



Avdelning för datavetenskap

Jonas Kihlström & Nikolay Johannessen

Konvertering av webbapplikationer från ASP till ASP.NET

Migration of web applications from ASP to ASP.NET

Examensarbete 10p
Dataingenjör

Datum: 06-01-19
Handledare: Johan Eklund
Examinator: Stefan Lindskog
Ev. löpnummer: C2006:05

Konvertering av webbapplikationer från ASP till ASP.NET

Jonas Kihlström & Nikolay Johannessen

Denna rapport är skriven som en del av det arbete som krävs för att erhålla en kandidatexamen i datavetenskap. Allt material i denna rapport, vilket inte är vårt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

Jonas Kihlström & Nikolay Johannessen

Godkänd, 060119

Handledare: Johan Eklund

Examinator: Stefan Lindskog

Sammanfattning

Arbetet som denna uppsats beskriver är utfört på uppdrag av Two i Sverige AB. Two är ett konsultbolag som i första hand arbetar med utveckling av hemsidor. Företaget har flera av sina färdiga applikationer för bl.a. uppbyggnad av hemsidor skrivna i språket ASP. Man har på företaget sett vinster med att konvertera dessa applikationer från ASP till den mer flexibla miljön ASP.NET. Motivering till konverteringen är att företaget anser att man efter konvertering till ASP.NET kommer att kunna förkorta leveranstiden av färdiga produkter till kund, samt lättare skapa hemsidor uppbyggda med moduler. Idag är Two:s samtliga applikationer skrivna i ASP. Företaget vill även ha en dokumentation över konverteringsprocessen för att spara tid och att skapa en gemensam standard inför kommande konverteringar.

Denna uppsats beskriver konverteringsprocessen för två av företags applikationer och även dokumentation för hur konverteringsprocessen bör se ut.

Resultatet av arbetet är två konverterade applikationer som fungerar på .NET ramverket och har samma funktionalitet som de ursprungliga applikationerna. Dokumentationen av konverteringsprocessen har resulterat i en hemsida som är belägen på företags intranät.

Migration of web applications from ASP to ASP.NET

Abstract

This dissertation is carried out on behalf of Two i Sverige AB. Two is a consulting agency which primarily works with the development of homepages. The company has several of its complete applications written in ASP e.g. the development of homepages. At the agency they have seen benefit migrating these applications from ASP to the more flexible environment ASP.NET. The motivation for the migration is that the company hopes that by using this technique they will be able to shorten the delivery time of completed products to customers and to develop homepages created in a more modular way. All of Two's applications are today written in ASP. The company has requested documentation of the migration process to save time and to create a common standard for forthcoming migrations.

This dissertation describes the migration process of two applications made by "Two i Sverige AB" developed for the Internet. Also included is documentation of how the migration process should be performed.

The result of the project is two migrated applications that work with the .NET framework and have the same functionality as the original applications. The documentation of the migration process has resulted in a homepage that is situated on the company's intranet.

Innehållsförteckning

| | | |
|----------|---|-----------|
| 1 | Inledning | 1 |
| 2 | Motivering till konvertering från ASP till ASP.NET | 3 |
| 2.1 | Introduktion | 3 |
| 2.2 | Målsättning | 3 |
| 2.3 | Kravspecifikation..... | 3 |
| 2.3.1 | Programspråk | |
| 2.3.2 | Utvecklingsmiljö | |
| 2.3.3 | Avgränsning | |
| 2.4 | Active server pages och Active server pages.NET | 4 |
| 2.4.1 | Active Server Pages | |
| 2.4.2 | .NET | |
| 2.4.3 | Active Server Pages.NET | |
| 2.4.4 | Jämförelse mellan Active Server Pages och Active Server Pages.NET | |
| 2.5 | Sammanfattning | 8 |
| 3 | Beskrivning av befintliga Webblikationer® | 9 |
| 3.1 | Introduktion | 9 |
| 3.1.1 | Nyhetsfunktion | |
| 3.1.2 | Webbenkät | |
| 3.1.3 | Processdokument | |
| 3.2 | Sammanfattning | 15 |
| 4 | Strukturering av uppgiften | 17 |
| 4.1 | Introduktion | 17 |
| 4.2 | Tänkbara lösningar | 17 |
| 4.3 | Motivering av val för detta projekt | 18 |
| 4.3.1 | Nyhetsfunktion | |
| 4.3.2 | Webbenkät | |
| 4.3.3 | Processdokument | |
| 4.4 | Sammanfattning | 20 |
| 5 | Implementation | 21 |
| 5.1 | Introduktion | 21 |
| 5.2 | Nyhetsfunktion | 21 |
| 5.2.1 | Beskrivning av konverteringsförloppet | |
| 5.2.2 | Inställningsmöjligheter | |

| | | |
|----------|---|-----------|
| 5.3 | Webbenkät | 22 |
| 5.3.1 | Beskrivning av konverteringsförloppet | |
| 5.3.2 | Importrutin | |
| 5.3.3 | Accesskontroll | |
| 5.4 | Processdokument | 23 |
| 5.5 | Sammanfattning | 25 |
| 6 | Resultat..... | 27 |
| 6.1 | Nyhetsfunktion | 27 |
| 6.2 | Webbenkät | 27 |
| 6.3 | Processdokument | 27 |
| 6.4 | Framtida utvecklingsarbete..... | 27 |
| 6.4.1 | Nyhetsfunktion | |
| 6.4.2 | Webbenkät | |
| 6.4.3 | Processdokument | |
| 7 | Slutsatser..... | 29 |
| 7.1 | Detta projekt | 29 |
| 7.1.1 | Nyhetsfunktion | |
| 7.1.2 | Webbenkät | |
| 7.1.3 | Processdokument | |
| | Referenser | 31 |
| A | Skärmbilder av processdokumentet..... | 33 |
| B | Implementationsbeskrivning för Webblikation® | 37 |

Figurförteckning

| | |
|---|----|
| Figur 1 Förloppet för en begäran av en ASP-sida..... | 5 |
| Figur 2 Illustration av ramverket för .NET | 6 |
| Figur 3 Modellbeskrivning av ASP.NET..... | 7 |
| Figur 4 Skärmbild av www.rydahls.se där nyhetsfunktionen används..... | 9 |
| Figur 5 Skärmbild av www.rydahls.se där nyhetsdetaljer visas. | 10 |
| Figur 6 Skärmbild av administrationssida för nyhetsfunktionen | 11 |
| Figur 7 Skärmbild från personlista för Webbenkät..... | 12 |
| Figur 8 Skärmbild från undersökningslista för Webbenkät | 13 |
| Figur 9 Skärmbild för hantering av sändande av undersökningar | 14 |
| Figur 10 Genereringsverktyg för ASP.NET..... | 18 |
| Figur 11 Översikt över processdokumentet | 24 |

Terminologi

.NET

Ett ramverk som innehåller färdiga objekt för utveckling och exekvering av programvara och som fungerar som ett ramverk.

API, *Application Programming Interface*

Gränssnitt som möjliggör för ett program att använda färdiga objekt.

ASP, *Active Server Pages*

Scriptbaserat språk som används för att skapa internetapplikationer

ASP.NET, *Active Server Pages.NET*

En utveckling av ASP. Den är en del av Microsofts .NET ramverk

CGI, *Common Gateway Interface*

Ett standardgränssnitt mellan http-server och externa program.

DLL, *Dynamic Link Libraries*

Fil som innehåller exekverbar kod som allokeras under körning istället för under länkning.

IIS, *Internet information server*

Microsoft-utvecklad internetserver.

ISAPI, *Internet Server Application Programming Interface*

Ett programgränssnitt från Microsoft.

PDF, *Portable Document Format*

Ett elektroniskt dokument som måste läsas med Adobe® Reader®.

SQL, *Structured Query Language*

Språk som används för att kommunicera med databaser.

VB, *Visual Basic*

Programspråk som skapats av Microsoft.

Webblikationer®

Applikationer som dels används som byggstenar för uppbyggnad av hemsidor och dels som fristående interaktiva applikationer för Internet. Webblikationer® är ett registrerat varumärke.

1 Inledning

Two i Sverige AB [1] är ett företag som specialiserar sig på utveckling av hemsidor. Företaget grundades 2001 och är ett resultat av en sammanslagning av två företag med breda kunskaper inom utveckling av hemsidor. Deras historia härstammar från databas- och programutveckling inom IT och reklam på konsultbasis. Under det senaste året har företaget expanderat och etablerat sig som ett ledande webbutvecklingsföretag i Värmland. Two har ett flertal färdiga applikationer som dels används som byggstenar för uppbyggnad av hemsidor, dels som fristående interaktiva applikationer för Internet. Exempel på sådana byggstenar är en så kallad Kalender och Nyhetsfunktion. Two har valt att kalla sina färdiga applikationer för Webblikationer® som är ett registrerat varumärke. Dessa är skrivna i Active Server Pages (ASP) [2].

