

Systemintegration i praktiken

Utveckling av ett offert- och kundhanteringssystem i ASP.NET

Micael Ebbmar
Jan-Åke Hedström

EXAMENSARBETE

Högskolan Trollhättan · Uddevalla
Institutionen för Informatik och Matematik

Uppsats för filosofie kandidat i Datavetenskap

Systemintegration i praktiken **Utveckling av ett offert- och kundhanteringssystem i** **ASP.NET**

Micael Ebbmar
Jan-Åke Hedström

Examinator:

Lena Pareto

Institutionen för Informatik och Matematik

Handledare:

Linn Gustavsson

Institutionen för Informatik och Matematik

Trollhättan, 2003

2003:D31

DEGREE PROJECT

System integration in reality Development of a system managing tenders and customer information in ASP.NET

Micael Ebbmar
Jan-Åke Hedström

Abstract

For this project we have developed a web based system to manage tenders, customer meetings and follow-ups within these two areas for Optinet Data AB, a computer security company in Gothenburg. Optinet has during some time been working with three different file registers to store and maintain information about tenders, customer meetings and follow-ups. Their working procedures have not been efficient and there is a need to integrate the three existing file registers into one solid system. Continuous meetings have been held with Optinet during the progress of the project to achieve a good and thorough result with a high degree of usability.

This report describes the analysis of how the company works, where we investigate how they store and maintain their information. It also describes their routines working with tenders, customer management and follow-ups. We have created a design based on the analysis of how a desirable system will look and how information will be managed. This design of the interface was created with help by the graphical tool in Microsoft Visual Studio .NET.

The work with the realization is also described in detail. The interface which was developed during the design is connected to the functionality, which we during the analysis concluded should be a part of the realization. Yet again Microsoft Visual Studio .NET have been used, this time together with the framework ASP.NET and the programming language Visual Basic .NET. The realization has been the most time consuming part of the project. It is followed by a thorough debugging and testing phase of the system both by users from Optinet as well as the project team. Eventually the system will be delivered to Optinet.

At the end, the result is discussed and compared to how well it meets the earlier mentioned goals. There are also recommendations to further development of the system.

Publisher:	University of Trollhättan · Uddevalla, Department of Informatics and Mathematics Box 957, S-461 29 Trollhättan, SWEDEN Phone: + 46 520 47 50 00 Fax: + 46 520 47 50 99	
Examiner:	Lena Pareto, HTU	
Advisor:	Linn Gustavsson, HTU	
Subject:	Computer science	Language: Swedish
Number:	2003:D31	Date: August 7, 2003
Keywords	Systemintegration, tender, customer, system, development, ASP.NET, Visual Basic .NET, Visual Studio .NET, SQL Server 2000, database, PIA, COM	

EXAMENSARBETE

Systemintegration i praktiken Utveckling av ett offert- och kundhanteringssystem i ASP.NET

Micael Ebbmar
Jan-Åke Hedström

Sammanfattning

Vi har utvecklat ett webbaserat system för hantering av offerter, nykundsmöten och uppföljning av dessa två verksamheter åt Optinet Data AB, ett datasäkerhetsföretag i Göteborg. Optinet har under en längre tid arbetat med tre separata filregister för att lagra och bearbeta information om offerter, kundmöten och uppföljning. Arbetssättet har inte varit effektivt och det finns behov för en integrering av de tre befintliga filregistren till ett system. Kontinuerliga möten med företaget har skett under arbetets gång för att uppnå ett bra och genomarbetat resultat med en hög användbarhet för Optinet.

Den här rapporten beskriver arbetet med att analysera hur företaget arbetar, där vi undersöker hur de lagrar och hanterar sin information samt rutinerna kring deras säljarbete. De rutinerna är offerthantering, nykundsbearbetning och uppföljning. Utifrån analysen har en design av gränssnittet och databasen skapats över det önskvärda systemet och dess informationslagring. Gränssnittet har skapats med hjälp av det grafiska verktyget i Microsoft Visual Studio .NET.

Vidare beskrivs också arbetet med realiseringen. Där kopplas det gränssnitt som under designfasen tog form till den funktionalitet som vi under analysfasen kom fram till skulle finnas. Även här har Microsoft Visual Studio .NET använts, nu tillsammans med ramverket ASP.NET och programspråket Visual Basic .NET. Realiseringen har varit den mest tidskrävande delen av arbetet. Den följs av en grundlig testfas där systemet testas av både Optinets användare och av projektgruppen. Till sist sker leverans av systemet till Optinet.

Slutligen diskuteras också resultatet och hur väl det överensstämmer med de uppsatta mål som har angivits. Dessutom ges rekommendationer till systemets fortsatta utveckling.

Utgivare:	Högskolan Trollhättan · Uddevalla, Institutionen för Informatik och Matematik Box 957, 461 29 Trollhättan Tel: 0520-47 50 00 Fax: 0520-47 50 99		
Examinator:	Lena Pareto, HTU		
Handledare:	Linn Gustavsson, HTU		
Huvudämne:	Datavetenskap	Språk:	Svenska
Nivå:	Fördjupningsnivå 1	Poäng:	10
Rapportnr:	2003:D31	Datum:	2003-08-07
Nyckelord:	Systemintegration, offert, kund, system, utveckling, ASP.NET, Visual Basic .NET, Visual Studio .NET, SQL Server 2000, databas, PIA, COM		

Förord

Det här examensarbetet uppkom som en idé av Optinet genom kontakten med en av projektets deltagare, Micael Ebbmar. Han hade året innan sökt sommarjobb på Optinet och då det var dags att leta efter ett examensarbete skickades en förfrågan dit. Optinets VD, Rune Wallgren svarade med att de hade ett förslag på ett utvecklingsarbete vilket ledde till det här examensarbetet.

Båda projektdeltagarna har gemensamt med stort engagemang och mycket tid fullgjort det här examensarbetet. Fördelningen av arbetet och den tid deltagarna lagt ner har varit lika. Arbetet har dock fördelats i två ansvarsområden. Micael Ebbmars ansvar var uppladdning av filer till databasen och interaktionen med Word. Jan-Åke Hedströms ansvar har varit databasstrukturen och visning av information i komponenten datagrid. Projektdeltagarna har varit insatta i båda ansvarsområdena och de beslut som fattats har tagits gemensamt, då de olika delarna har varit beroende av varandra.

Vi vill tacka Optinet för ett bra samarbete och ett lärorikt examensarbete. Dessutom vill vi tacka de personer som varit till hjälp för oss i examensarbetet. Däribland Jonas Amosson för hjälp med databasstrukturen, Andreas Boklund för hjälp med projektdirektiv, projektplan och övriga synpunkter samt Malte Hildingsson för hjälp med JavaScript.

Ett särskilt tack vill vi rikta till vår handledare Linn Gustavsson och vår examinator Lena Pareto. Utan er vägledning och värdefulla synpunkter hade inte arbetet och rapporten blivit vad de idag är.

Micael Ebbmar

Jan-Åke Hedström

Innehållsförteckning

Abstract.....	ii
Sammanfattning.....	iii
Förord.....	iv
Nomenklatur.....	1
1 Inledning.....	2
2 Bakgrund.....	2
3 Problemformulering.....	3
4 Förutsättningar.....	4
5 Målsättning.....	4
6 Avgränsningar.....	5
7 Metod.....	5
8 Genomförande.....	6
8.1 <i>Analys</i>	6
8.2 <i>Kravspecifikation</i>	14
8.3 <i>Design</i>	17
8.4 <i>Realisering</i>	29
8.5 <i>Testning</i>	40
8.6 <i>Leverans</i>	41
9 Reflektion.....	42
9.1 <i>Analys</i>	42
9.2 <i>Design</i>	42
9.3 <i>Realisering</i>	43
9.4 <i>Testning</i>	43
9.5 <i>Leverans</i>	43
10 Slutsats.....	44
10.1 <i>Rekommendationer till fortsatt utveckling</i>	44
Källförteckning.....	46
Figurförteckning.....	47

Appendix

Appendix A	Projektdirektiv	1
Appendix B	Projektplan	4
Appendix C	Offertregister	10
Appendix D	Nykundsbearbetningsregister	12
Appendix E	Flödesschema nuvarande arbetsgång	13
Appendix F	Flödesschema nuvarande arbetsgång	14
Appendix G	Entitets/rerelationsdiagram	15
Appendix H	Objekt/termkatalog	16
Appendix I	Navigering i gränssnitt	19
Appendix J	Namnstandard	20
Appendix K	SQL-skript	23
Appendix L	Datagrid	25
Appendix M	Systemdokumentation	26
Appendix N	Programkod	38

Nomenklatur

Offert - Ett tidsbegränsat anbud på en specifik tjänst eller produkt som erbjuds till kund.

Säljmöte - Det interna möte Optinet har vanligtvis två gånger i månaden, där de diskuterar den gångna periodens offerthistorik, nya kunder samt uppföljning av dessa två verksamheter.

Nykundsbearbetning - Innefattat arbetet med att boka det första besöket med en ny eller potentiell kund samt mötesuppföljning.

Offertregister - Ett Excelregister där offertspecifik information lagras; såsom kundnamn, säljare, avseende, ekonomisk data etc.

Uppföljning - Process där offert- och nykundshistorik går igen om och diskuteras för bestämning av nya verksamhetsmål.

Underlag - Den sammanställning av produkt/tjänst information som ligger till grund för offertgivning. Innehåller information såsom inköpspris, täckningsgrad, täckningsbidrag etc.

Avtal - Överenskommelse gällande tjänst. Specifik för typen av tjänst.

Offertfiler – Begrepp för alla filer rörande en offert. Inkluderar underlag, offert, avtal och övriga filer.

Totalbelopp - Försäljningssumma av produkt/tjänst.

Täckningsbidrag - Den vinst Optinet tjänar på en försäljning.

Täckningsgrad - Procentuell vinst på en försäljning.

Löpande tjänst - Innebär löpande inkomst från t.ex. webbhotell och är ingen engångsförsäljning.

Tagen offert - Besvarad offert som accepterats av kund.

Tappad offert - Besvarad offert som nekats av kund.

Obesvarad offert - Offert som ännu inte har besvarats. Detsamma som utestående offert.

Reviderad offert - Offert som förnyas och uppdateras, vanligtvis efter en kunds önskemål.

Optinet Data AB - Fullständigt namn för uppdragsgivaren. I texten används Optinet.

1 Inledning

För att företag idag skall kunna arbeta effektivt och möta marknadens konkurrens behövs ändamålsenliga verktyg. De är ofta datorbaserade och skall stödja den dagliga verksamheten samt hantera den information företagen använder och behöver. Kraven på verktyg varierar beroende på verksamhetens inriktning och företagets arbetssätt samt hur dess organisation är uppbyggd. En viktig faktor är att dessa verktyg är anpassade till att stödja den nuvarande verksamheten för att inte behöva anpassa verksamheten till verktygen.

Datorbaserade system hanterar information såsom t.ex. kundhantering, fakturering, offertgivning, verksamhetshistorik och målstyrning. Dessa områden är många gånger administrativa och det finns behov för att förenkla och automatisera dem för att kunna fokusera på företagets affärsidé. För att tillgodose ett företags alla behov krävs oftast ett flertal olika verktyg. De använder ofta delvis samma information och bör därför till viss del vara integrerade med varandra.

Ett datorbaserat verktyg är inte alltid tillräckligt för en hel verksamhet då de flesta verktyg är specialiserade inom ett specifikt område. När ett verktyg anses otillräckligt eller då ett företag till viss del byter arbetssätt, kan det bli kostsamt att införskaffa ett nytt verktyg. Därför är det lämpligt att utveckla kompletterande komponenter vilka kan integreras till de befintliga verktygen.

Vi presenterar ett företag som anser att ett av deras datorbaserade verktyg är otillräckligt. Det ger inte tillräckligt stöd till deras arbetssätt eller till det informationsbehov som finns. Vidare beskrivs utvecklingen av ett nytt verktyg som ett komplement till deras befintliga system. Projektet kommer bl.a. innehålla en studie av företaget med fokus på deras nuvarande arbetsrutiner och på hur information används. Resultatet analyseras och en lösning tas fram på ett verktyg anpassat till deras behov.

2 Bakgrund

Optinet Data AB (nedan kallat Optinet) är ett företag med nio anställda, placerat i Göteborg. Optinet är inriktat mot datarelaterade tjänster och produkter för små och medelstora företag, allt ifrån webbhotell till säkerhetsanalys och säkerhetsförebyggande åtgärder på Windowsplattformar.

Idag använder Optinet ett datorbaserat system, Visma Business (nedan kallat Visma) [1]. Visma är ett affärssystem med inriktning på ekonomistyrning och innehåller funktioner för kundhantering, fakturering och offerthantering. Enligt Optinet fungerar kundhanteringen och faktureringen bra, medan funktionen för offerthanteringen inte uppfyller deras behov och anses därför otillräcklig. I Visma saknas också helt funktioner för bearbetning av nya kunder samt uppföljning av säljarbete. För att kringgå dessa brister har de vid sidan om Visma skapat egna rutiner vid behandling av

informationen rörande offerthantering, nykundsbearbetning och uppföljning i programmet Microsoft Excel (nedan kallat Excel). De rutinerna består i att lagra informationen i tre separata filregister.

Anledningen till att Optinet inte införskaffat ett nytt komplett offerthanteringssystem är bl.a. att deras nuvarande arbetssätt har fungerat någorlunda tillfredställande för dem. De anser också att ett helt nytt verktyg troligtvis skulle innehålla för många funktioner som företaget inte skulle använda sig av. Arbetssättet och verktygen har en del brister och därför anser Optinet att behov finns för kompletterande verktyg. För två år sedan påbörjades utvecklingen av en webbapplikation för offerthantering. Den färdigställdes dock aldrig eftersom alla problem inte kunde lösas vid den tidpunkten.

3 Problemformulering

Att Optinet använder tre oberoende filregister medför ett antal problem. Det ger ett merarbete då delar av informationen är redundant. När informationen ändras i ett register måste användaren ibland även uppdatera de andra registren för att informationen skall vara konsistent. Här finns det en risk med att uppdateringar glöms bort vilket kan leda till att felaktig information lagras. Det saknas kontroller i deras nuvarande rutiner för att upptäcka felaktig och inkonsistent information.

Ytterligare problem är att åtkomsten av informationen blir begränsad då endast en användare åt gången kan arbeta med filregistren. Informationen är inte heller åtkomlig utifrån företaget. Det är en nackdel då Optinets säljare ibland behöver få tillgång till information om t.ex. offerter ute hos en kund.

Problem uppstår också vid uppföljning av säljarbete med avseende på offerthantering och nykundsbearbetning. Excel saknar delar av den funktionalitet Optinet anser sig behöva och det är främst Excels filtreringsfunktion tillsammans med summeringen av visad information de anser vara otillräcklig.

För att filregistren inte skall bli för omfattande skapas nya register för varje år. Det leder till flera register och svårigheter att snabbt ta fram kundhistorik samt att spåra äldre offerter. Flera filregister innebär att sökning av information blir tidskrävande och omständigt. Det saknas också helt versionshantering av reviderade offerter. Offertregistret innehåller endast information om den senaste versionen av en offert. Även det bidrar till svårigheter att ta fram kundhistorik.

Problemen medför ett arbetssätt där rutinerna inte är naturligt och stegvis utformade. Det ger ett manuellt merarbete samt större tidsåtgång vid hantering av information kring offerter, nykundsmöten och uppföljning än vad som är önskvärt.

4 Förutsättningar

Grunden för att åtgärda de problem som finns är att specificera vilka förutsättningar som finns för arbetet. Det finns en del givna förutsättningar i Optinets verksamhet. De inriktar sig mot Windows miljöer och de är *Microsoft Certified Partner*¹. Det är därför naturligt att projektet inriktar sig mot denna miljö. Resultatet av projektet kommer att placeras på en av företagets servrar där Microsoft Windows 2000 Server, Microsoft Internet Information Services 5 (nedan kallat IIS 5) och Microsoft SQL Server 2000 (nedan kallat SQL Server 2000) finns installerat. På servern finns också egna funktioner för autentisering och säkerhetskopiering.

Optinet vill fortsätta att använda Excel för att skapa offertunderlag och Microsoft Word (nedan kallat Word) för att skapa offerter. De är invanda vid det arbetssättet och det underlättar för säljarna att kunna kopiera information mellan dokumenten. Därför skulle det försvåra för dem om den funktionaliteten lades till i ett annat gränssnitt.

5 Målsättning

Målet är att det framtida systemet skall stödja Optinets verksamhet och förenkla deras rutiner. Systemet skall designas för att ta tillvara de bra aspekterna i det nuvarande arbetssättet och komma tillrätta med de problem och brister som finns idag.

Gränssnittet skall utformas så att en naturlig och för Optinet önskvärd arbetsgång finns. Arbetsgången skall förenklas och vara utformad stegvis där varje moment är ett naturligt steg. Risken för att eventuella uppdateringar glöms bort skall minskas.

De program som används idag kommer till viss del fortfarande att användas. Dock är det en målsättning att informationslagringen blir central, så att information ej blir redundant. Kontroller skall finnas för att inmatad information delvis är korrekt och att lagrad informationen är konsistent. Informationen skall vara åtkomlig av flera anställda samtidigt och även åtkomlig utanför företagets intranät. Detta p.g.a. att säljarna ibland behöver tillgång till information vid arbete hos kund.

För att kunna hantera olika versioner av reviderade offerter kommer versionshantering användas i systemet. Det ger möjlighet till att spåra äldre offerter, för att få en återblick och för att göra det möjligt med en komplett kundhistorik. Kundhistoriken kommer på sikt att växa och även antalet lagrade filer. Därför bör också en arkiveringsfunktion skapas, där äldre filer kan kopieras från systemet och arkiveras.

Vid Optinets interna säljmöten skall det finnas möjlighet att skapa ett korrekt diskussionsunderlag. Detta för att lättare kunna ta fram och diskutera resultatet sedan förra säljmötet och utifrån det bestämma nya mål med avseende på nya kunder och

¹ Innebär för Optinets del fullständig tillgång till Microsofts programvarubibliotek och fri teknisk support.

lämnade offerter. Sökkriterier bestämmer vilken information som visas och det skall vara möjligt att direkt uppdatera informationen.

6 Avgränsningar

Ingen kryptering vid kommunikation med Optinets webbserver är nödvändig och kommer därför inte att implementeras.

Inmatning av fullständig offertdata via gränssnittet till offertfilen skall inte implementeras då Optinet önskar fortsätta att arbeta med Excel och Word för att skapa offertunderlag och offerter.

7 Metod

En användarorienterad metod kommer att användas under projektets gång. Regelbundna möten och kontinuerlig kommunikation med uppdragsgivaren kommer att ske under hela projektets genomförande. Användarna kommer att vara delaktiga i utvecklingen och framtagandet av systemet för att ge dem en ökad möjlighet att påverka slutprodukten.

Vi har med Livscykelmodellen [2] som grund delat in arbetet i fem faser; analys, design, realisering, testning och leverans. Indelningen ger en övergripande bild av hur projektet kommer att genomföras. Det är en plan för vad som skall göras och i vilken ordning det skall ske. Varje fas i genomförandet av projektet ligger till grund för nästkommande fas vilket säkerställer projektets genomförande. En stor vikt läggs på analys- och designarbetet som är en viktig grund till det fortsatta genomförandet av projektet.

