

Eindverslag

Peter Langhout Reizen
Snowweb, een generieke boekingsmodule



Frank de Visser
99005364
Versie 5.1



Eindverslag

Frank de Visser

Instituut: Haagse Hogeschool

Opleiding: Informatica en Informatiekunde

Variant: Vormgeving en ontwerp van InterActie (VIA)

Peter Langhout Reizen

Bedrijfsmentor: Ing. V.G. van Hulst (tot begin september)

Bedrijfsmentor: Ing. O. Bollen (vanaf begin september)

Haagse Hogeschool

Examinator: Jim Graven

Examinator: Victor Hermans

Alphen a/d Rijn, 07 oktober 2004

Referaat

Dit verslag beschrijft het proces van het afstudeertraject “Snowweb, een generieke boekingsmodule” dat uitgevoerd is door Frank de Visser, student Vormgeving en ontwerp van InterActie aan de sector Informatica van de Haagse Hogeschool. De opdracht is uitgevoerd aan de hand van de IAD ontwikkelmethode.

Descriptoren:

- Graphical User Interface
- Gebruik(er)svriendelijkheid
- Databases
- SQL Server Scripting
- ASP.NET / VB.NET

Voorwoord

In de periode van 5 april 2004 t/m 8 oktober 2004 heb ik stage gelopen bij Peter Langhout Reizen en gewerkt aan mijn afstudeerproject.

Ik ben bij Peter Langhout Reizen terecht gekomen via een vriend, deze dacht wel een opdracht voor mij te hebben, maar was daar dusdanig vaag over dat ik eerst ben gaan kijken bij andere touroperators, omdat de reiswereld me wel leuk leek. Ik heb daarop gereageerd op een afstudeeropdracht bij Gogo Tours. Nadat ik hier op had gereageerd lagen de zaken voor me open. Peter Langhout Reizen wilde niet dat ik bij Gogo Tours kwam werken, en bood mij een goede afstudeeropdracht en meer salaris, waardoor de keuze voor mij snel was gemaakt.

De afstudeeropdracht bestond uit de ontwikkeling van het volledig nieuwe Snowweb. Met dit product zou Peter Langhout Reizen de wintersportwereld willen terug winnen en ook zo een generieke boekingsmodule hebben voor andere segmenten van de reisbranche. Het nieuwe idee achter Snowweb is eigenlijk dat de Hoteliers hun eigen kamers in mogen vullen en deze worden direct aan de reiziger aangeboden, met Peter Langhout Reizen slechts als controleur en doorgeefluik. De opdracht leek mij interessant omdat ik kennis zou op doen op het gebied van projectmatig werken, dieper ingaan op databases en kennismaking met de, voor mij nieuwe taal, ASP.NET / VB.NET

In het verslag zelf zullen een aantal termen gebruikt worden die u wellicht niet bekend voorkomen en specifiek voor de reiswereld zijn. Indien mogelijk verklaar ik de termen al in hetzelfde stuk als waar ze gebruikt worden, maar als extra hulp kunt u achter in het verslag een verklarende woordenlijst vinden.

Tot slot wil ik graag de mensen bedanken, die mij hebben geholpen bij mijn voorbereiding op de stage en tijdens mijn stage; Jannie Oosting voor haar steun tijdens mijn overgang van Openduaal (zie verklarende woordenlijst) naar mijn huidige stage, mijn examinatoren voor de input tijdens mijn stage en alle medewerkers van Peter Langhout Reizen die mij geholpen hebben, in het bijzonder Vincent van Hulst en Otje Bollen.

Frank de Visser

Inhoudsopgave

1	Inleiding	7
2	Achtergrondinformatie.....	8
2.1	Inleiding	8
2.2	Over Peter Langhout Reizen	8
2.3	Totstandkoming van de afstudeeropdracht.....	8
2.4	Organisatorische inrichting.....	9
2.5	ICT Situatieschets	11
2.5.1	De oude situatie weergegeven	11
2.5.2	De nieuwe situatie weergegeven.....	12
3	Opdrachtomschrijving.....	14
3.1	Totstandkoming van de probleemstelling.....	14
3.2	Vervaardigen van een doelstelling.....	14
3.3	Welke informatie was er beschikbaar?	14
3.4	Waarom de IAD methode?.....	15
3.5	Verbetering opdrachtomschrijving.....	15
4	Definitiestudie	16
4.1	Inleiding	16
4.2	Opstellen plan van aanpak.....	16
4.3	Hoe zijn de systeemeisen tot stand gekomen?	17
4.4	De vervaardiging van het systeemconcept.....	19
4.5	Opstellen pilotplan.....	21
5	Pilotontwikkeling.....	23
5.1	Inleiding	23
5.2	Startproblemen.....	23
5.3	Ontwikkeling van de database.....	24
5.3.1	Het OMT model	24
5.3.2	Het relationeel representatiemodel.....	26
5.3.3	Het relationeel implementatiemodel	26
5.3.4	Diagram	27
5.4	Ontwikkeling van de overige pilots.....	29
5.4.1	De verschillende taakdiagrammen	29
5.4.2	De styleguide en het prototype gebruikersinterface onder de loep	29
5.4.3	De XUAN modellen nader bekeken.....	32
5.4.4	De aanpassingen met betrekking tot de globaal organisatorische inrichting	32
6	Testplan en testrapport.....	34
6.1	Inleiding	34
6.2	Waarom SUMI?.....	34
6.3	Hoe zijn detail aspecten getest?	35
6.4	Hoe moeten de gegevens verwerkt worden?.....	36

7	De fictieve invoering beschreven.....	37
8	Evaluaties	38
8.1	Productevaluatie	38
8.1.1	De documentatie	38
8.1.2	De database.....	38
8.1.3	Het testplan.....	38
8.2	Procesevaluatie.....	39
8.2.1	Projectaanpak	39
8.2.2	ASP.NET in het kort.....	39
8.2.3	De database en zijn lange ontwikkeltijd.....	40
8.2.4	Wijziging stagebegeleider.....	41
8.2.5	Veranderingen in het traject met betrekking tot Tickotel	41
8.2.6	Het bedrijf is veranderd	41
9	Literatuurlijst.....	42
9.1	Boeken	42
9.2	Websites	42
10	Verklarende woordenlijst	43
11	Bijlagen	44

1 Inleiding

In dit verslag zal ik mijn afstudeerproject beschrijven, hoofdzakelijk zal het proces worden besproken. Het proces zal beschreven worden aan de hand van de producten die ik heb opgeleverd.

Ieder hoofdstuk bevat gedetailleerde uitleg hoe sommige keuzes zijn gemaakt en hoe deze binnen het project van toepassing waren.

In hoofdstuk 2 komt de achtergrondinformatie aan bod waar ik informatie geef over de huidige en toekomstige situatie binnen Peter Langhout Reizen. Hiermee zal de context van mijn project duidelijker worden.

Hoofdstuk 3 geeft uitleg over de totstandkoming van mijn opdrachtomschrijving procesmatig beschreven. Hierbij geef ik uitleg waarom ik bepaalde keuzes heb gemaakt.

Hoofdstuk 4 bevat het eerste grote document van de IAD ontwikkelmethode en zal uitvoerig beschreven worden. De definitiestudie bevat allerlei informatie die van belang is voor de opdrachtgever en de afstudeerder. Hoe deze informatie tot stand is gekomen wordt hier beschreven.

In hoofdstuk 5 zal de fase pilontwikkeling beschreven worden, deze fase bevat het pilotontwikkelplan met daarin de pilots. Met name de kernel pilot; database zal uitvoerig beschreven worden en iedere stap zal onderbouwd worden.

Hoofdstuk 6 beschrijft het testplan en hoe ik tot het testplan ben gekomen. Dit testplan zal gebruikt worden wanneer Snowweb af is. Hiermee zal de usability van Snowweb aan een aantal rigoureuze tests worden onderworpen om te kijken of Snowweb voldoende gebruikersvriendelijkheid biedt.

In hoofdstuk 7 wordt de fictieve invoering besproken, waar ik niet bij zal zijn en ook geen rapport van zal opleveren. Ik geef hier mijn eigen visie weer op de manier waarop de invoering in mijn ogen zou kunnen geschieden.

Hoofdstuk 8 bevat de proces- en productevaluatie. Hier beschrijf ik alle nevenactiviteiten die ik heb gedaan en hoe het proces binnen het bedrijf Peter Langhout Reizen is verlopen. Ik zal hier ook toelichten hoe ik wellicht de dingen beter kan aanpakken in een volgend project.

In hoofdstuk 10 staat een verklarende woordenlijst voor sommige woorden die ik heb gebruikt. Deze woorden verdienen meer uitleg.

In het laatste hoofdstuk van dit document staan de bijlagen. Deze geven meer informatie bij bepaalde onderwerpen die in mijn verslag voorkomen. De bijlagen zijn als volgt gedefinieerd:

- bijlage 1: Organogram van Peter Langhout Reizen
- bijlage 2: Databaseregels
- bijlage 3: Gedetailleerde taakdiagrammen
- bijlage 4: Usability aspecten

2 Achtergrondinformatie

2.1 Inleiding

In dit onderdeel zal ik wat meer uitweiden over de achtergrond van mijn afstudeeropdracht en Peter Langhout Reizen in het algemeen.

2.2 Over Peter Langhout Reizen

Peter Langhout Reizen is een reisorganisatie, gevestigd in Alphen a/d Rijn. Het bedrijf is opgericht in 1987. De organisatie is na vele reorganisaties geslonken van 140 werknemers naar 90 werknemers die efficiënter werk moeten afleveren. Peter Langhout Reizen organiseert diverse vakanties voor een brede doelgroep in alle soorten en maten bestemmingen in Europa, USA en nog enige andere werelddelen. Het bedrijf is een 'direct seller', wat betekent dat de klanten direct kunnen boeken zonder tussenpersoon.

2.3 Totstandkoming van de afstudeeropdracht

Peter Langhout Reizen is klanten aan het verliezen op het wintersportgebied. Dit omdat de meeste mensen hun eigen accommodatie boeken en er daarna met hun eigen spullen naar toe rijden en daar hun vakantie vieren. Peter Langhout Reizen biedt op dit vlak alleen maar pakketreizen aan.

Peter Langhout Reizen wil met een webmodule de markt heroveren, en het zou een module moeten worden waarbij de klanten hun reis geheel zelf zouden kunnen samenstellen zonder enige tussenkomst van Peter Langhout Reizen of andere instanties.

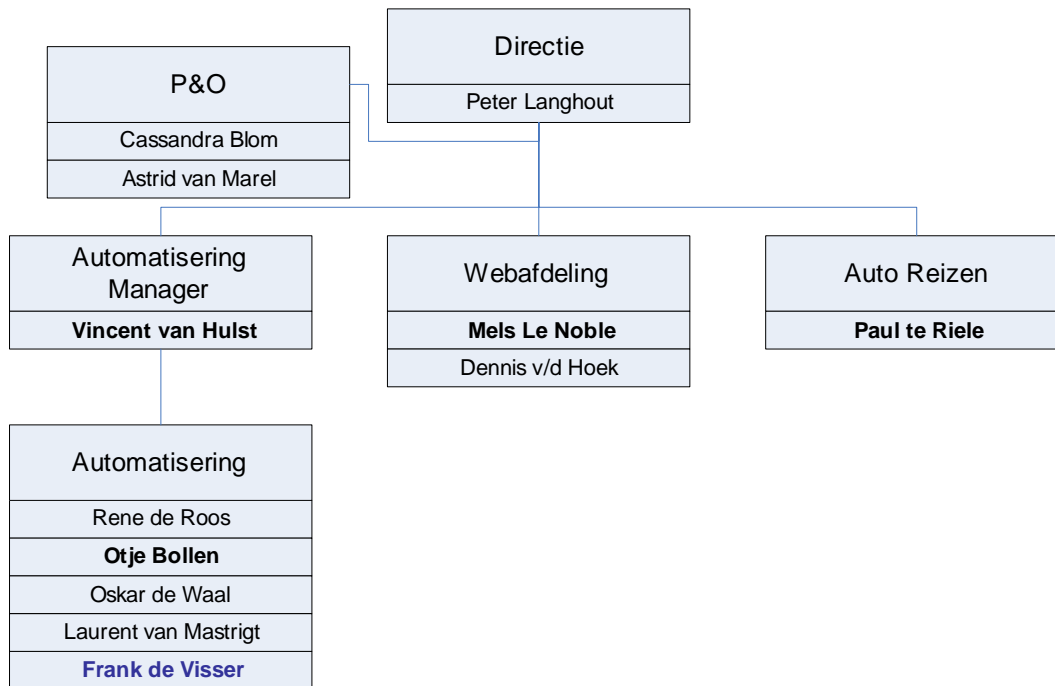
Deze module heeft de naam Snowweb gekregen en is aan mij toevertrouwd om geheel te definiëren, zo specifiek mogelijk, en om de portal van de reizigers te maken. Het bedrijf Tickotel zal alles aan de hand van mijn documentatie en sturing gaan maken.

2.4 Organisatorische inrichting

Het externe bedrijf dat mij zal bijstaan in dit project zijn in het vorige stukje kort beschreven. Ik zal hier een beeld scheppen hoe de interne organisatie mij heeft bijgestaan tijdens het project.

Omdat dit project voornamelijk documentatie- en database werk bevat is ervoor gekozen om mij te plaatsen binnen de afdeling automatisering waar ook de programmeurs, database experts en mijn stagebegeleider toe behoren. Hierdoor kon ik optimaal allerhande kennis aanspreken voor het welslagen van mijn project.

Hieronder het organogram met daarin mijn rol binnen de organisatie aangegeven, om een duidelijker beeld te scheppen, overigens laat dit figuur niet alle afdelingen zien, maar slechts de relevante. Ik heb hierin ook alle personen aangegeven die voor mij van belang waren voor mijn project en later in dit verslag nog terug komen (deze zijn vetgedrukt).



Figuur 1

In de bovenstaande figuur is te zien dat ik ben ingedeeld bij Automatisering. Hier heb ik directe hulp van ICT Manager Vincent van Hulst en database-expert Otje Bollen. De opdracht is vanuit Peter Langhout Reizen afkomstig en heeft in feite als contact persoon dhr Paul te Riele. Hij heeft de opdracht opgezet, maar is met de afhandeling, ná de definitiestudie niet bezig geweest en heeft slechts als 'start-up' voor het project gefungeerd. Vincent van Hulst de ICT Manager heeft zijn rol daarna overgenomen, omdat Dhr P. te Riele het te druk kreeg. Dhr P. te Riele is in het begin belangrijk geweest, omdat hij met het complete idee 'Snowweb' was gekomen. Hij wilde ervan profiteren met zijn afdeling Auto Reizen, want de meeste mensen zullen eerder geneigd zijn met de auto op wintersport te gaan.

De interface die later bovenop de functionaliteit zal worden gezet, wordt ontwikkeld door de 'webafdeling'.

Op een rijtje zijn de volgende personen van belang geweest, en zal ik ze als volgt benoemen in de rest van mijn documentatie:

Dhr Ing. V.G. van Hulst; ICT Manager, Stagebegeleider
Mevr. Ing. O. Bollen; Database-expert, Stagebegeleider
Dhr P. te Riele; Opdrachtgever
Dhr M. Noble; Webafdeling

Voor het totale organogram van het gehele bedrijf verwijs ik naar bijlage 1.

2.5 ICT Situatieschets

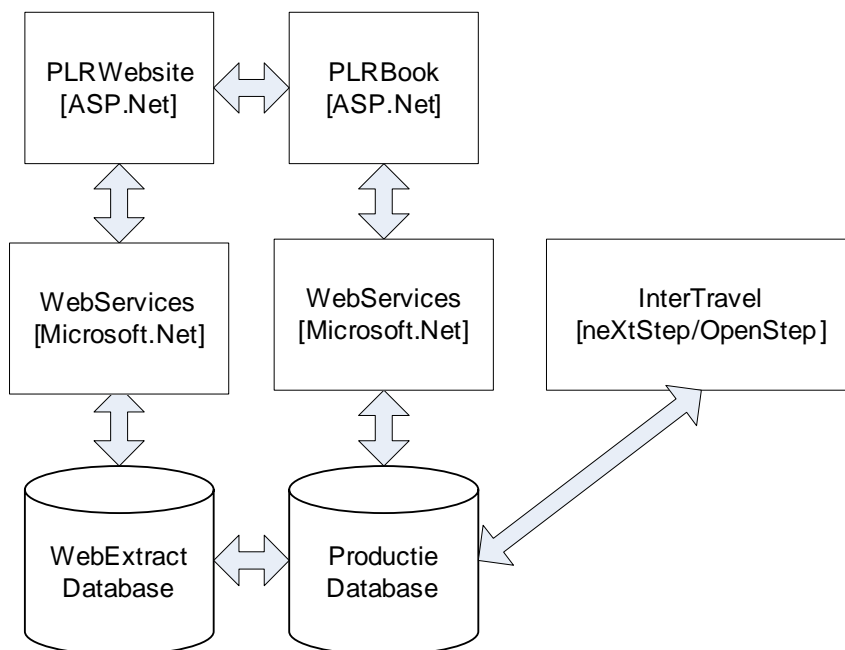
2.5.1 De oude situatie weergegeven

Om een duidelijker beeld te scheppen met betrekking tot de ICT situaties binnen Peter Langhout Reizen, zal ik hier zowel de oude (huidige) als de nieuwe (toekomstige) situatie beschrijven.

