvragen (10088)**

Aantal testcases om de functie Overzicht wesen + verlofaanvragen te testen of het voldoet aan de gestelde eisen.

1. Medewerker klikt op de link "Wensen".  
**Verwacht resultaat:** Medewerker krijgt een overzicht van zijn ingevoerde wensen die gelden voor de huidige dag of later. De wensen zijn gesorteerd op soort wens. Als er bij een bepaalde categorie geen wens(en) zijn dan wordt in die categorie de tekst "Geen wensen" getoond.
2. Medewerker klikt op de hyperlink "Nieuwe %wens invoeren".



**Verwacht resultaat:** Een pop-up scherm wordt getoond. De medewerker heeft de mogelijkheid om zijn wens in te voeren.

3. Medewerker klikt op "Ok" button in het pop-up scherm.

**Verwacht resultaat:** Wens wordt opgeslagen en wens wordt getoond in het Overzicht wensen. Het pop-up scherm wordt gesloten.

4. Medewerker klikt op "Cancel" button of op het kruisje in het pop-up scherm.

**Verwacht resultaat:** Wens wordt niet opgeslagen en niet getoond in het Overzicht wensen. Het pop-up scherm wordt gesloten.

## 4. Class diagram

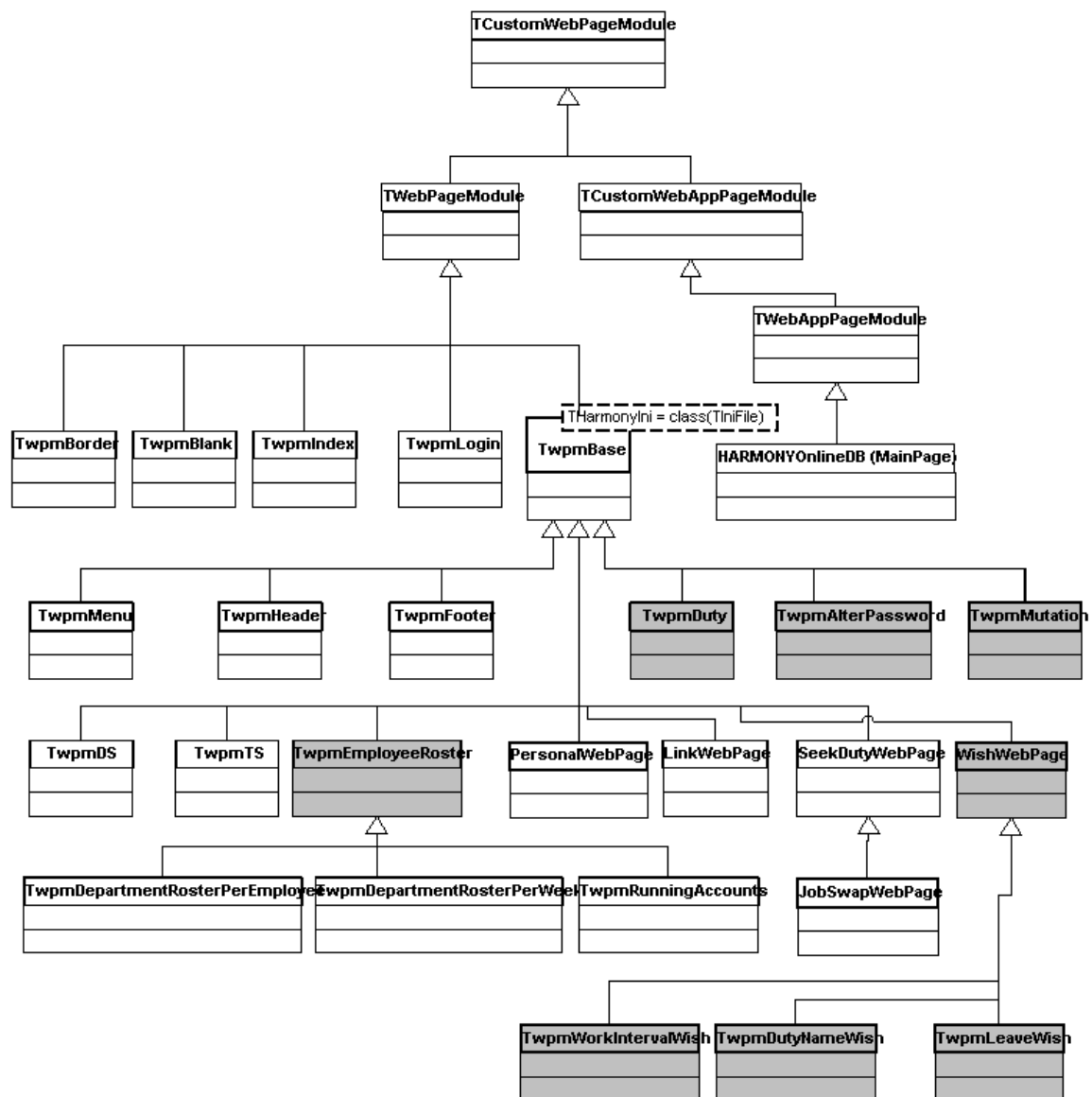
In dit hoofdstuk wordt door middel van class diagrams aangegeven hoe de structuur van HARMONY-WebAccess eruit ziet.

