

Eindverslag

Ontwikkelen van TO&I

Webbased-urenregistratie



Naam:	Assan Darwich
Student nummer:	99000050
Afstudeerblok:	2004-2.1
Bedrijf:	TU Delft, Faculteit Bouwkunde
Afdeling:	Technische Ontwerp en Informatica (TO&I)
Bedrijfsmentor1:	Dhr. dr. ir. R. Stouffs
Bedrijfsmentor2:	Dhr. ir. E. J. Janssen Groesbeek
Examinator 1:	Dhr. P.R.C. Breukel
Examinator 2:	Dhr. M.G. van Luijk

Colofon

Naam: Assan Darwich
Studentnummer: 99000050
Opleiding: Informatica en Informatiekunde
Afstudeerrichting: Ontwikkeling van software en technische infrastructuren (OSTI)
Afstudeerblok: 2004-2.1
Afstudeerperiode: 30 - augustus - 2004 / 14 - januari - 2005
Afstudeerproject: Ontwikkelen van het webbased-urenregistratiesysteem, voor de docentenassistenten van de Leerstoel Technisch Ontwerp & Informatica (TO&I), TU Delft, Faculteit Bouwkunde

Afstudeeromgeving: TU Delft, Faculteit Bouwkunde,
Leerstoel Technisch Ontwerp & Informatica (TO&I)

Bedrijfsmentoren: Dhr. dr. ir. R. Stouffs
Dhr. ir. E. J. Janssen Groesbeek

Examinatoren: Dhr. P.R.C. Breukel
Dhr. M.G. van Luijk

Contactgegevens Haagse Hogeschool

Bezoekadres:
Johanna Westerdijkplein 75
2521 EN Den Haag

Postadres:
Postbus 19
2520 CB Den Haag

Telefoon: +31 (0) 70 455 84 00
Fax: +31 (0) 70 455 84 05
Internet: <http://www.hhs.nl>

Contactgegevens TU Delft, Faculteit Bouwkunde

Bezoekadres:
Berlageweg 1
2628 CR Delft

Postadres:
Postbus 5043
2600 GA Delft

Telefoon: +31 (0) 15 278 41 84
Internet: <http://www.bk.tudelft.nl>

Referaat

Assan Darwich, Eindverslag, Ontwikkelen van het webbased-urenregistratiesysteem, voor de docentenassistenten van de Leerstoel Technisch Ontwerp & Informatica (TO&I), TU Delft, Faculteit Bouwkunde.

Dit verslag is geschreven in het kader van mijn afstudeerstage bij de leerstoel Technisch Ontwerp & Informatica aan de faculteit bouwkunde van de TU Delft. Deze stage is de laatste fase van de opleiding Informatica & Informatiekunde (I&I) aan de sector informatica van de Haagse Hogeschool. In de periode van 30 augustus 2004 t/m 14 januari 2005 heb ik me beziggehouden met het afstudeerproject genaamd “Ontwikkelen van het webbased-urenregistratiesysteem, voor de docentenassistenten van de Leerstoel Technisch Ontwerp & Informatica (TO&I), TU Delft, Faculteit Bouwkunde” bij de leerstoel technisch ontwerp & informatica.

In dit verslag wordt een beschrijving gegeven van de organisatie en de opdracht. Vervolgens wordt de gekozen aanpak en de uitgevoerde werkzaamheden aan de hand van de fasering van het project beschreven. Ten slotte wordt het afstuderen op het proces en de geleverde producten geëvalueerd.

Descriptoren:

- Eindverslag
- TU Delft
- Faculteit bouwkunde
- Haagse Hogeschool
- Data base
- webbased-urenregistratiesysteem
- PHP
- IAD, Iteratieve Application Development
- UML, Unified Modeling Language

Voorwoord

Dit verslag is het resultaat van mijn afstudeerproject bij de Leerstoel Technisch Ontwerp & Informatica aan de faculteit bouwkunde van de TU Delft. Dit project is bedoeld ter afsluiting van de vierjarige opleiding Informatica en Informatiekunde aan de sector Informatica van de Haagse Hogeschool.

In dit voorwoord wil ik gebruik maken van de gelegenheid iedereen te bedanken, die een bijdrage heeft geleverd aan het tot stand komen van dit verslag.

Ik wil mijn bedrijfsmentoren Dhr. R. Stouffs en Dhr. E. J. Janssen Groesbeek bedanken voor hun begeleiding en ondersteuning tijdens het project. Ten tweede wil ik Dhr. Henry Kixen bedanken voor zijn begeleiding en de adviezen die ik van hem kreeg, ook omdat hij me te woord heeft gestaan bij vragen en problemen. Ten derde dank aan alle personen van de leerstoel Technisch Ontwerp & Informatica binnen de afdeling “Building Technology” van de faculteit bouwkunde die mij in wat voor vorm dan ook geholpen hebben met het behaalde resultaat.

Verder wil ik ook mijn examiners Dhr. P.R.C. Breukel en Dhr. M.G. van Luijk bedanken voor hun begeleiding en de nuttige adviezen tijdens mijn afstudeerperiode.

Assan Darwich

Delft, 9 januari 2005

Inhoudsopgave

1	Inleiding	5
2	Organisatie 'TU Delft'	6
2.1	Algemene beschrijving van de organisatie	6
2.2	Leerstoel TO&I (Technisch Ontwerp en Informatica)	8
2.3	Betrokken partijen	9
2.4	Plaats van de afstudeerder in de organisatie	10
3	Opdracht (Ontwikkelen van een Webbased-Urenregistratiesysteem)	11
3.1	Probleemstelling	11
3.2	Doelstelling	11
3.3	Op te leveren producten	12
3.4	Doelstellingen op te leveren producten	12
4	Plan van aanpak	13
4.1	Ontwikkelmethode	13
4.2	Gekozen strategie	15
4.3	Techniek	17
4.4	Planning	18
5	Bepalen database	19
6	Definitiestudie	21
6.1	Huidige situatie	21
6.2	Opstellen systeemeisen	22
6.3	Samenstellen systeemconcept	24
6.4	Opstellen pilotplan	27
6.5	Iteratiestrategie pilotontwikkeling	29
6.6	Iteratie	29
7	Pilot 1 – Database TO&I DB-UR (Database Urenregistratie)	31
7.1	Eerste opzet	31
7.2	Specificatie van de database	32
7.3	Relationeel representatie model	34
7.4	Realiseren database	35
8	Pilot 2 – Realiseren administratiesysteem	37
8.1	Opstellen plan van aanpak	38
8.2	Ontwerpen en ontwikkelen user-interface	38
8.3	Ontwerpen functionaliteit	41
8.4	Ontwikkelen functionaliteit	46
9	Pilot 3 – Overzichten	51
9.1	Opstellen plan van aanpak	51
9.2	Ontwerpen en ontwikkelen user-interface	52
9.3	Ontwerpen functionaliteit	53
9.4	Ontwikkelen functionaliteit	55
9.5	Beoordelen en testen pilots	56
10	Evaluatie	57
10.1	Procesevaluatie	57
10.2	Productevaluatie	59
	Literatuurlijst	60
	Afkortingenlijst	61
	Figurenlijst	62
	Bijlagen	63
	INTERNE BIJLAGE 1 Opdrachtoomschrijving	64

1 Inleiding

Dit verslag is opgesteld naar aanleiding van de afstudeerstage, die ik heb uitgevoerd in opdracht van de opleiding Informatievoorziening & Informatietechnologie aan de Haagse Hogeschool. Deze praktische afstudeerstage heeft plaatsgevonden bij de leerstoel Technisch Ontwerp & Informatica aan de faculteit bouwkunde van de TU Delft. Het uitgevoerde afstudeerproject wordt in dit eindverslag beschreven.

Op basis van dit verslag moeten de examencommissie en een lid van de commissie van toezicht inzicht krijgen in de gebruikte aanpak en de werk- en denkwijze van de afstudeerder met betrekking tot het afstudeertraject. Hiermee moet tevens duidelijk worden hoe de eindproducten tot stand zijn gekomen.

Het afstudeerproject vond plaats tussen 30 augustus 2004 en 14 januari 2005 en dient afgesloten te worden met een eindverslag. Dit verslag is bestemd voor de examinatoren en de rijksgecommitteerde. Zij kunnen een uitspraak doen over de kwaliteit van de gehanteerde werkwijze van de uitgevoerde werkzaamheden.

Het verslag is als volgt opgebouwd:

- In hoofdstuk 2 wordt een beschrijving gegeven van de organisatie en de afdeling waar ik werkzaam was tijdens het afstudeerproject.
- In hoofdstuk 3 wordt de afstudeeropdracht beschreven.
- Hoofdstuk 4 beschrijft de totstandkoming van het plan van aanpak.
- Hoofdstuk 5 beschrijft hoe de database is bepaald
- Hoofdstuk 6 beschrijft de activiteiten, die hebben geleid tot de definitiestudie.
- Hoofdstuk 7 beschrijft de pilot 1 database.
- Hoofdstuk 8 en 9 beschrijven de pilot 2 administratiesysteem en pilot 3 overzichten. Hier worden de uitgevoerde werkzaamheden beschreven.
- In hoofdstuk 10 bevat uitgebreide product en proces evaluatie.

2 Organisatie ‘TU Delft’

Dit hoofdstuk omvat een omschrijving van de organisatie waarbinnen het afstudeerproject heeft plaatsgevonden. Vanuit een algemene visie op de afstudeeromgeving wordt steeds gedetailleerder ingegaan op de organisatie.

Naast de organisatie omschrijving komt in dit hoofdstuk ook de plaats van de afstudeerder binnen de organisatie aan bod.

2.1 Algemene beschrijving van de organisatie

In deze paragraaf staat de organisatie van de faculteit Bouwkunde globaal omschreven, en de afdeling Technisch Ontwerp & Informatica waar het afstudeerproject is volbracht wordt specifiek omschreven.

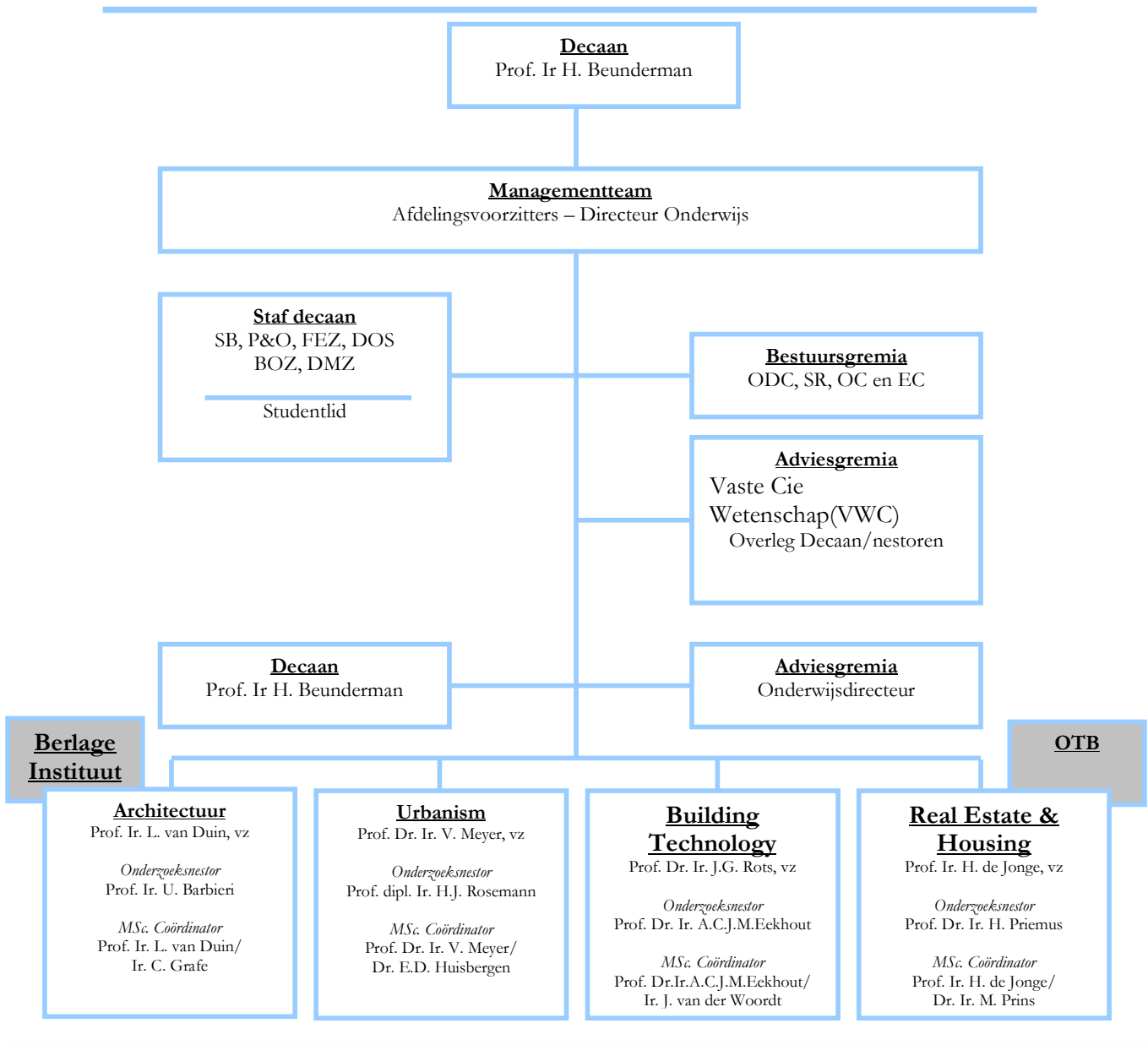
TU Delft

TU Delft staat voor Technische Universiteit Delft. De TU Delft werd opgericht in 1842 en is met meer dan 13.000 studenten, meer dan 2000 wetenschappers en 200 docenten de oudste, grootste en meest veelzijdige technische universiteit van Nederland.

De TU Delft werd alvorens zij de status universiteit verkreeg, gevormd door afdelingen. Na het verkrijgen van de universitaire status(1985), wordt gesproken van faculteiten. De TU Delft is op te splitsen in verschillende faculteiten. Deze faculteiten hebben ieder hun eigen werkgebied. De onderstaande faculteiten vormen samen de TU Delft. Binnen de faculteit Bouwkunde is het afstudeerproject “Ontwikkelen van het webbased-urenregistratiesysteem” uitgevoerd.

- **Bouwkunde**
- Civiele Techniek en Deelwetenschappen
- Elektrotechniek, Wiskunde en Informatica
- Luchtvaart- en Ruimtevaarttechniek
- Ontwerp, Constructie en Productie
- Techniek, Bestuur en Management
- Technische Natuurwetenschap

De faculteit Bouwkunde aan de TU Delft wordt gevormd door verschillende richtingen. In onderstaand organigram, is hiervan een overzicht weergegeven met de bijbehorende verantwoordelijke personen. Hieronder volgt het organigram van de faculteit bouwkunde.



Figuur 1. Organigram faculteit bouwkunde TU Delft

Uit het organigram is te zien dat de Faculteit Bouwkunde verschillende studierichtingen aanbiedt. In onderstaande tabel staan de richtingen weergegeven.

Richtingen binnen faculteit Bouwkunde	
Engelse benaming	Nederlandse benaming
• Architecturen • Building technologie • Reaal state & houding • Urbanisme	• Architectuur • Bouwtechnologie • Vastgoed & huisvesting • Verstedelijking

De afstudeerder heeft de opdracht binnen de afdeling Bouwtechnologie uitgevoerd. Ingezoomd op de afdeling Bouwtechnologie van de faculteit Bouwkunde kan onderscheid worden gemaakt in leerstoelen. Een leerstoel omvat een specifiek omschreven leeropdracht op het gebied van onderwijs en onderzoek. Een hoofdleraar bekleedt een leerstoel en is daarom ook verantwoordelijk voor de uitvoering van de bij leerstoel behorende leeropdracht. De afstudeerder heeft de opdracht uitgevoerd binnen de leerstoel Technisch Ontwerp en Informatica.

De leerstoelen binnen de afdeling Bouwtechnologie staan omschreven in onderstaande tabel. Deze staan allereerst omschreven in het Engels, aangezien deze termen worden gebruikt binnen de faculteit. Achter de Engelse benaming is de Nederlandse benaming weergegeven. Er wordt niet verder inhoudelijk op de leerstoelen ingegaan, omdat dit niet relevant is voor dit verslag.

Leerstoelen binnen faculteit Bouwkunde	
Engelse benaming	Nederlandse benaming
• Constructional integration & Coordination • Product development. • Environmental technical design. • Mechanics of building. • Building physics • Partitions & finishes • Technical design & Informatics • Installations • Structural design	• Constructie integratie & coördinatie. • Product ontwikkeling. • Natuurlijk technische ontwerp. • Mechanica van gebouwen. • Gebouw Natuurkunde. • Partities en finishes. • Technisch Ontwerp & Informatica. • Installaties. • Gestructureerd ontwerp.

2.2 Leerstoel TO&I (Technisch Ontwerp en Informatica)

De leerstoel TO&I is sinds 1993 opgericht door Prof. Dr. ir. arch. Sevil Sariyildiz. TO&I staat voor Technisch Ontwerp & Informatica. De leerstoel verstrekt onderwijs in de bachelor en in de masters Bouwtechnologie, Architectuur en Stedenbouw en doet onderzoek naar de inzet van ICKT (Informatie, Communicatie en Kennis Technologie) als tool, en al bestaande software programma's.

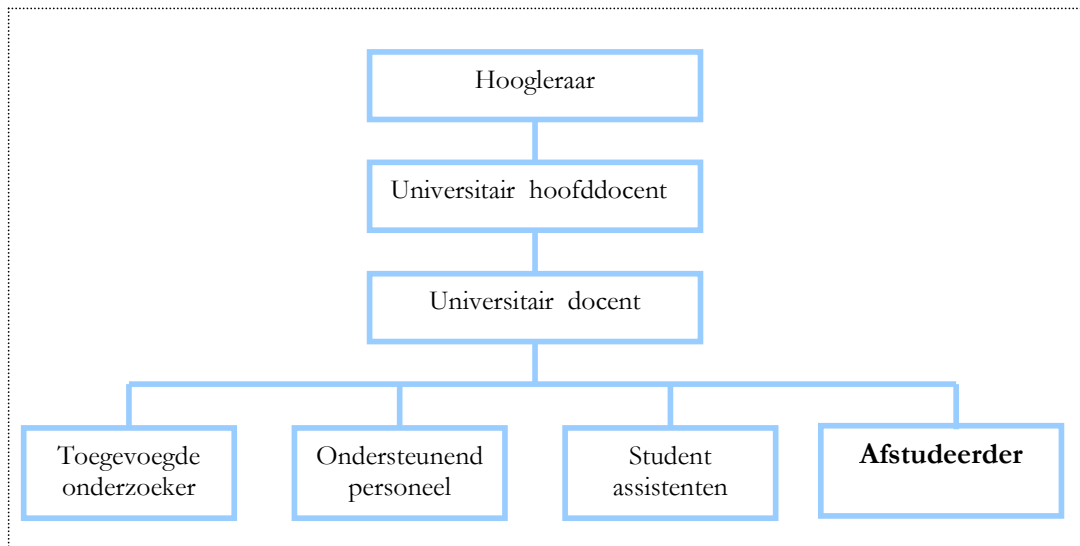
De leerstoel Technisch Ontwerp en Informatica is vooral gericht op het ontwerpen en materialiseren in het bouwproces. De leerstoel TO&I omvat ongeveer dertig werknemers. Waaronder een universitaire hoofddocent, universitair docent, hoofdleraar, toegevoegde onderzoekers, buitenlandse gastdocenten, student-assistenten, stagiaires en afstudeerders. (zie figuur 2 organigram TO&I)

2.3 Betrokken partijen

De partijen die bij het afstudeerproject betrokken zijn;

- ***Assan Darwich, afstudeerder***
Verantwoordelijk voor het uitvoeren van het project en het houden van contacten met de opdrachtgever, bedrijfsmentor en examiners.
- ***Dhr. dr. ir. R. Stouffs, opdrachtgever en bedrijfsmentor***
Verantwoordelijk voor het begeleiden van afstudeerder tijdens het project en op het technische vlak. Daarnaast contactpersoon binnen de leerstoel TO&I voor vragen en/of problemen van één van de betrokken partijen.
- ***Dhr. ir. E. J. Janssen Groesbeek opdrachtgever en bedrijfsmentor***
Verantwoordelijk voor het begeleiden van afstudeerder tijdens het project. Daarnaast contactpersoon binnen de leerstoel TO&I voor vragen en/of problemen van één van de betrokken partijen.
- ***Dhr. P.R.C. Breukel, examiner (Haagse Hogeschool)***
Beoordelende docent vanuit de Haagse Hogeschool.
- ***Dhr. M.G. van Luijk, examiner (Haagse Hogeschool)***
Beoordelende docent vanuit de Haagse Hogeschool.