Under analysfasen utreds hur Optinet arbetar idag och en nulägesbeskrivning skapas. I beskrivningen visas Optinets nuvarande arbetsgång och verktygens stöd till verksamheten. Beskrivningen tar också upp hur information lagras och vad som lagras. I kommunikation med arbetsgivaren diskuteras de aspekter som är bra och dåliga med det nuvarande arbetssättet. Utifrån detta skapas ett förslag på ett framtida arbetssätt, där de positiva aspekterna behålls och där nya delar infogas för att möta företagets behov och uppsatta mål.

I analysfasen kommer följande tre dokument att skapas vilka är viktiga för projektets genomförande:

Projektdirektiv (se Appendix A) – Ett initialt underlag skapas tillsammans med uppdragsgivaren för hur projektet kommer att genomföras och vad projektet omfattar. En kort övergripande bild ges om bakgrund, problem, önskemål och möjlig åtgärd. Projektet avgränsas, tillgängliga resurser specificeras och en tidsplanering skapas.

Projektplan (se Appendix B) – En utökning från projektdirektivet där bakgrund och projektbeskrivning är mer omfattande. Projektets avgränsningar, strategi och metodval specificeras samt delas in i aktiviteter vilka tidsplaneras veckovis. Projektplanen tar också upp den budgeten som finns tillgänglig och den organisation som finns runt projektet. Avslutningsvis specificeras önskad slutprodukt och önskvärd effekt av slutprodukten. Till dokumentet skapas två bilagor; Gantt-schema och riskanalys. Gantt-schemat är en förtydning av tidsplaneringen i diagramform. Riskanalysen är viktig för att identifiera de risker projektet kan drabbas av. Att vara medveten om dem gör projektdeltagarna förberedda på att snabbare kunna lösa oförutsedda problem.

Kravspecifikation (se kap. 8.2) – En beskrivning av vilka egenskaper det kommande verktyget skall ha och vilka funktioner som bör ingå. Här tas också upp dokumentation av verktyget och om eventuell utbildning kommer att ske.

Designfasen utgår från gjord analys och ett lösningsförslag skapas. Där specificeras teknisk lösning och utvecklingsmiljö samt programspråk väljs. Vi specificerar hur information skall lagras samt vad för information som skall lagras. Vidare skapas ett förslag på hur verktygets gränssnitt skall se ut och hur användarna kommer att interagera med det.

I realiseringsfasen skapas systemet utifrån resultatet av analys- och designfasen. Fasen utgörs till största delen av programmering. Skapade komponenter testkörs individuellt och fortlöpande under fasens genomförande.

Under testningsfasen testas den mer färdiga produkten på arbetsdatorn av både projektgruppen och Optinet's användare. Testningen rör funktionalitet i gränssnittet, databasen och förståbarhet. Resultatet av testningen skall fastslå om tidigare specificerade mål och krav har uppfyllts.

Leverans till Optinet av det färdiga systemet sker på en CD-skiva tillsammans med system- och användardokumentation.

8 Genomförande

Genomförandet av projektet sker enligt de steg specificerade i metodkapitlet (se kap. 7). Arbetet under faserna analys, design, realisering, testning och leverans beskrivs och de val, frågeställningar och problem som uppstått under arbetets gång redovisas.

8.1 Analys

En viktig del för att skapa en användbar produkt är att sätta sig in i och förstå hur uppdragsgivaren arbetar. Vid projektstart tillbringade vi två veckor hos Optinet där vi studerade deras arbetssätt och diskuterade de problem och begränsningar de upplever. Analysfasen leder till att en kravspecifikation tas fram inför designen av det nya systemet, vilket Optinet har valt att döpa till *Säljman*.

8.1.1 Hur arbetar Optinet idag?

Optinet har idag en kundbas på ca 500 kunder och deras verksamhet inriktar sig mot små och medelstora företag som till största delen verkar inom Västra Götalandsregionen. De inriktar sig endast mot Windowsmiljöer och erbjuder såväl produkter som tjänster. Produktsortimentet består av maskin- och programvara såsom arbetsstationer, servrar, nätverksutrustning samt applikationer. De tjänster som erbjuds är framförallt säkra Internetanslutningar med bl.a. VPN², brandväggar, viruskydd, säkerhetskopiering, säkra produktionsmiljöer och webbhotell.

De delar av Optinets verksamhet som berörs av detta projekt kan delas in i fyra delverksamheter; informationslagring, offerthantering, nykundsbearbetning samt uppföljning. Hur Optinet arbetar med dem skiljer sig och även fast de är närbesläktade specificeras delverksamheterna var för sig. Under informationslagring redovisas hur Optinet lagrar informationen främst med avseende på de tre andra delverksamheterna. Offerthantering avser arbetssättet och informationsbehandlingen kring den offertering Optinet utför gentemot deras kunder. Nykundsbearbetning innebär separat hantering av det första mötet med en ny kund och hur uppföljning av kunden skall ske. Den sista delverksamheten uppföljning innebär hur Optinet följer upp delverksamheterna offerthantering och nykundsbearbetning. Det sker under de återkommande säljmötena Optinet har.

8.1.1.1 Informationslagring

Optinets informationshantering sker huvudsakligen genom Visma, där information om kunder och fakturering hanteras. Visma lagrar informationen i en SQL Server 2000 databas. Den består av en tabell med 166 attribut och har en helt platt struktur utan relationer. Där lagras bl.a. namn på kunder, leverantörer och säljare i en kolumn som kallas *nm*. För att rätt information skall tas fram vid behov använder Visma andra kolumner för att identifiera om posten är en kund, leverantör eller säljare. Vid identifikation av kunder används kolumnen *CustNo* där ett sexsiffrigt kundnummer finns inmatat om det är en kund. På liknande sätt identifieras leverantörer med kolumnen *SupNo* som innehåller ett sju-siffrigt leverantörsnummer. Säljare identifieras via fältet *EmpNo* genom att det innehåller ett tal. Det talet startar på ett och räknas upp för alla Optinets anställda. De intressanta kolumnerna för projektets del visas i figur 8-1.

² Virtual Private Network – Är en krypterad förbindelse mellan två platser ofta tillhörande samma verksamhet över t.ex. Internet [3].

	Nm	CustNo	SupNo	EmpNo
▶	Optinet	0	1000001	0
	...	0	1000002	0
	...	0	1000003	0
	...	0	1000004	0
	...	100002	0	0
	...	100003	0	0
	...	100004	0	0
	...	100005	0	0
	...	0	0	2
	...	0	0	3
	...	0	0	7

Figur 8-1 Använda fält ur Vismas databas.

Kundhanteringen och faktureringen i Visma fungerar bra. Till offerthantering räcker dock inte Vismas inbyggda funktioner till och funktioner för nykundsbearbetning och säljmöte saknas dessutom helt. P.g.a detta har Optinet skapat egna rutiner för dessa tre funktioner vid sidan om Visma. De består i tre filregister skapade i Excel och finns lagrade på en server endast åtkomlig inom Optinet.

8.1.1.2 Offerthantering

En offert är ett prisförslag på en produkt eller tjänst och lämnas till kund. Kunden har sedan vanligtvis ca 20 dagar på sig att besvara offerten. För att skapa en offert måste Optinets försäljare först skapa ett underlag. Det görs i Excel (se figur 8.2, Appendix C för fullstorlek) och innehåller de kostnader Optinet har för att införskaffa de produkter och utföra de tjänster kunden efterfrågar. På de kostnaderna läggs sedan en vinstmarginal till. Offerten skapas i Word utifrån en .dot-mall³ genom att information klipps ut från underlaget och klistras in i offerten. Förutom dessa två filer skall också ett avtal skapas i de fall en kund önskar en löpande tjänst. Avtal finns färdiga beroende på tjänst och skickas med vid behov. I ca 50 % av fallen vid lämnande av offerter vill kunden revidera offerten. För att inte förlora informationen i den ursprungliga offerten skapas en ny version där ändringarna görs.

Vid offertering hanteras mycket information. Den lagras i ett offertregister skapat i Excel och består av; offertens löpnummer, försäljare, kundens namn, offertens avseende, ekonomisk information som totalbelopp och täckningsbidrag för produkter och inkomst av tjänst. Vidare lagras det datumet då offerten lämnades, hur länge den är giltig, om offerten är tagen eller tappad, länk till underlagsfilen och en länk till offertfilen. Registret uppdateras varje gång informationen om en offert ändras, t.ex. då den besvaras av kunden.

³ Skillnaden mellan Wordfilerna ".doc" och ".dot" är att då en .docfil som öppnas och redigeras ska sparas skrivs den ursprungliga .docfilen över. En .dot-fil öppnas som ett nytt dokument (doc) med de inställningar .dot-fil har. Då en .dot-fil sparas skrivs inte den ursprungliga filen över utan sparas som en .docfil. Det gör .dot-filen lämplig att använda som mall, då den inte riskerar att bli överskriven [4].

Alla filer tillhörande en offert lagras i en katalogstruktur på en av Optinets servrar i en katalog med kundens namn. I offertregistret är det dock bara den senaste giltiga versionen av offerten som syns. Har en offert reviderats finns inte länken till den gamla versionen kvar.

Löpnr.	Säljare	Kund	Avseende	Total belopp	TB	Tjänster	lämnad datum	går ut datum	tagen/tappad
731	prs		Driftsäker			122 250	2002-06-12	2002-06-22	tappad
732	prs		Virussäker			75 030	2002-05-30	2002-08-20	tappad
733	prs		Virussäker	49 740	49 740		2002-05-22	2002-06-15	tappad

Offertfil (insert, hyperlink)	Underlags fil (Ctrl+K)
OFFERTER\...doc	OFFERTER\...driftsäker 020514.xls
OFFERTER\...virussäker.doc	
OFFERTER\...20020522.doc	

Figur 8-2 Filregister för offerthantering.

8.1.1.3 Nykundsbearbetning

Nykundsbearbetning innebär att det första mötet med nya kunder eller potentiella kunder bokas och följs upp. Visma saknar helt en funktion för nykundsbearbetning. Det är en rutin skapad av Optinet för att följa upp första mötet med potentiella kunder och utfallet av dessa möten. För Optinet är det intressant att veta om mötet leder till att en offert skall skickas ut eller om kunden skall kontaktas vid ett senare tillfälle. Det är endast det första mötet som bokförs i det här registret för redovisning. Återkommande möten med kunden bokar säljarna i sina egna kalendrar i Microsoft Outlook (nedan kallat Outlook).

Informationen sparas i ett nykundsbearbetningsregister skapat i Excel (se figur 8-3, Appendix D för fullstorlek). Den information som lagras är kundnamn, försäljare, datumet då besöket bokades, datumet för besöket och datum för eventuell ombokning av besök. Vidare så lagras vad kunden är intresserad av, om offert skall skapas, om kunden har lagts till i Visma och om uppföljning skall göras.

Nykundsbearbetning 2002									
Kund	Säljare	bokat datum	besöksdatum	återb. datum	Intresserad av	Offert	I bearbetningsregister	Uppföljning	
	Prs	2002-08-23	2002-09-10	2004-03-09	Säkerhetsanalys	Ja	Ja	Ringa början mars	
	Prs	2002-09-04	2002-10-03		Drift		Ja		
	Tw	2002-09-11	2003-01-23		Allmänt	Nej	Ja		

Figur 8-3 Filregister för nykundsbearbetning.

8.1.1.4 Säljmöte

Cirka två gånger i månaden har Optinet säljmöte. Då samlas personalen för att diskutera verksamheten för den senaste perioden, d.v.s. sedan förra säljmötet. Med utgångspunkt från offertregistret diskuteras de offerter som skickats ut till kund samt de som har blivit besvarade. Genom Excels filtreringsfunktion filtreras offerterna per säljare och varje säljare redovisar sitt arbete.

På liknande sätt filtreras nykundsbehandlingsregistret och diskuteras. Här diskuteras utförda kundmöten och utgången av dem. Optinet vill veta om kunden är intresserad och i så fall av vilken tjänst eller produkt. Utifrån det bestäms om uppföljning skall ske av kunden och i så fall när.

När båda registren har gått igenom sammanräknas resultatet till ett uppföljningsregister (se figur 8-4, siffrorna är redigerade). Den information som lagras där är; datumet för säljmötet, summa försäljning av maskinvara/programvara, summa försäljning av tjänster, antal kunder under bearbetning (kunder som skall följas upp), täckningsbidrag och omsättning. Slutligen med utgångspunkt från förra periodens resultat så beräknas mål för försäljning av maskinvara/programvara samt tjänster.

	HW/SW	Tjänster	Kunder under bearbetning		Mål		HW	Oms	TB
20-jan	321	14	15		450	22	jan	650	122
03-feb	580	35			500	20	feb	425	102
17-feb	650	35			550	30	mar	185	35

Figur 8-4 Filregister för uppföljning.

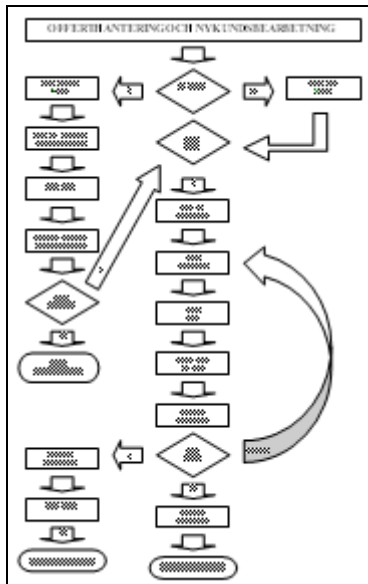
8.1.1.5 Optinets arbetsgång

Optinet har utvecklat en arbetsgång för att arbeta med offerter, kunder och uppföljning. Vi har valt att dela in den arbetsgången i två delar. En del med arbetsgången för nykundsbearbetning och offerthantering och en del med arbetsgången för uppföljning.

Arbetsgången vid kontakt med en ny potentiell kund (se Figur 8.5, Appendix E för fullstorlek) inleds med att kontrollera om kunden finns i Optinets register. Finns den inte det läggs kunden till i Visma och sedan i nykundsbehandlingsregistret. Säljaren har ett möte med kunden där de diskuterar vilka produkter eller tjänster kunden möjligtvis är intresserad av. Efter mötet uppdateras nykundsbehandlingsregistret. I de fall kunden inte är intresserad finns kunden fortfarande kvar i Visma. Vill kunden vid ett senare tillfälle göra affärer uppdateras nykundsbehandlingsregistret med en notering om när uppföljning skall göras och vad kunden är intresserad av. Om kunden vill inleda affärer redan vid mötet uppdateras nykundsbehandlingsregistret med att offert skall skapas och processen med att skapa en offert inleds.

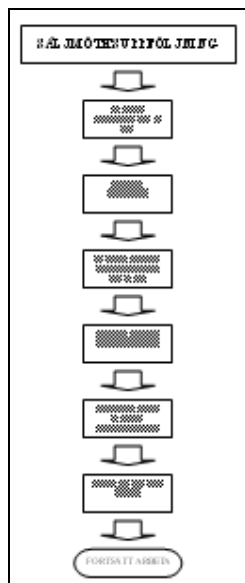
Då Optinet skall lämna en offert till en kund (se Figur 8.5, Appendix E för fullstorlek) kontrolleras det att nykundsbearbetning har skett samt att kunden finns i Visma och sedan skapas en post i offertregistret. Sedan inleds arbetet med att göra en underlagsfil för att undersöka vilka kostnader Optinet har för att leverera produkten eller utföra tjänsten, och vad kunden kommer att få betala. Utifrån underlagsfilen skapas en offert där relevant information från underlagsfilen klipps in och som sedan skickas till kund. Innan giltighetstiden gått ut svarar kunden förhoppningsvis på offerten. Svarar kunden

nej uppdateras endast offertregistret. Vill kunden revidera offerten skapas en ny underlagsfil och vidare en ny offertfil. Den nya offerten lämnas till kund och offertregistret uppdateras. Svarar kunden ja på offerten uppdateras offertregistret och tjänsten utförs.



Figur 8-5 Flödesschema över offerthantering och nykundsbearbetning.

Arbetsgången under säljmötet är enkel (se figur 8-6, Appendix F för fullstorlek). Optinets anställda samlas och går igenom resultatet av verksamheten sedan förra säljmötet. Offertregistret diskuteras post för post där varje säljare redovisar sin del och uppdatering av registret sker om behov finns. Efter offertregistret går nykundsbearbetningsregistret igenom på liknande sätt. Varje post diskuteras och säljarna redovisar sina poster. Även här görs uppdateringar om behov finns. Då genomgången av verksamheten är klar beräknas totalbeloppet fram, d.v.s. summan för försäljning av maskinvara/programvara, summan av täckningsbidraget och periodens omsättning samt antalet kunder under bearbetning. Dessa nyckeltal sammanställs till offertuppföljningsregistret och utifrån dem beräknas sedan mål inför nästa period. Målen avser försäljning av produkter och tjänster samt antal kunder under bearbetning och sammanställs till uppföljningsregistret.



Figur 8-6 Flödesschema över uppföljning vid säljmöte.

8.1.2 Nuvarande problem

Vad vi kan komma fram till under analysen av arbetssättet är att Optinets rutiner är utarbetade under en längre tid och fungerar bra. Det är verktygen Word och Excel samt de tre egenskapade registren som är otillräckliga för deras behov. Att arbeta med flera olika filregister medför en del problem. Det gör arbetet mer omständigt och det blir svårt att få en naturlig gång i arbetssättet. Det finns en risk att säljarna glömmet att uppdatera registren. Det innebär att informationen i registren kan bli inkonsistent och det finns inte något automatiskt sätt att upptäcka det på. Andra problem med att använda filer som register är att det är oöverskådligt och att åtkomligheten av dem är begränsad då endast en anställd åt gången kan använda dem samt endast inom företaget. En del kolumner i registren är också redundanta för att kunna identifiera varje post.

Vid hanteringen av reviderade offerter saknas en versionshantering och vid uppdatering av länkar i offertregistret skrivs de gamla länkarna över med de nya. Det bidrar till svårigheter med att söka fram en ordentlig kundhistorik, då säljarna dels måste leta i gamla offertregister och dels i den katalogstruktur där filerna lagras.

Under säljmötet används Excels filtreringsfunktion för att filtrera på bl.a. säljare. Vid varje filtrering sammanräknas den information som visas. Det är inte alltid sammanräkningen stämmer vilket ställer till problem och ödslar med tiden.

8.1.3 Optinets önskemål

Optinets önskemål är att effektivisera och standardisera hanteringen av information rörande offerter, nya kunder och uppföljning. De vill integrera de tre filregistrena; offertregistret, nykundsbearbetningsregistret och uppföljningsregistret till ett system.

Enkla och naturliga rutiner skall ligga till grund för deras arbetssätt och *Säljman* skall stödja dem fullt ut.

Då Visma använder SQL Server 2000 för att lagra information, anser Optinet att *Säljman* bör använda samma databaslösning. Optinet har rutiner för att ta backup av Vismas databas samt de anställda känner till hur SQL Server 2000 fungerar och är certifierade på den. Genom att använda en central databas erhålls fördelen med att flera anställda kan arbeta mot den samtidigt. Det är då också möjligt att införa kontroller för att minimera riskerna med bortglömda uppdateringar samt kontroller för att informationen i viss mån är korrekt. Det är tillåtet för det nya systemet att hämta information från Vismas databas, men inte att skriva till den. *Säljman* skall ha en egen databas och anledningen till det är att då supportpersonal från leverantören av Visma gör underhåll så installerar de ibland om databasen. Det skulle i så fall innebära att det *Säljmans* databas skulle försvinna.