In de tijd dat ik hier stage loop bij Peter Langhout Reizen is de structuur van de complete ICT onderhevig aan veranderingen. Dit omdat Peter Langhout Reizen overgaat op een compleet nieuw systeem om reizen in op te slaan, aan te bieden en te verwerken. Hier zal uiteindelijk ook mijn module op komen te draaien en de database die ik aan het ontwikkelen ben zal dan in dit nieuwe systeem worden opgenomen.

Op dit moment zijn de veranderingen nog niet doorgevoerd bij Peter Langhout Reizen en ziet de situatie er als volg uit; het bedrijf kent een technische infrastructuur die opgebouwd is uit een groot aantal verschillende componenten. De websites en de boekingsapplicatie worden volledig door Microsoft.Net aangestuurd. Daarnaast is er de kantoor applicatie InterTravel. Dit wordt door de PLR medewerkers gebruikt om de reizen en overige gegevens in te voeren en is geschreven in nextstep/openstep.

De volgende schets geeft weer hoe deze huidige structuur van de website van Peter Langhout Reizen er globaal uit ziet. Hierin zijn de meest relevante onderdelen weergegeven.



Figuur 2

Zoals in deze figuur te zien is worden de PLRBook (de boekingsapplicatie) en de PLRWebsite aangestuurd door Webservices. Deze webservices maken het mogelijk dat iedere applicatie een correcte verbinding met de database(s) kan leggen en zodoende van de vooraf gedefinieerd stored procedures kan gebruiken via XML. Dit maakt Webservices uitermate geschikt voor meerdere soorten sites in verschillende talen, die allemaal via XML (Webservices) verbinding maken met de database en de webservices op hun beurt de resultaten terug geven via XML.

Er zijn twee databases vanwege veiligheidsredenen. De databases zijn gelijk aan elkaar qua inhoud. De database webextract is echter 'read-only' en de websites maken gebruik van deze database. Hiermee wordt voorkomen dat er van buitenaf acties kunnen worden gedaan op deze database. De productiedatabase daarentegen wordt gebruikt door InterTravel en de boekingsapplicatie. Deze database is altijd up-to-date. Een ander voordeel van de twee databases is de ontlasting van de Webservices laag, omdat deze gescheiden functioneren.

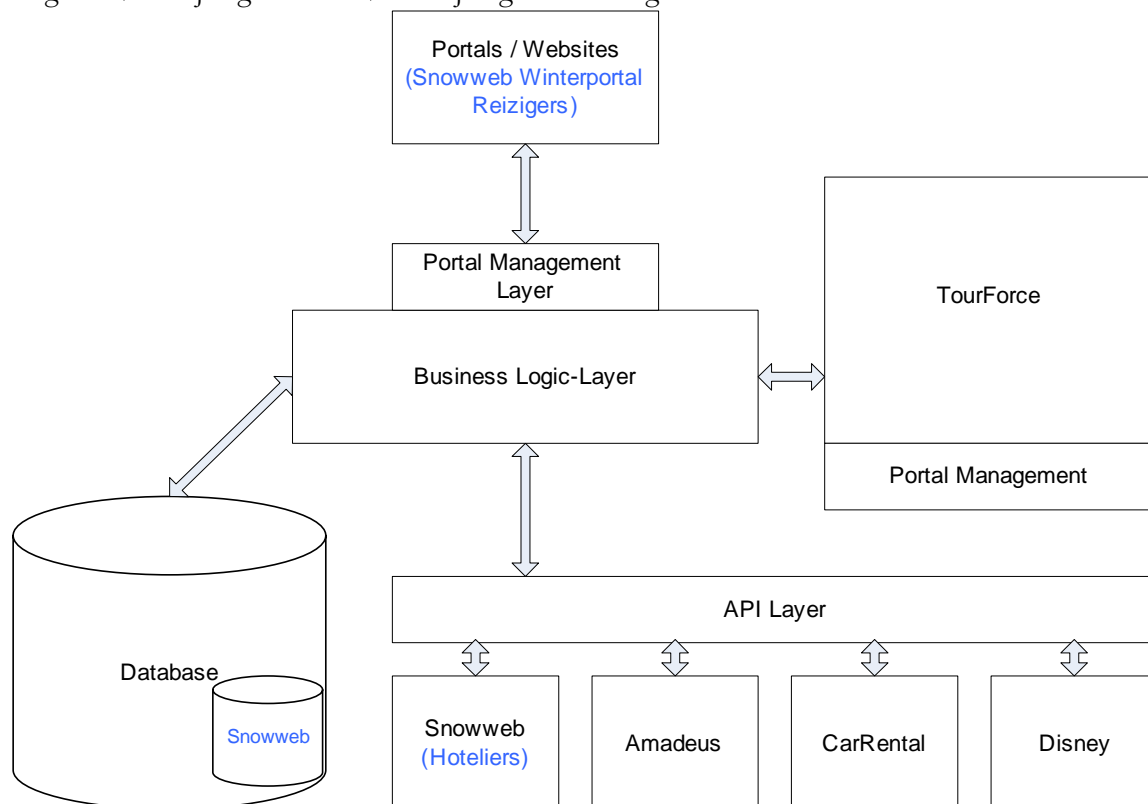
Zo werkt het huidige systeem van Peter Langhout Reizen op een veilige manier. Echter mijn module zal geïmplementeerd worden in de nieuwe toekomstige situatie.

2.5.2 De nieuwe situatie weergegeven

Om een beter beeld te geven van mijn module die in de nieuwe structuur zal komen, zal ik ook een schets maken van het nieuwe systeem. Het nieuwe systeem van Tickotel zal vele dingen moeten opvangen die op dit moment een probleem vormen met het huidige InterTravel systeem.

Tickotel is het bedrijf dat door Peter Langhout Reizen is gecontracteerd, vanwege hun 'betere' systeem ten opzichte van het huidige systeem. Dit nieuwe systeem wordt ook wel 'TourForce' genoemd. Het grootste minpunt van het huidige systeem is de beperking dat er alleen pakketreizen aangeboden worden. Met het nieuwe systeem moet het mogelijk zijn om de klanten zelf hun reizen te laten samenstellen.

De figuur hieronder geeft de nieuwe toekomstige situatie bij Peter Langhout Reizen weer. Op de volgende bladzijde geef ik een beschrijving van deze figuur.



Figuur 3

Tourforce is in deze figuur de backoffice applicatie die het huidige InterTravel gaat vervangen. Dit is een complete touroperator oplossing die van a tot z de verschillende administratieve processen ondersteunt, denk hierbij aan de inkoop van vervoer, verblijf en vermaak tot facturering naar de klant.

De API Layer is een interfacelaag die er voor zorgt dat third party's zonder veel inspanning aangesloten kunnen worden op Tourforce. Daarom staat "Snowweb (Hoteliers)" hier ook aangegeven, omdat deze ook gebruik zal gaan maken van third party databases.

De business logic layer draagt zorg voor alle zakelijke beslissingen, en dit gaat vrij ver. Denk hierbij aan het zelf samen stellen van een reis. Als mensen zelfstandig het transport mogen kiezen, dan moet er nagedacht worden over welke keuzemogelijkheden ze krijgen, zo wil je niet dat iemand die naar Salou gaat, in Madrid zal landen met het vliegtuig. Verder kun je denken aan yielding. Als de vraag stijgt, dat automatisch de prijs toeneemt, etc.

In de portal management layer wordt bepaald via welke portal het product zal worden verkocht. In mijn geval zal dit de snowweb portal zijn, en zal het product vanuit de snowweb database komen en dan via deze layer aan de klant aangeboden worden.

Het portal management is in feite de interface die aan Tourforce gekoppeld zit waarmee men de verschillende portals zou kunnen bewerken, denk hierbij aan functionaliteit en opties.

Ik zal hierbij dus de portal Snowweb, de API hoteliers en de complete database voor snowweb uitwerken en deze worden dan geïntegreerd binnen het gehele Tourforce systeem.

3 Opdrachtomschrijving

3.1 Totstandkoming van de probleemstelling

Peter Langhout Reizen heeft de afgelopen jaren te kampen gehad met een dalende wintersportmarkt. Dit is gekomen doordat men tegenwoordig alles zelf regelt en niet langer een zogenoemde pakketreis boekt, waar hotel, skipas, skiverhuur en vervoer al in zit. Men is meer geïnteresseerd in 'kleine' pensionnetjes, zelf reizen met de auto en zelf bepalen in welk gebied je gaat skiën.

Hierdoor is men gaan denken aan een manier waarbij Peter Langhout Reizen zelf deze reizen kan gaan aanbieden en de reiziger de volledige keuzevrijheid laat. Op dat moment kwam het idee Snowweb, wat een internationale portal zou moeten worden voor wintersporters.

3.2 Vervaardigen van een doelstelling

Het doel van de module was eigenlijk vrij simpel af te leiden vanuit de probleemstelling, namelijk het maken van een zo flexibel mogelijke reisboekingsmodule in ASP.NET.

Het probleem lag, zoals gezegd, bij de wintersportreizen, maar deze module zou natuurlijk ook voor andere doeleinden kunnen worden gebruikt, vandaar dat het een flexibele reisboekingsmodule is geworden. Echter de prioriteit lag bij de wintersportmarkt, en dat gaf mij ook een veel concreter beeld, waardoor het makkelijker was om me iets voor te stellen bij het project.

De belangrijkste factor in de doelstelling die ik eruit kon halen was eigenlijk; is dit project haalbaar voor mij binnen het gegeven tijdsbestek. En na een blik op de doelstelling leek het mij een prima uitdaging voor mijn afstudeerstage, ook vanwege de taal ASP.NET en het intensieve werken met Microsoft SQL Server.

3.3 Welke informatie was er beschikbaar?

Op het moment dat het traject van start was gegaan, was er al informatie in de zin van andere websites die al een soortgelijk product hadden en dit op een soortgelijke manier verkochten. Voor de techniek ben ik veel gaan kijken op *nachtjeweg.nl* en heb ik dit ook door de module heen als een voorbeeld gebruikt.

Er was ook een sneeuwsite in Amerika: *Snowweb.com*, die werd aangehaald door de opdrachtgever, maar deze vond ik niet geheel relevant. Wel omdat het over wintersport ging, maar het had absoluut niets met de techniek te maken. Dit heb ik zelf ook aangegeven bij de opdrachtgever, en deze vond dat ik wel even moest kijken op de site en daarna zien wat ik ermee deed.

De overige informatie was de informatie die ik direct vanuit Peter Langhout Reizen kon verkrijgen, deze was echter minimaal omdat men nog niet eerder een dergelijk project had gedaan.

Ik heb ook eerst besloten om eens goed op internet te zoeken naar dergelijke sites, maar kwam niet veel verder dan *nachtjeweg.nl* of *weekendjeweg.nl*.

Deze informatie heb ik gebruikt om een richtlijn te hebben voor de ontwikkeling van Snowweb.

3.4 Waarom de IAD methode?

Aan het begin van de stage heb ik gekozen voor de IAD-beheersmethode om het geheel projectmatig aan te kunnen pakken. Er bestaan ook vele andere methoden, maar van deze methode heb ik de meeste kennis en is in overleg met de ICT Manager goedgekeurd voor het project. De methode bestaat uit de volgende 3 fases:

- Definitiestudie
- Pilotontwikkeling
- Invoering

Binnen deze methode bestaan een aantal ontwikkelmethodes, bij het incrementeel ontwikkelen (RAD) worden de systeemeisen éénmalig in kaart gebracht; er wordt dus éénmalig een systeemconcept opgesteld. Vervolgens wordt het hele systeem gebouwd in een aantal iteraties tijdens de fase pilotontwikkeling, zonder dat er invoeringen plaatsvinden. Wanneer alle pilots ontwikkeld zijn, wordt het systeem als geheel in één keer ingevoerd.

Dit is een populaire vorm van de IAD-ontwikkel methode. De reden dat deze techniek veel gebruikt wordt, ligt in het feit dat deze manier van ontwikkelen heel snel gaat; de documentatie is bij deze vorm minimaal. Incrementeel ontwikkelen wordt ook wel aangeduid met RAD, Rapid Application Development. Omdat deze manier van ontwikkelen het meeste aansluit bij de wensen van de organisatie en de mogelijkheden van mij als afstudeerder (beperkte ontwikkeltijd) heb ik dan ook voor RAD gekozen.

Normaal gesproken wordt er ook een iteratie gedaan, maar omdat later bleek dat ik geen fysieke module aflever kan ik deze fase helaas niet doen. Ik heb geen pilots om te testen en vervolgens nog een keer mijn pilotontwikkelplan te maken.

3.5 Verbetering opdrachtoomschrijving

Na ongeveer een kwart van mijn stage te hebben gedaan is mijn opdrachtoomschrijving nog een keer herzien en bijgesteld. Dit had voor mij tot gevolg dat ik meer zou documenteren, dan programmeren. Tickotel zou alles ontwikkelen en ik zou hiervoor alle documentatie leveren.

Inmiddels was gebleken dat het project te groot was voor een afstudeerder. Dit bleek alleen al uit de hoeveelheid documentatie die nodig zou zijn om alles vast te leggen. Daarom werd de opdrachtoomschrijving ook bijgesteld. Dit gaf mij meer tijd om alles voor te bereiden en goede gedetailleerde documentatie te schrijven. Tijdens de bespreking met de leraren, die na de wijziging volgde, heb ik ook meer inzicht gekregen in de gehele situatie, zoals uitgelegd eerder in dit verslag bij de ICT Situatieschets. De ICT Manager heeft mij stap voor stap door het schema gepraat, waar dit eerder nog niet duidelijk was.

Mijn rol was in feite veranderd en de aanpak moest ook iets anders. Ik had nu in feite de verantwoording gekregen over andere programmeurs, dan slechts mijzelf. Ik had de rol van analist gekregen, degene die het ontwerp van de applicatie helemaal opstelt. Dit gaf mij ruimte, omdat ik niet een nieuwe taal hoefde toe te passen.

4 Definitiestudie

4.1 Inleiding

De definitiestudie heeft als doel een verantwoorde keuze te kunnen maken voor een bepaald systeem, door alle functionele en technische eisen op een rijtje te zetten. In de definitiestudie bevinden zich, de systeemeisen, het systeemconcept en de te ontwikkelen pilots. Na goedkeuring van de definitiestudie kan worden begonnen met de gedetailleerde pilotontwikkeling.

In dit hoofdstuk zal ik een beeld schetsen waarom ik bepaalde keuzes heb gemaakt tijdens de ontwikkeling van de definitiestudie en hoe het verdere proces is verlopen.

4.2 Opstellen plan van aanpak

Het plan van aanpak is het eerste onderdeel dat gemaakt moet worden volgens de IAD methode. Hierin worden de volgende onderdelen besproken:

- Probleemstelling
- Doelstelling
- Randvoorwaarden
- Beheersfactoren
- Gewenste eindresultaat
- Te gebruiken methoden en technieken
- Producten en activiteiten
- Fasering en planning

Het plan van aanpak is bedoeld om een duidelijk beeld te geven over wát er gaat gebeuren, hoe dit gaat gebeuren en wanneer dat gebeurt.

De probleemstelling en doelstelling had ik al vanuit de opdrachtschrijving gehaald, deze waren goedgekeurd door school en intern door de ICT Manager van Peter Langhout Reizen en ik heb daarop besloten deze niet aan te passen.

De randvoorwaarden en beheersfactoren kwamen direct vanuit de eisen van het bedrijf.

Het ging in dit plan van aanpak vooral om de planning, deze was belangrijk voor het gehele traject. Omdat ik eerder was begonnen met de afstudeerstage dan normaal gesproken gebeurd had ik een ruime planning opgesteld. Ik heb bij het maken van de planning de ervaring gebruikt van de ICT Manager die al eerder te maken had met dit soort projecten geven. Het feit dat ik meer ruimte in had gebouwd berustte ook op het feit dat ik kennis wilde maken met de programmeertaal ASP.NET en hier meer over wilde leren. Met een ruime planning had ik de tijd om hier en daar wat boeken te lezen en wat sites of programma's te schrijven in die taal. Door deze kleine applicaties heb ik, in feite buiten de opdracht om, toch wat kennis opgedaan over ASP.NET.