Legenda voor class diagrams:

- Een plusje (+) betekent dat de visibility van het attribuut, functie of procedure public is;
- Een streepje (-) betekent dat de visibility van het attribuut, functie of procedure private is;
- Een hekje (#) betekent dat de visibility van het attribuut, functie of procedure protected is.

### 4.1 Totaaloverzicht

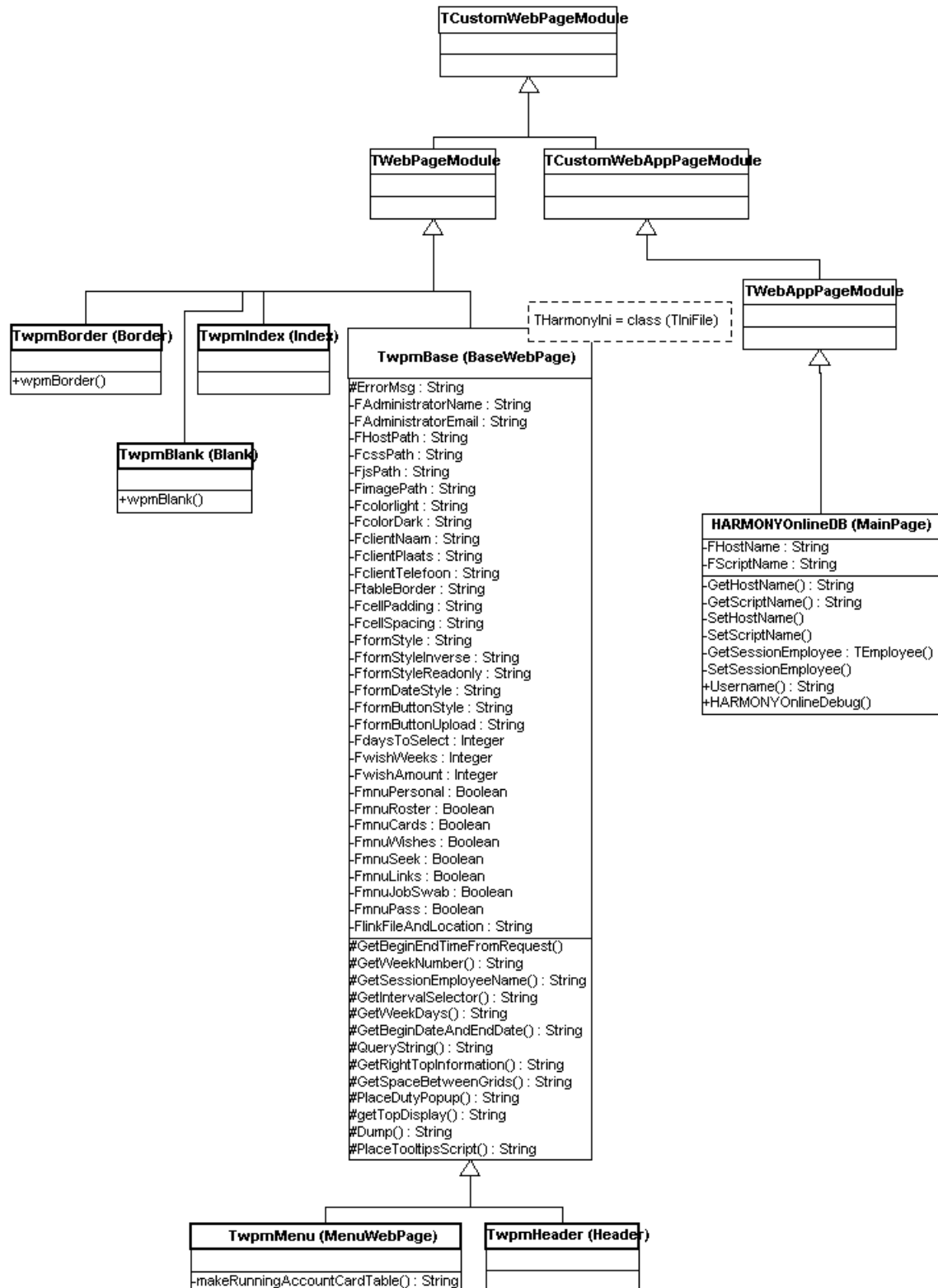
In dit overzicht worden alle classes getoond. De classes, functionaliteiten die nieuw zijn hebben donkere achtergrond.