För att underlätta att ta fram kundhistorik bör en sökfunktion finnas där ett flertal fält ur databasen skall kunna sökas på, bl.a. om en offert är tagen eller tappad, försäljare, datum, utförda eller kommande möten m.m. Versionshantering av alla filer tillhörande en offert bör också finnas så att en offerthistorik blir tillgänglig. Filerna bör också ha en enhetlig namnstandard och lagras centralt. Ett önskemål är att filerna skall skapas från gränssnittet genom att välja en mall och sedan kopieras till säljarens dator för bearbetning. Det skall då lagras i databasen att filen blivit skapad för att sedan kunna kontrollera att den har blivit upplagd efter bearbetningen och ge ett felmeddelande om så inte är fallet.

På Optinets säljmöten bör *Säljman* kunna erbjuda ett sökbart gränssnitt, där information om den gångna verksamheten lätt kan sökas fram och användas som ett diskussionsunderlag. Att kunna uppdatera informationen under mötets gång och visa korrekta sammanräkningar är också viktigt. Dessutom skall *Säljman* erbjuda en funktion för att beräkna verksamhetens nyckeltal samt lagra dessa och kommande mål till databasen. Dessa önskemål ligger senare till grund för kravspecifikationen (se kap. 8.2).

8.1.4 Val av lagrings- och utvecklingsmiljö

För att tillmötesgå Optinets önskemål kommer informationslagringen att göras i SQL Server 2000. Anledningen till det är att det är en programvara de har kunskap om och tillgång till. SQL Server 2000 möter också de krav Optinet har på åtkomligheten av informationen.

Vi har valt att implementera systemet som en webblösning, där utvecklingen sker i ramverket ASP.NET tillsammans med programspråket Visual Basic .NET. Optinet har som önskemål att *Säljman* inte bara skall vara tillgängligt inifrån företaget utan även utifrån hos t.ex. en kund. En webblösning gör det möjligt att använda *Säljman* om bara en Internetuppkoppling och en webbläsare finns tillgänglig. Ingen ytterligare programvara behöver då installeras. Optinet driver dessutom ett webbhotell som en del

av sin verksamhet. Den webbserverprogramvara som används där är IIS 5. Det vanligaste att använda tillsammans med IIS 5 är skriptspråket Active Server Pages (ASP) eller ramverket ASP.NET. Optinet har en del kunskaper i ASP, men har inga synpunkter på val av utvecklingspråk. Vi valde att använda ASP.NET för att det är en vidareutveckling av, och kommer på sikt att ersätta ASP. ASP.NET är mer kraftfullt och har en förbättrad prestanda gentemot ASP enligt *W3Schools* [5]. Till ASP.NET har nya kontroller för visning av information utvecklats med nya funktioner för uppdatering i realtid. Den förbättrade prestandan beror till stor del på att den kod som kompileras av .NET Framework lagras på webbservern så länge applikationen körs. Det ger minskade svarstider då t.ex. formulär återpostas till webbservern. I ASP.NET kan flera programspråk användas, bl.a. Visual Basic .NET, C# eller JScript.NET. Vi valde Visual Basic .NET därför att vi tidigare har kunskaper inom Visual Basic och VBScript.

Vid val av utvecklingsverktyg är inte urvalet stort. Vi valde Microsoft Visual Studio .NET (nedan kallat Visual Studio .NET) av anledningen att det är Microsoft, utvecklaren av ASP.NET, som har utvecklat applikationen samt att vi via Optinet har fri tillgång till den.

8.2 Kravspecifikation

Med utgångspunkt från nuvarande problem (se kap. 8.1.2) och Optinets önskemål (se kap. 8.1.3) skapades en kravspecifikation enligt Wisén m.fl. [6]. Den ger en detaljerad beskrivning av hur den slutgiltiga lösningen kommer att vara uppbyggd och vilka funktioner som skall implementeras. Kravspecifikationen presenteras här nedan i sin helhet (dock för rapportens skull något redigerad):

8.2.1 Bakgrund

Data- och systemsäkerhetsföretaget Optinet behöver ett nytt system för offerthantering och nykundsbearbetning samt uppföljning. Dels skall de kunna utföra hantering rörandes offerter och nykundsmöten oberoende av var de befinner sig, och dels skall det finnas en central lagringsplats där all information lagras. Vinsten med detta är att Optinet får en bättre kontroll på tagna, tappade och obesvarade offerter, belopp samt uppföljning av nykundsmöten. Dessutom skall de enkelt kunna ta fram information rörandes offerter och nya kunder vid sina säljmöten. Där skall det också finnas möjlighet till uppdatering under mötets gång. Nyckeltal skall kunna sammanställas för att se om de har nått sina uppsatta mål eller inte, vilket leder till bättre beslutsunderlag vid offertgivning.

8.2.2 Beskrivning av det nya systemet

Säljman skall ha som utgångspunkt att underlätta arbetet med nykundsbearbetning och offerthantering. Det ska ske genom en centralisering av informationslagringen till en databas för att komma ifrån de olika separerade filregister som finns idag. Under

rubriken funktioner specificeras de funktioner som företaget anser vara mest nödvändiga och de som anses vara önskvärda. Optinet kommer till viss del att fortsätta med sitt befintliga arbetssätt, såsom att använda sig utav underlagsfiler skapade i Excel och offertfiler skapade i Word. Den arbetsgången ger dem en viss flexibilitet eftersom en del varierande information återanvänds mellan offerterna. Den informationen kan kopieras och klistras in mellan dokumenten på ett flexibelt sätt. Då informationen som kan vara av intresse att återanvända, skiljer sig från offert till offert, skall den här funktionaliteten inte implementeras i det nya systemet. Kundrelaterad data såsom kundnamn, kundnummer och referens inhämtas från deras nuvarande system, Visma.

8.2.3 Funktioner

Här nedan följer en beskrivning över vilka nödvändiga funktioner systemet skall innehålla.

Nykundsbearbetning – Det skall vara möjligt att boka en tid för ett första möte med kunden, vad kunden är intresserad av etc. I efterhand skall uppdateringar kunna göras samt att minnesanteckningar och information om uppföljning skall kunna läggas till. Frekvens: ett par gånger i veckan.

Offertshantering – Erbjuder möjligheter att skapa och bearbeta offerter med tillhörande filer. Det skall vara möjligt att vid ändringar av en offerts status kunna uppdatera informationen om den. Offertfiler skall skapas automatiskt efter en vald mall dit sedan väsentlig information överförs. Den informationen är kundnamn, referens, datum och säljare. Varje offert skall ha möjlighet att knyta flera filer till sig. Frekvens: dagligen.

Versionshantering – Versionshantering för de filer som är knutna till en offert skall finnas. Vid revidering av en offert skall versionsnumret uppdateras för att göra det möjligt att gå tillbaka i filhistoriken, och se vad som har ändrats sedan förra versionen. Frekvens: vid skapande/revidering av fil.

Sökning - En av de viktigare funktionerna är möjligheten till att göra sökning av relevant information angående nykundsmöten och offerter. Nykundsmöten skall kunna visas baserat på säljare, datum och om uppföljning skall ske. Visning av offerter skall kunna ske beroende på om de är tagna, tappade eller obesvarade. Sökning av nykundsmöten och offerter skall också kunna ske med kundnamn, säljare och tidsperiod. Frekvens: (sökning) 2-3 gånger per vecka, (säljmöten) två gånger per månad.

Utöver dessa nödvändiga funktioner, är följande funktioner önskvärda:

Inloggning – Identifiering vid uppstart av systemet, så att användarens namn under efterföljande moment är förvalt.

Begynnelsebokstäver – Vid sökning av specifik kund skall begynnelsebokstäverna i ett kundnamn kunna skrivas in för att visa möjliga korresponderande kundnamn.

Komprimering – För att spara utrymme i databasen bör *Säljman* erbjuda komprimering av filer.

Hårddiskutrymme – Visning av totalt använt diskutrymme för databasen.

Namngivning – Vid skapande av offertfiler skall automatisk namngivning av dem ske för att minimera risken för namnkonflikter.

Arkivering – För att kunna begränsa databasens storlek bör arkivering av offertfiler vara möjlig. Ett senaste datum skall då kunna anges för att arkivera alla filer tillhörande de offerter som har besvarats. Frekvens: vanligtvis en gång per år.

8.2.4 Systemegenskaper

Arbetet med att underhålla systemet, såsom att göra säkerhetskopiering, skall hållas till ett minimum. Varje säljare är ansvarig för att korrekta och relevanta filer laddas upp till databasen och att ny informationen uppdateras. En kontinuerlig backup av systemet samt eventuell årlig arkivering av äldre offerter borgar för att övrig administration kan undvikas. *Säljman* kommer att ligga inom Optinets intranät och kommer inte att ha någon form av autentisering.

För att vid behov kunna utveckla *Säljman*, då t.ex. arbetssätt och behov förändras, skall systemet vara utvecklat på ett sådant sätt att förändringar av systemet underlättas. Detta skall ske med bl.a. god systemdokumentation, en väl följd namnstandard, och ett programspråk lämpat för uppgiften, som finns kvar inom en överskådlig framtid.

8.2.5 Funktionsegenskaper

Här nedan följer en beskrivning av de egenskaper ovanstående funktioner skall ha:

Offertshantering – Det skall inte implementeras någon gräns på hur många offerter som skall kunna lagras för varje kund. *Säljman* skall vara heller inte ha någon maxgräns för antalet filer som kan knytas till en offert. En naturlig begränsning finns i antalet filer som skapas per offert och att de skapade filerna är väsentliga. Detsamma gäller för storleken av offertfilerna där inga begränsningar implementeras.

Versionshantering – Ingen begränsning kommer att finnas på antalet versioner av en offertfil.

Sökning – Två typer av sökningar skall finnas; en för framtagande av offertinformation och en för framtagande av information rörande nykundsmöten. Användaren skall kunna söka offerter enligt dessa kriterier; tagen/tappad offert, obesvarad offert, säljare och tidsperiod. Sökning av kund skall kunna ske enligt följande; säljare, bokad av, bokad för, uppföljning och tidsperiod. Dessutom skall sökkriterierna kunna kombineras.

Arkivering – Det skall vara möjligt att arkivera alla filer t.o.m. ett visst datum. Med arkivering menas att de flyttas ut ur databasen och lagras till en katalog på servern.

Dock skall informationen om offerterna finnas kvar för att kunna erbjuda en komplett kundhistorik.

8.2.6 Dokumentation

En systemdokumentation skall skapas vilket beskriver den tekniska lösningen för att underlätta underhåll och framtida modifieringar av systemet. I systemdokumentationen skall det ingå en installationsanvisning samt en beskrivning av Säljmans alla filer samt dess funktioner och subrutiner.

8.2.7 Utbildning

Ingen långvarig utbildning skall krävas för att underhålla, installera, eller använda systemet. En viss datorvana och kunskap förutses att Optinets användare.

8.3 Design

Utifrån analysen av Optinets arbetssätt, behov och kravspecifikation presenteras nu en lösning av hur *Säljman* kommer att se ut. Vi har tagit tillvara på de önskemål och de förutsättningar Optinet har i arbetet med att designa *Säljman*. Designen delas in i två delar; dels strukturen på informationslagringen och dels hur gränssnittet kommer att utformas.

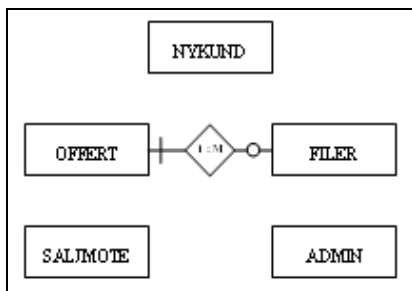
8.3.1 Informationslagring

Vid design av databasstrukturen har vi utgått från den information Optinet tidigare använt sig av för att utföra sin verksamhet rörande offerthantering, nykundsbearbetning och uppföljning. Deras arbetssätt har varit utarbetat under en längre tid och den informationen de lagrar i de tre filregistren kommer förutom med några få undantag att användas i det nya systemet. Initialt skissade vi databasens struktur på en *whiteboard* för att ta fram tabeller, relationer och fält. Med hjälp av standardmässiga normaliseringsregler strukturerades entiteternas relationer och tabellernas fält. Strukturen bearbetades och normaliserades sedan ytterligare av projektgruppen och referensgruppen i en iterativ process. Till sist överfördes strukturen till papper och diskuterades med Optinet.

8.3.1.1 Databasstruktur

Vi har valt att dela in databasen i fem tabeller; *nykund*, *offert*, *filer*, *saljmote* och *admin*. Tabellerna visas i ett *entitets/rerelationsdiagram* över databasen (se figur 8-7, Appendix G för fullstorlek). Tre av tabellerna; *nykund*, *saljmote* och *admin* är helt fristående medan de två andra tabellerna; *offert* och *filer* har en relation mellan sig. Det innebär att varje enskild fil som lagras i databasen är kopplad till en specifik offert. Tabellen *nykund* lagrar all information relaterad till kundens första möte. Varje kund kan bara ha en post i den här tabellen. Tabellen *offert* hanterar all information rörande offerter och

varje kund kan ha många offerter. Varje offert har ett löpnummer som gör den unik som också används i relationen till tabellen *filer* för att koppla offertens alla filer till sig. I tabellen *saljmote* lagras verksamhetens nyckeltal och mål och består av en post för varje säljmöte Optinet har. *Admin* innehåller den administrativa informationen som har betydelse för systemets funktionalitet, som sökvägar för arkivering och till offertmallar.



Figur 8-7 Entitets/rerelationsdiagram över databasens struktur.

För att mer i detalj visa de enskilda tabellernas uppbyggnad har vi skapat en objekt/termkatalog. Den finns i sin helhet i Appendix H, och beskriven med figurer för varje tabell i kapitel 8.3.1.2 – 8.3.1.6.

8.3.1.2 Tabellen *nykund*

Tabellen *nykund* (se figur 8-8) innehåller 12 fält. Primärnyckeln är *id_nykund* och är automatiskt uppräknande för varje ny post som skapas. Vid skapande av en post i tabellen matas följande information in; kundnummer, besöksdatum, bokningsdatum, vem som bokade, vem som skall gå på mötet och vad mötet avser. Kundnummer identifierar unikt varje kund och vi har medvetet valt att inte lagra kundens namn i tabellen då det inte är ovanligt att en kund byter namn vid t.ex. ett uppköp eller en sammanslagning. Kundnumret ändras inte och förblir alltid samma. Då mötet har utförts uppdateras information med en minnesanteckning, om kunden skall följas upp och i så fall när uppföljningen skall göras. Det sista fältet *saljmote* anger om kunden i fortsättningen skall visas under säljmötet för diskussion. Varje post i databasen har en storlek på upp till 383 byte.

Nyckel	Fältnamn	Datatyp	Storlek	Ej null	Beskrivning
Primär	id_nykund	Int	4	X	Autouppräknande primärnyckel
	kundnr	Int	4	X	Kundnummer
	besok	Smalldatetime	4		Datum för besöket
	bokat	Smalldatetime	4		Datum då besöket bokades
	bokat_av	Varchar	30		Namn på den anställda som bokade besöket
	bokat_for	Varchar	30		Namn på den anställda mötet är avsett för
	avseende	Varchar	100		Produkt/tjänst kunden är intresserad av
	minnesant	Varchar	100		Anteckning om mötet
	uppfoljning	bit	1		Lagrar om uppföljning ska ske
	upp_nar	Varchar	5		Vilket kvartal ska kunden följas upp, Q1-Q4 eller ingen uppföljning?
	uppf_text	Varchar	100		Vad gäller uppföljningen?
	saljmote	bit	1		Ska post visas på företagets säljmöte?

Figur 8-8 Specifikation av tabellen *nykund*.

8.3.1.3 Tabellen *offert*

I tabellen *offert* (se figur 8-9) finns det 14 fält där *id_lopnr* är automatuppräknande primärnyckel. Löpnumret är unikt för varje offert och varje kund kan ha flera offerter. Då en post skapas matas kundnummer, offertens avseende, kommentar, försäljare och kundens referens in i databasen. Då offerten skapats och skickats till kund uppdateras databasen med datumet då offerten skickades och offertens giltighetsdatum. Senare då kunden besvarar offerten uppdateras databasen med om offerten är tagen eller tappad och vilket datum den besvarades. Är offerten tagen matas dessutom offertens ekonomiska siffror som totalbelopp, täckningsbidrag och tjänst in. Varje inmatad post i databasen har en storlek på upp till 294 byte

Nyckel	Fältnamn	Datatyp	Storlek	Ej null	Beskrivning
Primär	id_lopnr	Int	4	X	Autouppräknande primärnyckel
	kundnr	Int	4	X	Kundnummer
	avseende	Varchar	100		Vilken produkt/tjänst avser offerten?
	kommentar	Varchar	100		Kommentar om offerten
	saljare	Varchar	30		Vem är ansvarig säljare?
	referens	Varchar	30		Företagets referens, hämtas från Visma
	lamnad	Smalldatetime	4		Datum då offerten lämnas till kund
	giltig	Smalldatetime	4		Slutdatum för offertens giltighetstid
	tagen	Bit	1		Är offerten tagen?
	tappad	Bit	1		Är offerten tappad?
	besvarad	Smalldatetime	4		Datumet offerten besvarades
	totalbelopp	Int	4		Totalsumma för försäljning
	tb	Int	4		Täckningsbidrag, försäljningens vinst
	tjanst	Int	4		Löpande månadsinkomst

Figur 8-9 Specifikation av tabellen *offert*.

8.3.1.4 Tabellen filer

Tabellen *filer* (se figur 8-10) innehåller 15 fält och har *id_fil* som autouppräknande primärnyckel. För att skapa en relation till tabellen offert används dess primärnyckel (*id_lopnr*) här som främmande nyckel (*fid_lopnr*). Då en fil skapas lagras filens benämning, datumet filen skapades, om filen är en offertfil, underlagsfil, avtalsfil eller en övrig fil, filens version, filens namn, contenttyp och filens längd till databasen. Fältet arkiverad används senare för att markera om filen har kopierats ut ur databasen. Varje skapad post i databasen har en storlek på upp till 100 224 byte om en medelstorlek för en fil antas vara 100 000 byte.

Nyckel	Fältnamn	Datotyp	Storlek	Ej null	Beskrivning
Primär	id_fil	Int	4	X	Autouppräknande primärnyckel
Främmande	fid_lopnr	Int	4	X	Främmande nyckel från Offert tabellen
	benamning	Varchar	50		Filens avseende
	skapad	Smalldatetime	4		Datumet då filen skapades
	underlag	Bit	1		Anger att filen är ett underlag
	offert	Bit	1		Anger att filen är en offert
	avtal	Bit	1		Anger att filen är ett avtal
	ovrig	Bit	1		Anger om filen inte är någon av de tre ovanstående
	version	Smallint	2		Filens version
	fil_uppladdad	Bit	1		Markeras automatiskt då fil skapas från webbgränssnitt
	filnamn	Varchar	100		Namnet på filen
	arkiverad	Bit	1		Anger om filen är arkiverad, filnamn finns kvar för att kunna hitta filen
	fil	Image	~100 000		Medelstorlek på uppladdad fil
	contenttyp	Varchar	50		Filens filtyp
	fillangd	Int	4		Filens storlek

Figur 8-10 Specifikation av tabellen *filer*.

