

# Afstudeerscriptie

HourPower urenregistratie



Auteur:	Jeffrey Veldkamp
Studentnummer:	1512926
Datum:	15 december 2009
Versie:	1.0
Afstudeerbedrijf:	n.v.t.
Bedrijfsbegeleider:	G�rard van Elsen
Eerste examiner:	Jeroen Weber
Onderwijsinstelling:	Hogeschool Utrecht
Opleiding:	Informatica

## Voorwoord

Deze scriptie is geschreven ter afsluiting van de Bachelor Informatica aan de Hogeschool Utrecht.

Mijn interesse in dit onderwerp is voortgekomen uit een project wat wij tijdens het eerste leerjaar uit hebben gevoerd. Wij mochten zelf een project bedenken om te realiseren en hier is toen het eerste idee voor HourPower ontstaan.

Op dat moment werd bij Feka ICT & Office gebruik gemaakt van AccountView om de gewerkte uren te registreren. Het was mij echter opgevallen dat je perse in moest loggen op een centrale server voordat de uren geregistreerd konden worden. Gezien het feit dat er nogal wat personeel in de buitendienst werkzaam is bij dit bedrijf bedacht ik me dat een online applicatie hiervoor een uitkomst zou kunnen bieden. Niet langer hoeft het personeel de hele dag te onthouden wat ze gedaan hebben, dit kunnen ze dan ter plekke doen.

Het schrijven van deze scriptie en het uitvoeren van dit project heb ik als een plezierige en leerzame tijd ervaren. Graag wil ik van deze gelegenheid gebruik maken om een aantal mensen te bedanken.

Allereerst wil ik Gérard van Elsen bedanken voor het feit dat hij de rol van bedrijfsbegeleider op zich wilde nemen. Doordat wij een eigen afstudeeropdracht hebben uitgevoerd moesten wij zelf op zoek naar een bedrijfsbegeleider. Hij heeft deze rol geaccepteerd en ons tijdens het project goede adviezen gegeven.

Ten tweede wil ik Jeroen Weber bedanken voor de begeleiding van school uit tijdens dit project. Niet alles is altijd even duidelijk geweest maar hij gaf altijd opheldering. Ook de feedback op mijn scriptie heeft geholpen om mijn scriptie naar een hoger niveau te tillen.

Naast de beide begeleiders wil ik Daniël van Zoelen bedanken voor de samenwerking tijdens dit project. Ik heb deze ervaren als zeer prettig en leerzaam. Vooral de spellen Uno in de pauze werkte ontspannend.

Vervolgens wil ik mijn ouders (Nel en Ger Veldkamp) bedanken voor hun steun tijdens mijn gehele studie. Vooral mijn moeder was oprecht geïnteresseerd in mijn bezigheden en heeft bijgedragen in het vinden van deelnemers aan ons onderzoek naar de wensen van verschillende bedrijven.

Een speciaal dankwoord gaat uit naar mijn zusje Sharissa Veldkamp en mijn vriendin Mélanie van den Acker. Zij hebben altijd voor mij klaargestaan en geprobeerd op verschillende manieren bij te dragen aan een goed eindresultaat van mijn studie. Daarnaast hebben zij regelmatig voor wat afleiding gezorgd om mijn hoofd even helemaal leeg te krijgen.

Ook wil ik hier graag de verschillende personen bedanken voor hun medewerking aan ons onderzoek naar de wensen van verschillende bedrijven. In het bijzonder Mario Ebbelaar van Accounting Plaza die ons in contact heeft gebracht met meerdere bedrijven.

Utrecht, december 2009.

Jeffrey Veldkamp.

## Managementsamenvatting

De aanleiding voor dit project komt voort uit een project wat tijdens het eerste leerjaar op de FNT van de Hogeschool Utrecht is uitgevoerd. Destijds was ik werkzaam bij een bedrijf waar mensen in de buitendienst werkzaam waren, maar de urenregistratie was hier niet voor geschikt. Om uren te registreren moesten medewerkers via Remote Desktop inloggen op een centrale server. Op basis van deze ervaring is toen besloten om een online urenregistratiesysteem (HourPower) te realiseren.

Het eindresultaat van dit project werd goed ontvangen en beoordeeld met een 8 als eindcijfer en een uitnodiging om het product namens de studierichting Informatica te presenteren op de open dag. Op basis hiervan is besloten om verder te gaan met dit product.

Gelukkig biedt de Hogeschool Utrecht de mogelijkheid om een duo-opdracht uit te voeren als afstudeerproject. Van deze gelegenheid is dan ook gebruik gemaakt om het eerder uitgevoerde project nieuw leven in te blazen. Destijds zijn functionaliteiten bedacht en direct gerealiseerd. Voor dit project zijn een aantal vragen gesteld en gezocht alvorens te beginnen met het realiseren van het systeem.

De vragen die zijn onderzocht zijn:

- Bestaan er al online urenregistratiesystemen, en zo ja: wat bieden deze pakketten?  
Om hier achter te komen zijn is een zoekopdracht gestart in de grotere zoekmachines (MSN/Bing, Yahoo en Google). Tussen deze resultaten verschenen zes pakketten die steeds terugkwamen. Op basis hiervan is besloten om te bekijken wat deze pakketten voor functionaliteit aanbieden.
- Hoe kan HourPower zich onderscheiden van deze pakketten?  
Om niet één van de zoveel te zijn is bekeken hoe HourPower onderscheidend kan zijn ten opzichte van de bestaande pakketten die zijn onderzocht. Onderdelen die niet terugkwamen in de onderzochte pakketten zijn:
  - Toegang vanaf de mobiele telefoon;
  - Import mogelijkheid vanuit bestaande pakketten;
  - Koppeling tussen onze agenda en de agenda van Outlook.

Op basis van het laatste punt is een onderzoek gestart naar de haalbaarheid. Hieruit bleek dat het realiseren hiervan niet zou lukken binnen de gestelde tijd van dit project.

- Welke functionaliteiten zijn een vereiste?  
Om ervoor te zorgen dat met HourPower een pakket wordt geboden wat ook aansluit bij de behoefte van bedrijven is op basis van de twee eerder genoemde onderzoeken een lijst met functionaliteiten samengesteld. Deze lijst is door 10 bedrijven ingevuld. Zo konden worden bepaald welke functionaliteiten absolute must-haves waren en welke functionaliteiten doorgestreept konden worden.
- Welke architectuur gaat worden gebruikt om dit te realiseren?  
Het was opgevallen dat de bestaande pakketten naast urenregistratie ook andere zaken

aanboden, zoals planning en facturatie. Eén van de vereisten voor het systeem is dus uitbreidbaarheid. Behalve uitbreidbaar moet het systeem ook onderhoudbaar zijn. Met deze eigenschappen in het achterhoofd hebben zijn vier architecturen onderzocht, te weten:

- Model-View-Controller;
- Model-View-Presenter;
- Presentation-Abstraction-Control;
- Document-View.

De uitkomst hiervan was dat het MVC model in combinatie met een Front Controller voor HourPower de meest geschikte architectuur bleek.

Op basis van de resultaten van de onderzoeken is het Functioneel Ontwerp gerealiseerd. Hierin is gebruik gemaakt van de modelleringstaal UiaML. Deze modelleringstaal is bedacht door de heer Alex Jongman van de FNT en tijdens dit project in de praktijk gebruikt om de volgende onderzoeksvraag te beantwoorden: Is UiaML als modelleringstaal geschikt voor het Functioneel Ontwerp bij grote applicaties? Daniël van Zoelen geeft hier antwoord op in zijn scriptie.

Doordat deze scriptie opgeleverd moest worden voor de deadline van het project kan nog niet het volledige product besproken worden. De gerealiseerde onderdelen die in deze scriptie aan bod komen zijn:

- Hourpower aanvragen;
- Inloggen;
- Dashboard;
- Afdelingen beheren;
- Medewerkers beheren en importeren;
- Klanten beheren en importeren;
- Projecten beheren en importeren;
- Activiteiten beheren en importeren;
- Uren registreren;
- Agenda beheren;
- Systeeminstellingen.

## Inhoudsopgave

Voorwoord .....	2
Managementsamenvatting .....	3
Inleiding .....	7
1 Probleemstelling en definitieve opdrachtomschrijving .....	8
1.1 Huidige situatie.....	8
1.2 Gewenste situatie.....	8
1.3 Projectopdracht.....	8
1.3.1 Fasering .....	8
1.3.2 Planning.....	9
1.3.3 Onderzoeksvragen.....	9
1.4 Reikwijdte .....	9
2 Uitgangssituatie.....	12
2.1 Overleg projectbegeleider.....	12
2.2 Goede werkomgeving .....	12
2.3 Kennis van de opdracht .....	12
2.4 Kennis van de ontwikkeltools.....	12
3 Methoden en technieken .....	13
3.1 Projectmethode.....	13
3.2 Ontwikkelmethode.....	13
3.2.1 Watervalmethode .....	13
3.2.2 DSDM.....	13
3.3 Ontwikkeltechniek.....	14
3.3.1 Server.....	14
3.3.2 Client.....	14
3.3.3 Database .....	15
3.3.4 Versiebeheer .....	15
3.3.5 Applicatieserver.....	15
3.3.6 Conventies.....	15
4 Uitvoering en resultaten .....	16
4.1 Onderzoeksfase .....	16
4.1.1 Bestaande pakketten.....	16
4.1.2 Onderscheiden .....	17

4.1.3	Wensen van de klant .....	18
4.1.4	Software architectuur.....	19
4.1.5	Microsoft Outlook add-on.....	20
4.2	Functioneel Ontwerp.....	20
4.2.1	Sitemap View.....	21
4.2.2	Content Area View .....	22
4.2.3	Previews .....	22
4.3	Technisch Ontwerp.....	22
4.3.1	Database ontwerp .....	22
4.3.2	Sequence diagrammen.....	24
4.3.3	Structuur.....	25
4.3.4	Architectuur.....	26
4.4	Realisatie .....	27
4.4.1	Afhankelijkheden.....	28
4.4.2	Resultaten.....	29
4.5	Testen .....	44
5	Conclusies en aanbevelingen .....	45
5.1	Aanbevelingen.....	45
6	Evaluatie van de procesgang.....	47
	Lijst met afkortingen .....	48
	Lijst van figuren en tabellen .....	49
	Literatuurlijst .....	50
	Bijlagen .....	51

## Inleiding

De aanleiding voor dit project komt voort uit een project wat tijdens het eerste leerjaar op de FNT van de Hogeschool Utrecht is uitgevoerd. Destijds was ik werkzaam bij een bedrijf waar mensen in de buitendienst werkzaam waren, maar de urenregistratie was hier niet voor geschikt. Om uren te registreren moesten medewerkers via Remote Desktop inloggen op een centrale server. Op basis van deze ervaring is toen besloten om een online urenregistratiesysteem (HourPower) te realiseren.

Het eindresultaat van dit project werd goed ontvangen en beoordeeld met een 8 als eindcijfer en een uitnodiging om het product namens de studierichting Informatica te presenteren op de open dag. Op basis hiervan is besloten om verder te gaan met dit product.

Om een zo goed mogelijk systeem op te leveren zijn vooraf de volgende onderzoeksvragen vastgesteld:

- Bestaan er al online urenregistratiesystemen, en zo ja: wat bieden deze pakketten?
- Welke functionaliteiten zijn een vereiste?
- Hoe kan HourPower zich onderscheiden van deze pakketten?
- Welke architectuur gaat worden gebruikt om dit te realiseren?
- Is UiaML als modelleringstaal geschikt voor het Functioneel Ontwerp bij grote applicaties?

De laatste vraag zal in de scriptie van Daniël van Zoelen worden behandeld.

Deze scriptie is geschreven ter afsluiting van de Bachelor Informatica aan de Hogeschool Utrecht en is als volgt opgebouwd:

- **Probleemstelling en definitieve opdrachtomschrijving.** In dit hoofdstuk is aan de hand van de huidige- en gewenste situatie een projectopdracht geformuleerd, samen met een overzicht van de verschillende projectfasen.
- **Uitgangssituatie.** Hier is kort beschreven aan welke voorwaarden moet worden voldaan om het project tot een succesvol einde te brengen.
- **Methoden en technieken.** In dit hoofdstuk is uitgelegd welke methoden en technieken wij hebben gebruikt tijdens de uitvoering van het project.
- **Uitvoering en resultaten.** In dit hoofdstuk staat per fase beschreven hoe wij te werk zijn gegaan en welke resultaten dit heeft opgeleverd. Tevens zijn hier de onderzoeksvragen beantwoord.
- **Evaluatie.** In dit hoofdstuk staat een beschrijving van de procesgang tijdens het uitvoeren van het project.
- **Conclusies en aanbevelingen.** Hier staan mijn conclusies op basis van het project en mijn aanbevelingen voor eventuele vervolgprojecten.

# 1 Probleemstelling en definitieve opdrachtomschrijving

## 1.1 Huidige situatie

Internet speelt tegenwoordig een steeds grotere rol in alles wat we doen. Iedereen beschikt wel over een verbinding via pc, laptop, mobiele telefoon en soms zelfs al vanuit de koelkast.

De gekste en de mooiste dingen kunnen gerealiseerd worden dankzij het Internet. Toch zijn er nog steeds bedrijven die hun urenregistratie lokaal bijhouden. Een voorbeeld hiervan is Feka ICT & Office. Hier staat op een centrale server het pakket AccountView geïnstalleerd en het personeel moet via Remote Desktop inloggen om o.a. de uren te kunnen registreren.

Bij dit bedrijf lopen ook een aantal mensen in de buitendienst welke hierdoor verplicht zijn om de hele dag te onthouden wat zij gedaan hebben doordat zij niet de mogelijkheid hebben dit tussendoor te registreren. Gezien de technieken van nu is dit natuurlijk een achterhaald systeem en kan dit veel eenvoudiger.

Na onderzoek is gebleken dat er reeds pakketten zijn welke de mogelijkheid bieden om online uren te registreren. Om deze reden is besloten onderzoek te doen naar een aantal van deze pakketten en te bekijken welke functionaliteiten zij aanbieden. Hieruit is gebleken dat deze pakketten naast urenregistratie vaak nog meer aanbieden, zoals planning en facturatie.

## 1.2 Gewenste situatie

De gewenste situatie is als een medewerker via zijn pc of mobiele telefoon in kan loggen op een online applicatie en daar eenvoudig zijn uren zou kunnen registreren.

## 1.3 Projectopdracht

Zoals aangegeven in de huidige situatie, is de urenregistratie nog niet bij alle bedrijven optimaal. Het zou voor zowel bedrijf als medewerker het beste zijn als gewerkte uren altijd geregistreerd kunnen worden. Voor het bedrijf levert dit nauwkeurige data op en de medewerker hoeft niet een hele dag zijn uren te onthouden.

De opdracht is om een applicatie te realiseren welke altijd bereikbaar is, zich onderscheidt ten opzichte van bestaande pakketten en aansluit bij de behoefte van bedrijven. Daarnaast moet de applicatie makkelijker uitbreidbaar en onderhoudbaar zijn en voor zoveel mogelijk mensen toegankelijk.

### 1.3.1 Fasering

Het project is opgedeeld in de volgende fasen:

- Onderzoek;
- Functioneel Ontwerp;
- Technisch Ontwerp;
- Realisatie;
- Testen;
- Oplevering.



### 1.3.2 Planning

Tijdens dit project is aan onderstaande planning gehouden.

Oplevering producten	
Product	Datum
Onderzoeksverslagen	6 november 2009
Functioneel Ontwerp	4 december 2009
Technisch Ontwerp	18 december 2009
HourPower	1 januari 2010
Testrapporten	8 januari 2010

Tabel 1 - Opleverdata

### 1.3.3 Onderzoeksvragen

Om tot een zo goed mogelijk resultaat te komen zijn vooraf de volgende vragen gesteld:

- Bestaan er al online urenregistratiesystemen en zo ja: wat bieden deze pakketten?
- Welke functionaliteiten zijn een vereiste?
- Hoe kan HourPower zich onderscheiden van deze pakketten?
- Welke architectuur gaat worden gebruikt om dit te realiseren?
- Is UiaML als modelleringstaal geschikt voor het Functioneel Ontwerp bij grote applicaties?

De eerste vier vragen zullen in hoofdstuk 4: “Uitvoering en resultaten” worden behandeld. De laatste vraag wordt behandeld in de scriptie van Daniël van Zoelen.

## 1.4 Reikwijdte

Om een systeem aan te bieden wat aansluit bij de behoefte van bedrijven is allereerst onderzoek gedaan naar de bestaande urenregistratiesystemen. De functionaliteit die zij aanbieden zijn in een MoSCoW tabel geplaatst en ingevuld door 10 bedrijven uit verschillende branches en met verschillende groottes.

Uit dit onderzoek zijn de volgende functionaliteiten naar voren gekomen als must have:

- **Basisfunctionaliteit.** Registreren van medewerkers, afdelingen, klanten, projecten, activiteiten en uren.
- **Rapportage.** De volgende rapportages zullen beschikbaar worden gesteld:
  - Geboekte uren op project/activiteit;
  - Geboekte uren op project/activiteit, per medewerker;
  - Geboekte uren vs. geplande uren, per project of medewerker;
  - Week, maand, kwartaal en jaaroverzichten per medewerker;
  - Afdrukken in PDF.
- **Rollen.** Het systeem kent meerdere rollen, welke hun eigen rechten hebben. Deze rollen zijn:

- **Medewerker:** De medewerker kan zijn eigen uren registreren, agenda bijhouden, klanten registreren en bewerken.
- **Manager:** De manager heeft dezelfde rechten als de medewerker maar kan daar bovenop o.a. projecten en activiteiten registreren en bewerken, medewerkers toevoegen en verwijderen uit projecten, uren accorderen en feedback geven op geregistreerde uren.
- **Beheerder:** De beheerder heeft alle rechten binnen het systeem.  
Voor een uitgebreid overzicht met alle rechten verwijst ik u naar het Use-case Diagram uit het Functioneel Ontwerp.
- **Opmerkingen.** Dit maakt het mogelijk voor medewerkers om opmerkingen te plaatsen bij de geregistreerde uren. De manager kan hier op zijn beurt feedback op geven.
- **Onderscheid externe- en interne projecten.** Mogelijkheid om project- en interne uren te scheiden.
- **Uren inleveren.** Per maand, week of zelf in te stellen periode.
- **E-mail notificatie.** Wanneer een medewerker geen uren heeft ingevoerd, of de uren niet heeft ingeleverd dan zal een e-mail verzonden worden ter herinnering.
- **Tarief van medewerkers.** Instellen wat het uurtarief van een medewerker is.
- **Exporteren.** Mogelijkheid om medewerkers, klanten, projecten, activiteiten en uren te exporteren naar XML en CSV.
- **Importeren.** Mogelijkheid om medewerkers, klanten, projecten en activiteiten te importeren vanuit een CSV bestand.
- **Accorderen.** Nadat de uren ingeleverd zijn kunnen deze door de project- of afdelingsmanager worden geaccordeerd.
- **Tarief van activiteiten.** Per activiteit in kunnen stellen wat het kost.
- **Vast bedrag voor projecten.** De mogelijkheid om per project een vast bedrag in te stellen. Door tarieven van medewerkers en activiteiten bij te houden kan vervolgens een overzicht worden getoond hoeveel het project heeft gekost en wat het heeft opgeleverd.
- **Tijd-voor-tijd regeling.** Door overuren bij te houden is het mogelijk voor de medewerker om deze tijd in te ruilen voor een vrije dag wanneer er bijvoorbeeld een keer wat minder werk is.
- **Feedback aan medewerker.** Zoals gezegd kunnen medewerkers opmerkingen toevoegen bij hun geregistreerde uren. De manager kan hierop feedback geven zodra de uren ingeleverd zijn om geaccordeerd te worden.
- **Meertalige interface.** Om ervoor te zorgen dat medewerkers die de Nederlandse taal niet (goed) beheersen ook gebruik kunnen maken van het pakket kan ook de Engelse taal geselecteerd worden.
- **Agenda bijhouden .** Binnen het systeem is het mogelijk om een agenda bij te houden. Het is mogelijk om een agenda met meerdere mensen te delen.
- **Taken in agenda direct als uren verwerken.** Om het geheel makkelijk te houden kunnen uren in de agenda direct als gewerkte uren worden verwerkt. Een gebruiker hoeft dan niet 2 keer dezelfde uren in te voeren.
- **Koppeling tussen agenda en Outlook.** Veel mensen maken gebruik van een agenda binnen Outlook en het is prettiger om 1 agenda bij te houden i.p.v. twee. Om deze is onderzoek

gedaan naar een koppeling tussen de Outlook agenda en de agenda van HourPower. Meer informatie over dit onderwerp is te vinden in hoofdstuk 4.1.4.

## **2   Uitgangssituatie**

Om het project, zoals deze in hoofdstuk 1 is gespecificeerd, tot een goed einde te brengen waren er een aantal voorwaarden waaraan moest worden voldaan. Deze voorwaarden worden in dit hoofdstuk besproken.

### **2.1   Overleg projectbegeleider**

Om ervoor te zorgen dat het project zonder te veel problemen verliep was het belangrijk om de projectbegeleider op de hoogte te houden van de voortgang. Dit gebeurde door wekelijks een highlight report op te stellen en op te sturen.

Op het moment dat er echt grote problemen of vragen de kop op staken dan waren de projectleden zelf verantwoordelijk voor een afspraak met de projectbegeleider. Tijdens zo'n afspraak werd het probleem besproken en (voor zover mogelijk) opgelost worden.

### **2.2   Goede werkomgeving**

Om goed aan het project te kunnen werken was een werkplek noodzakelijk. Tijdens het project werd gezocht naar een kantoor en tot die tijd is gebruik gemaakt van de mogelijkheden die geboden werden op Oudenoord 700 en Nijenoord 1.

### **2.3   Kennis van de opdracht**

Het was belangrijk dat beide projectleden goed op de hoogte waren van de opdracht. Om deze reden moesten alle documenten grondig doorgelezen en besproken worden om verwarring te voorkomen.

### **2.4   Kennis van de ontwikkeltools**

Tijdens de ontwikkeling is gebruik gemaakt van PHP, MySQL, AJAX (jQuery) en UiaML. Het was belangrijk dat er voldoende kennis was van deze zaken om onnodige vertraging te voorkomen.

## 3 Methoden en technieken

### 3.1 Projectmethode

Tijdens dit project is gebruik gemaakt van de projectmethode Prince2. Hiervoor is tijdens de minor E-Business het certificaat behaald en besloten om deze methode in de praktijk uitvoeren. De belangrijkste redenen waarom gebruik is gemaakt van Prince2 zijn:

- Gecontroleerd en georganiseerd start, midden en einde;
- Regelmatige evaluaties van vooruitgang ten opzichte van het plan en de business case;
- Flexibele besluitpunten;
- Betrokkenheid van management op passende punten;
- Scheiding van managementactiviteiten van technische activiteiten.

### 3.2 Ontwikkelmethode

Tijdens de studie Informatica hebben is gewerkt met de watervalmethode en DSDM om projecten te realiseren. Voor dit project is besloten om met DSDM aan de slag te gaan. In dit hoofdstuk worden beide methodes behandeld en uitgelegd waarom de keuze op DSDM is gevallen.

#### 3.2.1 Watervalmethode

De watervalmethode bestaat uit een aantal fasen waarin de ontwikkeling zich vloeiend naar beneden beweegt. De fasen van deze methode zijn:

- Analyse;
- Functioneel Ontwerp;
- Technisch Ontwerp;
- Bouw;
- Testen;
- Integratie;
- Beheer en onderhoud.

Deze methode is zo opgebouwd dat een fase afgerond moet zijn voordat een nieuwe fase gestart kan worden. Het nadeel hiervan is dat mensen binnen een project afhankelijk zijn van elkaar voordat gestart kan worden met de werkzaamheden. Bouwers zullen moeten wachten tot de ontwerpers klaar zijn. Dit kost een hoop tijd en geld.

Het is bijna een feit dat een opdrachtgever gedurende het project met wijzigingen komt. De watervalmethode is hier niet op berekend. Hierdoor moeten het complete Functionele- en Technische Ontwerp aangepast worden wanneer tijdens de bouwfase wijzigingen gemaakt moeten worden. Dit is wellicht het grootste nadeel aan de watervalmethode.

