



# Een waardevaste IT-investering voor Quote

VIRTUEEL KANTOOR MOGELIJK DOOR .NET FRAMEWORK EN WEB SERVICES

**QuoteOffice is een via internet, mobiele telefoon, i-mode en pocket PC bereikbaar virtueel kantoor gericht op zakelijk en privé-gebruik. Welke technologieën zijn gebruikt voor de ontwikkeling van dit virtueel kantoor op het internet? .NET laat zich bij de bouw van QuoteOffice van zijn beste kant zien.**

Voordat begonnen is met de bouw van QuoteOffice is gekeken wat de beste technologie zou zijn. Uitgangspunten die hierbij een rol speelden zijn gebruiksvriendelijkheid, goede performance en uitstekende beveiliging en toegang vanaf zoveel mogelijk apparaten waaronder de PC, pocket PC en mobiele telefoon i-mode.

Daarnaast waren er technische eisen opgesteld, namelijk:

- schaalbaarheid;
- 7 x 24 uur beschikbaarheid;
- onderhoudbaar;
- meerlagenarchitectuur;
- multimediaal;
- partners via web services integreerbaar;
- synchronisatie met Outlook over het internet.

Meerdere technieken zijn geëvalueerd zoals Oracle/Java op Unix/Linux, Microsoft DNA en Microsoft .NET. Na een uitvoerige selectie is gekozen voor het .NET Framework, dat destijds nog in bèta-vorm bestond. Doorslaggevend voor .NET waren vooral web service-ondersteuning, productiviteit, performance, beveiliging, de standaardondersteuning

voor meerdere apparaten en de kosten-efficiency.

## Architectuur

De architectuur is volledig componentgebaseerd en objectgeoriënteerd opgezet

waardoor de onderhoudskosten beheersbaar zijn geworden.

Delen van de Microsoft voorbeeld-architecturen (IBuySpy Store en Portal, Duwamish, PetShop en Fitch and Mather) zijn in de pilot getoetst en waar

### Wat houdt QuoteOffice in?

Alle QuoteOffice-abonnees kunnen op overzichtelijke wijze hun agenda, e-mail en bijvoorbeeld adresgegevens raadplegen én beheren. Hierbij wordt gebruik gemaakt van de Microsoft Mobile Toolkit, die onderdeel uitmaakt van het Microsoft .NET-ontwikkelplatform. De QuoteOffice-applicatie is overigens de eerste i-mode-toepassing die gebruik maakt van .NET in Europa. De QuoteOffice-applicatie ondersteunt meer dan 80 verschillende mobiele devices.

Daarnaast kunnen zakelijke gebruikers al hun gegevens, zoals bestanden, email, adresgegevens, taken en memo's delen met andere gebruikers die hiertoe rechten gekregen hebben. Groot-zakelijke gebruikers krijgen ook nog een CRM-mogelijkheid om hun klantenbestand efficiënt te beheren. Hierdoor kunnen groepen personen productief en projectmatig samenwerken waar en wanneer ze maar willen.

#### Feiten:

Oprichtgever	: QED BV
Projectuitvoerder	: BIG5 BV
Server software	: Windows 2000 Advanced Server, IIS 5, SQL Server 2000 Enterprise Edition
Technologieën	: Mobile Toolkit, Web Services, ASP.NET, C#, Web Forms, Server Controls, XML, XLST, .NET Framework, Visual Basic 6.0, SOAP Toolkit, Visual Studio.NET, QuikSoft, Crystal Reports.NET, SourceSafe
Ontwikkelmethodiek	: Rational Unified Process, DSDM



mogelijk hergebruikt om het project een goede start te geven.

De architectuur van QuoteOffice bestaat uit meerdere lagen:

- *Presentatielaag*

De presentatielaag is opgebouwd uit HTML en JavaScript code. Om een goede responsetijd te verwezenlijken, wordt met JavaScript functionaliteit uitgevoerd aan de client-kant waar dat zinvol is, zodat niet elke keer een round trip naar de server nodig is. ASP.NET detecteert automatisch welke browser gebruikt wordt en genereert Javascript waar mogelijk. De programmeur hoeft hier niks extra's voor te bouwen.