2.4 Plaats van de afstudeerder in de organisatie



Figuur 2. Organigram TO&I

De afstudeerder treedt gedurende het afstudeerproject binnen de leerstoel TO&I op als systeemontwerper -en ontwikkelaar ten behoeve van het project genaamd **“Ontwikkelen van webbased-urenregistratiesysteem”**.

Aangezien de docenten van de afdeling TO&I de eindegebruikers zijn van deze applicatie, is het samenwerken met deze docenten belangrijk om het afstudeerproject tot een goed einde te brengen. In hoofdstuk 3 wordt meer uitleg over het afstudeerproject gegeven.

3 Opdracht (Ontwikkelen van een Webbased-Urenregistratiesysteem)

In dit hoofdstuk wordt de afstudeeropdracht beschreven. Er wordt een weergave gemaakt van de probleemstelling, de doelstelling en de op te leveren producten. De definitieve opdrachtomschrijving is te vinden in de interne bijlage 1

3.1 Probleemstelling

De leerstoel Technisch Ontwerp & Informatica (TO&I) behoort tot de faculteit Bouwkunde van de Technische Universiteit Delft. Bij deze afdeling van TU-Delft zijn 25 medewerkers werkzaam. TO&I houdt zich vooral bezig met informatica toepassingen, technieken en methoden en is gericht op ontwerpen en materialiseren in het bouwproces. Tevens geeft TO&I onderwijs en doet onderzoek naar het gebruik van computersoftware bij het ontwerpen en ontwikkelen van gebouwen.

In het kader van de aangeboden projecten dienen de medewerkers (leerstoel TO&I) een urenregistratie bij te houden. Dat houdt in dat iedere medewerker per maand aangeeft hoeveel uren hij of zij elke dag aan een project besteed heeft.

Er zijn momenteel geen duidelijke overzichten van de uren besteding aan iedere project, het kan voorkomen dat aan een klein project meer uren besteed wordt, dan aan een groter project.

Door het rooster van de docenten, kan het voorkomen dat de studenten bij een zelfde project door verschillende docenten te woord gestaan worden. Om de uren van de docenten bij te houden, wil men een digitale omgeving aanbieden, waarbij een duidelijk beeld wordt gegeven van de uren die op het rooster staan, plus de extra uren die worden besteed aan de werkzaamheden die buiten het rooster vallen, zoals het voorbereiden van de lessen en het nakijken van de wekstukken.

Op dit moment bestaat hiervoor een Excel bestand dat eens per maand ingevuld en uitgeprint wordt. Echter, dit Excel bestand is niet overzichtelijk en heeft een beperkte functionaliteit en uitbreidingsmogelijkheden (zoals het koppelen met Outlook).

3.2 Doelstelling

Het doel van de afstudeeropdracht is het ontwikkelen van een Webbased-urenregistratiesysteem dat het mogelijk maakt voor elke medewerker per dag aan te geven hoeveel uren hij of zij aan een project besteed heeft. Verder dient per project en per medewerker een overzicht uitgedraaid te kunnen worden van de totale urenbesteding.

De Website dient zowel een eindgebruikersdeel als een administratief deel te bevatten.

Administratief moet bepaald kunnen worden wie toegang heeft tot de Website, welk projecten bestaan en wie aan welke activiteiten deelneemt. Daarbij dient ook gebruik gemaakt te kunnen worden van een onderwijsrooster (een overzicht van de vakken die de docenten geven, wanneer, waar en voor wie). Hiermee dienen de ingeroosterde uren automatisch te worden geregistreerd in het Webbased-urenregistratiesysteem, ook dienen alle benodigde overzichten gegenereerd en uitgeprint te kunnen worden, zoals de koppeling van de studentenassistenten aan de onderwijstaken. De gebruiker dient per dag en per project te kunnen aangeven hoeveel uren hij of zij besteed heeft aan een project en een maandelijks overzicht te krijgen. De authenticatie moet verlopen via een wachtwoord en een gebruikersnaam. De opslag van gegevens dient na een onderzoek ofwel verzorgd te worden door InfoBase of een eigen My SQL database.

Het systeem zou ook een zekere intelligentie moeten kennen:

- Rekening houden met de huidige datum (dag en maand).
- Via een koppeling met Excel automatisch informatie uit het onderwijsrooster inlezen.
- Via een koppeling met Outlook automatisch informatie uit een agenda inlezen (de informatie over de agenda van een docent moet opgevraagd kunnen worden via Outlook en deze informatie overgenomen worden in het urenregistratiesysteem).

3.3 Op te leveren producten

Volgens de opdrachtomschrijving zullen de volgende producten opgeleverd worden;

- Het plan van aanpak.
- De definitiestudie.
- De pilotontwikkelplannen.
- De applicatie “webbased-urenregistratiesysteem”.

3.4 Doelstellingen op te leveren producten

Plan van aanpak

Het doel van het Plan van Aanpak is het geven van een gedetailleerde beschrijving van de werkzaamheden die door de projectgroep uitgevoerd gaat worden en dient daardoor als leidraad voor de organisatie van het project.

Definitiestudie

Het doel van de Definitiestudie is om de systeemgrenzen en de inhoud van het te ontwikkelen (op maat maken) en te implementeren delen van het webbased-urenregistratiesysteem vast te stellen. Het beschrijven van de huidige en gewenste situatie en ook het ontwikkelscenario is hierbij van groot belang.

Zo wordt er bepaald via welk pilots het systeem op maat zal worden gemaakt en wat de onderlinge prioriteiten zijn.

De definitiestudie komt iteratief tot stand en worden er de diverse aspecten met betrekking tot het op maat maken van het systeem beschreven. Elke iteratie van de definitiestudie voegt iets toe aan het rapport.

Pilotontwikkeling

Het doel van de Pilotontwikkeling is het beschrijven van de ontwikkeling van een pilot. De reikwijdte van het pilot is gedefinieerd in de fase Definitiestudie en in het pilotplan.

4 Plan van aanpak

Om inzicht te krijgen in de uit te voeren stappen, is een plan van aanpak voor het gehele project opgesteld. Het plan van aanpak dient als leidraad voor de uitvoering van het project tot aan de implementatie. In het plan van aanpak waren onder meer punten waarbij ik een keuze moest maken. Deze punten heb ik gebaseerd op een oriëntatie, veelal gesprekken met de opdrachtgevers en begeleiders, over de opdracht. Het betreft de methode, techniek en de strategie die ik gekozen en toegepast heb.

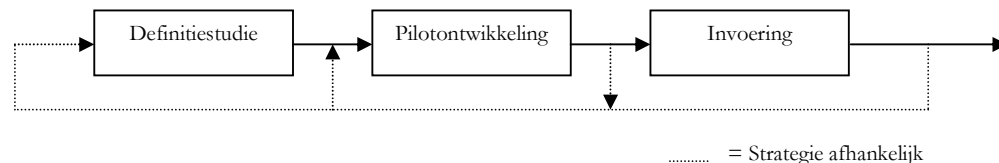
In dit hoofdstuk zullen werkzaamheden beschreven worden, die hebben geleid tot het plan van aanpak.

4.1 Ontwikkelmethode

Voor de ontwikkelmethode heb ik de methode IAD gekozen. IAD methode heb ik tijdens het laatste integrerend practicum (IP-03) van mijn opleiding toegepast. Hiermee had ik al ervaring opgedaan. Voordat ik de ontwikkelmethode heb gekozen, heb ik eerst de aandachtspunten binnen het project bekeken met betrekking tot het kiezen van een geschikte systeemontwikkelingmethode. Aan de hand van deze aandachtspunten heb ik kunnen kiezen welke methode het meest toepasbaar was. De aandachtspunten waarop de keuze gebaseerd zijn:

- De totale functionaliteit van de te ontwikkelen applicatie was in het eerste stadium nog niet definitief bekend. Hierdoor was er ook geen helder beeld over de user-interfaces van de applicatie.
- Er is een beperkte tijd van 15 weken om het project uit te voeren.

IAD (Iterative Application Development)



Figuur 3. Fasen systeemontwikkeling IAD

Iterative Application Development (IAD) is een methode die de gebruiker en de ontwikkelaar als gelijkwaardige partners ziet. De I van IAD staat naast iteratief (ontwikkelen in een aantal rondjes) voor interactief (samen met de gebruiker) en ook voor incrementeel (stapsgewijs uitbouwend tot een steeds grotere bruikbaarheid). IAD bestaat uit drie iteratieve fasen (figuur 4). Het doel van de definitiestudiefase is de analyse van de doelen, beperkingen en requirements, op basis waarvan een systeemconcept en een pilotplan worden opgesteld. Het pilotplan geeft aan welke pilots (incrementen) in welke iteratie worden ontwikkeld. In de pilotontwikkelingsfase worden het ontwerp en de implementatie van het systeem uitgewerkt. De uiteindelijk resulterende pilot wordt in de fase invoering operationeel gemaakt, geaccepteerd en ingevoerd in de organisatie.

Hieronder worden de voordelen en de nadelen van IAD verder toegelicht:

- Voordelen IAD

- Gebruikers en opdrachtgever worden nadrukkelijk bij het ontwikkelproces betrokken.
- Doordat cruciale delen van het systeem al in een vroeg stadium kunnen worden opgeleverd, kunnen hinderlijke knelpunten snel worden opgelost.
- Doordat het systeem niet eensklaps in zijn geheel opgeleverd wordt (watervalmethode), zijn de gebruikers al tamelijk vertrouwd met het systeem op het moment dat het volledig operationeel wordt.
- Het feit dat al in een vroeg stadium bruikbare delen van het systeem praktisch inzetbaar worden, heeft voordelen dat het project om wat voor redenen dan ook vroegtijdig wordt afgebroken.

- Nadelen IAD

- Oorspronkelijke doelstellingen van het informatiesysteem kunnen in de loop van het cyclische proces vervagen.
- Op de projectleider die een lineaire ontwikkelstrategie gewend is, kán de iteratieve aanpak in eerste instantie een desoriënterend effect hebben. Er zijn korte cycli, voortdurend bewegende doelen, veel minder mijlpaalproducten, een sterk toegenomen betrokkenheid van gebruikers en opdrachtgevers en korte, deels informele communicatielijnen.
- Als steeds het “korte termijnresultaat” voorop staat, zou dit kunnen leiden tot vervaging van de kwaliteit van de architectuur en de documentatie van het informatiesysteem.

(Bron: “IAD Het evolutionair ontwikkelen van informatiesystemen” R.J.H Tolido)

Binnen TO&I worden geen ontwerpmethoden en technieken gehanteerd bij de ontwikkeling van informatiesystemen. Er wordt een gegevensstructuur getekend en vervolgens wordt er gestart met het programmeren. Ik denk daarom, dat dit project een goede mogelijkheid biedt om TO&I te confronteren met een gestructureerde aanpak van systeemontwikkeling, waarbij verschillende fasen worden doorlopen, voordat men overgaat tot het bouwen van het systeem. Hierbij kan gedacht worden aan definitiefase, ontwerpfasen en implementatiefase.

Nadat ik deze methode heb gekozen heb ik een gesprek gehad met de opdrachtgever in dit gesprek heb ik de redenen uitgelegd die een rol hebben gespeeld bij mijn keuze.

De volgende redenen hebben een rol gespeeld bij mijn keuze.

- De mogelijkheid om eventuele feedback te verwerken heeft een belangrijke prioriteit waarbij de eindgebruikers meer bijbetrokken bij het ontwikkelproces hoeven te worden. IAD maakt dat mogelijk. Wijzigende eisen en wensen worden met IAD goed verwerkt in tegenstelling tot andere methoden. Bij mijn project wil ik wel de mogelijkheid hebben om gewijzigde wensen en/of eisen te kunnen doorvoeren.
- Bij eerdere ervaringen met IAD (IP-03 project) is ondervonden dat IAD een geschikte methode is om problemen die ontstaan zijn te vermijden door de iteratieve eigenschap van IAD. Door middel van de iteratiestrategie zoals Big-Bang wordt mogelijk om terug te schakelen naar vorige fase. Bij voorbeeld indien tijdens het ontwikkelen een eis vergeten is of een nieuwe eis bijkomt, kan dat nog opgenomen worden in de definitiestudie.

- Mijn project heeft zich binnen een innovatieve afdeling afgespeeld. Dit hield in dat de wensen en eisen van de applicatie van tevoren niet in zijn geheel vastgelegd konden worden. Dit kon verholpen worden door de gebruikers meer te betrekken bij de ontwikkeling van mijn project. Met IAD is dit mogelijk.

4.2 Gekozen strategie

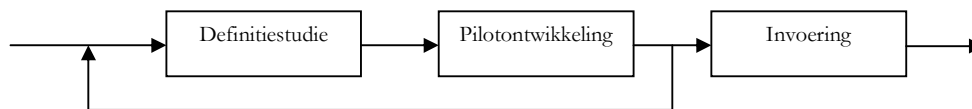
Voordat een planning gemaakt kon worden heb ik eerst een strategie bepaald die ik in mijn project kon gebruiken. Door middel van een strategie heb ik kunnen vastleggen hoe ik de fasen van IAD zou doorlopen. Ik zal de gekozen iteratiestrategie behandelen en valideren aan de hand van het boek **“IAD Het evolutionair ontwikkelen van informatiesystemen”** van R.J.H Tolido. IAD kent een viertal iteratiestrategieën. Deze zijn:

- Evolutionair ontwikkelen.
- Incrementeel opleveren.
- Incrementeel ontwikkelen.
- Big-bang invoeren

In het kader van dit project heb ik gekozen voor de strategie **‘Big-Bang invoeren’**.

Big-bang invoeren

Big-bang invoeren is een iteratiestrategie waarin de eerste twee fasen (definitiestudie en pilotontwikkeling) iteratief worden uitgevoerd, maar de laatste (invoering) niet. De invoering gebeurt pas na het laatste definitieve pilot. Een pilot is een stukje van het systeem wat onafhankelijk kan worden ontwikkeld en opgeleverd.



Figuur 4. Big-bang strategie van IAD

De keuze voor de gekozen iteratiestrategie is gebaseerd op de kenmerken van deze aanpak. De iteratiestrategie “Big-Bang invoeren” kan het beste gebruikt worden, indien:

- Er nog geen helder beeld is wat betreft de functionele eisen aan het gehele informatiesysteem, kan de aandacht gericht worden op de delen die al wel duidelijk zijn.
- Het te ontwikkelen informatiesysteem in een continu veranderende omgeving gaat functioneren, is het goed mogelijk dat ook de systeemeisen aan veelvuldige wijzingen onderhevig zullen zijn.
- Het systeem niet geschikt is voor partiele invoering. Er is geen behoefte om snel de delen van de applicatie operationeel te maken.
- De systeemeisen grotendeels bekend zijn.

De iteratiestrategie heb ik gekozen op basis van bovenstaande vier kenmerken. In overleg met de opdrachtgevers, is er besloten om de pilots afzonderlijk van elkaar te ontwikkelen en indien nodig te verbeteren door middel van iteratieslag. De verschillende pilotdelen worden in een testomgeving ontwikkeld. Die zullen ingevoerd worden als alle pilotdelen ontwikkeld zijn.

Aan deze iteratiestrategie zijn echter een aantal risico's verbonden waarmee rekening dient te worden gehouden, te weten:

- Te weinig feedback uit de praktijk:
Om dit risico te verkleinen zijn er wekelijkse vergaderingen gehouden met de personen welke bij dit project betrokken waren en projectbegeleider. Hierdoor kan dit risico vermeden worden.
- Invoering vindt in één keer plaats:
Het in een keer invoeren van een systeem is en blijft een moeilijk te besturen proces. Om dit risico te verkleinen is er getracht de opdrachtgever nauw te betrekken bij het ontwikkelproces.

Er is om de volgende redenen niet gekozen voor de andere strategieën binnen de IAD-methode:

- **Evolutionair ontwikkelen:**
Deze strategie wordt voornamelijk gebruikt wanneer er sprake is van onzekere systeemeisen, snel veranderende projectomgeving en wanneer de resultaten snel verwacht worden. De laatste eigenschap heeft geen betrekking tot de projecteigenschappen. Snel oplevering is namelijk niet gewenst.
- **Incrementeel opleveren:**
Deze strategie is goed toepasbaar bij duidelijke systeemeisen, een stabiele projectomgeving en bij oplevering van tussentijdse pilots. Omdat de tussentijdse oplevering van de pilots niet gewenst is, heb ik deze strategie niet gekozen.
- **Incrementeel ontwikkelen:**
Bij dit scenario gaat men ervan uit dat de systeemeisen en het systeemconcept van tevoren in grote lijnen zijn vastgesteld. Het systeem wordt vervolgens gerealiseerd in een aantal iteraties, waarna het systeem, als alle delen gerealiseerd zijn, in een keer ingevoerd kan worden in de organisatie.
Bij mijn project zijn er algemene wensen over de functionaliteit aanwezig, maar de specifieke systeemeisen en het systeemconcept zijn niet in grote lijnen vastgesteld.

4.3 Techniek



Nadat de keuze voor de methode is gemaakt, is er besloten welke technieken tijdens het uitvoeren van het project gebruikt zullen worden. De modellering van het systeem zal geschieden met behulp van de Unified Modeling Language (UML). UML is een objectgeoriënteerde modelleertaal om modellen te creëren van systemen. Gezien de ervaring die is opgedaan tijdens de opleiding is deze techniek gekozen.

Voordat deze techniek is gekozen, is dat met de opdrachtgevers besproken. De opdrachtgevers waren er mee eens met de keuze aangezien ze met deze techniek al bekend waren.

Naast UML heb ik de volgende technieken toegepast tijdens het afstudeerproject;

- **Interviewen**

De doelstelling van het interview is het vergaren van de wensen van de opdrachtgever en andere betrokkenen. Dit is zeer specifieke informatie die door de interviewer wordt gezocht. Welke informatie de interviewer wil weten is voor het grootste gedeelte van tevoren bekend. Het is de taak voor de interviewer om de inhoud te vergaren. Hierom valt de keuze op een onderzoeksinterview.

Ik heb interviewtechnieken gebruikt om over het project informatie in te winnen. Tijdens de inventarisatieronde heb ik een aantal gebruikers en opdrachtgever geïnterviewd. De resultaten van de interviews zijn als gegeven beschouwd om een beter beeld te kunnen krijgen over de huidige en de gewenste informatieverzorging naar aanleiding van het project. Tijdens het interviewen heb ik open vragen gehanteerd om de geïnterviewde volledig in zijn antwoorden vrij te laten.

Om de eisen en wensen af te bakenen en steeds duidelijker te maken kwamen er gesloten vragen achter. Voor elk onderwerp is er een hoofdvraag gehanteerd. Om de informatie te concretiseren en te preciseren werd er doorgevraagd waarmee de geïnterviewde gerust alle informatie over het onderwerp kon vertellen.

4.4 Planning

De planning van dit project heb ik opgesteld aan de hand van de uit te voeren activiteiten en is opgenomen in het plan van aanpak. (**zie externe bijlage A**). De planning is verdeeld in de te doorlopen fasen namelijk: het plan van aanpak, de definitiestudie en de pilotontwikkeling. De fase pilotontwikkeling heb ik in de planning ook onderverdeeld. De reden van deze onderverdeling is om een helder beeld te scheppen over tijdsinspanning van de verschillende activiteiten voor elke afzonderlijke fase.

Voor de definitiestudie heb ik 3 weken gepland, omdat het naar mijn mening voldoende was om inzicht te krijgen over de huidige situatie en het systeemconcept op te stellen.

Het pilotontwikkelplan neemt het grootste deel van mijn planning in beslag. Dit heeft vooral te maken met het feit dat voor elke pilot een apart ontwikkelplan opgesteld wordt en dat alle pilots ontwikkeld zullen worden in deze fase. De tijdspanne voor deze fase bestaat uit ongeveer 2,5 maand.

Ook voor het opstellen van het concepteindverslag heb ik één week ingepland. Dit was in het begin van dit project aangeraden door de examinatoren. Omdat ik over dit verslag wordt beoordeeld is het van belang dat de opbouw en inhoud van dit verslag voldoende idee geeft over de activiteiten die tijdens de afstudeerfase uitgevoerd zijn.

Hieronder ziet de globale planning die ik opgesteld heb, namelijk;

Id	Activiteiten	Begin	Eind	Duur	sep 2004					okt 2004					nov 2004					dec 2004					jan 2005		
					29-8	5-9	12-9	19-9	26-9	3-10	10-10	17-10	24-10	31-10	7-11	14-11	21-11	28-11	5-12	12-12	19-12	26-12	2-1	9-1	16-1		
1	Het opstellen van plan van aanpak	30-8-2004	3-9-2004	1w																							
2	Het opstellen van definitiestudie	6-9-2004	24-9-2004	3w																							
3	Het ontwikkelen van pilots	27-9-2004	19-11-2004	8w																							
4	Het schrijven van concept eindverslag	29-11-2004	3-12-2004	1w																							
5	Het ontwikkelen van pilots(vervolg)	29-11-2004	10-12-2004	2w																							
6	Het invoeren van het systeem	13-12-2004	17-12-2004	1w																							
7	Het bijwerken en opleveren van eindverslag	20-12-2004	7-1-2005	3w																							

Figuur 5. Planning

De gehele planning kunt u terugvinden in **externe bijlage A**, Plan van aanpak.