I dagsläget är inte Webblikationerna® så lättimplementerade som företaget skulle önska. Nyare sätt att utveckla program har lett till att företaget vill konvertera sina gamla program skrivna i ASP till den nyare teknologin Active Server Pages.NET (ASP.NET) [3]. Övergången från ASP till ASP.NET är en tidskrävande uppgift, som dock kan komma att ge vinster i form av kortare leveranstid mot kunden och ett mer effektivt sätt att arbeta på för utvecklarna. Hemsidor som företaget utvecklar kommer då att kunna vara mer modulatorierade, vilket innebär att varje Webblikation® i en hemsida är självständig. Detta gör det möjligt för varje del att fungera helt oberoende av andra delar och på så sätt kan man på ett smidigt sätt anpassa hemsidan efter begäran från kund.

Uppgiften, som denna uppsats behandlar, är både att utföra konverteringar av två enkla Webblikationer® från ASP till ASP.NET och att skriva ett dokument som beskriver en konverteringsprocess. De två Webblikationerna® är en så kallad Nyhetsfunktion och en så kallad Webbenkät. Dokumentet som beskriver konverteringsprocessen kommer att bestå av en hemsida på företagets intranät. Hemsidan kommer att användas som en enkel manual för framtida konverteringar av Webblikationer® inom företaget samt information om företagets standarder och krav. Manualen kommer även att innehålla mallar för hur dokumentation av Webblikationer® ska se ut.

När denna uppsats refererar till .NET, avses teknologierna i Microsoft.NET ramverket.

2 Motivering till konvertering från ASP till ASP.NET

2.1 Introduktion

I detta kapitel finns beskrivningar av både Active Server Pages (ASP) och ASP.NET, skillnader, fördelar och nackdelar är avsnitt som kommer att belysas. Dessutom kommer projektet i sin helhet att beskrivas samt motivet till att företaget vill göra förändringar.

2.2 Målsättning

Målet är att konvertera två Webblikationer® från ASP till ASP.NET samt att skapa och dokumentera en bra och väldefinierad process för konvertering av befintliga Webblikationer®. Dokumentationen avser att vara enkel att använda för den mer ovana konverteraren. Avsikten är att dokumentationen skall bli en steg-för steg-manual som alla konverterare i företaget bör använda.

2.3 Kravspecifikation

Följande krav beslutades gemensamt med uppdragsgivaren.

- Konvertera två Webblikationer® för Internet från ASP till ASP.NET.
 - I. Nyhetsfunktion
 - II. Webbenkät
- Beskriv och dokumentera en konverteringsprocess som senare kan användas som hjälp för framtida konverteringar. Denna dokumentation har valts att kallas processdokument.
- Lämna förslag på förbättringar på de Webblikationer® som konverterats.

2.3.1 Programspråk

De konverteringar som ska genomföras måste ske till ASP.NET. Detta är ett krav från uppdragsgivaren. Detta beror på att Two bara använder sig av Microsofts produkter och detta är därmed det enda alternativet för uppgradering.

2.3.2 Utvecklingsmiljö

All programmering och utveckling sker med utvecklingsverktyget Microsoft Visual studio.NET 2003. Two använder sig av versionshanteringssystemet Microsoft SourceSafe 6.0 [4] vilket Visual Studio.NET 2003 har ett inbyggt stöd för.

2.3.3 Avgränsning

De avgränsningar som beslutats i samförstånd med Two är:

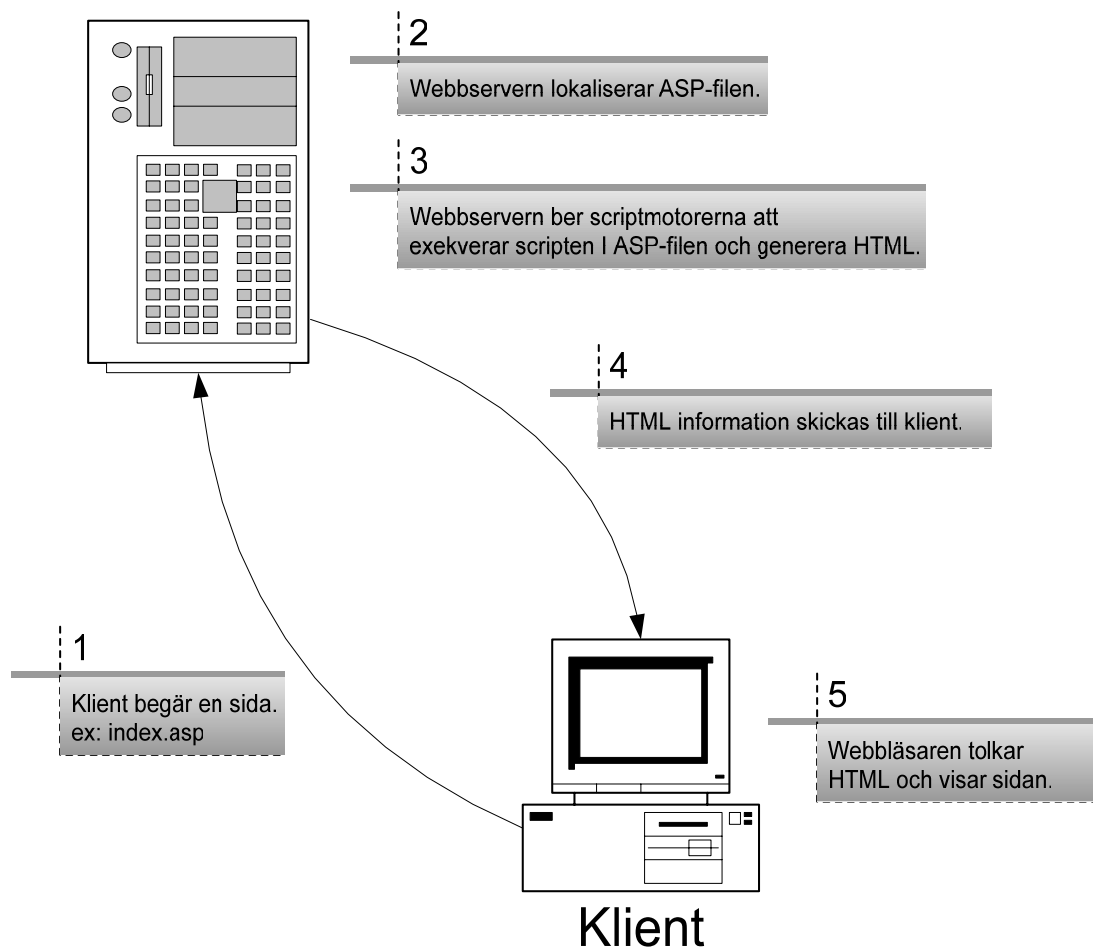
- Nyhetsfunktion och Webbenkät med samma funktionalitet som de befintliga Webblikationerna®.
- Processdokumentet kommer inte att innehålla instruktioner som beskriver specifika fall, utan generella riktlinjer.

2.4 Active server pages och Active server pages.NET

2.4.1 Active Server Pages

Active server pages (ASP) är skapat av Microsoft [5] och är ett scriptspråk för hemsidor som exekveras på webbservern och möjliggör skapande av interaktiva och dynamiska hemsidor för Internet som t.ex. *http://www.rydahls.se*. ASP används för att kunna kombinera hypertext markup language (HTML) och andra komponenter som t.ex. programkod för att kunna skapa dynamiska hemsidor. Visual basic script (VBScript) och JavaScript är de scriptspråk som ASP stödjer [6]. Genom att en sida på Internet har ändelsen ".asp" tolkar webbservern denna sida på ett annat sätt än en HTML-fil. När begäran från klienten kommer tolkar webbservern scriptet och skickar den genererade HTML-filen till klienten till skillnad från vanlig HTML-fil där allt innehåll skickas utan att tolkas av webbservern [6]. Både ASP-kod och HTML kan finnas på samma sida. ASP-filen tolkas inte innan sidan begärs av klienten utan detta sker direkt vid begäran. Ett exempel på en begäran av en hemsida skriven i ASP visas i Figur 1.