In de bronbestanden van het project zijn code en opmaak strikt gescheiden. ASP.NET biedt de hiervoor benodigde architectuur in de vorm van code behind. De logica die nodig is om een pagina correct te kunnen tonen, is met de taal C# in de code behind classes gebouwd. Deze code communiceert met de business logic-laag voor de pagina onafhankelijke code en data access.

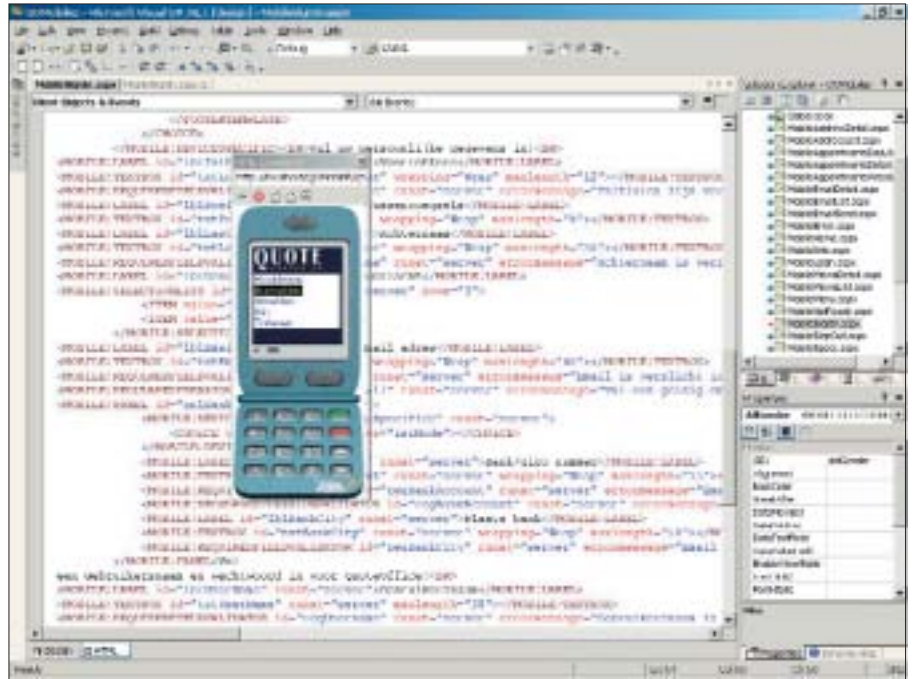
De keuze voor C# is gemaakt vanwege de in het projectteam aanwezige ervaring met Visual C++ en Java, waar de taal C# erg op lijkt wat syntax betreft. Overigens had ook VB.NET gebruikt kunnen worden, of een combinatie van C# en VB.NET. De .NET runtime omgeving is namelijk taalafhankelijk.

- *Business-laag*

Deze laag bevat de functionaliteiten die niet specifiek voor een pagina bedoeld zijn. Hierbij moet gedacht worden aan de logica die nodig is om een agenda-item te plaatsen, zoals de validaties die hiervoor nodig zijn en de aansturing van de data-laag om benodigde gegevens op te halen en weg te schrijven. Door middel van een classesstructuur is de code functioneel geordend. Dit komt herbruikbaarheid en onderhoudbaarheid ten goede.

De business laag is de enige laag die via de data access-laag gegevens manipuleert. Samen met een beveiligings-laag op de fysieke database, wordt hierdoor gegarandeerd dat er geen ongecontroleerde datawijzigingen plaatsvinden.