Een voordeel van de watervalmethode is dat het altijd duidelijk is in welke fase het project zich bevindt.

#### 3.2.2 DSDM

DSDM is een ontwikkelmethode die, in tegenstelling tot de watervalmethode, iteratief te werk gaat.

Tijdens elke iteratie worden kleine onderdelen toegevoegd aan het Functioneel- en Technisch Ontwerp waarna deze gerealiseerd en getest kunnen worden. Dit gaat net zo lang door tot alles naar wens is. Het voordeel hiervan is dat wijzigingen snel doorgevoerd kunnen worden en projectleden niet op elkaar zitten te wachten. Wanneer de bouwers bezig zijn met het realiseren van iteratie 1 zijn de ontwerpers al bezig met iteratie 2.

Een ander voordeel ten opzichte van de watervalmethode is het principe genaamd 'timeboxing'. Waar bij projecten met de watervalmethode de functionaliteit vast staat, daar staan bij DSDM de tijd en het geld vast. Functionaliteit is variabel. Wanneer het project boven de beschikbare hoeveelheid tijd en geld uit dreigt te komen wordt bekeken welke functionaliteit geschrapt moet worden. Het bepalen hiervan wordt met behulp van een MoSCoW tabel gedaan. Deze afkorting staat voor:

- Must have – moet worden gerealiseerd;
- Should have – zou gerealiseerd moeten worden;
- Could have – zou gerealiseerd kunnen worden;
- Would like to have – zou leuk zijn om te hebben maar hoeft nu niet gerealiseerd te worden.

Deze MoSCoW tabel is ook gebruikt om uit te vinden welke functionaliteiten perse gerealiseerd moesten worden tijdens dit project.

Het feit dat bij dit project de deadline vast stond is de aanleiding geweest om gebruik te maken van DSDM. Tijd en geld (hoewel het laatste nu niet van toepassing is) staan vast en functionaliteit is variabel. Om deze laatste reden hebben is gebruik gemaakt van een MoSCoW tabel om erachter te komen wat de must-haves voor het systeem waren. Meer informatie hierover is te vinden in hoofdstuk 4.1.3.

### **3.3 Ontwikkeltechniek**

Tijdens het project zijn een aantal verschillende ontwikkeltechnieken gebruikt. Er is gebruik gemaakt van een server, client- en databasetechniek. Ook is er een specifieke techniek voor versiebeheer gebruikt.

#### **3.3.1 Server**

Omdat de applicatie via internet bereikbaar is wordt gebruik gemaakt van een techniek die hier ondersteuning voor biedt. Om deze reden is besloten om gebruik te maken van PHP5 in combinatie met AJAX.

PHP5 maakt het mogelijk om server-side en op een objectgeoriënteerde manier te programmeren.

#### **3.3.2 Client**

Op de machine van de eindgebruiker wordt gebruik gemaakt van AJAX. Doordat AJAX gebruik maakt van JavaScript is het een vereiste dat de gebruiker deze optie aan heeft staan in de webbrowser. Dit moet echter geen probleem zijn gezien het feit dat ook websites als Hyves en Youtube gebruik maken van Javascript. Hierdoor is de aanname gedaan dat er weinig gebruikers op de website zullen komen die Javascript niet ingeschakeld hebben.

Een aantal voordelen die AJAX biedt zijn onder andere:

- Verzenden van data zonder refresh;
- Directe controle na het invoeren van data, zonder het formulier te hoeven verzenden.

Ondersteuning van Javascript op mobiele telefoons verschilt per merk en type, dus hiervoor zal een lichtgewicht variant komen die dit niet bevat.

### 3.3.3 Database

Voor de database is gebruik gemaakt van een MySQL database.

### 3.3.4 Versiebeheer

Tijdens de ontwikkeling van de applicatie is gebruik gemaakt van versiebeheer. Hiervoor wordt gebruik gemaakt van TortoiseSVN.

### 3.3.5 Applicatieserver

De applicatie draait op een server van Hosting2Go. Zij bieden ondersteuning voor alle eerder genoemde technieken, met uitzondering van het versiebeheer. Voor zowel de live versie als de ontwikkelomgeving is een apart domein beschikbaar.

Momenteel draait het systeem op een shared server. Zolang de grootte van het systeem binnen de limieten van deze server valt zal dit voldoende zijn. Mocht het systeem in de toekomst deze limieten overschrijden dan kan op ieder gewenst moment over worden gestapt naar een dedicated server.

### 3.3.6 Conventies

Tijdens het project wordt uitgegaan van een aantal conventies waarmee tijdens de lessen Java kennis is gemaakt, te weten:

- Naamgeving van bestanden;
- Naamgeving van variabelen;
- Naamgeving van methoden;
- Gebruik van hoofd- en kleine letters;
- Gebruik van haakjes en accolades;
- Gebruik van witregels, spaties en tabs.

Tijdens het project is zoveel mogelijk rekening gehouden met deze conventies.

Deze code conventies kunnen gevonden worden op

<http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html>.

## 4 Uitvoering en resultaten

### 4.1 Onderzoeksfase

In de onderzoeksfase zijn een viertal onderzoeken uitgevoerd. Er is onderzoek gedaan naar reeds bestaande online urenregistratiesystemen, de beste software architectuur voor de applicatie en de mogelijkheid voor een Microsoft Outlook add-on.

Op basis van de resultaten van de bestaande pakketten is een onderzoek gestart naar de wensen van de klant. Hierdoor werd duidelijk welke functionaliteit noodzakelijk is binnen de applicatie.

Nadat alle onderzoeksresultaten bekend waren is het functioneel ontwerp gerealiseerd.

#### 4.1.1 Bestaande pakketten

In dit hoofdstuk wordt de onderzoeksvraag: “Bestaan er al online urenregistratiesystemen, en zo ja: wat bieden deze pakketten?” beantwoord.

Eén van de doelen is om onderscheidend te zijn van de pakketten die reeds bestaan. Om deze pakketten te onderzoeken is een zoekopdracht naar online urenregistratie gedaan in meerdere zoekmachines.

Uit de lijst die hieruit naar voren is gekomen zijn een zestal resultaten gekozen om te onderzoeken wat zij aanbieden.

Deze pakketten zijn:

- Mijnnuren Online;
- 4ever Uren;
- Clockwise;
- NuTijd;
- Comsult Urenregistratie;
- Sumeris Urenregistratie.

Deze zes pakketten zijn gekozen omdat deze destijds hoog in de resultatenlijst eindigden bij de verschillende zoekmachines.

Bij sommige pakketten verliep dit onderzoek zeer goed, aangezien zij een demo beschikbaar hadden op hun website. Andere bedrijven hadden vrijwel geen informatie beschikbaar en hiervoor moest een demo aangevraagd worden of konden video's en screenshots bekeken worden.

Wat als eerste opviel bij een aantal pakketten is dat deze meer bieden als alleen urenregistratie. Sommigen bevatten ook onderdelen als planning en facturatie. Dit is wellicht iets om in de toekomst mee aan de slag te gaan. Eerst moet HourPower een goed werkend urenregistratiesysteem worden.

Op basis van de aangeboden functionaliteit is een lijst opgesteld met functionaliteiten die veel terugkwamen, nodig werden geacht en onderscheidend zijn.

Naast alleen functionaliteit is ook gekeken naar de gebruiksvriendelijkheid van de applicaties (voor zover beschikbaar). Dit is gedaan door de 5 basisregels van Krug te vergelijken met de applicaties.



De vijf basisregels van Krug zijn:

1. **Create a clear visual hierarchy.** Hier wordt gekeken of er een duidelijke hiërarchische structuur te zien is op de website. Grote kopteksten, logo's en afbeeldingen worden als zeer belangrijk gezien omdat deze als eerst de aandacht trekken. Hoe kleiner de tekst, hoe onbelangrijker.
2. **Take advantages of conventions.** Haal voordeel uit conventies. Door bijvoorbeeld een vergrootglas te tonen bij een tekstveld snapt vrijwel iedereen dat daarmee gezocht kan worden. Zo hoeft een bezoeker van de pagina niet perse de taal te spreken, maar hij begrijpt wel wat ermee bedoelt wordt.
3. **Break up pages into clearly defined areas.** Door ervoor te zorgen dat pagina's zijn opgedeeld in verschillende gebieden blijft het overzicht behouden.
4. **Make it obvious what's clickable.** Het is belangrijk om duidelijk te maken wat klikbaar is op een website. Dikgedrukte tekst kan ook zijn om ergens de nadruk op te leggen. Het is het beste om ervoor te zorgen dat klikbare tekst onderstreept is en een andere kleur heeft dan de rest van de tekst.
5. **Minimize noise.** Ruis is iets wat als zeer vervelend wordt beschouwd en jaagt bezoekers zelfs weg. Het gebruik van veel verschillende (knal)kleuren, bewegende plaatjes etc. wordt daarom sterk afgeraden.

Hieruit kon worden geconcludeerd dat er niet 1 pakket was wat aan alle regels voldeed. Iets waar HourPower wel aan moet voldoen.

Het volledige onderzoek is te vinden in bijlage B1.

#### 4.1.2 Onderscheiden

In dit hoofdstuk wordt de onderzoeksvraag: "Hoe kan HourPower zich onderscheiden van deze pakketten?" beantwoord.

Op basis van het resultaat van het onderzoek naar bestaande pakketten is onderzocht op welk gebied HourPower zich kan onderscheiden.

We zijn uiteindelijk op de volgende onderdelen uitgekomen:

- Importeerfunctie voor medewerkers, klanten, projecten en activiteiten;
- Toegang vanaf de mobiele telefoon;
- Koppeling tussen agenda en de agenda van Outlook.

Deze onderdelen worden door geen van de bekeken pakketten aangeboden en verwacht wordt dat dit waardevolle toevoegingen zijn in het systeem.

Door het mogelijk te maken reeds bestaande data te importeren in het systeem wordt het makkelijker voor bedrijven om over te stappen wanneer zij al gebruik maken van een bepaald pakket. Zonder deze functie verplicht je eigenlijk dat bestaande data met de hand ingevoerd wordt of dat een bedrijf weer vanaf 0 begint.

Ook toegang vanaf de mobiele telefoon vinden wij belangrijk voor een systeem als deze. Niet iedereen heeft altijd de beschikking over een computer om de uren te registreren. Voor deze mensen is het ideaal om even snel met de mobiele telefoon verbinding te maken en alsnog de uren in te kunnen voeren. Uit onderzoek van Forrester gebleken dat een vijfde van de Nederlanders gebruik maakt van internet op de mobiele telefoon. Dit komt neer op een aantal van ongeveer 2,25 miljoen mensen. Een ander onderzoek van Forrester geeft aan dat verwacht wordt dat het gebruik van internet op de mobiele telefoon in West-Europa de komende jaren zal stijgen. Van 13% in 2008 naar 39% in 2014.

De koppeling tussen de agenda van ons systeem en die van Outlook is ook een onderdeel waarvan verwacht wordt een meerwaarde te zijn. In het bedrijfsleven wordt veelal gebruik gemaakt van Microsoft Outlook als mailprogramma en agenda. Door een koppeling te realiseren blijven beide agenda's altijd gelijk aan elkaar en kunnen afspraken via Outlook in het systeem worden opgeslagen en vice versa.

Er is een apart onderzoek uitgevoerd naar de haalbaarheid van deze Outlook add-on. Dit onderzoek wordt besproken in hoofdstuk 4.1.5.

Deze functionaliteiten zijn opgenomen in het onderzoek naar de wensen van de klant. Dit onderzoek wordt in het volgende hoofdstuk besproken.

#### **4.1.3 Wensen van de klant**

In dit hoofdstuk wordt de onderzoeksvraag: "Welke functionaliteiten zijn een vereiste?" beantwoord.

Zoals gezegd is op basis van het vorige onderzoek een lijst met functionaliteiten opgesteld. Deze lijst is te vinden in bijlage B2.

Door gebrek aan ervaring was het in eerste instantie de vraag waar begonnen moest worden met een dergelijk onderzoek. Op advies van dhr. R. Raatgever van Utrecht Inc. is in de directe omgeving geïnformeerd.

Deze aanpak werkte en dankzij vrienden en familie zijn eenvoudig 10 bedrijven gevonden die de opgestelde lijst met functionaliteiten hebben beoordeeld middels een MoSCoW tabel. Ten tijde van het onderzoek was nog niet bekend hoeveel resultaten genoeg zouden zijn voor goed onderzoek. Als gevolg hiervan is een afspraak gemaakt met de projectbegeleider, meneer van Elsen. In dit gesprek is gezegd dat 10 bedrijven zouden volstaan.

Uiteindelijk is er contact geweest met 13 bedrijven waarvan de volgende hebben meegewerkt aan ons onderzoek:

- Feka ICT & Office;
- Admium;
- De Mandemakers Groep;
- Atos Origin;
- Het Spiegelend Koetje;
- Accounting Plaza;
- Peak IT;

- Sansa;
- The Human Effect;
- Zin Kraamzorg.

Met onderstaande bedrijven is contact geweest maar zij hebben aangegeven niet mee te willen doen, of hebben nooit een reactie gegeven:

- New Tell Essence;
- EDS;
- Emolife.

Het moge duidelijk zijn dat het zeer uiteenlopende bedrijven zijn waar contact mee is geweest. Dit is echter een bewust keuze geweest. HourPower moet zich niet richten op één soort bedrijf maar een systeem zijn wat voor elk bedrijf, van elke grootte van toepassing kan zijn.

Het resultaat wat hieruit is voortgekomen is te vinden in hoofdstuk 1.3 Reikwijdte.

#### **4.1.4 Software architectuur**

In dit hoofdstuk wordt de onderzoeksvraag: “Welke architectuur gaat worden gebruikt om dit te realiseren?” beantwoord.

Uit het onderzoek naar bestaande pakketten is gebleken dat het geregeld voorkomt dat er meer aangeboden wordt dan alleen urenregistratie. Mede hierdoor is het een vereiste dat het systeem makkelijk uitbreidbaar en onderhoudbaar is. Dit zijn dus niet-functionele eisen die van belang zijn voor het systeem. Er is gezocht naar architectuur- patronen en stijlen die hierbij ondersteuning konden bieden.

Na wat speurwerk naar verschillende software architecturen zijn er een aantal boven water gekomen welke zijn onderzocht. Te weten:

- Model-View-Controller;
- Presentation-Abstraction-Control;
- Document-View;
- Model-View-Presenter.

Na al deze architecturen bekeken te hebben is de conclusie getrokken dat het MVC model in combinatie met een Front Controller voor dit systeem de beste mogelijkheden biedt. Niet alleen is er reeds voldoende ervaring met het MVC model, vrijwel elk bekend PHP framework werkt op basis van het MVC model. Ten tijde van dit onderzoek was nog niet bekend of een framework gebruikt zou worden dus deze optie is bewust open gehouden.

De reden dat naast het MVC model ook gebruik gemaakt wordt van een Front Controller is omdat deze het makkelijker maakt om operaties uit te voeren die overal gelden. Doordat de Front Controller altijd als eerste aangeroepen wordt kan deze voor o.a. controle van sessies worden gebruikt.

Het volledige onderzoek is te vinden in bijlage B3.

#### 4.1.5 Microsoft Outlook add-on

Eén van de onderdelen waarin HourPower zich kan onderscheiden is een plug-in voor Microsoft Outlook. Het was echter de vraag of dit haalbaar zou zijn binnen de gestelde tijd. Om deze reden is een onderzoek gestart om te kijken wat er benodigd is om een dergelijke add-on te realiseren.

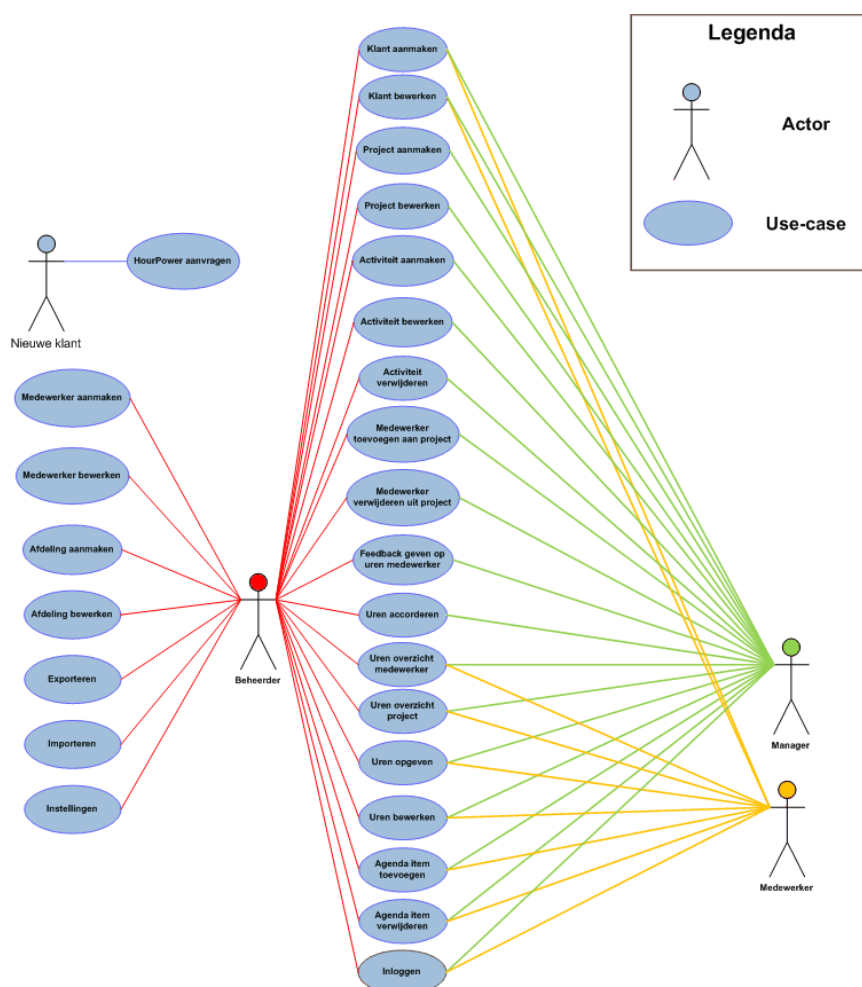
Middels dit onderzoek is de conclusie getrokken dat er een hoop verschillende manieren zijn om een dergelijke add-on te realiseren. Daarnaast is geconcludeerd dat er momenteel niet voldoende tijd en kennis beschikbaar is om dit onderdeel tot een goed resultaat te kunnen brengen gedurende dit project.

Het is echter een zeer waardevolle (en onderscheidende) uitbreiding op het systeem.

Het volledige onderzoek is te vinden in bijlage B4.

## 4.2 Functioneel Ontwerp

In het Functioneel Ontwerp wordt de functionaliteit van het urenregistratiesysteem beschreven. Dit is gedaan met behulp van een use-case diagram (zie Figuur 1). Deze is opgesteld uit het resultaat van het onderzoek naar de gewenste functionaliteit.



Figuur 1 - Use case diagram

Deze use-cases zijn vervolgens verder uitgewerkt in UiaML. Dit is een standaard welke bedacht is door Alex Jongman. Hier is kennis mee gemaakt tijdens de specialisatie Web-Essentials.

De heer Jongman heeft deze standaard bedacht om de functionele werking van de applicatie in beeld te brengen. Om dit beeld nog duidelijker te krijgen zijn er meerdere weergaves binnen UiaML, de Sitemap View (SMV) en de Content Area View (CAV).

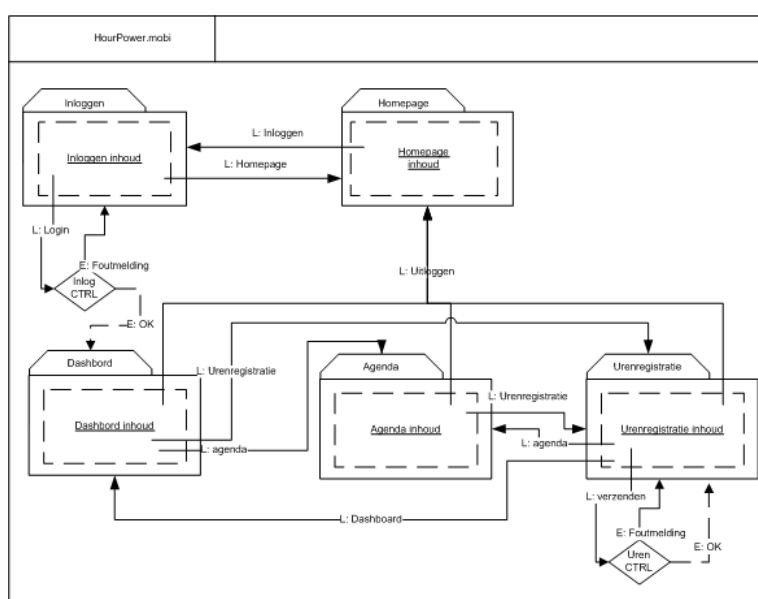
De SMV is een weergave van het totaalplaatje. Dit is een overzicht van het complete systeem met alle onderlinge relaties tussen pagina's en een overzicht van eventuele events van een pagina.

Per use-case wordt een use-case specificatie geschreven met daarbij een CAV en een preview. In een dergelijke specificatie staat beschreven welke situaties voor kunnen komen na bijvoorbeeld het versturen van een formulier.

Het complete Functioneel Ontwerp was te groot om bij te voegen als bijlage. Op verzoek kan deze worden verkregen.

#### 4.2.1 Sitemap View

Op Figuur 2 is de SMV van de mobiele website te zien. Ondanks dat de mobiele website slechts 5 pagina's telt waarvan 2 met events begint het overzicht al vrij druk te worden. Om deze reden heb ik ook besloten om deze afbeelding te tonen in plaats van de SMV van het gehele systeem. Deze is inmiddels zo groot dat hij niet meer op 1 pagina past en de overzichtelijkheid is vrijwel volledig verdwenen.

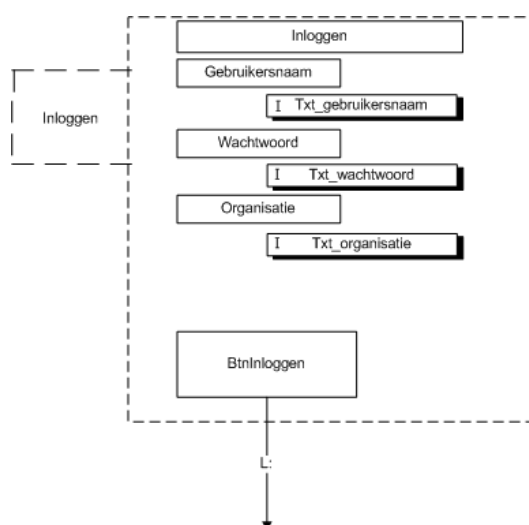


Figuur 2 - Sitemap View

Daniël van Zoelen heeft dan ook besloten om UiaML in zijn scriptie onder de loep te nemen en te kijken waar de sterke en zwakke punten liggen.

### 4.2.2 Content Area View

De CAV is een weergave van een enkele pagina. Het laat zien welke onderdelen zich op de betreffende pagina bevinden, met bijbehorende naam en type. Tevens wordt aangegeven welke velden verplicht zijn om in te vullen. Zoals te zien is op Figuur 3



Figuur 3 - Content Area View

Deze gegevens kunnen door de programmeur worden overgenomen wanneer hij bezig gaat met het realiseren van de betreffende pagina.

### 4.2.3 Previews

Zoals te zien is op Figuur 3 laat de Content Area View alleen zien wat er op de pagina moet komen, maar niet hoe het eruit moet komen te zien.

Om die reden zijn er ook previews toegevoegd aan het Functioneel Ontwerp. De preview van deze Content Area View is te zien op Figuur 4.

## 4.3 Technisch Ontwerp

Uit het Functioneel Ontwerp valt op te maken hoe de applicatie eruit komt te zien. Om de werking van de applicatie duidelijk te maken is een Technisch Ontwerp opgesteld. Hierin staan het database ontwerp, de sequence diagrammen van de use cases, de structuur en de gebruikte architectuur.