8.3.1.5 Tabellen saljmote

Tabellen *saljmote* (se figur 8-11) innehåller 16 fält och har *id_saljmote* som autouppräknande primärnyckel. I den här tabellen lagras datumet för säljmötet, verksamhetens nyckeltal samt mål. Nyckeltalen som beräknas utifrån den gångna periodens verksamhet är; summa totalbelopp, summa täckningsbidrag summa tjänst, antal tagna offerter, tappade offerter, lämnade offerter, antal genomförda möten, antal kunder som skall följas upp och antal bokade kommande möten. Dessutom lagrar tabellen de mål Optinet sätter upp inför nästkommande period. Målen är; summa totalbelopp, summa täckningsbidrag, summa tjänst samt antal bokade kommande besök. Varje skapad post i tabellen har en storlek på 64 byte.

Nyckel	Fältnamn	Datotyp	Storlek	Ej null	Beskrivning
Primär	id_saljmote	Int	4	X	Autouppräknande primärnyckel
	fid_nykund	Int	4	X	Senaste värdet från tabellen nykund
	fid_offert	Int	4	X	Senaste värdet från tabellen offert.
	mote	Smalldatetime	4	X	Säljmötets datum
	sum_totalbelopp	Int	4	X	Summa försäljning
	sum_tb	Int	4	X	Summa täckningsbidrag
	sum_tjanst	Int	4	X	Summa tjänst
	ant_tagna	Int	4	X	Antal tagna offerter
	ant_tappade	Int	4	X	Antal tappade offerter
	ant_lamnade	Int	4	X	Antal lämnade offerter
	ant_gen_moten	Int	4	X	Antal genomförda möten
	ant_upp_moten	Int	4	X	Antal kunder för uppföljning
	ant_kom_moten	Int	4	X	Antal kommande möten
	ant_bok_moten	Int	4	X	Antal bokade möten.
	mal_totalbelopp	Int	4		Mål för totalbelopp
	mal_tb	Int	4		Mål för täckningsbidrag

Figur 8-11 Specifikation av tabellen *saljmote*.

8.3.1.6 Tabellen *admin*

Tabellen *admin* (se figur 8-12) består av fem fält. Det kommer bara att finnas en post i den och därför sätts primärnyckeln *id_admin* till 1. Anledningen till det är att hämtning och lagring till databasen förenklas då man på förhand känner till id-värdet i tabellen. De övriga fyrafälten lagrar sökvägar. Tre sökvägar till vart underlagsfiler, offertfiler och avtalsfiler finns lagrade i serverns katalogstruktur. Den sista sökvägen är till vart arkiverade filer sparas vid kopiering ut ur databasen. Tabellens storlek uppgår till max 404 byte.

Nyckel	Fältnamn	Datotyp	Storlek	Ej null	Beskrivning
Primär	id_admin	Int	4	X	Primärnyckel, satt till 1
	sokvag_underlag	Varchar	100		Vart underlagsmallar finns
	sokvag_offert	Varchar	100		Vart offertmallar finns
	sokvag_avtal	Varchar	100		Vart avtalsmallar finns
	sokvag_arkivering	Varchar	100		Vart filer ska arkiveras

Figur 8-12 Specifikation av tabellen *admin*.

8.3.1.7 Databasens storlek

För att få en uppfattning om databasens storlek efter ett års användande har vi gjort en del antaganden om Optinets verksamhet, inräknat en viss marginal (se figur 8-13). Vi antar att de får 100 nya kunder och skapar 200 offerter per år. Varje offert innehåller i sin tur fem filer. Vi räknar också med att de har säljmöte två gånger i månaden. Viktigt att komma ihåg är attfälten av typen *varchar* och *image* inte uppgår till sin maximala storlek utan endast till storleken av texten eller filen som är inmatad. Fälten av typ *int*, *smalldatetime* och *bit* är alltid av samma längd oavsett innehåll.

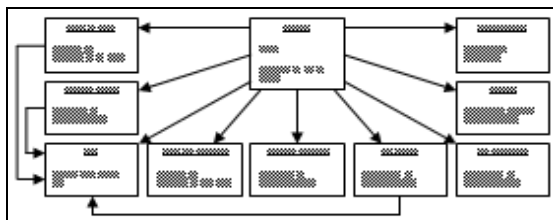
Databasens storlek kommer maximalt under det första året att uppgå till 100 323 040 bytes, där 99.7 % består av filer.

Tabellnamn	Antal poster (st)	Poststorlek (byte)	Tabellstorlek (byte)	Kommentar
Nykund	100	383	38 300	Antal nya kunder
Offert	200	294	58 800	Antal nya offerter
Filer	1 000	100 224	100 224 000	Fem filer per offert gånger antal offerter
Saljmate	24	64	1 536	Antal säljmöten
Admin	1	404	404	Administrativ information
Summa			100 323 040	

Figur 8-13 Databasens storlek.

8.3.2 Gränssnitt och funktionalitet

Innan arbetet med att realisera gränssnittet startade skapade vi en skiss på papper över hur navigeringen skulle kunna ske inom gränssnittet. Navigeringen (se figur 8-14, Appendix I för fullstorlek) delades upp utifrån analysen av de tre verksamhetsområden inom Optinet som berör projektet; offerthantering, nykundsbearbetning och uppföljning. Det är en för företaget naturlig indelning och vi kom därför fram till att *Säljman* skulle bestå av följande delar; en del för offerthantering, en del för nykundsbearbetning och en del för uppföljning. Förutom de sidorna finns även en gemensam meny, en startsida samt en sida för administrering av *Säljman*.



Figur 8-14 Skiss över navigeringen inom *Säljman*.

Förslaget diskuterades tillsammans med Optinet och ett beslut togs på att fortsätta med designen av *Säljman* i Visual Studio .NET. Vi skrev ingen kod utan använde det grafiska verktyget i Visual Studio .NET för att placera ut komponenter. Vi har använt oss av absolut positionering för att ha större möjligheter att styra utseendet på formulären. Det sparar tid och arbete gentemot att styra layouten med hjälp av tabeller. Efter arbetet med att skapa gränssnittet fick Optinets användare testa och gå igenom vårt förslag. Det framkom inga större synpunkter utan projektet fortskred.

Designen av gränssnittet och den funktionaliteten som kommer att finnas på varje sida (se kap. 8.3.2.1 – 8.3.2.10) är beroende av varandra och därför tas de bitarna upp här gemensamt.

8.3.2.1 *Meny och Startside*

Den första sidan säljaren möts av erbjuder en meny i den vänstra ramen och en välkomstsida i den mittre (se figur 8-15). Menyn finns hela tiden åtkomlig då säljaren använder *Säljman*. Via den ges han åtkomst till alla de funktioner han behöver.

På förstasidan har användaren möjlighet att välja sitt namn i en rullgardinsmeny. Det skall inte vara något krav för säljaren att logga in, men gör han det visas endast hans egen information vid bearbetning av offerter och nykundsmöten. Loggar användaren in visas namnet under menyn till vänster som inloggad säljare.

Startside

Skapa Ny Offert

Bearbeta Offerter

Offertfiler

Boka Nytt Kundmöte

Bearbeta Kundmöten

Sök Offerter

Sök Kundmöten

Säljmöte

Administrering

Inloggad som:

Logga in:

Figur 8-15 Gränssnitt för *Menyn* och *Startsidan*.

8.3.2.2 *Skapa ny offert*

På sidan *Skapa ny offert* finns ett formulär där säljaren skapar en ny offert (se figur 8-16). I en rullgardinsmeny skall alla Optinets kunder finnas tillgängliga för val vid skapande av offert. Det innebär en stor mängd kunder så två genvägar skall finnas för att snabbare kunna välja rätt kundnamn; en ruta där säljaren matar in begynnelsebokstäverna i kundens namn och en där inmatning av kundnummer kan ske. Endast den eller de kunder som matchar inmatningen skall då visas i rullgardinsmenyn. För att kunna skapa en ny offert för en kund är det ett krav att kunden finns med i Vismas databas. Finns den inte det måste den först läggas till där. Då säljaren väljer en kund skall kundens referens automatiskt hämtas från Visma och visas i formuläret. Säljaren väljer sedan sitt eget namn ur en rullgardinsmeny. Namnet är förvalt om inloggning skett på förstasidan. Offertens avseende och en kommentar fylls i och

knappen *Skapa offert* trycks in för att lagra informationen till databasen. Säljaren skall även ha möjlighet att härifrån navigera till sidan *Offertfiler* för att skapa offertfiler.

Början på kundnamn: Kundnummer: Kundnamn: *

Referens: * Säljare: *

Avseende: * Kommentar:

Figur 8-16 Gränssnitt för sidan *Skapa ny offert*.

8.3.2.3 Bearbeta offerter

Bearbeta offerter är tänkt att användas i säljarnas dagliga arbete med offerter. Via den här sidan kan arbete ske med alla i *Säljmans* skapade offerter (se figur 8-17). Säljaren har möjlighet att direkt välja den kund som skall arbetas med antingen via kundens begynnelsebokstäver, via kundnummer eller via rullgardinsmenyn där alla kunder som har en offert inmatad i *Säljman* visas. Det skall också gå att välja en specifik offert genom att mata in offertens löpnummer. Sökkriterier för att hämta flera offerter för redigering skall också finnas. Öppna offerter kan visas enligt tre kriterier; de som har lämnats till kund, de som ännu inte har lämnats till kund och de offerter där giltighetstiden gått ut men som ännu inte har besvarats. Ytterligare skall besvarade offerter kunna sökas på enligt kriterierna tagna eller tappade. Alla offerter skall också kunna sökas på. Har säljaren valt att logga in på startsidan visas endast den säljarens offerter.

Början av kundnamn: Kundnummer: Kundnamn:

Löpnummer:

Öppna Tagna
 Lämnade Tappade
 Ej lämnade
 Utgångna Alla

Figur 8-17 Gränssnitt för sidan *Bearbeta offert*.

8.3.2.4 Offertfiler

För att komma till sidan *Offertfiler* (se figur 8-18) finns det tre vägar. Säljaren kan välja att navigera hit via menyn i vänsterkanten. I det fallet måste löpnumret matas in på den offert vars filer skall bearbetas. Säljaren kan också komma hit via en knapp på sidan *Skapa offert* som visas då en offert precis har skapats. Det tredje sättet är att via sidan

Bearbeta offerter klicka på knappen *Filer* vid listningen av varje offert. Då användaren klickar sig till sidan *Offertfiler* enligt de två senare metoderna följer offertens löpnummer med till sidan och vald offerts filer visas direkt.

Förutom att visa filer erbjuds också möjligheten att skapa underlagsfiler, offertfiler och avtalsfiler eller att ladda upp övriga filer. De filerna knyts till den offert som för tillfället visas och filerna laddas ner till säljarens dator för bearbetning. Då filerna är klara laddas de upp till databasen via sidan *offertfiler*. Vid skapande av filer genereras automatiskt ett filnamn för att lätt kunna särskilja vilken typ av fil och vilken version det är. Vid uppladdning till databasen sker då också en kontroll på att rätt fil lagras till rätt post.

Löpnummer:

Välj underlagsmall:

Välj offertmall:

Välj avtalsmall:

Figur 8-18 Gränssnitt för sidan *Offertfiler*.

8.3.2.5 *Boka nytt kundmöte*

Via sidan *Boka nytt kundmöte* (se figur 8-19) sker bokningar med nya kunders första möte. Kundens namn väljs enligt samma princip som vid *Skapa offert* och *Bearbeta offert*. Användaren matar in mötets avseende, vem som bokat det och vem som skall gå på mötet. Har säljaren loggat in på startsidan visas säljarens namn i båda rullgardinsmenyerna. Vidare matas bokningsdatum och mötesdatum in via två kalenderkontroller vilka är nyutvecklade för ASP.NET.

Början på kundnamn:	Kundnummer:	Kundnamn: *
<input type="text"/>	<input type="text"/>	<input type="text"/>
Avseende: *		
<input type="text"/>		
Bokat av: *		Bokat för: *
<input type="text"/>		<input type="text"/>
Bokningsdatum: *	Mötesdatum: *	
< juli 2003 >	< juli 2003 >	
må ti on to fr lö sö	må ti on to fr lö sö	
30 1 2 3 4 5 6	30 1 2 3 4 5 6	
7 8 9 10 11 12 13	7 8 9 10 11 12 13	
14 15 16 17 18 19 20	14 15 16 17 18 19 20	
21 22 23 24 25 26 27	21 22 23 24 25 26 27	
28 29 30 31 1 2 3	28 29 30 31 1 2 3	
4 5 6 7 8 9 10	4 5 6 7 8 9 10	

Figur 8-19 Gränssnitt för sidan *Boka nytt kundmöte*.

8.3.2.6 Bearbeta kundmöten

Som motsvarighet till *Bearbeta offerter* är även sidan *Bearbeta kundmöten* tänkt att användas till säljarnas dagliga arbete med nykundsmöten (se figur 8-20). Har säljaren valt att logga in på startsidan visas endast de möten som är kopplade till den inloggade säljaren. I formuläret väljs kundens namn enligt tidigare beskriven princip med begynnelsebokstäver. Här finns också möjlighet att välja att visa möten enligt sökkriterierna; utförda möten, kommande möten eller om uppföljning skall ske och i så fall när. Optinet har valt att dela in när uppföljning skall ske kvartalsvis. De val som finns är då Q1, Q2, Q3, Q4 eller alla kunder som är markerade för uppföljning.

The screenshot shows a web form for editing a customer meeting. It includes three input fields at the top: 'Början på kundnamn:', 'Kundnummer:', and 'Kundnamn:'. Below these are two checkboxes, 'Kommande' and 'Utförda', and a dropdown menu for 'Uppföljning:'. A 'Visa Poster' button is located at the bottom right.

Figur 8-20 Gränssnitt för sidan *Bearbeta kundmöte*.

8.3.2.7 Sök offerter

Söksidan för offerter (se figur 8-21) och även för kundmöten (beskrivs i kapitel 8.3.2.8) är inte tänkt att användas under det dagliga arbetet med offerter eller möten. Här kan Optinets anställda söka offerter enligt vissa kriterier och inloggningen på startsidan används inte på dessa sidor. Huvudsyftet med sidorna är att använda dem vid Optinets säljmöten för framtagande av diskussionsunderlag.

Om kundhistorik skall visas kan en specifik kund väljas ur rullgardinsmenyn eller med hjälp av begynnelsebokstäverna. Den vanligaste sökningen är dock efter flera offerter baserat på samma sökkriterier som *Bearbeta offert* med avseende på öppna och besvarade offerter. Det är också viktigt att kunna söka på en viss säljares offerter eller att tidsbestämma sökningen för att inte visa alltför stor mängd information då *Säljman* körts under en längre tid.

The screenshot shows a search form for offers. It features input fields for 'Början på kundnamn:', 'Kundnamn:', 'Säljare:', 'Startdatum (ÅÅÅÅMMDD):', and 'Slutdatum (ÅÅÅÅMMDD):'. There are also four checkboxes: 'Öppna', 'Tagna', 'Tappade', and 'Alla'. A 'Sök Offerter' button is positioned at the bottom right.

Figur 8-21 Gränssnitt för sidan *Sök offerter*.

8.3.2.8 Sök kundmöten

Söksidan för kundmöten (se figur 8-22) har samma funktion som söksidan för offerter, att vara ett diskussionsunderlag eller en söksida för Optinets personal. Kundnamn kan inte väljas specifikt här då kundhistoriken för nykundsmöten endast uppgår till ett möte. Snarare vill de kunna söka fram kunder där uppföljning skall ske. De sökkriterier som är möjliga att specificera är vem som bokade mötet och vem som har utfört det. Även här är det viktigt att kunna ange datum för sökningen för att ta fram aktuell information. Det viktigaste sökkriteriet är när uppföljning skall göras. Kunden visas då endast om den är

markerad för uppföljning. Vid inmatning av ny kund till databasen markeras ett fält i databasen för diskussion på säljmöte. Det fältet har säljaren inte möjlighet att avmarkera från sidan *Bearbeta nykundsmöte*. Avsikten med det är att alla nykundsmöten skall tas upp för diskussion på säljmötet där det bestäms om fältet skall avmarkeras eller finnas kvar för fortsatt diskussion.

Bokat av: Bokat för:

Startdatum (ÅÅÅÅMMDD): Slutdatum (ÅÅÅÅMMDD):

Uppföljning:

Figur 8-22 Gränssnitt för sidan *Sök kundmöten*.

8.3.2.9 *Säljmöte*

Sidan *Säljmöte* (se figur 8-23) har en enkel design och används endast under företagets säljmöten. Då knappen *Beräkna* klickas på sker en sammanräkning av verksamhetens nyckeltal (se kap. 8.3.1.5). Nyckeltalen beräknas utifrån förra säljmötet och dagens datum.

För offerter beräknas summa av totalbelopp, täckningsbidrag och tjänst utifrån datumet då de besvarades. Det gör även antal tagna och tappade offerter. Antal lämnade offerter beräknas utifrån det datum offerten lämnades. För nykundsmöten beräknas antal genomförda möten, antal bokade möten och antal kommande möten i en jämförelse med bokningsdatum. Dessutom beräknas alla kunder under bearbetning, d.v.s. de som är markerade för uppföljning.

Sammanräkningen visas på skärmen tillsammans med möjlighet att mata in mål för den kommande perioden. Målen är summa totalbelopp, summa täckningsbidrag, summa tjänst och antal bokade kommande möten. På skärmen skall även en säljmötes historik visas.

Figur 8-23 Gränssnitt för sidan *Säljmöte*.

8.3.2.10 *Administrering*

Sidan *Administrering* är skapad för att förenkla administreringen och underhållet av *Säljman* (se figur 8-24). Två funktioner kommer att finnas; arkivering och sökvägar. Då

Säljman använts under en längre tid kommer databasens storlek att ha ökat. Den största delen av informationen är filer, och arkiveringsfunktionen ger en möjlighet att kopiera filerna ut ur databasen och markera dem som arkiverade. Filerna är sedan inte åtkomliga via databasen, men däremot finns informationen om dem kvar för att erbjuda kundhistorik. Den andra funktionen ställer in sökvägar till den katalog där underlagsmallar, offertmallar och avtalsmall finns samt även dit arkivering skall ske.

Arkivering

Slutdatum för arkivering:

Sökvägar

Underlagsmall: <input type="text"/>	<input type="button" value="Bläddra"/>	Offertmall: <input type="text"/>	<input type="button" value="Bläddra"/>
Avtalsmall: <input type="text"/>	<input type="button" value="Bläddra"/>	Arkivering: <input type="text"/>	<input type="button" value="Bläddra"/>

Figur 8-24 Gränssnitt för sidan *Administrering*.

8.3.3 Namngivning

Efter leveransen av *Säljman* kommer Optinets personal själva att underhålla systemet. Det är därför viktigt att programmeringen sker på ett sådant sätt att programkoden blir läsbar i efterhand

Innan realiseringen av *Säljman* startade skapade vi en namnstandard (se Appendix J) på hur programkoden skall se ut. Den styr vilket språk som skall användas och hur namngivningen av variabler, instanser, komponenter, subrutiner, funktioner och HTML-taggar skall gå till.