5 Bepalen database

Om te kunnen bepalen welke database is het beste voor de applicatie ‘webbased-urenregistratiesysteem’ moest ik een duidelijk beeld hebben over het Infobase project en Infobase-Studentwork om dit beeld te kunnen krijgen heb ik informatie verzameld door middel van het interviewen met verschillende projectmedewerkers. Ik heb verschillende projectmedewerkers geïnterviewd. Tijdens de wekelijkse vergaderingen met mijn opdrachtgevers heb ik ook de mogelijkheid gehad om informatie in te winnen over het Infobase project. Volgende hoofdvragen heb ik gebruikt om een beeld te krijgen over het Infobase project;

- **Wat is het doel van Infobase project?**
- **Hoe is het project opgebouwd?**
- **Wat is het doel van het onderdeel "infobase studentwork"?**
- **Wat zijn de huidige functionaliteiten van infobase?**
- **Wie maakt er gebruik van infobase? (gebruikersdoelgroep)**
- **Uitgebreide informatie over de interfaces van infobase?**

Conclusie

Na het verwerken van de informatie heb ik het volgende geconstateerd.
De volgende functionaliteiten maken deel uit van het bestaande deel:

- Een gebruiker kan op de bestaande delen van de applicatie inloggen.
- Er is sprake van een rechtenstructuur. Iedere gebruiker kan per project verschillende rechten hebben. Afhankelijk hiervan krijgt de gebruiker meer of minder mogelijkheden te zien. De rechten van de gebruikers worden door systeemadministrator toegekend.
- Afhankelijk van de status die de gebruiker heeft, krijgt de gebruiker verschillende rechten. Bij "infobase studentwork" is de gebruiker een student of een docent.
- Een student van de faculteit bouwkunde kan haar/zijn eindproduct voor een willekeurig vak via de inleverpagina van Infobase-Studentwork inleveren.
- Een docent kan de ingeleverde producten beoordelen.

De projectgegevens kunnen beheerd worden. Deze worden door de docenten beheerd. Het is afhankelijk van de rechten die de docenten hebben. Om een project te kunnen beheren hoort de ingelogde gebruiker "Admin" rechten te hebben. Hierbij gaat het om het beheren van de gegevens zoals de periode, het semester, de opgaven en het beoordelen van deze opgaven die bij het project horen

Na dit onderzoek heb ik de voordelen en de nadelen van infobase op een rijtje gezet;

Voordelen

- Er is een bestaande database.
- Er zijn tabellen die gebruikt kunnen worden in mijn database (zoals project).
- Er zijn bestaande tools en APIs (application program interface) die ik kan gebruiken.

Nadelen

- De bestaande database is te complex, het zal me veel tijd kosten om te begrijpen hoe deze database in elkaar zit.
- Te complexe datamodel voor mijn applicatie (het datamodel bevat veel tabellen met veel relaties).
- Meer dan 90% van infobase sluit niet op mijn applicatie (het datamodel bevat veel tabellen terwijl ik van 1 of 2 tabellen gebruik zal maken).
- Een simpel model omvormen naar infobase vergt veel tijd en moeite.
- Bij het wijzigen van de inhoud van een of ander tabel in infobase kan gevolgen hebben op mijn applicatie of de andere onderdelen die onder infobase draaien.
- Infobase wordt beheerd door systeemadministrator extern (in mijn geval door docent admin intern).

Na het vergelijken van de voordelen en de nadelen heb ik geconcludeerd dat het niet verstandig is om van Infobase gebruik te maken.

De redenen hiervoor waren:

- Het is niet moeilijk voor mij om een nieuwe My SQL database te creëren.
- Ik heb een korte tijd voor mijn project, daarom wilde ik niet veel tijd besteden om een complexe database te gaan bestuderen terwijl ik van een of twee tabellen gebruik zal maken.
- Bij het wijzigen van de inhoud van een of ander tabel in infobase kan gevolgen hebben op mijn applicatie of de andere onderdelen die onder infobase draaien.
- Als ik van de bestaande tools en APIs (application program interface) gebruik maak, zal dat me zoveel tijd kosten aan het bestuderen en aanpassen als nieuwe code zelf schrijven.

Om de gevonden resultaten met de opdrachtgever te bespreken heb ik met hem een afspraak gemaakt. Na dit gesprek was hij overtuigd van mijn gevonden resultaat en daarom hebben we besloten om een eigen My SQL database te ontwikkelen.

6 Definitiestudie

Na het bepalen welke methode en strategie ik zou gebruiken in mijn project, ben ik mij gaan verdiepen in de inhoudelijke kant van het project. De definitiestudie was de start daarvoor. De definitiestudie geeft inzicht in het verdere verloop van het project; **webbased-urenregistratiesysteem**. Het doel van deze definitiestudie is, overeenstemming met de opdrachtgever te bereiken over de definities en grenzen van dit project.

In dit hoofdstuk worden de werkzaamheden beschreven die zijn uitgevoerd in het kader van het rapport definitiestudie. De huidige situatie, de systeemeisen, het systeemconcept en het pilotplan worden uitvoerig besproken. De definitiestudie is als externe bijlage opgenomen.

De volgende punten zijn opgenomen in de definitiestudie en worden toegelicht omdat deze belangrijkste punten zijn uit de definitiestudie voor dit project:

- Huidige situatie.
- Systeemeisen.
- Systeemconcept.
- Pilotplan.

6.1 *Huidige situatie*

Dit hoofdstuk beschrijft de huidige situatie en de huidige functionaliteit van de huidige werkwijze en is tevens het uitgangspunt van het te ontwikkelen product.

In de opleiding van de faculteit Bouwkunde worden in elk semester projecten (onderzoeksprojecten, onderwijsprojecten) aangeboden voor de studenten. Bij deze projecten worden de studenten begeleid door docenten. Door het rooster van de docenten kan het voorkomen, dat de studenten bij een andere docent voor hetzelfde project te woord gestaan worden. Iedere docent is verantwoordelijk voor het bijhouden van eigen uren besteding zowel de rooster uren als de extra gemaakte uren. De extra gemaakte uren kunnen besteed worden aan de werkzaamheden die buiten de rooster activiteiten vallen, zoals het nakijken van de werkstukken, en het voorbereiden van de lessen. Aan het eind van ieder maand/project worden de bestede uren doorgegeven aan de docent admin. Docent admin is een docent die verantwoordelijk is voor het bijhouden van de uren. Deze docent gebruikt een Excel bestand voor het opslaan van deze uren.

Docent admin is ook verantwoordelijk voor het opstellen en het bijhouden van het onderwijsrooster (een overzicht van de vakken die de docenten geven, wanneer, waar en voor wie). Dit onderwijsrooster wordt in Excel opgesteld. Na het eind van iedere project worden de extra uren en de roosteruren opgeteld om inzicht te krijgen over de tijdbesteding van een project. De gebruikte Excel bestanden hebben de volgende nadelen heeft:

- Niet overzichtelijk.
- Kan geen overzichten genereren.
- Heeft weinig wijzigingen mogelijkheden.
- Moeilijk te beheren.
- Er kan niet achter gehaald worden hoeveel tijd een docent aan een project besteed heeft.
- Er kan niet achter gehaald worden hoeveel precies een project tijd kost.
- Geen zoek optie (docent/project)

Vanwege voorgenoemde nadelen wil men een digitale omgeving aanbieden, waarbij gebruik gemaakt zal worden van een urenregistratiesysteem. De afstudeerder zal zorgen van het ontwikkelen en implementeren van deze applicatie (webbased-urenregistratiesysteem).

6.2 Opstellen systeemeisen

Alle systeemeisen met betrekking tot het ontwerpen en ontwikkelen van de producten worden bij een project op basis van de wensen en de eisen van de gebruiker geformuleerd. Hierbij is het van belang dat de systeemeisen concreet opgesteld moeten worden.

Bij het bepalen en opstellen van de systeemeisen heb ik de eisen en de wensen met betrekking tot het ontwikkelen van Webbased-urenregistratiesysteem als uitgangspunt gehanteerd. In het begin van de afstudeerperiode kreeg ik een lijst met geïnventariseerde algemene functionaliteiten. Deze wensen waren bepaald tijdens een workshop waar docenten en student-assistenten van de afdeling TO&I aanwezig waren. Hieronder ziet u de algemene eisen, die de eerste basis hebben gevormd voor de uiteindelijke systeemeisen;

- Het moet mogelijk zijn voor een ingelogde gebruiker zijn/haar eigen extra gemaakte uren te registreren.
- Het moet mogelijk zijn om de docenten te kunnen beheren, toevoegen/wijzigen.
- Het moet mogelijk zijn om de projecten te kunnen beheren, toevoegen/wijzigen.
- Het moet mogelijk zijn om het rooster te kunnen beheren, toevoegen/wijzigen.
- Het genereren van overzichten per docent met betrekking tot de gemaakte uren per project.

Om deze algemene wensen verder te concretiseren heb ik met mijn opdrachtgever en begeleider wekelijks gesprekken gevoerd. Uitgaande van deze algemene wensen, was het mijn doel om specifieke systeemeisen te bepalen. Tijdens deze gesprekken heb ik aangegeven dat in verband met de voortgang van het project met de bovenstaande criteria rekening gehouden moest worden.

Aanvullend op deze gesprekken heb ik aan de hand van deze eisen een paar docenten geïnterviewd. Op basis van de totale informatie die ik heb verzameld, heb ik de uiteindelijke systeemeisen bepaald en gegroepeerd. Ik heb de groepen aan de hand van het verschillende type systeemeisen ingedeeld. Deze groepen bestaan uit:

- Operationele eisen
- Functionele eisen
- Userinterface-eisen

Per categorie zijn de geïnventariseerde eisen die in mate van belangrijkheid zijn opgedeeld. De reden hiervan is om duidelijkheid te scheppen in welke volgorde de systeemeisen geïmplementeerd worden. Daarnaast geeft het ook een helder beeld van de eisen die niet opgenomen worden als het project uitloopt.

Er zijn drie niveau's te onderscheiden, namelijk:

- *Basis*
De eisen dienen te allen tijde opgenomen te worden in het systeem.
- *Comfort*
De eisen mogen geïmplementeerd worden, mits de tijdsplanning dit toestaat. Deze eisen zijn geen overbodige luxe.
- *Luxe*
Eisen die het systeem uitbreiden met luxe functionaliteit die niet noodzakelijk is voor de werking van het systeem.

De indeling van de systeemeisen over de verschillende niveaus werd tijdens de wekelijkse gesprekken bepaald.

Hieronder ziet u de systeemeisen, die de applicatie aan moet voldoen:

Functionele eisen

Dit zijn de belangrijkste functionele eisen waaraan het gehele systeem moet voldoen voor het gehele systeem eisen wordt u verwezen naar de **externe bijlage B** Definitiestudie.

Functionele eisen	Prioriteit
Het moet mogelijk zijn met behulp van een inlogpagina te kunnen inloggen. (docent, docent admin)	Basis
Het moet mogelijk zijn voor een ingelogde gebruiker om zijn/haar identiteit te specificeren door het afdrukken van de gebruikersnaam op het scherm. (docent, docent admin)	Basis
Het moet mogelijk zijn om de docenten te kunnen beheren, toevoegen/wijzigen. (docent admin)	Basis
Het moet mogelijk zijn om de projecten te kunnen beheren, toevoegen/wijzigen. (docent admin)	Basis
Het moet mogelijk zijn om het rooster te kunnen beheren, toevoegen/wijzigen. (docent admin)	Basis
Het invullen van het rooster wordt verzorgd door het importeren van het Excel bestand (rooster) naar de nieuwe applicatie. (docent admin)	Comfort
De ingeroosterde uren moeten automatisch worden geregistreerd (2 uur per dagdeel).	Basis
Het moet mogelijk zijn voor een ingelogde gebruiker zijn/haar eigen extra gemaakte uren te registreren/wijzigen. (docent)	Basis
Het genereren van overzichten per docent met betrekking tot de gemaakte uren per project/datum. (docent, docent admin)	Basis
Het genereren van overzichten per project met betrekking tot de gemaakte uren per docent/datum. (docent, docent admin)	Basis
Het moet mogelijk zijn om naar een bepaald project of projecten te zoeken met behulp van een select menu. (docent, docent admin)	Basis

Operationele eisen

Operationele eisen hebben betrekking op de eisen die waarborgen dat het systeem kan blijven draaien. De volgende eisen op operationeel gebied zijn achterhaald:

ID	Operationele eisen	Prioriteit
O1	De applicatie dient een internetapplicatie te zijn.	Basis
O2	De applicatie dient met PHP gerealiseerd te worden.	Basis
O3	Er dient gebruik gemaakt te worden van een centrale database. Hierbij wordt MySQL database gebruikt.	Basis
O4	De applicatie moet tegelijk voor meerdere gebruikers toegankelijk zijn.	Basis

User-Interface eisen

De volgende eisen hebben betrekking op de interface van de applicatie. Hiermee wordt de interactie met de gebruiker bedoeld en de eventuele uitvoer bestemd voor de gebruiker.

ID	User-Interface eisen	Prioriteit
U1	De taal die gebruikt wordt moet Nederlands zijn.	Basis
U2	De interface dient overzichtelijk te zijn.	Basis
U3	De schermen dienen consistent te zijn. Hierbij kan gedacht worden aan lettertype, de kleuren, lettergrootte, drukknoppen of hyperlinks.	Basis

6.3 Samenstellen systeemconcept

Aan de hand van de systeemeisen heb ik kunnen vastleggen hoe de gewenste situatie voor mijn applicatie eruit zou komen te zien. De stap die ik daarna heb gedaan was het samenstellen van de use-cases. Om tot de definitieve use-cases te komen heb ik de volgende activiteiten uitgevoerd.

- Bepalen actors.
- Schetsen van de senario's voor elke bepaalde actor.
- Samenstellen use-case en beschrijven van de use-cases door middel van use-case templates.
- Opstellen use-case diagram.

Een actor is een entiteit die buiten het systeem staat en die direct communiceert met het systeem. Een actor kan zowel een mens als een ander systeem zijn.

Om de actors te bepalen heb ik met mijn opdrachtgever en begeleider gesprekken gehouden. Uit deze gesprekken is gebleken dat bij mijn applicatie twee actors betrokken waren, namelijk;

- Docent.
- Docent admin.

Om een beter inzicht te krijgen over de betrokkenheid van de actors bij het systeem heb ik voor iedere actor een scenario geschetst. Hieronder worden de scenario's verder toegelicht.

Scenario voor docent:

De docent meldt zich aan bij het systeem(inlog procedure). Iedere docent krijgt zijn/haar eigen gebruikersnaam te zien op het scherm.

De docent krijgt de mogelijkheid om zijn rooster te bekijken en uit te printen.

De docent kan zijn eigen extra gemaakte uren registreren, bekijken en beheren.
De docent kan zoeken naar een project, als deze docent deel neemt aan dit project krijgt hij/zij een overzicht van zijn totale uren besteding aan dit project.
Als laatste kan een docent verschillende overzichten van zijn/haar projecten uitprinten.

Scenario voor de docent admin:

De docent admin zorgt voor het onderhouden van de applicatie en de database. Een systeemadministrator heeft alle rechten in de applicatie. Als er een nieuw project, rooster of docent geregistreerd moet worden, wordt dat door de docent admin uitgevoerd.
De docent admin is de beheerder van de uren die aan het rooster verbonden zijn of die als extra worden geregistreerd, dus de docent admin kan de uren van alle docenten wijzigen of uren voor een docent toevoegen.
Naast het beheren van de applicatie kan docent admin overzichten genereren en uitprinten van alle docenten en projecten.

Samenstellen Use-cases:

Aan de hand van de systeemeisen en de scenario's van de gebruikers heb ik de use-cases samengesteld. Een usecase is een beschrijving van een reeks van interacties tussen één of meerdere gebruikers en het systeem. Use-cases zijn een middel om de functionele systeemeisen met bijbehorende actoren gedetailleerd weer te geven.

Voor het samenstellen van use-cases heb ik gekeken naar de functionele eisen die ik bepaald heb. Alle functionele eisen heb ik gegroepeerd op basis van de scenario's. Op basis van de groepering heb ik de use-cases samengesteld. Door middel van use-case templates heb ik de use-cases beschreven.

Om een voorbeeld te geven hoe een functionele eis vertaald wordt naar een use-case, zijn er functionele eisen gekozen die uitgewerkt zijn tot een use-case template.

Functionele eisen	Prioriteit
Het moet mogelijk zijn voor een ingelogde gebruiker zijn/haar eigen extra gemaakte uren te registreren/Wijzigen. (docent)	Basis
Het moet mogelijk zijn om de geregistreerde uren te beheren toevoegen/wijzigen. (docent admin)	Basis
Het moet mogelijk zijn voor een ingelogde gebruiker zijn/haar totale gemaakte uren te bekijken. (docent)	Basis

Hieronder ziet u de usecase-template "beheren uren";

Docent admin

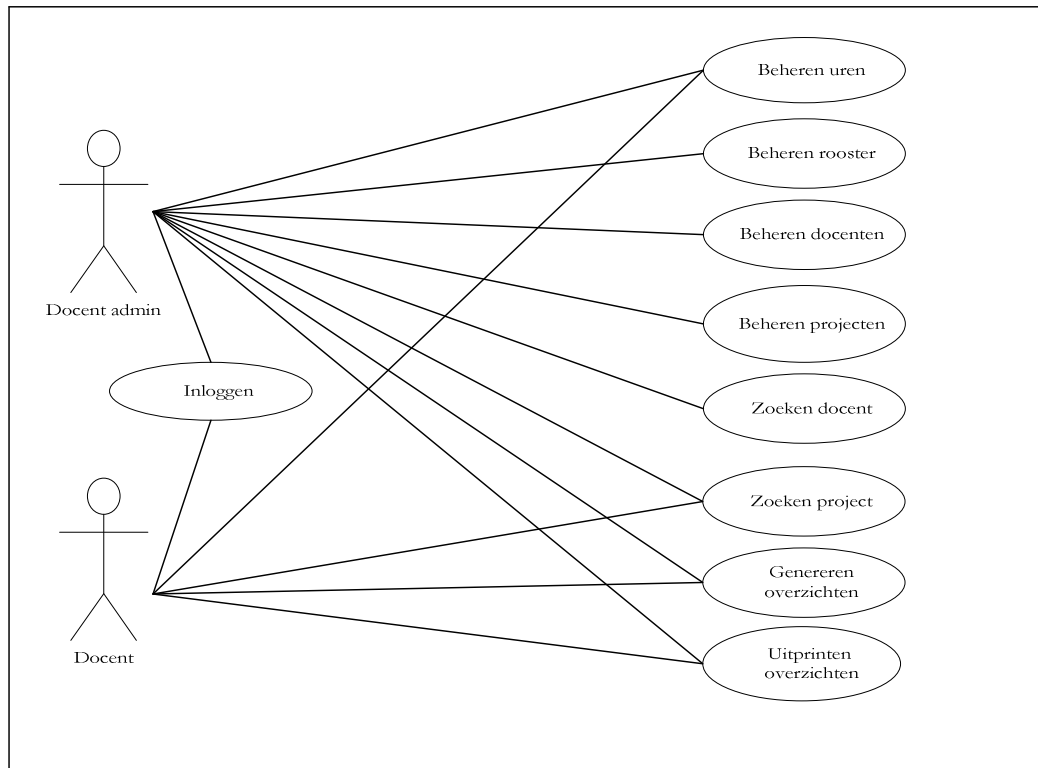
Use-case	:	Beheren uren
Actor(en)	:	Docent admin
Aanname(n)	:	De docent admin is ingelogd in het systeem en wil de uren van een docent toevoegen of wijzigen.
Beschrijving	:	(1) Een ingelogde gebruiker (docent admin) vraagt de uren van een docent aan. (2) De applicatie controleert of de betreffende docent al persoonlijk uren heeft. Indien deze docent al uren heeft, wordt dat ook op het scherm getoond. (3) De docent admin krijgt de mogelijkheid om de uren van alle gebruikers te wijzigen. Bij het wijzigen van de uren worden er de oude uren vervangen. (4) Als de uren worden toegevoegd, is er een mogelijkheid om deze uren te wijzigen.
Uitzondering	:	Er wordt gecontroleerd of de ingevulde waarde correct is of het aantal uren groter is dan 10. Bij beide gevallen wordt er een foutmelding gegeven.
Resultaat	:	• De uren zijn toegevoegd. • De uren zijn gewijzigd.

Docent

Use-case	:	Beheren uren
Actor(en)	:	Docent,
Aanname(n)	:	De docent is ingelogd in het systeem en wil zijn/haar eigen persoonlijke uren toevoegen of wijzigen.
Beschrijving	:	(1) Een ingelogde gebruiker (docent) vraagt zijn uren aan. (2) De applicatie controleert of de gebruiker al persoonlijk uren heeft. Indien de gebruiker al uren heeft, wordt dat ook op het scherm getoond. (3) De gebruiker krijgt een mogelijkheid om zijn uren te wijzigen. Zodra de wijzigingen worden aangebracht, krijgt de gebruiker de mogelijkheid om deze uren aan zijn/haar account toe te voegen. Bij het wijzigen van de uren worden er de oude uren vervangen. (4) Als de uren worden toegevoegd, is er een mogelijkheid om deze uren te wijzigen
Uitzondering	:	Er wordt gecontroleerd of de ingevulde waarde correct is of het aantal uren groter is dan 10. Bij beide gevallen wordt er een foutmelding gegeven.
Resultaat	:	• De uren zijn toegevoegd. • De uren zijn gewijzigd.