Figuur 4 - Inloggen preview

### 4.3.1 Database ontwerp

Op basis van het onderzoek naar de gewenste functionaliteit is het mogelijk geweest een database ontwerp kunnen maken. Hierover hebben is nog een discussie gevoerd op welke manier we dit aangepakt zou worden.

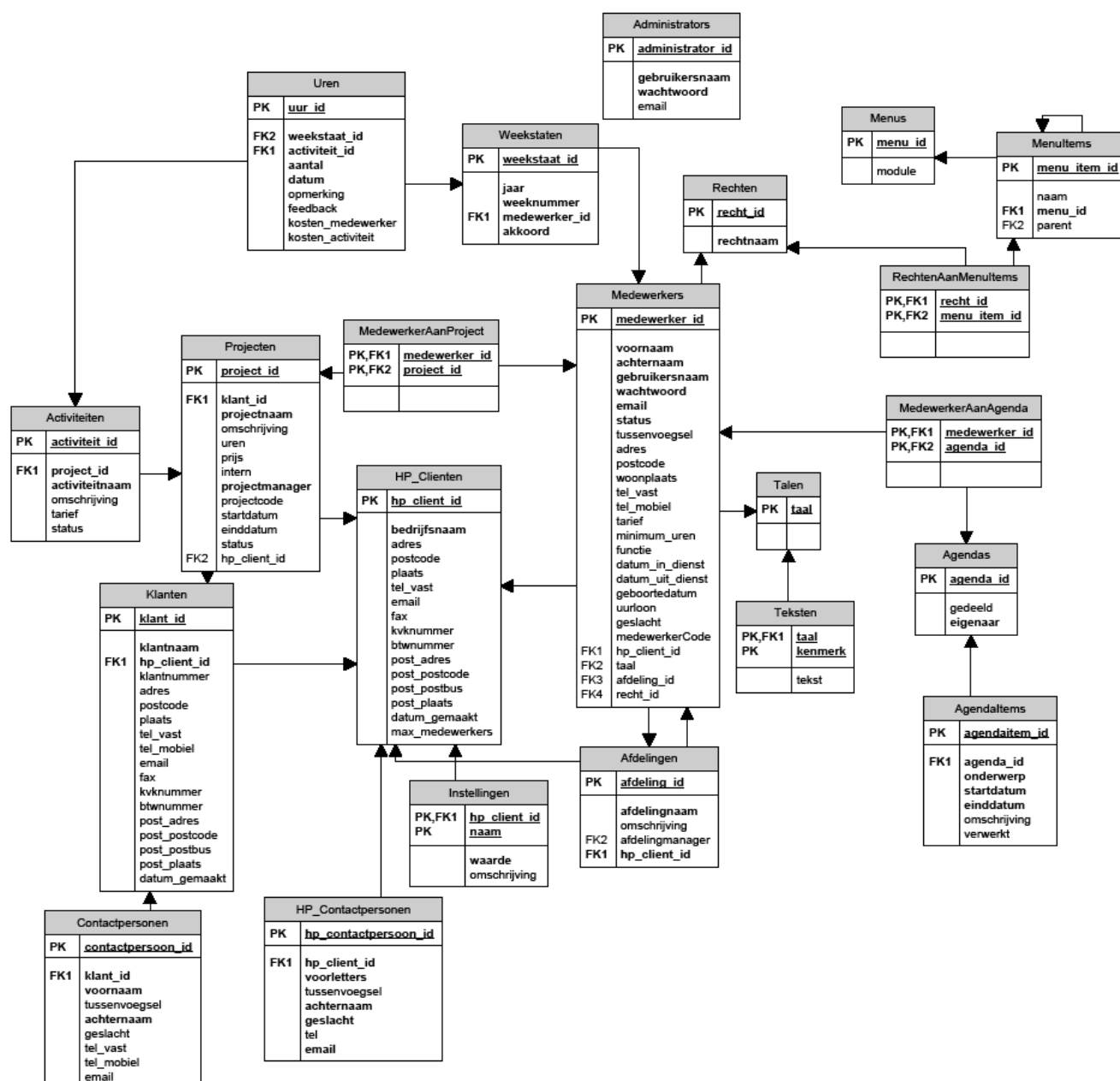
Optie 1 was om elke klant een eigen database te geven waarin alle data geregistreerd zou worden. Optie 2 was om 1 database te maken waarin alle klanten hun data zouden registreren.

De voor- en nadelen van beide opties zijn met elkaar vergeleken en uiteindelijk is besloten voor optie 2 te gaan.

Het belangrijkste punt is dat het systeem op de manier van optie 2 veel beter te onderhouden en uit te breiden is. Er hoeft slechts 1 database aangepast te worden wanneer dit nodig is. Bij optie 1 zouden dit x databases zijn op basis van x klanten.

Het feit dat de database hierdoor wat groter en complexer is maakt in deze weinig uit.

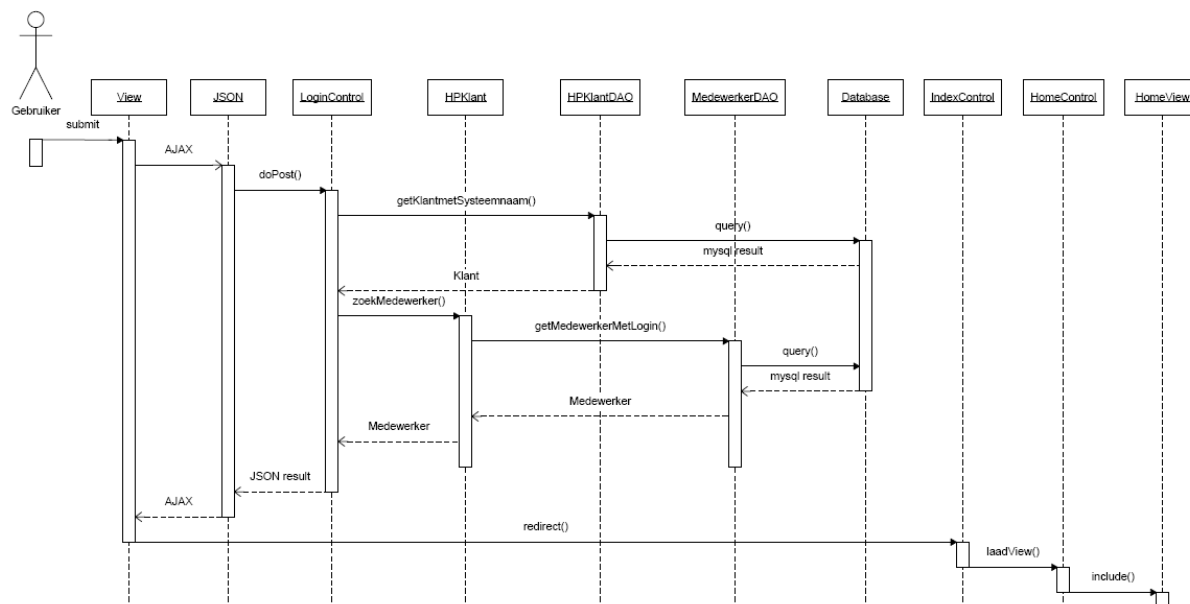
Het databasemodel is te zien op Figuur 5.



Figuur 5 - Database ontwerp

### 4.3.2 Sequence diagrammen

Om duidelijk te maken hoe de applicatie werkt is gebruik gemaakt van sequence diagrammen. Deze geven per use case aan welke functies aangeroepen worden in welke bestanden. Hierdoor wordt een duidelijk beeld van de werking van een use case geschetst en kan bekeken worden hoe deze verbeterd of uitgebreid kan worden.



Figuur 6 - Sequence diagram inloggen

Op Figuur 6 is het sequence diagram van het inloggen te zien.



### 4.3.3 Structuur

In de structuur van het systeem is heel bewust rekening gehouden met het feit dat er in de toekomst mogelijk meerdere modules toegevoegd moeten worden. Ook is rekening gehouden met eventuele wensen voor maatwerk.

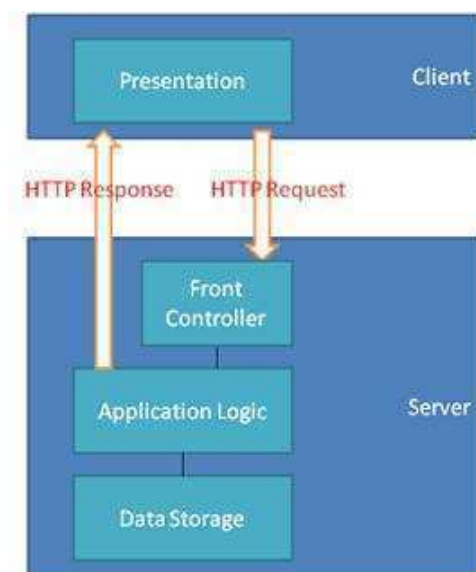
In Tabel 2 is de structuur van het systeem te zien.

Map	Gebruik
<b>/htdocs</b>	Dit is de standaard map waarin gezocht wordt naar de startpagina (index.php)
<b>/modules</b>	Dit is de map die de verschillende modules bevat, zoals urenregistratie
<b>/urenregistratie</b>	Bevat alle code van de module urenregistratie
<b>/classes</b>	Hierin staan alle model classes
<b>/controllers</b>	Hierin staan de controllers. Deze verwerken data en starten de views
<b>/views</b>	Hierin staan de views. Deze tonen de data aan de gebruiker
<b>/images</b>	Bevat alle afbeeldingen van het systeem
<b>/klantnaam</b>	Dit wordt een map met de naam van de klant. In deze map worden o.a. logo's van de klant bewaard
<b>/css</b>	Bevat alle bestanden die de opmaak van de views verzorgen
<b>/js</b>	Bevat alle JavaScript bestanden
<b>/maatwerk</b>	Deze map wordt gebruikt voor maatwerk oplossingen die door de klant aangevraagd kunnen worden
<b>/klantnaam</b>	Wederom een map met de naam van de klant. Op deze manier wordt maatwerk van verschillende klanten gescheiden
<b>/classes</b>	Bevat alle model classes van het maatwerk
<b>/controllers</b>	Bevat de controllers van het maatwerk
<b>/views</b>	Bevat de views van het maatwerk

Tabel 2 - Mappenstructuur

#### 4.3.4 Architectuur

Zoals eerder vermeld (zie hoofdstuk 4.1.4) is onderzoek gedaan naar de architectuur voor het systeem. Op basis van dit onderzoek is gekozen voor het MVC model met gebruik van een Front Controller.



Figuur 7 - Architectuur

De Front Controller vangt als eerste de request van de client op. Vervolgens wordt de ingevoerde URL bekeken en de bijbehorende controller geladen. De controller voert op zijn beurt de benodigde bewerkingen uit en stuurt de response naar de client (zie Figuur 7).

Wanneer de ingevoerde URL niet bekend is wordt een error pagina getoond.

#### Verzenden van een formulier

Het verzenden van een formulier gaat iets anders dan alleen het bezoeken van een pagina.

Waar een normaal formulier de data verzendt en de pagina ververs, daar maakt HourPower gebruik van een AJAX form. Het AJAX form wordt op dezelfde manier opgebouwd als ieder ander formulier op een website. Er wordt echter een klein stukje Javascript (jQuery) aan toegevoegd:

```

$(document).ready(function() {
    $('#login').ajaxForm({
        dataType: 'json',
        success: validateLogin
    });
});
  
```

`$('#login').ajaxForm` zorgt ervoor dat het formulier behandeld wordt als een AJAX form.

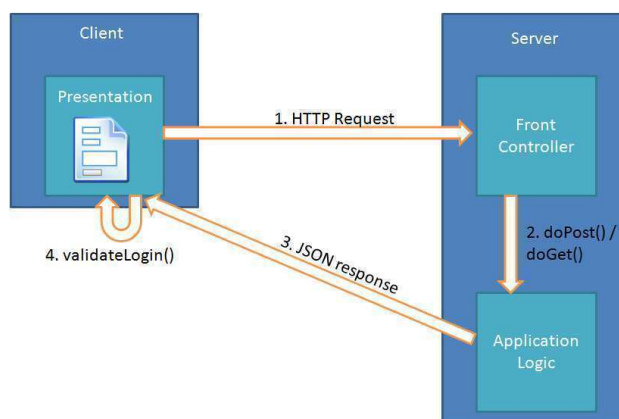
Hierdoor wordt de data verzonden en ontvangen zonder dat een refresh van de pagina noodzakelijk is. `dataType: 'json'` geeft aan welk content type verwacht wordt als antwoord van de server. In dit geval is dat dus een JSON response. Een JSON response kan er als volgt uitzien:

```
{
  "firstName": "John",
  "lastName": "Smith",
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021"
  },
  "phoneNumbers": [
    { "type": "home", "number": "212 555-1234" },
    { "type": "fax", "number": "646 555-4567" }
  ]
}
```

`success: validateLogin` geeft aan welke functie aangeroepen moet worden nadat de response van de server is ontvangen.

Na het verzenden van het formulier wordt allereerst de Front Controller aan het werk gezet. Wederom wordt bekeken wat de URL van de request is, om vervolgens de juiste controller aan te roepen. Vervolgens wordt gekeken of het om een POST of een GET request gaat (afhankelijk van de instellingen van het formulier) en op basis daarvan wordt een `doPost` of `doGet` aangeroepen bij de betreffende controller.

De controller gaat vervolgens aan de slag met de validatie en verwerking van de data en stuurt een JSON reponse terug. Op basis van deze response wordt de functie `validateLogin` aangeroepen, welke op zijn beurt fout- of succesmeldingen laat zien (zie Figuur 8).



**Figuur 8 - Verzenden van het inlog formulier**

## 4.4 Realisatie

Om het systeem te realiseren is iteratief gewerkt. Elke use case is als aparte iteratie genomen om zo het systeem stap voor stap op te bouwen.

Tijdens het realiseren is gebruik gemaakt van een ontwikkelomgeving welke draait op een PHP 5 server en een MySQL database server. De ontwikkelomgeving wordt gebruikt om nieuwe onderdelen

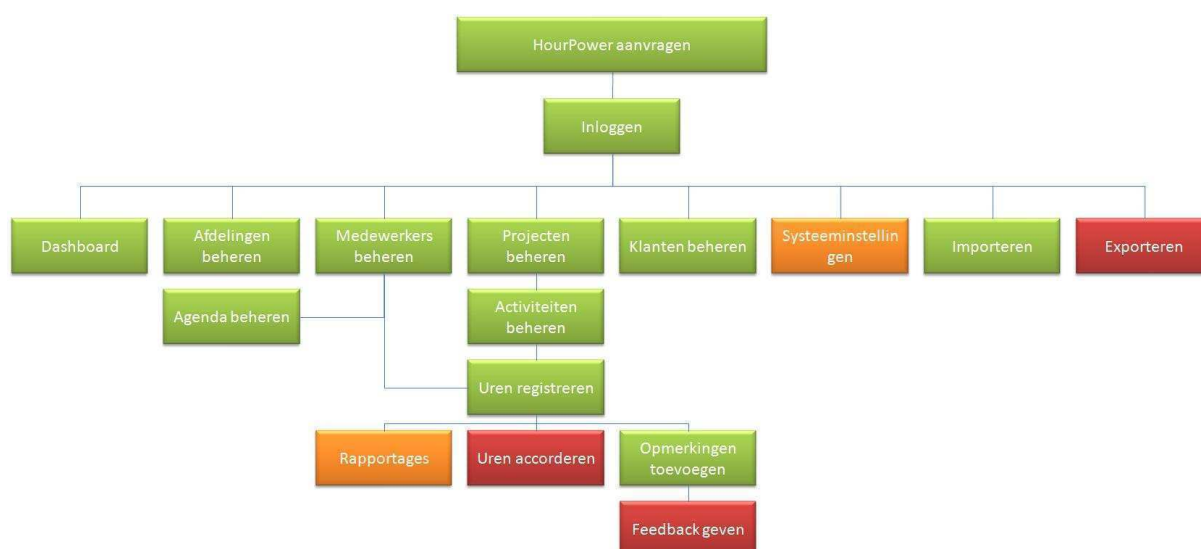
aan het systeem toe te voegen en bestaande onderdelen aan te passen. Ook het testen van het systeem vindt plaats in deze omgeving.

Voordat werd begonnen met programmeren is overwogen of het gebruik van een framework een serieuze optie zou zijn voor het systeem. Uiteindelijk is besloten dat dit niet het geval is. Dit omdat een framework een eigen structuur heeft waar vaak niet veel aan gewijzigd kan worden. Doordat HourPower gebruik maakt van een eigen structuur (zie hoofdstuk 4.3.3) om het programma makkelijk uitbreidbaar te maken en code van verschillende modules te scheiden bood een framework teveel restricties.

#### 4.4.1 Afhankelijkheden

In het systeem zitten functionaliteiten die van elkaar afhankelijk zijn om goed te kunnen werken. Zo is het onmogelijk om in te loggen zonder dat eerst een pakket is aangevraagd, dient er eerst ingelogd te worden voordat medewerkers toegevoegd kunnen worden, moet een medewerker bestaan voordat deze uren kan registreren etc.

Hieronder is een overzicht van deze afhankelijkheden opgenomen:



**Figuur 9 - Afhankelijkheidsdiagram**

Aan de hand van deze afhankelijkheden is het systeem opgebouwd. Allereerst is het aanvragen van het pakket gerealiseerd, gevolgd door het inloggen en zo stapsgewijs een niveau naar beneden.

Door dit overzicht wordt ook snel duidelijk welke functionaliteit van het systeem veel afhankelijkheden kent en dus een belangrijke stap in het realisatietraject is.

Omdat nog niet alle functionaliteit gerealiseerd was tijdens het schrijven van deze scriptie is met behulp van verschillende kleuren onderscheid aangebracht in de status. De groene onderdelen zijn afgerond, de oranje worden aan gewerkt en de rode moeten nog gerealiseerd worden.

#### 4.4.2 Resultaten

In dit hoofdstuk zullen alle gerealiseerde onderdelen van Figuur 9 worden besproken.

##### Hourpower aanvragen

Voordat het mogelijk is om uren te registreren moet er eerst een aanvraag worden gedaan om het systeem te kunnen gebruiken.

Hiervoor zijn een aantal velden verplicht om in te vullen zodat het systeem de aanvraag op de juiste manier kan verwerken. Zo is het bijvoorbeeld verplicht om de voornaam, achternaam en e-mail van de contactpersoon op te geven. Dit omdat het systeem de contactpersoon zal gebruiken om een beheerder account aan te maken.

Het e-mailadres van de contactpersoon wordt gecontroleerd op basis van het domein wat ingevoerd wordt. Veel scripts gebruiken reguliere expressies om te kijken of er een @ in staat en of het e-mailadres geen vreemde tekens bevat. Wij kijken echter naar nog iets extra's, het domein. Met reguliere expressies zou het adres example@example.com toegestaan zijn. Bij ons is dit niet het geval omdat van example.com geen MX records opgevraagd kunnen worden. Dit houdt in dat het domein niet bestaat of geen mailserver bevat.

Vervolgens gaat het systeem automatisch een gebruikersaccount aanmaken voor de contactpersoon welke zal bestaan uit voornaam.achternaam. Jeffrey Veldkamp zal dan jeffrey.veldkamp worden en Daniël van Zoelen wordt daniel.vanzoelen.

Hetzelfde geldt voor de bedrijfsnaam. Wanneer hier Jeffrey B.V. wordt ingevuld, dan wordt er een systeemnaam aangemaakt met jeffreybv. Dit is weer van belang op het moment dat er wordt ingelogd.

Wanneer alle velden correct zijn ingevuld wordt er een e-mail verzonden naar de contactpersoon met de bijbehorende inloggegevens. Ook zal er een CC worden verzonden naar [info@hourpower.eu](mailto:info@hourpower.eu) dat er een nieuwe klant geregistreerd is.

Een klein detail wat aan het systeem is toegevoegd is dat het KvK nummer na het invoeren direct gecontroleerd wordt. Wanneer het geen juist formaat is of reeds bestaat dan zal direct een melding verschijnen. In de toekomst kan dit wellicht worden gebruikt om direct de gegevens van het bedrijf op te halen.

##### Inloggen

Alleen het aanvragen van HourPower is natuurlijk niet voldoende. De volgende stap is om in te loggen op het zojuist aangevraagde systeem. Om in te loggen zijn 3 onderdelen vereist: Een gebruikersnaam, wachtwoord en naam van de organisatie. (Zie Figuur 11).

##### Aanvragen

Bedrijfsnaam*	<input type="text"/>
Adres*	<input type="text"/>
Postcode*	<input type="text"/>
Plaats*	<input type="text"/>
Telefoon*	<input type="text"/>
Fax	<input type="text"/>
KvK-nummer*	<input type="text"/>
BTW-nummer	<input type="text"/>

##### Contactpersoon

Voornaam*	<input type="text"/>
Tussenvoegsel	<input type="text"/>
Achternaam*	<input type="text"/>
Geslacht*	<input checked="" type="radio"/> M <input type="radio"/> V
Direct nummer	<input type="text"/>
E-mail*	<input type="text"/>

##### Post adres

Adres	<input type="text"/>
Postbus	<input type="text"/>
Postcode	<input type="text"/>
Plaats	<input type="text"/>

\* VERPLICHT INVULLEN

**Verzenden**

Figuur 10 - HourPower aanvragen

## Login

Gebruikersnaam

Wachtwoord

Organisatie

Inloggen

Figuur 11 - Inloggen

De reden dat er ook een naam van de organisatie opgegeven moet worden is dat het de snelheid bevordert. In dit geval is direct bekend bij welke organisatie gekeken moet worden of de medewerker bekend is. Anders moesten alle systemen doorgezocht worden op zoek naar de medewerker die probeert in te loggen.

Een ander bijkomend voordeel hiervan is dat namen van medewerkers vaker als 1 keer voor kunnen komen. Een medewerker met als gebruikersnaam john.doe kan dan bij zowel bedrijf A als bedrijf B voorkomen.

## **Dashboard**

Wanneer succesvol is ingelogd wordt het dashboard getoond. Het dashboard laat een overzicht zien met nieuwe mededelingen, zoals feedback op ingeleverde uren, en toont een mini agenda met de items van de huidige dag.

Het dashboard is te zien op Figuur 12.

## **Dashboard**

Geen mededelingen

Donderdag 03 dec
Geen afspraken vandaag

**Figuur 12 - Dashboard**

## Afdelingen beheren

Voor de medewerkers met de rechten van beheerder is het mogelijk om afdelingen te beheren. Dat wil zeggen: aanmaken en bewerken.

Bij het aanmaken van een afdeling is er maar 1 onderdeel vereist, de naam. Het kiezen van een afdelingmanager is optioneel evenals het geven van een omschrijving.

### Afdeling aanmaken

Afdelingsnaam\*

Manager 

▼

Omschrijving

\* VERPLICHT INVULLEN

Verzenden

Figuur 13 - Afdeling aanmaken

Niet iedere medewerker kan een afdelingmanager worden. Hiervoor moet de medewerker minimaal manager of beheerder rechten hebben.

Wanneer een afdeling is aangemaakt, verschijnt deze in het afdeling overzicht. Hier kan op de naam van een afdeling worden geklikt waarna een overzicht verschijnt van de afdeling.

In dit overzicht verschijnen de naam, manager en omschrijving en alle medewerkers die zijn ingedeeld in deze afdeling.

# Afdeling

<b>Afdelingnaam</b>	Directie
<b>Afdelingmanager</b>	Jeffrey Veldkamp
<b>Omschrijving</b>	Omschrijving van de afdeling.

## Personeelsleden

Naam	Functie	Status	Urenoverzicht	Projecten overzicht
<a href="#">Jeffrey Veldkamp</a>		actief	Overzicht	Overzicht
<a href="#">Daniël van Zoelen</a>		actief	Overzicht	Overzicht

Figuur 14 - Afdeling detailpagina

Het indelen van een medewerker in een afdeling gebeurt bij de medewerker zelf.



## Medewerkers beheren

Ook het beheren van medewerkers is alleen beschikbaar voor beheerders.