8.4 Realisering

Efter avslutad designfas inleds realiseringen av *Säljman*. Det har varit den stora biten i projektet och vi har även här valt att dela in beskrivningen i två delar; databasen samt gränssnitt och funktionalitet. De problem som uppstått redovisas under respektive kapitel.

8.4.1 Databasen

När arbetet med att realisera databasen påbörjades var vi tvungna att bestämma tillvägagångssättet. Då realiseringen av *Säljman* inte gjordes direkt på Optinets servrar var vi medvetna om att databasen var tvungen att både installeras på arbetsdatorn och på Optinets SQL Server. Vi såg tre möjligheter till hur implementeringen av databasen skulle gå till; att manuellt skapa databasen och tabellerna både på arbetsdatorn och på Optinets Server, att manuellt skapa databasen och tabellerna på arbetsdatorn och sedan exportera databasen eller att manuellt skapa databasen i SQL Server 2000 och köra ett SQL-skript där tabellerna skapas med hjälp av SQL (Structured Query Language)⁴.

Vi valde det sista alternativet, att manuellt skapa en databas och sedan skapa tabellerna med hjälp av ett SQL-skript. Det är ett mycket snabbt sätt att skapa de tabeller *Säljman* behöver använda sig av och skriptet kan användas för både nyinstallation och ominstallation av tabellerna. Viktigt att tänka på är dock att inte köra skriptet utan att först ha gjort en backup på lagrad information, annars kommer den att försvinna då nya tabeller skapas.

Till vår hjälp för att skapa SQL-skriptet (se figur 8-25, Appendix K för komplett SQL-skript) använde vi oss av boken *SQL på 10 minuter* [7] för SQL-syntaxen. Skriptet består av tre sektioner; ta bort data ur befintliga tabeller, ta bort tabeller och skapa tabeller. Den information som anges i SQL-kommandot vid skapande av tabeller är fältets namn, vilken datatyp det är och om värdet får vara *null*, d.v.s. inte innehålla något värde alls, eller *not null*. Vi anger vilket fält som skall vara primärnyckel och om det fältet skall vara automatiskt uppräknande.

```
DELETE FROM Nykund

DROP TABLE Nykund

CREATE TABLE Nykund (
  id_nykund      int          NOT NULL      IDENTITY(1,1) PRIMARY KEY,
  kundnr        int          NOT NULL ,
  besok         smalldatetime NULL ,
  bokat         smalldatetime NULL ,
  bokat_av      varchar (30)  NULL ,
  bokat_for     varchar (30)  NULL ,
  avseende     varchar (100)  NULL ,
  minnesant    varchar (100)  NULL ,
  uppf_nar     varchar (5)    NULL ,
  uppf_text    varchar (100)  NULL ,
  uppfoljning  bit           NULL ,
  saljmote     bit           NULL )
```

Figur 8-25 Utdrag ur SQL-skriptet.

⁴ SQL (Structured Query Language) är ett språk för att kommunicera med databaser där det är möjligt att hämta information från databasen och att lagra information till databasen [7].

Då vi bestämde vilka datatyper som skulle användas tog vi hjälp från webbplatsen *MSSQL* [8] och SQL Server 2000 inbyggda hjälpfunktion. De datatyper tabellerna innehåller är *int* eller *smallint* för heltal, *varchar* för text, *smalldatetime* för datum, *bit* för booleska värden och *image* för filer. Att vi valde *int* som datatyp till de flesta heltalsfält beror på att de kan lagra tal upp till en storlek av 2 147 483 647. *Smallint* kan endast lagra upp till 32767 bytes vilket i många fall kan vara begränsande. Datatypen *varchar* använde vi för lagring av text då den inte tar upp en fast storlek i databasen utan endast mängden inmatade tecken. För lagring av filer fanns inget annat alternativ än *image*, den kan lagra filer med en storlek på upp till 2 147 483 647 bytes. Då andra alternativ är begränsade i mängden data som kan lagras, exempelvis *binary* som endast kan lagra upp till 8000 bytes, var de ej aktuella.

Databasen skapades på arbetsstationen enligt standardmässigt förfarande via SQL Servers administrationsverktyg *Enterprise Manager*. Den gavs namnet *Exjobb* och via ytterligare ett av SQL Servers administrationsgränssnitt *Query Analyzer* kördes sedan skriptet och tabellerna skapades.

För att kunna programmera *Säljman* krävdes inte bara att databasen *Exjobb* fanns tillgänglig utan även Vismas databas behövdes. Optinet exporterade sin databas till Microsoft Access format vilken sedan importerades till arbetsstationen

8.4.2 Gränssnitt och funktionalitet

Arbetet med att programmera *Säljman* i Visual Studio .NET har inte varit helt problemfritt. Hur vi har gått tillväga för att skapa den funktionalitet vi i analysfasen kommit fram till, och koppla den till det gränssnitt vi skapade i designfasen redovisas i en beskrivning för varje sida. Där redovisar vi de val som har uppstått och de problem vi har stött på under realiseringens gång. All programkod finns att läsa på den bifogade CD-skivan, Appendix N.

8.4.2.1 Litteratur

Inledningsvis läste vi in oss på ASP.NET för att färre problem skulle uppstå under arbetets gång. Det skedde parallellt med analys- och designfasen och den litteratur vi har använt oss av under inläsningen och även vid senare problemlösning är *Teach Yourself ASP.NET in 21 Days* [9], *ASP.NET Unleashed* [10], *ASP.NET Kick Start* [11] och *Beginning Asp.NET Databases using VB.NET* [12].

Under arbetets gång har vi även haft stor hjälp av *ASP.NET Forums* [13] och sökmotorn *Google* [14]. Boken *Mastering Regular Expressions* [15] har varit till nytta vid implementeringen av felkontroller

8.4.2.2 ASP.NET, VB.NET och Visual Studio .NET

ASP.NET är en del av ramverket .NET som tillhandahålls gratis av Microsoft. Detta innebär att ASP.NET kan utnyttja hela .NET:s klassbibliotek som innehåller ett stort antal funktioner. Om man tidigare i ASP ville hämta data från en databas och visa den i en enkel rutnätskontroll var man tvungen att skriva en hel del kod själv. I ASP.NET finns det nu färdiga funktioner och komponenter som bl.a. datagrid⁵. Man slipper att skriva kod för att visa informationen, det räcker med att skriva kod för att binda data till en datagrid som sedan renderar en tabell automatiskt. En annan fördel ASP.NET har gentemot ASP, är komponenter för felkontroller vid inmatning, vilket sparar tid genom att man slipper att skriva kontrollslingor för hand. All kod i ASP.NET kompileras före den exekveras och körs på webbservern. Det leder till att den blir snabbare i jämförelse med icke kompilerad kod som bl.a. används av skriptspråket VBScript tillsammans med ASP. Dock måste ASP.NET-komponenter för t.ex. en tabell omslutas av ASP-taggar såsom `<asp:table>` för att särskilja dem från standardkomponenter i HTML som t.ex. `<table>`. Ofta läggs dessutom attributet `runat="server"` till, för att komponenten skall köras på webbservern och inte på klienten. Dessa är bara en del av del fördelar ASP.NET har gentemot ASP, dock finns det även nackdelar som brist på dokumentation och kunskap eftersom ASP.NET är relativt nytt.

Nytt är också VB.NET (Visual Basic .NET), vilket är programspråket vi har valt att programmera i. Valet stod mellan C# och VB.NET vilka bägge stöds i ASP.NET. En av de största anledningarna till att vi valde just VB.NET är att den mesta litteraturen om ASP.NET handlar om VB.NET. Ett flertal förbättringar har gjorts sedan VB6. Det är fullständigt objektorienterat och felhanteringen har förbättrats i och med implementationen av `try` och `catch` som även förekommer i t.ex. programspråket Java. Viktigt att notera är att man inte skall förledas att tro att VB.NET har något gemensamt med VBScript, som användes tillsammans med ASP. VBScript är som namnet antyder ett skriptspråk, med stora skillnader från ett programspråk som VB.NET.

Utvecklingsverktyget vi har använt är Visual Studio .NET Professional, vilket är det främsta verktyget för utveckling med VB.NET. Det har inbyggt stöd för bl.a. ASP.NET, VB.NET och C#. Det är ett komplext verktyg vilket tog ett par veckor att kunna arbeta obehindrat i. Vi hade nytta av de böcker som fanns till förfogande, i synnerhet *ASP.NET Kickstart* [11] som tog upp just utveckling i Visual Studio .NET. En av flera bra funktioner i Visual Studio .NET är att ASP.NET-komponenter är lättåtkomliga från en gömd panel på vänstersidan av skärmen, och med en enkel knapptryckning drar man ut vald komponent på skärmen. Dessutom finns färdiga komponenter att använda sig av för att skapa databaskopplingar. Dock skapas i vårt tycke en väl omständig kod av Visual Studio .NET, och därför har vi valt att själva programmera databaskopplingarna. Varje webbsida som Visual Studio .NET skapar består av en designsida som innehåller all HTML-kod och alla ASP.NET-komponenter samt en sida som innehåller VB.NET

⁵ Komponent i ASP.NET, används för att presentera data (ofta från en databas).

koden. Fördelen med detta är att en person kan arbeta med designen och en annan person kan arbeta med logiken till samma sida samtidigt. En annan uppskattad funktion är möjligheten till att kunna sätta in brytpunkter i programmet. När det körs stannar programmet upp vid dessa brytpunkter, och då finns möjlighet att bl.a. kontrollera variabelvärden vilket underlättar vid felsökning. Ett mindre problem som uppstod efter vi arbetat med Visual Studio .NET, var att HTML-koden på designsidan formaterats om, indentering ändrats och den blev svårsläst. Detta löstes enkelt genom att välja bort automatisk formatering under *Options*. Ytterligare problem som vållade en del bekymmer i början, var när bägge projektdeltagarna arbetade på samma projekt genom två olika instanser av Visual Studio .NET på samma dator. Det resulterade ofta i en del märkliga felmeddelanden och för att kunna fortsätta arbeta skapade vi ett nytt projekt där en av oss arbetade separat med funktioner och sidor, vilka vi sedan infogade i det första projektet. D.v.s. vi jobbade med vart sitt projekt, som mot slutet av realiseringen fogades ihop till ett.

8.4.2.3 Allmänt om Säljman

Säljman är uppbyggt av HTML-ramar i ett ramverk. Initialt fanns det fem ramar, en ram i mitten omgiven av smalare ramar på kanterna runtomkring. Detta kom senare under realiseringen att innebära en begränsning i hur mycket information som kunde visas på skärmen utan att använda rullningslistan. Av den anledningen togs den nedre och den högra ramen bort. De tre ramarna som finns kvar och används är menyramen till vänster, en ram på ovansidan med Optinets logo och mittenramen där informationen visas. För att styra utseende på innehållet har vi använt oss av *Cascading Style Sheets* (CSS). Då designen av formulären på webbsidorna är klar (se kap. 8.3.2) återstår programmeringen av funktionaliteten (se Appendix N för *Säljmans* kod).

Under programmeringsarbetet har vi försökt lägga de funktioner som används av flera sidor i en egen modul. Det är inte alltid det har varit möjligt, då programkoden inne i funktionen ibland påverkar komponenter på sidan. Då måste funktionen finnas på den sidan komponenten finns.

8.4.2.4 Meny och startsida

Arbetet med att realisera funktionaliteten i menyn och på startsidan vållade inga större problem. Vid inladdningen av startsidan hämtas säljarnas namn från Vismas databas. Då kolumnen *EmpNo* (se kap. 8.1.1.1) i Visma anger säljare, användes det först för att identifiera namnen och binda dem till en rullgardinsmeny. Det resulterade i att för många namn visades, bl.a. på tidigare anställda och anställda vilka inte kommer att behöva använda *Säljman*. För att ytterligare specificera vilka namn som skulle visas används fältet *inf7* i Vismas databas. Det är ett tidigare oanvänt reservfält och om Optinet vill att en anställd skall kunna logga in i *Säljman* matar de in värdet 1 i *inf7*. Vid inloggning lagras namnet i en sessionsvariabel som har fått en veckas giltighetstid.

Detta enligt Optinets önskemål för att inte säljarna skall behöva logga in flera gånger varje dag. Då säljarnas namn hämtas från databasen lagras de i en cache som används på flera ställen i *Säljman*. Då säljaren väljer sitt namn visas det under menyn (se kap. 8.3.2.1). Det problemet vi hade var att få menyramen att uppdateras så att säljarens namn syntes. Det löstes med att anropa en JavaScript funktion där ny inläsning av sidan sker. Problemet löstes med hjälp av sökningar på *Google* [14].

8.4.2.5 Skapa offert

Vid realiseringen av sidan *Skapa offert* fanns en del problem med databaskopplingar. Sidan var den första där de implementerades och det var tidskrävande att få det att fungera. Med litteraturen (se kap. 8.4.2.1) som stöd löste vi det och arbetade sedan med funktionaliteten i formuläret. Vid inladdning av sidan kopplas först säljarnas cachade namn till rullgardinsmenyn och om säljare är inloggad blir dennes namn förvalt. Vidare hämtas alla kundnamn från Vismas databas, också dem till en rullgardinsmeny. Det är många kunder och det finns två genvägar till att välja rätt kund. Antingen matar användaren in ett kundnummer eller kundens begynnelsebokstäver. För varje ändrad inmatning i de fälten förfrågas Vismas databas efter ny information och rullgardinsmenyn uppdateras med de kunder som matchar. Vid inmatning av begynnelsebokstäver kan flera bokstäver anges och alla matchande kundnamn visas medan inmatning av ett kundnummer endast genererar en kund. En kontroll sker av inmatningen av kundnumret för att se till att bara siffror anges. Då användaren väljer en kund i rullgardinsmenyn hämtas kundens referens från Vismas databas och visas i en textruta.

Innan offertposten skapas i databasen *Exjobb* sker en kontroll av formuläret. Det som kontrolleras är att kund och säljare är vald och att offertens avseende är inmatad. Inmatning till databasen sker inte förrän informationen är korrekt. Då posten sparas returneras ett löpnummer som visas på skärmen. En länk visas också till sidan *Offertfiler* där säljaren kan börja skapa offertens filer.

8.4.2.6 Bearbeta offerter

Då vi hade löst problemet med hur databaskopplingarna skapades gick det lättare att på sidan *Bearbeta offerter* implementera förfrågningar till databasen. Ett annat problem uppstod rörande kundens namn och dess kundnummer. I databasen *Exjobb* lagras bara kundnumret och vid visning på skärmen skall även kundens namn visas. Att skicka en förfrågan till två databaser visade sig vara besvärligt. Med hjälp av *ASP.NET Forums* [13] löste vi det problemet. Ytterligare ett problem var visningen av informationen på skärmen. Vi hade tre komponenter att välja mellan; repeater, datalist eller datagrid [9, 12].

Repeatern är den enklaste komponenten för visning informationen där uppdateringar inte är möjliga. Komponentens datalist är mer avancerad och har inbyggda funktioner för

uppdatering och lämpar sig för enkla databasstrukturer. Datagriden (se figur 8-26, Appendix L för fullstorlek) är den mest avancerade komponenten för visning av data från mer komplexa databaser. Den har kraftfulla inbyggda funktioner för uppdatering och har störst möjligheter till skräddarsydd design [9, 12]. Vi valde att använda oss av komponenten datagrid p.g.a. de inbyggda funktionerna för uppdatering och dess designmöjligheter. Litteraturen erbjöd inte tillräckligt med information angående formateringen av datagriden så med hjälp av *An Extensive Examination of the DataGrid Web Control* [16] läste vi in oss på komponenten. I arbetet med den använde vi oss också av *ASP.NET Forums* [13] och *Google* [14].

Löpnr: 1005	Kund: 100592 A+A Corporate Design AB		
Säljare: Rune Wallgren	Referens: Anders Andersson		
Avseende: Serverköp		Ändra	Ta bort
Kommentar: Är nog också intresserad av Säker Internet			Filer
Lämnad: 2003-06-01	Giltig: 2003-06-21		
Tagen: <input checked="" type="checkbox"/> Tappad: <input type="checkbox"/>	Besvarad: 2003-06-15		
Totalbelopp: 24000	Täckningsbidrag: 4500	Tjänst:	

Löpnr: 1006	Kund: 100003 Zimmer Scandinavia AB		
Säljare: <input type="text" value="Rune Wallgren"/>	Referens: <input type="text" value="Per Persson"/>		
Avseende: <input type="text" value="Säker Internet"/>		Uppdatera	Avbryt
Kommentar: <input type="text"/>		Ta bort	Filer
Lämnad (YYYY-MM-DD): <input type="text" value="2003-07-15"/>	Giltig (YYYY-MM-DD): <input type="text" value="2003-09-05"/>		
Tagen: <input type="checkbox"/> Tappad: <input type="checkbox"/>	Besvarad: <input type="text"/>		
Totalbelopp: <input type="text"/>	Täckningsbidrag: <input type="text"/>	Tjänst: <input type="text"/>	

Figur 8-26 Visning av information i en datagrid.

Via formuläret väljer säljaren de sökkriterier som behövs för att visa den information som skall bearbetas. Informationen visas i datagriden där det finns möjlighet att ta bort enskilda poster och uppdatera dem. Vid varje borttagning eller ändring av information postas formuläret till servern där databasen uppdateras och den uppdaterade informationen skickas tillbaka till datagriden. Uppdateringen sker utan att säljaren navigerar bort från sidan. Då säljaren vill skapa eller ladda upp filer till en offert navigerar säljaren från datagriden till sidan *Offertfiler* via knappen *Filer*. Då skickas offertens löpnummer med och filerna till rätt offert visas direkt.

Vid listningen av information sker en kontroll av varje post. Den kontrollerar så att informationen är konsistent. Felmeddelande visas vid följande händelser; om giltighetsdatumet är innan lämnatdatumet, tagen eller tappad är ikryssad men besvaratdatum saknas, offerten är besvarad men något av beloppen saknas och om besvaratdatum är inmatat men tagen eller tappad inte är angivet.

8.4.2.7 Offertfiler

Från sidan *Offertfiler* har användaren åtkomst till de filer som tillhör en specifik offert. Även skapandet av nya filer till en offert sker härifrån. Användaren börjar med att mata in offertens löpnummer, därefter visas poster med eventuella tillhörande filer. Det som

visas är löpnummer, filnamn, benämning, version, kategori, skapad datum, om filen är uppladdad samt om den är arkiverad. Varje fil tillhör en specifik kategori beroende på dess syfte. Vi har i samarbete med Optinet valt att dela in filtyperna i fyra kategorier; underlag, offert, avtal och övrig. Underlag behövs för att ha ett prisunderlag där inköps- och försäljningspris, kvantitet m.m. finns angivet. Detta dokument skickas inte med till kund utan är bara till för Optinets säljare. Kategorin offert består av ett Worddokument som skickas ut till kund. Till varje offert skickas vanligtvis ett avtal med där olika avtal hör till olika tjänster. Slutligen finns en övrig kategori, där filer som inte passar in i de andra kategorierna kan läggas till, t.ex. bildfiler, pdf dokument m.m.