Om performance-redenen zijn veel functies als static functies gecodeerd. Dit



Afbeelding 1. Ontwikkelen voor mobiele devices met VS.NET

scheelt het instantiëren van een class, en zolang het een niet-afhankelijke functieaanroep betreft, is dit acceptabel. Wel moet rekening gehouden worden met situaties waarin veel bewerkingen plaatsvinden. De reden hiervan is dat in multithreading-situaties het gevaar bestaat dat er onverwachte resultaten optreden doordat dezelfde code door verschillende threads tegelijkertijd wordt doorlopen (zogenoemde race-condities). In die gevallen kan het lock() statement hulp bieden. Een betere oplossing (vanuit de objectgeoriënteerde gedachte) is echter om in zulke gevallen een class te instantiëren.

- *Data access-laag*

De data-laag biedt een geabstraheerde toegang tot de verschillende datastores. Hierdoor is het mogelijk om op een duidelijke wijze uit verschillende databronnen te putten, zonder dat hiervoor de functionaliteiten in business-laag aangepast moeten worden. Binnen QuoteOffice wordt gebruik gemaakt van SQL Server, XML-bestanden en web services. De opvolger van ADO, ADO.NET biedt standaard de mogelijkheid om met SQL databases en XML-bestanden te communiceren. Beide methoden leveren uiteindelijk een gevulde dataset op die binnen de business-laag gebruikt kan worden. Dezelfde dataset wordt gevuld met de XML gegevens die na

een web services-aanroep worden ontvangen.

## Database met Stored Procedures

Om security- en performanceredenen gebruikt QuoteOffice stored procedures. Stored procedures zijn gecompileerde stukken code die binnen de database-omgeving draaien en gegevens teruggeven. De aanroep van stored procedures is zeer eenvoudig geworden door met name Visual Studio.NET. Deze ontwikkelomgeving heeft veel geïntegreerde mogelijkheden om met de database server te communiceren en een database object aan te maken en te wijzigen. De code om bijvoorbeeld met de database te communiceren kan automatisch gegenereerd worden, hetgeen de ontwikkelaar veel tijd scheelt.

## Visual Studio.NET, Rational XDE en SourceSafe

Om een applicatie als QuoteOffice te kunnen ontwikkelen, is een diversiteit aan tools gebruikt. Visual Studio.NET is een ontwikkelomgeving waar een ontwikkelaar alles kan vinden wat nodig is van ontwerp tot deployment. Door de integratie met SourceSafe en bijvoorbeeld Rational XDE voor het modelleren is alles in één oogopslag te zien. Wanneer code geschreven is, dan hoeft een ontwikkelaar maar drie

slashes (///) in te typen en een basis voor de automatisch te genereren documentatie wordt getoond. Taken kunnen overall in de code ingevoegd worden en kunnen in een centrale lijst bijgehouden worden. Voor QuoteOffice betekent dit dat onze ontwikkelaars de modellen, documentatie, ASP.NET -code, C#-code, stored procedures en de testbrowser vanuit één centrale omgeving kunnen beheren.

### Mobiel internet

De Microsoft Mobile Internet Toolkit integreert volledig met Visual Studio.NET. De ontwikkelaar gebruikt dezelfde omgeving om applicaties te maken die op een groot aantal mobiele apparaten kan werken. De ontwikkelaar hoeft de specifieke details per device niet te weten, daar zorgt de toolkit voor. Wanneer een mobiel apparaat een pagina aanroept, detecteert de afhandelingsroutine automatisch welke code gegenereerd moet worden. Zo is voor QuoteOffice een i-mode-variant gemaakt, door de presentatielaag te veranderen. Afbeelding 1 toont een ontwikkelpagina van QuoteOffice voor een i-mode-pagina en de emulator waarop getest kan worden, wanneer geen mobiel apparaat voorhanden is.

### Performance tuning

Om ervoor te zorgen dat de gebruiker een goede responsetijd krijgt van de

webpagina's, zijn de standaard tuning-mogelijkheden van het .NET Framework gebruikt. Via het instellen van parameters is in zeer korte tijd een indrukwekkende performancewinst te behalen.