Class diagram van alle classes

## 4.2 Overzicht Toplaag

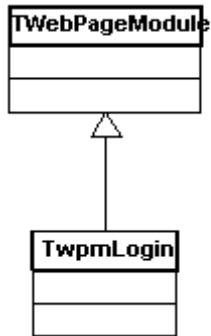
In het overzicht toplaag wordt de basis van HARMONY WebAccess getoond.



*Class diagram van Toplaag (uitgebreidere beschrijving van de classes)*

### 4.3 Overzicht Login

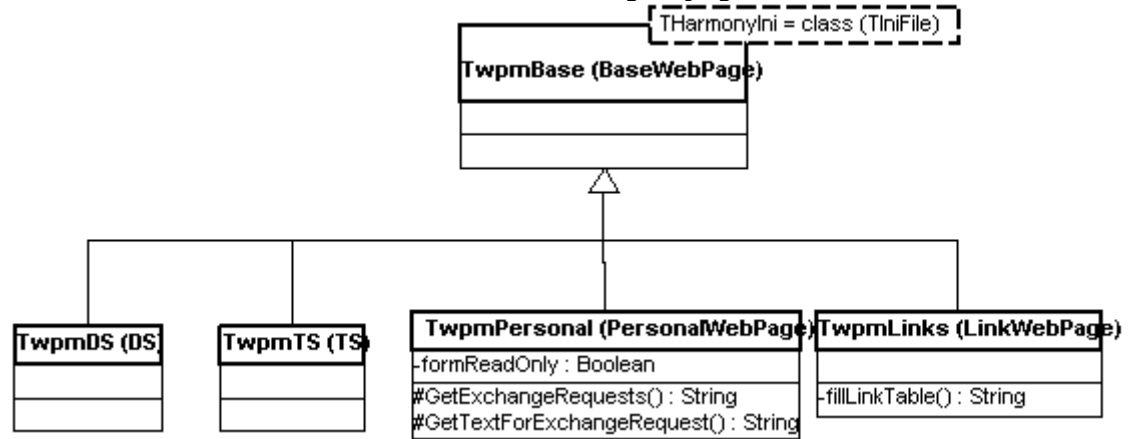
Dit overzicht toont de classes die de login-functionaliteit gebruikt.