Webbserver



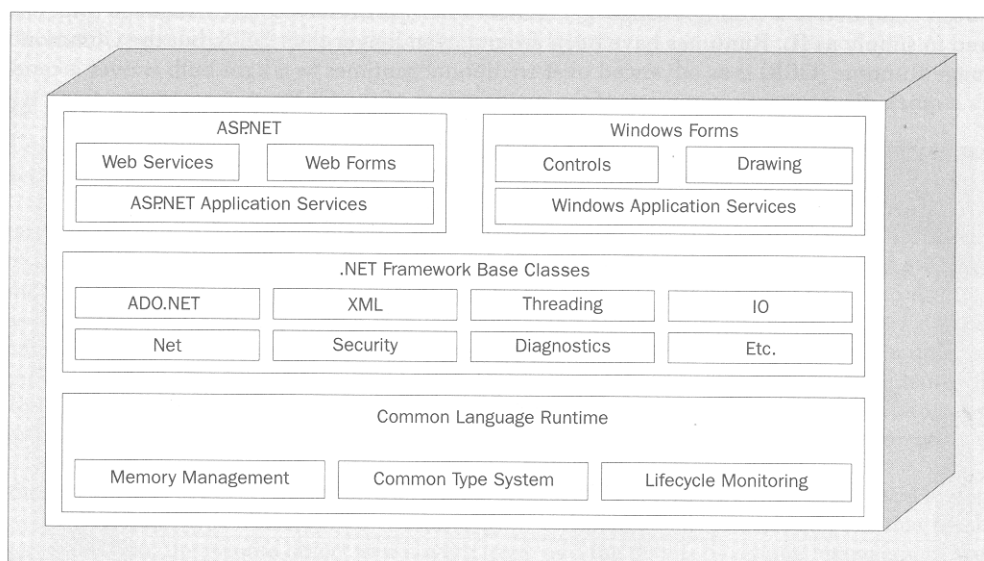
Figur 1 Förloppet för en begäran av en ASP-sida

Microsoft introducerade under samma tid som ASP även något som kallas för Internet Server Application Programming Interface (ISAPI) [7] vilket ASP använder sig av. Detta skulle vara ett alternativ till det äldre formatet Common Gateway Interface (CGI) [8] som är ett sätt för servern att kommunicera på ett enkelt sätt med externa program. CGI-applikationer har den negativa egenskapen att servern skapar separata instanser av det externa programmet varje gång applikationen begärs. Detta kan leda till att servern blir överbelastad. ISAPI mildrar problemet genom att använda sig av Dynamic Link Library (DLL) filer, som är komplicerad kod. Det är bara första gången DLL-filen initieras som den allokerar nytt minne, den kan därefter återanvändas vid flera anrop utan att behöva allokeras mer minne. DLL-filen kommer att allokeras i samma minnesområde som servern för att därigenom öka prestanda.

När DLL-filerna har allokerats finns de kvar i minnet tills de blir aktivt borttagna, vilket leder till att man sparar minne och datorkraft [9].

2.4.2 .NET

.NET tillhandahåller ett nytt Application programming interface (API), ny funktionalitet och nya verktyg för att skapa Windows- och internetapplikationer, komponenter och tjänster. .NET-ramverket lägger sig ovanpå operativsystemet som en virtuell maskin och isolerar program som är utvecklade med .NET från de flesta operativsystemspecifika tekniker som t.ex. filhantering och minneshantering. Figur 2 visar de viktigaste delarna av .NET ramverket, man kan se hur det logiskt är indelat i tre lager.



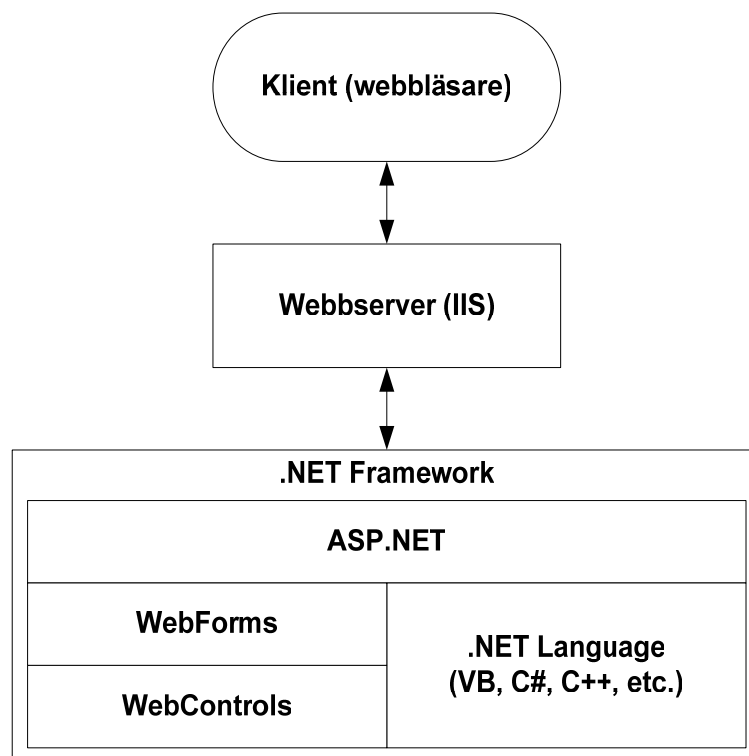
Figur 2 Illustration av ramverket för .NET

I bottenlagret ligger Common Language Runtime (CLR), vilket är själva hjärtat till .NET ramverket [10]. Den innehåller t.ex. en gemensam samling av datatyper och gemensam minneshantering. I tillägg till att sköta minneshantering tar CLR även hand om referensräkning för objekt och skräphantering (eng. garbage collector). Datatyperna följer ett standardgränssnitt vilket möjliggör för ramverket att stödja ett flertal programspråk, som t.ex. C#, VB, C++. .NET är objektorienterat uppbyggt vilket medför att alla programspråk som stöds av .NET måste stödja objektorientering [11]. I mellanlagret finns grundläggande klasser för hantering av data, processer och andra grundläggande klasser som t.ex. säkerhet och kommunikation. Dessa klasser är standardiserade vilket gör dem programspråksberoende.

Topplagret i ramverket inkluderar komponenter för gränssnitt i olika miljöer, som t.ex. webbläsare, Windowskonsollen och Windows.

2.4.3 Active Server Pages.NET

ASP.NET [3] är en del av Microsofts .NET ramverk för skapande av internetapplikationer. Sidor som innehåller grafisk information om en hemsida skriven i ASP.NET kallas för webbformulär och har filändelsen aspx. De kompilerade ASP.NET sidorna exekveras på serversidan och genererar HTML, Wireless Markup Language (WML) [12] eller eXtensible Markup Language (XML) [13] som skickas till klienten. Det är Internet Information Services (IIS) [14] som kommunicerar med .NET-ramverket för att generera HTML och exekvera kod för hemsidor som ska visas. Detta samband visas i Figur 3. ASP.NET sidorna använder en kompilerad, händelsebaserad programmodell vilket innebär förbättrad kontroll från servern genom att webbservern hela tiden får reda på vilka händelser som inträffar på klientsidan. Detta förbättrar prestanda på serversidan och möjliggör separation av programlogik och användargränssnitt.



Figur 3 Modellbeskrivning av ASP.NET

2.4.4 Jämförelse mellan Active Server Pages och Active Server Pages.NET

Genom att programmera objektorienterat kan klasser skapas som enkelt kan implementeras i ett projekt. Denna möjlighet gäller för alla programspråk .NET stödjer. Eftersom ASP.NET-sidor är kompillerade innan exekvering betyder det att prestandamässigt är det snabbare med ASP.NET än ASP då de ej kompileras utan kräver tolkning. ASP.NET använder sig av .NET ramverket med dess API, vilket medför att ASP.NET har ett större utbud på färdigbyggda komponenter, som t.ex. färdiga klasser för hantering av XML, Structured Query Language (SQL) och threading. Detta leder till tidsvinster i utvecklingsfasen då man inte behöver återskapa algoritmer som finns i .NET. En annan fördel med ASP.NET, är möjligheten att skapa användarkontroller. Dessa används för att kunna skapa objekt av hemsidor som senare kan integreras inuti andra hemsidor. Dessa objekt fungerar som klasser, vilket medför att en användarkontroll kan återanvändas på flera ställen. Ett exempel på detta är nyhetsfunktionen som används på flera ställen på hemsidan.

2.5 Sammanfattning

Detta kapitel har beskrivit vilka krav som ställts och de mål som satts upp för detta projekt. Kapitlet innehåller även en beskrivning av det programmeringsspråk som används i företagets programutveckling idag och en beskrivning av den miljö som Webbligationerna® ska konverteras till.

3 Beskrivning av befintliga Webblikationer®

3.1 Introduktion

I detta avsnitt beskrivs de två Webblikationer® som ska konverteras från ASP till ASP.NET, samt en beskrivning av hur framtida konverteringar kan utföras. De två Webblikationerna® är dels en nyhetsfunktion och dels en webbenkät.