4.3 Hoe zijn de systeemeisen tot stand gekomen?

Alle systeemeisen die zijn vastgelegd in de definitiestudie komen vanuit een vergadering over Snowweb samen met de ICT Manager en opdrachtgever. De vergadering eindigde als een brainstormsessie. Normaal gesproken ben ik gewend een advies uit te brengen over de eisen die de opdrachtgever op tafel legt, maar na enkele minuten kwam al boven tafel dat het handig zou zijn als we eerst alle ideeën per persoon op papier zouden zetten en die met zijn allen bespreken. Het was voor mijn gevoel dus niet echt een vergadering, maar een brainstormsessie.

De resultaten van deze sessie zal ik hier kort in een lijstje samenvatten:

- Snowweb moet duidelijk opgezet worden voor de gebruiker
- Snowweb moet snel zijn
- Snowweb heeft als doel de gebruiker zoveel mogelijk opties te bieden waardoor deze beter een persoonlijke keus kan maken i.p.v. een pakketreis.
- Snowweb moet gebruik maken van de Peter Langhout Reizen database voor de eerste informatie, en daarna pas verder zoeken.
- Snowweb moet flexibel zijn, zodat de module eventueel met een andere interface ook als een boekingsmodule voor andere dingen ingezet kan worden.
- Snowweb moet behalve een boekingsmodule zijn, ook gedetailleerde informatie kunnen aanbieden over sneeuwhoogtes, reisverslagen, en gebieden.
- Snowweb moet gebruik gaan maken van Third party databases.
- Snowweb moet mogelijkheden bieden voor Hoteliers om hun eigen kamers in te voeren

Als eerste in de definitiestudie ben ik uitgegaan van de doelgroep, dit omdat ik vanaf dat punt een beter beeld kan vormen over het geheel van de applicatie, en zodat de opdrachtgever ook meteen kan zien hoe de doelgroep in elkaar zit en daar eventueel marketing op kan aanpassen. Ik heb de doelgroep geformuleerd vanuit mijn eigen ervaring als fanatiek skiër en snowboarder, wat ik inmiddels al 16 jaar doe.

De doelgroep is in feite opgebouwd uit drie groepen, omdat van alle kanten van de module gebruikers zitten die de module voor een ander doel gebruiken. Hieronder een korte beschrijving van de drie doelgroepen:

Als eerste heb je de mensen die op zoek zijn naar een wintersportvakantie, waarbij ze alles zelf kunnen regelen, maar wel op één plek.

De leeftijd van deze groep zal over het algemeen tussen de 23 en 50 liggen. Het zal hier gaan om de wat meer ervaren wintersporter.

Deze mensen hebben doorgaans al met webinterfaces gewerkt, maar zullen geen IT professionals zijn. Hier moet dus goed rekening gehouden worden met de gebruikersvriendelijkheid van de interface en het gebruiksgemak.

De tweede groep mensen zijn de hoteliers en de inkopers van Peter Langhout Reizen. Deze instanties willen hun kamers op de website zetten, zodat de eerste groep deze kan boeken. Het betreft hier ondernemers met veelal een eigen hotel of pension met een leeftijd van ongeveer 27 tot 65. Hier geldt dat er weinig 'computerkennis' aanwezig is.

Als derde groep zijn er de medewerkers van Peter Langhout Reizen die de management functie moeten gaan bedienen. De leeftijd hiervan kan uiteenlopen van 20 tot 60. Degenen die de management functie gaan bedienen beschikken meestal over een redelijke hoeveelheid kennis van de techniek en computers in het algemeen.

Nadat de groepen waren vastgesteld kon ik verder met de eisen met betrekking tot de module. Uit de brainstormsessie was naar voren gekomen dat de module Snowweb de mogelijkheid aan Hoteliers zou moeten bieden, dat deze hun eigen kamers zelfstandig kunnen invoeren.

Qua usability en interface gold het volgende: De Hoteliers en reizigers mogen hun weg niet kwijt raken, of verkeerde dingen invullen. Wanneer deze fouten vaak voorkomen, dan moet het management van PLR teveel tijd besteden aan het corrigeren van andermans fouten. Daarom heb ik in de definitiestudie geschreven dat de usability eisen en interface eisen aan hoge eisen moeten voldoen. Hiervoor heb ik gebruik gemaakt van GUIDE (zie literatuur), met daarin de acht gouden regels van Schneiderman. Ter verduidelijking zal ik de 8 regels kort neerzetten:

- Striving for consistency (Het streven naar consistentie binnen de module)
- Enable frequent users to use shortcuts (Shortcuts voor veelgebruikte acties)
- Offer informative feedback (De feedback op een actie moet informatief zijn)
- Design dialogs to yield closure (Maak geen open dialogen, maar geef de gebruiker zo min mogelijk opties)
- Offer simple error handling (Geef begrijpelijke foutcodes en uitleg en eventueel oplossingen)
- Permit easy reversal of actions (Sta toe dat de acties makkelijk zijn terug te draaien)
- Support internal locus of control (Voorkom onverwachte systeemacties, de gebruiker wil controle houden)
- Reduce short-term memory load (Geef het menselijk brein niet teveel informatie om tegelijk te onthouden (de regel is 7 items tegelijk))

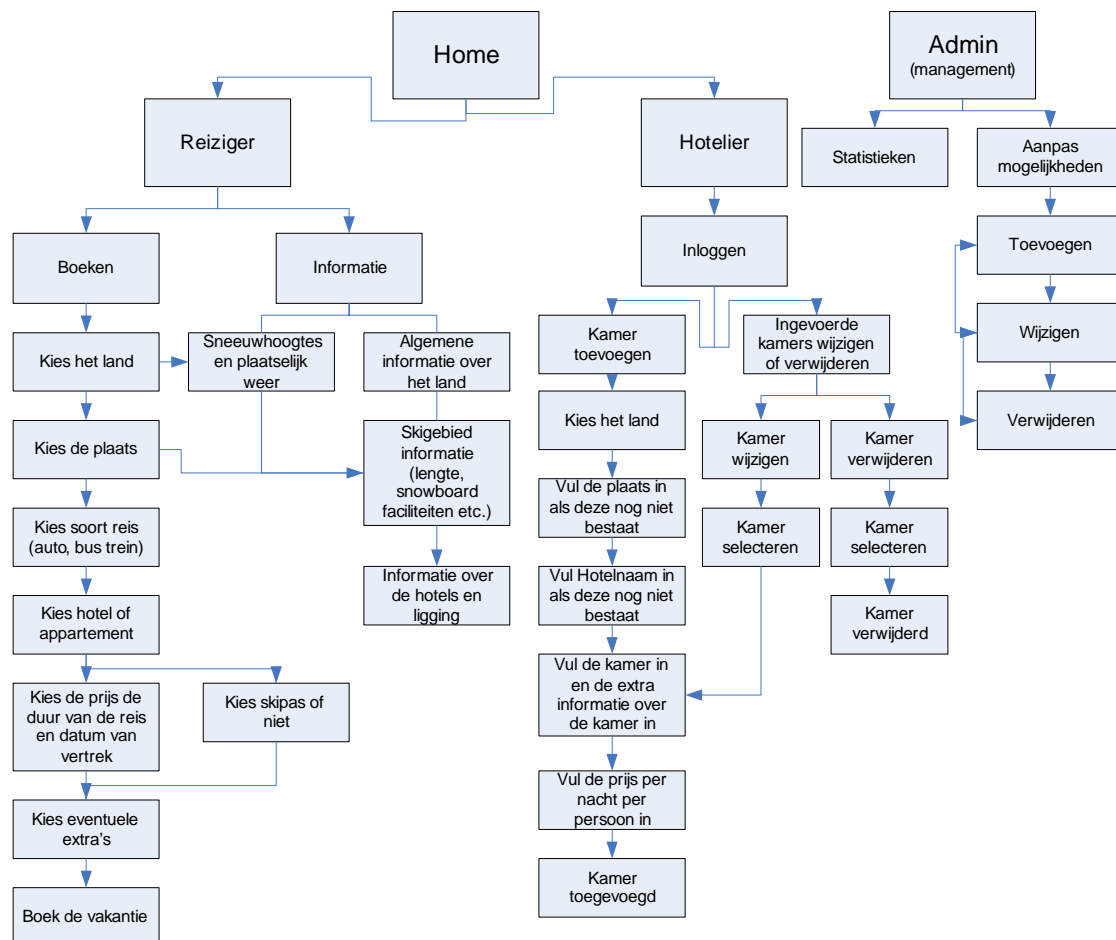
Deze 8 regels heb ik gebruikt als minimale eisen voor de usability en interface, om een zo hoog mogelijke kwaliteit te waarborgen. Daarnaast heb ik de 'wizard style' aangehouden. Uit mijn ervaring, die ik heb opgedaan toen ik part time webdesigner was bij een reclame bureau, is gebleken dat je mensen goed moet sturen, en via een 'wizard' is dat perfect te doen, omdat je ze steeds een zeer beperkte keuze geeft (2 of 3 keuzemogelijkheden), dit heeft ook weer te maken met de regel 'Design dialogs to yield closure'. Deze 'wizard' geldt overigens alleen voor de reizigers, aangezien de hoteliers meer vrijheid nodig hebben om hun kamers juist in te voeren. Zij hebben bijvoorbeeld een hotel of wellicht een camping, en hierin zit een duidelijk verschil wat niet direct vastgelegd kan worden. De verschillende factoren zijn constant anders. Bovendien zullen de hoteliers vaker een kamer invoeren en op den duur zal een 'wizard' dan irritatie oproepen, omdat ze steeds hetzelfde moeten doorlopen. Deze constructie was overigens ook een idee waar de ICT Manager al ervaring mee had en wat hij goedkeurde.

De performance van de module moest ook hoog liggen, omdat er van meerdere zijden van de module wordt ingevoerd en dit niet ten koste mag gaan van de snelheid, zodat klanten niet gaan klagen over de module zelf. Hiervoor zijn snelle servers en een 'schone' code nodig, beide aspecten lagen niet in mijn hand, maar zijn wel duidelijk door mij gedocumenteerd.

In de operationele eisen zijn opgenomen dat er gebruik wordt gemaakt van Webservices om het geheel snel te laten werken. Webservices stellen bestaande, ongelijksoortige applicaties op willekeurige locaties in staat samen te werken. Webservices zijn van belang bij het implementeren van de module binnen Peter Langhout Reizen, omdat alle modules en applicaties via deze Webservices gekoppeld zijn en query's kunnen uitvoeren op de centrale database. Webservices verhogen ook weer de performance van de applicatie, omdat men via externe applicaties (niet de website zelf) de stored procedures en databasehandelingen afhandelt en de resultaten naar de website worden gestuurd.

4.4 De vervaardiging van het systeemconcept

De volgende stap in de definitiestudie was het systeemconcept, het systeemconcept heb ik zelf gemaakt aan de hand van de gegevens die ik had verkregen uit de vergadering met de ICT Manager en de opdrachtgever, zoals eerder in dit hoofdstuk aangegeven. Het eerste navigatieschema had als volgt vorm gekregen:



Figuur 4

Dit schema in figuur 4 heb ik ontwikkeld aan de hand van de eisen en wensen die naar voren waren gekomen in de brainstormsessie en mijn eigen ervaring als webdesigner. Ook heb ik gebruik gemaakt van de kennis die ik had opgedaan door het kijken naar *nachtjeweg.nl*. Daarom heb ik ook voor de navigatie gekozen zoals hierboven beschreven. Ik ben uitgegaan van een 'narrow-down' situatie, zodat de wachttijden voor de gebruiker zo minimaal mogelijk blijven en de query's ook in de hand blijven. Er wordt bijvoorbeeld bij de reiziger eerst gezocht op het land, daarna de plaats, etc. Het wordt steeds kleiner, waardoor je niet alleen de reiziger goed kunt sturen, door hem een steeds beperktere keuze te geven, maar ook zelf het proces niet al te groot laat worden.

Het schema is na overleg met de ICT Manager goedgekeurd, maar later in de Pilotontwikkelplanfase bleken er toch wat haken en ogen te zitten aan dit navigatieschema. Sommige onderdelen bleken niet gedetailleerd genoeg. Daarom hebben de ICT Manager en ik ervoor gekozen om in het pilotontwikkelplan per pilot een nieuwe navigatie te maken en deze zo te laten omdat hij wel de situatie goed weergeeft, maar in mindere mate de scheiding tussen de verschillende onderdelen en een aantal details mist, zoals het snelzoeken van bepaalde hotels.

Hierna ben ik begonnen aan het objectmodel, wat al erg lijkt op het OMT Object model van de database, maar wel de objecten landen en gebieden bevat. Later bleken deze 2 objecten niet langer nodig binnen Snowweb omdat deze ook in de algemene database in Tourforce al worden opgenomen.

Hierbij had ik zelf ook bedacht dat een systeem waarbij de mensen een waardering aan de hotels kunnen geven ook van pas kan komen omdat het mensen een beter beeld geeft van wat goed is en wat niet goed is. Dit ben ik op meerdere sites tegengekomen en het geeft de mensen een wat beter idee van 'kwaliteit'. Dit heb ik voorgelegd aan de ICT Manager samen met het objectmodel en deze vond het een goed idee.

In het systeemconcept was ook het globaal storyboard opgenomen om de opdrachtgever een beter idee te geven van de functies en hoe deze waarschijnlijk in de praktijk door mensen werden afgehandeld. Daarvoor heb ik enkele scenario's geschreven hoe de doelgroep de module waarschijnlijk zou doorlopen. Hieronder een voorbeeld van een dergelijk scenario:

Boeken reis

Bij het boeken van een reis moet de reiziger een aantal stappen trapsgewijs doorlopen, waardoor hij op een 'wizard' wijze tot een boeking komt. Ik zal hieronder de stappen beschrijven.

De reiziger kiest eerst een land waar hij heen wil. Vervolgens de plaats in dat land en daarna de vervoersmethode. Er worden hem nu een aantal opties geboden van hotels met de kamers en de prijzen. Hij kan nu direct een kamer boeken. Zodra de kamer is gekozen en de duur van de reis gaat men naar de volgende keuze. Hier wordt gevraagd of er een skipas bij in moet zitten of niet, daarna worden de eventuele extra dingen nog gekozen en dit wordt uiteindelijk allemaal in een soort van winkelwagentje gegooit. Hierna kan de reiziger boeken.

De scenario's heb ik opgesteld aan de hand van mijn kennis die ik op school en in de praktijk heb opgedaan. Mijn OpenDuaal periode en de stages die ik heb gelopen bleken later van goede toepassing bij het doorontwikkelen van de module, omdat aan de hand van deze scenario's ook een test kon worden opgesteld.

4.5 Opstellen pilotplan

Als laatste belangrijke onderdeel is er het pilotplan in de definitiestudie. Het pilotplan geeft een korte beschrijving en planning per pilot en een kosten/baten analyse. Om tot goede pilots te komen is er nog een brainstormsessie gehouden waarin ieder onderdeel stuk voor stuk is besproken. De brainstormsessie zelf is kort geweest en heeft slechts een paar uitbreidingen op de vorige sessie gebracht. Hieronder kort een lijstje van de uitkomsten:

- De database moet volledig werken en voorbereid zijn op enige toekomstige aanpassingen.
- Uit de praktijk is gebleken dat een database super flexibel opgezet moet worden opdat er altijd extra items bij geplaatst kunnen worden in de tupels.
- De pilot Hoteliers zal aan de API komen te hangen van het model zoals Tickotel dat heeft opgesteld. (deze staat beschreven in figuur 3, paragraaf 2.5.2)

Met deze extra informatie ben ik begonnen aan het opstellen van het pilotplan. De pilots heb ik opgesteld aan de hand van de informatie die al eerder in de definitiestudie was besproken, en ik heb goed gekeken naar 'hoe' de gehele module mooi opgedeeld kon worden. Hieruit zijn de pilots reizigers, hoteliers en management gekomen, daar deze de meest bruikbare coherente delen vormen. De pilot database is geen officiële pilot, maar een 'kernel' pilot. Dit houdt in dat deze pilot ontwikkeld moet worden als basis voor de andere 'echte' pilots en dat is bij de database het geval.

Als belangrijk knelpunt voor de ontwikkeling van de pilots was daar de koppeling met de database van derde partijen zoals *Expedia.nl*. Op dat moment had ik nog geen zicht op de situatie, wanneer er andere databases bij zouden komen. Het nieuwe systeem dat geïntroduceerd zal worden door Tickotel zal via de API, zoals uitgelegd in het begin van dit document, een koppeling kunnen maken met de databases. Deze informatie had ik echter nog niet ten tijde van het schrijven van de definitiestudie, waardoor het voor mij een knelpunt vormde. Ik heb later na overleg met de ICT Manager deze niet verder uitgewerkt in het pilontwikkelplan.