Bij het aanmaken van een medewerker zijn een aantal velden verplicht om in te vullen. In dit geval zijn dat: voornaam, achternaam, geslacht en e-mail. Deze worden gebruikt om een nieuwe gebruikersnaam te genereren en om deze samen met een wachtwoord op te sturen naar het opgegeven e-mailadres.

Net als bij het aanvragen van het pakket wordt ook hier het e-mailadres van de contactpersoon gecontroleerd op basis van het opgegeven domeinnaam.

Wanneer er al een combinatie bestaat van voornaam en achternaam dan wordt er gekeken hoeveel dit er zijn en zal er een cijfer achter deze combinatie worden geplakt. Wanneer jan.janssen dus al een keer bestaat binnen het bedrijf, dan zal een nieuwe Jan

Janssen jan.janssen1 als gebruikersnaam krijgen.




Naast deze gegevens is het ook noodzakelijk om te selecteren wat de rechten van de nieuwe medewerker zijn. Hierbij kan men kiezen uit beheerder, manager en gebruiker. Wat de verschillen zijn tussen deze rollen staat beschreven in het Use-case Diagram uit het Functioneel Ontwerp.

Behalve dat kunnen er nog meer zaken opgegeven worden, zoals het uurloon en het uurtarief. Wanneer deze ingevuld worden dan kan op basis van de geregistreerde uren een overzicht worden gegenereerd hoeveel de medewerker gekost heeft en wat hij heeft opgeleverd.

Ook is het mogelijk om een minimum aantal uren in te voeren (met een maximum van 40) waaraan een medewerker wekelijks moet voldoen.

Zodra alle (verplichte) velden correct zijn ingevoerd, wordt de medewerker toegevoegd en zal verschijnen in het medewerker overzicht. Net als bij afdelingen is de naam van de medewerker klikbaar. Wanneer hierop geklikt wordt verschijnt een detail pagina van de medewerker (zie Figuur 16).

## Medewerker aanmaken

Voornaam*	<input type="text"/>
Tussenvoegsel	<input type="text"/>
Achternaam*	<input type="text"/>
Geslacht*	<input checked="" type="radio"/> Man <input type="radio"/> Vrouw
Adres	<input type="text"/>
Postcode	<input type="text"/>
Woonplaats	<input type="text"/>
Telefoon vast	<input type="text"/>
Telefoon mobiel	<input type="text"/>
E-mail*	<input type="text"/>
Functie	<input type="text"/>
Afdeling	<input type="text"/>
Geboortedatum	<input type="text"/> 
Datum in dienst	<input type="text"/> 
Datum uit dienst	<input type="text"/> 
Uurloon	<input type="text"/>
Uurtarief	<input type="text"/>
Minimale uren	<input type="text"/>
Rechten	Beheerder <input type="text"/>
Medewerker code	<input type="text"/>

\* VERPLICHT INVULLEN

**Verzenden**

Figuur 15 - Medewerker aanmaken

## Medewerker

<b>Voornaam</b>	Jeffrey	<b>E-mail</b>	veldkampj@gmail.com
<b>Tussenvoegsel</b>		<b>Functie</b>	Directeur
<b>Achternaam</b>	Veldkamp	<b>Afdeling</b>	Directie
<b>Geslacht</b>	M	<b>Geboortedatum</b>	26-11-2009
<b>Adres</b>	Weegbree 10	<b>Datum in dienst</b>	18-11-2009
<b>Postcode</b>	3641 VC	<b>Datum uit dienst</b>	06-11-2009
<b>Woonplaats</b>	Mijdrecht	<b>Status</b>	actief
<b>Telefoon vast</b>	0297242275	<b>Uurloon</b>	€ 25,00
<b>Telefoon mobiel</b>	0652381718	<b>Uurtarief</b>	€ 60
<b>Rechten</b>	Beheerder	<b>Minimale uren</b>	40

### Externe projecten

Project	Code	Start	Deadline	Uren
---------	------	-------	----------	------

Nog geen projecten

### Interne projecten

Project	Code	Start	Deadline	Uren
---------	------	-------	----------	------

Nog geen projecten

### Uren per week

Figuur 16 - Medewerker detailpagina

Op deze pagina staat een overzicht van alle gegevens die zijn ingevuld tijdens het registreren van de medewerker. Ook is daar een overzicht te vinden van de in- en externe projecten waarbij de medewerker is ingedeeld en de geregistreerde uren van de laatste 3 weken.

Wanneer het nodig is kan een medewerker worden gewijzigd. Er wordt dan een pagina getoond die vrijwel identiek is als het aanmaken van een medewerker met als verschil dat er een keuzemogelijkheid is voor actief en inactief. Ook kan op deze pagina het wachtwoord van de medewerker worden gewijzigd.

Om het systeem ook toegankelijk te maken voor bedrijven die al gebruik maken van een urenregistratiesysteem is het mogelijk om medewerkers te importeren vanuit een CSV (Comma Separated Values) bestand.

Een CSV bestand ziet er als volgt uit:

```
medewerkerID,voornaam,achternaam,adres,postcode,woonplaats
1,Jeffrey,Veldkamp,Weegbree 10,3641 VC,Mijdrecht
```

Er is gekozen voor dit formaat omdat vrijwel elke database de mogelijkheid biedt om de inhoud te exporteren naar een CSV bestand.

Het importeren van medewerkers bestaat uit 2 stappen. De eerste is om het bestand op te geven. Hier worden alleen CSV bestanden geaccepteerd.

## Medewerkers importeren

Bestand:

Figuur 17 - Medewerkers importeren stap 1

Wanneer het bestand opgegeven en geaccepteerd is zal stap 2 verschijnen.

## Medewerkers importeren

Medewerkercode	<input type="text"/>
Voornaam*	<input type="text"/>
Tussenvoegsel	<input type="text"/>
Achternaam*	<input type="text"/>
E-mail	<input type="text"/>
Adres	<input type="text"/>
Postcode	<input type="text"/>
Woonplaats	<input type="text"/>
Telefoon vast	<input type="text"/>
Telefoon mobiel	<input type="text"/>
Geboortedatum	<input type="text"/>
Geslacht	<input type="text"/>

\* VERPLICHT INVULLEN

Figuur 18 - Medewerkers importeren stap 2

Omdat het vrijwel onmogelijk is dat het systeem exact gelijk is aan een CSV bestand van een klant moet opgegeven worden welk veld uit het bestand bij welk onderdeel binnen het systeem hoort.

Zodra dit is gedaan, en de verplichte velden ook geselecteerd zijn, zal de inhoud van het bestand worden opgeslagen in de database.

Na het opslaan van de medewerkers zal gekeken worden of er een e-mailadres is opgeslagen. Als dit het geval is dan zal de medewerker op dit e-mailadres een e-mail ontvangen met de bijbehorende inloggegevens. Is dit niet het geval dan zal de beheerder die de import uitgevoerd heeft een e-mail ontvangen met deze gegevens.

## Klanten beheren

Het aanmaken van klanten is voor iedereen beschikbaar. Het importeren mag alleen door beheerders worden gedaan.

Bij het aanmaken van een klant zijn een aantal velden verplicht om in te vullen: Klantnummer, klantnaam, adres, postcode, plaats en telefoon. Ook is het een vereiste om naam en e-mail van een contactpersoon op te geven.

Nadat een klant is toegevoegd aan het systeem zal deze in het overzichtschermbild verschijnen. In dit scherm kan op de naam van de klant geklikt worden waarna een detailpagina getoond wordt.

Op de detailpagina worden alle gegevens getoond zoals ingevuld bij het aanmaken van de klant. Onderaan de pagina staat echter een overzicht met projecten van de klant.

Net als met medewerkers is het mogelijk om klanten te importeren. Ook hier is de eerste stap om een CSV bestand te uploaden. Vervolgens verschijnt er wederom een scherm waarbij per onderdeel aangegeven moet worden welk veld erbij hoort.

Hierbij is het overigens niet verplicht om een

contactpersoon op te geven, in tegenstelling tot het aanmaken en bewerken van een klant. Dit hebben we gedaan omdat niet zeker is dat er altijd een contactpersoon aanwezig is in het CSV document wat geïmporteerd wordt.

## Klant aanmaken

Klantnummer*	<input type="text"/>
Klantnaam*	<input type="text"/>
Adres*	<input type="text"/>
Postcode*	<input type="text"/>
Plaats*	<input type="text"/>
Telefoon*	<input type="text"/>
Fax	<input type="text"/>
KvK-nummer	<input type="text"/>
BTW-nummer	<input type="text"/>

### Contactpersoon

Voornaam*	<input type="text"/>
Tussenvoegsel	<input type="text"/>
Achternaam*	<input type="text"/>
Geslacht*	<input checked="" type="radio"/> M <input type="radio"/> V
Vast nummer	<input type="text"/>
Mobiel nummer	<input type="text"/>
E-mail*	<input type="text"/>

### Post adres

Adres	<input type="text"/>
Postbus	<input type="text"/>
Postcode	<input type="text"/>
Plaats	<input type="text"/>

\* VERPLICHT INVULLEN

**Verzenden**

Figuur 19 - Klant aanmaken

## Projecten beheren

In tegenstelling tot afdelingen beheren en medewerkers beheren is projecten beheren beschikbaar voor zowel beheerders als managers.

Zoals in het afhankelijkheidsdiagram (Figuur 9) te zien is, zijn projecten enorm belangrijk binnen het systeem. Zonder projecten kunnen er geen activiteiten gemaakt worden en dus geen uren worden geregistreerd.

Bij het aanmaken van projecten zijn slechts 2 velden verplicht om in te vullen, namelijk de naam en of het project intern is of niet. Zo niet dan moet ook de klant worden geselecteerd. Wanneer intern aangeklikt wordt (zie Figuur 20) dan zal het menu met klanten onzichtbaar worden.

In tegenstelling tot een afdelingmanager kan een projectmanager wel een willekeurige medewerker zijn, ongeacht zijn rechten. Wanneer deze geselecteerd wordt zal de medewerker in de lijst ook direct geselecteerd worden. Door medewerkers in deze lijst te selecteren worden ze toegevoegd aan het project. Bij het bewerken van een project kunnen extra medewerkers worden toegevoegd aan een project, of juist medewerkers uit een project worden gehaald.

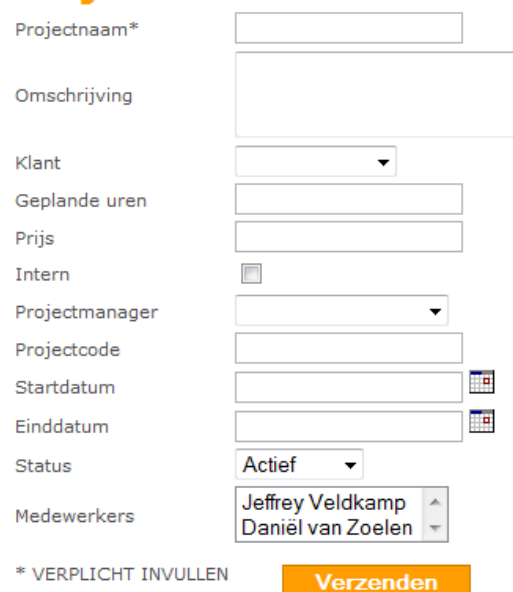
Medewerkers koppelen is noodzakelijk om ze uren te kunnen laten schrijven op projecten. Wanneer ze niet zijn ingedeeld in een project is uren schrijven hierop dus onmogelijk.

Voordat het project wordt opgeslagen wordt een controle uitgevoerd of de einddatum ook daadwerkelijk na de startdatum valt. Zo niet dan zal het project niet worden opgeslagen en een foutmelding worden getoond.

Net als bij de vorige onderdelen hebben ook projecten een detail pagina. Op deze pagina wordt de algemene informatie van het project getoond maar ook hoeveel uur er reeds geschreven is, welke medewerkers ingedeeld zijn bij dit project, welke activiteiten dit project bevat en het is mogelijk om nieuwe activiteiten toe te voegen.

Ook projecten kunnen geïmporteerd worden vanuit een CSV bestand. Dit werkt op dezelfde manier als het importeren van medewerkers en klanten.

## Project aanmaken





Figuur 20 - Project aanmaken

## Project

Projectnaam	HourPower
Omschrijving	Afstudeeropdracht voor de HU
Klant	
Uren begroot	250
Uren gemaakt	0
Prijs	€ 0,00
Intern	Ja
Projectmanager	Jeffrey Veldkamp
Projectcode	HP
Startdatum	01-09-2009
Einddatum	31-01-2010
Status	actief

### Activiteiten

Activiteit	Uren begroot	Uren gemaakt	Bewerken
Programmeren	250	0	
Ontwerpen	150	0	

Activiteit toevoegen  Activiteiten importeren

### Medewerkers

Medewerker	Functie	Status
Jeffrey Veldkamp	Directeur	actief
Daniël van Zoelen		actief

Figuur 21 - Project detailpagina

## Activiteiten beheren

Net als projecten beheren is dit toegankelijk voor zowel beheerders als managers.

Om zo nauwkeurig mogelijk te registreren wordt gebruik gemaakt van activiteiten. Zo kan niet alleen per project, maar ook per activiteit bekeken worden hoeveel tijd eraan besteed wordt en hoeveel ervoor begroot is.

Bij het aanmaken van een activiteit is alleen de naam verplicht. Andere zaken als omschrijving, uren en tarief zijn optioneel. Wanneer een tarief ingesteld wordt bij een activiteit dan zal dit verkozen worden boven het uurtarief van een medewerker.

### Activiteit toevoegen

Activiteitnaam*	<input type="text"/>
Omschrijving	<input type="text"/>
Uren	<input type="text"/>
Tarief	<input type="text"/>
Status	<input type="text" value="Actief"/>

\* VERPLICHT INVULLEN

Figuur 22 - Activiteit toevoegen

Evenals de andere onderdelen bevatten activiteiten ook een detailpagina. Op deze pagina staan de algemene gegevens zoals ingevuld bij het aanmaken en de geregistreeerde uren op deze activiteit (zie Figuur 23).

Ook activiteiten kunnen geïmporteerd worden. Op een projectdetail pagina staat een overzicht van de bijbehorende activiteiten (zie Figuur 21). Onder dit overzicht kan gekozen worden om activiteiten aan te maken, maar ook om deze te importeren. Het importeren van activiteiten gaat op dezelfde manier zoals bij medewerkers staat beschreven.

### Activiteit

Activiteitnaam	Programmeren
Omschrijving	Code kloppen
Geplande uren	250
Tarief	€ 50,00
Status	actief

#### Geregistreeerde uren

Figuur 23 - Activiteit detailpagina

## Uren registreren

Uiteraard is dit voor iedereen beschikbaar.

Bij het openen van dit venster wordt gekeken welke week het is en vervolgens worden de bijbehorende data getoond. Onder project zijn alle projecten beschikbaar waarbij een medewerker is ingedeeld. Zodra een project geselecteerd wordt, wordt er een AJAX request verzonden waarna de bijbehorende activiteiten geladen worden. Wanneer geprobeerd wordt om uren op te slaan zonder dat een activiteit is geselecteerd dan zal er een rode rand verschijnen om de betreffende activiteit.

## Uren opgeven



Week ≤ Week 46 - 2009 ≥

Project*	Activiteit*	ma 09-11	di 10-11	wo 11-11	do 12-11	vr 13-11	za 14-11	zo 15-11	Week totaal
									0

Dag totaal 0 0 0 0 0 0 0

Nieuwe regel toevoegen + Opslaan

Figuur 24 - Uren opgeven

Na het invoeren van de gewerkte uren wordt bekeken of de ingevoerde waarde een getal is en of deze groter is dan 0 en kleiner dan (of gelijk aan) 24. Wanneer dit het geval is dan worden direct het weektotaal en dagtotaal aangepast. Door op een ballon naast het invoerveld te klikken verschijnt een venster waarin opmerkingen toegevoegd kunnen worden (zie Figuur 25). Standaard zijn deze ballonnen grijs gekleurd. Wanneer een opmerking is toegevoegd dan zal deze oranje worden. Op deze manier blijft duidelijk bij welke uren opmerkingen zijn toegevoegd.

## Uren opgeven



Week ≤ Week 46 - 2009 ≥

Project*	Activiteit*	ma 09-11	di 10-11	wo 11-11	do 12-11	vr 13-11	za 14-11	zo 15-11	Week totaal
									0

Dag totaal

Nieuwe regel toevoegen + Opslaan

Figuur 25 - Opmerking toevoegen

Wanneer geprobeerd wordt om een opmerking toe te voegen dan wordt eerst gecontroleerd of er al een activiteit geselecteerd is. Zo niet dan zal boven het invoerveld een melding verschijnen dat dit eerst moet gebeuren.

Tijdens het realiseren van het toevoegen van opmerkingen zijn een aantal lastige punten naar voren gekomen. Voor het toevoegen van een opmerking wordt namelijk gebruik gemaakt van een apart formulier. Deze wordt dus los verzonden van de uren zelf.

Op het moment dat reeds uren geregistreerd zijn en opmerkingen worden toegevoegd levert dit geen problemen op. In dit geval is het een kwestie van die uren updaten. Het werd echter lastiger op

het moment dat geprobeerd werd een opmerking toe te voegen bij uren die nog opgeslagen moesten worden.

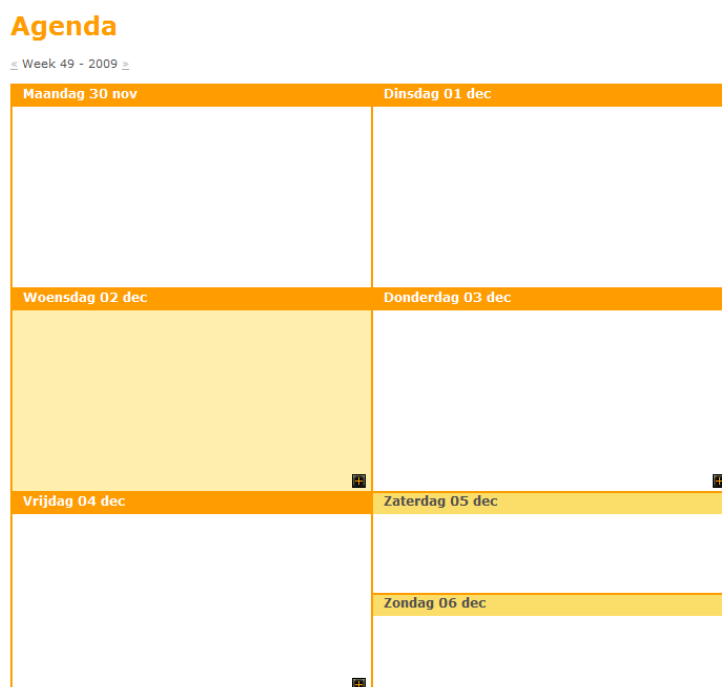
Dit is opgelost door deze opmerkingen tijdelijk in een sessie te zetten met een referentie naar de activiteit en de datum. Wanneer de uren dan opgeslagen worden wordt in de sessie gekeken of er nog opmerkingen staan die toegevoegd moeten worden. Mocht dit het geval zijn dan worden deze opmerkingen toegevoegd en uit de sessie verwijderd.



## Agenda beheren

De agenda is voor alle rollen binnen het systeem beschikbaar.

Op het moment dat een medewerker wordt toegevoegd aan het systeem wordt er ook direct een agenda voor hem aangemaakt. Figuur 26 toont de agenda van het systeem.



Figuur 26 - Agenda

Net als het registreren van de uren wordt bij de agenda eerst gekeken in welke week we op dit moment zitten. Deze week wordt standaard geladen. Met de pijlen naast het weeknummer en het jaartal kan worden gewisseld van week.

Zoals te zien is op Figuur 26 is de huidige dag anders gekleurd. Dit is gedaan om direct duidelijk te maken welke dag het is. Wat ook opvalt is het feit dat het niet mogelijk is om in het verleden agenda items toe te voegen. Voor de zaterdag en zondag hangt dit af van de instellingen van het systeem.

Op het moment dat bovenstaande afbeelding werd gemaakt stond de instelling “Zet zaterdag en zondag aan in het registratiesysteem” uit. Hierdoor is het niet mogelijk om agenda items toe te voegen op deze dagen.

Wanneer op de + wordt geklikt verschijnt onderstaand venster:



The form is titled 'x' in the top right corner. It contains the following fields and controls:

- Datum:** A text input field containing '02-12-2009'.
- Onderwerp\*:** A text input field.
- Starttijd:** Two dropdown menus showing '00' and '00' separated by a colon.
- Eindtijd:** Two dropdown menus showing '00' and '00' separated by a colon.
- Omschrijving:** A large text area.
- Verwerken in gewerkte uren:** A checkbox.
- \*Verplicht invullen:** A label indicating that the subject field is mandatory.
- Opslaan:** An orange button to save the item.

**Figuur 27 - Agenda item toevoegen**

Er zijn een aantal regels waar aan moet worden voldaan voordat een agenda item toegevoegd wordt. Zo is het onderwerp verplicht en moet de eindtijd **na** de starttijd vallen.

Wanneer aan de voorwaarden is voldaan dan verdwijnt het venster en wordt de afspraak in de agenda getoond (zie Figuur 28).

## Agenda

≤ Week 49 - 2009 ≥

Maandag 30 nov	Dinsdag 01 dec
<b>Woensdag 02 dec</b> 00:00 - 01:00 <a href="#">Afspraak</a> <span style="color: red;">x</span>	<b>Donderdag 03 dec</b> 
<b>Vrijdag 04 dec</b> 	<b>Zaterdag 05 dec</b> 
	<b>Zondag 06 dec</b> 

**Figuur 28 - Agenda met agenda item**

Een toegevoegd agenda item kan bewerkt worden, mits deze niet in het verleden staat. Is dit wel het geval dan kan de afspraak alleen worden bekeken. Hetzelfde geldt voor het verwijderen ervan.

## Systeeminstellingen

De instellingen zijn alleen beschikbaar voor beheerders.

Op het moment van schrijven waren er slechts 3 beschikbare instellingen voor het systeem. Deze zijn te zien op Figuur 29. De instellingen gelden voor het volledige systeem en voor alle medewerkers.



Figuur 29 - Systeeminstellingen

De instellingen spreken eigenlijk voor zich maar voor de duidelijkheid worden deze hieronder toch besproken.

### *Feedback kunnen geven op ingeleverde uren*

Deze instelling zorgt ervoor dat project- en afdelingmanagers feedback kunnen geven op ingeleverde uren of opmerkingen bij ingeleverde uren. Op het moment dat er feedback is gegeven op ingeleverde uren dan zal hier een melding over verschijnen op het dashboard.

### *Accordeersysteem: Week inleveren en goedkeuren*

Niet bij elk bedrijf hoeven uren geaccordeerd te worden. Deze instelling zorgt ervoor dat bij het opgeven van uren wel of geen knop verschijnt om de uren in te leveren om te worden geaccordeerd.

### *Zet zondag en zaterdag aan in het registratiesysteem*

Zoals besproken bij de agenda is het mogelijk om het registreren van uren en agenda items op zaterdag en zondag uit te schakelen. Standaard staat deze optie uitgeschakeld. Mochten bedrijven toch de mogelijkheid willen bieden om ook op zaterdag en zondag uren te registreren (bijv. omdat zij op zaterdag geopend zijn) dan is dat hier mogelijk.

In het verloop van dit project is het de bedoeling dat er meerdere instellingen toegevoegd zullen worden. Enkele ideeën hiervoor zijn:

- Meldingen tonen in het dashboard (ja of nee);
- Klantnummer laten beginnen bij een op te geven getal;
- Opmerkingen toevoegen bij geregistreerde uren (ja of nee).

## 4.5 Testen

Uiteraard moesten de gerealiseerde onderdelen ook getest worden. Tijdens het realiseren zijn tussendoor tests uitgevoerd om zo te kunnen oordelen op de goede weg te zijn. Na het realiseren werd aan een ander projectlid gevraagd om een goede test uit te voeren en deze te documenteren.

Deze tests zijn vastgelegd in een testrapport. Doordat de realisatie nog niet volledig was afgerond tijdens het schrijven van deze scriptie, was ook het testrapport nog niet volledig. Om deze reden is deze niet toegevoegd als bijlage.