3.1.1 Nyhetsfunktion

Nyhetsfunktionen har som uppgift att visa nyheter på en hemsida. En administratör kan lägga till, ändra och ta bort nyheter med hjälp av en lösenordsskyddad administrationssida.

För varje nyhet visas datumet då nyheten inträffar. Bredvid datumet visas den rubrik som tillhör nyheten. Denna rubrik är även en hyperlänk som länkar vidare till nyheten. Ett exempel på nyhetsfunktionen visas längst ner till höger med fet stil i Figur 4. Under rubrikraden ligger ingressen till nyheten, som är en sammanfattning av nyheten.



Figur 4 Skärmbild av www.rydahls.se där nyhetsfunktionen används

När användaren valt att klicka på länken för en nyhet, visas hela nyheten i en ny sida med eventuella bilder och dokument. Ett exempel på detta visas i Figur 5.

The screenshot shows the website for Rydahls, a specialist in brake systems. The page is titled 'Bromsdagen 2005-10-11' and reports on the company's second annual brake day. The article includes a photograph of a meeting and a link to a PDF document. The website has a dark blue header with the Rydahls logo and navigation links. A sidebar on the left lists various services, and a right sidebar contains other news items.

Rydhals.se » Tunga fordon » Nyheter

2005-10-13

Bromsdagen 2005-10-11

Rydhals genomförde sin andra Bromsdag för året.



Rydhals genomförde på Tisdagen sin andra Bromsdag för året med tyngdpunkten på Tunga Fordon. Rydhals samlade kunder och leverantörer för en trevlig och givande dag i Lofbergs Lila Arena efter en inledande rundvandring på Rydhals huvudkontor i Karlstad. Med gästande föreläsare från Wabco, Sun Maskiner samt f d förbundskaptenen i friidrott Ulf Karlsson blev dagen väldigt givande. Kvällen bjöd sedan på elitserie hockey där Färjestad på ett bra sätt rundade av kvällen med att besegra Djurgården.

Dokument: Fler bilder från Bromsdagen.pdf
Fordons avd.

Övriga nyheter

051107 - Rydhals certifierade av WABCO
Rydhals är numera en certifierad distributör av reservdelar från WABCO.

051013 - Bromsdagen 2005-10-11
Rydhals genomförde sin andra Bromsdag för året.

050831 - Utökning i Göteborg & Stockholm
Rydhals förstärker distributionen med utökning i Stockholm & Göteborg

AB ERNST HJ. RYDAHL BROMSBANDFABRIK
Brisgatan 19, Box 12, 651 02 Karlstad Tel. 054 - 85 62 00
Fax. 054 - 85 10 58 Kundservice Fordon 0771 - 27 66 72

Figur 5 Skärmbild av www.rydahls.se där nyhetsdetaljer visas.

Administrationssidan som endast administratörer kommer åt, vilket visas i Figur 6, innehåller ett flertal fält som ger möjlighet att ändra eller lägga till information till en nyhet. Det finns möjlighet att klicka på "lägg till bild" där en ny sida visas. Här kan administratören ladda upp en bild som ska visas på nyheten.

Administration - Ändra Nyhet

Du kan infoga en länk i ingress eller brödtext genom att kopiera den här texten och byta ut [adress](#) och [Rubrik](#).
 Rubrik

| | | |
|---------------------------|---|---|
| Rubrik | <input type="text" value="Bromsdagen 2005-10-11"/> | |
| Ingress | <input type="text" value="Rydahls genomförde sin andra Bromsdag för året."/> | <input type="button" value="Hämta Bild"/> |
| Brödtext | <input type="text" value="Rydahls genomförde på Tisdagen sin andra Bromsdag för året med tyngdpunkten på Tunga Fordon. Rydahls samlade kunder och leverantörer för en trevlig och givande dag i Löfbergs Lila Arena efter en inledande rundvandring på Rydahls huvudkontor i Karlstad. Med gästande föreläsare"/> | <input type="button" value="Dokument"/> |
| Bild | <input type="text" value="mote.jpg"/> | |
| Länk till dokument | <input type="text" value="Fler bilder från Bromsdagen.pdf"/> | |
| Länktext | <input type="text"/> | |
| E-mail | <input type="text"/> | |
| Underskrift | <input type="text" value="Fordons avd"/> | |
| Datum | <input type="text" value="2005-10-13"/> | |
| Visas mellan | <input type="text" value="2005-10-13"/> - <input type="text" value="2005-12-12"/> | |
| Visa på Startsidan | <input checked="" type="checkbox"/> | |

Figur 6 Skärmbild av administrationssida för nyhetsfunktionen

Databssystemet Microsoft SQL-server [15] används för att lagra nyheterna. I denna server lagras en databas med en tabell som innehåller information om varje nyhet.

3.1.2 Webbenkät

Webbenkäten används för företag som vill kunna göra en enkät, som i denna rapport är synonymt med undersökning, bland t.ex. sina kunder. Webblikationen® är uppdelad i två delar, dels en administrationsdel, dels en klientel. Från administrationssidorna kan man skapa och skicka enkäter. Det första som en administratör måste göra är att skapa eller importera personer till webbenkäten. Väljer administratören att importera, finns en importrutin för kommaseparerade filer. De importerade personerna blir sedan de till vilka är möjligt att skicka

enkäten till. Detta är en viktig funktion som måste fungera enkelt. Genom att klicka på länken personer i menyn på vänster sida som visas i Figur 7, listas alla personer i webbenkäten.

WEBBENKÄT

Personer

Välj efternamn: [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#) [Å](#) [Ä](#) [Ö](#) [Alla](#)

Visa enbart inaktiva personer

| Aktiv | Namn | E-mail | Storlek Företag | Befattning | Företag |
|----------|--------------------------------------|------------------------|-----------------|------------------|-----------------------|
| Ej aktiv | Udd, Conny | jonas.kihlstrom@two.se | 11-25 | Kontaktperson | Rejlers Ingenjörer AB |
| Aktiv | Udd, Conny | jonas.kihlstrom@two.se | 11-25 | Kontaktperson | Rejlers Ingenjörer AB |
| Ej aktiv | Uddh, Pernilla | jonas.kihlstrom@two.se | 26-50 | Kontaktperson | PROFFICE Sweden AB |
| Aktiv | Uddh, Pernilla | jonas.kihlstrom@two.se | 26-50 | Kontaktperson | PROFFICE Sweden AB |
| Aktiv | Ullbrandt, Per | jonas.kihlstrom@two.se | 1-5 | VD/kontaktperson | PuTec |
| Ej aktiv | Ullbrandt, Per | jonas.kihlstrom@two.se | 1-5 | VD/kontaktperson | PuTec |
| Ej aktiv | Ullenius, Christina | jonas.kihlstrom@two.se | 601- | VD | Karlstads universitet |
| Aktiv | Ullenius, Christina | jonas.kihlstrom@two.se | 601- | VD | Karlstads universitet |
| Ej aktiv | Ullenius, Urban | jonas.kihlstrom@two.se | 6-10 | VD/kontaktperson | Ullenius Ateljéer AB |
| Aktiv | Ullenius, Urban | jonas.kihlstrom@two.se | 6-10 | VD/kontaktperson | Ullenius Ateljéer AB |
| Ej aktiv | Unander Scharin, Per | jonas.kihlstrom@two.se | 1-5 | VD/kontaktperson | Pemasi Soft Care AB |
| Aktiv | Unander Scharin, Per | jonas.kihlstrom@two.se | 1-5 | VD/kontaktperson | Pemasi Soft Care AB |
| Aktiv | Uvefalk, Patric | jonas.kihlstrom@two.se | 11-25 | VD/kontaktperson | Uvefalk Bil AB |
| Ej aktiv | Uvefalk, Patric | jonas.kihlstrom@two.se | 11-25 | VD/kontaktperson | Uvefalk Bil AB |

Figur 7 Skärmbild från personlista för Webbenkät

När administratören har lagt till de personer som ska vara med i enkäten kan en undersökning skapas. All hantering av undersökningar sker i menyn undersökningar. I undersökningsdelen kan administratören välja att skapa, kopiera eller radera undersökningar. Det är även här som frågor skapas och länkas ihop med den undersökning som önskas. Det finns tre typer av frågor: envalsfråga, flervalsfråga och fritextsvar. Envalsfråga är en fråga där bara ett svar kan ges som t.ex. en ja eller nej fråga. Flervalsfråga är en fråga där flera alternativ ges och flera alternativ kan väljas som t.ex. val av flera kandidater. Den sista frågetypen är fritext, denna typ av fråga besvaras med ett helt fritt svar i form av text. När hela undersökningen är klar och alla frågorna länkade till den kan undersökningen skickas iväg. Genom att klicka på brevet bredvid undersökningen i Figur 8 kan personer som ska delta i enkäten väljas.