För dessa fyra kategorier finns det ett antal olika mallar som läggs upp på webbservern vilka används för skapande av nya filer. När användaren vill skapa en ny underlagsfil eller avtalsfil, väljer han en mall från en rullgardinsmeny och skriver en frivillig kommentar om filen. När användaren klickar på knappen *Skapa* görs en kopia av vald mall, ett automatiskt filnamn genereras, en post skapas i databasen med information om löpnummer, filnamn, kommentar, version, typ av fil samt vilket datum den skapades. Dock läggs själva filen inte till i databasen, utan den skickas till användaren som laddar ner den lokalt på sin arbetsstation. Användaren kan då ändra och lägga till i filen och när han är nöjd, ladda upp den till databasen. Först då läggs själva filen till i den post som tidigare skapats. Vid skapande av en offertfil görs inte en kopia av mallen utan offertspecifik data såsom kundnamn, referens, säljare och giltighetsdatum läggs automatiskt till i offertfilen. Detta presenteras mera ingående under stycket ”Interaktion med Word”. Dock är rutinen annars likvärdig med skapande av underlags- och avtalsfiler i övriga hänseenden. Vid skapande av en fil i sista kategorin, övrig, skiljer sig händelseförloppet gentemot tidigare kategorier. Det finns ingen mall som användaren använder sig av, utan filen laddas upp direkt till databasen via en uppladdningssida. På denna sida får användaren lägga till en frivillig kommentar om filen, sedan välja fil att ladda upp. En post om filen skapas samtidigt som filen laddas upp till databasen.

Vid uppladdning av filer till tidigare skapade poster finns kontroll att användaren av misstag inte laddar upp fel fil av misstag. Detta görs genom en enkel namnkontroll av filen mot filnamnet i posten. Är filnamnet fel kommer inte filen att kunna laddas upp till databasen.

8.4.2.7.1 Namngivning

Namnen på filerna som skapas följer en standard, utarbetad tillsammans med Optinet, för att underlätta hanteringen av dem när de befinner sig på säljarens dator. Först kommer kundnamnet, där alla blanksteg har ersatts med ett understreck (_), därefter ett minus, löpnummer, ytterligare ett minus, kategori (underlag, offert, avtal, övrig), minus, ett v och versionsnumret. Till detta läggs filsuffixet till, vilket är beroende på filtyp, t.ex. .doc för Word, .xls för Excel o.s.v. Ett par exempel på hur filnamn kan se ut är ”ab_dahls_konditori-1013-avtal-v1.xls” och ”everts_snickeri-1013-offert-v2.doc”. Namnet automatgenereras från *Säljman* vilket har varit ett önskemål från Optinet. Filer

under kategorin övrig, som laddas upp samtidigt som post skapas i databasen, döps automatiskt om till att följa namnstandarderna.

8.4.2.7.2 Versionshantering

Det händer i upp till 50 % av fallen att en offert som är utskickad kommer tillbaka för revidering. Anledningar till detta kan vara omförhandlingar i pris, tjänst eller kvantitet. Därför är versionshantering en önskvärd funktion i *Säljman*. Den innebär att man kan lägga till nya versioner av en fil till en offert. Fördelen med att ha en versionshantering är att man automatiskt kan se vilken fil som är den nyaste och därmed mest aktuell. Versionshantering är implementerad för alla fyra filkategorier. Versionshantering är automatisk, vilket menas att den sker utan påverkan från användarna. T.ex. om användaren skapar en ny offertfil, får den ett högre versionsnummer än de(n) som redan finns. Undantag är om en post är skapad men fil inte uppladdad, då meddelas användaren att ny version inte kan skapas före fil till tillhörande post först har laddats upp.

8.4.2.7.3 Interaktion med Word

En önskvärd funktion som tidigare nämnts är möjligheten att överföra data automatiskt från webbgränssnittet till ett Word dokument. Detta för att Optinet skall kunna ha olika mallar dit kundspecifik information skall överföras. Även om det går enkelt att själv manuellt lägga till kundnamn, referens, säljare och datum är det här ett typiskt exempel på arbetsuppgifter datorer är bra på, nämligen att automatisera enklare göromål. Att låta en webbsida kommunicera med en applikation, i det här fallet Word är dock inte helt trivialt. Det finns en del viktiga aspekter som måste tas i beaktning vid en sådan här lösning, som säkerhet, rättigheter m.m.

Installationen gick tillväga som så att först gjordes en standardinstallation av Office XP på webbservern. Det medför en risk att om applikationen kraschar eller hänger sig, kan i värsta fall hela webbservern bli lidande. Men utan Office XP installerat, hade den här önskvärda funktionen inte funnits. För att kunna manipulera Word genom ASP.NET krävdes installation av *Office XP Primary Interop Assembly (PIA)* vilket är sätt att anropa *unmanaged (COM⁶) code* från *managed (.NET) code* med hjälp av .NET Framework [17]. Word och Office XP räknas som *unmanaged code*, eftersom det är inte uppbyggt av ramverket .NET. Nästa version av Office kommer att vara uppbyggt av .NET, vilket innebär att mellanliggande program som Office XP PIA inte längre är nödvändiga vid programutveckling.

Efter installationen av PIA, var en referens till den tvungen att läggas in i projektet i Visual Studio .NET. Slutligen var vi tvungna att ge ASPNET-användaren⁷ större rättigheter för att den skulle kunna skapa och modifiera Word dokument. Därefter var

⁶ Component Object Model (COM) – Teknologi för att interagera med applikationer.

⁷ En användare som läggs till i gruppen Användare när man installerar Visual Studio .NET.

det möjligt att anropa Word från ASP.NET och utnyttja Words funktioner för att skapa nytt dokument baserat på vald mall. För att veta var i dokumentet information skulle läggas in, var man tvungen att modifiera Word-mallarna, genom att lägga till s.k. bokmärken (*bookmarks*). Fem bokmärken lades in; företagsnamn, referens, säljare, offertdatum och slutdatum. Dessa namn används sedan i ASP.NET-koden för att specificera vart i dokumentet information skall placeras.

8.4.2.8 Boka nytt kundmöte

Boka nytt kundmöte skiljer sig inte så mycket från *Skapa offert* (se kap. 8.4.2.5). Kod med databaskopplingar kan kopieras in här från den sidan där ändringar i koden görs för att passa formuläret på sidan *Boka nytt kundmöte*.

Formuläret innehåller kontroller för att fälten kundnamn, avseende, vem som bokat och vem som skall gå på mötet är inmatat. Även att datumet för mötet är efter bokningsdatumet kontrolleras. Utan korrekt inmatning postas inte formuläret, utan felmeddelande ges.

Då formuläret postas och databasen uppdateras visas ett meddelande på skärmen med information om mötesbokningen; vilken kund mötet är bokat med och vilket datum.

8.4.2.9 Bearbeta kundmöten

Bearbeta kundmöten och *Bearbeta offerter* (se kap. 8.4.2.6) är likartade programmeringsmässigt. Delar av programkoden gällande databaskopplingar och visning av information i datagriden kopierades från sidan *Bearbeta offert*.

I formuläret anger säljaren de sökkriterier han behöver för att visa information. Kriterierna är kundnamn, kommande möten, utförda möten och uppföljning. Används kommande eller utförda möten kan flera möjliga träffar genereras medan kundnamn endast genererar en träff. I rullgardinsmenyn väljs typ av uppföljning: antingen alla som är markerade för uppföljning, kvartalsvis via Q1, Q2, Q3 och Q4 eller alternativet ej intressanta som visar alla kundmöten där uppföljning är avmarkerat. Formulärets felkontroll tillåter inte bokstäver i fältet kundnummer.

Då informationen visas i datagriden sker en kontroll av att informationen är konsistent. Det kontrolleras också om kunden är markerad för uppföljning men det är inte specificerat vilken sorts uppföljning det är.

8.4.2.10 Sök offerter

Funktionaliteten på söksidan av offerter är likartad den på sidan *Bearbeta offerter* (se kap. 8.4.2.6). Programkoden kopierades därifrån och ändrades för att passa den här sidan. En skillnad gentemot *Bearbeta offerter* är att sökningen kan tidsbegränsas. Detta för att endast informationen om offerter från den gångna verksamhetsperioden skall

visas. Tidsbegränsningen jämförs mot offertens lämnatdatum eller giltighetsdatum, beroende på vilka sökkriterier för öppna och besvarade offerter som har angivits.

Ytterligare en skillnad från *Bearbeta offerter* är att sist i datagriden skall en sammanräkning göras av totalt antal offerter, antal tagna offerter och summan av totalbelopp, täckningsbidrag och tjänst. Det löstes med att för varje rad i datagriden anropa en funktion där sammanräkningen sker.

Felkontroller som sker vid visning av information i datagriden är samma som den på *Bearbeta offerter*.

8.4.2.11 Sök kundmöten

Sök kundmöten har en liknande funktionalitet som *Bearbeta kundmöten*. Programkoden kopierades därifrån och ändrades till att passa den här sidan. Sökningen kan även här tidsbegränsas, som på sidan *Sök offerter*. Tidsbegränsningen jämförs mot det datum mötet utfördes.

På samma sätt som på *Sök offerter* sker en sammanräkning i datagriden. Informationen som beräknas är antal utförda möten och antal möten som skall följas upp.

Felkontroller som sker vid visning av information i datagriden är samma som den på *Bearbeta kundmöten* (se kap. 8.4.2.9).

8.4.2.12 Säljmöte

Implementeringen av funktioner för uppföljning under sidan *Säljmöte* löpte utan större problem. Funktionaliteten är enkel. Då knappen *Säljmöte* trycks in beräknas alla nyckeltal genom en rad förfrågningar till databasen. De redovisas på webbsidan där mål matas in. När det är klart lagras informationen till databasen.

Sist på sidan visas en summering av tidigare genomförda säljmöten i en datagrid. Detta för att ge Optinet en helhetsbild av den gångna verksamheten.

8.4.2.13 Administrering

Funktionaliteten på *Administrering* är begränsad. Den innehåller inställningar för vart i katalogstrukturen filer skall arkiveras samt sökvägar till de mallar företaget använder för offertering. För att ställa in sökvägen skrivs rätt sökväg in i textrutan. En kontroll sker då på att sökvägen är giltig och lagring sker till tabellen *admin*. Vid inladdning av sidan *Offertfiler*, (se kap. 8.4.2.7) läses sökvägarna in och lagras i sessionsvariabler. Programmeringen av sökvägarna till mallarna innefattade inga nämnvärda problem. Vid implementeringen av arkiveringsfunktionen uppstod problem med automatisk kopiering av flera filer från databasen utan användarens inblandning. Funktionen är enligt kravspecifikationen inte nödvändig utan endast önskvärd. Funktionen har inte kunnat implementeras inom tidsramen för projektet.

8.5 Testning

Under realiseringens gång har vi testat varje enskild komponent för sig då vi har programmerat dem. Tyvärr innebär inte det att när systemet är klart kommer det att fungera utan fel. Testningsfasen är en viktig del av arbetet där mycket energi skall läggas för att hitta så många som möjligt av *Säljmans* fel. Under arbetets gång har vi hela tiden arbetat med projektet på arbetsdatorn och testkört via Visual Studio .NET. I testningsfasen anropar vi *Säljman* via webbläsaren från en annan dator. Anledningen till det är för att se att interaktionen mellan Microsoft Internet Explorer, IIS 5, SQL Server 2000 och .NET Framework fungerar som det ska.

Metoden vi använde under testningen var att först göra en ominstallation av databasen med hjälp av SQL-skriptet och sedan sida för sida arbeta oss igenom *Säljman*. Testningen har delats in i två delar; testning av gränssnitt och testning av funktionalitet. Testningen av funktionaliteten innebär inte bara att det skall fungera utan också att det är rätt funktionalitet och att det överensstämmer med de mål som har satts upp. Testningen har också skett i två omgångar; en gång endast av projektgruppen och en gång i samarbete med Optinets användare. De fel vi har hittat under testningsfasen har också korrigerats.

8.5.1 Gränssnitt och användarvänlighet

Vi upptäckte att tabbordningen på komponenterna inte var korrekt inställt på alla *Säljmans* webbsidor. Tabbordningen har som funktionalitet att låta användaren navigera på ett enkelt sätt i formulären med tabulatorknappen utan att använda musen.

Vidare uppkom synpunkter på textstorleken i listningen av filer och säljmöten i datagridarna på sidorna *Offertfiler* respektive *Säljmöte*. Storleken var för stor vilket innebar att en rullningslist visades och att användaren fick rulla sidan horisontellt för att läsa all information. Textstorleken minskades samt sidans ramverk ändrades så att bredden på menyramen minskades.

Vi har under arbetets gång haft återkommande kontakter med Optinet där gränssnittet diskuterats och genomarbetats. Inga övriga synpunkter framkom om gränssnittet vid testningen.

8.5.2 Funktionalitet

Vid testningen av funktionaliteten upptäckte vi att sidorna *Bearbeta offert* och *Sök offerter* inte kunde lista sökta poster från databasen. Orsaken till felet var att *Säljman* läste in *null*-värden från databasen och kunde inte konvertera dessa till booleska variabler. Exakt varför felet uppstod efter ominstallationen av databasen vet vi inte men felet åtgärdades genom att skriva om funktionen.

På ovan nämnda sidor upptäckte vi också att felkontrollerna (se kap. 8.4.2.6 och 8.4.2.10) av databasens information inte var helt korrekta vilket skulle kunna leda till att informationen inte förblir konsistent. Kontrollfunktionen sågs över och korrigerades.

Då vi testade att boka ett möte märkte vi att felkontrollen av inmatade datum i kalenderfunktionen inte stämde. Det var möjligt att mata in ett mötesdatum som var tidigare än bokningsdatumet i de fall bokningsdatumet var det standardvärde formuläret initierades med. ASP.NET:s inbyggda funktioner för kontroll av komponenter gick inte att använda på kalenderkontroller. Därför skrevs valda datum till osynliga textrutor där kontrollen skedde. Vid initieringen av formuläret skrevs dock inte det förvalda datumet till textrutorna. Det ledde till att kontrollen inte fungerade när användaren inte valde båda datumen.

Ett fel som kunde ha lett till att fylla webbserverns hårddiskutrymme var att då användaren laddade upp en fil till databasen, mellanlagrades den på servern. Filen togs inte bort från webbservern efter uppladdning vilket åtgärdades.

De fel som hittades under testningsfasen var starkt hämmande för *Säljmans* funktionalitet, men erbjöd inga större svårigheter att lösa. Optinet användare var belåtna med funktionaliteten i systemet när väl ovanstående problem hade rättats till.

8.6 Leverans

Då *Säljman* var realiserat och testat återstod att leverera produkten till Optinet. Det skedde på en CD-skiva tillsammans med systemets dokumentation. Webbplatsen ligger i en katalog på skivan vilket skall kopieras till webbservern. Den dokumentation som ingick var uppbyggnaden av databasen i form av entitets/rerelationsdiagram (se Appendix G) och objekt/termkatalog (se Appendix H) samt SQL-skript för installation av databasens tabeller (se Appendix K).

Vidare ingick systemdokumentationen (se Appendix M) bestående av en installationsanvisning, en listning av alla projektets filer samt en listning av alla subrutiner och funktioner i dessa filer tillsammans med beskrivningar.

9 Reflektion

Då den metod vi har använt innebär en indelning av arbetet i fem faser, har vi valt att dela in analysen i samma fem faser där vi går igenom utfallet av arbetet. Vi tar upp hur arbetet har förflutit gentemot planeringen och vilka avvikelser som har inträffat under arbetets gång.

9.1 Analys

Valet att göra ett projektdirektiv och en projektplan i början av projektet har lett till en förståelse av projektets innehåll och har underlättat arbetet i de efterkommande faserna. Den tidsplanering vi gjorde inledningsvis visade sig inte stämma då den var för snålt tilltagen. Anledningen till detta var en missbedömning av tidsåtgången för de olika faserna. Vi har haft problem under arbetets gång, men omfattningen och konsekvenserna av dem i riskanalysen har överensstämmt med verkligheten. Dock glömde vi helt att ta med tidsplaneringen som en risk för projektet. I efterhand visade det sig vara den största risken.

Analysen av Optinets verksamhet och arbetsätt med avseende på offerthantering, nykundsbearbetning och uppföljning har gått som planerat. Den har varit en viktig grund för skapande av kravspecifikation och de avgörande beslut vi har tagit i efterkommande faser. Under projektets gång har vi regelbundet stämt av arbetet mot kravspecifikationen för att säkerställa *Säljman*s funktionalitet.

De val vi gjorde i slutet av analysfasen med avseende på informationslagring, utvecklingsmiljö och programspråk har i efterhand visat sig vara korrekta val. Dock har valet av ASP.NET inneburit en oväntat hög inlärningströskel vilket också bidragit till att tidsplaneringen brast. Hade vi valt t.ex. ASP som vi då behärskade bättre, hade vi haft större möjligheter att fullfölja projektet inom tidsramarna, men en del av funktionaliteten i *Säljman* hade gått förlorad.

9.2 Design

Designfasen löpte utan större komplikationer. Vi designade databasens struktur och uppbyggnad. Arbetet med strukturen av databasen var inledningsvis svårt. Det var en iterativ process där vi gick igenom och finslipade strukturen flera gånger. Vi tog även hjälp av referensgruppen vid framtagande av databasstrukturen vilket bidrog till att det senare skapandet av SQL-frågor förenklades.

Då vi skapade designen av gränssnittet gjordes först ett förslag på papper som diskuterades med Optinet. Utifrån det skapade vi designen i Visual Studio .NET vilket visade sig vara ett lätt sätt att skapa, ändra och utvärdera gränssnittet både på arbetsdatorn och vid möten hos Optinet. Än så länge låg projektet i fas tidsmässigt.

9.3 Realisering

Realiseringen av databasen i SQL Server 2000 innebar inga överraskningar. Skapandet av SQL-skript och uppsättning databasapplikationen förlöpte planenligt tack vare tidigare kunskaper inom de områdena.

Vid realiseringen av *Säljman*, d.v.s. programmeringen där funktionaliteten kopplades till gränssnittet innebar större problem än väntat. Inlärningströskeln var hög för ASP.NET, VB.NET samt Visual Studio .NET. Vi hade införskaffat en del litteratur (se kap. 8.4.2.1) om ämnet och de visade sig inte vara tillräckligt detaljerade för att lösa de specifika problem som uppstod i den här fasen. Ny litteratur införskaffades för att lösa dessa problem tillsammans med sökningar på Internet.

Från början använde vi Visual Studio .NET:s inbyggda funktioner för att skapa databaskopplingar. Vi hade räknat med att den funktionaliteten Visual Studio .NET erbjöd skulle räcka till. Det gjorde den inte utan skapade onödigt mycket kod som dessutom var väldigt svårförståelig. För att komma runt det problem skapades alla databaskopplingar för hand. Inledningsvis var det tidskrävande men sparade en del tid totalt i realiseringsfasen. Till att tidsplaneringen sprack bidrog också det tidskrävande arbetet med visning av information i datagridar, filuppladdning till databas samt kopplingen till Word.

Namnstandarderna vi skapade under designfasen som hjälp till en enhetlig programmering kunde ha följts bättre. Efter att den skapades föll den i glömska. Trots det skedde den mesta namngivningen enligt standarden. Vi fick dock i efterhand göra en del korrigeringar av variabel- och funktionsnamn

9.4 Testning

Projektgruppens testning skedde enligt plan, men testningen tillsammans med Optinet blev inte så omfattande som vi först hade planerat. Det var på grund av att projektet drog ut på tiden och att Optinets användare var på semester då *Säljman* färdigställdes. Dock testades *Säljman* tillsammans med Optinet under en dag. Utfallet av testningen var trots det positivt. I efterhand borde planeringen av testningstillfällena ha gjorts med större framförhållning.