Class diagram van Login

### 4.4 Overzicht ongewijzigde classes (datum/tijd, persoonlijk, links)

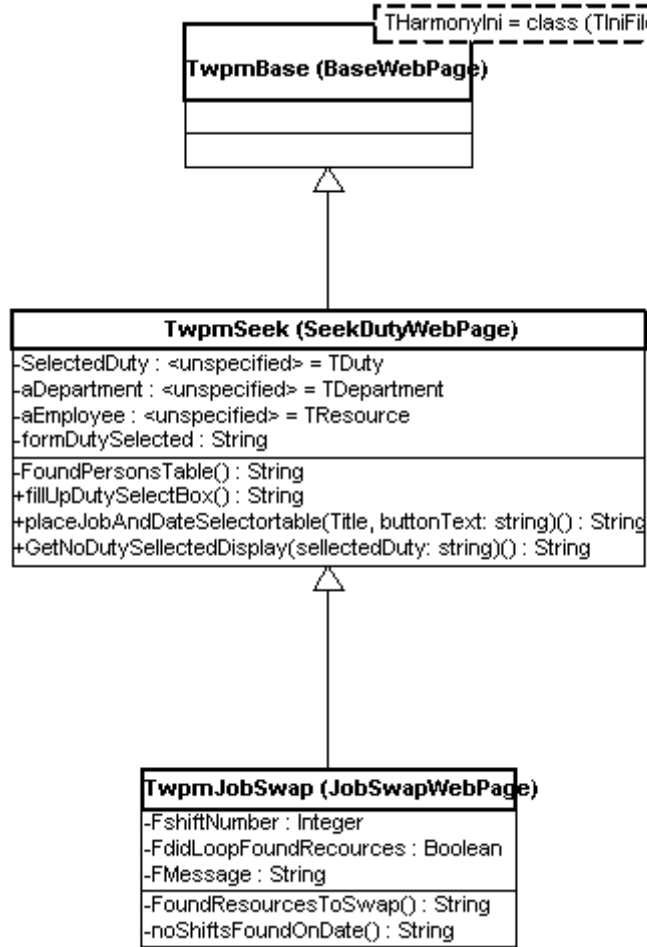
De classes in dit overzicht zullen niet worden gewijzigd.



Class diagram van de ongewijzigde classes

## 4.5 Overzicht Zoeken en Ruiling aanvragen

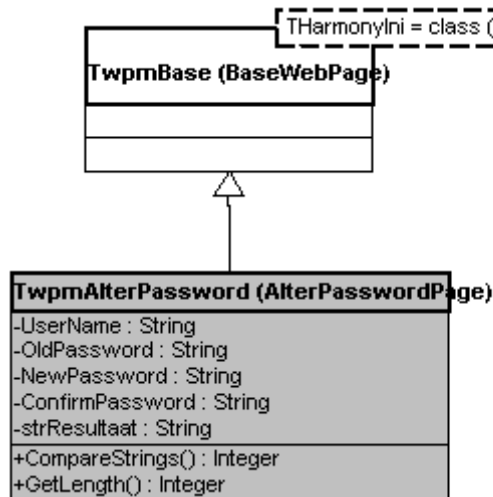
Dit overzicht toont de classes die gebruikt worden bij het zoeken naar diensten en bij het aanvragen van ruilingen.