WEBBENKÄT 


- Personer
 - Storlek Ftg
 - Befattningar
 - Importrutin
- **Undersökningar**
- Resultat
- Administratörer
- Hjälp
- Logga ut

Kopiera undersökning **Lägg till undersökning**

| Undersökning | Ansvarig | Tillgänglig | |
|--|-----------------|---------------------|-------------------------------|
| <input checked="" type="checkbox"/>  35. Sofias test | Jonas Kihlström | 05-12-01 - 05-12-22 | Till frågorna |
| <input checked="" type="checkbox"/>  34. Kopia av Buggetestning | Nikolay & Jonas | 05-11-26 - 05-12-26 | Till frågorna |
| <input checked="" type="checkbox"/>  33. Buggetestning | Nikolay & Jonas | 05-11-26 - 05-12-26 | Till frågorna |
| <input checked="" type="checkbox"/>  32. Massive Test | Team Two | 05-12-01 - 05-12-08 | Till frågorna |
| <input checked="" type="checkbox"/>  31. Test Skogsstyrelsen | Jan Back | 05-11-14 - 05-12-14 | Till frågorna |
| <input checked="" type="checkbox"/>  25. Kopia av test 2005-01 | Jan Back | 05-02-04 - 06-03-06 | Till frågorna |
| <input checked="" type="checkbox"/>  23. Kopia av test 2005-01 | Jan Back | 05-02-04 - 05-03-06 | Till frågorna |
| <input checked="" type="checkbox"/>  5. test 2005-01 | Jan Back | 05-02-04 - 06-03-06 | Till frågorna |
| <input checked="" type="checkbox"/>  4. TEst | Jan Back | 04-12-10 - 05-01-09 | Till frågorna |

Figur 8 Skärmbild från undersökningslista för Webbenkät

För att välja vilka personer som ska få undersökningen via e-post väljer administratören de personer som önskas genom att klicka på deras namn. När en person valts läggs deras namn upp på listan i den undre delen av sidan som visas i Figur 9.

WEBBENKÄT 

- Personer
 - Storlek Ftg
 - Befattningar
 - Importrutin
- Undersökningar
- Resultat
- Administratörer
- Hjälp
- Logga ut



Välj Personer till undersökning: 34. Kopia av Buggtestning




Sök personer: -- Välj Storlek på företag -- VD

Välj personer: [Alla personer](#) [Alla personer som visas i listan nedanför](#)

| Namn | Stl Ftg | Befattning | Företag |
|--|---------|------------|--------------------------------------|
| Westerberg, Magnus | 1-5 | VD | HusvagnsPoolen i Värmland AB |
| Andersson, Johan | 1-5 | VD | Qell Designbyrå AB |
| Andersson, John-Erik | 11-25 | VD | Årjängs Järn AB |
| Andersson, Leif | 26-50 | VD | Plaståtervinning i Arvika AB |
| Andersson, Nicklas | 51-100 | VD | Union Wood Ek.förening |
| Andersson, Olle | 1-5 | VD | Vesuvius Scandinavia AB |
| Andersson, Roland | 1-5 | VD | Begravningsbyrån Roland Andersson AB |
| Asklund, regionchef, Johan | 101-200 | VD | Nordea Region Värmland |
| Bemler, Ulf | 51-100 | VD | Byggbolaget i Värmland AB |
| Ben-Hadj, Mohamed | 1-5 | VD | MTD Karlstad AB |
| Bergkvist, Hans | 11-25 | VD | KAMIC Karlstad Automatic AB |
| Bexelius, Per | 1-5 | VD | Bex Komponent AB |
| Blomgren, Lars | 101-200 | VD | Fimek AB |
| Boberg, Jan | 51-100 | VD | Outkumpu Stainless Tube AB |

Valda Personer

 = Ta bort personen från undersökningen  Personen har fått undersökningen
Om en person har svarat på en undersökning ligger resultatet kvar.

| | Namn | Stl Ftg | Befattning | Företag | |
|---|----------------------|---------|---------------|---------|-----------------------------|
|   | Johannessen, Nikolay | 601- | VD | Two AB | Skicka igen |
|  | kihlström, Jonas | 101-200 | Kontaktperson | two | Skicka |

Figur 9 Skärmbild för hantering av sändande av undersökningar

När rätt personer valts kan undersökningen skickas antingen individuellt eller till alla på en gång. Det går även att skicka en påminnelse till de personer som inte svarat. En ikon kommer upp för de personer som fått en undersökning skickad till sig. För att sedan se resultatet av en undersökning kan administratören gå in på resultat. På resultatsidan visas svaren från alla användare både med procent tillsammans med de kommentarer som lämnats. Man kan även filtrera svaren efter olika grupper som t.ex. storlek på företag och befattning.

Webbenkäten innehåller även funktioner för att administrera rättigheter för administratörer. Administratörer kan antingen vara superadministratörer eller vanliga administratörer. Superadministratörer kan se alla undersökningar och även skapa administratörer medan vanliga administratörer bara kan se sina egna undersökningar och inte ändra sina rättigheter.

Till sist finns även en hjälp med beskrivning över hur webbenkäten fungerar och en knapp för utloggning.

3.1.3 Processdokument

I dagens läge har inte företaget några dokument som beskriver hur deras internetapplikationer ska designas. Eftersom inga Webblikationer® har konverterats tidigare finns det heller inga dokument som beskriver en sådan konvertering. Behovet är dock stort, eftersom efterfrågan på en snabb leverans av en ny produkt hela tiden ökar [16]. Ett dokument med bra riktlinjer för hur konverteringen skulle kunna utföras kan spara mycket tid för programmeraren och därmed förkorta leveranstiden.

3.2 Sammanfattning

I detta kapitel finns en beskrivning av de två Webblikationer® som ska konverteras, deras utseende samt deras funktionalitet. Även bakgrunden till ett processdokument har förklarats i detta kapitel.

4 Strukturering av uppgiften

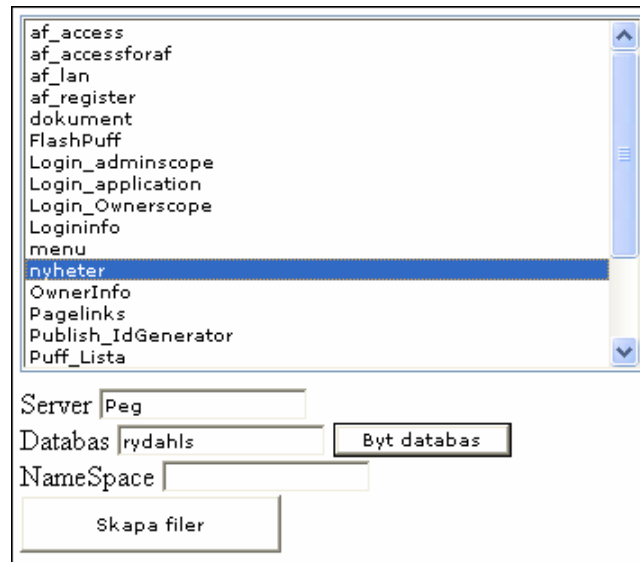
4.1 Introduktion

I detta avsnitt diskuteras olika tänkbara tillvägagångssätt för att lösa de uppgifter som kravspecifikationen innefattar. Olika val som t.ex. vad som bör konverteras och vad som bör lösas på ett nytt sätt förklaras.

4.2 Tankbara lösningar

Ett sätt att konvertera den gamla Webblikationen® är att genom rad för rad anpassa gammal kod till ny kod och se till att den har samma funktionalitet som innan. Det finns verktyg idag som t.ex. "ASP to ASP.NET Migration Assistant" [17], som kan anpassa ASP kod till ASP.NET bara genom några enkla knapptryckningar. Dessa verktyg för dock med sig att man inte tar till vara på flera av fördelarna med ASP.NET som t.ex. färdiga kontroller för inmatning. Microsofts konverteringsverktyg anpassar bara koden så att den fungerar på en ASP.NET plattform, den optimerar inte.

En annan lösning är att skriva om hela applikationen från början. Genom att undersöka den gamla applikationens egenskaper och funktioner, kan man skriva om dessa i det nya språket. Denna lösning kan leda till att man återuppfinner algoritmer, man skapar algoritmer som redan finns. Programmeraren som utför konverteringen kan uppleva samma problem som utvecklaren som skapade ursprungskoden hade. Fördelen är att man får en väldigt klar bild över vilka komponenter som behövs. Är programmeraren en van ASP.NET utvecklare kan han/hon använda sig av de kunskaperna om ASP.NET för att snabbare konvertera utan att titta på de lösningar som redan finns i den ursprungliga implementationen av Webblikationen®.