9.5 Leverans

Leveransen av produkten var okomplicerad. Det skedde som planerat på en CD-skiva med tillhörande dokumentation. Dock kunde ett installationskript ha skapats för att automatisera installationen av en del av *Säljman*. Vi ansåg inte att det var nödvändigt att skapa ett installationskript då det bara gällde att kopiera filerna till webbservern.

10 Slutsats

Vi anser att vårt arbete uppnått de kvantifierbara mål som är uppställda av projektgruppen tillsammans med Optinet samt att *Säljman* uppfyller de krav som ställts i kravspecifikationen.

Gränssnittet är utformat för att skapa ett stegvis arbetssätt kring offerthantering, nykundsbearbetning och uppföljning. Informationslagringen sker centralt och kontroller har införts i *Säljman* för att säkerställa att lagrad information är konsistent och att redundansen har minskat. Dessutom har åtkomligheten av informationen förbättrats då det är möjligt för användarna att komma åt *Säljman* utifrån företaget, samt att flera samtida användare stöds.

Genom *Säljman* har möjligheterna till att enkelt ta fram kundhistorik ökat genom att införa versionshantering och sökmöjligheter. Versionshanteringen gör det möjligt att spåra reviderade offerter för en komplett tillbakablick. Sökningen underlättar också framtagandet av kundhistorik, samt även framtagandet av diskussionsunderlag vid Optinets säljmöten där informationen är uppdateringsbar under mötets gång.

Jämfört med kravspecifikationen (se kap. 8.2) så har alla nödvändiga funktioner implementerats i *Säljman*. Inloggning på startsidan, sökning med hjälp av kundens begynnelsebokstäver och automatisk namngivning har implementerats. Däremot har inte komprimering av filer för lagring till databasen av anledningen att SQL Server 2000 inte stöder komprimering. Inte heller har visning av totalt använt databasutrymme samt arkiveringsfunktionen implementerats p.g.a. tidsbrist.

Den metod vi har använt för att fullfölja projektet har fungerat bra och i stort sett har den följts. Metoden och de inledningsvis täta kontakterna tillsammans med Optinet har lagt grunden för *Säljmans* utveckling. Trots den grundliga analysen förstod vi dock inte vidden av projektet och den tid som krävdes för att lösa de problem som uppstod. I efterhand inser vi att projektet skulle ha avgränsats mer med tanke på de resurser som stod till förfogande.

10.1 Rekommendationer till fortsatt utveckling

När det gäller offerthantering och nykundsbearbetning är *Säljman* anpassat till Optinets nuvarande verksamhet och behov. Därför har vi inga större rekommendationer i dagsläget till fortsatt utveckling av de områdena förutsatt att Optinet inte ändrar sitt arbetssätt. Offerthanteringen skulle i framtiden kunna utökas till att endast hantera filer på webben utan att först kopiera dem till användarens dator. Vinsten skulle vara att slippa hanteringen av lösa filer. Det skulle kunna göras möjligt i Office 2003 vilket kommer vara integrerat med ramverket .NET. Mötesfunktionen kan i framtiden utökas till att hantera bokningen av alla kundmöten och inte bara det första. Vinsten med det är att Optinets användare då slipper använda två system för olika typer av bokningar; *Säljman* och Outlook.

Säljmötesfunktionen är den funktionen som i dagsläget har störst utvecklingsmöjligheter. Den skulle kunna samla in mer statistik om verksamheten för att sedan kunna visas överskådligt med grafer.

Säkerheten i *Säljman* kan utökas på flera sätt. Rättigheter kan läggas till för att styra användarnas åtkomst av t.ex. sidorna administrering och säljmöte. Vidare så kan kontroll ske på varje sida av att användaren är en giltig användare med hjälp av sessionsvariabler. För att ytterligare öka kontrollen över *Säljman* kan loggning av användare och anslutningar införas.

Säljman har många databaskopplingar vilket vid många förfrågningar kan belasta nätverket eller ge långa svarstider. Det kan då vara lämpligt att byta dessa mot lagrade procedurer. SQL Server tar då själv hand om bearbetningen vid förfrågningarna och avlastar webbservern.

Slutligen kan *Säljman* utökas med de två önskvärda funktionerna från kravspecifikationen som inte implementerats. De är arkivering och visning av totalt utrymme för databasen. Projektgruppen kommer efter projektets avslutande att fortsätta samarbetet med Optinet för att implementera dessa två funktioner samt att vidareutveckla *Säljman*.

Källförteckning

- 1 Visma Software AB (2003). Visma. [Elektronisk]. Tillgänglig: <<http://www.visma.se/>> [2003-03-25].
- 2 Andersen, Erling S. (1994). *Systemutveckling: Principer, Metoder och Tekniker*. Lund: Studentlitteratur AB.
- 3 What?is.com : Sökning på VPN (senast uppdaterad 2003-02-11). [Elektronisk]. Tillgänglig: <<http://whatis.techtarget.com/>> [2003-03-25].
- 4 Klarspråk HB (1997). Tips för dig som arbetar med Microsoft Word. [Elektronisk]. Tillgänglig: <<http://www.klarsprak.se/wrd.htm>> [2003-07-22].
- 5 W3 Schools (2003). ASP .NET Tutorial. [Elektronisk]. Tillgänglig: <<http://www.w3schools.com/aspnet/default.asp>> [2003-07-17].
- 6 Wisén, Jan & Lindblom, Börje. (2001). *Effektivt Projektarbete*. Stockholm: Norstedts Juridik AB.
- 7 Forta, Ben (2000). *SQL på 10 minuter*. Malmö: PC Boken.
- 8 Database Journal (2003). MSSQL. [Elektronisk]. Tillgänglig: <<http://www.databasejournal.com/features/mssql/article.php/1442341>> [2003-06-20].
- 9 Payne, Chris (2002). *Teach Yourself ASP.NET in 21 Days*. USA: SAMS Publishing.
- 10 Walther, Stephen (2002). *ASP.NET Unleashed*. USA: SAMS Publishing.
- 11 Walther, Stephen (2003). *ASP.NET Kick Start*. USA: SAMS Publishing.
- 12 Kauffman, John et al. (2002). *Beginning ASP.NET Databases using VB.NET*. USA: WROX Press.
- 13 Microsoft ASP.NET : ASP.NET Forums. [Elektronisk]. Tillgänglig: <<http://asp.net/forums/>> [2003-07-08].
- 14 Google: Search Engine. [Elektronisk]. Tillgänglig: <<http://www.google.com/>> [2003-07-23].
- 15 Friedl, Jeffrey E.F. (2002). *Mastering Regular Expressions*. Sebastopol USA: O'Reilly & Associates Inc.
- 16 ASP.NET.4GuysfromRolla.com (2003). An Extensive Examination of the DataGrid Web Control. [Elektronisk]. Tillgänglig: <<http://aspnet.4guysfromrolla.com/articles/040502-1.aspx>> [2003-07-15].
- 17 Microsoft MSDN (2002). Working with the Office XP Primary Interop Assemblies [Elektronisk]. Tillgänglig: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnoxpta/html/odc_oxppias.asp> [2003-07-28].

Figurförteckning

Figur 8-1	Använda fält ur Vismas databas.....	8
Figur 8-2	Filregister för offerthantering.....	9
Figur 8-3	Filregister för nykundsbearbetning.....	9
Figur 8-4	Filregister för uppföljning.....	10
Figur 8-5	Flödesschema över offerthantering och nykundsbearbetning.....	11
Figur 8-6	Flödesschema över uppföljning vid säljmöte.....	12
Figur 8-7	Entitets/rerelationsdiagram över databasens struktur.....	18
Figur 8-8	Specifikation av tabellen <i>nykund</i>	19
Figur 8-9	Specifikation av tabellen <i>offert</i>	19
Figur 8-10	Specifikation av tabellen <i>filer</i>	20
Figur 8-11	Specifikation av tabellen <i>saljmote</i>	21
Figur 8-12	Specifikation av tabellen <i>admin</i>	21
Figur 8-13	Databasens storlek.....	22
Figur 8-14	Skiss över navigeringen inom <i>Säljman</i>	22
Figur 8-15	Gränssnitt för <i>Menyn</i> och <i>Startsidan</i>	23
Figur 8-16	Gränssnitt för sidan <i>Skapa ny offert</i>	24
Figur 8-17	Gränssnitt för sidan <i>Bearbeta offert</i>	24
Figur 8-18	Gränssnitt för sidan <i>Offertfiler</i>	25
Figur 8-19	Gränssnitt för sidan <i>Boka nytt kundmöte</i>	26
Figur 8-20	Gränssnitt för sidan <i>Bearbeta kundmöte</i>	27
Figur 8-21	Gränssnitt för sidan <i>Sök offerter</i>	27
Figur 8-22	Gränssnitt för sidan <i>Sök kundmöten</i>	28
Figur 8-23	Gränssnitt för sidan <i>Säljmöte</i>	28
Figur 8-24	Gränssnitt för sidan <i>Administrering</i>	29
Figur 8-25	Utdrag ur SQL-skriptet.....	30
Figur 8-26	Visning av information i en datagrid.....	35

Appendix A Projektdirektiv

Bakgrund

Optinet Data AB är ett IT-företag i Göteborg med inriktning mot drift och säkerhet. De har ca 10 anställda och erbjuder webbhotell och datorspecifika tjänster åt små och medelstora företag. Idag använder de ekonomi- och kundhanteringssystemet Visma Business. Där lagras och administreras viktiga funktioner som kunddata och fakturering. För funktionerna nykundsbearbetning och offerthantering används idag Excel och Word utan koppling till Visma Business databas.

Problemformulering

Idag när Optinet Data AB knyter nya kundkontakter läggs deras data in manuellt i ett Exceldokument. Där anges vem som har kontakt med kunden, bokade datum, den tjänst kunden är intresserad av, om offert skickats ut och om uppföljning skall ske. Vid utskick av offert och ändring av bokade datum görs detta manuellt vilket innebär merarbete. För offerter lagras dess specifika information i ett annat Exceldokument med information såsom kundinformation, tjänstens avseende, belopp, täckningsbidrag, datum rörande offertens giltighet, om offerten är tagen eller tappad och en länk till offertfilen (skriven i Word) och en underlagsfil (skriven i Excel). Vid försäljarnas internmöten används ytterligare ett Exceldokument med försäljningsmål av tjänster och produkter samt antalet bokade kundmöten diskuteras. Vid diskussionen används Excels inbyggda filteringsfunktion för att ta fram nyckelvärden. Detta förfarande anser Optinet Data AB vara omständigt samt att informationen inte kan komma åt utifrån företaget.

Åtgärd

För att effektivisera och samordna informationshanteringen kring nya kundkontakter, offerthantering och uppföljning önskar Optinet Data AB integrera ovanstående funktioner till ett system. Det webbaserade systemet skall ge åtkomst åt framförallt två sidor; nykundsbearbetning och offerthantering.

Förutom två nämnda sidor, skall även en funktion implementeras som gör det möjligt att ta fram nyckelvärden såsom budgeterade mål, nådda mål, antal bokade kundmöten etc. via ett webbgränssnitt.

Under nykundsbearbetning skall man lägga till och redigera sådana data som behövs för kontakt med nya och potentiella kunder. Exempel på detta är planerade besöksdatum, vad kunden är intresserad av, säljare, m.m. För att kunna arbeta med potentiella kunder under nykundsbearbetning krävs att de är redan inlagda i kunddatabassystemet (Visma Business). All denna information skall sparas på ett sådant sätt att den är lätt att inhämta och redigera via webbgränssnitt (t.ex. i databasform).

I offerthanteringen skall ingå visning av utsända offerter, någon form av revisionshantering om man behöver uppdatera offert, inläggning och utskrift av nya offerter, olika sorterings och filtreringsmöjligheter såsom visa vilka offerter inte blivit besvarade, när de förfaller, m.m. Den här informationen skall också sparas på liknande sätt som ovanstående.

Målet är att all ovanstående information skall kunna kommas åt från en gemensam plats på deras intranät.

Avgränsningar

Autentisering till det webbaserade systemet kommer inte att implementeras då det kommer att ligga inom Optinet Data AB:s eget intranät.

Underhåll, support och drift av systemet kommer inte att ske efter projektets avslut.

Projektdeltagare

Micael Ebbmar micke@ebbmar.net 0520 - 42 83 46

Jan-Åke Hedström jan-ake.hedstrom@telia.com 0521 - 602 13, 0707 - 56 87 32

Resurser

Den tid som finns till projektets förfogande uppgår till 400 timmar per deltagare, sammanlagt 800 timmar.

Bil för transport till och från Optinet Data AB.

Datorer: Tillgång till stationär dator med utvecklingsmiljö finns på Optinet Data AB. Projektdeltagarna tillhandahåller själva en bärbar dator.

Nödvändiga programvaror finns att tillgå från Optinet Data AB.

Den litteratur som behövs står projektdeltagarna för.

Tidsplanering

Gantt-schema		Vecka															
		7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Delmoment	Projektbeskrivning till HTU																
	Projektdirektiv till arbetsgivare																
	Projektplan till arbetsgivare																
	Systemering																
	Realisering av databas																
	Utveckling av webbgränssnitt																
	Skapande av databaskopplingar																
	Sök- och Filtreringsfunktioner																
	Koppling till Word																
	Inlämning av preliminärt utkast																
	Inlämning av examensarbete																
	Redovisning av examensarbete																
	Dokumentation																

Samråd

Samråd om systemet kommer att ske med personal på Optinet Data AB samt HTU angående systemets funktion och tekniska detaljer.

Uppdragsgivare: Rune Wallgren, Optinet Data AB, VD

rune@optinet.se

031 - 26 55 60, 0708 - 64 09 97

Referensgrupp: Rune Wallgren, Optinet Data AB, VD

Jan Mellqvist, Optinet Data AB, SQL Server ansvarig

janne@optinet.se

031 - 26 55 60, 0708 - 64 09 90

Andreas Boklund, HTU

andreas.boklund@htu.se

0520 - 41 51 77

Jonas Amoson, HTU

jonas.amoson@htu.se

0520 - 47 53 27

Examinator: Lena Pareto

lena.pareto@htu.se

0520 - 47 51 19

Handledare HTU: Linn Gustavsson

linn.gustavsson@htu.se

0520 - 47 51 60

Appendix B Projektplan

Bakgrund

Optinet Data AB är ett IT-företag i Göteborg med inriktning mot drift och säkerhet. De har ca 10 anställda och erbjuder webbhotell och datorspecifika tjänster åt små och medelstora företag. Idag använder de ekonomi- och kundhanteringssystemet Visma Business för befintliga kunder. Där lagras och administreras viktiga funktioner som kunddata och fakturering. Offerthantering sker separat där Excel och Word används utan koppling till Visma Business databas. Ytterligare sker hantering av nya potentiella kunder i Excel där det anges vilken tjänst de är intresserade av samt om offert utgått tillsammans med information om fortsatt kontakt samt värdet på tjänsterna. De företag som väljer att bli kunder läggs därefter in i Visma Business. Då Excel och Word används saknas möjlighet för flera anställda att arbeta med filerna samtidigt.

Projektbeskrivning

Systembeskrivning

Uppdragsgivarens önskemål är att komplettera sitt befintliga system, Visma Business, med ett webbaserat system för hantering av offerter. Offerter kommer att kunna skapas och sparas samt att versionshantering av offerter kommer att finnas. Ytterligare information om offerter som lagras är; säljare, giltighetstid, värde på tjänst, om offerten är tagen eller tappad samt underlagsfil.

Vidare kommer funktioner skapas för att lagra information om kundbesök med information som; ansvarig säljare, bokade datum och eventuell uppföljning.

Utifrån dessa två utökningar av systemet skall även sökfunktioner finnas för att kunna ta fram information om antalet utestående offerter, deras värde och antalet bokade kundbesök. Sökningen skall kunna göras för hela Optinet Data AB eller per försäljare och skall även kunna tidsbegränsas.

Flera anställda skall även kunna arbeta med systemet samtidigt.

Avgränsningar

Optinet Data AB:s befintliga system arbetar på en Microsoft plattform där SQL-server 2000 och IIS (Internet Information Service) 5.0 är grunden för deras intranät. Därför skall utökningarna av systemet använda den plattformen.

Systemet kommer att ligga på Optinet Data AB:s intranät vilket har en egen autentiseringsmekanism och därför kommer ingen autentisering skapas inom ramen för projektet.

Projektgruppen kommer inte att lämna kontinuerlig driftsupport efter avslutat projekt. Inte heller kommer backup av relevanta databaser att tas. Dessa uppgifter övertas av uppdragsgivaren.

Strategi och metodval

Vi har valt en deduktiv ansats till projektet där vi går från det generella till det specifika. En helhetsbild över systemet skapas med en syn på övergripande funktionalitet. Vidare ökar vi gradvis detaljnivån mot det mer specifika för att lättare förstå och kunna skapa varje komponent i systemet. Under den här delen av arbetet väljs utvecklingsmiljö och programspråk som av projektgruppen anses lämplig och som faller inom ramen för avgränsningarna. För att vi skall vara väl förberedda för eventuella problem som kan uppstå har vi identifierat de risker som kan uppkomma under projektets gång (se Appendix B).

Under hela projektets gång sker en kontinuerlig kommunikation och verifiering av arbetet med uppdragsgivaren och HTU:s handledare. Projektet kommer också löpande att dokumenteras och tidsredovisas.

Den strategi vi har valt för att realisera projektet är processororienterad och vi har delat in arbetet i fem faser. Dessa faser är analys, design, realisering, testning och leverans.

- **Analys** – Här undersöker vi hur Optinet Data AB arbetar idag och vilka funktioner som idag finns i systemet. Vidare skall vi se på vilka nya funktioner som skall skapas för att utöka systemet enligt uppdragsgivarens önskemål.
- **Design** – Framtagande av databasstruktur med relevanta dokument såsom objektbeskrivning med relationer samt objekt/term katalog. Vi skapar också en namngivningsstandard för variabler, funktioner och filnamn. Efter det skapas en beskrivning innehållande systemets webbsidor tillsammans med hur navigering sker samt av webbsidornas funktionalitet. Designfasen är användarorienterad och innan realisering verifieras designen med uppdragsgivaren.
- **Realisering** – Med stöd från beskrivningen om databasens uppbyggnad skapas ett SQL-skript för en enkel implementering av databasen. Vidare skapas utifrån designen av systemet en webbplats med användargränssnitt samt nödvändiga databaskopplingar. Realiseringen kommer att ske på den dator vi fått tilldelat oss hos Optinet Data AB samt även på HTU. Mindre testning av enskilda komponenter sker under arbetets gång. Dokumentation från den här fasen kommer vara dokumentering av skript, trädstruktur, filbeskrivningar och funktionsbeskrivningar.
- **Testning** – Då realiseringen är färdig kommer systemet att flyttas över till Optinet Data AB:s server och testas grundligt samt att eventuella buggar fixas.

- Leverans – Systemet kommer att implementeras hos uppdragsgivaren under projektets gång. Dessutom kommer en CD-skiva att levereras med dokumentation om databas, webbgränssnitt, funktioner och installation.