Tests zoals deze vastgelegd zijn in het testrapport zien er als volgt uit:

Pagina: Project aanmaken							
Testnummer	Projectnaam*	Klant**	Intern**	Geplande uren	Prijs	Startdatum	Einddatum
1							
2	HourPower						
3	HourPower		Ja				
4	HourPower	Jeffrey B.V.					
5	HourPower		Ja	-1			
6	HourPower	Jeffrey B.V.			-1		
7	HourPower	Jeffrey B.V.				31-12-2009	30-12-2009
8	HourPower	Jeffrey B.V.				30-12-2009	31-12-2009

Tabel 3 - Testrapport, ingevoerde waardes

\* verplicht veld

\*\* één van deze velden moet worden geselecteerd

Testnummer	Verwachting	Resultaat
1	Foutmelding op projectnaam, klant en intern	Verwachting klopt
2	Foutmelding op klant en intern	Verwachting klopt
3	Geen foutmelding	Verwachting klopt
4	Geen foutmelding	Verwachting klopt
5	Foutmelding op geplande uren	Project wordt opgeslagen
6	Foutmelding op prijs	Project wordt opgeslagen
7	Foutmelding op einddatum	Verwachting klopt
8	Geen foutmelding	Verwachting klopt

Tabel 4 - Testrapport, resultaten

Met bovenstaande test wordt de use-case niet als afgerond beschouwd. Er zijn nog twee situaties waarbij de verwachte uitkomst anders is dan het resultaat. Deze moeten eerst worden opgelost voordat gestart mag worden met een nieuwe use-case.

Het is de bedoeling om het systeem ook te laten testen door mensen die er in de toekomst gebruik van zouden kunnen maken. Op het moment van schrijven is dat nog niet gelukt. Het doel is om dit gedaan te hebben voor het einde van dit project.

## 5 Conclusies en aanbevelingen

In hoofdstuk 2 van deze scriptie zijn een aantal succesfactoren genoemd die van essentieel belang waren voor een goed verloop van het project:

- **Overleg met projectbegeleider.** Als ergens vragen over ontstonden dan moest een afspraak worden gemaakt met de projectbegeleider. Dit is meerdere malen gebeurd tijdens het project, waaronder ook een tussentijdse presentatie om de voortgang te tonen. Ook werd wekelijks een highlight report gestuurd om de projectbegeleider op de hoogte te houden.
- **Goede werkomgeving.** Aan het begin van het project is geprobeerd een ruimte bij Cube te krijgen. Dit is helaas niet gelukt. Gelukkig was het mogelijk om gedurende het hele project vrijwel elke dag in hetzelfde projecthokje kunnen werken. Dit had als bijkomend voordeel dat de docentbegeleider dicht in de buurt was.
- **Kennis van de opdracht.** Door de uitgevoerde onderzoeken en het gemaakte Functioneel Ontwerp is in de eerste weken van het project een duidelijk beeld ontstaan van het product wat opgeleverd moest worden. Hierdoor was de kennis bij beide projectleden voldoende aanwezig.
- **Kennis van de ontwikkeltools.** De meeste gebruikte technieken om het systeem te realiseren zijn tijdens de opleiding al meerdere keren aan bod gekomen. Daniël van Zoelen had soms wat problemen met AJAX maar daar heb ik hem bij kunnen helpen.

Tijdens dit project is geleerd dat er reeds online urenregistratiesystemen bestaan. Deze systemen bieden over het algemeen meer aan dan alleen urenregistratie. Hier is tijdens de realisatie rekening mee gehouden door een architectuur te gebruiken welke goed onderhoudbaar en uitbreidbaar is.

Ondanks het feit dat een Outlook add-on niet haalbaar was tijdens dit project is het wel gelukt om een systeem te realiseren wat zich onderscheidt. Zo is het mogelijk om met de mobiele telefoon toegang tot het systeem te krijgen en kan bestaande data eenvoudig worden geïmporteerd.

Doordat niet zomaar functionaliteiten gerealiseerd zijn, maar onderzoek is gedaan naar de wensen van verschillende bedrijven middels een MoSCoW tabel, biedt het systeem onderdelen aan waar ook werkelijk vraag naar is.

Bovenstaande punten zorgen ervoor dat HourPower een pakket is wat aan voorafgestelde eisen voldoet. Het systeem is makkelijk te onderhouden en uit te breiden, weet zich te onderscheiden van bestaande systemen en sluit aan bij de behoefte van bedrijven.

### 5.1 Aanbevelingen

Tijdens het project is onderzoek gedaan naar de haalbaarheid van een Outlook add-on om HourPower te onderscheiden van de bestaande online urenregistratiesystemen. In dit onderzoek is de conclusie getrokken dat dit binnen dit project niet haalbaar is. In de toekomst moet overwogen worden om hier alsnog een project van te maken.

Wat in de toekomst zeker gedaan moet worden is een beheer gedeelte realiseren. Op het moment is in de database wel een tabel aanwezig voor administrators maar hier is nog geen apart beheersysteem voor. Nu moet alles nog rechtstreeks in de database worden toegevoegd en bekeken.

Nog een onderdeel wat in de toekomst wellicht gerealiseerd kan worden is een klanten login. Dit verzoek is onlangs binnengekomen van een potentiële klant. Hij is op zoek naar een systeem als HourPower maar zou ook graag zien dat klanten in kunnen loggen en daar de geboekte uren op hun projecten kunnen bekijken.

Zoals gezegd bieden de bestaande pakketten meer aan dan alleen urenregistratie. Het is belangrijk om naast een goed urenregistratiesysteem ook te zorgen dat het systeem extra functies als facturatie en planning gaat bevatten. Hierdoor kan HourPower nog beter de concurrentie aangaan met de bestaande online urenregistratiesystemen.

## 6 Evaluatie van de procesgang

Deze evaluatie wil ik beginnen bij de begeleiding van de projectbegeleider, Gérard van Elsen. We hadden aan het begin van het project de afspraak gemaakt dat er overleg plaats zou vinden op verzoek van mij en Daniël van Zoelen. Daarnaast zouden wij wekelijks een highlight report opsturen met de geboekte voortgang en eventuele problemen waar wij tegenaan waren gelopen.

Persoonlijk vind ik dit een fijne manier van werken en dit heb ik dan ook als positief ervaren. Het gebeurt nog wel eens dat wekelijks overleg wordt gepleegd waarin soms niets te bespreken valt. Dit vind ik zeer vervelend en zonde van mijn tijd.

Op de momenten dat wij wel gesprekken hadden met meneer van Elsen dan had hij altijd nuttig advies waar wij mee verder konden.

Naast de begeleiding van meneer van Elsen was er ook begeleiding van onze docentbegeleider, Jeroen Weber. Hij heeft tijdens het project meerdere malen naar mijn scriptie gekeken en voorzien van feedback. Dankzij deze feedback heb ik mijn scriptie naar mijn idee flink kunnen verbeteren.

Het uitvoeren van het project zelf is vrij soepel verlopen en volgens de planning verlopen. De uitgevoerde onderzoeken en het Functioneel Ontwerp waren zelfs ver voor de deadline al afgerond. Hierdoor hebben wij aardig wat tijd gehad om aan de andere onderdelen van het project te werken.

Ook de samenwerking met Daniël van Zoelen heb ik als prettig ervaren. Wij kennen elkaar inmiddels 8 jaar en hebben in die periode meerdere projecten samen afgerond. Ook dit project heeft weer een prettige samenwerking opgeleverd. Dit is een mooi vooruitzicht gezien het feit dat wij samen een bedrijf op willen starten.

Alles bij elkaar opgeteld is dit een positief en leerzaam afstudeerproject geweest waarin ik veel geleerd heb.

Mijn dank aan meneer van Elsen en de Hogeschool Utrecht.

Jeffrey Veldkamp.

## Lijst met afkortingen

<b>AJAX</b>	Asynchronous JavaScript and XML. Dit is een taal waarmee interactieve web applicaties gebouwd kunnen worden. Zo is het bijvoorbeeld mogelijk om formulieren te valideren zonder dat deze verzonden hoeven te worden.
<b>CAV</b>	Content Area View. Een weergave in UiaML waarin een (web)pagina wordt uitgewerkt en gespecificeerd.
<b>CSV</b>	Comma Separated Values. Een bestand waarin de waardes worden gescheiden door een kommateken. Dit formaat wordt door vrijwel elke database ondersteund.
<b>DSDM</b>	Dynamic Systems Development Method. Een ontwikkelmethode welke iteratief te werk gaat. Deze wordt voornamelijk gebruikt voor de ontwikkeling van gecomputeriseerde informatiesystemen.
<b>HTML</b>	Hypertext Markup Language. Een standaard welke gebruikt wordt om de opmaak van webpagina's te verzorgen
<b>JSON</b>	JavaScript Object Notation. Wordt gebruikt voor het uitwisselen van datastructuren in met name webapplicaties die asynchroon data ophalen van de webserver. Een voorbeeld hiervan is AJAX.
<b>MVC</b>	Model-View-Controller. Een ontwerp patroon dat het ontwerp van complexe toepassingen opdeelt in drie onderdelen met ieder verschillende verantwoordelijkheden.
<b>PDF</b>	Portable Document Format. Een universele bestandsindeling waarmee lettertypen, afbeeldingen en lay-out van elk willekeurig brondocument behouden blijven. Ongeacht het programma of platform waarmee het document werd gemaakt.
<b>PHP</b>	PHP: Hypertext Preprocessor (voorheen: Personal Home Page). Een scripttaal welke op de webserver wordt uitgevoerd. De output is over het algemeen in HTML.
<b>SMV</b>	Sitemap View. Een weergave in UiaML waarin het volledige systeem getoond wordt met afhankelijkheden, events en bijbehorende (fout)meldingen.
<b>SVN</b>	Subversion. Een versiebeheersysteem wat het mogelijk maakt om met meerdere mensen tegelijk in hetzelfde document te werken. Ook is het mogelijk om oudere revisies van bestanden terug te halen.
<b>SQL</b>	Structured Query Language. Een taal die wordt gebruikt om selecties uit te voeren op een database. Bekende commando's zijn SELECT, INSERT, UPDATE en DELETE.
<b>UiaML</b>	Universal information architecture Modeling Language. Een modelleringstaal welke gebruikt wordt om de functionele werking van een applicatie in beeld te brengen.
<b>XML</b>	Extensible Markup Language. Een standaard van het World Wide Web Consortium waarmee gestructureerde gegevens weergegeven kunnen worden in de vorm van platte tekst. Dit is leesbaar voor zowel mens als machine.



## Lijst van figuren en tabellen

Figuur 1 - Use case diagram .....	20
Figuur 2 - Sitemap View.....	21
Figuur 3 - Content Area View .....	22
Figuur 4 - Inloggen preview .....	22
Figuur 5 - Database ontwerp.....	23
Figuur 6 - Sequence diagram inloggen .....	24
Figuur 7 - Architectuur .....	26
Figuur 8 - Verzenden van het inlog formulier .....	27
Figuur 9 - Afhankelijkheidsdiagram .....	28
Figuur 10 - HourPower aanvragen .....	29
Figuur 11 - Inloggen.....	30
Figuur 12 - Dashboard .....	31
Figuur 13 - Afdeling aanmaken.....	32
Figuur 14 - Afdeling detailpagina .....	32
Figuur 15 - Medewerker aanmaken .....	33
Figuur 16 - Medewerker detailpagina .....	34
Figuur 17 - Medewerkers importeren stap 1 .....	34
Figuur 18 - Medewerkers importeren stap 2 .....	35
Figuur 19 - Klant aanmaken.....	36
Figuur 20 - Project aanmaken .....	37
Figuur 21 - Project detailpagina .....	37
Figuur 22 - Activiteit toevoegen .....	38
Figuur 23 - Activiteit detailpagina .....	38
Figuur 24 - Uren opgeven .....	39
Figuur 25 - Opmerking toevoegen.....	39
Figuur 26 - Agenda .....	41
Figuur 27 - Agenda item toevoegen .....	42
Figuur 28 - Agenda met agenda item .....	42
Figuur 29 - Systeeminstellingen .....	43
 Tabel 1 - Opleverdata .....	 9
Tabel 2 - Mappenstructuur.....	25
Tabel 3 - Testrapport, ingevoerde waardes .....	44
Tabel 4 - Testrapport, resultaten .....	44

## Literatuurlijst

Buschmann, F, Meunier, R, Rohnert, H, Sommerlad, P, Stal, M, *Pattern: Presentation-Abstraction-Control* - <http://www.vico.org/pages/PatronsDisseny/Pattern%20Presentation%20Abstra/>

Core J2EE Pattern Catalog, *Core J2EE Patterns – Front Controller* - <http://java.sun.com/blueprints/corej2eepatterns/Patterns/FrontController.html>

Diract IT, *Waterval methode* - <http://www.diract-it.nl/OverDiract/Projecten-en-ethoden/Waterval-Ontwikkelmethode>

DSDM Consortium, *White Paper: Using DSDM and PRINCE2* - [http://www.dsdm.org/knowledgebase/download/32/using\\_dsdm\\_and\\_prince2\\_white\\_paper.pdf](http://www.dsdm.org/knowledgebase/download/32/using_dsdm_and_prince2_white_paper.pdf)

Florijn, G, *Whitepaper Software architectuur* - [http://cibit.nl/site.nsf/0/AAAD6925A3453EBDC1256E43003908E4/\\$file/Whitepaper\\_Software\\_architectuur.pdf](http://cibit.nl/site.nsf/0/AAAD6925A3453EBDC1256E43003908E4/$file/Whitepaper_Software_architectuur.pdf) (pagina 22)

Haugney, D, *MoSCoW Method* - <http://www.projectsart.co.uk/moscow-method.html>

Husson, T, *Western European Mobile Forecast, 2009 to 2014* - <http://www.forrester.com/Research/Document/Excerpt/0,7211,53717,00.html>

JSON, *JSON in JavaScript* - <http://www.json.org/js.html>

Krug, S, *Don't make me think! A Common Sense Approach to Web Usability* – New Riders Publishing, 2005

Marston, T, *The Model-View-Controller (MVC) Design Pattern for PHP* - <http://www.tonymarston.net/php-mysql/model-view-controller.html>

MSDN, *Document/View Architecture* - <http://msdn.microsoft.com/en-us/library/4x1xy43a%28VS.80%29.aspx>

Ormerod, M, *Defining the OpenEdge® Reference Architecture - Presentation: Model-View-Controller Pattern* - <http://communities.progress.com/pcom/docs/DOC-14628%3B>

phpPatterns, *Design: The front controller and php* - [http://www.phppatterns.com/docs/design/the\\_front\\_controller\\_and\\_php](http://www.phppatterns.com/docs/design/the_front_controller_and_php)

Schall, D, *MVC vs MVP* - <http://www.darronschall.com/weblog/2004/06/mvc-vs-mvp.cfm>

Snyder, T, *MVC or MVP Pattern – Whats the difference?* - [http://blogs.infragistics.com/blogs/todd\\_snyder/archive/2007/10/17/mvc-or-mvp-pattern-whats-the-difference.aspx](http://blogs.infragistics.com/blogs/todd_snyder/archive/2007/10/17/mvc-or-mvp-pattern-whats-the-difference.aspx)

Wokke, A, *Forrester: een vijfde Nederlanders gebruikt internet op telefoon* - <http://tweakers.net/nieuws/61431/forrester-een-vijfde-nederlanders-gebruikt-internet-op-telefoon.html>

## **Bijlagen**

- B1      Onderzoek bestaande pakketten**
- B2      Lijst met functionaliteiten**
- B3      Onderzoek software architectuur**
- B4      Onderzoek Outlook add-on**

**Bijlage B1**

# Onderzoek: reeds bestaande pakketten

Welke pakketten zijn beschikbaar en wat kunnen ze?

Jeffrey Veldkamp  
Daniël van Zoelen  
3-9-2009

## Inhoudsopgave

Inleiding .....	5
Mijnuren.nl .....	6
Functionaliteit .....	6
Modules.....	6
Urenregistratie .....	6
Verlofregistratie .....	6
Planning en budgetten .....	7
Facturatie.....	7
Export .....	8
Usability .....	8
Create a clear visual hierarchy .....	9
Take advantage of conventions.....	9
Break up pages into clearly defined areas .....	9
Make it obvious what's clickable.....	9
Minimize noise .....	9
Tarieven.....	9
Mijnuren online .....	10
Mijnuren standalone server .....	10
Mijnuren managed services .....	10
4ever-uren.nl .....	11
Functionaliteit .....	11
Invullen en bijwerken urenlijst .....	11
Rapportages genereren.....	11
Administratief beheer van het systeem .....	11
Tabellen .....	12
Projecten .....	12
Gebruikers .....	12
Gebruikersgroepen.....	13
Kosten en tarievenstructuur.....	13
Usability .....	13
Create a clear visual hierarchy .....	13
Take advantage of conventions.....	13
Break pages up into clearly defined areas .....	14

Make it obvious what's clickable.....	14
Minimize noise .....	14
Tarieven.....	14
Clockwise.info.....	16
Functionaliteiten .....	16
Urenregistratie .....	16
Urenmanagement .....	16
Rapportages .....	16
Factureren .....	16
Agenda.....	17
Gebruikersprofielen .....	17
Usability.....	17
Create a clear visual hierarchy .....	18
Take advantage of conventions.....	18
Break up pages into clearly defined areas .....	18
Make it obvious what's clickable.....	19
Minimize noise .....	19
Tarieven.....	19
Hosting door ClockWise .....	19
Aanschaf van ClockWise.....	19
NuTijd .....	20
Functionaliteit .....	20
Usability.....	20
Create a clear visual hierarchy .....	21
Take advantage of conventions.....	21
Break up pages into clearly defined areas .....	21
Make it obvious what's clickable.....	22
Minimize noise .....	22
Tarieven.....	22
Consult.....	23
Usability.....	23
Tarieven.....	23
Sumeris.....	24
Usability.....	24

Tarieven.....	24
Conclusie .....	25

## Inleiding

Het concept van online urenregistratie is niet nieuw. Er bestaan reeds een aantal pakketten die deze service bieden. Om deze reden gaan wij onderzoeken welke pakketten dit zijn en wat voor services zij bieden.

Op basis van het resultaat van dit onderzoek kan bekeken worden welke functionaliteiten veel voorkomend zijn en op welk gebied HourPower zich kan onderscheiden van deze pakketten.

Een simpele zoekopdracht in de grote zoekmachines geeft een aardige lijst weer met pakketten die o.a. online urenregistratie aanbieden. De pakketten die wij onder de loep zullen nemen zijn:

- Mijnueren.nl;
- 4ever-uren.nl;
- Clockwise.info;
- Nutijd.nl;
- Comsult.nl;
- Sumeris.nl.



## Mijnuren.nl

Mijnuren.nl biedt een pakket voor urenregistratie, verlof, planning en facturatie.

### Functionaliteit

De functionaliteiten die mijnuren.nl biedt zijn o.a.:

- Registratie van medewerkers, NAW, tarieven (inkoop/kostprijs);
- Registratie van projecten, activiteiten, budgetten, projectmanagers, tarieven (verkoop);
- Registratie van gewerkte uren via een maandstaat of weekstaat;
- Periodeafsluiting, inclusief e-mails naar collega's met openstaande timesheets. Indien nodig accorderen door afdelingsmanagers;
- Overzichten van geboekte uren op project/activiteit en per medewerker;
- Urenstaten in huisstijl van de gebruiker;
- Online mededelingen voor uw personeel;
- Facturatie van geboekte uren;
- Facturen in huisstijl van de gebruiker;
- Facturatie van orders;
- Genereren van herinneringen volgens organisatie-, klant- of factuurspecifieke regels;
- Meertalige gebruikersinterface;

Behalve deze vrij standaard functionaliteiten bieden zij per module nog een breed scala aan functionaliteit. Deze zullen hieronder besproken worden.

## Modules

### Urenregistratie

Bij het registreren van uren kan gekozen worden om de timesheet als maandstaat of weekstaat te tonen. Overige functionaliteiten van de urenregistratie zijn:

- Nieuwsberichten voor communicatie;
- Importeermogelijkheid van klanten, gebruikers en projecten;
- Bulkmutaties;
- Standaard activiteiten zodat projecten snel ingericht kunnen worden;
- Kopieer mogelijkheid van projecten en activiteiten;
- Bijhouden van verjaardagen van collega's;
- Niet-uren gerelateerde activiteitstypes;
- Tijd-voor-tijd regelingen om werkelijke overuren over te hevelen naar verlof cq salaris;
- Kentekenmodule voor rittenregistratie t.b.v. leaseauto's;
- Productdefinitie voor snel aanmaken standaard orders;
- Attenderingen per e-mail voor niet afgesloten timesheets;

### Verlofregistratie

De verlof module van mijnuren.nl biedt de volgende functionaliteit aan de eindgebruiker:

- Verlofrechten, inclusief automatische ophoging per jaar;
- Uitgebreide verlofregelingen per jaar;

- Meerdere verloftypen mogelijk;
- Verlofaanvragen per e-mail;
- Mutatiemogelijkheid;
- Tijd-voor-tijd regeling;
- Volledig inzicht in historie van aanvragen, mutaties en saldoopbouw;
- Koppeling beschikbaarheid aan planningmodule;
- Aanwezig-blok op startpagina;

### Planning en budgetten

De planning en budgetten module van mijnuren.nl biedt de volgende functionaliteit aan de eindgebruiker:

- Globale projectplanning: uren invoeren per project per activiteit per dag of week;
- Toekenning projectplanning: uren invoeren per project per activiteit per medewerker per dag of week;
- Medewerker planning: inzien van de planning van de medewerker;
- Overzicht van planning vs. geboekte uren;
- Overzicht van beschikbaarheid personeel;

### Facturatie

Naar eigen zeggen is de facturatie module van mijnuren.nl ontzettend uitgebreid. Een aantal functionaliteiten die deze module biedt zijn:

- Order-, uur- en handmatige facturen;
- Automatisch genereren factuurvoorstellen per gekozen periode;
- Automatisch genereren van herinneringen op basis van project- of klantspecifieke betalingstermijnen;
- Automatische factuurnummer codering op basis van maskers;
- Factuurnummer per afdeling;
- Herhalende orders;
- Automatische BTW berekening;
- Meerdere valuta;
- Organisatiespecifieke factuur templates;
- Genereren factuur in PDF.

Deze facturatie module wordt op de volgende manieren toegepast:

- Het factureren van gewerkte uren volgens budgetafpraak;
- Het factureren van fixed-price projecten;
- Het factureren van eenmalige en herhalende orders;
- Het factureren van gewerkte uren tegen diverse uurtarieven;
- Het factureren van gewerkte uren in meerdere landen en in meerdere talen voor drie verschillende bedrijven met behulp van meerdere valuta;

Daarnaast gaan ze deze module verder uitbreiden met elektronisch/digitaal factureren, mogelijkheden voor klantenportals en meer.

## Export

Op dit moment kunnen de volgende gegevens worden geëxporteerd naar xml, csv of xls:

- Afdelingen;
- Klanten;
- Medewerkers;
- Medewerker tarieven;
- Projecten;
- Projectactiviteiten;
- Projecttarieven;
- Uren.

De gegevens die geëxporteerd kunnen worden zijn afhankelijk van welke modules in gebruik zijn.

## Usability

Naast functionaliteit is ook usability een belangrijk aspect voor websites. Om deze reden zullen we gaan bekijken of de applicaties voldoen aan de vijf aspecten van het Billboard design 101 van Steve Krug:

1. Create a clear visual hierarchy;
2. Take advantage of conventions;
3. Break pages up into clearly defined areas;
4. Make it obvious what's clickable;
5. Minimize noise.

Er is geen klikbare demo aanwezig op de website van Mijnnuren. Wel staan er een aantal video's waarin getoond wordt hoe een aantal functionaliteiten binnen het programma werken (zie Figuur 1).



Figuur 1 – Aanmaken van een project

### Create a clear visual hierarchy

Volgens deze regel van Krug zijn grote teksten het belangrijkste en kleine teksten onbelangrijk. We scannen pagina's namelijk en lezen ze niet helemaal. Het is dus belangrijk om een goede hiërarchie te hebben om mensen niet op het verkeerde spoor te brengen.