Figur 10 Genereringsverktyg för ASP.NET

Företaget har skapat ett ASP.NET verktyg, som visas i Figur 10, som automatiskt genererar, utifrån en tabell i en databas, ett antal s.k. .aspx-filer vilket är xml-filer som innehåller grafisk information om hemsidor, vilka kallas Webbformulär. Filerna innehåller en koppling mot en databas, en listningssida och en redigeringsida. Listningssidan är en sida som visar innehållet från en tabell i databasen. Redigeringsidan är en sida för att redigera de fält som finns i tabellen. På redigeringsidan kan man också radera och lägga till rader i tabellen. Detta verktyg kräver att programmeraren lägger till och implementerar metoder i den automatiskt genererade koden för att anpassa applikationen efter behoven. Denna typ av uppslagning mot en databas sker på flera av de Webblikationer® som Two har.

4.3 Motivering av val för detta projekt

4.3.1 Nyhetsfunktion

Valet blev att kombinera företagets genereringsverktyg, med att skriva om delar av applikationen från början. Klasser med databaskoppling från den automatiskt genererade koden kommer att användas, sedan byggs nyhetsfunktionen manuellt utifrån denna. Detta ansågs vara det bästa sättet att utveckla nyhetsfunktionen, eftersom man då lättare tar vara på de funktioner som finns i .NET. I och med att man skriver applikationen från början kan man lättare strukturera upp uppbyggnaden, samt skapa en mer lättimplementerad Webblikation®.

Nyhetsfunktionen är en Webblikation® som till största delen representerar en databastabell. Den kod som genereras från genereringsverktyget som nämns i stycke 4.2 passar bra till nyhetsfunktionen då den skapar presentationen automatiskt. Det som sparas från den genererade koden är den uppkoppling som sker mot databasen. Den innehåller all information som behövs för att göra uppkopplingen mot databasen som t.ex. tabellnamn, kolumnnamn och även de enklare kommandona som uppdatera, spara och radera.

Under konverteringens gång återanvänds inte alla algoritmer som existerar i den kod som ska konverteras. Det är bara i de fall där programmeraren som konverterar inte hittar en enkel lösning som återanvändning av algoritmer bör förekomma.

4.3.2 Webbenkät

Även här blev valet att kombinera företagets verktyg, med att skriva om applikationen från början. Då denna Webblikation® är betydligt större än nyhetsfunktionen, samt innehåller fler tabeller bestämdes det att en klass för hantering av generella SQL-frågor måste skapas. Detta ansågs nödvändigt eftersom det finns relationer mellan ett flertal tabeller i databasen och genereringsverktyget bara kan skapa klasser som hanterar en tabell åt gången.

Webbenkäten är en Webblikation® som är mer fristående än vad nyhetsfunktionen är, men liknar nyhetsfunktionen i sin uppbyggnad. Likheterna är att de delar som Webblikationen® består av är listor och redigeringssidor precis som i nyhetsfunktionen. På så sätt passar genereringsverktyget utmärkt för denna Webblikation®.

Det finns ett antal SQL-frågor och algoritmer som redan existerar i den nuvarande Webblikationen®, vilka bör användas under konverteringsförloppet där man anser att det behövs. Det kan finnas fall där man bör skriva om algoritmen helt för att dra nytta av .NET-ramverket.

4.3.3 Processdokument

Innan projektet startades diskuterades det att processdokumentet skulle skrivas som ett Worddokument. Detta dokument skulle vara det enda dokument som beskriver hur konverteringsprocessen skulle gå till. Problemet med detta tillvägagångssätt var dock att det skulle bli komplext och för mycket text. Tanken är att processdokumentet ska vara enkelt och

lättillgängligt vilket ett dokument av denna typ troligen inte skulle bli. Detta skulle i förlängningen betyda att dokumentet inte används.

För att kunna tillgodose de krav som ställs från företaget, skapas en hemsida. Fördelen med denna hemsida är att den kan vara lättillgänglig via företagets intranät.

Först i processdokumentet måste det finnas en uppmaning om att programmeraren måste förstå funktionaliteten i den Webblikation® som ska konverteras och även instruktioner om hur konverteringen ska gå till. Detta är en viktig förutsättning för en väl genomförd konvertering. Efter detta bör det komma en beskrivning av hur katalogstrukturen ska se ut i framtida Webblikationer®, detta för att skapa en gemensam struktur för hur programuppbyggnaden ska se ut för hela företaget. Vidare bör det komma instruktioner om vilka klasser som kommer att användas. Nu är det också viktigt att klargöra att man kan använda befintliga moduler och klasser som finns tillgängliga. Dokumentet kommer att ge tips på hur programmeringen kan gå till. I detta dokument kan man även ta med punkter som återanvändning av kod, dokumentation och andra viktiga frågeställningar som kan uppstå under programmeringen.

När konverteringen är klar måste programmeraren skriva ett implementationsdokument till programmet, som innehåller vilken funktion Webblikationen® har, hur den implementeras och vilka inställningar som kan göras. Till sist bör processdokumentet innehålla en checklista över de viktigaste punkterna som programmeraren måste ta hänsyn till under konverteringen. Detta för att programmeraren på ett enkelt sätt ska kunna kontrollera att Webblikationen® är rätt konverterad enligt given standard.

4.4 Sammanfattning

I detta kapitel har valen för lösning av de olika uppgifterna i kravspecifikationen beskrivits. Det har varit olika val mellan designer för att lösa ett och samma problem. Lösningarna som beskrivits är de lösningar som ansågs vara de bästa innan konverteringen och processdokumentet påbörjades.

5 Implementation

5.1 Introduktion

Detta kapitel innehåller en beskrivning av de applikationer som konverterats samt beskrivning och utdrag ur processdokumentet som förklarar konverteringsprocessen. Vissa funktioner beskrivs mer noggrant då dessa lösningar skiljer sig programmeringsmässigt från det nuvarande.

5.2 Nyhetsfunktion

5.2.1 Beskrivning av konverteringsförloppet

I början av konverteringen skapades en katalogstruktur som ansågs vara så generell att den kommer att kunna användas som en standard för Webblikationer® skrivna i ASP.NET. Denna struktur separerar administrationsdelen och klientdelen i två delar. I respektive del delas även objekt upp som t.ex. klasser och bilder.

En hjälpklass för databasuppkopplingar skriven i VB.NET fanns redan i företaget, denna klass sköter uppkopplingen mot databasen. Den är generellt skriven och kan hantera olika typer av databasuppkopplingar mot t.ex. SQL och Access-databaser. Det är denna hjälpklass som de genererade klasserna är beroende av när uppkoppling sker eftersom det är denna klass som innehåller uppkopplingsrutiner. För nyhetsfunktionen genererades enbart en klass då Webblikationen® endast använder sig av en tabell. Denna klass passar väl in på de krav på metoder som behövs för att göra de nödvändiga databasoperationer, som t.ex. skapande av en nyhet.

För att kunna implementera nyhetsfunktionen på ett enkelt sätt på en hemsida valdes användarkontroller. Användarkontroller förklaras i stycke 2.4.4. Denna lösning valdes för att kunna integrera nyhetsfunktionen på en redan existerande hemsida. För att kunna testköra alla användarkontroller skapades hemsidor. Detta på grund av att användarkontroller inte kan visas som en hemsida, utan måste integreras på en hemsida för att kunna visas.

5.2.2 Inställningsmöjligheter

På grund av den nya möjligheten till användarkontroller i ASP.NET kan inställningar ske under körning. I web.config, som är en konfigurationsfil som används av servern, lagras ett flertal inställningar för Webblikationen®, dessa inställningar kan t.ex. vara sökvägar, uppkopplingssträngar, osv.

Användarkontrollen för översikten av nyheter har tre inställningsmöjligheter för att ändra på utseendet. Den första inställningsmöjligheten är hur många nyheter som ska visas. Den andra inställningen anger det maximala antal tecken den kortfattade beskrivningen av nyheten får ha. Tredje och sista inställningen är möjligheten att bestämma om en nyhet ska visas på startsidan.

5.3 Webbenkät

5.3.1 Beskrivning av konverteringsförloppet

Webbenkäten har liknande struktur och uppbyggnad som nyhetsfunktionen. De har samma katalogstruktur och i de flesta fall samma typ av genererade databaskopplingar. Skillnaden är att webbenkät inte blir en del av en annan hemsida utan kommer att vara fristående. Detta gör att valet blev att inte använda användarkontroller i lika stor grad.