Voor alle use-case templates wordt u naar **externe bijlage B** Definitiestudie verwezen.

Om een duidelijkheid te verschaffen over de hele applicatie zijn alle use-cases met de bijbehorende actors opgenomen in een diagram, de use-case diagram. Hieronder ziet u het uiteindelijke use-case diagram.



Figuur 6. Use Case diagram

6.4 Opstellen pilotplan

Het pilotplan bestaat uit een geprioriteerde lijst van de pilots die ontwikkeld en ingevoerd zullen worden. Een pilot is een coherente subset van het uiteindelijke beoogde informatiesysteem die als zelfstandige, bruikbare eenheid kan worden ingezet in de organisatie. Hierbij heeft elke pilot betrekking op een bepaald deel van het ontworpen systeemconcept. Voor elke pilot wordt een inschatting van de benodigde tijd en middelen gemaakt. Het pilotplan fungeert als leidraad voor het ontwikkeltraject.

Aan de hand van de systeemeisen en use-cases heb ik te ontwikkelen onderdelen van webbased-urenregistratiesysteem in de eerste instantie in 4 pilots verdeeld. Het betreft de volgende pilots;

1. Database.
2. User-interface
3. Administratiesysteem.
4. Overzichten.

Aangezien het **webbased-urenregistratiesysteem** op een database moet werken en de andere pilots hier afhankelijk van zijn, heb ik die als eerste pilot genomen. Er zal gebruik gemaakt worden van MySQL database om de applicatie uit te breiden.

Tijdens het bepalen van de pilots is naar voeren gekomen dat het niet zinvol zou zijn om de user-interface als aparte pilot te definiëren. Het bepalen van alle user-interfaces van de gehele applicatie in een vroeg stadium kan leiden tot het itereren van de een bepaalde user-interface, wat

meer tijd in beslag neemt dan is gepland. Om de tijdverspilling te vermijden heb ik besloten om de user-interface over drie-en te verdelen.

Voor de andere pilots zijn de systeemeisen gecategoriseerd en gegroepeerd. Voor elke pilot zijn van tevoren de gerelateerde systeemeisen, het doel, relatie met andere pilots, planning en prioriteit vastgesteld. Dit is gedaan om vooraf de belangrijkste punten voor een pilot vast te stellen. Ten slotte zijn tijdens deze fase de volgende pilots opgesteld;

Pilot(s)
1. Database
2. Administratie
3. Overzichten

Hieronder worden de pilots gedefinieerd;

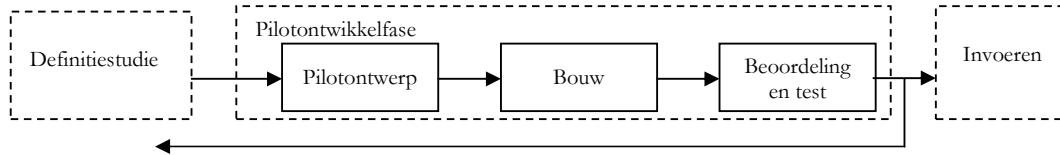
Pilot 1	:	Database
Doel van de pilot	:	Ontwerpen en realiseren van de database.
Relatie met andere pilots	:	Pilot 2, pilot 3
Planning	:	2 weken
Delen van de pilot	:	
Prioriteit	:	Basis

Pilot 2	:	Administratiesysteem
Doel van de pilot	:	Het ontwikkelen en implementeren van een administratiesysteem voor de gerelateerde systeemeisen.
Relatie met andere pilots	:	Pilot 1, pilot 3
Planning	:	4 weken
Delen van de pilot	:	• User interfaces. • Inlogpagina. • Beheersysteem voor de docenten • Beheersysteem voor de projecten • Beheersysteem voor het rooster • Beheersysteem voor de geregistreerde uren.
Prioriteit	:	Basis

Pilot 3	:	Overzichten
Doel van de pilot	:	Het ontwikkelen en implementeren van interfaces voor de overzichten die de gebruikers willen genereren.
Relatie met andere pilots	:	Pilot 1, pilot 2
Planning	:	4 weken
Delen van de pilot	:	• User interfaces. • Overzichten voor het rooster. • Overzichten van de geregistreerde uren.
Prioriteit	:	Basis

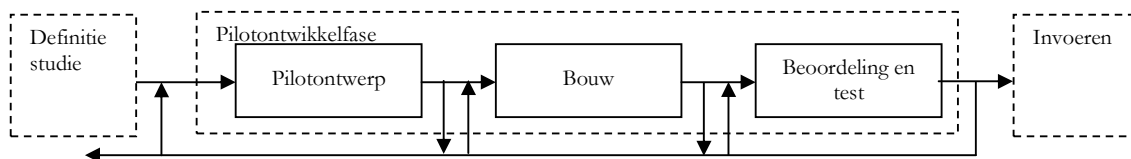
6.5 Iteratiestrategie pilotontwikkeling

Net als dat ik bepaald heb welke strategie ik zou gebruiken voor het project (big-bang), vond ik het ook nuttig een iteratiestrategie te kiezen voor de fase pilotontwikkeling. De situatie volgens big-bang was als volgt:



Figuur 7. Big-bang iteratiestrategie

Om ervoor te zorgen dat na elke stap de mogelijkheid bestond dat ik kon itereren zonder dat ik de gehele fase hoefde mee te nemen, heb ik besloten een iteratiestrategie voor de pilotontwikkelfase te bepalen.



Figuur 8. Big-bang iteratiestrategie + volledige iteratie pilotontwikkeling

6.6 Iteratie

Tijdens de ontwikkelfase van de pilot 2 administratie zijn er nog een paar eisen bijgekomen. Hierbij gaat het om 'het zoeken naar projectoverzichten' en 'het zoeken naar enkele docentoverzichten'. Hierdoor zijn er twee nieuwe functionele eisen en use-cases toegevoegd aan de definitiestudie. Hierbij gaat het om de volgende eisen;

Functionele eisen	Prioriteit
Het moet mogelijk zijn om naar een bepaald project of projecten te zoeken met behulp van een select menu die alle projecten bevat.	Basis
Het moet mogelijk zijn om naar een bepaalde docent of docenten te zoeken met behulp van een select menu die alle docenten bevat.	Basis

Hieronder worden de nieuwe use-cases weergegeven;

Use-case	:	Zoeken project
Actor(en)	:	Docent, docent admin
Aanname(n)	:	Docent is ingelogd in het systeem en wil naar projectoverzichten zoeken.
Beschrijving	:	(1)Gebruiker opent de zoekpagina. (2) Gebruiker kiest een zoekterm om naar een project te zoeken. (3)Vervolgens worden de zoekresultaten op het scherm getoond. De gebruiker ziet de projectnaam, de naam van docenten die deel nemen aan dit project aantal urenbesteding van iedere docent, en uiteindelijk de totale urenbesteding van het gehele project.
Uitzondering	:	Als voor dit project nog geen uren worden geregistreerd, wordt er overzicht getoond waarvan de totale urenbesteding gelijk is aan nul.
Resultaat	:	Gevonden resultaten worden op het scherm getoond.

Use-case	:	Zoeken docent
Actor(en)	:	Docent admin.
Aanname(n)	:	Docent admin is ingelogd in het systeem en wil naar een docent zoeken.
Beschrijving	:	(1)Docent admin krijgt de zoekpagina te zien. (2)Docent admin kiest een zoekterm om naar een docent te zoeken. (3)Vervolgens worden de zoekresultaten op het scherm getoond. De docent admin ziet de naam van de docent, en aan welke project deel heeft genomen, en hoeveel uren aan iedere project heeft besteedt, en uiteindelijk de totale urenbesteding van deze docent.
Uitzondering	:	Als voor deze nog geen uren worden geregistreerd, wordt er overzicht getoond waarvan de totale urenbesteding gelijk is aan nul.
Resultaat	:	Gevonden resultaten worden op het scherm getoond.

Door deze iteratie is het use-case diagram ook aangepast.

7 Pilot 1 – Database TO&I DB-UR (Database Urenregistratie)

De eerste pilot in de ontwikkelfase werd gevormd door het realiseren van een database structuur voor de applicatie. De database is de meest essentiële pilot, omdat de basis vormt voor de hierna komende pilots; het administratiesysteem en de overzichten. In dit hoofdstuk wordt toegelicht hoe de geschikte database tot stand gekomen is.

7.1 Eerste opzet

Aangezien op de afdeling TO&I gebruik wordt gemaakt van MySQL database, werd mij in het begin van het afstudeerproject duidelijk dat de te ontwikkelen applicatie van MySQL database gebruik dient te maken. Daarom had ik geen keuze om een andere DBMS te kiezen. Hierbij wordt MySQL als DBMS gebruikt.

Een andere motivatie om MySQL te kiezen was dat ook binnen TO&I kennis hiervan aanwezig was.

Om deze keuze te bevestigen heb ik op het Internet gezocht naar de beste combinatie met PHP (omdat PHP voor scripting zal gebruikt worden) gezocht. Uit dat onderzoek kwam duidelijk naar voren dat PHP en MySQL één van de populairste combinaties is.

Om een duidelijk beeld te krijgen over hoe de database eruit komt te zien en wat voor gegevens worden er opgeslagen, heb ik de betrokken partijen geïnterviewd. Na dat interview heb ik de tabellen bepaald en ben ik begonnen met het modelleren van de database omdat het een duidelijker en overzichtelijker idee geeft. Voor het modelleren van de database had ik twee manieren in mijn gedachten. De eerste manier was om de database met een klassendiagram te ontwerpen. De tweede manier is het ERD (entity relation diagram). Om de database te ontwerpen heb ik het ERD gekozen omdat ik tijdens mijn opleiding in verschillende projecten ERD techniek had toegepast en dat was me altijd goed bevallen. Het ERD wordt in de komende paragraaf gedetailleerd weergegeven.

In de definitiestudie zijn een aantal systeemeisen opgesteld die betrekking hebben op de database. De eerste stap bij deze pilot was het verzamelen van de systeemeisen die specifiek aan de database zijn gesteld en per systeemeis een beschrijving geven van een mogelijke oplossing om die eis te kunnen implementeren in het product. Hieronder geef ik de specifieke eisen voor de database.

7.2 Specificatie van de database

Na het verzamelen van de systeemeisen wordt de specificatie en de afbakening van de database bepaald. Hieronder zijn de belangrijke functionele eisen weer gegeven waaraan deze pilot moet voldoen.

Functionele eisen	Prioriteit
Het moet mogelijk zijn met behulp van een inlogpagina te kunnen inloggen. (docent,docent admin)	Basis
Het moet mogelijk zijn voor een ingelogde gebruiker om zijn/haar identiteit te specificeren door het afdrukken van de gebruikersnaam op het scherm. (docent,docent admin)	Basis
Het moet mogelijk zijn om de docenten te kunnen beheren. (docent admin)	Basis
Het moet mogelijk zijn om de projecten te kunnen beheren. (docent admin)	Basis
Het moet mogelijk zijn om het rooster te kunnen beheren. (docent admin)	Basis
De ingeroosterde uren moeten automatisch worden geregistreerd(2 uur per dagdeel). (docent)	Basis
Het moet mogelijk zijn voor een ingelogde gebruiker zijn/haar eigen extra gemaakte uren te registreren. (docent)	Basis
Het moet mogelijk zijn om deze uren te beheren toevoegen/wijzigen. (docent,docent admin)	Basis
Het genereren van overzichten per docent met betrekking tot de gemaakte uren per week/maand/jaar. (docent,docent admin)	Basis
Het moet mogelijk zijn om naar een bepaalde docent of docenten te zoeken met behulp van een select menu. (docent admin)	Basis

Na het analyseren en groeperen van de bovengenoemde eisen ben ik na een overleg met mijn begeleider aan de volgende entiteitstypen gekomen. Hierbij gaat het om de volgende entiteitstypen:

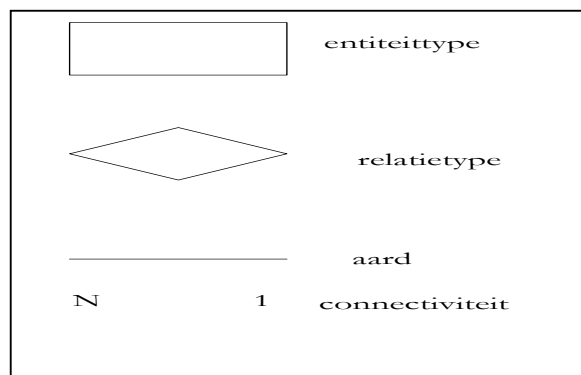
Hieronder zijn de gebruikte entiteitstypen en eigenschappen weergegeven.

Entiteitstypen	Eigenschap
Docentfull	Tabel docentfull bevat de personen. Hier worden de gebruikersgegevens geregistreerd. Hierin is de gebruiker met zijn persoonlijke ID en gebruikersnaam bekend. Deze tabel bevat aantal attributen, deze attributen zijn bij voorbeeld; Voornaam, achternaam, email, telefoon....
Project	Tabel project bevat de projectgegevens. Hier worden de projectgegevens geregistreerd. Deze tabel bevat aantal attributen, deze attributen zijn bij voorbeeld; Projectnaam, begindatum, einddatum....
Rooster	Tabel Rooster bevat de roostergegevens. Hier wordt een rooster opgesteld, daarna wordt dit rooster aan een docent gekoppeld. Als een docent aan een rooster wordt gekoppeld worden de geroosterde uren automatisch geregistreerd in het account van deze docent (twee uur per dagdeel). Deze tabel bevat aantal attributen, deze attributen zijn bij voorbeeld; Datum, ruimte, vcode....

Uren-reg	Tabel Uren-reg bevat de urengegevens. Hier worden de uren van de docenten geregistreerd. Hierbij kan gedacht worden aan twee soorten registraties, de eerste de roosteruren die automatisch worden geregistreerd op het moment dat een docent aan een rooster wordt gekoppeld, de tweede de extra gemaakte uren die worden besteed aan een project. Hier moet de gebruiker zelf zijn eigen uren registreren/beheren. Deze tabel bevat aantal attributen, deze attributen zijn bij voorbeeld; Datum, aantaluren,
----------	--

Het ERD (Entity relation Diagram)

ERD (Entity Relation Diagram) is een diagram dat de relatie tussen de verschillende entiteiten weergeeft. De entiteiten zijn getekend in een rechthoek. De relaties van de verschillende entiteiten zijn aan elkaar gekoppeld met een associatietype. Hierin wordt met een werkwoord aangegeven hoe de entiteiten aan elkaar gekoppeld zijn.



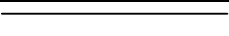
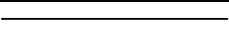
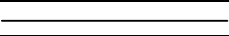
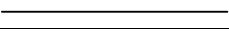
Figuur 9. Legenda ERD

Uitleg relaties

Een relatietype bestaat uit 2 associatietypen. Die geven specifiek aan wat het verband, en het omgekeerde verband, tussen de entiteitstypen is.

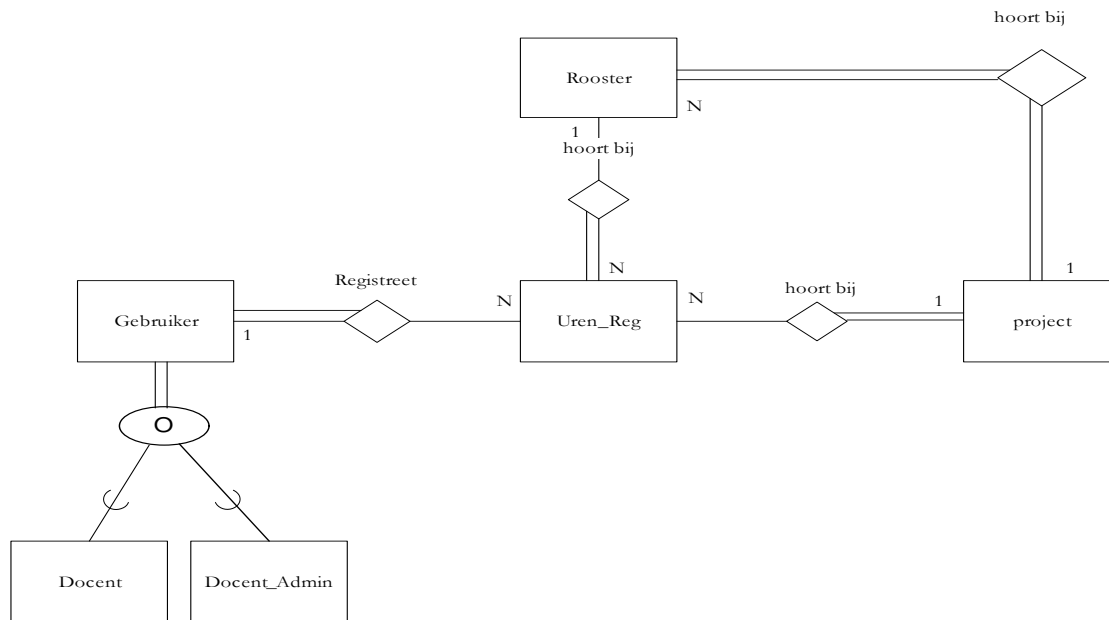
Belangrijke kenmerken van associatietypen zijn de *aard* en de *connectiviteit*. De aard van een associatietype geeft aan of het verplicht (dubbele streep) is of niet verplicht (optioneel, enkele streep). De connectiviteit van een associatietype geeft aan of er maximaal 1 of juist meer entiteiten bij een bepaalde entiteit aan de 'andere kant' kunnen horen.

De volgende tabel licht dit toe.

 1	precies één
 N	minstens één, tenminste één
 1	hoogstens één; nul of één, kan een ...
 N	nul of meer, kan een aantal ...,

Figuur 10 Relatietypes

Hieronder wordt het ERD van de gebruikte database weergegeven;



Figuur 11 Het ERD

7.3 Relatieve representatie model

De volgende stap die ik heb gedaan was het vertalen van het ER-model naar een relationeel representatie model. Dit model geeft precies weer, wat de relaties zullen zijn in de database. Tevens geeft het model precies aan wat de verbanden tussen de relaties zijn door middel van de vreemde sleutels. Ook wordt aangegeven of de vreemde sleutels wel of niet null-waarden mogen bevatten.

Voor de opzet van het model is gebruik gemaakt van het DB-01 moduleboek en het boek “Databasesystemen in de praktijk” van Vandenbulcke.

Het relationeel representatie model is een tussenstap tussen het ER-model en het relationeel implementatiemodel.

Het relationeel representatie model

DOCENT (docent_id, naam, achternaam, straat, postcode, plaats, telefoon, mobile, geboortedatum, studienr, e-mail, homepage, bank/giro, woonplaats, functie, wachtwoord, is_admin)

Primary key (docent_id) null-waarden niet toegestaan

PROJECT (project_id, project_naam, begin_datum, eind_datum, vcode)

Primary key (project_id) null-waarden niet toegestaan.

ROOSTER (**rooster_id**, datum, dagdeel, ruimte, project_id)

Primary key (roostert_id) null-waarden niet toegestaan

Foreign key (project_id) refereert aan project_id in PROJECT null-waarden niet toegestaan

UREN_REG (**uren_reg_id**, docent_id, project_id, rooster_id, datum, aant_uren, commentaar)

Primary key (uren_reg_id) null-waarden niet toegestaan

Foreign key (docent_id) refereert aan docent_id in DOCENT null-waarden niet toegestaan

Foreign key (project_id) refereert aan project_id in PROJECT null-waarden niet toegestaan

Foreign key (rooster_id) refereert aan rooster_id in PROJECT null-waarden wel toegestaan

7.4 Realiseren database

Het implementatiemodel is bedoeld als script voor invoering van de fysieke database geïmplementeerd. Om voorbeeld te geven zijn hieronder een paar SQL scripts weergegeven. Deze scripts kan men op de database uitvoeren en hiermee worden de tabellen aan gemaakt.

Tijdens het vertalen van het relationeel representatie model naar een implementatiemodel heb ik de vreemde sleutels niet kunnen definiëren aangezien het niet mogelijk is om bij MySQL vreemde sleutels te definiëren. Dit wordt tijdens implementatie mogelijk gemaakt.