Als we met deze regel in ons achterhoofd gaan kijken naar Figuur 1 dan kunnen we zien dat in dit geval het logo van Mijnnuren Online en het bedrijfslogo de dingen zijn die als belangrijkste onderdelen aangegeven worden. Vervolgens wordt de tekst 'Beheer project' als minder belangrijk gezien. De rest van de tekst wordt gezien de grootte als onbelangrijk gezien volgens de regels van Krug.

### Take advantage of conventions

Ook deze regel zit verwerkt in Figuur 1. Er wordt op de pagina gebruik gemaakt van een knop met daarop de afbeelding van iets wat op een kalender lijkt. Wanneer erop gedrukt wordt verschijnt een pop-up met inderdaad een kalender.

### Break up pages into clearly defined areas

Deze regel zit goed verwerkt in Figuur 1. Er is bovenaan de pagina duidelijk een gebied voor de logo's met daaronder het hoofdmenu. Vervolgens is er een klein gebied met daarin de naam van de pagina en onderaan een gebied met tabbladen en bijbehorende formulieren.

### Make it obvious what's clickable

Op dit punt wordt minder goed gescoord. Op zich is het duidelijk dat de pagina tabs bevat waar dus zeer waarschijnlijk op geklikt kan worden. Het hoofdmenu zou echter verbeterd kunnen worden. Momenteel heeft het hoofdmenu dezelfde stijl als de kop 'Voeg nieuw project toe', welke niet klikbaar is.

### Minimize noise

Ruis is er eigenlijk niet op de pagina. Er staan geen overbodige tierlantijnen op de pagina waardoor alles overzichtelijk blijft.

### Tarieven

Het pakket van mijnnuren wordt op verschillende manieren aangeboden:

- Mijnnuren online;
- Mijnnuren standalone server;
- Mijnnuren managed services.

## Mijnuren online

Jaarlijkse licentiekosten					
Modules	Aantal gebruikers				
	t/m 10	t/m 25	t/m 50	t/m 75	t/m 150
<b>Standaard</b>	€ 504,-	€ 1.181,25	€ 2.047,50	€ 2.835,-	€ 4.725,-
<b>Afdeling</b>	inclusief	inclusief	inclusief	inclusief	inclusief
<b>Excel</b>	inclusief	inclusief	inclusief	inclusief	inclusief
<b>Tarieven</b>	€ 105,-	€ 131,25	€ 157,50	€ 183,75	€ 262,50
<b>Klanten</b>	€ 84,-	€ 105,-	€ 126,-	€ 147,-	€ 210,-
<b>Agenda</b>	€ 84,-	€ 105,-	€ 126,-	€ 147,-	€ 210,-
<b>Accorderen</b>	€ 84,-	€ 105,-	€ 126,-	€ 147,-	€ 210,-
<b>Verlof</b>	€ 105,-	€ 131,25	€ 157,50	€ 183,75	€ 262,50
<b>Planning</b>	€ 147,-	€ 183,75	€ 220,50	€ 257,25	€ 367,50
<b>Kilometer</b>	€ 84,-	€ 105,-	€ 126,-	€ 147,-	€ 210,-
<b>Export</b>	€ 105,-	€ 131,25	€ 157,50	€ 183,75	€ 262,50
<b>Facturatie</b>	€ 336,-	€ 420,-	€ 504,-	€ 588,-	€ 840,-

## Mijnuren standalone server

Gebruikers	Jaarlijkse licentiekosten		
	Basis	Expert	Enterprise
1 t/m 25	€ 875,-	€ 1.750,-	€ 3.575,-
26 t/m 50	€ 1.225,-	€ 2.100,-	€ 3.925,-
51 t/m 100	€ 1.531,25	€ 2.406,25	€ 4.231,25
101 t/m 250	€ 2.712,50	€ 3.587,50	€ 5.412,50
251 t/m 500	€ 3.718,75	€ 4.593,75	€ 6.418,75
501 t/m 1000	€ 6.125,-	€ 7.000,-	€ 8.825,-
1001 t/m 2500	€ 15.662,50	€ 16.537,50	€ 18.362,50
+ 100	€ 1.190,35	€ 1.256,85	€ 1.395,55

## Mijnuren managed services

Van deze dienst zijn geen tarieven bekend.

## 4ever-uren.nl

### Functionaliteit

De drie hoofdfuncties van dit pakket zijn:

- Invullen en bijwerken van de urenlijst;
- Rapportages genereren;
- Administratief beheer van het systeem.

#### Invullen en bijwerken urenlijst

Met 4ever-uren is het mogelijk om tot op een kwartier nauwkeurig registreren hoeveel tijd men op welke dagen besteed heeft aan een bepaalde activiteit in een bepaald project.

Binnen dit systeem zit geen uitgebreide verlof module zoals we gezien hebben bij mijnuren.nl. Wanneer verlof geregistreerd moet worden dan moet gekozen worden voor de projectactiviteit “Verlof – Zomervakantie”.

#### Rapportages genereren

Deze functionaliteit biedt een aantal mogelijkheden waarmee de gebruiker projectrapportages en medewerkerrapportages kan genereren. De voorwaarde hiervoor is wel dat een gebruiker de zogeheten ‘superuser’ rechten moet hebben.

Per rapportage kunnen, per op te geven periode, 2 soorten rapportages gegenereerd worden. Deze zullen hieronder besproken worden.

#### Projectrapportage

De twee rapportages die hiervan gegenereerd kunnen worden zijn:

- Overzicht met activiteiten en hoeveelheid geboekte uren;
- Overzicht met activiteiten en hoeveelheid geboekte uren uitgesplitst naar medewerker.

#### Medewerkerrapportage

De twee rapportages die hiervan gegenereerd kunnen worden zijn:

- Overzicht per project met de activiteiten en de geboekte uren van 1 bepaalde medewerker;
- Overzicht per project met de activiteiten en geboekte uren van alle medewerkers.

Naast alleen de geboekte uren geven deze rapportages ook het uurtarief en de kosten weer.

#### Administratief beheer van het systeem

Het is mogelijk om 1 of meer gebruikers rechten te geven om het systeem administratief te beheren. Dit betekent dat men alle tabellen kan bijwerken. Op deze manier is het mogelijk om o.a. nieuwe projecten, activiteiten of gebruikers in te voeren.

In het volgende hoofdstuk zullen deze tabellen aan bod komen.

## Tabellen

### Projecten

Bij projecten kunnen medewerkers hun uren per activiteit registreren. Deze activiteiten zijn weer aan een bepaald project gekoppeld. Per project kunnen de volgende zaken aangegeven worden:

- Projectcode;
- Projectnaam;
- Omschrijving;
- Begin- en einddatum;
- Wel of geen fixed price project;
- Maximaal te spenderen aantal uren;
- Bijbehorende activiteiten.

### Gebruikers

Met gebruikers wordt iedereen bedoeld die het systeem moet kunnen gebruiken. Het systeem kent de volgende 3 gebruikers:

- User;
- Superuser;
- Administrator;

Hieronder zal kort worden beschreven wat deze gebruikers voor rechten hebben binnen dit systeem.

#### User

Dit omvat de gebruikers die enkel en alleen uren kunnen registreren en bijwerken.

#### Superuser

Deze gebruikers kunnen hun uren registreren en bijwerken maar ook rapportages samenstellen.

#### Administrator

De administrator heeft alle rechten die de superuser ook heeft en kan daarnaast ook het volledige systeem inrichten.

Per gebruiker kunnen de volgende zaken worden aangegeven:

- Gebruikersnaam;
- Wachtwoord;
- Naam;
- E-mailadres;
- Telefoonnummer;
- Standaard uurtarief;
- Soort gebruiker (user, superuser, admin);
- Actief, of niet meer;
- Koppeling met gebruikersgroepen.

## Gebruikersgroepen

Een gebruikersgroep is eigenlijk niets anders als een afdeling binnen een bedrijf. Gebruikers kunnen in 1 of meer gebruikersgroepen worden ingedeeld. Per gebruikersgroep kunnen de volgende zaken aangegeven worden:

- Groepsnaam;
- Per projectactiviteit of de leden hun uren op die projectactiviteit mogen schrijven.

## Kosten en tarievenstructuur

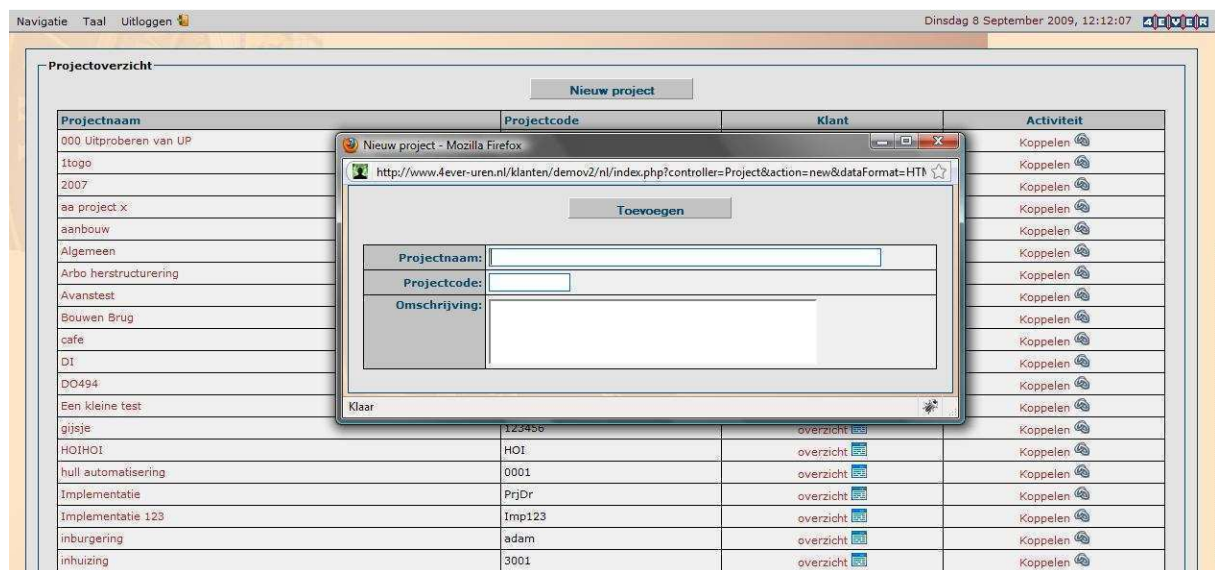
Het systeem werkt met tarieven (ex. BTW) per uur. De berekening van de werkelijke kosten geschiedt op basis van de geboekte uren of gedeelten daarvan (minimaal per kwartier).

Er zijn meerdere niveaus waarop uurtarieven kunnen worden vastgelegd. Deze niveaus zijn:

- Per projectactiviteit (mits als factuurabel aangemerkt);
- Per gebruiker voor een bepaalde projectactiviteit;
- Per gebruiker (standaardtarief voor deze gebruiker).

## Usability

Wanneer een blik op de website van 4ever-uren wordt geworpen dan heeft het meer weg van een willekeurige hobby website als van een website waar een product mee verkocht moet worden. Ook de demo die beschikbaar is op de website maakt op ons weinig indruk (zie Figuur 2).



Figuur 2 - Aanmaken van een project

## Create a clear visual hierarchy

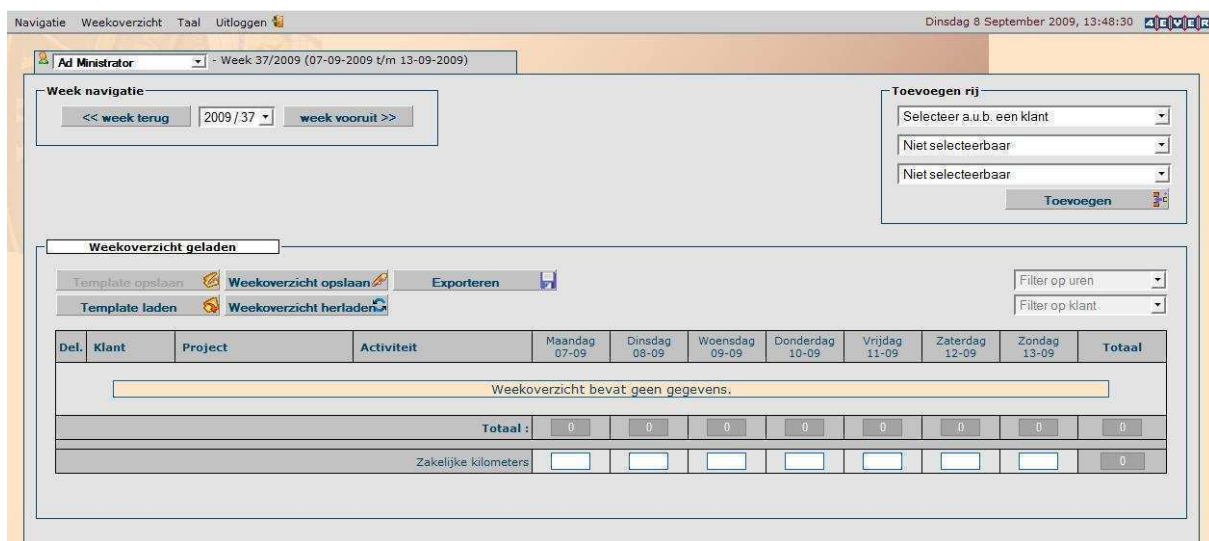
Van een duidelijke hiërarchie is eigenlijk geen sprake. Alle teksten zijn vrijwel even groot (klein) en zelfs het logo van het pakket is nauwelijks groter. Volgens de regels van Krug is alles wat op deze pagina staat even (on)belangrijk.

## Take advantage of conventions

Er wordt op de website gebruik gemaakt van aannames. Op de pagina getoond op Figuur 2 zijn deze te zien achter 'overzicht' en 'koppelen'. De afbeeldingen maken eigenlijk redelijk duidelijk wat ermee



wordt bedoeld. Ook op andere pagina's gebeurt dit door bijv. een floppy te laten zien achter 'exporteren' (zie Figuur 3), hoewel wij deze persoonlijk eerder zouden gebruiken voor opslaan.



Figuur 3 - Het dashboard

### Break pages up into clearly defined areas

Deze regel zit er redelijk in verwerkt zoals te zien is op Figuur 2 en 3. Bovenaan de pagina's bevindt zich een menu en daaronder staat een groot grijs vlak met inhoud. Op Figuur 2 is dat een overzicht met projecten, op Figuur 3 bevat dit vlak nog een aantal onderdelen welke duidelijk aangegeven worden.

### Make it obvious what's clickable

Ook aan deze regel voldoet de applicatie. Over het algemeen zijn het voornamelijk knoppen waarop (vanzelfsprekend) gedrukt kan worden, maar zoals op Figuur 2 te zien is krijgt klikbare tekst een andere kleur dan normale tekst.

Het hoofdmenu bovenaan de pagina doet dit ook goed. Wanneer de muis over het menu komt, verschijnt er een rand om duidelijk te maken dat erop geklikt kan worden. In het uitklapmenu worden de items blauw met witte tekst waardoor ook duidelijk gemaakt wordt dat het klikbaar is.

### Minimize noise

Wat ons betreft voldoet de applicatie hier niet aan. De achtergrond op alle pagina's zorgt voor een hoop onrust op de pagina terwijl dit niet nodig is. Door een witte achtergrond te gebruiken zou het stukken rustiger over kunnen komen.

### Tarieven

Het gebruik van 4ever-uren wordt maandelijks berekend. Op de website staan de maandtarieven van 2006. Doordat deze nog steeds op de website staan nemen wij aan dat deze tarieven nog steeds geldig zijn.

Aantal gebruikers	Prijs (per maand)
1 – 50	€ 49,-
51 – 100	€ 99,-
Per 10 gebruikers extra	€ 5,-

Naast deze maandelijkse tarieven rekent 4ever-uren jaarlijks € 65,- hosting kosten. Daarentegen worden geen set-up kosten in rekening gebracht.

## Clockwise.info

Net als mijnuren.nl biedt ClockWise meer als alleen urenregistratie. Naast alleen urenregistratie bieden ze ook urenmanagement, rapportages, factureren, agenda en gebruikersprofielen.

## Functionaliteiten

### Urenregistratie

De volgende functionaliteit is beschikbaar bij de urenregistratie:

- Uren schrijven en inleveren per week, maand of zelf in te stellen periode;
- Meerdere uren schrijven los van elkaar per projectonderdeel per dag;
- Uren importeren uit agenda;
- Opmerkingen plaatsen bij geschreven uren;
- Schrijven van zakelijke kilometers, woonwerk kilometers, reiskosten en onkosten;
- Tijd voor tijd systeem;
- Week inleveren alleen als het gewenste aantal uren is geschreven;
- Mogelijkheid informatie over het project te bekijken:
  - Wanneer is de deadline;
  - Hoeveel uren zijn er begroot voor de gebruiker;
  - Hoeveel uren zijn er reeds geschreven.

### Urenmanagement

- Ingeleverde uren kunnen per week of per maand worden ingeleverd en goedgekeurd;
- Mogelijkheid om feedback te geven aan gebruiker;
- Maandoverzichten per medewerker;
- Accorderen door linemanager, projectmanager of klant;
- Overzichten afdrukken in PDF;
- Jaaroverzicht;
- Onderscheid tussen project en interne uren.

### Rapportages

- Rapportages op klant-, project-, onderdeel-, afdelings-, en medewerkerniveau en over iedereen;
- Wat is de verhouding tussen begrote uren en geschreven uren;
- Wie heeft wanneer vrij genomen;
- Wie heeft volgens de planning nog tijd om op een nieuw project gezet te worden;
- Wat is de onderhanden werkpositie;
- Zoeken in de geschreven uren op basis van project, onderdeel, medewerker of periode;
- Uitgebreid zoeken op goedgekeurde uren, gefactureerde uren of factuurnummer;
- Sommatie van uren, onkosten, uren \* tarief en meer per afdeling, medewerker, klant/project en periode.

### Factureren

- Facturen maken op basis van vast bedrag, geplande uren, geschreven uren of goedgekeurde uren;

- Uurtarieven mogelijk per project, per medewerker en voor iedere medewerker voor ieder project verschillend;
- Specificatie van de uren meesturen;
- Uren afboeken die niet gefactureerd gaan worden;
- Creditnota's maken en gefactureerde uren vrijgeven;
- Herinneringen maken en opslaan los van factuur;
- Debiteuren administratie bijhouden;
- Weergeven in Word, .html en .rtf formaat;
- Gefactureerde uren worden op gefactureerd gezet;
- Doorberekenen reiskosten, reiskilometers \* tarief en onkosten;
- Toevoegen extra onderdelen in de factuur;
- BTW % instellen.

### Agenda

- Afspraken met medewerkers kunnen maken;
- Agenda's naast elkaar leggen;
- Mailen uitgenodigde medewerkers;
- Exporteren via iCal of vCal;
- Agenda items in het urenscherm direct als gemaakte uren importeren;
- Week, maand en jaaroverzichten van de afspraken.

### Gebruikersprofielen

- Gebruikersgroepen aanmaken die verschillende rollen hebben binnen ClockWise. Per groep kan het volgende ingesteld worden:
  - Welke onderdelen zijn beschikbaar;
  - Wat is het rapportage niveau. Rapportage over 1 persoon (gebruiker), over de afdeling (afdelingrapport) of over iedereen (beheerder);
  - Wat is de kleurinstelling;
  - Mogen uren geschreven worden na de deadline en boven de begrote uren;
  - Mogen meer uren geschreven worden dan de contract uren per week.

### Usability

Op de website is een demo beschikbaar van urenregistratie (zie Figuur 4).

Figuur 4 – Aanmaken van een project

### Create a clear visual hierarchy

Volgens deze regel van Krug zijn grote teksten het belangrijkste en kleine teksten onbelangrijk.

Daarvan uitgaand, is de titel van de pagina als belangrijkste wordt gezien, en vervolgens de titels die onderdelen van de lijst aangeven en daarna de link na de titel. Daarop volgend komt het menu, die andere kleuren gebruikt en andere tekst dikte heeft. De rest, mappen lijst en de inputveld titels, van de tekst wordt als onbelangrijk beschouwd

### Take advantage of conventions

Hier is de kalender icoon weer te zien, en ook hier verschijnt er een pop-up kalender als er op gedrukt wordt (figuur 2)

Figuur 5 - kalender

### Break up pages into clearly defined areas

De pagina is verdeeld in 3 gedeeltes: bovenaan de pagina is het hoofdmenu, aan de linkzijde een lijst met mappen en taken en vervolgens in de hoofdpagina met daarin de naam van de pagina daaronder.

### Make it obvious what's clickable

Het menu zien er uit als tabs en zien er klikbaar uit, er is duidelijk verschil met de actieve tabblad en de inactieve tabblad. de linkjes in de mappenlijst op de linkerhelft van de pagina worden pas duidelijk wanneer de muis er over heen zweeft, en omdat ze veel op de rest van de tekst lijkt, lijkt het niet snel klikbaar.

### Minimize noise

Er is geen ruis aanwezig op deze pagina.

### Tarieven

Er zijn twee verschillende manieren om gebruik te maken van dit pakket. De eerste is door middel van een hosting abonnement bij ClockWise. De tweede is door middel van de aanschaf van ClockWise en installatie op een eigen (web)server.

#### Hosting door ClockWise

Aantal gebruikers	Prijs (per maand)
1	€ 15,-
2 – 5	€ 40,-
6 – 10	€ 60,-
11 – 15	€ 75,-
16 – 20	€ 86,-
25	€ 97,-
>25	Contact opnemen

#### Aanschaf van ClockWise

Ook hier is de prijs afhankelijk van het aantal gebruikers. Hier hoeft echter pas contact opgenomen te worden wanneer het om meer als 50 gebruikers gaat.

Aantal gebruikers	Prijs
1 – 5	€ 550,-
10	€ 925,-
15	€ 1175,-
20	€ 1425,-
25	€ 1675,-
50	€ 2675,-
>50	Contact opnemen

## NuTijd

Op de website van nuTijd staat qua functionaliteit minder beschreven als de voorgaande websites. Wel worden de onderdelen van het systeem genoemd.

- Urenregistratie;
- Planning;
- Facturatie;
- Rapportage;
- Klanten;
- Projecten;
- Medewerkers;
- Afdelingen;
- Accorderen;
- Herinneringen;
- Tarieven;
- Export

## Functionaliteit

Zoals eerder gezegd staat er vrij weinig informatie over de functionaliteit van het pakket op de website. Hieronder worden de functionaliteiten opgesomd die wel op de website staan:

- Registratie op basis van klant, project, tarief, activiteit en uren;
- Controle en accordering door HR/manager;
- Automatische herinnering per e-mail bij ontbrekende uren;
- Planningsoverzicht voor interne planning en afspraken;
- Klanten, contactpersonen en projecten beheren;
- Rapportage per dag, week, maand en jaar;
- Selectie op klant, project, afdeling en medewerker;
- Rapporten bekijken in HTML, Word of Excel formaat;
- Weekstaten afdrukken voor accordering;
- Flexibel factureren op basis van uren en extra kosten;
- Export naar externe pakketten;
- Rollen/rechten systeem (medewerker, manager, beheerder);
- Maatwerk uitbreidingen mogelijk.

## Usability

Ook op de website van nuTijd is een demo beschikbaar welke wij zullen onderwerpen aan de usability test.