Door gebruik te maken van sessions kunnen gegevens voor een langer traject bewaard blijven. Door het instellen van parameters geeft men aan waar deze gegevens bewaard worden. Hierdoor is het up en out scalen van de applicatie heel eenvoudig.

Er zijn meerdere mogelijkheden om de performance te verhogen door caching. Daar ASP.NET van zichzelf al snel is, zijn deze caching-mechanismen pas belangrijk voor zware hits op het systeem. In QuoteOffice is gebruik gemaakt van:

- *Output caching.* Per request en per browser wordt voor zolang als nodig de webpagina bewaard zodat deze niet opnieuw gegenereerd moet worden als het niet nodig is. Dit caching-mechanisme wordt ingezet bij de pagina's die niet steeds veranderen zoals de login- en FAQ-pagina's.
- *Fragment caching.* Delen van pagina's die gecached kunnen worden, worden gemarkeerd zodat je deze niet elke keer opnieuw hoeft te genereren. Fragment caching is ingezet voor onder meer de menustructuren die niet of zelden veranderen. Door het instellen

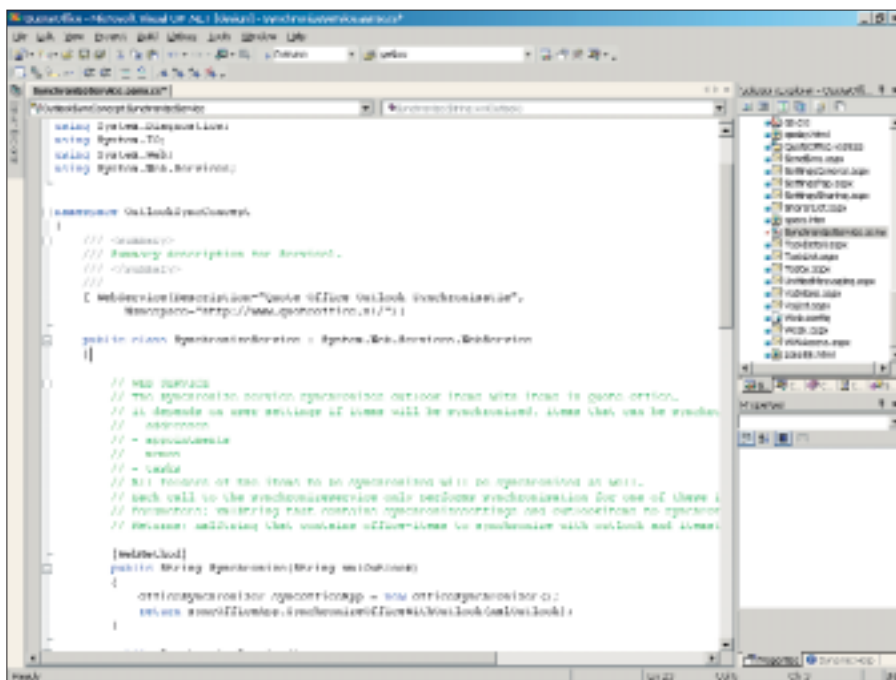
van parameters is het mogelijk om de menu's opnieuw uit de data source te halen bij veranderingen.

- *Input caching.* Informatie van de databronnen worden bewaard zolang deze niet veranderen. Hier is geen gebruik van gemaakt binnen QuoteOffice omdat de standaard cachingmogelijkheden van Microsoft SQL Server 2000 Enterprise Edition prima werken.
- *Data caching.* Alle gegevens die bewaard kunnen worden tijdens sessies worden bewaard in sessie-objecten. Dit is met name handig voor relatief statische databasegegevens, zoals lookuplijsten en dergelijke. Bij QuoteOffice is met stress-testen bekeken hoe deze objecten zich gedragen. Hoewel data caching heel stabiel is, hebben we toch zoveel mogelijk gekozen voor de andere caching-mechanismen en .NET-mogelijkheden zoals de viewstate.