Hieronder ziet u het implementatiemodel voor de tabel “Uren-reg”;

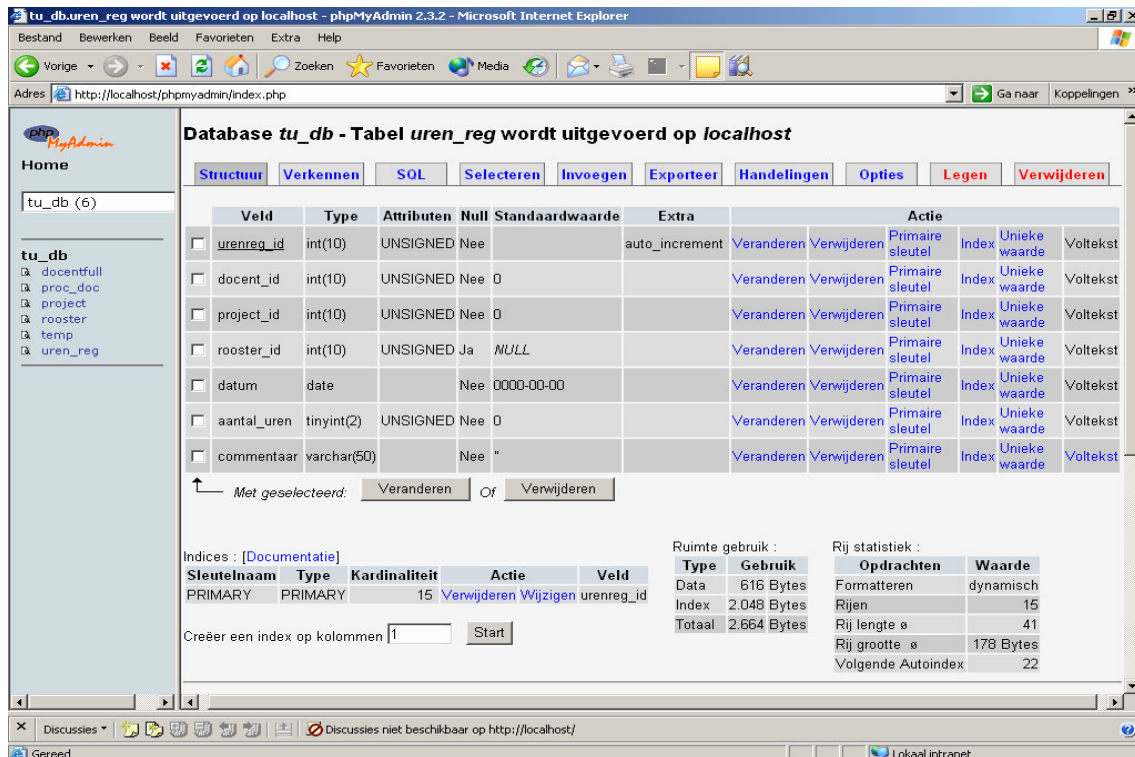
CREATE TABLE `uren\_reg` (

```
`uren_reg_id` int(10) unsigned NOT NULL auto_increment,  
`docent_id` int(10) unsigned NOT NULL default '0',  
`project_id` int(10) unsigned NOT NULL default '0',  
`rooster_id` int(10) unsigned default NULL,  
`datum` date NOT NULL default '0000-00-00',  
`aantal_uren` tinyint(2) unsigned NOT NULL default '0',  
`commentaar` varchar(50) NOT NULL default ''
```

PRIMARY KEY (`uren\_reg\_id`)

) TYPE=MyISAM AUTO_INCREMENT=1 ;

Ter illustratie wordt de structuur van de gecreëerde tabel “**uren_reg**” in de MySQL database weergegeven. Alle andere tabellen zijn op deze manier gecreëerd.



Figuur 12. Structuur van de tabel “uren_reg”

Voor het invullen van de tabel **docentfull** heb ik gebruik gemaakt van een Excel bestand, dit bestand bevat de personeelgegevens. Om de rest van de tabellen te kunnen testen heb ik zelf voor de invulling gezorgd.

Na het importeren van het Excel bestand personeellijst heb ik de inhoud van de tabel **docentfull** getest met deze SQL query

```
” SELECT *
FROM `docentfull`
ORDER BY naam
LIMIT 0, 100.”
```

Hierbij kan ik het resultaat van de query niet weergegeven aangezien het privé informatie bevat van de medewerkers van TO&I.

Voor het hele implementatiemodel wordt u naar **externe bijlage C** Pilotontwerpproject database verwezen.

8 Pilot 2 – Realiseren administratiesysteem

Na het afronden van de pilot database ben ik gestart met de pilot administratiesysteem. In dit hoofdstuk wordt het realiseren van de pilot administratiesysteem beschreven. De pilot is administratiesysteem genoemd aangezien de functionele eisen die betrekking hebben op het administratie gedeelte, in deze pilot gerealiseerd worden.

Deze pilot bevat de delen voor de docent admin en de docenten. Het Administratiesysteem is een product dat de gebruikers de mogelijkheid geeft zich te identificeren. Hierbij is er de mogelijkheid om persoonlijke uren aan het gebruikersaccount toe te voegen. Bovendien geeft het systeem de docent admin de mogelijkheid om de docenten, projecten, rooster en uren bij te houden en te beheren voor verschillende opgaven. Tevens is er de mogelijkheid om naar een project en naar een docent te zoeken ten behoeve van de uren registratie.

Tijdens het realiseren van deze pilot heb ik me met de volgende pilotdelen beziggehouden. Hieronder staan de pilotdelen die in het pilotplan bepaald zijn. (zie definitiestudie).

- User-interface.
- Inloggen.
- Beheersysteem voor docenten.
- Beheersysteem voor projecten.
- Beheersysteem voor rooster.
- Beheersysteem voor uren.
- Zoeken naar projecten en docenten.

Om deze pilotdelen te realiseren heb ik de volgende activiteiten verricht;

- Opstellen plan van aanpak.
- Ontwerpen en ontwikkelen user-interface.
- Ontwerpen en ontwikkelen functionaliteit inloggen.
- Ontwerpen en ontwikkelen functionaliteit het beheersysteem voor docenten.
- Ontwerpen en ontwikkelen functionaliteit het beheersysteem voor projecten.
- Ontwerpen en ontwikkelen functionaliteit het beheersysteem voor het rooster.
- Ontwerpen en ontwikkelen functionaliteit het beheersysteem voor de uren.
- Ontwerpen en ontwikkelen functionaliteit voor zoekpagina projecten.
- Ontwerpen en ontwikkelen functionaliteit voor zoekpagina docenten.

8.1 Opstellen plan van aanpak

Het doel van deze activiteit is duidelijke afspraken vast te stellen over inhoud en omvang van de op te leveren producten en daarvoor uit te voeren werkzaamheden. In het plan van aanpak van pilot 2 heb ik in het kort het doel en een planning beschreven. Dit plan van aanpak heeft als overzicht gefunctioneerd tijdens het ontwikkelen van deze pilot.

Tijdsduur:

De tijdsduur voor het ontwikkelen van pilot 2 is 4 weken.

Time-boxing:

Bouweenheid	Prioriteit	Kosten
Ontwerpen en ontwikkelen user-interfaces.	Basis	2 dagen
Ontwerpen en ontwikkelen functionaliteit inloggen.	Basis	1 dag
Ontwerpen en ontwikkelen functionaliteit het beheersysteem voor docenten.	Basis	2 dagen
Ontwerpen en ontwikkelen functionaliteit het beheersysteem voor projecten.	Basis	3 dagen
Ontwerpen en ontwikkelen functionaliteit het beheersysteem voor rooster.	Basis	4 dagen
Ontwerpen en ontwikkelen functionaliteit het beheersysteem voor uren.	Basis	5 dagen
Ontwerpen en ontwikkelen functionaliteit voor zoekpagina projecten.	Basis	2 dagen
Ontwerpen en ontwikkelen functionaliteit voor zoekpagina studenten.	Basis	1 dag

Timeboxing is een projectmanagementtechniek die met succes gebruikt kan worden om de iteratieve ontwikkelcyclus te beheersen. Het uitgangspunt van timeboxing is dat de benodigde tijd en opleverdatum van een pilot wordt gefixeerd. De inhoud van de pilots wordt ingedeeld in kleine eenheden. Die eenheden worden vervolgens gecategoriseerd en gesorteerd op prioriteit, waarna ze een voor een ontwikkeld worden.

8.2 Ontwerpen en ontwikkelen user-interface

In deze paragraaf wordt het pilotdeel “het ontwerp van de user-interface” van de pilot administratiesysteem beschreven. Dit pilotdeel is opgesteld omdat het nog niet duidelijk was hoe de user-interfaces van deze pilot eruit moest zien. Het doel van dit pilotdeel is het bepalen van de user-interface voor de pilot administratiesysteem.

Allereerst heb ik de systeemeisen die voor dit pilotdeel relevant zijn, verzameld. Het gaat om de volgende eisen:

ID	User-Interface eisen	Prioriteit
U1	De taal die gebruikt wordt moet Nederlands zijn.	Basis
U2	De interface dient overzichtelijk te zijn.	Basis
U3	De schermen dienen consistent te zijn. Hierbij kan gedacht worden aan lettertype, de kleuren, lettergrootte, drukknoppen of hyperlinks.	Basis

Om tot een geschikte user-interface te komen heb ik me de volgende twee vragen gesteld;

- Wat krijgt de gebruiker te zien.
- Hoe krijgt de gebruiker de informatie te zien.

Wat krijgt de gebruiker te zien

Om te bepalen welke en hoeveel informatie de gebruiker te zien zal krijgen, heb ik de use-cases uit het rapport definitiestudie als uitgangspunt genomen. Hiernaast heb ik ook voor deze pilot relevante eisen geïnventariseerd om een beeld te krijgen over de uiteindelijke vorm van user-interface. Bij deze use-cases wordt beschreven welke acties de actoren kunnen uitvoeren. Nadat ik de use-cases uitvoerig heb bekeken, heb ik de volgende acties bepaald. De acties zijn als volgt;

Acties binnen use-cases

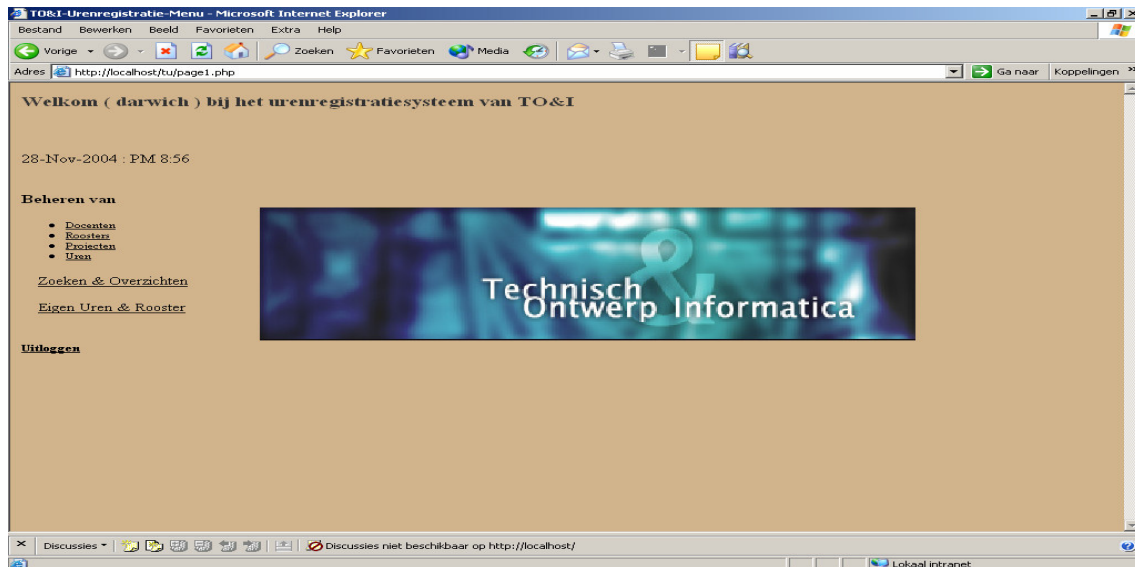
Docent admin
+ Inloggen()
+ docentenlist()
+ toevoegen docent()
+ wijzigen docent()
+ projectlist()
+ toevoegen project()
+ wijzigen project()
+ roosterlist()
+ toevoegen rooster()
+ wijzigen rooster()
+ urenlist()
+ toevoegen uren()
+ wijzigen uren()
+ zoeken project()
+ zoeken docent()

Docent
+ Inloggen()
+ Eigen urenlist()
+ toevoegen eigen uren()
+ wijzigen eigen uren()
+ zoeken eigen project()

De acties die ik heb bepaald, hebben mij een idee gegeven welke functionaliteiten de schermen aan de gebruiker moesten bieden. Omdat ik in aantal functionaliteiten heb vastgesteld, heb ik besloten voor iedere actie een scherm te ontwerpen om deze functionaliteiten mogelijk te maken. Nadat ik een globale schets van deze schermen gemaakt heb, heb ik dit met mijn opdrachtgever en begeleider bekeken. Hieruit is gebleken dat er in ieder geval eerst een optie geselecteerd moest worden en daarna de bovengenoemde acties uitgevoerd moesten worden.

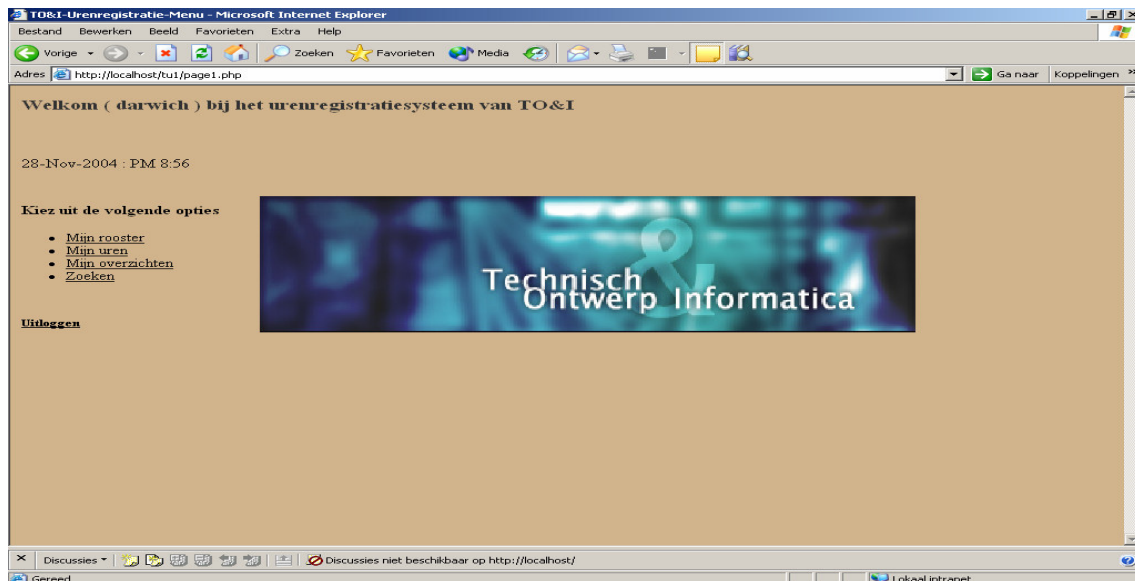
De schermen moeten aan elkaar gekoppeld worden door middel van hyperlinks. Bovendien moeten alle schermen gekoppeld worden aan de inlogpagina. Om gebruik te maken van de functionaliteiten moet de gebruiker altijd inloggen met behulp van inlogpagina.

Hierdoor zijn er nog twee schermen bijgekomen om de programma's te selecteren een voor de beheerder (docent admin) en een voor de docenten. Om dit aan te tonen heb ik hieronder screenshots opgenomen.



Figuur 13. Het hoofdscherm om de applicatie te beheren

Zoals het in figuur 13 is te zien dat de gebruikersnaam, de tijd en de datum zijn getoond. Als docent admin een actie wil uitvoeren, dient hij eerst een optie te kiezen. Als docent admin de optie 'Eigen uren & Rooster' kiest, wordt hij door geschakeld naar het hoofdscherm van de docenten. Om dit aan te tonen heb ik hieronder een screenshot opgenomen.



Figuur 14. Het hoofdscherm voor de docenten

Nadat ik met mijn opdrachtgever en begeleider over het ontwerp van de schermen tot een overeenstemming gekomen ben, heb ik me verdiept in de volgende vraag: hoe krijgt de gebruiker de informatie te zien?

Hoe krijgt de gebruiker de informatie te zien

De manier waarop informatie wordt gepresenteerd aan de gebruiker is een ander aspect voor de user interface. Hierbij gaat het om de lay-out waarbij het vooral om de aspecten als kleur, lettergrootte, drukknoppen of hyperlinks aan orde komen.

Met betrekking tot de user-interface waren er aantal eisen gesteld. Deze user-interface eisen heb ik in het begin van de paragraaf aangegeven.(zie paragraaf 8.2). Eén van de eisen is als volgt;

ID	User-Interface eisen	Prioriteit
U3	De schermen dienen consistent te zijn. Hierbij kan gedacht worden aan lettertype, de kleuren, lettergrootte, drukknoppen of hyperlinks.	Basis

Om de user-interfaces consistent te ontwikkelen heb ik bij alle user-interfaces standaard kleur, lettertype en lettergrootte gehanteerd. Ik heb van standaard knopen gebruik gemaakt.



Figuur 15. Drukknoppen

Naar aanleiding van het ontwerp user-interface heb ik deze interfaces ontwikkeld. De user-interfaces zijn in HTML ontwikkeld met daarin statische gegevens. De verschillende HTML pagina's zijn aan elkaar gekoppeld door middel van hyperlinks.

8.3 Ontwerpen functionaliteit

Voordat de applicatie ontwikkeld wordt, is het belangrijk om te weten welke functionaliteiten de applicatie moet hebben. In hoofdstuk 6.2 van het verslag heb ik aangegeven welke functionele eisen gesteld worden aan de applicatie. Als eerste heb ik de functionele eisen die betrekking hebben op deze pilot verzameld en onderverdeeld in belangrijkheidsgraad. De onderstaande systeemeisen hebben betrekking tot de functionaliteit van deze pilot. De prioriteitverdeling is gedaan op basis van de gebruikte verdeling in de definitiestudie. Dat was met de opdrachtgever bepaald. Hieronder geef ik de functionele eisen weer.

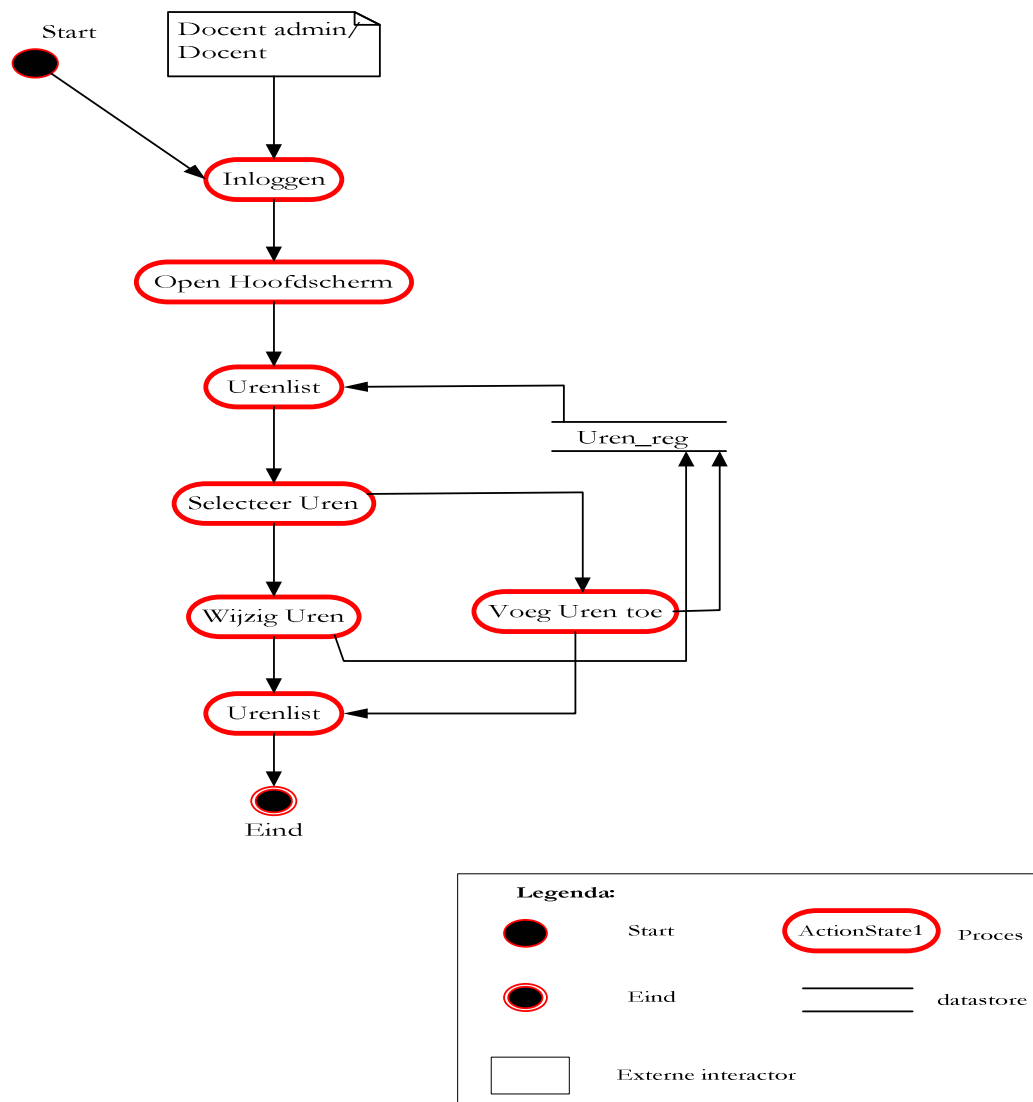
Functionele eisen	Prioriteit
Het moet mogelijk zijn met behulp van een inlogpagina te kunnen inloggen. (docent,docent admin)	Basis
Het moet mogelijk zijn om de docenten te kunnen beheren, toevoegen/wijzigen. (docent admin)	Basis
Het moet mogelijk zijn om de projecten te kunnen beheren, toevoegen/wijzigen. (docent admin)	Basis
Het moet mogelijk zijn om het rooster te kunnen beheren, toevoegen/wijzigen. (docent admin)	Basis
De ingeroosterde uren moeten automatisch worden geregistreerd(2 uur per dagdeel). (docent)	Basis
Het moet mogelijk zijn voor een ingelogde gebruiker zijn/haar eigen extra gemaakte uren te registreren. (docent)	Basis
Het moet mogelijk zijn om deze uren te beheren toevoegen/wijzigen. (docent,docent admin)	Basis
Het moet mogelijk zijn om naar een bepaald project of projecten te zoeken met behulp van een select menu. (docent,docent admin)	Basis
Het moet mogelijk zijn om naar een bepaalde docent of docenten te zoeken met behulp van een select menu. (docent admin)	Basis

Nadat ik de eisen verzameld heb, ben ik begonnen met het ontwerp. Hiervoor heb ik de volgende diagrammen gemaakt, namelijk:

- Activity diagram.
- Sequence diagram.