De pilot Reizigers is voortgekomen uit de eerste sessie en hieruit was gebleken dat de reizigers het zo makkelijk mogelijk moesten hebben. De pilot Reizigers bestaat uit een interface die de informatie zo aan de reizigers aanbiedt dat deze het gemakkelijk kunnen selecteren en boeken. Verder zou er ook nog informatie opgezocht moeten kunnen worden, zoals de sneeuwhoogtes in een bepaald gebied. Om dit te kunnen realiseren moeten de PLR medewerkers deze informatie wel up-to-date houden. Ik heb namelijk vaker meegemaakt in het verleden dat ik hele mooie management systemen had gemaakt en daarbij konden bedrijven nieuws invoeren en dergelijke. Dit werd niet tot nauwelijks gebruikt. Hierdoor verouderd een website heel snel en haal je het dynamische eruit. Om de mensen te blijven trekken moet je nieuwe informatie blijven aanbieden. De ICT Manager heeft deze voorwaarde voor mij gecommuniceerd naar de organisatie.

De pilot Hoteliers was gedefinieerd als een interface waarbij de hoteliers een inlognaam en wachtwoord kregen en via deze hun kamers konden invoeren, wijzigen en verwijderen. Een belangrijk punt hierbij was dat de hoteliers in goede banen werden geleid bij het invoeren van de kamers, de interface en functionaliteit zijn hier ook van groot belang. Ze moesten echter ook de nodige bewegingsvrijheid krijgen, om alles zo correct mogelijk in te kunnen voeren. Om ervoor te zorgen dat de interface ook daadwerkelijk goed zou worden, zijn er afspraken gemaakt met de afdeling 'webdesign' om de interface zo goed mogelijk te maken. Een belangrijke afspraak hierbij was om de GUIDE methode als richtlijn te gebruiken zoals eerder besproken.

De management pilot is een pilot die eigenlijk bestaat uit een verzameling statistieken die vanuit de database worden gegenereerd. Deze pilot is bedoeld voor de medewerkers van Peter Langhout Reizen om de gegevens van de reizigers en hotels te kunnen controleren en eventueel aan te passen. Ook moeten de statistieken die vanuit de database zijn af te leiden worden weergegeven in de interface van het management. De management pilot heeft ook eigenlijk geen tabellen of dergelijke binnen de database. Ik heb ervoor gekozen om alle statistieken via bepaalde queries naar boven te halen i.p.v. hiervoor een extra tabel aan te maken, omdat dit weer ruimte scheelt in de database.

Vanuit de grove kosten en baten analyse die ik heb gemaakt is gebleken dat wanneer Snowweb online zou gaan Peter Langhout een grotere winst zou kunnen behalen. Dit komt eigenlijk voort uit een tweetal punten:

- Er is minder personeel nodig voor de inkoop van de accommodaties waardoor er kosten bespaard worden.
- De nieuwe manier van boeken en keuzes moet ervoor zorgen dat de mensen meer gaan boeken via Snowweb, op deze manier krijgt Peter Langhout Reizen meer omzet, omdat er meer mensen boeken dan voorheen het geval was dankzij het puur en alleen aanbieden van pakketreizen en het via deze module mogelijk is om verschillende keuzes samen te voegen.

Er zijn natuurlijk ook een aantal kostenposten waar aandacht aan besteed moet worden:

- De ontwikkeling van de module kost geld
- Er zullen hoteliers moeten worden aangetrokken om hun kamers te verkopen via Snowweb.
- De communicatie met transportmaatschappijen moet beter, omdat het hier geen vooraf ingekochte stoelen betreft, maar stoelen die 'on-site' worden geboekt.

De definitiestudie was op dat moment compleet en deze heb ik zo aangeleverd bij de opdrachtgever, deze heeft hem uitvoerig bekeken en goedgekeurd. Ik kon gaan beginnen aan het pilotontwikkelplan.

5 Pilotontwikkeling

5.1 Inleiding

In dit hoofdstuk zal ik beschrijven hoe de ontwikkeling van de documentatie van de pilots is verlopen en in het bijzonder de kernel pilot Database, zal ik uitgebreid beschrijven, aangezien ik hier het meeste werk in heb moeten steken. Ik heb tijdens de pilotontwikkelfase een compleet logboek bijgehouden waarin ik de verschillende problemen heb gedocumenteerd. Deze problemen zal ik hier bespreken.

5.2 Startproblemen

Toen ik het pilotontwikkelplan voor het eerst aan het maken was rezen al gelijk de problemen. Er was mij op dat moment nog niets bekend over de Third Party koppeling met een database die zou komen. Ik heb in overleg met mijn stagebegeleider besloten om de Pilot Third Party daarom ook weg te laten, deze is later ook niet meer teruggekomen in het pilotontwikkelplan, omdat ik deze niet meer zou maken. De koppeling zou via de API van Tickotel gebeuren, zoals in de vorige hoofdstukken uitgelegd.

Tot op het moment dat de opdrachtomschrijving voor de tweede keer werd besproken en gewijzigd had ik veel open vragen die op dat moment niet beantwoord werden. De definitiestudie was goedgekeurd, maar wat er nu precies moest gebeuren met de pilots was mij niet compleet duidelijk. Nadat de opdrachtomschrijving was besproken, heb ik een veel beter beeld gekregen. Er werden mij allerlei duidelijkere taken toegewezen en een aantal diagrammen zoals terug te vinden in de ICT Schets verhelderden de situatie voor mij nog meer.

Op dat moment had ik eigenlijk de taken liggen die ik moest gaan doen, de belangrijkste taken hierbij waren:

- Het pilotontwikkelplan moest volledig zijn en zo duidelijk dat Tickotel er direct mee aan de slag kon, want het was niet langer het geval dat ik daadwerkelijk zou gaan programmeren, maar meer een fictieve module zou neerzetten volledig uitgewerkt op papier.
- De database moest zeer goed gemaakt worden, omdat dit het hart zou zijn van de hele module.
-

Hierbij moet worden opgemerkt, dat de Third Party koppeling volledig weg viel, omdat Tickotel deze al via een API in Tourforce ging koppelen.

Met deze twee punten heb ik besloten een kwalitatief en gedetailleerd pilotontwikkelplan te maken, wat iedere programmeur kan aanspreken als hij gaat ontwikkelen. Daarnaast had ik als doelstelling een haast perfecte database te maken. Daartoe heb ik een extra document naast het pilotontwikkelplan gemaakt dat de database volledig tot in de puntjes beschrijft.

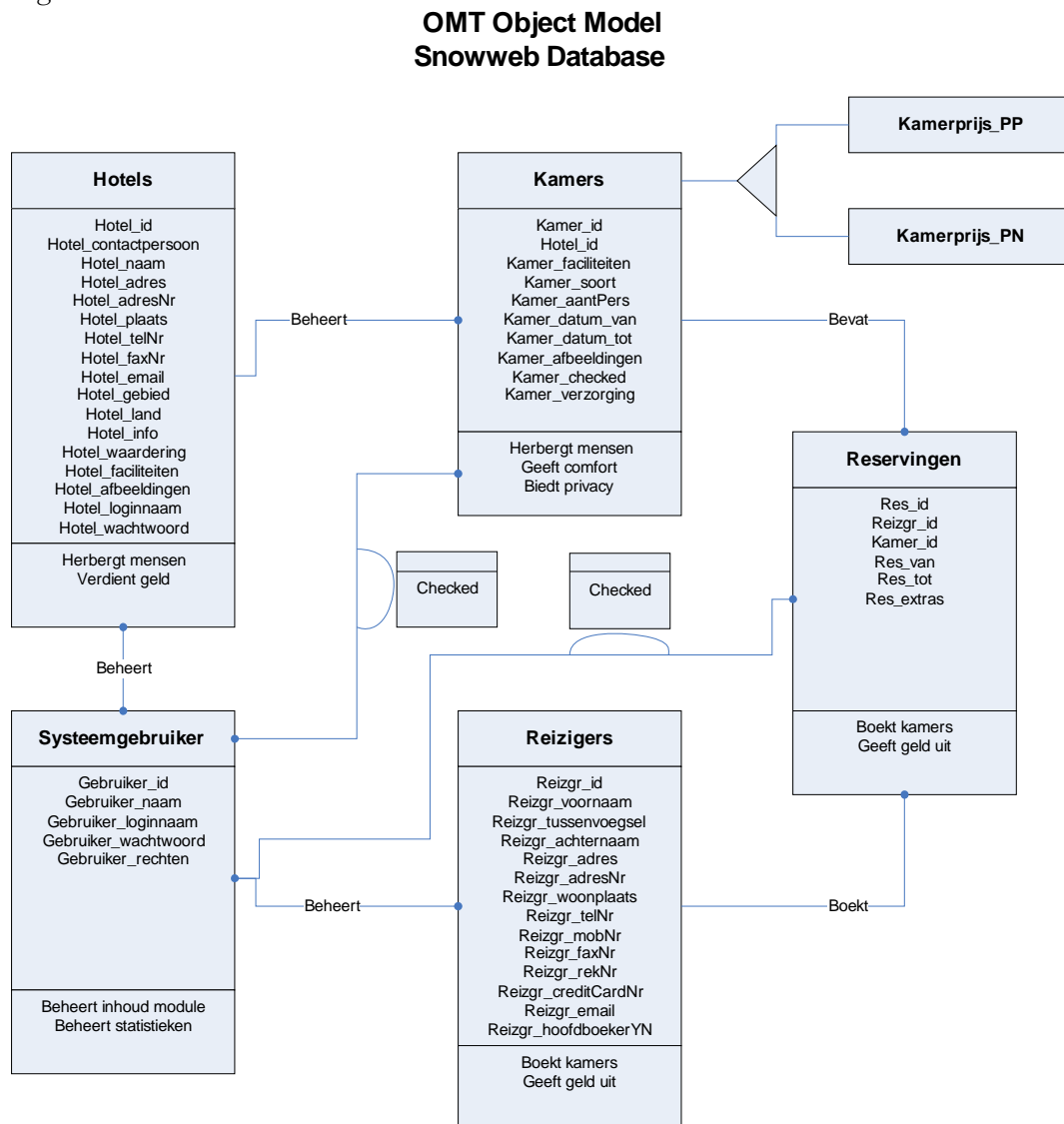
Vanaf dit punt kon ik daadwerkelijk met concrete eisen en doelen aan de slag.

5.3 Ontwikkeling van de database

Toen ik net begonnen was aan de stage had ik het idee dat ik 'wel even een database' zou maken. Ik had in het verleden al databases gemaakt, maar deze waren nooit zo uitgebreid geweest als deze, maar ik had geen enkel vermoeden dat het zoveel werk zou zijn. Ik ben na het voorgaande gesprek begonnen aan het database document . Meer over de problemen die ik tegen ben gekomen tijdens de database ontwikkeling zijn te vinden in de procesevaluatie aan het einde van dit verslag.

5.3.1 Het OMT model

Nadat mijn opdrachtschrijving was aangepast ben ik begonnen aan de database. Ik heb daartoe de volledige situatie bekeken en de verschillende objectmodellen die ik al had gemaakt voor de definitiestudie gebruikt om een zo gedetailleerd mogelijk document te maken. Ik ben begonnen met het OMT Model wat hieronder te zien is.



Figuur 5

Ik heb dit model gemaakt met behulp van het boek: Object-Oriented Modeling and Design. In dit boek worden een tweetal modellen beschreven. Ik was eerst begonnen aan een ER model, maar toen las ik het volgende in het boek: *“We have compared OMT object modeling to the entity-relationship (ER) information modeling notation. In essence, OMT object modeling is an enhanced form of ER. OMT object modeling improves on ER in the areas of expressiveness and readability.”* Ik heb hierna ook zelf de modellen bekeken en vergeleken en inderdaad bleek voor mij ook het OMT model duidelijker, daarom heb ik besloten om een OMT model te maken, want alles moet zo duidelijk en overzichtelijk mogelijk zijn.

Het OMT model is bedoeld om op een grafische en overzichtelijke manier de werkelijkheid aan te geven voor de module. Zoals te zien is zijn onderdelen zoals kamers, hotels, reizigers en reserveringen opgenomen. Er is ook een tabel systeemgebruiker. Dit is eigenlijk het management van Peter Langhout Reizen. Deze tabel wordt echter niet gebruikt in de uiteindelijke Snowweb database, omdat de systeemgebruikers al ergens anders gedefinieerd staan door Tickotel. Ik heb hem er wel in gezet om duidelijkheid over de situatie te geven en hoe deze gebruikers de inhoud controleren.

Ik zal in het kort het model uitleggen qua acties:

- De reiziger reserveert kamer.
- Een reservering bevat een kamer passend bij een hotel
- Een kamer wordt beheerd door een hotel
- De kamerprijs wordt gesplitst in Prijs per Nacht (PN) of Prijs per Persoon (PP)
- De systeemgebruiker beheert alle tabellen*

*Hoe het beheer van de tabellen precies gaat werken is niet bekend, of de controle ging via dubbele tabellen voor de hotels en kamers, of dat er een aparte database exact hetzelfde werd gemaakt die na verloop van tijd werd gesynchroniseerd met de live database. Ik ben van het tweede geval uit gegaan, aangezien PLR dit systeem al gebruikt (database synchronisatie tussen de webextract database en de productie database).

Ik had in de eerste modellen ook de gebieden, sneeuwhoogten en steden opgenomen. Dit is er echter uitgehaald nadat ik mijn eerste model had voorgelegd aan de database-expert en mijn stagebegeleider, omdat deze al standaard in de Tourforce database zaten.

Ik ben tot dit huidige model gekomen door het constant perfectioneren van de modellen die ik maakte en deze te laten zien aan de database-expert. Hier heb ik misschien een trial-error methode gebruikt, maar ik wilde alles zo goed mogelijk hebben. En zo'n soort module was er nog nooit gemaakt binnen Peter Langhout Reizen, dus ook de database-expert moest af en toe even iets zien om vervolgens te kunnen corrigeren, in plaats van het uit het hoofd te doen.

Ik kwam ook het probleem tegen hoe ik de wijze van prijsberekening moest oplossen, want deze had ik eerst in dezelfde tabel gezet, maar later bleek dat ik ze er via een disjunctie uit moest halen. In het model staat het driehoekje ook voor disjunctie, wat zoveel betekent als; het is óf PN óf PP.

Voor de informatie om het OMT model goed te maken heb ik intensief gebruik gemaakt van hoofdstuk 12 in het boek 'Object-oriented Modeling and Design'.

Nadat het OMT model helemaal compleet was ben ik met de volgende stap begonnen, het relationele representatiemodel.

5.3.2 Het relationeel representatiemodel

Het relationeel representatiemodel geeft in feite in schrijftaal en in het kort weer hoe tabellen eruit komen te zien en wat hierbij de primary en foreign keys zijn.

Het model was meer voor mezelf bedoeld en om ook goed de opbouw te zien van de database. Echter ze zouden waarschijnlijk niet gebruikt worden ten tijde van het echte ontwikkelproces. Ze zijn wel handig om het uiteindelijke implementatiemodel terug te kunnen leiden. Ik heb het geheel vanaf de 0^e normaalvorm tot de Boyce-Codd normaalvorm genormaliseerd. Ik heb op internet nog gelezen over een 4^e en 5^e normaalvorm, maar vond dit onnodig en onbekend. Deze 2 normaalvormen worden alleen nog gebruikt in de wiskundige wereld om te bewijzen dat een database niet redundant is. Als je goed hebt genormaliseerd volgens de Boyce-Codd normaalvorm, dan is je database al optimaal en niet redundant.

Tijdens dit stadium ben ik ook bekend gemaakt met de 'huisregels' van Peter Langhout Reizen met betrekking tot het ontwikkelen van een database. Deze regels maakten het voor mij weer een stuk onduidelijker, omdat ik zelf een database had gemaakt en daarna alle naamgeving en dergelijke kon aanpassen. Uiteindelijk na een goede uitleg van de database expert, zijn het wel goede regels gebleken. Deze regels zijn te vinden in bijlage 2.

De regels hebben als grootste voordeel dat alles makkelijk te vinden is in de database, ook wanneer je deze niet hebt gemaakt. Het is een standaard binnen de organisatie.

5.3.3 Het relationeel implementatiemodel

Het relationeel implementatiemodel, is het uiteindelijke model wat precies weergeeft hoe de database eruit komt te zien in tekstvorm. Deze tekst zou je dan eventueel in de database als script uit kunnen voeren waardoor alle nodige tabellen en sleutels automatisch worden aangemaakt.

Omdat alles binnen PLR ontwikkeld wordt voor Microsoft SQL server, moest ik hier ook gebruik van maken. Daar heb ik de tabellen gemaakt en hierbij een diagram gemaakt (waar alle relaties tussen de tabellen worden aangegeven). Hiervan heb ik door MS SQL Server een script laten draaien waar precies netjes alle code wordt gegenereerd om de database opnieuw te maken via een script.