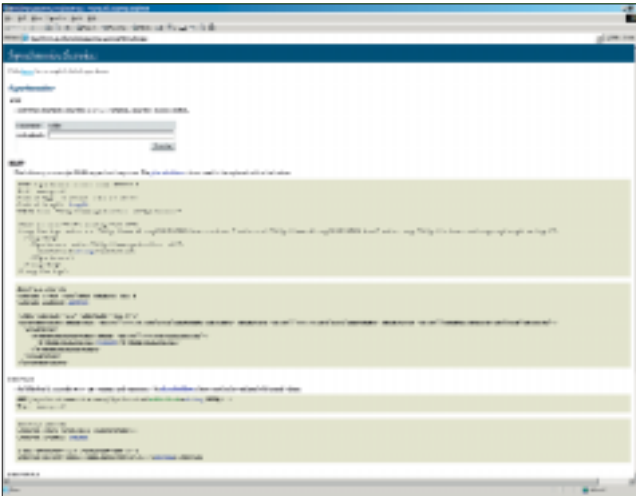
Doordat je ASP.NET -code vooraf compileert, is de performance van een ASP.NET -applicatie vele malen beter dan andere webgebaseerde oplossingen. Voor QuoteOffice betekende dit onder meer een besparing in hardwarekosten.

### Web services

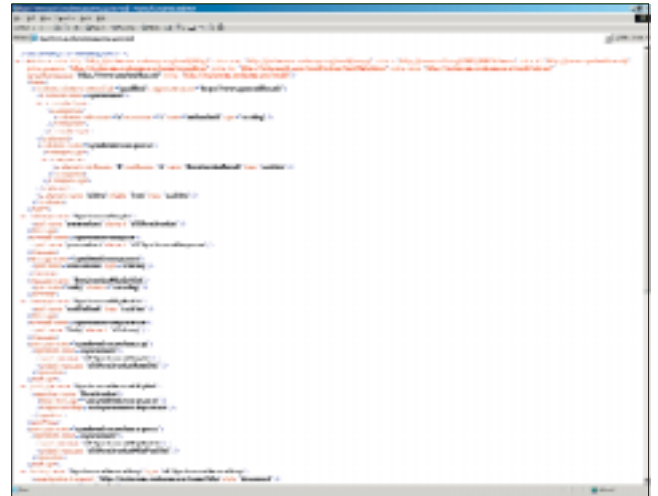
QuoteOffice bevat functionaliteit om te synchroniseren met Outlook. Deze synchronisatie werkt twee kanten op. Deze functionaliteit is vanwege de beoogde uitbreidbaarheid voor andere Organizer clients met een web service gerealiseerd. Een sterke kant van van .NET is de bibliotheek van functies die het ontwikkelen en het gebruik van web services eenvoudig maken. In Visual Studio.NET is met een paar muisklikken een .asmx file te maken waarin de definitie van een web service staat. In afbeelding 2 is te zien wat het resultaat hiervan is. De gemaakte klasse overerft van System.Web.Services.WebService. De methodes die de web service aan de buitenwereld beschikbaar stelt, worden voorzien van het [WebMethod] attribuut. De rest verzorgt de compiler en VS.NET. Web services worden door .NET uitgerust met een standaard testpagina. Wanneer iemand de web service via een browser benadert, resulteert dat in een scherm dat de methoden van de service



Afbeelding 2. Het .asmx bestand voor de QuoteOffice Outlook-synchronisatie web service



Afbeelding 3. De testpagina voor de synchronisatie-web service van QuoteOffice



Afbeelding 4. De Outlook synchronisatie web service client

toont. Wanneer je een methode selecteert, kan deze online getest worden. In het geval van QuoteOffice wordt het scherm van afbeelding 3 getoond. Voor het maken van deze pagina's is geen extra werk nodig, want ze worden automatisch meegegenereerd.