INFO

URENREGISTRATIE

PLANNING

FACTURATIE

RAPPORTAGE

BEHEER

AFMELDEN

KLANTEN

MEDEWERKERS

AFDELINGEN

TARIEVEN

BTW

ACTIVITEITEN

HERINNERINGEN

EXPORT

WACHTWOORD

INSTELLINGEN

Klanten

Nieuw record +

<< < Pagina 1 / 1 > >>

Naam ↑ ↓	Klantnr ↑ ↓	Bezoekplaats ↑ ↓	Contactpersonen ↑ ↓	Projecten ↑ ↓	Status ↑ ↓	
abcde	50064		0 (Toon details)	1 (Toon details)	Actief	
Blauw VOF	50049	Utrecht	1 (Toon details)	1 (Toon details)	Actief	
Gazonmaaiers VOF	50050		0 (Toon details)	0 (Toon details)	Actief	
Groninger Techniek	50048	Groningen	0 (Toon details)	2 (Toon details)	Actief	
H.J. van de Smit B.V.	50047	Amsterdam	0 (Toon details)	0 (Toon details)	Actief	
Intern	00001		0 (Toon details)	0 (Toon details)	Actief	
Janssen	50061	Janssentown	0 (Toon details)	0 (Toon details)	Actief	
leve de zzp-er	50057	amsterdam	0 (Toon details)	2 (Toon details)	Actief	
test man	50062		0 (Toon details)	0 (Toon details)	Actief	
Tuinder	50056		0 (Toon details)	1 (Toon details)	Actief	
Van der Vlucht	50055	Maastricht	0 (Toon details)	2 (Toon details)	Actief	
XS	50060	Lelystad	0 (Toon details)	0 (Toon details)	Actief	
zomaar een clubje	50063		0 (Toon details)	0 (Toon details)	Actief	

Medewerker: Erik van der Ploeg  
Laatste login: 08-09-2009 14:18

Figuur 6 - Klanten beheren


## Create a clear visual hierarchy

Volgens de regels van Krug is op deze pagina het bovenste menu het belangrijkste, gevolgd door het submenu. Verder ontbreekt eigenlijk een logo op de pagina om nog eens te benadrukken op welke website je bezig bent.

## Take advantage of conventions

Deze regel wordt toegepast in de knoppen die zichtbaar zijn op Figuur 6. Een + achter 'Nieuw record' en een X achter een record zegt eigenlijk voldoende. Ook op andere pagina's worden deze aannames toegepast.

Het mooiste voorbeeld is misschien wel te vinden op de pagina van de facturen. Hier staan onder de koppen weergave en export de logo's van Internet Explorer, Word, Excel en Adobe Reader (zie Figuur 7).

Factuur ↑ ↓	Status ↑ ↓	Factuurdatum ↑ ↓	Klant ↑ ↓	Weergave	Export	
50813	Openstaand	14-08-2009	leve de zzp-er	  		
50812	Openstaand	04-08-2009	Blauw VOF	  		

Figuur 7 - Gebruik van aannames

## Break up pages into clearly defined areas

Elke pagina in de applicatie wordt op dezelfde wijze getoond als te zien is op Figuur 6. Bovenaan het hoofdmenu, met daaronder het submenu (wanneer deze bestaat) en daaronder de data die erbij hoort.

Eigenlijk heel simpel, maar wel duidelijk.



### Make it obvious what's clickable

Dit punt is voor verbetering vatbaar wat ons betreft. Het hoofdmenu is vrij duidelijk doordat deze items wat weg hebben van knoppen. Het submenu daarentegen bestaat uit wat afbeeldingen met tekst erop, maar nergens wijst uit dat erop te klikken valt. Zeker voor onervaren gebruikers kan dit voor verwarring zorgen.

Wat ze overigens wel weer goed doen is klikbare tekst blauw gekleurd maken, zoals de klantnamen van Figuur 4. Dit zou echter nog verbeterd kunnen worden door de tekst te onderstrepen.

### Minimize noise

Ruis is eigenlijk niet aanwezig in de applicatie. Het is allemaal strak met prettige kleuren en veel wit. Hier scoren ze wat ons betreft zeer goed.

### Tarieven

Ook bij nuTijd wordt het pakket aangeboden in abonnementsvorm en gaan de prijzen per aantal medewerkers.

Pakket	Starter	Basic	Small	Medium	Large	Pro
Gebruikers	1	5	10	25	50	51+
Set-up kosten	€ 0,-	€ 0,-	€ 0,-	€ 0,-	€ 0,-	Offerte
Facturering per kwartaal	n.v.t.	€ 48,75 p/m	€ 71,50 p/m	€ 98,75 p/m	€ 147,50 p/m	Offerte
Facturering per jaar	€ 8,19 p/m	€ 44,85 p/m	€ 65,78 p/m	€ 90,85 p/m	€ 135,70 p/m	Offerte

## **Comsult**

Over het urenregistratie pakket van Comsult is helaas weinig informatie beschikbaar. Een aantal functionaliteiten die genoemd worden zijn:

- Per project uren en tarief registreren;
- Periodieke export naar XML of Excel;
- Declarabiliteit per medewerker;
- Budgetuitputting;
- Burn rate per project / medewerker;
- Projectomzetten.

## **Usability**

Er is helaas geen online demo beschikbaar waarmee we zelf een kijkje kunnen nemen in het programma. Om een demo te krijgen moet contact opgenomen met Comsult.

## **Tarieven**

Deze staan niet op de website aangegeven.

## Sumeris

Ook over het urenregistratie pakket van Sumeris is weinig informatie beschikbaar op de website. Zij noemen zelfs helemaal geen functionaliteit van het pakket.

## Usability

Er is helaas ook bij Sumeris geen online demo beschikbaar. Evenals bij Comsult dient contact opgenomen te worden om een demo aan te vragen.

## Tarieven

Tarieven staan wel op de website van Sumeris en zijn als volgt:

Pakket	Prijs (per maand)
Basis financieel	€ 20,-
Urenbeheer	€ 5,-

Bij Sumeris is het dus mogelijk om voor 25 euro per maand een urenregistratie pakket te hebben, ongeacht het aantal gebruikers. Wel wordt er eenmalig € 75,- berekend voor een E-token.

## Conclusie

Het eerste wat we kunnen concluderen na dit onderzoek is dat de pakketten die wij hebben onderzocht over het algemeen meer bieden als alleen urenregistratie. Planning en facturering is een veel terugkomend iets dus in dat opzicht weten we dat we nog een flinke slag te maken hebben. Echter, voordat we hiermee aan de slag gaan willen we eerst een goede basis hebben liggen met de urenregistratie.

Vrijwel alle functionaliteit die wij in gedachten hebben zijn al gerealiseerd door één of meerdere systemen. Zo dachten wij o.a. aan een koppeling tussen HourPower en andere pakketten (o.a. AccountView) en e-mail notificaties bij het niet invoeren van uren. Dus in hoeverre kunnen wij ons onderscheiden van al deze pakketten?

Als eerste antwoord op deze vraag verwijzen wij naar de usability tests die wij hebben uitgevoerd op de beschikbare demo systemen. Niet één systeem bevatte echt alle onderdelen voldoende. Dit lijkt misschien een nietszeggend iets, echter een goed overzichtelijke website is enorm belangrijk. Kijk bijvoorbeeld naar Google. Het was niet de eerste zoekmachine, maar inmiddels wel wereldwijd de meest gebruikte. Het succes hiervan begint door de enorm eenvoudige vormgeving, in tegenstelling tot andere zoekmachines waarbij de pagina's vol staan met advertenties en andere (overbodige) toepassingen.

Het is ons ook opgevallen dat geen van de pakketten een koppeling biedt met Outlook. Een agenda kunnen beheren is makkelijk, maar dit zou nog makkelijker kunnen door deze met Outlook te koppelen. Het is echter nog de vraag of het momenteel haalbaar is voor ons om een dergelijke koppeling te realiseren. Hier hebben wij onderzoek naar gedaan en voor het resultaat verwijs ik u graag naar dit onderzoek (Onderzoek: Outlook Add-on).

Daarnaast is ons opgevallen dat al deze pakketten wel online te bereiken zijn, maar niet één biedt de dienst ook aan voor de mobiele telefoon. Iemand die ooit op internet is geweest met de mobiele telefoon weet dat het niet te doen is wanneer een website hier niet voor gemaakt is.

Uit onderzoek van Forrester [1] is gebleken dat een vijfde van de Nederlanders gebruik maakt van internet op de mobiele telefoon. Dit komt neer op een aantal van ongeveer 2,25 miljoen mensen. Uit een ander onderzoek van Forrester [2] blijkt tevens dat verwacht wordt dat het gebruik van mobiel internet de komende jaren nog zal stijgen in West-Europa, van 13% in 2008 naar 39% in 2014.

Wij verwachten dan ook dat hier kansen liggen om ons te onderscheiden van de andere pakketten, door een lichtere variant van de applicatie beschikbaar te maken voor mobiel internet.

Het volgende punt waarmee wij ons denken te kunnen onderscheiden heeft wederom met de mobiele telefoon te maken. Vrijwel alle pakketten bieden een e-mail notificatie systeem aan om personeel te helpen herinneren dat er nog uren ingevuld moeten worden. Het nadeel van e-mail is echter dat het nog wel eens gebeurt dat het niet aankomt of (te) laat gelezen wordt.

Dit willen wij voorkomen door de mogelijkheid aan te bieden om gebruik te maken van een sms dienst. Deze dienst hoeft niet alleen te dienen als herinnering voor het invullen van uren, maar zou ook gebruikt kunnen worden om meerdere mensen tegelijk belangrijke mededelingen te sturen.

Het voordeel hiervan ten opzichte van e-mail is het feit dat men over het algemeen altijd de mobiele telefoon bij zich heeft en deze berichten ook kan lezen wanneer er geen computer in de buurt is.

Een extra functionaliteit die wij willen bieden is om vanuit HourPower de mogelijkheid te bieden mensen te bellen door middel van Skype.

## Ons pakket

Op basis van de onderzochte pakketten en de onderdelen waarmee wij ons willen onderscheiden is het inmiddels mogelijk om een beeld te schetsen van hoe ons pakket er qua functionaliteit uit zal komen te zien. Met deze lijst zullen wij vervolgens contact opnemen met bedrijven om te kijken wat er nog ontbreekt, of welke functionaliteit overbodig is en hoe belangrijk de functionaliteit is.

- Registreren van medewerkers, afdelingen, klanten, projecten en activiteiten;
- Registratie van gewerkte uren;
- Agenda bijhouden;
- Taken in agenda direct als uren verwerken;
- Koppeling tussen agenda en Outlook;
- Rollen maken en toekennen aan medewerkers (bijv. beheerder, gebruiker);
- Tarief van medewerkers instellen;
- Importeer mogelijkheid voor medewerkers, klanten, projecten en activiteiten;
- Exporteren van medewerkers, klanten, projecten, activiteiten en uren (xml, csv);
- Kopieer mogelijkheid van projecten en activiteiten;
- Tarief van activiteit instellen;
- Vast bedrag voor een project instellen;
- Accorderen door project- of afdelingmanager;
- Online mededelingen voor personeel;
- Meertalige interface (EN/NL);
- Tijd-voor-tijd regeling;
- Opmerkingen plaatsen bij geschreven uren;
- Mogelijkheid tot feedback aan medewerker;
- Onderscheid tussen projecten en interne uren;
- Uren inleveren per week, maand of zelf in te stellen periode;
- Overzicht geboekte uren op project/activiteit en per medewerker;
- Overzicht geboekte uren vs. geplande uren, per project of medewerker;
- Week, maand, kwartaal en jaaroverzichten per medewerker;
- Overzichten afdrukken in PDF;
- E-mail notificatie bij niet geregistreerde uren;
- SMS notificatie bij niet geregistreerde uren en mededelingen;
- Toegang vanaf mobiele telefoon;
- Skype implementatie;
- Mogelijkheid tot maatwerk uitbreiding.

## Bronvermelding

[1] Wokke, A (2009), *Forrester: een vijfde Nederlanders gebruikt internet op telefoon* - <http://tweakers.net/nieuws/61431/forrester-een-vijfde-nederlanders-gebruikt-internet-op-telefoon.html>

[2] Husson, T (2009), *Western European Mobile Forecast, 2009 to 2014* - <http://www.forrester.com/Research/Document/Excerpt/0,7211,53717,00.html>

Bijlage B2

# MoSCoW tabel

Jeffrey Veldkamp  
Daniël van Zoelen

## MoSCoW Tabel - HourPower

	Must have	Should have	Could have	Would like to have
Registreren van medewerkers, afdelingen, klanten, projecten en activiteiten				
Rollen maken en toekennen aan medewerkers (bijv. beheerder, gebruiker)				
Tarief van medewerkers instellen				
Importeer mogelijkheid voor medewerkers, klanten, projecten en activiteiten				
Exporteren van medewerkers, klanten, projecten, activiteiten en uren (xml, csv)				
Kopieer mogelijkheid van projecten en activiteiten				
Tarief van activiteit instellen				
Vast bedrag voor een project instellen				
Accorderen door project- of afdelingmanager				
Online mededelingen voor personeel				
Meertalige interface (EN/NL)				
Tijd-voor-tijd regeling				
Opmerkingen plaatsen bij geschreven uren				
Mogelijkheid tot feedback aan medewerker				
Onderscheid tussen projecten en interne uren				
Uren inleveren per week, maand of zelf in te stellen periode				
Overzicht geboekte uren op project/activiteit en per medewerker				
Overzicht geboekte uren vs. geplande uren, per project of medewerker				
Week, maand, kwartaal en jaaroverzichten per medewerker				
Overzichten afdrukken in PDF				
E-mail notificatie bij niet geregistreerde uren				
SMS notificatie bij niet geregistreerde uren en mededelingen				
Toegang vanaf de mobiele				



telefoon
Skype implementatie
Mogelijkheid tot maatwerk uitbreiding
Agenda bijhouden
Taken in agenda direct als uren verwerken
Koppeling tussen agenda en Outlook

## Ontbrekende functionaliteit

Mocht u nog functionaliteiten missen in dit pakket, voeg deze dan toe in onderstaande lijst:

-

**Bijlage B3**

# Onderzoek: Software architectuur

Welke te gebruiken voor HourPower

Jeffrey Veldkamp  
Daniël van Zoelen

## Inhoudsopgave

Inleiding .....	3
Model-View-Controller (MVC) .....	4
Model .....	4
Passive Model.....	4
Active Model .....	5
View .....	5
Controller.....	5
Front Controller .....	5
Voordelen .....	6
Nadelen .....	6
Model-View-Presenter (MVP) .....	7
Model .....	7
View .....	7
View interface .....	7
Presenter .....	7
Voordelen .....	7
Nadelen .....	8
Presentation-Abstraction-Control (PAC) .....	9
Presentation .....	10
Abstraction .....	10
Control.....	10
Voordelen .....	10
Nadelen .....	10
Document-View .....	11
Document .....	11
View .....	11
Voordelen .....	11
Nadelen .....	11
Conclusie .....	12
Bronvermelding .....	13
Literatuurlijst .....	14

## Inleiding

Om software te ontwikkelen moet rekening gehouden worden met de juiste software architectuur. Wanneer dit niet wordt gedaan kan dit ernstige gevolgen hebben op het eindresultaat. Zo is het mogelijk dat het project volledig stuk loopt of enorm uitloopt wanneer zomaar begonnen wordt. Een andere mogelijkheid is dat het uiteindelijk enorm lastig is, of zelfs onmogelijk, wanneer (delen van) de software uitgebreid of hergebruikt moet worden.

Om zelf niet tegen deze problemen aan te lopen hebben we besloten om onderzoek te doen naar de beste software architectuur voor onze applicatie. Hierbij moet rekening gehouden worden met herbruikbaarheid, uitbreidbaarheid en onderhoudbaarheid.

In een whitepaper van cibit.nl [1] wordt een drietal standaardarchitecturen genoemd die gericht zijn op het opzetten van interactieve systemen. Deze architecturen zijn:

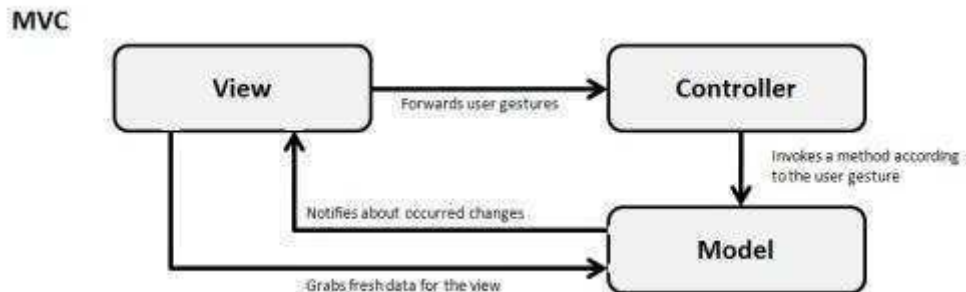
- Model-View-Controller;
- Presentation-Abstraction-Control;
- Document-View.

Na een simpele zoekopdracht in Google naar het MVC treffen wij nog een andere architectuur, namelijk het MVP. Ook deze architectuur zal door ons onderzocht worden.

In de rest van dit document zullen we alle architecturen bespreken om vervolgens tot een conclusie te komen welke architectuur het best bij onze software past.

## Model-View-Controller (MVC)

Het MVC is een veelgebruikte architectuur voor applicaties die gebruik maken van gebruikersinteractie. Het voornaamste idee is het scheiden van de user interface van een applicatie van de domein logica. Het verdeelt functionaliteit over objecten welke betrokken zijn bij het onderhouden en presenteren van de data.



Figuur 1 - Het MVC, <http://dotnetslackers.com/articles/designpatterns/Evergreen-but-still-topical-MVC-vs-MVP.aspx>

Het MVC bevat drie componenten met verschillende verantwoordelijkheden:

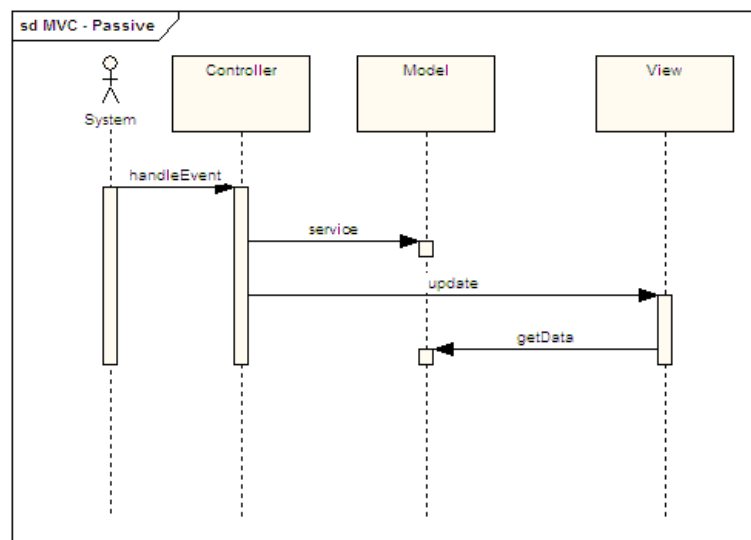
### Model

Het model bevat de data van de applicatie en de business logica welke bepaalt hoe de data gewijzigd en opgevraagd wordt. De opslag van data wordt gedaan met behulp van een database. Het ophalen en wegschrijven van data gaat via een datalaag (Data Access Object). Deze datalaag is niet per se een onderdeel van het MVC.

### Passive Model

Behalve bovenstaand model bestaat er nog een andere variant van het model, het Passive Model. In deze variant verdwijnt de relatie tussen model en view. Het updaten van de view gebeurt dan via de controller.

Het Passive Model wordt veel gebruikt voor web MVC. Doordat de view van webpagina's altijd volledig opnieuw geladen wordt is het niet noodzakelijk om een active model te gebruiken.

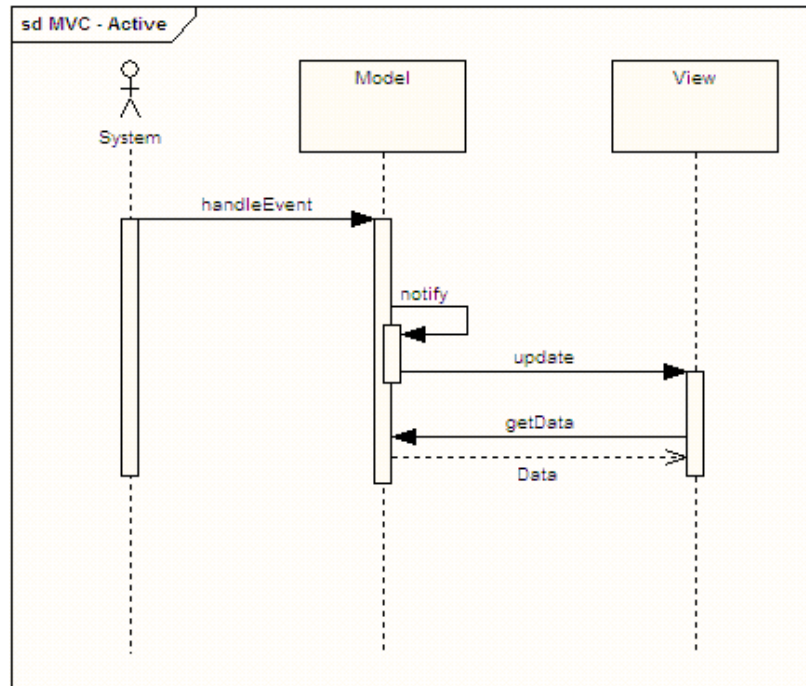


Figuur 2 - Werking van het Passive Model, <http://communities.progress.com/pcom/docs/DOC-14628%3B>

## Active Model

Bij het Active Model wordt gebruik gemaakt van een Observer. Dit model wordt gebruikt wanneer het model wijzigt zonder inbreng van de controller.

Een voorbeeld hiervan is een thermometer. In dit geval wijzigt de temperatuur (de data) en de view moet deze verandering laten zien.



Figuur 3 - Werking van het Active Model, <http://communities.progress.com/pcom/docs/DOC-14628%3B>

## View

De view wordt gebruikt om de data aan de gebruiker te tonen. De view doet geen verwerkingen van de getoonde data.

## Controller

De controller verwerkt en reageert op handelingen van gebruikers, zoals het versturen van een formulier. Deze handelingen worden opgevangen en vervolgens wordt het model aangepast waarna de gewijzigde data door het model teruggegeven wordt aan de view.

In het MVC is de controller **geen** mediator tussen de view en het model. Zowel de controller als de view hebben gelijke mogelijkheden om het model aan te spreken. De controller zal ook geen data van het model naar de view kopiëren. Wel mag de controller waarden in het model plaatsen en de view vertellen dat het model is aangepast.

## Front Controller

De front controller is een patroon wat bij webapplicaties gebruikt wordt. Deze vangt in eerste instantie het HTTP request op, analyseert de data en voert bewerkingen uit alvorens de benodigde controller aan te roepen.

Een eenvoudig voorbeeld hiervan kan zijn:

```
<?php
switch($_GET['page']) {
    case 'news':
        include('pages/news.php');
        break;
    default:
        include('pages/home.php');
        break;
}
?>
```

Het voordeel van het gebruik van een front controller is dat bepaalde functies niet elke keer opnieuw geschreven hoeven te worden, bijvoorbeeld het controleren van de geldigheid van een sessie.

## Voordelen

De voordelen van het gebruik van het MVC zijn:

- We hebben reeds ervaring met het gebruik van het MVC;
- Code is makkelijk herbruikbaar;
- Makkelijk uitbreidbaar;
- Domein onafhankelijk;
- Gescheiden verantwoordelijkheden (model, view, controller) waardoor de code beter te begrijpen is;
- Bevordert multi-team development. Programmeurs kunnen zich bezig houden met de business logica terwijl de ontwerpers zich bezig kunnen houden met de views zonder de code te beschadigen;
- Veel informatie beschikbaar over MVC in PHP;
- Vrijwel elk PHP framework maakt gebruik van het MVC model.

## Nadelen

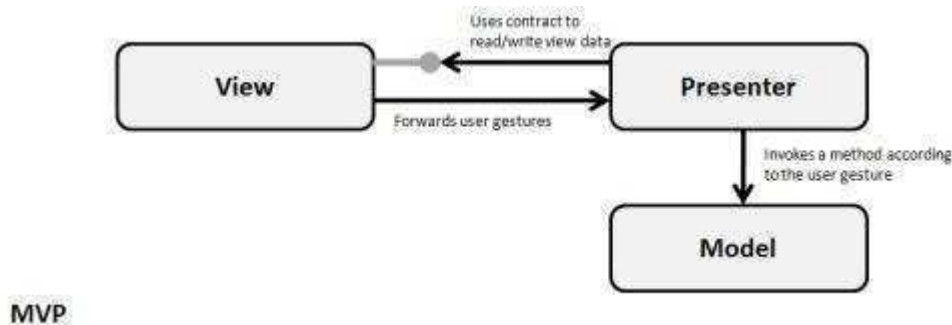
De nadelen van het gebruik van het MVC model zijn:

- Verhoogde complexiteit. Extra classes/code door de gescheiden verantwoordelijkheden;

## Model-View-Presenter (MVP)

Het MVP model is gebaseerd op het concept van het MVC model. Het MVP model scheidt de verantwoordelijkheden echter over 4 componenten:

- Model;
- View;
- View interface;
- Presenter.