Activity Diagram

Een activity diagram is een functioneel model en heeft het voordeel dat het een gecombineerde kijk geeft op de processen en de gegevens. Hiermee is een duidelijk beeld ontstaan van de gewenste situatie. Om aan te tonen hoe een activity diagram eruit ziet, is hieronder een voorbeeld gegeven van “Beheren uren” van use case “Beheren uren”.



Figuur 16. Activity diagram “beheren uren”

Sequence diagram

De belangrijke informatie wordt op het scherm getoond, sommige functies verlopen echter achter de schermen, de gebruiker zal zich niet bewust zijn van deze functies. Echter, voor de ontwikkelaar is het van cruciaal belang om te weten hoe de informatiestromen achter de schermen verlopen. Om de informatiestromen aan te geven worden de sequence diagrammen binnen UML gemaakt.

Een sequence diagram wordt gebruikt om een interactie tussen een aantal objecten weer te geven, gerelateerd aan de tijd. Het sequence diagram wordt weergegeven aan de hand van objecten die horizontaal gerangschikt zijn, hun tijdslijn die verticaal is aangegeven en de interacties tussen de objecten. Op de verticale as wordt het verloop in tijd weergegeven, waarbij de tijd verloopt van boven naar beneden. Bij sequence diagrammen lopen de informatiestromen tussen de objecten. De meeste gemeenschappelijke van deze objecten zijn de gebruikers, applicatie, opslag objecten zoals database, tekstbestand of geheugen object.

Voordat ik de sequence diagrammen gemaakt heb, heb ik de acties en de objecten bepaald. De acties staan in hoofdstuk 8.1 beschreven. Voor duidelijkheid worden de acties hieronder nogmaals weergegeven.

Docent admin
+ Inloggen()
+ docentenlist()
+ toevoegen docent()
+ wijzigen docent()
+ projectlist()
+ toevoegen project()
+ wijzigen project()
+ roosterlist()
+ toevoegen rooster()
+ wijzigen rooster()
+ urenlist()
+ toevoegen uren()
+ wijzigen uren()
+ zoeken project()
+ zoeken docent()

Docent
+ Inloggen()
+ eigen urenlist()
+ toevoegen eigen uren()
+ wijzigen eigen uren()
+ zoeken eigen project()

Hierna heb ik de objecten bepaald. Namelijk;

- Gebruiker.
- Webbased-urenregistratiesysteem
- Database.

Hieronder worden de objecten gedetailleerd beschreven;

Objecten	Beschrijving
Gebruiker	De eindgebruiker is de gebruiker die direct met de applicatie communiceert. In dit geval kan de gebruiker zowel docent als docent admin zijn.
Webbased-urenregistratiesysteem	De applicatie die uiteindelijk ontwikkeld zal worden. De applicatie en de user-interface's zijn als één object gedefinieerd. De applicatie bevat aantal files die opgeslagen zijn in de server. In deze files zijn PHP functies opgeslagen, deze functies zorgen voor de communicatie tussen de GUI's en de database.
Database	Dit is het opslag object, waar uiteindelijk de gegevens in worden opgeslagen en/of uit worden opgevraagd. Dit is het resultaat van pilot 1.

Voor de namen van de acties die de objecten webbased-urenregistratiesysteem en Database uitvoeren, is een bepaalde naamstructuur vastgelegd.

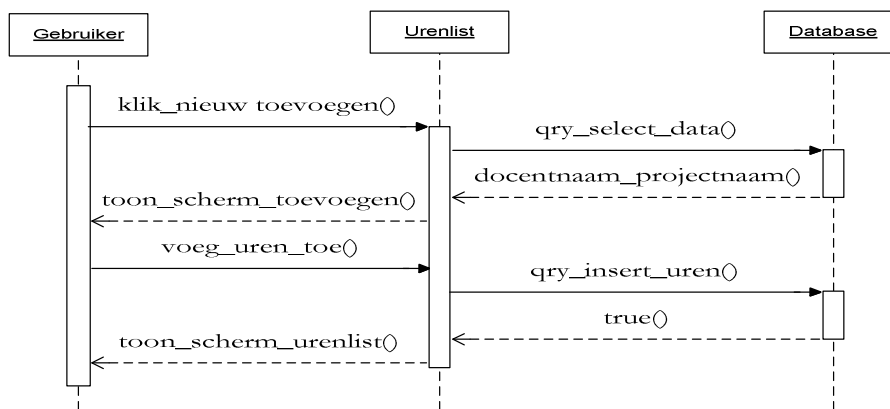
De naamgeving van elke actie, die van het object webbased-urenregistratiesysteem naar het object Database gaat, is afhankelijk van de query's binnen de functies die in de Webbased-urenregistratiesysteem geschreven zullen worden;

Bij het opvragen van de gegevens begint het met **qry_select..**

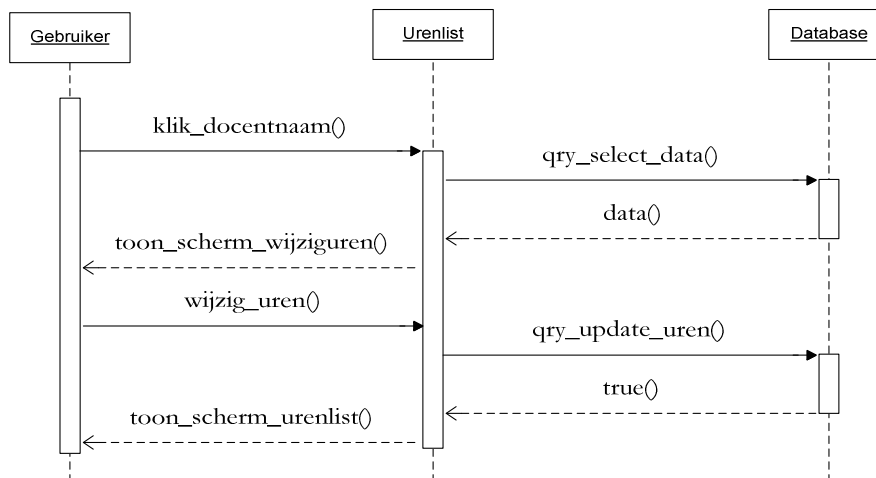
Bij het toevoegen van de gegevens begint het met **qry_insert..**

Bij het wijzigen van de gegevens begint het met **qry_update..**

Hieronder wordt een voorbeeld gegeven van het sequence diagram voor “**beheren uren**” om te laten zien hoe de sequence diagrammen eruit zien.



Figuur 17. Sequence diagram “toevoegen uren”



Figuur 18. Sequence diagram “wijzig uren”

Toelichting sequence diagram

In het figuur 17 en 18 is te zien dat het object gebruiker (docent admin) op het urenlist scherm van Webbased-urenregistratiesysteem op de knop **“toevoegen”** klikt om het scherm te zien voor nieuwe uren registreren. Als gebruiker docent admin is krijgt hij te zien op dat scherm een select menu van alle docenten (docent krijgt alleen de mogelijkheid om zijn eigen uren aan te passen), een select menu voor alle projecten, lege velden voor datum, aantal uren en een commentaar. Na het invullen van dit scherm klikt het object gebruiker op **“Ok”** om de data te verzenden. Het object webbased-urenregistratiesysteem ontvangt de actie van het object gebruiker. In object webbased-urenregistratiesysteem worden functies geschreven en de functies bevatten de SQL query's. Via de query, genaamd query_insert_uren worden de gegevens in de database opgeslagen

Als docent admin de bestaande uren wil wijzigen, dient hij op de naam van de gewenste docent te klikken om het wijzig scherm te krijgen. Het wijzigsscherm bevat de bestaande uren van de geselecteerde docent. Docent admin heeft alle rechten, hij kan alle geregistreerde uren wijzigen en/of aan andere docent, ander project of andere datum toevoegen. Als de gebruiker op de knop **Ok** drukt, worden de nieuwe gegevens in de database opgeslagen.

Als een docent zijn bestaande uren wil wijzigen, dient hij op zijn naam te klikken om het wijzigsscherm te krijgen. Het wijzigsscherm bevat zijn naam en de projecten waar hij aanwerkt, het aantal uren en een commentaar. Docent heeft rechten om geregistreerde uren te wijzigen en/of aan ander project of andere datum toe te voegen. Als de gebruiker op de knop **Ok** drukt, worden de nieuwe gegevens in de database opgeslagen.

Het object webbased-urenregistratiesysteem ontvangt de actie van het object gebruiker. In object webbased-urenregistratiesysteem worden functies geschreven en de functies bevatten de SQL query's. Via de query, genaamd query_update_uren worden de nieuwe uren in de database opgeslagen.

Voor de rest van de activity en sequence diagrammen wordt u naar de **externe bijlage D** het pilotontwikkelrapport van pilot 2 verwezen.

8.4 Ontwikkelen functionaliteit

Naar aanleiding van het ontwerp van deze pilot ben ik begonnen met het ontwikkelen van de functionaliteit. Voor het ontwikkelen van de functionaliteit heb ik gebruik gemaakt van scripttaal PHP. De keuze voor PHP was omdat PHP het standaard scripttaal is binnen TO&I. Het was ook door de opdrachtgever gewenst om de applicatie in PHP te programmeren.

Tijdens het bepalen van user-interfaces had ik besloten om de user-interfaces op basis van de pilotdelen te ontwikkelen. Op basis van de schermen die ik had ontwikkeld, heb ik besloten om de functionaliteit ook op dezelfde manier te ontwikkelen. Ik heb per pilotdeel geprogrammeerd. Tijdens het programmeren heb ik de eerder bepaalde functionele eisen en het ontwerp van deze pilotdelen als uitgangspunt gehanteerd. Ten behoeve van het ontwikkelen van de functionaliteiten heb ik de volgende werkzaamheden uitgevoerd.

- Realiseren inlogpagina.
- Realiseren beheren docenten.
- Realiseren beheren projecten.
- Realiseren beheren rooster.
- Realiseren beheren uren.
- Zoeken naar projecten en docenten.

Bij het realiseren van de pilotdelen heb ik van een bepaalde structuur gebruik gemaakt. De query's worden met behulp van PHP functies aan de database aangeboden. Bij het opvragen, ophalen en verwerken van de gegevens wordt er van deze functies gebruik gemaakt.

Voor de connectie met de database heb ik de file `tucon.php` ontwikkeld, een beschrijving van deze file volgt hieronder

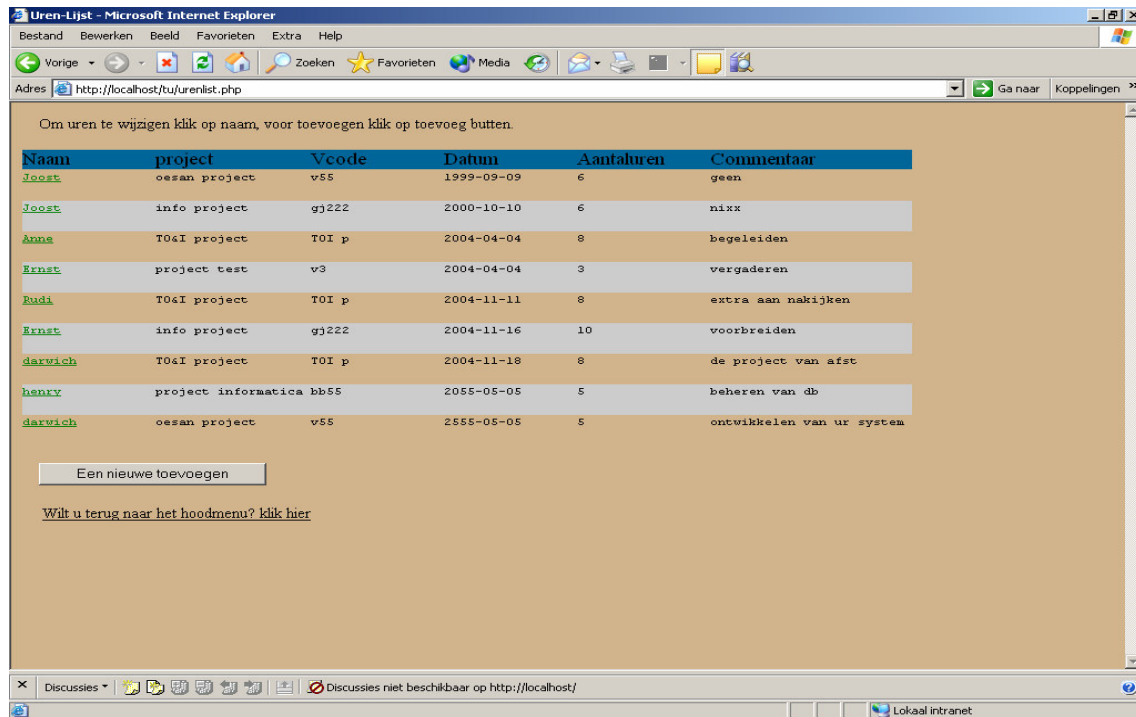
- **tucon.php**; hierin wordt de databaseconnectie geregeld. Dit bestand wordt bij alle pagina's aangeroepen, met behulp van "**require_once("tucon.php")**", waar een connectie met de database noodzakelijk is. Ik heb voor deze manier gekozen om de code niet iedere keer in iedere pagina te herhalen.

In dit bestand bestaan ook de functies om bijvoorbeeld te kunnen inloggen, outloggen en de rechten te bepalen.

Deze file bevat belangrijke gegevens, zoals passwords die toegang geven tot de database. Voor het beveiligen van deze gegevens heb ik deze file in een speciale directory gezet, dit heeft voordelen, de directory is namelijk niet direct te bereiken via de webserver maar alleen via tussenkomst van PHP.

Beheren uren

Het onderstaande scherm is één van de schermen die ik ontwikkeld heb.



Figuur 19. Urenlist admin

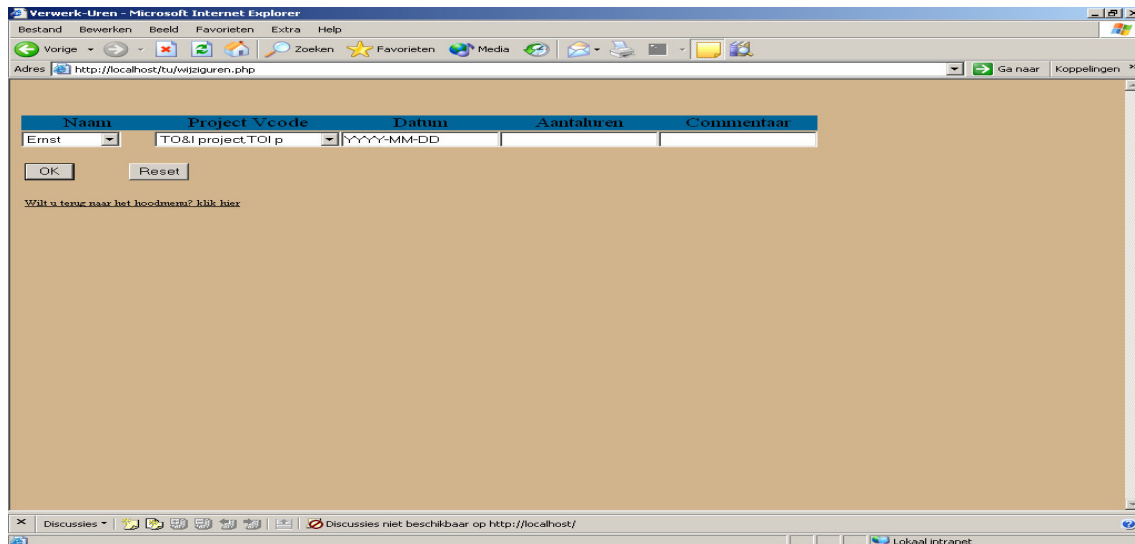
Bij dit scherm krijgt de ingelogde gebruiker (in dit geval docent admin) een lijst van alle docenten en hun bijhorende extra gemaakte uren op het scherm getoond. De gebruiker heeft de mogelijkheid om de uren van alle docenten te wijzigen of nieuwe uren voor een docent te registreren. Bij het ontwikkelen deze pagina heb ik van de onderstaande PHP code gebruikt:

```
<?php require_once("tucon.php");
adminlogin();
mysql_select_db($database_tucon, $tucon);
$query_Recordset2 = " SELECT      urenreg_id, docent_id, naam, uren_reg.project_id,
                                projectnaam, vcode, datum, aantal_uren, commentaar
                                FROM
                                uren_reg, docentfull, project
                                WHERE
                                uren_reg.docent_id=docentfull.id
                                AND
                                uren_reg.project_id=project.project_id
                                AND
                                ISNULL(rooster_id) order by uren_reg.datum ";

$Recordset2 = mysql_query($query_Recordset2, $tucon) or die(mysql_error());

$totalRows_Recordset2 = mysql_num_rows($Recordset2);
?>
```

Als docent admin nieuwe uren wil toevoegen, dient hij op de knop 'Een nieuwe toevoegen' te drukken om de 'toevoegen pagina' te krijgen. De docent admin heeft vrije keuze wat datum, aantaluren en commentaar betreft, maar voor docentnaam, projectnaam en vcode dient hij gebruik te maken van een select menu. Als de gebruiker op de knop 'Ok' drukt, worden de gegevens in de database opgeslagen.



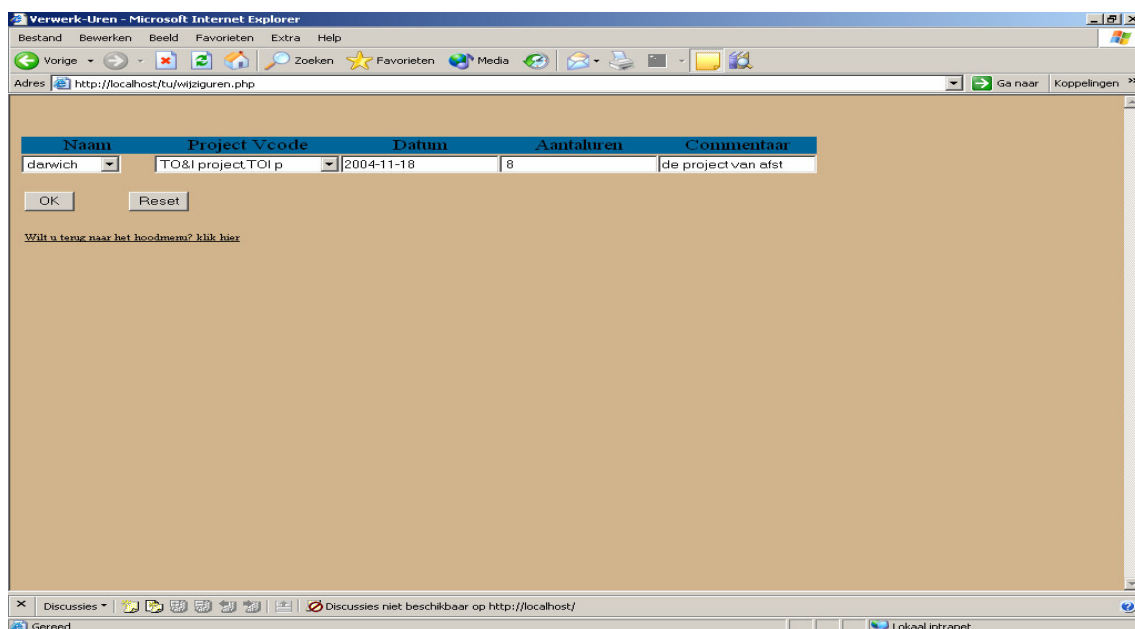
Naam	Project	Vcode	Datum	Aantaluren	Commentaar
Ernst	TO&I project TOI p	TOI p	YYYY-MM-DD		

OK Reset

[Wilt u terug naar het hoofdmenu? klik hier](#)

Figuur 20. Toevoegen uren admin

Als docent admin bestaande uren wil wijzigen, dient hij op de naam van de gewenste docent te klikken om het 'wijzig scherm' te krijgen. Het 'wijzig scherm' bevat de bestaande uren van de geselecteerde docent. Docent admin heeft alle rechten hij kan de geregistreerde uren wijzigen en/of aan andere docent, ander project of andere datum geven. Als de gebruiker op de knop 'Ok' drukt, worden de nieuwe gegevens in de database opgeslagen.