Class diagram van Zoeken & Ruiling aanvragen

## 4.6 Overzicht Wachtwoord wijzigen

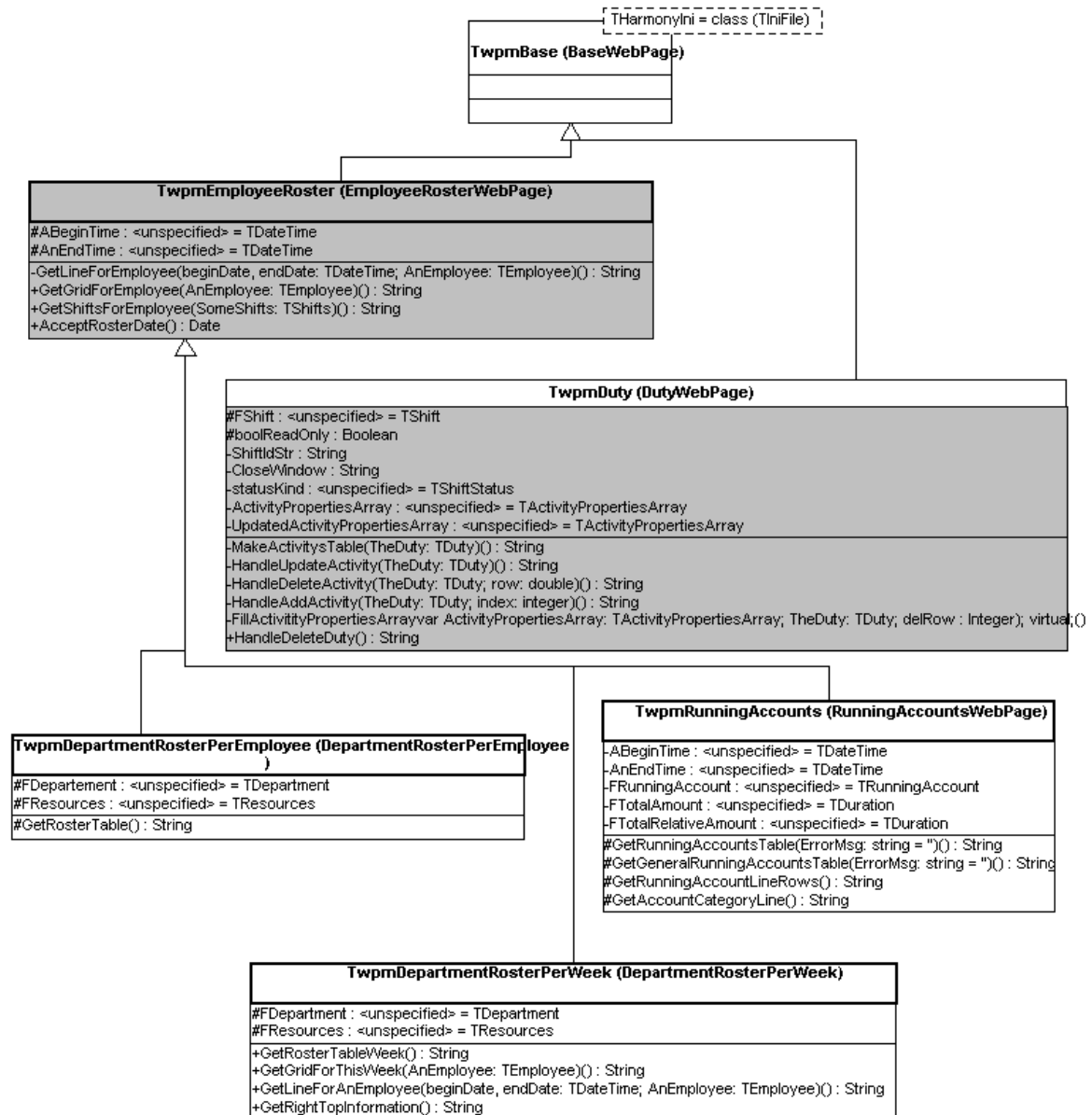
Dit overzicht laat de class zien gebruikt wordt bij het wijzigen van het wachtwoord. De medewerker kan zelf zijn wachtwoord wijzigen.