I de fall där det finns relationer mellan tabeller har det visat sig vara snabbare och mindre krävande för servern att använda sig av mer komplexa SQL-frågor som t.ex. frågor som spänner över flera tabeller, än de enklare frågor som de genererade klasserna erbjuder. En generell klass skapades för att kunna hantera denna typ av komplexa frågor.

5.3.2 Importrutin

Importrutinen som finns i den befintliga webbenkäten innehåller många kodrader med många tidskrävande algoritmer. Beslut togs i tidigt skede att den skulle skrivas med hjälp av .NET-metoder. I .NET finns metoder för att importera kommaseparerade filer till minnet med endast ett fåtal rader kod. Denna metod ledde också till att importering av Excel-blad möjliggjordes, vilket uppdragsgivaren påpekat som en bra förbättring.

Importrutinen innehåller kontroller för alla fält såsom E-post och namn för att säkerställa fältens validitet innan lagring i databas. Kontroller finns även för att alla fält har fyllts i. Att denna kontroll sker är viktigt för att uppehålla databasens integritet. För att

underlätta konfigurering av importrutinen finns en konfigurationsfil som innehåller information om den kommaseparerade filen. Konfigurationsfilen kan ändras under exekvering utan att användaren behöver ändra i källkoden. Den konfigurationsmöjlighet som finns är möjligheten att ha sidhuvud i den kommaseparerade filen som ska importeras. Sidhuvud är första raden i den kommaseparerade filen som namnger varje kolumn, finns inget sidhuvud måste varje kolumn definieras manuellt i konfigurationsfilen.

5.3.3 Accesskontroll

För att kontrollera behörigheten hos en användare för administrationsdelen används .NETs ramverk. Detta sker genom att i web.config ange vilka kataloger och filer som inte ska ha anonym åtkomst. När användaren loggar in kontrolleras användarnamn och lösenord mot databasen. En s.k. cookie lagras på klientens dator för att användas som en typ av nyckel för åtkomst till de skyddade hemsidorna.

5.4 Processdokument

Processdokumentet delades upp i åtta steg. Första sidan som möter programmeraren är en översikt över stegen med länkar till en mer detaljerad beskrivning av varje steg. Skärmbilder av varje steg finns i Bilaga B. Översikten över processdokumentet kan ses i Figur 11.

| |
|--|
| Portning av webblikationer |
| 1. Lär känna webblikationen |
| Några tips före konvertering |
| Mer info |
| 2. Skapa katalogstrukturen |
| Mall för katalogstruktur |
| Mer info |
| 3. Vilka klasser behövs? |
| Information om klasser |
| Mer info |
| 4. Färdiga funktioner |
| Ta reda på om några av de funktioner som du behöver redan finns inom företaget som är skrivna i ASP.NET. |
| Mer info |
| 5. Egen kod |
| Riktlinjer för kodning |
| Mer info |
| 6. Dokumentation |
| Mall för dokumentation |
| Mer info |
| 7. Checklista |
| Kolla in före release! |
| Mer info |
| 8. Lista över webblikationer skrivna i .NET |
| Lägg till modulen som du precis portat i listan över ASP.NET webblikationer. |
| Mer info |

Figur 11 Översikt över processdokumentet

Första steget i processdokumentet förklarar nödvändigheten av att förstå webblikationen som ska konverteras. Efter steg ett kommer instruktioner och riktlinjer för uppbyggnaden av katalogstrukturen för att få en gemensam och lättigenkännlig struktur. När dessa steg är avklarade belyser steg tre frågan om vilka klasser som behövs? Den innehåller även råd om vad man bör tänka på när man ska designa klasstrukturen för en Webblikation®.

Nästkommande steg kommer att vara ett register över funktioner och algoritmer som ska vara lätta att återanvända i framtida projekt. Tips om hur man kan gå till väga när man ska börja skriva programkod ges i steg fem. När konverteringen är klar medföljer en mall för implementationsdokumentet. Denna mall finns i bilaga B. För att konverteraren sedan ska kunna kontrollera den konverterade Webblikationen®, finns en checklista. Denna checklista ska användas innan Webblikationen® kan anses vara klar för att läggas till i listan över konverterade Webblikationer®.

I det sista steget lägger konverteraren in den konverterade Webblikationen® i en lista med de Webblikationer® som är skrivna i .NET som företaget har till sitt förfogande.

5.5 Sammanfattning

Detta kapitel beskriver implementationsmetoder för nyhetsfunktionen och webbenkäten, även en beskrivning av vad processdokumentet innehåller har gåtts igenom. Bara de metoder som skiljer sig väsentligt åt från den ursprungliga Webblikationen® är beskrivna.

6 Resultat

6.1 Nyhetsfunktion

Nyhetsfunktionen färdigställdes utan några större problem och inom den tidsram som bestämts. Resultatet blev en nyhetsfunktion konverterad till ASP.NET som har samma funktioner och egenskaper som den ursprungliga Webblikationen®

6.2 Webbenkät

Webbenkäten har efter konverteringen blivit snabbare och lättare att vidareutveckla än den tidigare versionen skriven i ASP. I och med att den skrevs om helt från början kunde mycket tid och energi läggas ner på att möjliggöra för enklare vidareutveckling. Resultatet blev en webbenkät konverterad till ASP.NET med samma funktioner och egenskaper som Webblikationen® hade innan.

6.3 Processdokument

Processdokumentet har färdigställts enligt de krav som ställdes och inom den tidsram som sattes. Dokumentet finns nu tillgängligt för Two's medarbetare på företagets intranät. Planen för detta processdokument har varit att under hela examensarbetet parallellt arbeta och utveckla dokumentet samtidigt som konverteringsarbetet, vilket utförts.

6.4 Framtida utvecklingsarbete

6.4.1 Nyhetsfunktion

Eftersom nyhetsfunktionen är en enkel Webblikation® som endast innehåller elementära funktioner anses det inte nödvändigt att utöka funktionaliteten. Tanken med nyhetsfunktionen är att den ska vara lättimplementerad och en utökning kan få negativa effekter för implementationen.

6.4.2 Webbenkät

Denna Webblikation® är självständig och det leder till att det finns större utrymme för vidareutveckling och att lägga till funktionalitet. Här är funktioner som anses vara bra att lägga till:

Förbättring:

- Icke sekventiell förflyttning mellan frågor.
- Fler filtreringsmöjligheter i personlistorna.
- Skapa Portable Document Format (PDF) [18] filer från resultaten.
- Tillbakaknapp på sidan där frågorna listas
- Mall för E-post.
- Radera flera personer på samma gång.

Förklaring/Exempel:

Det bör finnas möjlighet att skapa svarsvägar som t.ex. Vid nej-svar hoppa till fråga 10.

Kunna filtrera personer på fler sätt. T.ex. ”Visa endast personer från Volvo.”

PDF-filerna genereras utifrån de svar som finns för en specifik undersökning

Administratören bör kunna skapa och utforma brevet som skickas ut till personer som idag är hårdkodat.

Administratören kan bara radera en person åt gången. Bra egenskap skulle kunna vara att T.ex. ”Radera personer från Saab”

6.4.3 Processdokument

Funktioner för att kunna registrera och lagra Webblikationer® och även algoritmer som kan återanvändas bör läggas till i processdokumentet. Detta anses nödvändigt för att enkelt hålla ordning och underlätta användandet av processdokumentet för utvecklaren.

7 Slutsatser

7.1 Detta projekt

Arbetet hos Two har varit väldigt lärorikt och givande. Arbetet har fullföljts enligt överenskommen kravspecifikation. Projektet slutfördes inom den givna tidsramen.

7.1.1 Nyhetsfunktion

Konverteringen av nyhetsfunktionen var ett projekt som gav en bra grundförståelse för vad konverteringsprocessen innebär. Anledningen till att nyhetsfunktionen var ett bra projekt att börja med, är att den innehöll ett flertal grundläggande funktioner som återfinns i flera av företagets Webblikationer®. Nyhetsfunktionens storlek har också haft en betydande roll. På grund av att nyhetsfunktionen har varit liten har det inneburit att om misstag påträffats har projektet kunnat startas om utan att förlora mycket tid.

Nyhetsfunktionen har även varit ett bra projekt p.g.a. dess enkelhet. Det har lett till att mer tid har kunnat spenderas på hur dynamiska hemsidor på Internet fungerar och hur information skickas mellan hemsidor. Denna kunskap har underlättat konverteringen av webbenkäten.