Naam	Project	Vcode	Datum	Aantaluren	Commentaar
darwich	TO&I project TOI p	TOI p	2004-11-18	8	de project van afst

OK Reset

[Wilt u terug naar het hoofdmenu? klik hier](#)

Figuur 21. Wijzig uren admin

Als docent admin voor een docent kiest om zijn uren te wijzigen, krijgt hij een waarschuwing om zeker te zijn van zijn actie. Hier onder is de waarschuwing.



Figuur 22. Extensie controle wijziguren admin

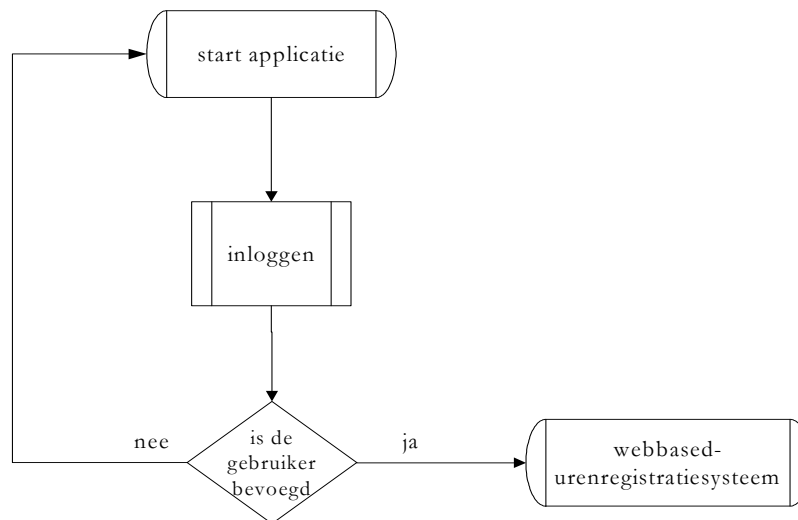
Op deze manier heb ik ook de andere functionaliteiten ontwikkeld. Voor deze pilot had ik een periode ingepland van vier weken. De functionaliteiten die ik wilde verwerken in deze vier weken heb ik niet kunnen halen, aangezien ik in het begin weinig kennis had over programmeertaal PHP. Het oefenen en opdoen van ervaring heeft iets langer geduurd dan de geplande tijd. Hierdoor heb ik besloten om de comfort eisen die aan deze pilot gesteld waren te laten vervallen. Alle basis systeemeisen die aan deze pilot zijn gesteld, waren wel ontwikkeld.

Inlogpagina

In overleg met de opdrachtgevers heb ik besloten om bij alle delen van ontwikkelde functionaliteiten gebruik te maken van de inlogpagina. Een gebruiker moet via een loginscherm toegang kunnen krijgen tot deze applicatie.

De gebruiker gaat inloggen op de applicatie. De gebruiker is een docent admin of een docent. Er zal nagegaan worden of de gebruiker geautoriseerd is om toegang tot het systeem te krijgen.

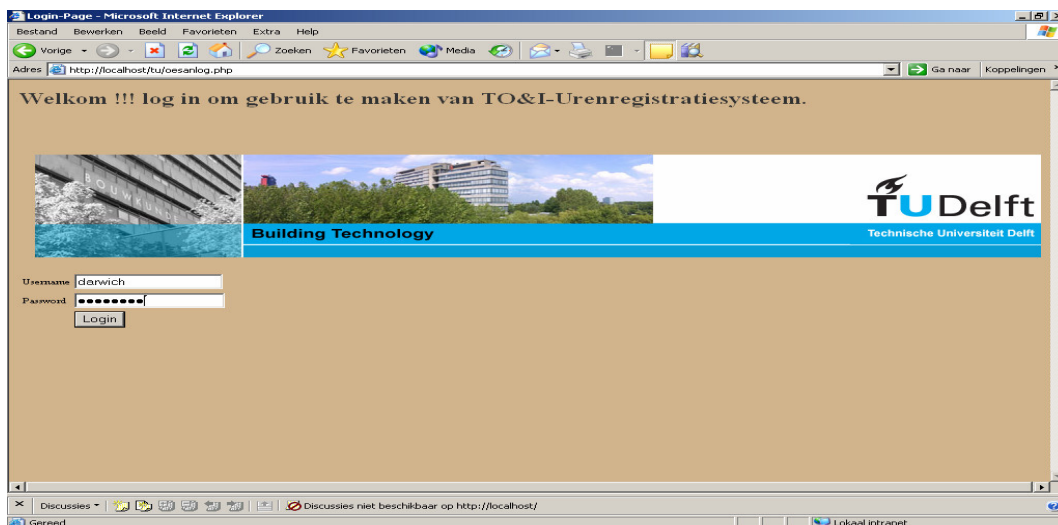
Indien de gebruiker niet bevoegd is om het systeem binnen te komen, zal hij een foutmelding te zien krijgen. Na het controleren van de autorisatie wordt de gebruiker toegelaten.



Figuur 23. Inlogprocedure

Hierbij is de huidige inlogpagina van de bestaande applicatie gebruikt. Dit scherm is een inlogscherm voor de gebruiker. De gebruiker voert eerst zijn gebruikersnaam en password in. Er zal nagegaan worden of de gebruiker geautoriseerd is om toegang tot de applicatie te krijgen.

Indien de gebruiker niet bevoegd is om het systeem binnen te komen, zal hij een foutmelding te zien krijgen.



Figuur 24. Inlogscherm webbased-urenregistratiesysteem

9 Pilot 3 – Overzichten

Na het afronden van de pilot administratiesysteem ben ik gestart met de pilot overzichten. In dit hoofdstuk wordt het realiseren van de pilot overzichten beschreven.

Deze pilot bevat de delen voor de docent admin en de docenten. Pilot overzichten is een product dat de gebruikers de mogelijkheid geeft om verschillende overzichten te genereren.

Om deze pilotdelen te realiseren heb ik de volgende activiteiten verricht;

- Opstellen plan van aanpak.
- Ontwerpen user-interface
- Ontwikkelen user-interface.
- Ontwerpen functionaliteit overzichten
- Ontwikkelen functionaliteit overzichten.

9.1 Opstellen plan van aanpak

Het doel van deze activiteit is duidelijke afspraken vast te stellen over inhoud en omvang van de op te leveren producten en daarvoor uit te voeren werkzaamheden. In het plan van aanpak van pilot 3 heb ik in het kort het doel en een planning beschreven. Dit plan van aanpak heeft als voerzicht gefunctioneerd tijdens het ontwikkelen van deze pilot.

Tijdsduur:

De tijdsduur voor het ontwikkelen van pilot 2 is 4 weken.

Tijdens het plannen van deze pilot moest ik ook met het schrijven van eindverslag rekening houden.

Time-boxing:

Bouweenheid	Prioriteit	Kosten
Ontwerpen en ontwikkelen user-interfaces.	Basis	2 dagen
Ontwerpen functionaliteit overzichten.	Basis	5 dagen
Ontwikkelen functionaliteit overzichten.	Basis	8 dagen

9.2 Ontwerpen en ontwikkelen user-interface

In deze paragraaf wordt het pilotdeel ontwerp van de user-interface van de pilot overzichten beschreven. Dit pilotdeel is opgesteld omdat het nog niet duidelijk was hoe de user-interface van dit pilotdeel eruit moest zien. Het doel van dit pilotdeel is het bepalen van de user-interface voor de pilot overzichten.

Allereerst heb ik de systeemeisen die voor dit pilotdeel relevant zijn, verzameld. Het gaat om de volgende eisen:

ID	User-Interface eisen	Prioriteit
U1	De taal die gebruikt wordt moet Nederlands zijn.	Basis
U2	De interface dient overzichtelijk te zijn.	Basis
U3	De schermen dienen consistent te zijn. Hierbij kan gedacht worden aan lettertype, de kleuren, lettergrootte, drukknoppen of hyperlinks.	Basis

Om tot een geschikte user-interface te komen heb ik me de volgende twee vragen afgevraagd;

- Wat krijgt de gebruiker te zien.
- Hoe krijgt de gebruiker de informatie te zien.

Wat krijgt de gebruiker te zien

Om te bepalen welke en hoeveel informatie de gebruiker te zien zal krijgen, heb ik de use-cases uit het rapport definitiestudie als uitgangspunt genomen. Hiernaast heb ik ook voor dit pilot relevante eisen geïnventariseerd om een beeld te krijgen over de uiteindelijke vorm van user-interface. Bij deze use-cases wordt beschreven welke acties de actoren kunnen uitvoeren. Nadat ik de use-cases uitvoerig heb bekeken heb ik de volgende acties bepaald. De acties zijn als volgt;

Acties binnen use-cases

Docent	Student
+ inloggen()	+ inloggen()
+ zoeken op docent()	+ zoeken eigen overzichten()
+ zoeken op project()	+ genereren eigen overzichten()
+ zoeken op datum()	
+ selecteren docent()	
+ selecteren project()	
+ invullen datum()	
+ genereren overzichten()	

Bij de schermen heb ik onderscheid gemaakt tussen docent admin en docent. Eén docent admin heeft meer rechten dan één docent. Dat wil zeggen dat een docent allen maar overzichten van zichzelf mag bekijken terwijl een docent admin alle overzichten mag bekijken. Hierdoor krijgt de docent admin de mogelijkheid om eerst een docent of een project te zoeken om de desbetreffende overzichten te bekijken.

Hoe krijgt de gebruiker de informatie te zien?

De manier waarop informatie wordt gepresenteerd aan de gebruiker is een ander aspect voor de user interface. Hierbij gaat het om de lay-out waarbij het vooral om de aspecten als kleur, lettergrootte, drukknoppen of hyperlinks aan orde komen.

Met betrekking tot de user-interface waren er aantal eisen gesteld. Deze user-interface eisen heb ik in het begin van de paragraaf aangegeven. Één van de eisen is als volgt;

ID	User-Interface eisen	Prioriteit
U3	De schermen dienen consistent te zijn. Hierbij kan gedacht worden aan lettertype, de kleuren, lettergrootte, drukknoppen of hyperlinks.	Basis

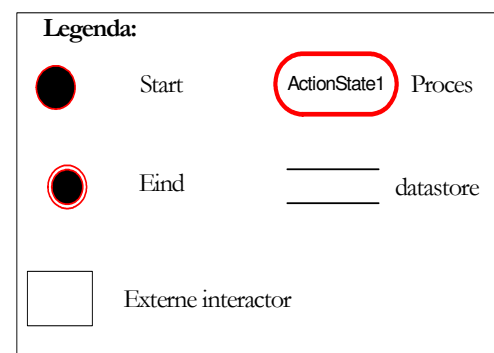
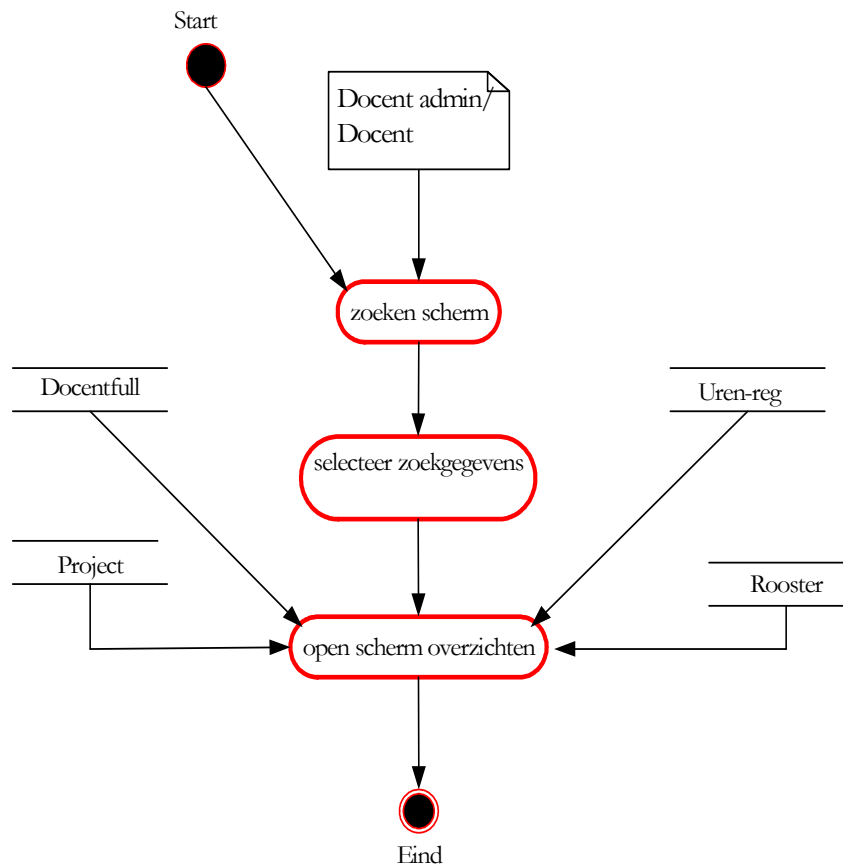
Naar aanleiding van het ontwerp user-interface heb ik de interfaces ontwikkeld. De user-interfaces zijn in HTML ontwikkeld met daarin statische gegevens. De verschillende HTML pagina's zijn aan elkaar gekoppeld door middel van hyperlinks.

9.3 Ontwerpen functionaliteit

Voordat de applicatie ontwikkeld wordt, is het belangrijk om te weten welke functionaliteiten de applicatie moet hebben. In hoofdstuk 6.2 van het verslag heb ik aangegeven welke functionele eisen gesteld worden aan de applicatie. Als eerste heb ik de functionele eisen die betrekking hebben op deze pilot verzameld. Hieronder geef ik de belangrijke functionele eisen met betrekking tot deze pilot weer.

Functionele eisen	Prioriteit
Het moet mogelijk voor iedere docent zijn rooster te kunnen bekijken	Comfort
Het genereren van overzichten per docent met betrekking tot de gemaakte uren per project.	Basis
Het genereren van overzichten per docent met betrekking tot de gemaakte uren per week/maand/jaar.	Basis
Het genereren van overzichten per project met betrekking tot de gemaakte uren per docent.	Basis
Het genereren van overzichten per project met betrekking tot de gemaakte uren per week/maand/jaar.	Basis

Activity diagram



Figuur 25. Activity diagram “overzichten”

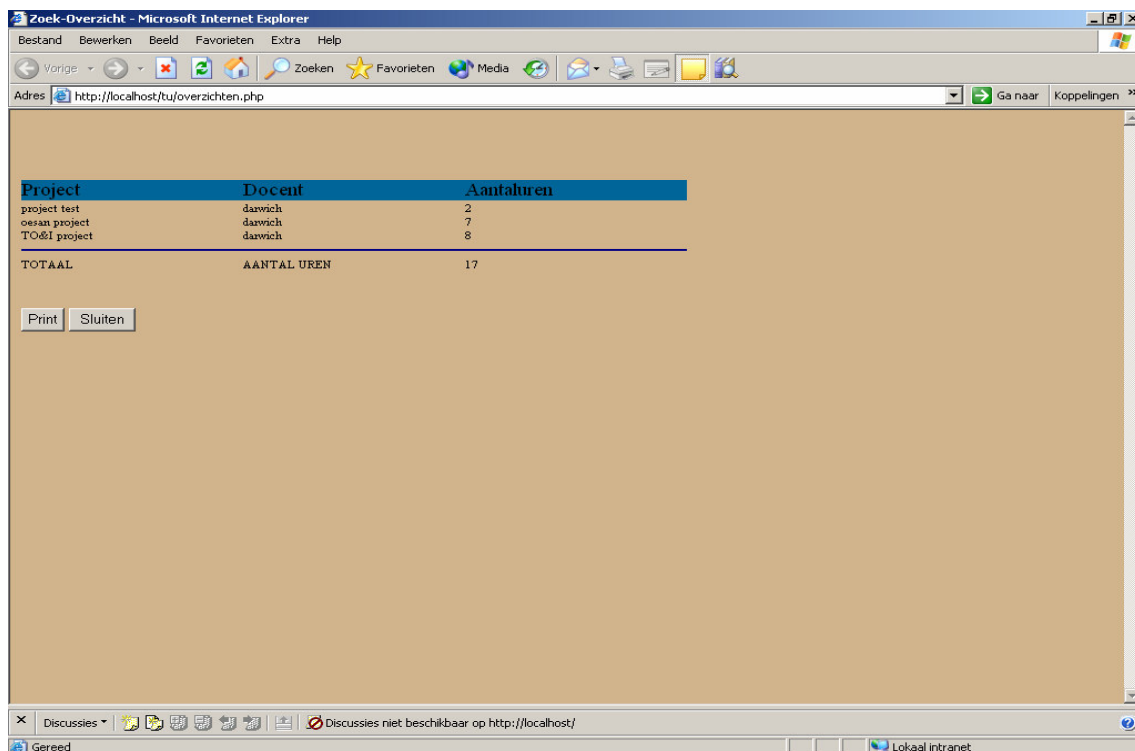
9.4 Ontwikkelen functionaliteit

Na het ontwerpen van de functionaliteit van deze pilot ben ik begonnen met het ontwikkelen van de functionaliteit. Tijdens het ontwikkelen heb ik me beziggehouden met het realiseren van verschillende overzichtspagina's voor docent admin en docenten. Ik heb de eerder bepaalde functionele eisen en het ontwerp als uitgangspunt gehanteerd.

Hieronder zijn twee voorbeelden van de overzichtspagina's die ik ontwikkeld heb.

Urenoverzicht per docent

Om een urenoverzicht te genereren moet de gebruiker eerst de zoekpagina openen, daarna moet docent admin één docentnaam kiezen (docent krijgt alleen zijn eigen naam) waarvan hij de urenbesteding overzicht wil bekijken. Na het zoeken op een docent wordt een overzicht van zijn totale uren besteding getoond. Op dat overzicht staat zijn naam, de naam van het project waar hij deel heeft genomen, de totale uren besteding (de roosteruren plus de extra uren) aan dit project en als laatste de totale urenbesteding overzicht van deze docent. Als de gebruiker op de knop 'print' drukt, wordt dit overzicht naar de printer gestuurd de knop sluiten voor het sluiten van het scherm.



The screenshot shows a Microsoft Internet Explorer window titled 'Zoek-Overzicht - Microsoft Internet Explorer'. The address bar shows 'http://localhost/tu/overzichten.php'. The main content area displays a table with three columns: 'Project', 'Docent', and 'Aantahuren'. The table lists three projects: 'project test' (2 hours), 'ocean project' (7 hours), and 'TO&I project' (8 hours). Below the table, there is a summary row: 'TOTAAL' (AANTAL UREN) with a value of 17. At the bottom of the table, there are two buttons: 'Print' and 'Sluiten'.