Class diagram van Wachtwoord wijzigen

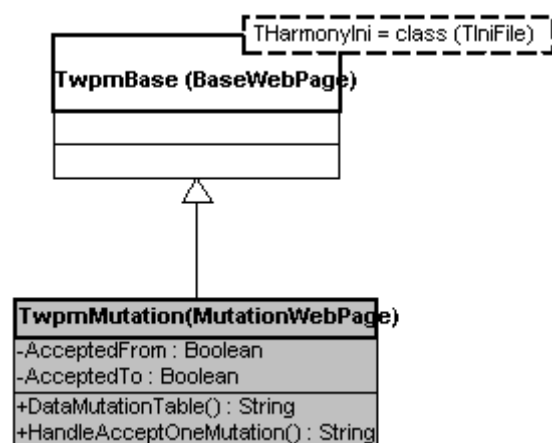
## 4.7 Overzicht Rooster

Dit overzicht toont de classes die nodig zijn voor het tonen van het individuele rooster, rooster per groep, rooster groep per week, eventuele kaarten en voor het bekijken van (gepubliceerde) diensten.

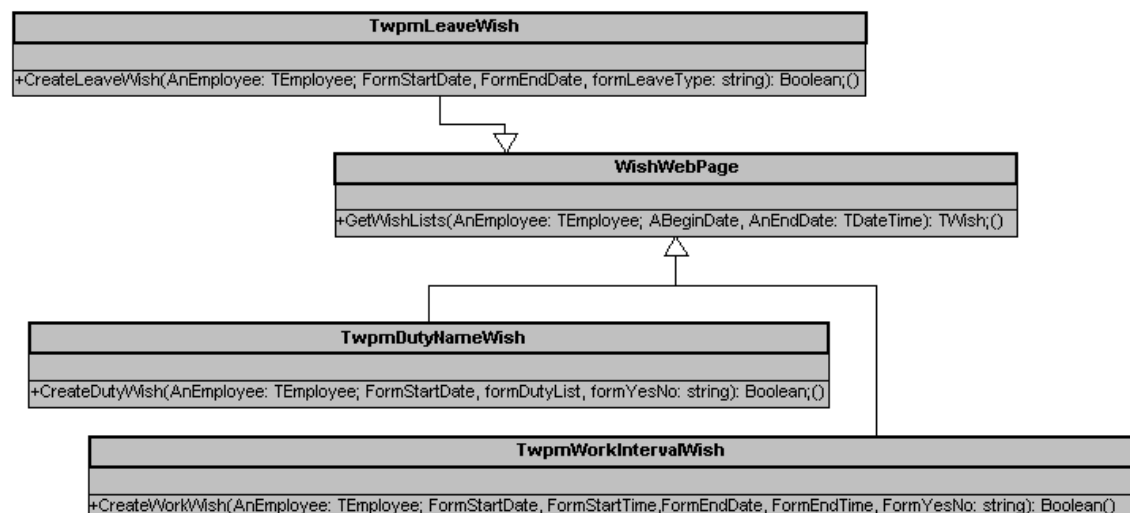


Class diagram van Roostergegevens

## 4.8 Overzicht mutaties

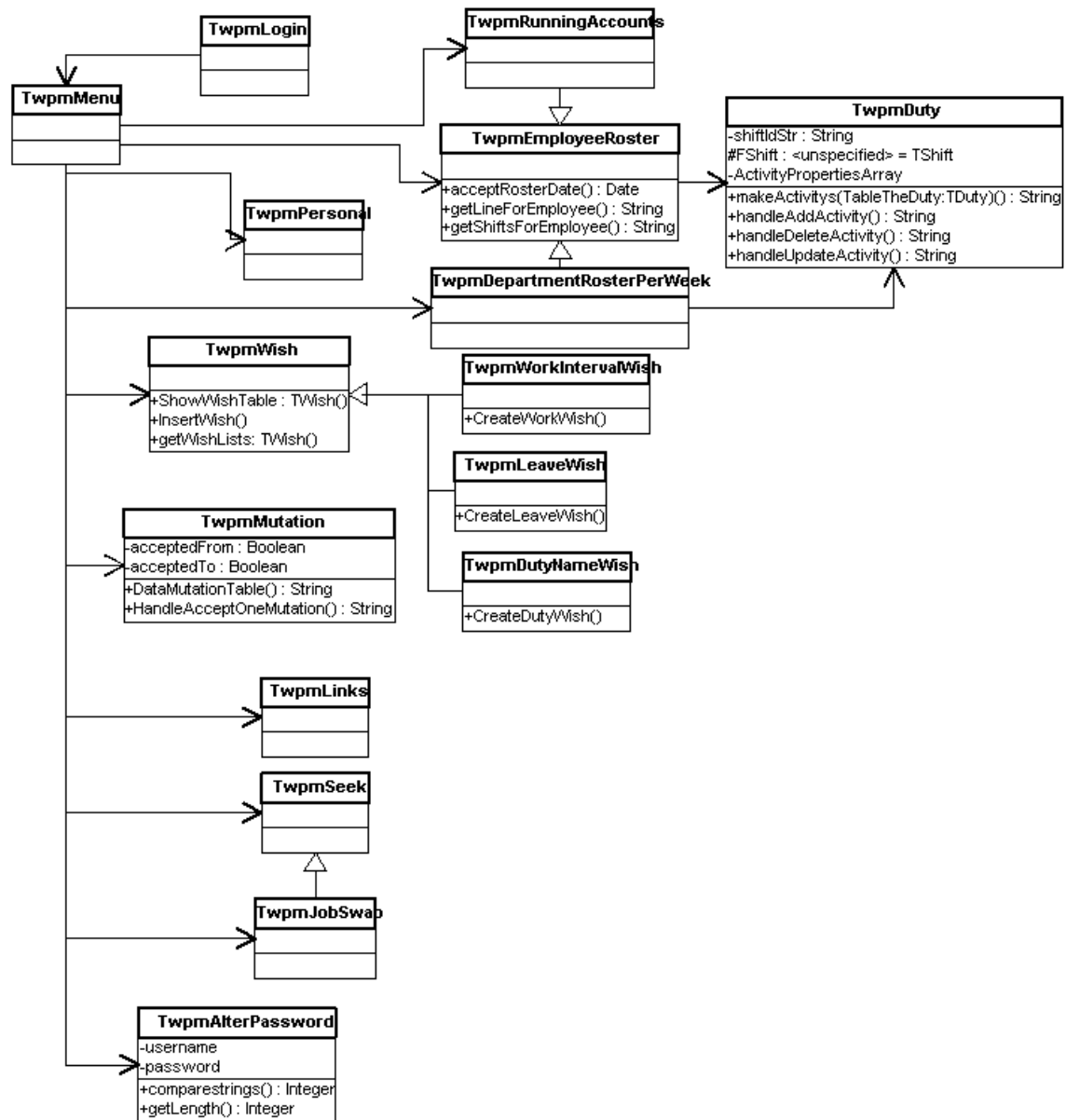


## 4.9 Overzicht wensen



#### 4.10 Overzicht HARMONY WebAccess

Dit class diagram toont de relaties tussen alle classes van HARMONY WebAccess. De relaties met TwpmMenu zijn geen echte relaties, deze relatie geeft alleen aan dat vanuit TwpmMenu de webpagina opent die bij die class hoort.





#### 4.11 Overzicht HARMONY

HARMONY WebAccess stuurt informatie tussen de browser en HARMONY door, waarbij een vertaalslag plaatsvindt. Een HTTP-request van de browser wordt omgezet in DCOM-request en een DCOM-response wordt omgezet in een HTTP-response. Het class diagram Overzicht HARMONY toont de classes die voor het gebruik van HARMONY van belang zijn.