Figuur 4 - Het MVP, <http://dotnetslackers.com/articles/designpatterns/Evergreen-but-still-topical-MVC-vs-MVP.aspx>

### Model

Het model bevat, net zoals het model bij MVC de data van de applicatie en voorziet methodes om de data beschikbaar te maken. Daarnaast bevat het model ook de business logica.

In tegenstelling tot het MVC is het niet het model die de view update maar wordt dit door de presenter gedaan. Over het algemeen wordt dit als voordeel beschouwt omdat in deze situatie de view **geen** kennis heeft van het model.

### View

In tegenstelling tot het MVC model is de view niet alleen verantwoordelijk voor het tonen van data aan de gebruiker. Hiernaast vangt de view in eerste instantie de gebruikershandelingen op waarna deze doorgegeven worden aan de presenter.

### View interface

De view interface wordt gebruikt om de view aan de presenter te koppelen, waarna deze met elkaar kunnen communiceren.

### Presenter

Waar de view verantwoordelijk is voor het tonen van de data daar is de presenter verantwoordelijk om het model aan te passen als reactie op de input van de gebruiker. De input van de gebruiker wordt opgevangen door de view en doorgespeeld aan de presenter, welke op zijn beurt het model bewerkt waar nodig.

### Voordelen

Het MVP model heeft, doordat het vrijwel identiek is, dezelfde voordelen als het MVC model. Er is echter een punt waar MVP (iets) beter scoort dan MVC, namelijk het maken van unit tests. Met MVC



is dit al vrij eenvoudig maar met MVP is dit nog eenvoudiger omdat de interactie met de view via een interface verloopt.

### **Nadelen**

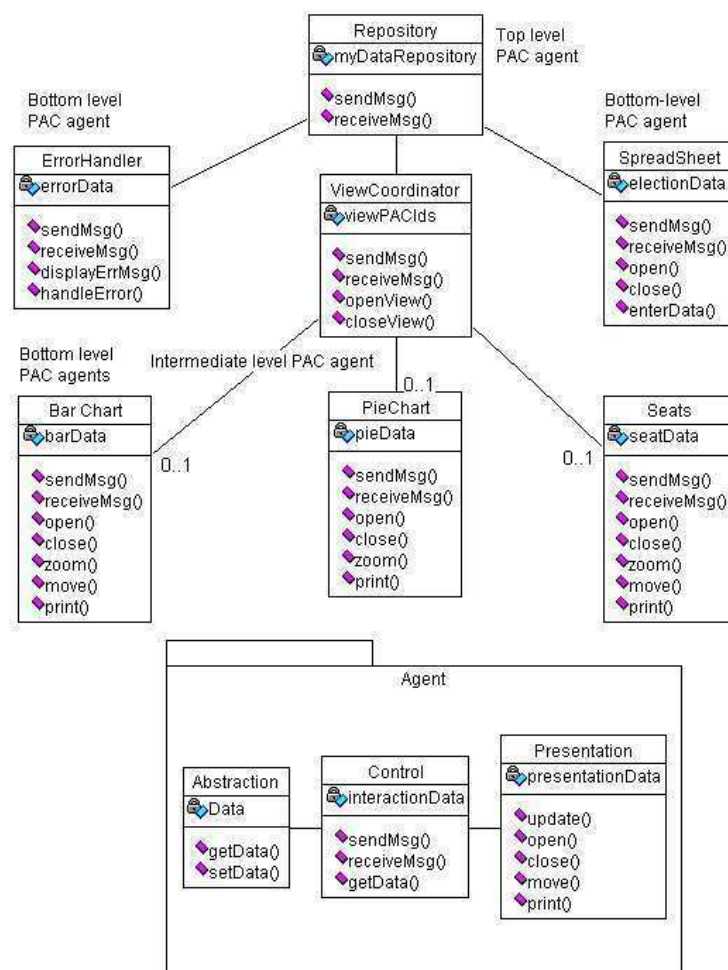
- Verhoogde complexiteit. Extra classes/code door de gescheiden verantwoordelijkheden;
- Geen ervaring onder de projectleden met dit model;

## Presentation-Abstraction-Control (PAC)

Het PAC is een verdere ontwikkeling van het MVC model. Het MVC is beperkt tot eenvoudige GUI's waar één of meerdere views gebruik maken van hetzelfde model. Wanneer het model bestaat uit substructuren welke ieder een eigen manier van interactie nodig hebben is een complexere architectuur nodig.

Het PAC model heeft, in tegenstelling tot het MVC, niet het model als belangrijkste onderdeel maar een hiërarchische structuur van PAC componenten. Elk PAC component bestaat uit de volgende onderdelen:

- Presentation;
- Abstraction;
- Control.



Figuur 5 - PAC architectuur, <http://www.vico.org/pages/PatronsDisseny/Pattern%20Presentation%20Abstra/>

In bovenstaande afbeelding is een voorbeeld van het PAC te zien. Elk onderdeel hiervan bestaat uit de drie PAC componenten. Zo heeft bijv. Seats een SeatsAbstraction, SeatsControl en SeatsPresentation. De SeatsControl communiceert met de ViewCoordinatorControl, welke op zijn beurt communiceert met de RepositoryControl.

## Presentation

Presentation is exact hetzelfde als de view in het MVC. Het toont data aan de gebruiker maar doet geen verwerkingen.

## Abstraction

Abstraction is vergelijkbaar met het model van het MVC, echter kan het zijn dat het slechts een onderdeel is van de volledige data structuur van de applicatie. Daarnaast speelt hij geen actieve rol in de notificatie van wijzigingen.

## Control

De control heeft ook overeenkomsten met de C van het MVC. De control zorgt voor de verwerking van gebruikershandelingen en update het model. Daarnaast is het de taak van de control om de presentation te updaten na een handeling.

Het verschil van deze control ten opzichte van de control van het MVC is het feit dat deze de gemaakte wijzigingen doorgeeft aan het ouder (parent) PAC component.

## Voordelen

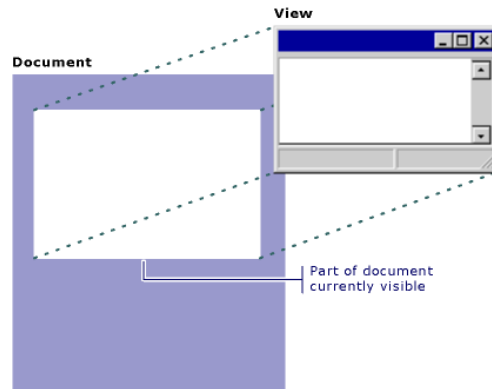
- Gescheiden code;
- Duidelijke architectuur voor (zeer) complexe applicaties;
- Eenvoudig uitbreidbare code;
- Code is herbruikbaar;

## Nadelen

- Nog complexere architectuur dan het MVC;

## Document-View

De Document-View architectuur is geïntroduceerd in MFC 2.0. In deze architectuur wordt de data van de applicatie gerepresenteerd als een document object. De views van deze data worden gerepresenteerd als view objecten.



Figuur 6 - Document-View, <http://msdn.microsoft.com/en-us/library/4x1xy43a%28VS.80%29.aspx>

## Document

Het document is exact zoals de naam zegt, het document. Het document kan bestaan uit meerdere views. In een Excel document zouden dit grafieken, tabellen etc. kunnen zijn.

Het enige doel wat het document heeft is het beheren van de applicatiedata. Dit betekent dat het document zich bezighoudt met het opslaan en verwerken van data en het updaten van de views.

## View

Zoals eerder gezegd zou een view binnen een programma als Excel bijvoorbeeld een grafiek kunnen zijn. De view is verantwoordelijk voor het tonen van de data en het verzorgen van gebruikersinteractie hiermee, waaronder selectie en bewerking.

## Voordelen

- Op een eenvoudige manier meerdere views tonen;
- Gescheiden code;
- Herbruikbaarheid en uitbreidbaarheid van code;
- Eenvoudig toe te passen middels het MFC framework.

## Nadelen

- De rol van de controller uit het MVC model is verdeeld over de document en de view;
- De architectuur is speciaal voor het MFC framework;
- De architectuur is voornamelijk bedoeld voor gebruik met C++.

## Conclusie

Het moge duidelijk zijn dat er voldoende architecturen aanwezig zijn om op een goede, gestructureerde manier een (web) applicatie te bouwen. Aan ons nu de opgave om de juiste architectuur voor onze applicatie te kiezen.

Op de laatste plaats is wat ons betreft de Document-View architectuur terecht gekomen. Het is een architectuur die wel code scheidt, maar tegelijkertijd wordt overal aangegeven dat het een architectuur is voor MFC en C++. Aangezien wij ons bezig gaan houden met het ontwikkelen van een webapplicatie is deze architectuur onbruikbaar.

De derde plaats is veroverd door de Presentation-Abstraction-Control architectuur. Het is een architectuur die duidelijk is en voldoet aan bijna alle eisen die we vooraf hebben gesteld (herbruikbaarheid, uitbreidbaarheid). Echter hebben wij het idee dat op basis van wat wij gezien en geleerd hebben van deze architectuur het veel te uitgebreid is voor de applicatie die wij voor ogen hebben waardoor het onderhouden ervan een lastige taak is.

Zoals we vaker op internet hebben aangetroffen gaat het tussen MVC en MVP. Er zijn flink wat websites die aandacht besteden aan de beide architecturen. Verschillen zijn er eigenlijk nauwelijks waardoor het lastig te zeggen is welke architectuur beter is.

Zowel MVC als MVP voldoen aan de eisen welke we vooraf hebben gesteld. Beiden bieden makkelijk herbruikbare, uitbreidbare en onderhoudbare code op een gestructureerde manier. Op welke architectuur is de keuze dan uiteindelijk gevallen? Wij gaan voor het MVC Passive Model in combinatie met een Front Controller.

De reden dat we voor het Passive Model van het MVC kiezen is omdat dit model veel gebruikt wordt voor webapplicaties en de architectuur overeenkomt met het MVP. Het enige verschil is dat de view data ophaalt uit het model wat bij MVP niet het geval is. Voor de rest is de werking ervan vrijwel gelijk. Een andere reden om voor deze architectuur te kiezen is het feit dat alle bekende PHP frameworks werken met het MVC model. Het is nog niet zeker of we een framework gaan gebruiken, maar door voor deze architectuur te kiezen blijft de mogelijkheid in elk geval bestaan.

De Front Controller gaan we gebruiken omdat dit een zeer handige toevoeging is op het MVC model. Zoals eerder gezegd heeft dit als voordeel dat we o.a. de controle van de sessie op één plaats kunnen regelen. Dit zorgt er weer voor dat de code overzichtelijk en makkelijk onderhoudbaar blijft.

## Bronvermelding

[1] Florijn, G (1998), *Whitepaper Software architectuur* -

[http://cibit.nl/site.nsf/0/AAAD6925A3453EBDC1256E43003908E4/\\$file/Whitepaper\\_Software\\_architectuur.pdf](http://cibit.nl/site.nsf/0/AAAD6925A3453EBDC1256E43003908E4/$file/Whitepaper_Software_architectuur.pdf) (pagina 22).

## Literatuurlijst

Bergen, P van, *Model-View-Controller* - [http://www.dossier-andreas.net/software\\_architecture/mvc.html](http://www.dossier-andreas.net/software_architecture/mvc.html)

Bergen, P van, *Presentation-Abstraction-Control* - [http://www.dossier-andreas.net/software\\_architecture/pac.html](http://www.dossier-andreas.net/software_architecture/pac.html)

Buschmann, F, Meunier, R, Rohnert, H, Sommerlad, P, Stal, M (2000), *Pattern: Presentation-Abstraction-Control* - <http://www.vico.org/pages/PatronsDisseny/Pattern%20Presentation%20Abstra/>

*Design: The front controller and php* - [http://www.phppatterns.com/docs/design/the\\_front\\_controller\\_and\\_php](http://www.phppatterns.com/docs/design/the_front_controller_and_php)

*Document/View Architecture* - <http://msdn.microsoft.com/en-us/library/4x1xy43a%28VS.80%29.aspx>

Esposito, D (2009), *Evergreen but still topical: MVC vs. MVP* - <http://dotnetslackers.com/articles/designpatterns/Evergreen-but-still-topical-MVC-vs-MVP.aspx>

Fowler, M (2006), *Passive View* - <http://martinfowler.com/eaDev/PassiveScreen.html>

*Front Controller* - [http://www.phpwact.org/pattern/front\\_controller](http://www.phpwact.org/pattern/front_controller)

Marston, T (2004), *The Model-View-Controller (MVC) Design Pattern for PHP* - <http://www.tonymarston.net/php-mysql/model-view-controller.html>

*Model View Controller* - [http://www.phpwact.org/pattern/model\\_view\\_controller](http://www.phpwact.org/pattern/model_view_controller)

Ormerod, M (2009), *Defining the OpenEdge® Reference Architecture - Presentation: Model-View-Controller Pattern* - <http://communities.progress.com/pcom/docs/DOC-14628%3B>

Schall, D (2004), *MVC vs MVP* - <http://www.darronschall.com/weblog/2004/06/mvc-vs-mvp.cfm>

Snyder, T (2007) - *MVC or MVP Pattern – Whats the difference?* - [http://blogs.infragistics.com/blogs/todd\\_snyder/archive/2007/10/17/mvc-or-mvp-pattern-whats-the-difference.aspx](http://blogs.infragistics.com/blogs/todd_snyder/archive/2007/10/17/mvc-or-mvp-pattern-whats-the-difference.aspx)

**Bijlage B4**

# Onderzoek: Outlook Add-on

Een toevoeging voor HourPower

Daniël van Zoelen  
Jeffrey Veldkamp



## Inhoudsopgave

Inleiding .....	3
Het onderzoek .....	4
Microsoft .....	4
Visual Basic .....	4
Visual Studio .....	4
Visual Studio Tools for Office .....	4
Add-in Express .....	4
De talen .....	4
VBA .....	5
Visual Basic.NET .....	5
.NET .....	5
C++ .....	5
C# .....	5
Kosten .....	5
Is het mogelijk? .....	6
Conclusie .....	7
Bronvermelding .....	8

## Inleiding

Outlook is een van de meest gebruikte e-mail applicatie in de zakelijke wereld, dit omdat naast e-mail het ook kan figureren als agenda, taak manager, contact manager en notitieblok. Met een add-on is het mogelijk om Outlook te beheren op een BlackBerry of telefoon.

Om meer functionaliteit toe te voegen aan het, nog te ontwikkelen, urenregistratiesysteem onderzoeken we of het mogelijk is om een add-on te ontwikkelen die een verbinding maakt tussen onze applicatie en Outlook.

Een van de gewenste functies voor de add-on is het aanpassen van de Outlook agenda zodra er een verandering is in de agenda van de klant in het urenregistratiesysteem en vice versa.

## Het onderzoek

Om een add-on te maken kan je gebruik maken van verschillende bewerkingssoftware op de markt, ieder met een eigen structuur en manier van aanpak.

### Microsoft

Microsoft heeft de volgende programma's ontwikkeld voor het ontwikkelen van een add-on voor Outlook en Office Outlook

- Visual Basic
- Visual Studio
- De add-on voor Visual studio: Tools for Office - voor Outlook 2003 en later

### Visual Basic

De eerste versies van Visual Basic waren een programmeringomgeving. Met Visual Basic kon men snel en op een visuele manier grafische applicaties bouwen zonder het invoeren van programmacode.

Later werd Visual Basic als programmeercode opgenomen in het .Net framework, wat gebruikt wordt in o.a. Visual Studio.

### Visual Studio

Visual Studio is een programmeer ontwikkel applicatie van Microsoft, waarmee het mogelijk is om programma's te ontwikkelen voor in de Windows omgeving. Visual Studio maakt gebruik van het .NET framework. In dat framework zitten de talen die nodig zijn voor het maken van een add-on: Visual Basic.Net, C# en Visual C ++. De huidige versie van Visual studio is alleen in staat om add-ons te maken tot Outlook 2003, om latere versie te kunnen bewerken moet er een add-on geïnstalleerd worden, Visual Studio Tools for Office.

### Visual Studio Tools for Office

In Visual Studio is het mogelijk om een extra add-on te installeren, Visual Studio Tools for Office(VSTO). Deze add-on maakt het mogelijk om zelf add-ons te schrijven voor alle applicaties in Microsoft Office 2003 en later. Een van de voordelen met de VSTO add-on is dat de gemaakte add-ons geschreven kunnen worden in de .Net framework talen. Voorheen was het alleen mogelijk om add-ons te maken in DLL of in COM, geschreven in Visual Basic, C++ of VBA . VSTO is standaard opgenomen in Visual Studio Professional Edition.

### Add-in Express

Add-in Express is zowel een alternatief als toevoeging voor Microsoft Visual Studio Tools for Office, voor het ontwikkelen van uitbreidingen voor Office. Het programma maakt het makkelijker om add-ons te ontwikkelen omdat het programma het coderen van de user interface voor zijn rekening neemt, en het helpt ontwikkelaars met prototypen en debuggen.

Add-in Express geeft de ontwikkelaar drie keuzes: werken met het .NET framework , met VCL (Visual Component Library) of als add-on voor Visual Studio Tools for Office.

### De talen

Deze programma's maken gebruik van deze talen voor het maken van een add-on.

- VBA
- VB.NET
- .NET

- C++
- C#

## VBA

Visual Basic for Applications (VBA) is de programmeertaal van Visual Basic. Met behulp van VBA is het mogelijk toepassingen te automatiseren en de functionaliteit van de applicatie kan daardoor enigszins worden uitgebreid.

Het is niet mogelijk om met VBA een losstaande applicatie te bouwen, het is alleen bedoeld om een applicatie uit te breiden. Een groot nadeel aan VBA is dat het gebruikt kan worden om macro's te schrijven met kwade intenties. Nu kan dit met meerdere programmeertalen, echter met VBA ligt de beveiliging in handen van de gebruiker, niet in die van de auteur. Vooral onervaren computer gebruikers kunnen hierdoor ontzettend veel schade aanrichten terwijl ze het zelf niet door hebben.

## Visual Basic.NET

Sinds 2002 is Visual Basic opgenomen in het .Net framework. Het is sindsdien een objectgeoriënteerde programma taal geworden. Omdat het is opgenomen in het .NET framework is het niet meer "backwards compatible". Dat wil zeggen dat het niet meer mogelijk is om onderdelen die nu in Visual Basic worden gemaakt niet meer geopend en bewerkt kunnen worden in vorige versies van Visual Basic.

## .NET

Het .Net framework is ontwikkeld door Microsoft en geldt als ontwikkelingsplatform voor Windows applicaties. Het framework is ontworpen om een samenwerking te creëren tussen applicaties en bibliotheken, geschreven in verschillende programmeertalen, waaronder: C#, C++, .NET, Visual Basic.NET).

## C++

C++ is een objectgeoriënteerde programma taal, en is een uitbreiding op C. Het wordt beschouwd als de meest populaire programmeertaal en wordt gebruikt voor applicatiesoftware tot video games.

## C#

Gebaseerd op C++ is C# ("C sharp") een objectgeoriënteerde programmeertaal, ontwikkeld door Microsoft als deel van het .Net initiatief. Het bevat elementen van diverse andere programmeertalen (voornamelijk Delphi en Java). De taal C# is samen met het .NET framework geïntroduceerd, en wordt beschouwd als een van de belangrijkste talen voor het platform.

## Kosten

Hier een overzicht met de aanschafprijzen van de pakketten.

Pakket	Aanschafprijs
Visual Studio 2008 Standard Edition	\$ 299,-
Visual Studio 2008 Professional Edition	\$ 799,-
Add-in Express 2009 Standard	\$ 349,-*
Add-in Express 2009 Professional**	\$ 449,-*
Add-in Express 2009 Premium	\$ 949,-*

\*prijzen voor de .NET, VCL en VSTO zijn gelijk

\*\* Geen Professional versie van VSTO

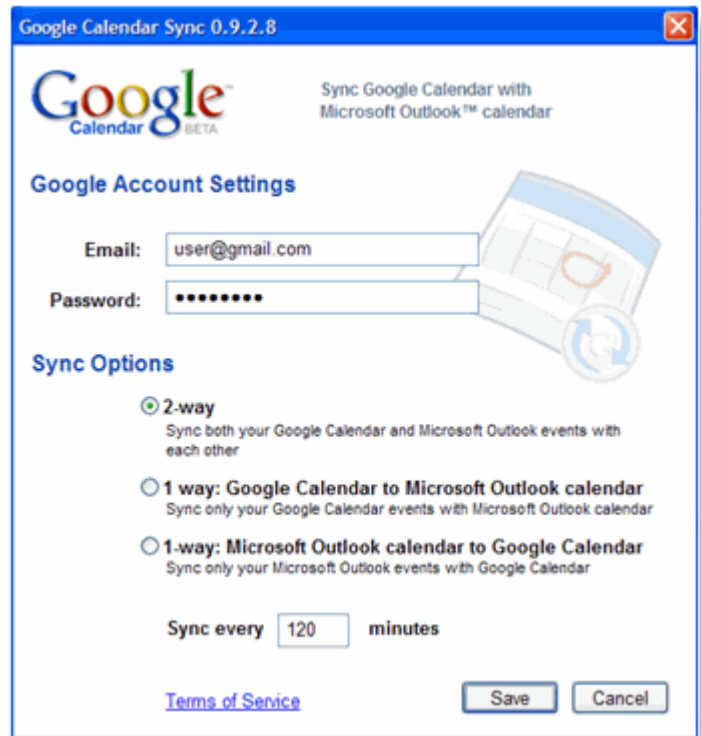
Om gebruik te maken van Add-in Express VSTO, moet men gebruik maken van Visual Studio Professional. De totaalprijs van beide pakketten is dan \$ 1.148,- voor de Add-in Express Standard en \$ 1.748,- voor de Add-in Express Premium.

### Is het mogelijk?

Inmiddels weten we dat het mogelijk is om een add-on te maken. Nu is het ook belangrijk te weten of het mogelijk is om de add-on te laten doen wat we willen dat hij doet.

Dus kunnen we met een add-on een connectie maken met een website, en is het mogelijk om gegevens terug te sturen naar de website?

Ja, dat is mogelijk. Een add-on genaamd Google Calander Sync synchroniseert de gebruikers agenda van Outlook met de online agenda van Google.



Figuur 1 – installatiescherm

<http://www.google.com/support/calendar/bin/answer.py?hl=en&answer=89955>

## Conclusie

Het is mogelijk om een add-on te maken en het te laten doen wat we willen, echter het maken van de add-on is nog onbekend terrein. Omdat de talen waarin een add-on gebouwd wordt onbekend voor ons zijn moeten we deze eerst onder de knie krijgen. Hetzelfde geldt ook voor het gebruiken van de ontwikkelingsprogramma's.

Het is op dit moment voor ons niet mogelijk om een add-on te maken voor onze applicatie. Hiervoor zou een nieuw project gestart moeten worden.

Mochten we er voor kiezen om te starten met het project, dan zullen gebruik gaan maken van Add-in Express .NET of Add-in Express VCL. Het is niet de goedkoopste oplossing, maar het programma vereenvoudigt het ontwikkelen van een add-on dusdanig dat wij hier de meerwaarde wel van zien.

## Bronvermelding

- <http://support.microsoft.com/?kbid=291163>
- <http://www.add-in-express.com/add-in-net/>
- <http://www.add-in-express.com/add-in-delphi/>
- <http://www.add-in-express.com/add-in-vsto/>
- <http://www.outlookcode.com/article.aspx?id=36>
- <http://www.google.com/support/calendar/bin/answer.py?answer=89955>
- <http://www.microsoft.com/visualstudio/en-us/howtobuy/default.aspx#price-comparison-chart>