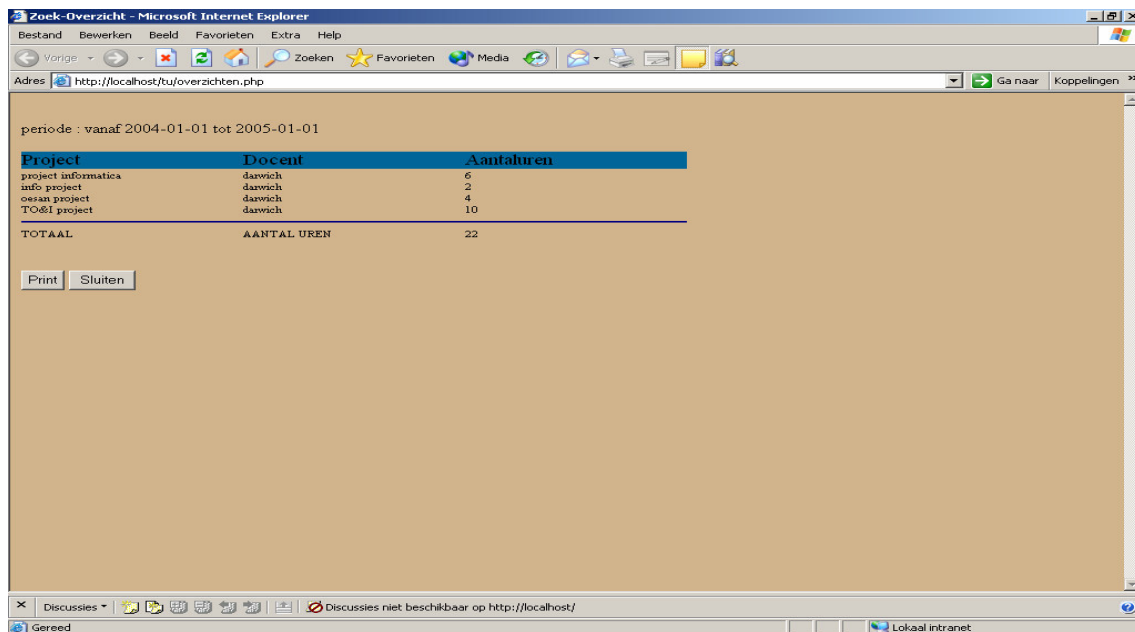
Project	Docent	Aantahuren
project test	darwich	2
ocean project	darwich	7
TO&I project	darwich	8
TOTAAL	AANTAL UREN	17

Print Sluiten

Figuur 26. Urenoverzichten per docent

Urenoverzicht per docent per datum

Om een urenoverzicht voor een docent binnen bepaalde tijd te genereren moet de gebruiker eerst de zoekpagina openen, daarna moet docent admin de gewenste docent kiezen het gewenste begin en eind datum invoeren (docent krijgt een overzicht alleen van zichzelf in de aangegeven datum) waarvan hij de urenbesteding overzicht wil bekijken. Na het zoeken op datum wordt een overzicht van de totale datum urenbesteding getoond. Op dat overzicht staat de docentnaam, de naam van de projecten waaraan deze docent deel heeft genomen, de totale uren besteding (de roosteruren plus de extra uren) aan dit project en als laatste de totale urenbesteding die een docent binnen de aangegeven datum besteed heeft. Als de gebruiker op de knop 'print' drukt wordt dit overzicht naar de printer gestuurd de knop 'Sluiten' is voor het sluiten van het scherm.



periode : vanaf 2004-01-01 tot 2005-01-01

Project	Docent	Aantahuren
project informatica	darwich	6
info project	darwich	2
oesaa project	darwich	4
TO&I project	darwich	10
TOTAAL	AANTAL UREN	22

Print Sluiten

Figuur 27. Urenoverzichten per docent per datum

Voor de rest van overzichten wordt u naar de **externe bijlage E** het pilotontwikkelaapport van pilot 3 verwezen.

9.5 Beoordelen en testen pilots

Tijdens het realiseren van de pilots heb ik wel per pilotdeel getest. Ik was tijdens de implementatie begonnen met het testen. Zodra er een functionaliteit werd toegevoegd, werd dit direct getest. Deze informele test was bedoeld om te vergelijken of de werking van de applicatie overkwam met de functionaliteiten die beschreven stonden in de use-cases. Alle functionele eisen en use-cases werden stuk voor stuk doorlopen met de opdrachtgever en mijn programmeerbegeleider om te kijken of de gemaakte functionaliteit overeenkwam met de use-cases. Als dat goed liep ging ik door naar de volgende functionele eis. Dat heb ik bij alle functionele eisen toegepast die ik ontwikkeld heb.

10 Evaluatie

In dit hoofdstuk evalueer ik de werkzaamheden die zijn verricht tijdens mijn afstudeerperiode. De evaluatie van de afstudeerperiode is opgesplitst in een procesevaluatie, waarin wordt teruggeblikt naar de procesgang die ik heb gevolgd, en een productevaluatie, waarin de kwaliteit van de producten wordt geëvalueerd.

10.1 Procesevaluatie

Vanaf het begin tot het einde van mijn afstudeerperiode heb ik een soort logboek bijgehouden waarin ik dagelijks heb beschreven welke activiteiten ik heb uitgevoerd en wat voor problemen ik ben tegengekomen tijdens het uitvoeren van het project. Bij het evalueren van het proces heeft dat logboek mij geholpen.

In het begin van de afstudeerperiode heb ik met de opdrachtgevers wekelijkse gesprekken gepland. In plaats van de workshops heb ik deze gesprekken gehouden. Tijdens deze gesprekken ging het vooral om de voortgang, de tussenproducten, eventuele vragen en problemen die ik had met betrekking tot het project. Dat heeft een belangrijke rol gespeeld om de voortgang van het project en eventuele problemen bij te houden. Op deze manier heb ik wekelijks de voortgang van het project kunnen bijhouden. Ik had ook de mogelijkheid om feedback te krijgen op de ingeleverde producten en documenten.

Aanvullend op deze gesprekken heb ik alle andere medewerkers van TO&I kunnen spreken als er vragen of problemen waren. Er zijn hoofddocenten, docenten, student-assistenten en onderzoekers werkzaam bij de afdeling waar ik mijn afstudeerstage heb volbracht. De gebruikersdoelgroep van mijn project zijn de docenten en studenten. Hierdoor kon ik indien nodig ook de andere medewerkers raadplegen. Ik ben zeer tevreden aangezien ik steeds de mogelijkheid heb gehad om met alle medewerkers op de afdeling TO&I ideeën en informatie uit te wisselen in het kader van mijn project.

Methode en techniek

Tijdens het uitvoeren van mijn afstudeerproject heb ik gebruik gemaakt van de ontwikkelmethode IAD. IAD is een ontwikkelmethode waarbij veel interactie met de gebruiker is.

Binnen TO&I worden geen specifieke ontwikkelmethoden gehanteerd bij de ontwikkeling van de informatiesystemen. Er wordt een gegevensstructuur getekend en vervolgens wordt er gestart met het programmeren. Ik denk daarom, dat dit project een goede mogelijkheid is geweest om TO&I te confronteren met een gestructureerde aanpak van systeemontwikkeling, waarbij verschillende fasen worden doorlopen, voordat men overgaat tot het bouwen van het systeem.

Ik vond het niet zo lastig om een keuze te maken, omdat ik vlak voor de afstudeerperiode de IAD methode op school toegepast had. Ik vond IAD bij het project goed passen. Ik heb het project in een informele organisatie uitgevoerd. Door de gebruikers nauw te betrekken in het ontwikkelproces heb ik de mogelijkheid gehad om feedback te krijgen en te verwerken. Omdat er met IAD de mogelijkheid is om terug te vallen op eerdere fasen. Dit gaf mij de mogelijkheid om flexibel te werken tijdens de ontwikkeling. Er was bijvoorbeeld de mogelijkheid om de systeemeisen in een later stadium te herzien.

Het modelleren van het ontwerp van de pilots met UML ging soepel. Door de ervaring die ik op school heb opgedaan met UML, kon ik dit tijdens mijn project goed toepassen. Af en toe zijn er situaties geweest waar ik moeite mee had, hiervoor heb ik het UML boek geraadpleegd. (Zie literatuurlijst).

Strategie

In mijn project heb ik gebruik gemaakt van de “big-bang” iteratiestrategie. In de strategie worden de fasen definitiestudie en pilotontwikkeling opgehouden. De fase invoering wordt pas afgehandeld wanneer de vorige twee fasen in zijn geheel waren afgerond.

De ontwikkelde pilots zijn niet ingevoerd, omdat het na de afstudeerperiode zal plaatsvinden. Ik had dus alleen de optie om de definitiestudie in de cycli mee te nemen. Ik had namelijk nog de vrijheid om veranderingen aan te brengen in de definitiestudie. Dat vond ik nodig omdat niet elk punt van te voren kon worden vastgesteld.

In de fase pilotontwikkeling heb ik besloten om binnen deze fase een strategie toe te passen die mij de mogelijkheid gaf om elk onderdeel te kunnen itereren.

Achteraf vind ik dat het een goede keuze is geweest. Doormiddel van volledige iteratie had ik de vrijheid om te itereren in het hele ontwikkelingsproces. Ik hoefde niet een hele fase te itereren wanneer dat nodig was.

Ontwikkelomgeving

Tijdens het bouwen van de functionaliteiten heb ik van PHP - HTML en MySql gebruik gemaakt, aangezien ik ervaring met HTML en SQL heb, heb ik daar geen moeite mee gehad. Begin van de afstudeerperiode heb ik wel moeite gehad met PHP. In het begin van mijn afstudeerperiode heb ik een paar oefeningen gedaan om vertrouwd te raken met PHP. De bestaande HTML kennis heb ik ook moeten uitbreiden. Tijdens mijn stage had ik met ASP gewerkt. ASP is ook een programmeertaal (script taal) om webapplicaties te ontwikkelen. Gezien de ervaring met ASP had ik al basiskennis over het programmeren van webapplicaties.

Aangezien ik in de zelfde werkplek met de programmeurs van TO&I heb gezeten, heb ik de mogelijkheid gehad om vragen te stellen over programmeren. Dat vond ik een groot voordeel. Naarmate mijn project vorderde, raakte ik meer vertrouwd met PHP en verliep het programmeren een stuk soepeler.

Planning

Het opstellen van een goede planning is en blijft een lastig onderdeel. Ik had namelijk het idee dat ik alles met voldoende tijd had gepland, maar dit bleek achteraf toch niet het geval te zijn. Op zich heb ik alle werkzaamheden volgens de planning uitgevoerd. Maar aangezien ik bijvoorbeeld bij pilot 2 weinig tijd had om alle functionele eisen uit te voeren heb ik de comforteisen latenvallen om op tijd te kunnen beginnen aan de pilot 3. Ik had beter inwerktijd kunnen plannen. Het verwerken van de feedback op de tussenproducten heeft ook tijd gekost. Maar dat heeft geen vertraging opgeleverd.

Als ik de planning opnieuw zou opstellen, zou ik met een korte inwerktijd rekening houden. Om eventuele feedback te verwerken zou ik ook per fase een paar dagen meer plannen.

10.2 Productevaluatie

Plan van aanpak

Het doel van het plan van aanpak is voornamelijk om duidelijkheid te scheppen voor zowel de opdrachtgever als de afstudeerder over de omvang, diepgang en de planning van de opdracht. Het opstellen van het plan van aanpak was een belangrijk aspect voor mij.

Het plan van aanpak heeft door het hele project heen als leidraad gediend. Hierin stond onder andere de doelstelling die gerealiseerd moest worden. Het plan van aanpak was het eerste deelproduct dat opgeleverd is aan de opdrachtgever. Na feedback is dit product een definitief deelproduct geworden. Ik vind dat aan dit product extra aandacht moest worden gegeven om een beeld te krijgen van wat het project inhield en wat voor hulpmiddelen gebruikt moesten worden om het project uit te voeren. Eén van de belangrijkste punten was het bepalen van de methoden en technieken. Ik ben wel tevreden over het plan van aanpak dat ik heb opgesteld.

Definitiestudie

Het opstellen van het definitiestudie rapport verliep goed. Na veel besprekingen met begeleider / opdrachtgever heb ik in het begin van mijn project een eerste opzet van dit rapport kunnen samenstellen. In de 3 week was ik klaar met mijn definitiestudie. Tijdens de ontwikkeling van de pilots zijn er nieuwe eisen naar voren gekomen waardoor ik de definitiestudie moest aanpassen. Deze eisen heb ik in de definitiestudie opgenomen. De definitiestudie werd in de loop van het project enkel iets uitgebreid.

Pilotontwerpen

Ik ben positief te spreken over het resultaat uit de fase pilotontwikkeling.

Pilot 1 is prima gelukt, dat kwam doordat de database en SQL kennis van mij een sterk punt was tijdens het afstuderen, ik kon de kennis aantonen en toepassen tijdens mijn afstudeerproject. De pilot administratiesysteem heeft het meeste tijd in beslag genomen. Tijdens het programmeren heb ik wat problemen gehad omdat ik eerst met de programmeertaal PHP moest oefenen. Hierdoor moest ik de comforteisen laten vallen om deze pilot binnen de gepande tijd af te maken. Alle basiseisen met betrekking tot deze pilot zijn gerealiseerd. Het eind resultaat sluit goed aan bij datgene wat ik aan het begin voor de ogen had. Hier ben ik ook positief over.

Slot

Over het algemeen kan ik met een goed gevoel terugkijken naar het verloop van het totale project mede doordat ik een hoop kennis heb opgedaan in de afgelopen afstudeerperiode. Het is voor mij de eerste ervaring geweest om in een onderwijsinstelling aan een deelproject te werken.

Ik vond het leuk om mijn afstudeeropdracht binnen de TU Delft uit te voeren. Ik ben dan ook veel dank verschuldigd aan allen die mij hebben bijgestaan tijdens de uitvoering van dit project.

Literatuurlijst

- IAD, Het evolutionair ontwikkelen van informatiesystemen.
R.J.H Tolido Academic Service, Schoonhoven 2001
- Praktisch UML
Jos Warmer & Anneke Kleppe Salland de Lange, Deventer 1999
- Moduleboek DB-01, Relationale Databases
T.H.M. Spaan Haagse Hogeschool, Den Haag 1998
- Databasesystemen voor de praktijk
J.A. Vandenbulcke, Deventer 1997
- HTML 4
Steyer, Ralph Nathan.
- Complete handboek PHP
Academic service
- Reader Communicatieve vaardigheden / AV-03 (909)
Haagse Hogeschool, den haag 1998

Internet

- <http://www.phpfreakz.nl>
- <http://www.php.net>
- <http://www.phphulp.nl>
- <http://www.mysql.com>
- <http://php.pagina.nl>
- <http://www.hotscripts.com>
- <http://msdn.microsoft.com/library>
- <http://www.google.nl>

Afkortingenlijst

DBMS	: Database management systeem
ERD	: Entity relation diagram
HTML	: HyperText Markup Language
IAD	: Iteratieve Application Development
IP-03	: Integrerend practicum van de opleiding informatica en informatiekunde aan de Haagse Hogeschool
PHP	: Hypertext Preprocessor
SQL	: Structured query language
TU Delft	: Technische universiteit Delft
TO&I	: Technische ontwerp & Informatica
UML	: Unified Modeling Language

Figurenlijst

Figuur 1	: Organigram faculteit bouwkunde TU Delft	4
Figuur 2	: Organigram TO&I	7
Figuur 3	: Fasen systeemontwikkeling IAD	10
Figuur 4	: Big-bang strategie van IAD	12
Figuur 5	: Planning	15
Figuur 6	: Use case diagram	24
Figuur 7	: Big-bang iteratiestrategie	26
Figuur 8	: Big-bang iteratiestrategie + volledige iteratie pilotontwikkeling	26
Figuur 9	: Legenda ERD	30
Figuur 10	: Relatie types	30
Figuur 11	: Het ERD	31
Figuur 12	: Structuur van de tabel “Uren-reg”	33
Figuur 13	: Hoofd scherm om de applicatie te beheren	37
Figuur 14	: Hoofd scherm voor docenten	37
Figuur 15	: Drukknoppen	38
Figuur 16	: Activity diagram “beheren uren”	39
Figuur 17	: Sequence diagram “toevoegen uren”	41
Figuur 18	: Sequence diagram “wijzig uren”	41
Figuur 19	: Urenlit admin	44
Figuur 20	: Toevoegen uren admin	45
Figuur 21	: Wijzig uren admin	45
Figuur 22	: Extensie controle	46
Figuur 23	: Inlogprocedure	47
Figuur 24	: Inlogschermb	47
Figuur 25	: Activity diagram “overzichten”	51
Figuur 26	: Urenoverzicht per docent	52
Figuur 27	: Urenoverzicht per docent per datum	53

Bijlagen

- Interne bijlage 1 : opdrachtomschrijving
- Externe bijlage A : plan van aanpak
- Externe bijlage B : definitiestudie
- Externe bijlage C : pilot 1 database
- Externe bijlage D : pilot 2 administratiesysteem
- Externe bijlage E : pilot 3 overzichten

INTERNE BIJLAGE 1 Opdrachtschrijving

Assan Darwich

Kenmerk: DOA2004-2.1

Wijzigingen in de student- en/of bedrijfsgegevens (j/n):Nee

zo ja, welke:

Omschrijving aangepast door: Assan Darwich

Ontwikkelen van het webbased-urenregistratiesysteem, voor docenten bouwkunde TU Delft.

De leerstoel Technisch Ontwerp & Informatica (TO&I) behoort tot de faculteit Bouwkunde van de Technische Universiteit Delft. Bij deze afdeling van TU-Delft zijn 25 medewerkers werkzaam. TO&I houdt zich vooral bezig met informatica toepassingen, technieken en methoden en is gericht op ontwerpen en materialiseren in het bouwproces. Tevens geeft TO&I onderwijs en doet onderzoek naar het gebruik van computersoftware bij het ontwerpen en ontwikkelen van gebouwen.

In het kader van de aangeboden projecten dienen de medewerkers (leerstoel TO&I) een urenregistratie bij te houden. Dat houdt in dat iedere medewerker per maand aangeeft hoeveel uren hij of zij elke dag aan een project besteed heeft.

Er is momenteel geen duidelijke overzichten van de uren besteding aan iedere project, het kan voorkomen dat een klein project meer uren besteedt dan wat grotere projecten.

Door het rooster van de docenten kan het voorkomen dat de studenten bij een zelfde project door verschillende docenten te woord gestaan worden. Om de uren van de docenten bij te houden, wil men een digitale omgeving aanbieden, waarbij een duidelijk beeld wordt gegeven van de uren die op het rooster staan, plus de extra uren die worden besteed aan de werkzaamheden die buiten het rooster vallen, zoals het voorbereiden van de lessen en het nakijken van de wekstukken.

Het doel van de afstudeeropdracht is het ontwikkelen van een Webbased-urenregistratiesysteem dat het mogelijk maakt voor elke medewerker per dag aan te geven hoeveel uren hij of zij aan een project besteed heeft. Verder dient per project en per medewerker een overzicht uitgedraaid te kunnen worden van de totale urenbesteding.

De Website dient zowel een eindgebruikersdeel als een administratief deel te bevatten.

Administratief moet bepaald kunnen worden wie toegang heeft tot de Website, welk projecten bestaan en wie aan welke activiteiten deelneemt. Daarbij dient ook gebruik gemaakt te kunnen worden van een onderwijsrooster (een overzicht van de vakken die de docenten geven, wanneer, waar en voor wie). Hiermee dienen de ingeroosterde uren automatisch te worden geregistreerd in het Webbased-urenregistratiesysteem, ook dienen alle benodigde overzichten gegenereerd en uitgeprint te kunnen worden, zoals de koppeling van de studentenassistenten aan de onderwijstaken. De gebruiker dient per dag en per project te kunnen aangeven hoeveel uren hij of zij besteed heeft aan een project en een maandelijks overzicht te krijgen. De authenticatie moet verlopen via een wachtwoord en een gebruikersnaam. De opslag van gegevens dient na een onderzoek ofwel verzorgd te worden door InfoBase of een eigen My SQL database.

Het systeem zou ook een zekere intelligentie moeten kennen:

- Rekening houden met de huidige datum (dag en maand).
- Via een koppeling met Excel automatisch informatie uit het onderwijsrooster inlezen.
- Via een koppeling met Outlook automatisch informatie uit een agenda inlezen (de informatie over de agenda van een docent moet opgevraagd kunnen worden via Outlook en deze informatie overgenomen worden in het urenregistratiesysteem)

De volgende software zal gebruikt worden:

- Windows NT.
- MS-Office.
- Visio 2000.
- MySQL.
- PHP.

De volgende rapporten zijn beschikbaar:

- Documentatie over de huidige applicatie InfoBase.

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

- Opstellen van een plan van aanpak:
Beheersaspecten.
Risicoanalyse.
Aanpak van het project.
Planning.
- Bepalen database (onderzoeken naar de geschikte database)
- Opstellen van een definitiestudie:
Definiëren van de huidige situatie.
Definiëren van het ontwikkelscenario.
Definiëren van systeemeisen.
Bepalen van een systeemconcept.
Beschouwen van de technische structuur.
Beschouwen van de organisatorische inrichting.
Opstellen van een pilotplan.
- Ontwikkelen van pilots:
Vorbereiden en houden van een pilotontwerp workshop.
Specificeren van de globale organisatorische inrichting van de pilot.
Specificeren van de globale functionele structuur van de pilot.
Specificeren van de globale technische structuur van de pilot.
Opstellen van een pilotontwikkelplan.
Ontwerpen van software-bouweenheden.
Realiseren van software-bouweenheden.
Integreren van bouweenheden.
Beoordelen en testen van het pilotdeel.

- Invoeren van webbased-urenregistratiesysteem:
Invoeren van de applicatie webbased-urenregistratiesysteem.
Opstellen van een testplan.
Uitvoeren van een acceptatietest.

Bij de uitvoering van de opdracht zal IAD gehanteerd worden. De volgende technieken zullen gebruikt worden:

- Interviewtechnieken.
- De technieken van UML
- ERD.

De volgende producten zullen opgeleverd worden:

- Plan van aanpak.
- Definitiestudie.
- Documentatie van de pilots.
- De applicatie het webbased-urenregistratiesysteem