Het aanroepen van een web service en het ontvangen van het resultaat ervan, gebeurt via een SOAP-bericht (Simple Object Access Protocol). De structuur hiervan wordt getoond op de testpagina. Om deze QuoteOffice web service te kunnen aanroepen moet je een XML-bestand als parameter string meesturen. Deze XML string is natuurlijk niet een willekeurige. De clients van deze web service dienen de specificaties te weten, om een juist formaat XML te kunnen sturen via het SOAP-bericht (en ook het resulterende XML-bestand te kunnen verwerken). Op deze manier is een generiek synchronisatiemechanisme aanwezig, zonder dat QuoteOffice voor elke offline client aangepast hoeft te worden! De synchronisatie-client die momenteel voor QuoteOffice beschikbaar is, is de Outlook-synchronisatie-client. Deze kan van de QuoteOffice site worden gedownload.

De testpagina's zijn bedoeld voor menselijke interactie, terwijl de technische specificatie van een web service in een WSDL-bestand (Web Service Description Language) staat. Deze kan opgevraagd worden door de asmx-pagina van de web service op te vragen met de '?wsdl' parameter. Een deel van WSDL voor de synchronisatie-web service staat in afbeelding 4. Deze definitie is met name

van belang bij het maken van een client-applicatie, ook wel een webservice consumer genoemd.

Omdat QuoteOffice ook met oudere versies van Outlook moet synchroniseren, kon de oplossing niet geheel in .NET geschreven worden. Het is op dit moment ook niet zo dat iedereen de benodigde .NET runtime op zijn machine geïnstalleerd heeft staan. Natuurlijk staat het concept van web services los van .NET en gelukkig biedt Microsoft voor niet .NET-omgevingen ook een oplossing. Door middel van de SOAP toolkit en Visual Basic 6.0 is het ook mogelijk om web services te consumeren. Hoewel de SOAP toolkit de meeste complexiteit voor zijn rekening neemt, is het maken van een web service client in .NET met VS.NET eenvoudiger en vooral intuïtiever. Via een web reference naar de WSDL-URL wordt de web service-definitie in het project gelinkt, als was het een zelfgeschreven class. Daarna kunnen de methoden aangeroepen worden, zoals dat ook op 'gewone' objecten gebeurt. Het enige waar je op moet letten, is dat de web reference ververst moet worden wanneer de web service definitie is gewijzigd. Dit kan niet automatisch gebeuren.

## Performance testing

Door het gebruik van Homer (Web Application Stress Test) is de QuoteOffice-applicatie vele tientallen miljoenen keren getest op meerdere multi processor-systemen onder zware load en zelfs in de bèta 1 versie gaf bovengenoemde archi-

tektuur geen krimp. De nieuwe ADO.NET access-laag naar SQL 2000 Enterprise Edition heeft een gemiddelde performance boost gegeven van meer dan 300% ten opzichte van de vorige ADO-versie. Hierdoor zijn minder servers nodig voor dezelfde performance-criteria. Ook leverde de gecompileerde ASP.NET -pagina's een serieuze winst op ten opzichte van ASP 3.0 (met en zonder COM+ integratie). Zie <http://www.gotdotnet.com> voor extra performance tuning tips.

## Een waardevaste IT-investering

Quote heeft dankzij BIG5 en .NET een unieke value-propositie in de markt kunnen zetten tegen een fractie van de oorspronkelijk budgetteerde kosten. Door de toepassing van standaardproducten en -componenten is Quote verzekerd van een waardevaste IT-investering die moeiteloos de veranderingen binnen .NET nu en in de toekomst bij kan benen.

Door de efficiëntere manier van programmeren binnen Visual Studio.NET en C# worden projecten sneller afgerond waardoor Quote voor minder geld meer functionaliteit krijgt. De gewenste functionaliteit voor QuoteOffice had nooit gebouwd kunnen worden binnen de gestelde budgetten zonder de vernieuwende technologie van het .NET Framework en Visual Studio .NET.

### Nuttige internetadressen

- <http://www.qo.nl>
- <http://www.gotdotnet.com>
- <http://msdn.microsoft.com>