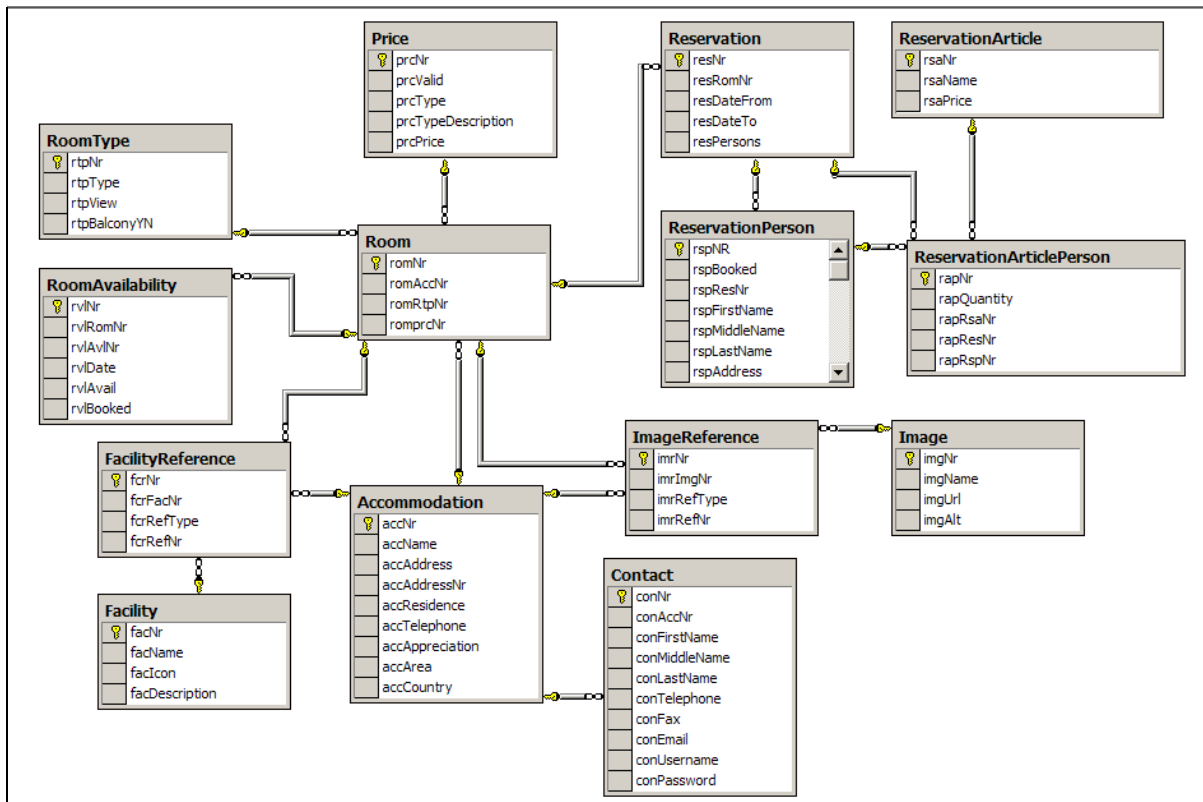
Ik heb gekozen voor deze manier, omdat de scripts die gegenereerd worden uniek zijn voor Microsoft SQL Server, maar wel een relationeel implementatiemodel zijn. Op deze manier heb ik de database voor altijd precies goed op papier staan, want er zal in de toekomst ook nog steeds gebruik gemaakt worden van Microsoft SQL Server.

Hieronder zal ik een klein stukje ter voorbeeld laten zien, zodat er een beeld ontstaat hoe zo iets er precies uit ziet.

```
CREATE TABLE [dbo].[Accommodation] (
    [accNr] [bigint] IDENTITY (1, 1) NOT NULL,
    [accName] [varchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [accAddress] [varchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [accAddressNr] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [accResidence] [varchar] (250) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [accTelephone] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [accAppreciation] [numeric] (18, 0) NULL,
    [accArea] [bigint] NULL,
    [accCountry] [bigint] NULL
) ON [PRIMARY]
GO
```

5.3.4 Diagram

Het diagram is in mijn stappenplan eigenlijk datgene wat ik eerder had gemaakt dan het relationeel implementatiemodel, om van daaruit het implementatiemodel te genereren. Echter het diagram hoort officieel later te komen dan het relationeel implementatiemodel en is slechts een grafische weergave van het relationeel implementatiemodel. Ik wil aan de hand van dit diagram een uitleg geven waarom er gekozen is voor bepaalde oplossingen.



Figuur 6

Zoals te zien is, zijn het aantal personen in Reservation aangegeven in 'resPersons'. Dit is natuurlijk redundant en dit had ik eerst niet in de database, maar de database expert raadde mij aan het er wel in te zetten, omdat het gewoon in de praktijk query's en snelheid scheelt.

Verder had ik ook een probleem met de oplossing voor plaatjes en faciliteiten. Ik had hier eerst aparte tabellen per Accommodation en Room gemaakt. Dit bleek echter niet efficiënt, want in dit geval wil je juist wel weer zo weinig mogelijk in de database opslaan. Dus is er in overleg met de database expert gekozen voor een generieke oplossing waarbij er een FacilityReference en ImageReference tabel aanwezig zijn met daarin de refType en refNr. De refType geeft in dit geval aan naar welke tabel hij verwijst (acc of rom). Het refNr geeft het nummer in de tabel aan. Op deze manier kun je plaatjes en faciliteiten goed hergebruiken voor zowel accommodatie als kamer.

Verder had ik in mijn eerste opzetten de kamers volledig apart opgenomen, zonder generalisatie. Hiermee bedoel ik, dat de kamertabel (Room) alle informatie bevatte, zoals kamertype, of deze geboekt was etc. Op die manier krijg je natuurlijk enorm grote tabellen, en is er later voor

gekozen om meer standaarden in te brengen. Nu is de prijs, beschikbaarheid en soort kamer uit elkaar getrokken tot verschillende tabellen. Er kan zich dus in de tabel 'RoomType' een type bevinden zoals; 2 kamer appartement. Dit type kan vervolgens voor meerdere kamers worden gebruikt door een simpele verwijzing vanuit de tabel 'Room'. Deze oplossing bood een betere schaalbaarheid, betere performance, standaardisatie en geen vervuiling van de database.

Het moeilijkste onderdeel was een zo efficiënt mogelijke manier zoeken om de beschikbaarheid van een kamer te verwerken, zonder een enorme hoeveelheid aan onnodige data op te slaan per datum. Dit was voor mij eigenlijk een voortdurend proces van aanpassen, voorstellen, aanpassen. Uiteindelijk hebben we er dan voor gekozen om de tabel wel per datum te vullen, en vervolgens per datum aan te geven hoeveel van die kamers beschikbaar zijn en hoeveel er al geboekt zijn. Dit was nog de meest efficiënte oplossing. Wellicht komt hiervoor in de toekomst een betere oplossing, maar mijn kennis reikt niet zo ver en ook de database expert had er nog geen beter passende oplossing voor.

Het was erg moeilijk om alle acties voor je in het hoofd te halen, om zo te bepalen hoe iets in de database wordt opgeslagen of eruit gehaald moet worden. Hierbij moet je rekening houden met allerlei factoren, zoals "wat voor attributen heeft een kamer" en "hoe wordt een bepaald kamertype toegekend" en de meeste factoren waren voor mij tot dat moment nog onbekend. Deze factoren zijn in de reiswereld specifiek en zijn wellicht zelfs Peter Langhout Reizen specifiek waren. Ik heb inmiddels wel veel kennis opgedaan over het maken van een database en dat je met ongelooflijk veel dingen rekening moet houden.

5.4 Ontwikkeling van de overige pilots

5.4.1 De verschillende taakdiagrammen

Zoals eerder besproken is er na het taakdiagram in figuur 4, paragraaf 4.4, besloten om in het pilotontwikkelplan meer gedetailleerde taakdiagrammen te maken. Deze 3 diagrammen zijn bijgesloten in bijlage 3 en ik zal er een aantal stukjes uitlichten, waarom ik bepaalde keuzes heb gemaakt in vergelijking tot het vorige taakdiagram.

Bij de pilot reizigers heb ik een duidelijke scheiding gemaakt tussen het selecteren van een hotel en het opvragen van informatie. Het taakdiagram is opgesteld volgens de inmiddels bekende 'wizard' style, en beschrijft duidelijk de stappen die de gebruiker onderneemt voordat deze bij de eindstreep is. Omdat mij werd gezegd dat niet iedereen direct op zoek gaat naar een land, maar vaak naar een hotel, heb ik de navigatie enigszins aangepast. Men kan nu via 'informatie opvragen' uiteindelijk via land – gebied – stad bij een hotel terecht komen en vanaf daar de boekingsprocedure in gaan, of de gebruiker kiest voor een hotel en kan direct de boekingsprocedure in. Op deze manier ondervind ik de onnodige stappen die in mijn eerste taakdiagram wel bestonden, alleen stuur ik de gebruiker nog steeds.

Bij de hoteliers is het taakdiagram ook drastisch gewijzigd, omdat bleek dat bij het eerste taakdiagram er geen mogelijkheid bestond tot invoering van een nieuw hotel door een hotelier, hier had ik destijds nog niet over nagedacht. Hier heb ik de mogelijkheid voor gemaakt in mijn gedetailleerde taakdiagram. Een hotel kan altijd worden toegevoegd, maar wordt pas daadwerkelijk in de live database gezet na controle door het Management van PLR. Hierna kan de hotelier via gebruikersnaam/login zijn kamers toevoegen aan het hotel. Zoals te zien in de pilot database is er de mogelijkheid meerdere contactpersonen per hotel aan te maken, dus het betreft hier niet altijd 1 persoon. Hier heb ik voor gekozen, omdat ik me kan voorstellen dat een Hotelier niet zijn wachtwoord aan iedereen zou willen geven, en daarom meerdere contactpersonen zou willen aanmaken die ook kamers kunnen invullen.

Het taakdiagram voor het management heeft in feite geen wijzigingen ondergaan, maar er is slechts een gedetailleerder beeld ontstaan nadat ik alle stappen beter had uitgeschreven. Dit leek mij juist, omdat het pilotontwikkelplan zo gedetailleerd mogelijk moest zijn.

5.4.2 De styleguide en het prototype gebruikersinterface onder de loep

De styleguide is het deel in het pilotontwikkelplan wat alle uiterlijke factoren van een pilot beschrijft.

De styleguide ontwikkelen was nog een van de uitdagingen van het pilotontwikkelplan. Het design zou in samenwerking gebeuren met de afdeling WebDesign, maar ik moest eerst mijn eigen visie van de styleguide naar voren brengen. Dit vonden zij prettig om nieuwe inzichten te krijgen. Ik heb daarop een aantal algemene dingen van een website als Snowweb gedefinieerd in een CSS (Cascading Style Sheet). Ik had zoals eerder vermeld al ervaring in de webdesign wereld, dus ik had een al een idee hoe het eruit zou moeten komen te zien. Het zou uiteraard een winterachtig uiterlijk moeten krijgen, doch overzichtelijk en gebruikersvriendelijk.

Ik heb zelf een klein logo gemaakt bij Snowweb, wat naar mijn idee Snowweb goed kon weergeven. Ik heb hierbij gebruik gemaakt van blauwe kleuren en een abstract gebergte en een 'koud' lettertype. Nu is een 'ontwerp' altijd het mooist in de ogen van degene die het gemaakt heeft, maar toch bleef mijn mening dat dit een goed ontwerp was voor een logo.



Figuur 7

Vervolgens moest ik nadenken over titels van stukken tekst. Ik ben zelf altijd een voorstander van mooi geordende rechthoeken met informatie en daarboven een titel. Nu is het probleem van ontwerpen met lettertypen voor websites, dat de keuzemogelijkheid altijd vrij beperkt blijft. Het is namelijk zo dat de meeste pc's niet altijd alle lettertypen hebben die jij graag wil laten zien en deze pc die de website bekijkt vervangt hem dan door een standaard lettertype wat wellicht het hele design niet ten goede zou komen. Een plaatje kan ook gebruikt worden als titel, maar vanuit mijn oogpunt gaat dat teveel ten koste van de dynamiek en performance van de website. Een kleine aanpassing is dan niet meer zo makkelijk gedaan.

Hieronder mijn stukje CSS code waarin ik de titel en de inhoud van zo'n stuk rechthoek met informatie zou willen weergeven.

```
.TITLE1
{
    width: 50%;
    font-family: Broadway, Verdana;
    font-size: 12px;
    vertical-align: top;
    background-color: #EEEEEE;
    border: solid 1px black;
}

.CONTENT1
{
    width: 50%;
    font-family: Verdana;
    font-size: 11px;
    vertical-align: top;
}
```

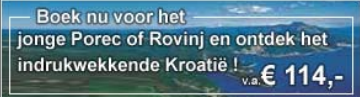

Zoals te zien wil ik de 'title' altijd weergeven met het lettertype Broadway en mocht deze niet beschikbaar zijn, dan Verdana. Ik vind dit lettertype goed passen bij Snowweb vanwege zijn vormen.

De knoppen heb ik als volgt gedefinieerd:

```
.globalMenuButtons
{
    font-family: Verdana;
    font-size: 11px;
    width: 125;
    background-color: #EEEEEE;
    color: Black;
    border-style: outset;
}
```

Ik heb gekozen om de knoppen te definiëren in een class, omdat ik de standaard knoppen die windows weergeeft absoluut niet mooi vind. Vandaar dat ik de knoppen enigszins heb nagmaakt met een ander lettertype en lettergrootte. Deze zijn ook gemakkelijk aan te passen voor de webdesigners en ook hier geldt weer dat snelheid een belangrijke factor is. Ik had ook plaatjes kunnen gebruiken, maar dat gaat wederom ten koste van de dynamiek en performance.

Om dit kracht bij te zetten met de inhoud en titel heb ik daarop ook een klein design gemaakt waarin het zichtbaar wordt.

Fictels		
Home	NieuwsBrief	Gebied Informatie
Vind Hotel	Aanbiedingen	Over Ons
Welkom bij Snowweb Hier kunt u alles vinden wat snowweb u te bieden heeft		Snelzoeken Gebied: <input type="text" value="Salzburgerland"/> Prijs: <input type="text" value=" < 99"/> Hotel: <input type="text" value="TestHotel"/> <input type="button" value="Zoek!"/>
Aanbiedingen 	Aanbiedingen	Aanbiedingen 

Figuur 8

In dit ontwerp is duidelijk te zien hoe de verschillende vakken zijn opgebouwd. Hierbij moeten natuurlijk nog veel meer kleuren worden gebruikt, maar dat laat ik over aan de Webdesign afdeling. Dit geeft ook slechts een indicatie van mijn idee van Snowweb, omdat mijn taak niet geheel lag bij de ontwikkeling van het design, maar ze wel graag mijn idee hierover wilden horen.

5.4.3 De XUAN modellen nader bekeken

Het XUAN model beschrijft alle acties en gevolgen van een actie, welke belangrijk zijn om een systeem te bouwen. Hiermee kan men direct zien wat het moet kunnen, en hoe deze acties gevolg hebben. Een belangrijk onderdeel van het pilotontwikkelplan, omdat het precies laat zien wat er gebeurd binnen de module. Ik zal een klein voorbeeld geven om een beeld te scheppen hoe zoiets eruit ziet:

Taak	User Action	Window Action	System Response	Database Action
Kies extra's	Select	Load pagina met extra's zoals skipassen etc		Select * From ReservationArticle

Toen ik begonnen was aan het eerste XUAN model voor de reizigers, ontstonden er gelijk een aantal problemen met betrekking tot de database. Ik moest nu een aantal query's gaan maken terwijl de database op dat moment nog onderhevig was aan verandering. Ik heb dit opgelost door het XUAN model alvast in te vullen zonder databaseacties. Deze acties heb ik later ingevuld. Dat heb ik zo gedaan voor alle pilots.

De verdere ontwikkeling van de XUAN modellen ging aardig goed. Ik denk dat er altijd een aantal acties zullen zijn die er wellicht later bijkomen wanneer men aan het ontwikkelen is. Misschien dat ze extra functies willen toevoegen. De basisfuncties van de pilots zoals ik deze heb gemaakt staan volledig beschreven in het XUAN model

5.4.4 De aanpassingen met betrekking tot de globaal organisatorische inrichting

De globaal organisatorische inrichting beschrijft wat de gebruikers doen, wat ze moeten kunnen, en hoe hun rollen en opleiding aangepast moeten worden om optimaal gebruik van de module te kunnen maken.

In feite was er voor de reizigers en hoteliers geen verandering in gebruikersrol, omdat de reizigers al meer op internet hun vakanties boeken en de hoteliers er een functie bij krijgen, namelijk de mogelijkheid om via internet hun kamers te verkopen dan slechts via bestaande verkoopkanalen en touroperators. Wat betreft het management van Peter Langhout Reizen gaat er wel wat veranderen, aangezien deze een wijziging van rol zullen krijgen. Er zullen een aantal mensen op de module gezet moeten worden om deze te monitoren zodat alles in goede banen wordt geleid. Ik ben hier uitgegaan van een drietal mensen, ieder uit een van de verschillende afdelingen van PLR. Een van autoreizen, een van vlieguren en een van busreizen. De rolwijziging zal hoogstwaarschijnlijk plaatsvinden bij de huidige inkopers van reizen, die hierna moeten controleren of de hoteliers hun gegevens wel correct invoeren en onderhoud uitvoeren aan de database items. Daar wil ik mee zeggen, dat er bijvoorbeeld nieuwe typen kamers zijn gekomen die moeten worden ingevuld in de database. Overigens zal de rol additief zijn, en geen vervangende rol.