7.1.2 Webbenkät

Den tidsplan som lades i början av projektets visades sig vara allt för optimistisk. Webbenkäten uppskattades ta ca en vecka mer än Nyhetsfunktionen, denna feluppskattning berodde på dålig förundersökning då konverteringen tog två veckor längre än planerat. Webbenkätens fulla omfång hade ej uppfattats och det visades sig innehålla fler funktioner än vad som först antogs.

Webbenkäten är av större komplexitet än nyhetsfunktionen. De erfarenheter som arbetet med nyhetsfunktionen gav, var väsentliga för konverteringen av webbenkäten. Ex. på erfarenheter är listning av data och insättning i databas i ASP.NET.

7.1.3 Processdokument

Att Two är i behov av ett gemensamt dokument för hur konverteringsprocessen ska se ut är något som upptäcktes under konverteringen av nyhetsfunktionen och webbenkäten. Redan vid ett tidigt stadium sågs behovet av att sätta standarder och riktlinjer. För att kunna uppnå en bra process för att bygga hemsidor krävs det att alla Webblikationer® följer en väldefinierad standard som medför att Webblikationerna® enkelt kan sättas ihop med varandra.

Förhoppningarna är stora att företaget kan ta till sig av den hemsida som är tänkt att fungera som en informationskälla och eventuell riktlinje för företaget.

Referenser

- [1] Two's hemsida, 2005: Om Two. (2005.09.08) <http://www.two.se>
- [2] Alex Homer, Dave Sussman, Brian Francis, 2000, Professional Active Server Pages 3.0
- [3] Microsoft, 2005: ASP.NET. (2005.09.24) <http://msdn.microsoft.com/asp.net/>
- [4] Microsoft Visual SourceSafe 6.0 (2006.02.02)
<http://msdn.microsoft.com/vstudio/previous/ssafe/productinfo/default.aspx>
- [5] Microsoft, 2005: Active Server Pages. (2005.09.25)
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnanchor/html/activeservpages.asp>
- [6] Chris Payne, 2001: *Lär dig ASP.NET på 3 veckor*. ISBN 91-636-0693-3
- [7] Templeman, Julian, 2002, *Visual Studio.Net Black Book*
- [8] Rafe Colburn, 1998, *CGI-Programmering på 1 vecka*. ISBN 91-636-0516-3
- [9] A.Keyton Weissinger, 1999, *ASP in a nutshell*. California
- [10] Steven Roman, Ron Petrusha, Paul Lomax, 2002, *VB.NET Language in a Nutshell*. ISBN 0-596-00308-0
- [11] Thuan Thai, Hoang Q. Lam, 2001, *.NET Framework Essentials*
- [12] Wap Introduction, (2005.12.5) http://www.w3schools.com/wap/wap_intro.asp
- [13] Michael J.Young, 2002, *XML Steg för Steg 2:a Upplagan*. ISBN 91-636-0713-1
- [14] Anthony Northrup, 1999, *Introducing Windows 2000 Server*. ISBN 1-57231-875-9
- [15] Ryan K. Stephens, Ronald R. Plew, 2000, *Lär dig SQL på 3 veckor*. 3 uppl.
- [16] Muntlig källa: Allan Hallén, VD på Two
- [17] ASP to ASP.NET Migration Guide (2005.11.09)
<http://msdn.microsoft.com/asp.net/using/migrating/aspmig/>
- [18] Why PDF? (2005.12.01)
<http://www.adobe.com/products/acrobat/adobepdf.html>

A Skärmbilder av processdokumentet

Lär känna webblikationen

Det är ytterst viktigt att du som ska porta en webblikation kan och känner till alla de funktioner som finns.

Ställ dig dessa frågor:

- Vilka funktioner har webblikationen? (Lista gärna dessa innan)
- Hur administreras sidan?
- Är den kopplad mot någon databas?

Allt arbete du lägger ner här får du lätt tillbaka i slutet.

Tillbaka

Skärmbild av steg ett, "Lär känna webblikationen".

Skapa katalogstrukturen

Two använder en gemensam katalogstruktur på alla program. Det är bra att skapa denna struktur i Visual studio innan man börjar programmera för att få en stabil grund men även få en idé om hur programmet ska struktureras.

Så här ska katalogstrukturen se ut:

| | |
|------------------|---|
| [-] Webblikation | Detta är roten på strukturen. Det ska vara namnet på webblikationen |
| [-] Classes | Klasser ska sparas här (.vb filer) |
| [-] Images | Bilder som hör till modulen |
| [-] Pages | Här ska aspx sidor lagras. |
| [-] Admin | Adminsidor i aspx |
| [-] Usercontrols | Användarkontroller (.ascx filer) |
| [-] Admin | Adminanvändarkontroller |

Tillbaka

Skärmbild av steg två, "Skapa katalogstrukturen".

Vilka klasser behövs

Bestäm vilka klasser som behövs för projektet. Rita gärna ett diagram för att få en klar bild över programmet.

Exempel på klasser:

- Uppkopplingsklasser mot databaser
- Verktygsklasser som används på flera ställen i projektet

Tips!

Använd gärna färdiga hjälpklasser från Microsoft s.k Applikation Blocks

Tillbaka

Skärmbild av steg tre, "Vilka klasser behövs?"

Färdiga funktioner

Ta reda på om några av de funktioner som du behöver redan finns inom företaget som är skrivna i ASP.NET

Här finns alla färdig funktioner/moduler:

| | |
|--|----------------|
| Kalenderdialog | Nyhetsfunktion |
| Importrutin från CSV-fil och Excel-fil | Webbenkät |
| Tabbelgenererare (Rekomenderas vid Datalister) | Webbenkät |
| Lösenordsgenerare | Webbenkät |

OBS!

Fyll på denna lista med nya funktioner!

Tillbaka

Skärmbild av steg fyra, "Färdiga funktioner".

Egen kod

Börja skriv kod på egen hand utan att kolla på den befintliga koden. Håll hela tiden i åtanke:

- Uppfinn inte hjulet igen, använd gamla algoritmer från ursprungskoden om du har svårigheter med hur du ska lösa ett problem.
- Kommentera hela tiden koden. Använd VBCommenter och Ndoc
- Var noga med att namnge objekt med självförklarande namn.
- Håll koden snygg, inga onödiga bortkommenteringar.

Tillbaka

Skärmbild av steg fem, "Egen kod".

Dokumentation

Kom ihåg att alltid dokumentera den kod som skrivs.
Använd denna mall för att skriva implementationsdokumentationen:

Implementationsbeskrivning för (Modulnamn).doc

Spara den i samma katalog som projektet

Tillbaka

Skärmbild av steg sex, "Dokumentation".

Checklista

Kolla:

- Alla länkar så de fungerar från andra datorer.
- Resurser delas på rätt sätt
- Se till att inte ha några "Magic numbers" i koden, nummer utan beskrivning.

Tillbaka

Skärmbild av steg sju, "Checklista".

Lista över webblikationer skrivna i .NET

| Namn: | Plats: | Ansvarig: |
|----------------|-------------------|------------------|
| Nyhetsfunktion | \\green\somewhere | Nikolay & Jonas |
| Webbenkät | \\green\somewhere | Nikolay & Jonas |

Tillbaka

Skärmbild av steg åtta, "Lista över portade webblikationer".

B Implementationsbeskrivning för Webblikation®

Beskrivning

Kortfattad beskrivning av modulen.

Krav

Beskriver vilka förutsättningar som måste uppfyllas innan modulen implementeras.

Ex.

Connection.vb uppkoppling mot databas måste finnas i projektet.

Katalogstruktur

<Katalog>\<Katalog>\

Ex.

News\Classes\

News\Usercontrols\Admin\

Konfigurering

Här ska nödvändiga inställningar skrivas ner. Som t.ex. web.config och Global.asax.

Ex.

web.config (exempel värden)

<appSettings>

 <add key="PageDocumentsUploader" value="/Html/News/Pages/PageDocumentsUploader.aspx"/>

 <add key="CalendarPath" value="/Html/News/Pages/Calendar.aspx"/>

</appSettings>

Global.asax

Public Const CfgKeyPageImageUploader *As String* = "PageImageUploader"

Public Const CfgKeyCalendarPath *As String* = "CalendarPath"

Implementation

Lista användarkontroller och förklara deras funktion och inställningsmöjligheter.

<**Filnamn**>

<Enkel beskrivning>

Inställningsmöjligheter:

<**Inställning**> <Förklaring av inställning>

Ex.

newsList.ascx

Användarkontrollen används för att visa en lista över nyheter.

Inställningsmöjligheter:

nrOfNews

Antal nyheter som ska visas

introLength

Längden på ingressen

StyleSheet taggar

Nya stylesheettaggar som används i modulen.