Wat betreft de opleidingsbehoeften van de reizigers heb ik de statistieken er op nageslagen. Uit een aantal websites, die te vinden zijn in hoofdstuk 8 paragraaf 8.2, bleek dat de mensen op internet, en met name de doelgroep die online aankopen doet (waaronder ook online vakanties boeken valt) steeds stijgt en al meer dan de helft van de Nederlanders deze manier gebruikt. Ik mag er dan ook vanuit gaan dat de mensen die hun vakantie via Snowweb boeken (aangegeven bij doelgroepen in de definitiestudie, paragraaf 4.3) al gebruik hebben gemaakt van dergelijke internetdiensten en daarom geen extra opleiding nodig hebben. Het is bovendien ook niet mogelijk om deze mensen een opleiding te bieden. De interface zal een makkelijke intuïtieve interface moeten worden, wat het gebruik vergemakkelijkt.

Voor de hoteliers zal er een korte beschrijving worden gemaakt waarmee zij aan de slag kunnen, omdat zij een ingewikkeldere taak hebben die niet volgens een bepaald pad hoeft te lopen. Met deze korte beschrijving moeten zij voldoende uit de voeten kunnen om alles correct in te vullen.

Het management zal een korte workshop krijgen zodat hun opleidingsniveau wordt verhoogd om zodoende gebruik te kunnen maken van de module. Zij zijn namelijk de laatste controlerende factor en mogen geen fouten maken. Mochten er fouten ontstaan, dan moeten ze in staat zijn deze op te kunnen lossen. Overigens bedoel ik hier natuurlijk geen technische aspecten mee, maar wel of de hoteliers bijvoorbeeld wel de juiste kamer hebben gekozen en daarbij de juiste faciliteiten hebben aangegeven.

Op deze manier moeten alledrie de gebruikersgroepen in staat zijn om goed gebruik te kunnen maken van de module.

6 Testplan en testrapport

6.1 Inleiding

Het testplan is door mij ontwikkeld om Snowweb zo goed mogelijk te kunnen testen en hierbij een rapport te kunnen opleveren wat uitslag geeft of Snowweb goed is of niet. Een dergelijk uitgebreid testplan is in mijn opinie altijd een goede manier om een stuk programmatuur te testen of deze voldoet aan de usability eisen. In dit geval moet de gebruikersvriendelijkheid dus hoog liggen. De test zal worden uitgevoerd door minimaal 10 personen, dit is een gegeven vastgesteld door SUMI en hiermee zal er een waarheidsgetrouwe conclusie ontstaan.

Er zal door mij geen rapport worden opgeleverd, omdat de test zal worden uitgevoerd nádat Snowweb ontwikkeld is tot een werkbaar geheel. Dit zal echter pas een aantal maanden ná mijn stage gebeuren.

6.2 Waarom SUMI?

Ik heb in het testplan als hoofdtest de SUMI (Software Usability Measurement Inventory) test opgenomen. Deze test is de enige beschikbare vragenlijst voor het beoordelen van usability van software die is ontwikkeld, gevalideerd en gestandaardiseerd op Europees niveau. SUMI vragenlijsten zijn beschikbaar in het Engels, Frans, Duits en Nederlands.

Het is dus een internationaal erkende test, waarbij ik mag aannemen dat deze de usability ten volste test.

SUMI is bedoeld om van een aantal aspecten een score te kunnen berekenen. Het aantal aspecten waar je op test is als volgt: Learnability, Helpfulness, Affect, Flexibility en Robustness.

Omdat ik alleen een SUMI nog niet voldoende vind voor de volledige test van een module, heb ik vanuit mijn ervaring op school met testen besloten om de 2 aspecten die bij de SUMI test het minste scoren te onderwerpen aan een tweetal extra testen, om zo een gedetailleerder beeld te krijgen van datgene wat er mis is.

Omdat ik helaas zelf de test niet kan doen heb ik op basis van het gehele traject van tevoren al gekozen voor een tweetal aspecten die in mijn ogen grondig getest moeten worden. Flexibility en Learnability. Ik heb voor flexibility gekozen. De module moet namelijk vooral flexibel zijn en dit is ook een belangrijk aspect voor het slagen van de module. Ik heb gekozen voor learnability, omdat learnability staat voor het gemak waarmee nieuwe gebruikers effectieve interactie kunnen beginnen en maximale performance kunnen bereiken. Dit is uiteraard een belangrijk onderdeel, des te sneller en makkelijker mensen met de module kunnen opschieten, des te eerder men gaat boeken.

Voor meer informatie over de usability aspecten verwijs ik naar Bijlage 4.

6.3 Hoe zijn detail aspecten getest?

Om de learnability te kunnen testen heb ik gebruik gemaakt van de Think Aloud testmethode. Hierbij moeten de gebruikers een bepaald scenario doorlopen en denken ze hardop. Er zit hier vervolgens iemand bij die dit alles noteert en aan de hand van bepaalde op- of aanmerkingen kan achterhaald worden hoe het met de learnability gesteld is. Verder worden alle handelingen en acties van de gebruiker genoteerd door een of twee observatoren. Hiervoor zijn een aantal criteria te vinden:

- learning time (subjectief).
- Aantal keren dat help of documentatie is gebruikt.
- Verhouding goede keuzes / foute keuzes.
- Aantal foute herhalingen.
- Aantal keren dat de gebruiker door de module op het verkeerde spoor wordt gezet.

Belangrijk voor de learnability is het criterium: Aantal keren dat de help of documentatie is gebruikt.

Ik heb voor deze test gekozen, omdat mensen ongewild soms dingen zeggen die van groot belang zijn voor de module en hieruit kan lering worden getrokken. De andere criteria die genoteerd worden zijn ook van belang, aangezien hieruit blijkt hoe intuïtief de module is. Als de documentatie vaak wordt gebruikt, dan blijkt dat de intuïtiviteit nog veel te wensen over laat. Zo is dit ook het geval bij de andere criteria. Ik had hiervoor ook een andere test kunnen doen zoals de Cognitieve Walkthrough, waarbij de gebruiker door een aantal vaste scenario's loopt. Echter deze test geeft niet zo gedetailleerd weer wat de gebruiker nu echt daadwerkelijk vervelend, onjuist of onduidelijk vindt.

Voor de flexibility heb ik daarentegen wél gekozen voor de Cognitieve Walkthrough. Hierbij worden een aantal scenario's doorlopen om te kijken of de module de gebruikers ondersteund bij hun taken en om te kijken of de module flexibel genoeg is om bepaalde gebruikerskeuzen te ondervangen. De te vinden criteria hierbij zijn:

- Aantal keren dat help of documentatie is gebruikt.
- Verhouding goede keuzes / foute keuzes.
- Aantal keren dat de gebruiker zich afvraagt wat hij/zij moet doen (na vreemde programma output).
- Aantal keren dat de gebruiker niet weet wat hij/zij moet doen.

Belangrijk voor de flexibiliteit is het criterium: Aantal keren dat de gebruiker zich afvraagt wat hij zij moet doen (na vreemde programma output)

Ik heb in dit geval voor deze test gekozen, omdat aan de hand van de scenario's die staan gedefinieerd in het pilotontwikkelplan een generiek pad voor de gebruiker kan worden neergelegd. Een actie die vaak wordt gedaan kan zo perfect worden onderzocht, om te kijken of er geen rare dingen ontstaan terwijl de gebruiker bezig is. Ook zal de gebruiker in dit geval bepaalde waarden willen invullen, en de module moet dan zo flexibel mogelijk zijn om deze gegevens juist te interpreteren. Iedere gebruiker zal dit anders willen doen. De scenario's zijn ook vrij algemeen gehouden zodat de gebruiker in feite zelf kan beslissen (voor zover dit mogelijk is met de wizard style die is geïmplementeerd in de module).

De keuze voor twee verschillende testen lijkt misschien raar, omdat de Think Aloud test voor beide onderdelen geschikt zou kunnen zijn. De scenario's zijn echter verschillend, bij de learnability zijn de scenario's gemakkelijker, omdat het hier slechts interface betreft en niet

direct functionaliteit. Bij de ene test gaat het om het gemak, en bij de andere test om de flexibiliteit. Ook houden twee verschillende tests de gebruiker eerder bij de les, dan twee dezelfde tests waarbij onderscheid moeilijker word.

6.4 Hoe moeten de gegevens verwerkt worden?

Voor de SUMI test bestaan standaard verwerkingsprogramma's zoals SPSS. Je voert de resultaten van je SUMI in en er komt een grafiek uit met daarin een voorstelling van de gegevens en de resultaten.

Voor de twee specifieke testmethoden zal het aan moeten komen op een objectieve conclusie door de verschillende observatoren. Er zullen opmerkingen door gebruikers worden vergeleken en deze gegevens zullen allen tezamen in een document komen met daaraan een conclusie verbonden. Aan de hand van deze gegevens kan bepaald worden of het product uiteindelijk goed is of niet goed.

7 De fictieve invoering beschreven

De invoering kan ik helaas niet zelf doen en er zal hier ook geen rapport van verschijnen, want deze wordt volledig door het bedrijf Tickotel uitgevoerd. Echter ik wil hier wel een licht op werpen, aangezien ik wel een idee heb hoe een dergelijke invoering wellicht het beste zou kunnen geschieden.

Aangezien de module Snowweb geïntegreerd moet worden binnen Tourforce lijkt het mij verstandig om eerst een test Tourforce omgeving op te zetten en hierbinnen Snowweb te integreren.

Hiermee kunnen de nodige technische tests worden gedaan, bijvoorbeeld een stress test en andere performance testen. Hierna kan het testplan wat ik heb gemaakt worden uitgevoerd door een aantal gebruikers om het systeem te testen op gebruikersniveau. Zodra alle tests voorspoedig zijn verlopen kan Snowweb geïntegreerd worden in de 'live' Tourforce omgeving. Hierna kan er gebruik worden gemaakt van de module.

8 Evaluaties

8.1 Productevaluatie

8.1.1 De documentatie

Zoals is gebleken uit het gehele verslag, is de documentatie bij mijn afstudeerstage een zeer belangrijk onderdeel geweest en bevat alles om de module mee te ontwikkelen, behalve de code en slechts een gedeelte van het uiterlijk. Ik denk dat de documentatie op hoog niveau is geschreven en dankzij een nauwe samenwerking met de ICT Manager en de database-expert plus de tijd die erin is gestoken, het geheel een compleet en goed stuk documentatie is geworden.

Beide medewerkers waren ook tevreden over de documentatie dat allemaal is samengevoegd in één groot ontwikkeldocument en wat gebruikt gaat worden voor de ontwikkeling van Snowweb.

Het was uiteindelijk een hele klus om alle situaties goed op orde te krijgen en van tevoren al rekening te houden met allerlei situaties die zich kunnen voordoen bij een module zoals Snowweb. De moeilijkheid lag voor mij vooral in het 'niet' programmeren. Meestal kom ik tijdens de programmeerfase ook nog dingen tegen die ik dan ter plekke verbeter of aanpas, maar dit maal moest het allemaal vanuit de documentatie naar voren komen en moest ik er in deze zelfde documentatie direct een oplossing voor bedenken. Er dient ook rekening gehouden te worden met de mogelijkheden van de programmeertaal en de database engine.

8.1.2 De database

De database is toch eigenlijk wel het hoogtepunt geworden van Snowweb. Ik heb hier enorm veel tijd aan besteed. Omdat de database het absolute fundament is voor een goede module vond ik ook dat ik dit volledig en supergedetailleerd moest uitwerken. Het is na vele pogingen een database geworden die efficiënt en flexibel is.

De database bevat voor bijna ieder mogelijke denkbare situatie in Snowweb een oplossing. Zo zijn bijvoorbeeld alle plaatjes, faciliteiten en kamertypes generiek in te vullen, wat de querytijd verkort en de databaseomvang beperkt houdt. Ook is hij flexibel door verschillende tussentabellen te maken, wat een koppeling tussen bijvoorbeeld skipas en unieke gebruiker per reservering mogelijk maakt.

Een goed product, volledig uitgeschreven, en uiteindelijk in een dusdanige scripttaal omgezet, zodat deze meteen geïmplementeerd kon worden in de Microsoft SQL Server.

8.1.3 Het testplan

Het testplan is gedetailleerd en bevat dieptetesten zoals de Cognitieve Walkthrough om bepaalde onderdelen nog beter te kunnen testen. Ik heb het testplan volledig en secuur gemaakt. Er is echter één minpuntje. Ik kan de test natuurlijk niet zelf uitvoeren. Hierdoor heb ik van tevoren al een tweetal items gekozen om extra testen op te doen. Ik heb goed nagedacht over deze twee items, en heb besloten dat aan de hand van het project, deze waarschijnlijk de meeste aandacht zullen verdienen. In het ergste geval zullen er twee andere items uit de SUMI test komen die meer aandacht verdienen, maar ook daarvoor kunnen beide testen die ik heb besproken gebruikt worden.

Verder ben ik, samen met de stagebegeleider van mening dat dit testplan voldoende informatie bevat om de test op usability goed uit te voeren.

8.2 Procesevaluatie

8.2.1 Projectaanpak

De aanpak / methode die ik gekozen heb is de IAD ontwikkelmethode. Naast het feit dat ik hier tijdens de opleiding veelvuldig mee in aanraking gekomen ben en dus ook redelijk wat ervaring opgedaan heb met deze ontwikkelmethode, leent deze methode zich prima voor webapplicatie ontwikkeling. Ik ben dan ook groot voorstander van een pilotgeoriënteerd ontwikkeltraject. IAD zorgde voor een duidelijk stramien met bijbehorende planning wat als houvast functioneerde tijdens het ontwikkeltraject.

Van de IAD ontwikkelmethode zijn niet alle projectonderdelen uitgevoerd aangezien die niet nodig waren of er geen mogelijkheid voor was, zoals de iteratie.

8.2.2 ASP.NET in het kort

In het voorgaande verslag wordt er niet veel aandacht besteed aan de programmeertaal ASP.NET, maar hier moest wel degelijk rekening mee worden gehouden. Ik heb bij aanvang van de afstudeerstage mij voorgenomen een nieuwe taal te leren. Ik was op dat moment al vrij ver met PHP en ASP. Deze twee talen konden mijn nieuwsgierigheid naar nieuwe mogelijkheden niet langer bekoren.

Toen ik bij de aanvang van deze stage kreeg te horen dat er gebruik gemaakt ging worden van ASP.NET leek mij het zeer interessant om van deze taal alvast de basis te leren. Niet alleen omdat het veel gevraagd zal worden later in mijn beroep, maar ook omdat ik van ontzettend veel nieuwe mogelijkheden hoorde wat met ASP en PHP niet zo maar mogelijk was. Ook bood deze taal me uiteraard het inzicht in het project, zodat ik wist hoe ik mijn documentatie zou kunnen richten op de mogelijkheden van deze taal.

Ik heb ongeveer 1,5 à 2 maanden van mijn stage kunnen toewijden aan het leren van ASP.NET. Dit kon onder meer omdat ik eerder was begonnen met de stage dan normaal gesproken de bedoeling was, en deze kans heb ik direct aangegrepen. Ik ben begonnen met het boek VB.NET in 21 dagen. VB.NET is eigenlijk een taal binnen het geheel van ASP.NET, aangezien ASP.NET (ook wel bekend als C#) zelf eigenlijk meer op Java lijkt. Deze variant leek voor mij nog het meest leek op het oude ASP. Ik heb dit direct aangegrepen om zo een snellere start te kunnen maken, vanwege de herkenning.

Die eerste paar maanden ben ik bezig geweest met het maken van kleine webapplicaties, en heb ook van mijn ICT manager geleerd hoe ik met VB.NET gebruik kon maken van webservices. Deze kleine webapplicaties hadden weinig te doen met het project wat ik ging doen, maar ik heb er wel veel van geleerd, en dat was ook mijn bedoeling.

Ik heb inmiddels de basis van VB.NET onder de knie en kan al een bescheiden website opzetten die alle functionaliteit bezit om items uit een database te halen of deze erin terug te plaatsen.

Ik ben van plan om in mijn verdere loopbaan meer kennis op te doen over ASP.NET en heb hiermee alvast een basis gezet.

8.2.3 De database en zijn lange ontwikkeltijd

De database heeft een erg lang ontwikkeltraject gehad, mede dankzij het grote detail dat ik erin wilde hebben, en het feit dat ik nog nooit een dergelijke database binnen een reisorganisatie had ontwikkeld. Ook de database-expert had nog nooit zoiets gemaakt voor de reiswereld, ook al had zij duidelijk wel ontzettend veel ervaring met databases binnen de reiswereld, maar ook daarbuiten.

Ik was begonnen aan de database en al snel bleek dat er constant aanpassingen nodig waren. Dit begon al bij het OMT model. Hierin bevonden zich items die weg moesten of veranderd moesten worden. Als voorbeeld: Ik had de gebieden en landen erin staan, deze mochten later weer weg.

Omdat ik zo onervaren was in het maken van een database binnen de reiswereld heb ik constant de veranderingen doorgesproken met de database-expert. Zo was er het klassieke voorbeeld dat ik niet nagedacht had over een prijs per persoon of een prijs per nacht. Dit werd later opgesplitst in 2 onderdelen als je kijkt naar het OMT model. Na vele aanpassingen zijn de database-expert en ik uiteindelijk overeengekomen over het OMT model dat de werkelijkheid benaderd.

Hierna ben ik begonnen met het uitschrijven van de normaalvormen en daarna het maken van de database in Microsoft SQL Server. Ook hier heb ik steeds weer een model gemaakt en deze in overleg met de database-expert besproken. Ik heb hier zoals eerder uitgelegd van een trial-error methode. Steeds als ik een aanpassing had doorgevoerd bleek er ook wel weer iets anders niet goed te zijn. Heel veel modellen die ik heb gemaakt waren direct eigenlijk al niet goed, omdat ik in mijn hoofd bepaalde acties na ging lopen en vervolgens op een fout stuitte in de database. Zo had ik bijvoorbeeld in de eerste paar modellen nog geen mogelijkheid om meerdere gebruikers allemaal een unieke skipas of ander extra te geven, maar dit zag ik pas toen ik de actie in mijn hoofd maakte. Dit moest dus weer aangepast worden, en zo ben ik uiteindelijk tot een database gekomen die alle mogelijke acties kan ondervangen. Dit was dus mede dankzij mijn eigen input, aangezien mijn blik op sommige situaties een verfrissende was voor de database-expert.

Zo vond ik het logisch om de faciliteiten in het algemeen op te nemen, omdat je deze dingen steeds hergebruikt binnen de module. Hier had de database-expert nog niet over nagedacht, omdat de faciliteiten op dit moment slechts als stukje tekst bij een accommodatie worden geschreven. Het leek mij sneller en efficiënter om hier vaste items van te maken met een icoontje erbij, zodat je vooral een saamhorigheid in het design zou krijgen i.p.v. stukjes tekst die steeds anders zijn. Ook hoeven de hoteliers die iets invoeren op deze manier alleen maar wat aan te klikken, en niet zelf te schrijven. Dat geeft meer controle aan de kant van Peter Langhout Reizen.

De uiteindelijke database is tot een goede database geworden dankzij mijn inspanning en de hulp van de database expert. Ik heb hierbij bepaalde keuzes (zoals hierboven beschreven) moeten maken en na moeten denken over specifieke onderdelen, zodat ik nu meer begrip heb van een werkelijke database i.p.v. de theorie op school.

Het proces van de trial-error ontwikkeling is een zeer langdradige methode, maar geeft je wel het meeste inzicht in de situaties, omdat je voor alle fouten gaandeweg een goede oplossing bedenkt. Als je alles in één keer goed doet, en alleen een trial-error doet bij een OMT model, dan zal later blijken dat de database toch niet alles goed ondervangt en dan ben je al live en is je module al in werking. Het kost dan geld en tijd dat menig bedrijf er dan niet meer voor over heeft.

Ik moest nu ook af en toe grote delen in mijn eentje op lossen, omdat de database-expert niet altijd tijd had. Dat kost veel tijd. Om te voorkomen dat het weer zoveel tijd gaat kosten, lijkt het mij goed om zulke dingen in een projectteam te doen met mensen die nog meer verstand van de

databases hebben. Ook zullen de problemen waarschijnlijk eerder boven komen, als je dit met meerdere mensen bespreekt.

8.2.4 Wijziging stagebegeleider

De eerste 6 maanden van mijn stage bij Peter Langhout Reizen was dhr. Ing. Vincent van Hulst mijn stagebegeleider. Deze heeft echter eind augustus ontslag genomen en is nu ergens anders werkzaam. Peter Langhout heeft nu besloten dat Otje Bollen, de database-expert met wie ik uitgebreid heb samengewerkt, mijn stagebegeleider wordt voor de laatste maand dat ik hier zit. Men vindt terecht dat de naam Peter Langhout Reizen verbonden moet zijn aan mijn stage, en daarom kan een oud-werknemer PLR niet vertegenwoordigen tijdens de examenzitting.

Deze wijziging heeft voor mij weinig gevolgen, omdat Otje al goed op de hoogte was van alles wat ik deed en heb gedaan en ook de diploma's in huis heeft om mij te kunnen begeleiden. Ik had er persoonlijk geen problemen mee, daar ik ook goed met Otje kan opschieten.

8.2.5 Veranderingen in het traject met betrekking tot Tickotel

In de laatste maand van het traject, toen ik net klaar was met de eerste versie van mijn concept verslag is het onmogelijke gebeurd. De groep Tickotel is het na 8 maanden van overleg en voorbereiding niet eens geworden met Peter Langhout over de 'sourcecode'. Tickotel wil het product ook extern als product gaan marketen, terwijl Peter Langhout de beschikking wil hebben over de 'sourcecode', zodat hier intern eventuele wijzigingen kunnen worden doorgevoerd wanneer dat nodig blijkt te zijn.

Als ze het hier niet over eens kunnen worden, zal het project worden afgeblazen en is het des te belangrijker dat mijn documentatie goed in elkaar steekt, zodat een eventuele andere groep hiermee verder kan. Deze wijziging heeft voor mij gelukkig verder geen grote gevolgen, maar ik vond het belangrijk genoeg om het even te vermelden.

8.2.6 Het bedrijf is veranderd

In mijn tijd dat ik bij PLR mijn stage heb gedaan is het bedrijf onderhevig aan verandering geweest, want de markt veranderd voortdurend en je móet mee. Ik heb veel mensen zien gaan, en de volledige structuur binnen het bedrijf was aan het veranderen.

Dit heeft mij vooral geleerd dat niets vaststaat. Ook mijn project heeft veranderingen ondergaan, en is een aantal keer opnieuw gedefinieerd, zo was er eerst een Third Party, dit was echter niet meer nodig. Een aantal tabellen in de database hoefde ik niet meer te doen en het project bleek groter dan verwacht, waardoor ik alleen documentatie heb kunnen doen.

De veranderingen hebben mij meer inzicht gegeven in het bedrijfsleven en hoe onstabiel het kan zijn. Een project zal vrijwel nooit vaststaan en ontwikkeld worden, zoals het aan het begin is opgesteld. Wees altijd bereid voor eventuele drastische veranderingen en probeer jezelf zo goed mogelijk daaraan aan te passen.

9 Literatuurlijst

9.1 Boeken

Graphical User Interface Design and Evaluation (GUIDE)

Auteur(s): David Redmond – Pyle & Alan Moore

Eerste druk, 1995

IAD, Het evolutionair ontwikkelen van informatiesystemen

Auteur(s): R.J.H. Tolido

Tweede oplage, augustus 1997

Object-Oriented Modeling and Design

Auteur(s): James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorensen

Eerste druk, 1991

ASP.NET in 21 dagen

Auteur(s): C. Payne

2002

VB.NET in 21 dagen

Auteur(s): D. MacKenzie, K. Sharkey

2002

9.2 Websites

www.nachtjeweg.nl

© Bookit B.V. 2003

Voorbeeldsite voor ontwikkeling Snowweb

www.weekendjeweg.nl

© Bookit B.V. 2004

Voorbeeldsite voor ontwikkeling Snowweb

www.snowweb.com

© Sierra Web Design

Voorbeeldsite voor ontwikkeling Snowweb

msdn.microsoft.com

© Microsoft 2004

Technische website met veel kennisartikelen over programmeren in ASP.NET

www.m4n.nl/news/news_articles_5.jsp

© M4N 2004

Informatieve website met statistieken over internetgebruik

www.improveqs.nl/sumi.htm

Improve Quality Services B.V. © 2004

Informatieve site over de SUMI test

10 Verklarende woordenlijst

WizardStyle

Dit is een bepaalde manier van interfacing, waarbij je de gebruiker exact in een richting stuurt waar jij als ontwikkelaar de gebruiker wil hebben. Bij een wizardstyle zijn de keuzemogelijkheden ook altijd beperkt en betreft het vaak een 'vorige' en 'volgende' knop, om de gebruiker zo tussen de keuzemogelijkheden door te loodsen. Op deze manier kun je bij de volgende stap alweer rekening houden met de wensen van de gebruiker en deze op een bepaalde manier presenteren i.p.v. dat je alles op één enkele pagina neerzet.

Intertravel

De huidige kantoorapplicatie van Peter Langhout Reizen. Dit systeem verzorgt alleen in de verkoop van pakketreizen en heeft geen verdere uitbreidingsmogelijkheden. Het systeem is zes jaar geleden ontwikkeld in nextstep/openstep en heeft inmiddels de nodige ouderdomskuren.

Tourforce

Het nieuwe reisboekingsysteem van het bedrijf Tickotel. Dit nieuwe systeem moet meer flexibiliteit bieden en mogelijkheid om af te stappen van de pakketreizen.

Openduaal

Dit is een opleidingsvariant op de Haagse Hogeschool. Het is hierbij de bedoeling dat je drie dagen in de week werkt en twee dagen in de week naar school gaat. Dit gebeurt bij één bedrijf, en het is in principe de bedoeling dat je hier ook je proefafstuderen en uiteindelijke afstuderen doet.

Live

Het woord 'live' geeft binnen de ICT context altijd iets aan wat aan de benaderbaar is door de gebruikers en als hoedanig gebruikt kan worden waarvoor de desbetreffende module of applicatie is bedoeld. De items die 'live' staan zijn ook uit de testfase en gereed voor gebruik. Een beta, of test item kan nooit 'live' staan.

Database Engine

De database engine is in feite de applicatie waarmee databases gebouwd en gebruikt kunnen worden. De database zelf bevat de informatie, en de database engine zorgt ervoor dat de database het doet. Een aantal voorbeelden van database engines zijn: MySQL, MS SQL Server en Oracle.

Query

De query is als het ware de 'vraag' die je stelt aan een database, waarmee je bepaalde informatie uit de database wil halen of er in terug stoppen. Het word in de taal SQL geschreven en geeft je alle besturingsmogelijkheden die je wil (afhankelijk van de database engine).

ASP.NET

ASP.NET is de volgende generatie taal van Microsoft's Active Server Pages(ASP). ASP.NET verschilt op twee belangrijke punten van ASP: Het ondersteunt code die geschreven is in gecompileerde talen zoals: Visual Basic (VB.), C++ en Perl en het bevat de mogelijkheid om code te scheiden van inhoud.

11 Bijlagen

Hier bevinden zich alle bijlagen die genoemd zijn in het verslag. Hieronder nog een korte opsomming van deze bijlagen en hun betekenis.

Bijlage 1; Organogram van Peter Langhout Reizen

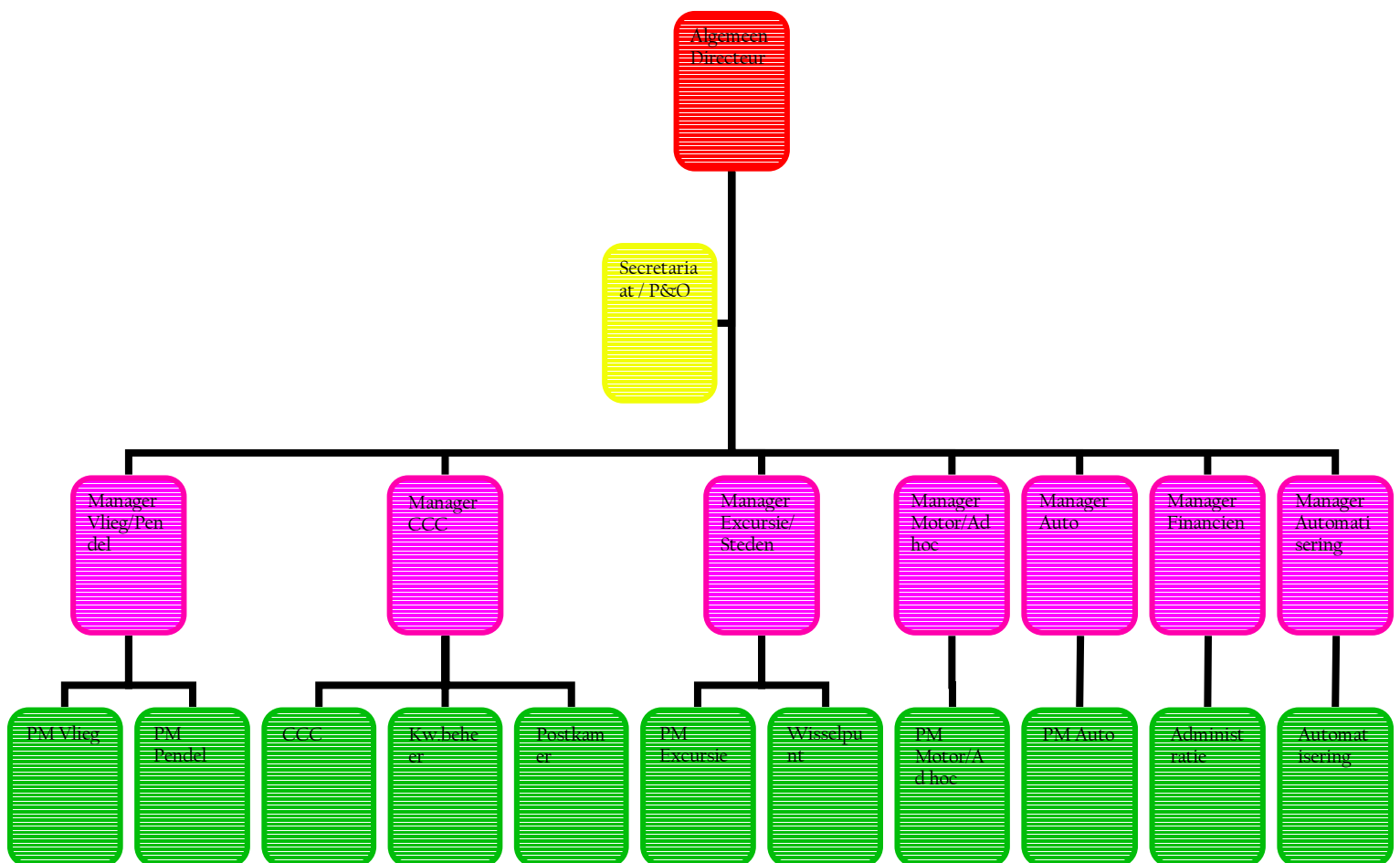
Bijlage 2; Databaseregels die gelden bij de ontwikkeling van een database voor Peter Langhout Reizen.

Bijlage 3; Gedetailleerde taakdiagrammen van de pilots Reizigers, Hoteliers en Management

Bijlage 4; Usability aspecten

Bijlage 1

Het organogram van Peter Langhout Reizen.



Bijlage 2

Project	:	Database-regels
Onderdeel	:	
Datum	:	7-4-2004
Ontwikkelaar	:	Otje

Algemeen

- Voor elk script / StoredProcedure / Function komt een header te staan met minimaal de volgende informatie (voor template kun je bij Otje/René terecht:
 - Naam script/storedProcedure/function
 - Auteur
 - Aanmaakdatum
 - Doel van het script/storedProcedure/Function
 - Voor het bijhouden van de aanpassingen aan de scripts: Persoon, datum aanpassing.
- Bij het maken van een StoredProcedure/Function overleggen met René/Otje over de inhoud.

Nieuw Project

Elke project heeft zijn eigen 2 of 3 letter afkorting. Voor elk object wat in de database wordt gemaakt, moet deze afkorting in kleine letters verschijnen. Dus voor Stored Procedures, Tabellen, Views en Functies.

Momenteel zijn de de volgende projecten:

fsp	FlightSeatPlanning
aru	Applicatie Reis Uitvoering
ir	InfoRay
rs	Reporting Services

Nieuwe Tabellen

Bij het maken van nieuwe tabellen altijd overleggen met René (bij afwezigheid met Otje) in welke database de nieuwe tabellen moeten komen.

Elke nieuwe tabel heeft een 2 of 3 lettercode in kleine letters vooraf, om aan te geven bij welk project een tabel hoort.

Aan de volgende regels moeten nieuwe tabellen voldoen.

- Alles in het engels (tabelnamen, veldnamen, storedprocedures etc.)
- Elke tabel heeft een unieke drie-letter code voorafgaand aan ieder veld.
 - Bijv.. tabel Trip = trp, tabel Reservation = res
 - Overleg voor het vinden van de unieke code met Otje!
- Primary Key van de tabel heeft eerst
 - de drie-letter code
 - vervolgens 'Nr'.
 - Bijv. tabel Trip = trpNr, tabel Reservation = resNr
- Foreign Key van de tabel heeft eerst
 - de drie-letter code
 - vervolgens drie-letter code van de tabel waarnaar verwezen wordt
 - vervolgens 'Nr'

- Bijv. Tabel Reservation naar tabel Trip: resTrpNr
- De drie-letter code staat in Kleine letters, vervolgens start elk woord met een hoofdletter
 - Bijv. Trip veld datum geldig vanaf: trpDateFrom
- Booleans (type bit) worden aan het eind aangegeven door 'YN'
 - Bijv. Reservation veld is een ticket geprint? = resPrintTicketYN (1 = ja, 0 = nee)
- Bij ieder veld een description erbij plaatsen. Zie template voor het maken van de tabellen. Zodoende kan een ieder eenvoudig het doen van een veld later terugvinden.
- Gebruik van enkele velden die vaak voorkomen (xxx staat voor de drie-letter code):
 - xxxRefType drie-letter code naar tabel waarnaar verwezen wordt
 - xxxRefNr Primary Key van deze verwijzing
 - Bv. xxxRefType = 'trp' en xxxRefNr = 3489 => 3489 is het trpNr
 - xxxCode varchar(10), afkorting code
 - xxxName varchar(50, Bijhorende naam
 - xxxDateFrom geldig vanaf
 - xxxDateTo geldig t/m
 - xxxQuantity aantal
 - xxxActualYN is het record nog actual. Te gebruiken als je records niet wilt verwijderen, maar wel wil aangeven dat ie niet meer 'valid' is.
- Probeer logische namen te gebruiken die wat zeggen.
 - bv. xxxDate zegt niets, maar xxxDateBooked geeft aan boekingsdatum.
- Elke tabel heeft de volgende extra velden:
 - xxxUser nvarchar(25) NOT NULL gebruiker die record laatst heeft gewijzigd
 - xxxProgram nvarchar(40) NOT NULL programma die record laatst heeft gewijzigd
 - xxxLastDate Datetime NOT NULL Datum waarop record laatst is gewijzigd.
- De tabellen moeten worden gemaakt via een script. Dit is veel intikwerk, maar mocht de database verloren gaan, dan hebben we de scripts waarmee de tabellen opnieuw gegenereerd kunnen worden.
Een template van dit script kun je krijgen bij René/Otje
- Het maken van de indexen gebeurt ook via een script. Ook dit template kun je krijgen bij René/Otje. Overleg met René waar de indexen komen te staan.

Nieuwe View

Naamgeving: een view heeft als voorloop letter de hoofdletter V.

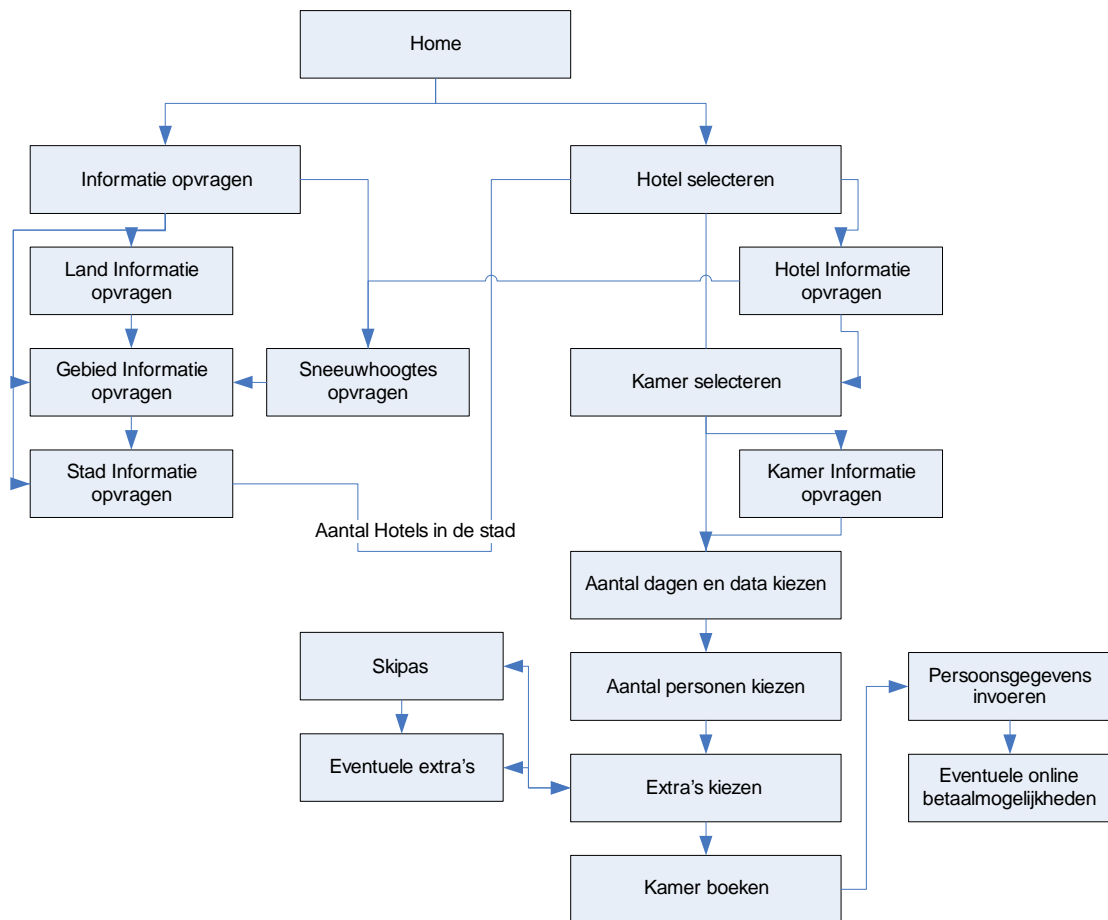
Dus de naam is bv. fspV<ViewNaam>. Als eerste de 2/3 voorloopletters van het project en vervolgens een V en vervolgens de naam vd View.

Nieuwe Functie

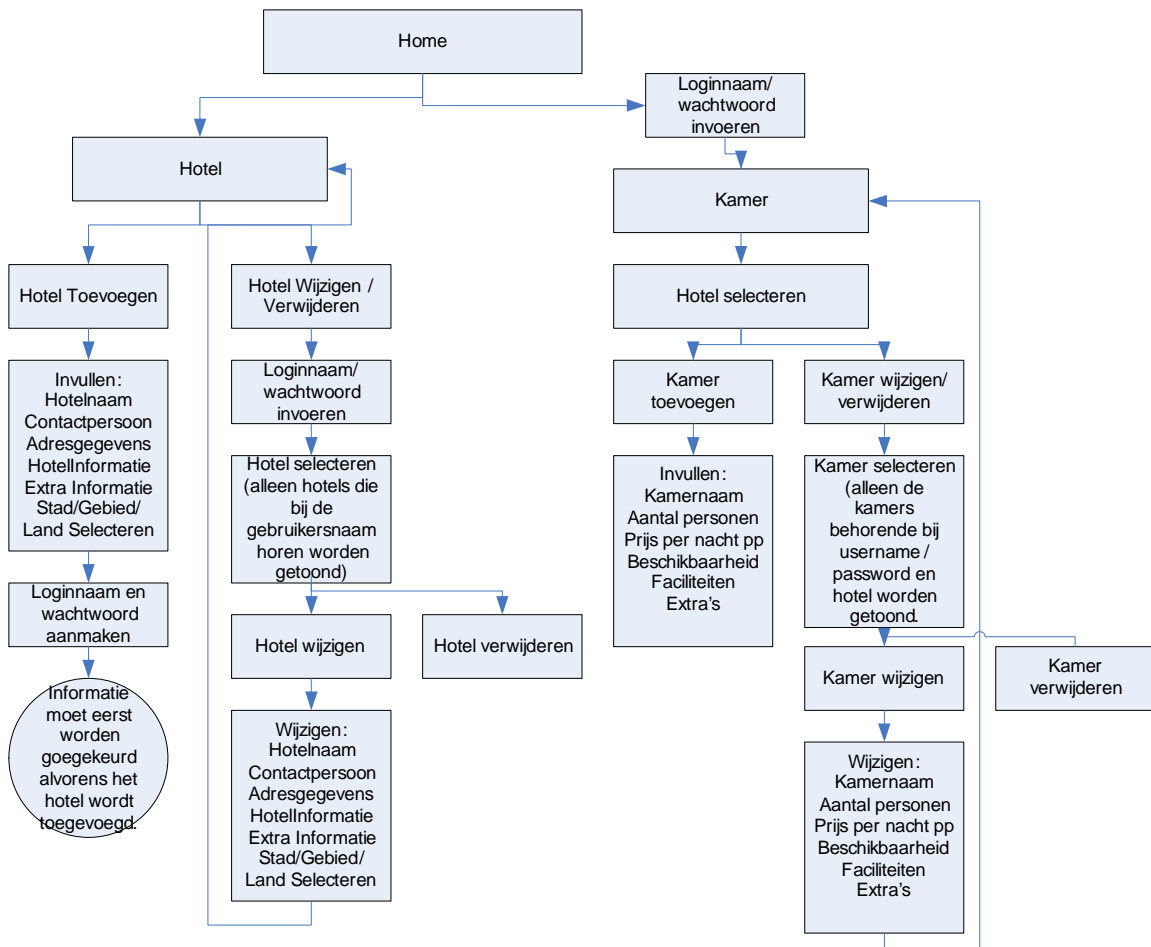
Naamgeving: Een functie krijgt als voorloop letter de hoofdletter F.

Dus de naam is bv. fspF<Functienaam>. Als eerste de 2/3 voorloopletters van het project en vervolgens een F en vervolgens de naam vd Functie.

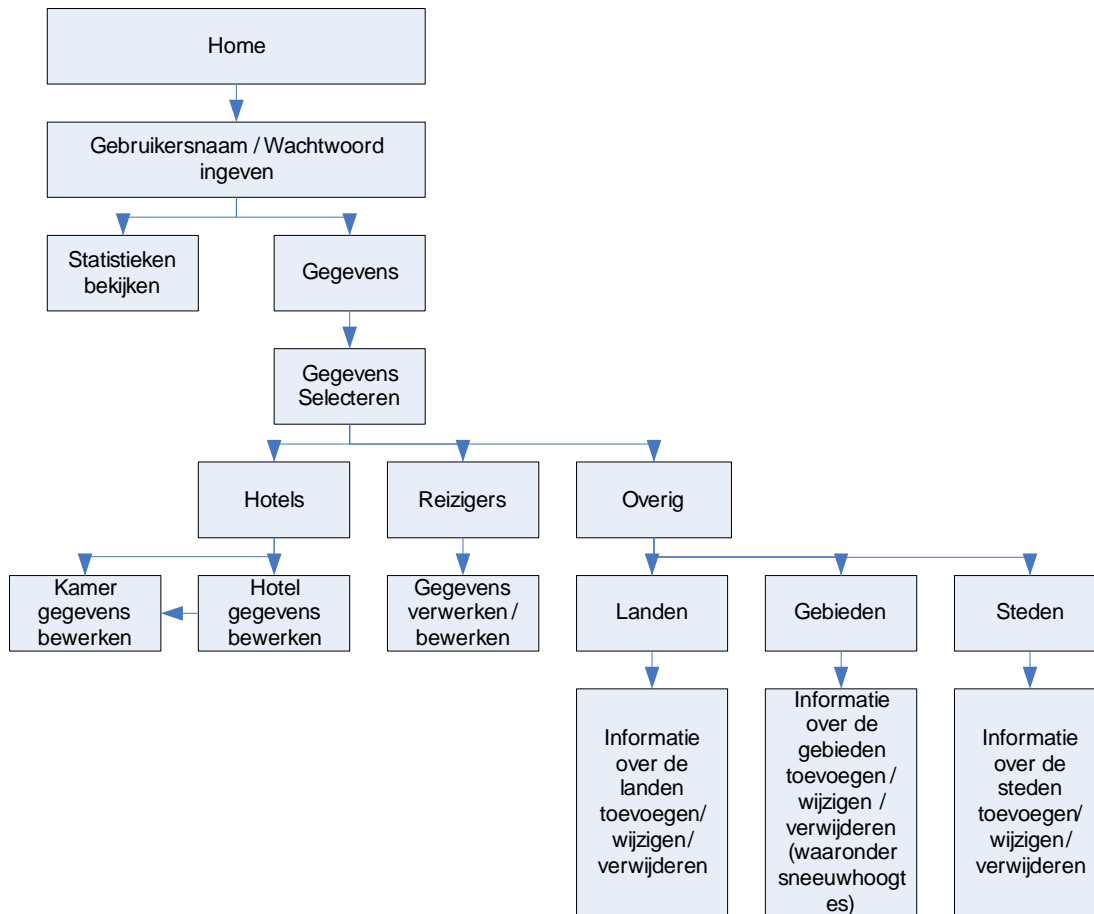
Bijlage 3



Figuur 9 : Pilot Reizigers



Figuur 10 : Pilot Hoteliers



Figuur 9 : Pilot Management

Bijlage 4

Usability-aspecten

In dit hoofdstuk zal worden ingegaan op de theoretische achtergrond van usability en de usability-aspecten. Van de te testen aspecten wordt aangegeven, wat ze betekenen, hoe je ze kunt meten en hoe de criteria daarvoor te vinden zijn.

Usability

Wat is nou eigenlijk usability? Usability kan beschreven worden als de nauwkeurigheid/volledigheid, de mate waarin de resources gebruikt worden om de doelen te gebruiken en het gemak, voldoening enzovoort waarmee specifieke gebruikers hun doelen volbrengen in willekeurige omgevingen.

Met usability kom je dus eigenlijk tegemoet aan de behoefte en het gemak van de gebruiker.

Het doel dat usability met zich mee brengt is onder andere het verminderen van fouten door de gebruiker en het verhogen van de bevrediging van de gebruiker.

Usability aspecten

Usability kan opgesplitst worden in een aantal aspecten, deze zullen hieronder beschreven worden.

Learnability

Learnability staat voor het gemak waarmee nieuwe gebruikers effectieve interactie kunnen beginnen, en maximale performance kunnen bereiken. Dit aspect kan echter ook weer worden opgedeeld in een aantal deelaspecten, deze zullen hieronder kort beschreven worden:

- Predictability: Ondersteuning voor gebruikers om het effect van een te nemen actie te voorspellen.
- Synthesizability: Ondersteuning voor gebruikers om het effect van een gedane actie op de huidige toestand te verklaren.
- Familiarity: Vermogen van de gebruiker om uit ervaring met andere applicaties bij deze applicatie te gebruiken.
- Generalizability: Mogelijkheid die het systeem biedt om user actions bij andere taken hier toe te passen.
- Consistency: Vergelijkbaarheid in de interactie bij gelijksoortige taken.

Helpfulness

De omvang waarin het product de gebruiker ondersteuning biedt is de omschrijving die hoort bij Helpfulness. Het is dus de mate van ondersteuning die de gebruiker krijgt. Hierbij kan je denken aan feedback die de gebruiker krijgt.

Affect

Het emotionele gevoel dat de gebruiker krijgt tegenover de software. Het kan dus zijn dat de gebruiker zich erg aangetrokken voelt tot de software of natuurlijk juist niet.

Flexibility

Dit aspect verwijst naar de verschillende manieren waarop gebruikers en “systeem” informatie uitwisselen. Ook dit aspect bestaat uit een aantal deelaspecten welke hieronder uitgewerkt zullen worden:

- Dialogue initiative (initiatief bij de dialoog): In hoeverre kan de gebruiker zelf beslissen wat/ waar gebeurd.
- Multi-threading: In hoeverre kan de gebruiker verschillende taken tegelijk uitvoeren.
- Task migratability (overdraagbaarheid van taken): De mogelijkheid om (delen van) taken aan het systeem of andere gebruikers over te laten (tegelijkertijd). Hierbij kan je denken aan een spellingcontrole.
- Substitutivity: Toestaan dat er gelijkwaardige manieren bestaan om verschillende soorten input/ output te kunnen geven. Denk hierbij aan bijvoorbeeld de hoogte van een tabel op te geven in cm of inches.
- Customizability: De mogelijkheid om de user interface naar eigen voorkeur in te richten.

Robustness

Robustness geeft aan in hoeverre de taken in de relatie gebruikerssysteem onder alle omstandigheden goed uitgevoerd kunnen worden. Hieronder zullen een aantal deelaspecten van robustness beschreven worden:

- Observability: Het vermogen van de gebruiker om de interne toestand van het systeem te weten.
- Recoverability: Het in staat stellen om een fout te herkennen en te repareren.
- Responsiveness: De mate waarin de communicatie tussen systeem en gebruiker optreedt (de stabiliteit ervan, hoeveelheid, snelheid enz.)
- Task conformance: De mate waarin het systeem de gebruiker ondersteunt bij de taak uitvoering en daarbij aansluit bij de taakmodellen en mentale modellen van de gebruikers.