Översiktlig aktivitets- och tidsplan

Här redovisas huvudpunkterna för projektet och när i tiden aktiviteterna beräknas bli utförda. Projektet kommer officiellt att pågå från 17 mars till 20 maj. För mer översiktlig tidsplanering (se Gantt-schema, Bilaga 1).

- Vecka 12 Projektplan till arbetsgivare
- Vecka 12 - 13 Start av systemering
- Vecka 13 Realisering av databas
- Vecka 14 – 18 Utveckling av webbgränssnitt
- Vecka 16 – 17 Skapande av databaskopplingar
- Vecka 17 (21 april) Halvtidsinlämning, HTU
- Vecka 17 – 18 Skapande av sökfunktioner
- Vecka 19 Koppling till externa program
- Vecka 20 – 21 Testning
- Vecka 21 Leverans
- Vecka 21 (20 maj) Inlämning till första uppsatspresentationen
- Vecka 22 (26 – 28 maj) Seminarium med uppsatspresentation

Projektbudget

I projektet finns budgeterat 400 arbetstimmar per projektdeltagare, sammanlagt 800 timmar.

Inga kostnader finns för hårdvara. Optinet Data AB har ställt en arbetsdator till förfogande på företaget samt att det på HTU finns tillgång till datorer med utvecklingsverktyg. HTU har även bärbara datorer till utlåning om behov finnes.

Enligt överenskommelse med Rune Wallgren så har ekonomisk kompensation beviljats för reskostnader till och från Göteborg.

Projektorganisation

- Uppdragsgivare: Rune Wallgren, Optinet Data AB, VD
rune@optinet.se, 031 - 26 55 60, 0708 - 64 09 97
- Examinator: Lena Pareto, HTU
lenna.pareto@htu.se, 0520 - 47 51 19
- Handledare: Linn Gustavsson, HTU
linn.gustavsson@htu.se, 0520 - 47 51 60
- Referensgrupp: Rune Wallgren, Optinet Data AB, VD
Jan Mellqvist, Optinet Data AB, SQL Server ansvarig
janne@optinet.se, 031 - 26 55 60, 0708 - 64 09 90
Andreas Boklund, HTU
andreas.boklund@htu.se, 0520 - 41 51 77
Jonas Amoson, HTU
jonas.amoson@htu.se, 0520 - 47 53 27
Linn Gustavsson, HTU

Information och inflytande frågor

Under projektets gång kommer projektgruppen att ha kontakt med uppdragsgivare, examinator, handledare och referensgrupp.

Förväntad slutprodukt

Projektet kommer att utmynna i ett webbaserat system för hantering av offerter och kundkontakter. Varje säljare kommer att kunna skapa och hantera versioner av offerter, lagra information om tjänsternas värde, boka kundbesök och hantera kundkontakt. Systemet kommer att innehålla sökfunktioner för att kunna ta fram information om tagna/tappade offerter, deras värde, samt antalet bokade kundbesök. Informationen skall vara sökbar både på företagsnivå och per försäljare samt att sökningen skall kunna tidsbegränsas.

Förväntad effekt

Systemet kommer att ge Optinet Data AB en ökad kontroll av offerter, kundbesök och möjlighet att kunna ta fram väsentliga nyckeltal för att kunna målstyra sin verksamhet. Anställda skall ha möjlighet att komma åt systemet både inom företaget och utifrån. Ytterligare en vinst är att flera anställda har möjlighet att samtidigt arbeta med systemet.

Bilaga 1 - Gantt-schema

Gantt-schema		Vecka																
		7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
Delmoment	Projektbeskrivning till HTU																	
	Projektdirektiv till arbetsgivare																	
	Projektplan till arbetsgivare																	
	Systemering																	
	Realisering av databas																	
	Utveckling av webbgränssnitt																	
	Skapande av databaskopplingar																	
	Sök- och Filtreringsfunktioner																	
	Koppling till externa program																	
	Testning																	
	Leverans																	
	Inlämning av preliminärt utkast																	
	Inlämning av examensarbete																	
	Redovisning av examensarbete																	
Dokumentation																		

Bilaga 2 – Riskanalys

Id	Risk	Sannolikhet	Konsekvens	Riskvärde	Åtgärd	Nytt riskvärde
1	Sjukdom	2	3	6	Omfördela arbetet	1 x 3 = 3
2	Hårdvaruproblem	1	4	4	Kontinuerlig backup	1 x 4 = 4
3	Mjukvaruproblem	3	3	9	Kontinuerlig backup, ominstallation	1 x 3 = 2
4	Webbgränssnitt	1	4	4	Inläsning av referenslitteratur	1 x 4 = 4
5	Programmering	4	5	20	Inläsning av referenslitteratur	2 x 5 = 10
6	Webb-server	1	5	5	Inläsning av referenslitteratur, kontakter	1 x 5 = 5
7	SQL-server	2	5	10	Inläsning av referenslitteratur, SQL- ansvarig hos uppdragsgivare	1 x 5 = 5
8	Integrering med externa applikationer	3	3	9	Inläsning	2 x 3 = 6
9	Projektgenomförande	2	4	8	Skapa tidsplanering och gör uppföljning	1 x 4 = 4
10	Risk för omarbetning	3	4	12	Täta kontakter med uppdragsgivare	2 x 4 = 8
11	Uppdragsgivare svår att nå	2	3	6	Framförhållning, boka möte i tid	1 x 3 = 3
12	Handledare och examinator, HTU svåra att nå	3	3	9	Framförhållning, boka möte i tid	2 x 3 = 6
13	Datortillgång	2	4	8	Planera behovet av datorer	1 x 4 = 4
14	Transportproblem	2	2	4	Planera resandet, åka tåg	1 x 2 = 4

Appendix C Offertregister

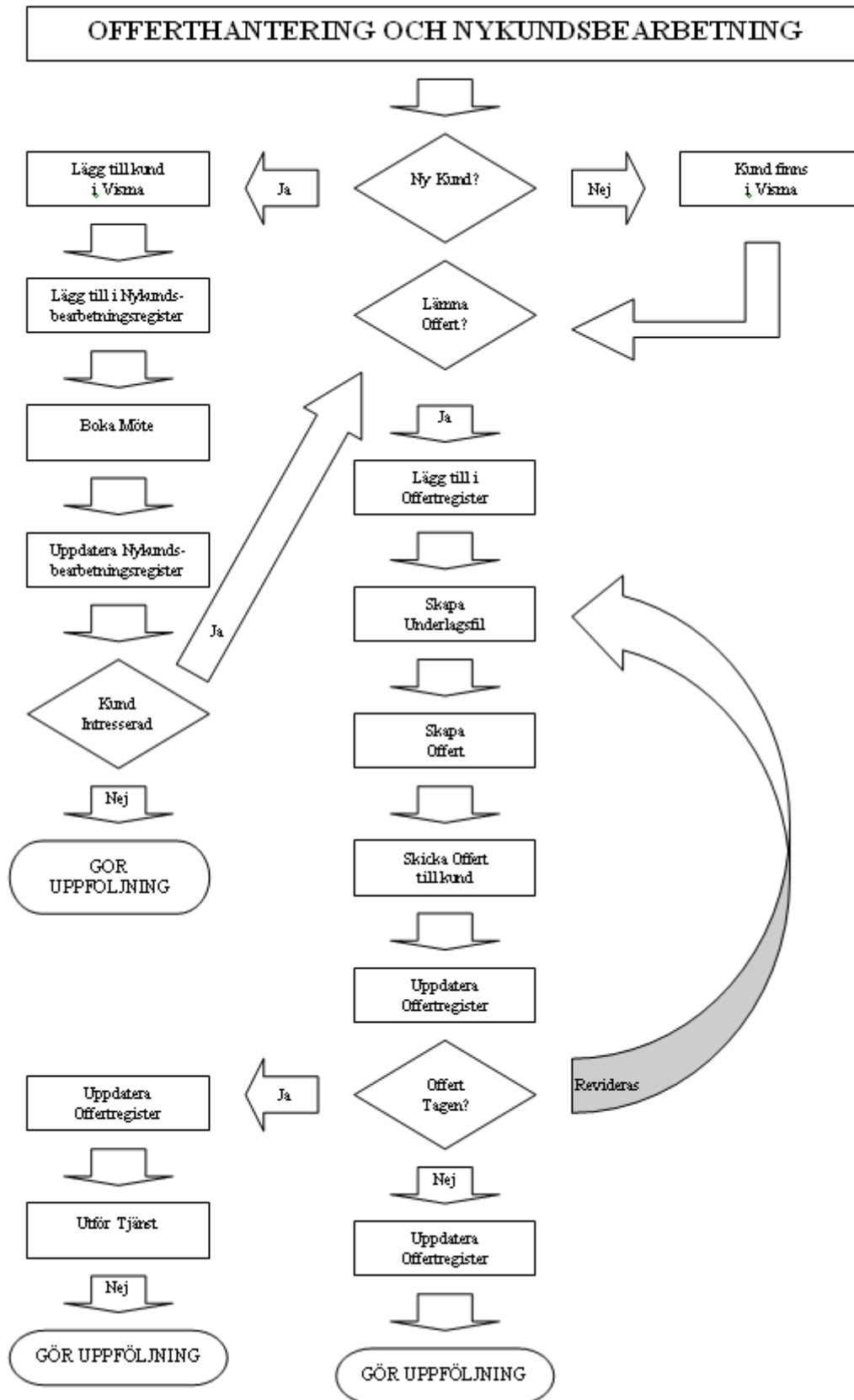
Löpnr.	Säljare	Kund	Avsände	Total belopp	TB	Tjänster	lämnad datum	går ut datum	tagen/tappad
731	prs		Driftsäker			122 250	2002-06-12	2002-06-22	tappad
732	prs		Virussäker			75 030	2002-05-30	2002-08-20	tappad
733	prs		Virussäker	49 740	49 740		2002-05-22	2002-06-15	tappad

Offertfil (insert, hyperlink)	Underlags fil (Ctrl+K)
OFFERTER\...doc	OFFERTER\...driftsäker 020514.xls
OFFERTER\...virusssäker.doc	
OFFERTER\...20020522.doc	

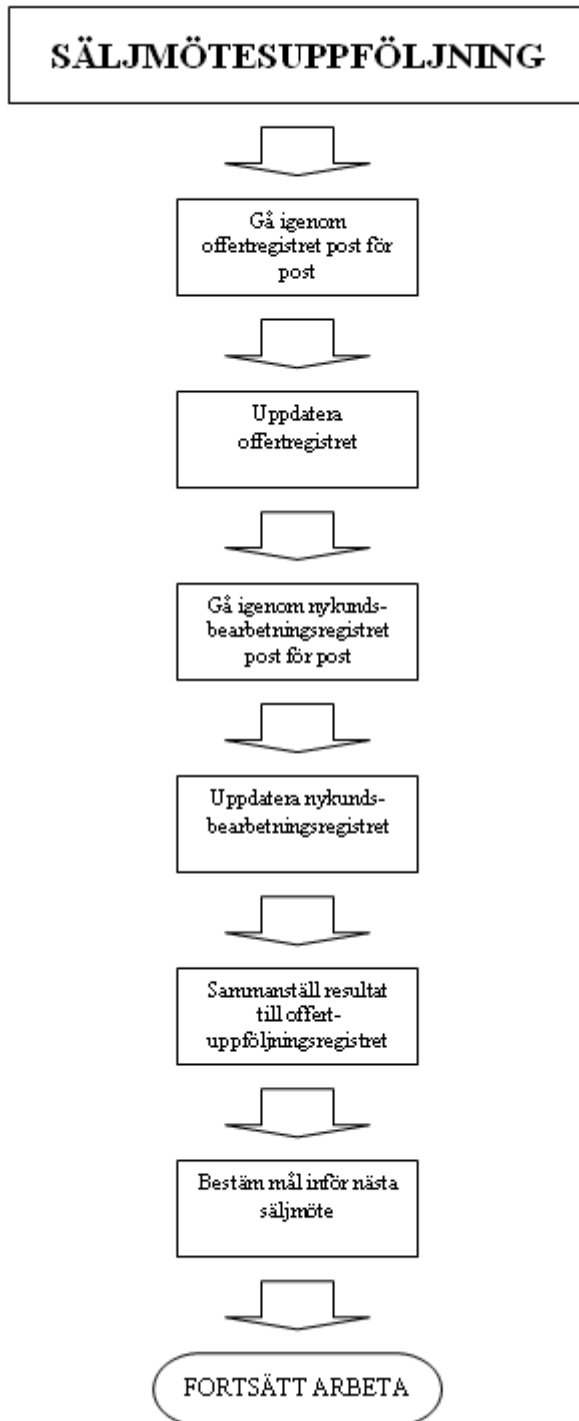
Appendix D Nykundsbearbetningsregister

Nykundsbearbetning 2002									
Kund	Säljare	bokat datum	besöksdatum	återb. datum	Intresserad av	Offert	I bearbetningsregister	Uppföljning	
	Prs	2002-08-23	2002-09-10	2004-03-09	Säkerhetsanalys	Ja	Ja	Ringa början mars	
	Prs	2002-09-04	2002-10-03		Drift		Ja		
	Tw	2002-09-11	2003-01-23		Allmänt	Nej	Ja		

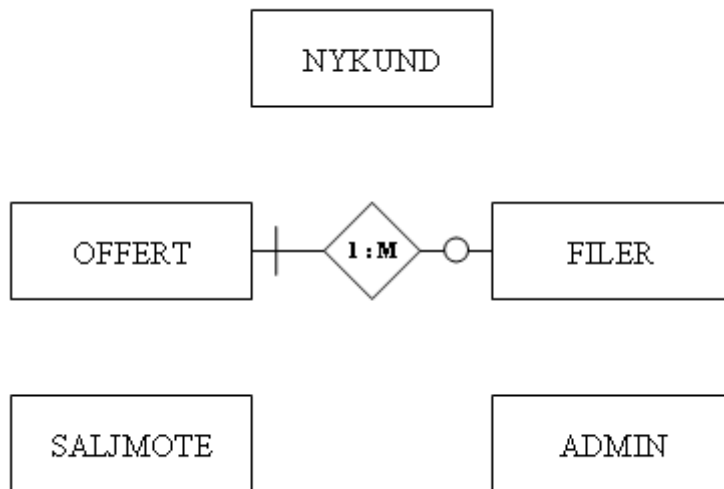
Appendix E Flödesschema nuvarande arbetsgång



Appendix F Flödesschema nuvarande arbetsgång



Appendix G Entitets/rerelationsdiagram



Appendix H Objekt/termkatalog

Objekt: Offert
Beskrivning: Lagrar information om offerter
Poststorlek: 294 byte

Nyckel	Fältnamn	Datotyp	Storlek	Ej null	Beskrivning
Primär	id_lopnr	Int	4	X	Autouppräknande primärnyckel
	kundnr	Int	4	X	Kundnummer
	avseende	Varchar	100		Vilken produkt/tjänst avser offerten?
	kommentar	Varchar	100		Kommentar om offerten
	saljare	Varchar	30		Vem är ansvarig säljare?
	referens	Varchar	30		Företagets referens, hämtas från Visma
	lamnad	Smalldatetime	4		Datum då offerten lämnas till kund
	giltig	Smalldatetime	4		Slutdatum för offertens giltighetstid
	tagen	Bit	1		Är offerten tagen?
	tappad	Bit	1		Är offerten tappad?
	besvarad	Smalldatetime	4		Datomet offerten besvarades
	totalbelopp	Int	4		Totalsumma för försäljning
	tb	Int	4		Täckningsbidrag, försäljningens vinst
	tjanst	Int	4		Löpande månadsinkomst

Objekt: Filer
Beskrivning: Hanterar offerternas alla filer
Poststorlek: 100 224 byte

Nyckel	Fältnamn	Datotyp	Storlek	Ej null	Beskrivning
Primär	id_fil	Int	4	X	Autouppräknande primärnyckel
Främmande	fid_lopnr	Int	4	X	Främmande nyckel från Offert tabellen
	benamning	Varchar	50		Filens avseende
	skapad	Smalldatetime	4		Datomet då filen skapades
	underlag	Bit	1		Anger att filen är ett underlag
	offert	Bit	1		Anger att filen är en offert
	avtal	Bit	1		Anger att filen är ett avtal
	ovrig	Bit	1		Anger om filen inte är någon av de tre ovanstående
	version	Smallint	2		Filens version
	fil_uppladdad	Bit	1		Markeras automatiskt då fil skapas från webbgränssnitt
	filnamn	Varchar	100		Namnet på filen
	arkiverad	Bit	1		Anger om filen är arkiverad, filnamn finns kvar för att kunna hitta filen
	fil	Image	~100 000		Medelstorlek på uppladdad fil
	contenttyp	Varchar	50		Filens filtyp
	fillangd	Int	4		Filens storlek

Systemintegration i praktiken
Utveckling av ett offert- och kundhanteringssystem i ASP.NET

Objekt: Nykund
Beskrivning: Hanterar mötesbokningar och uppföljning av nya kunder
Poststorlek: 383 byte

Nyckel	Fältnamn	Datotyp	Storlek	Ej null	Beskrivning
Primär	id_nykund	Int	4	X	Autouppräknande primärnyckel
	kundnr	Int	4	X	Kundnummer
	besok	Smalldatetime	4		Datum för besöket
	bokat	Smalldatetime	4		Datum då besöket bokades
	bokat_av	Varchar	30		Namn på den anställda som bokade besöket
	bokat_for	Varchar	30		Namn på den anställda mötet är avsett för
	avseende	Varchar	100		Produkt/tjänst kunden är intresserad av
	minnesant	Varchar	100		Anteckning om mötet
	uppfoljning	bit	1		Lagrar om uppföljning ska ske
	upp_nar	Varchar	5		Vilket kvartal ska kunden följas upp, Q1-Q4 eller ingen uppföljning?
	uppf_text	Varchar	100		Vad gäller uppföljningen?
	saljmote	bit	1		Ska post visas på företagets säljmöte?

Objekt: Säljmote
Beskrivning: Lagrar sammanräkningar av offertdata samt mål, används endast vid säljmöte
Poststorlek: 64 byte

Nyckel	Fältnamn	Datotyp	Storlek	Ej null	Beskrivning
Primär	id_saljmote	Int	4	X	Autouppräknande primärnyckel
	fid_nykund	Int	4	X	Senaste värdet från tabellen nykund
	fid_offert	Int	4	X	Senaste värdet från tabellen offert.
	mote	Smalldatetime	4	X	Säljmötets datum
	sum_totalbelopp	Int	4	X	Summa försäljning
	sum_tb	Int	4	X	Summa täckningsbidrag
	sum_tjanst	Int	4	X	Summa tjänst
	ant_tagna	Int	4	X	Antal tagna offerter
	ant_tappade	Int	4	X	Antal tappade offerter
	ant_lamnade	Int	4	X	Antal lämnade offerter
	ant_gen_moten	Int	4	X	Antal genomförda möten
	ant_upp_moten	Int	4	X	Antal kunder för uppföljning
	ant_kom_moten	Int	4	X	Antal kommande möten
	ant_bok_moten	Int	4	X	Antal bokade möten.
	mal_totalbelopp	Int	4		Mål för totalbelopp
	mal_tb	Int	4		Mål för täckningsbidrag

Systemintegration i praktiken
Utveckling av ett offert- och kundhanteringssystem i ASP.NET

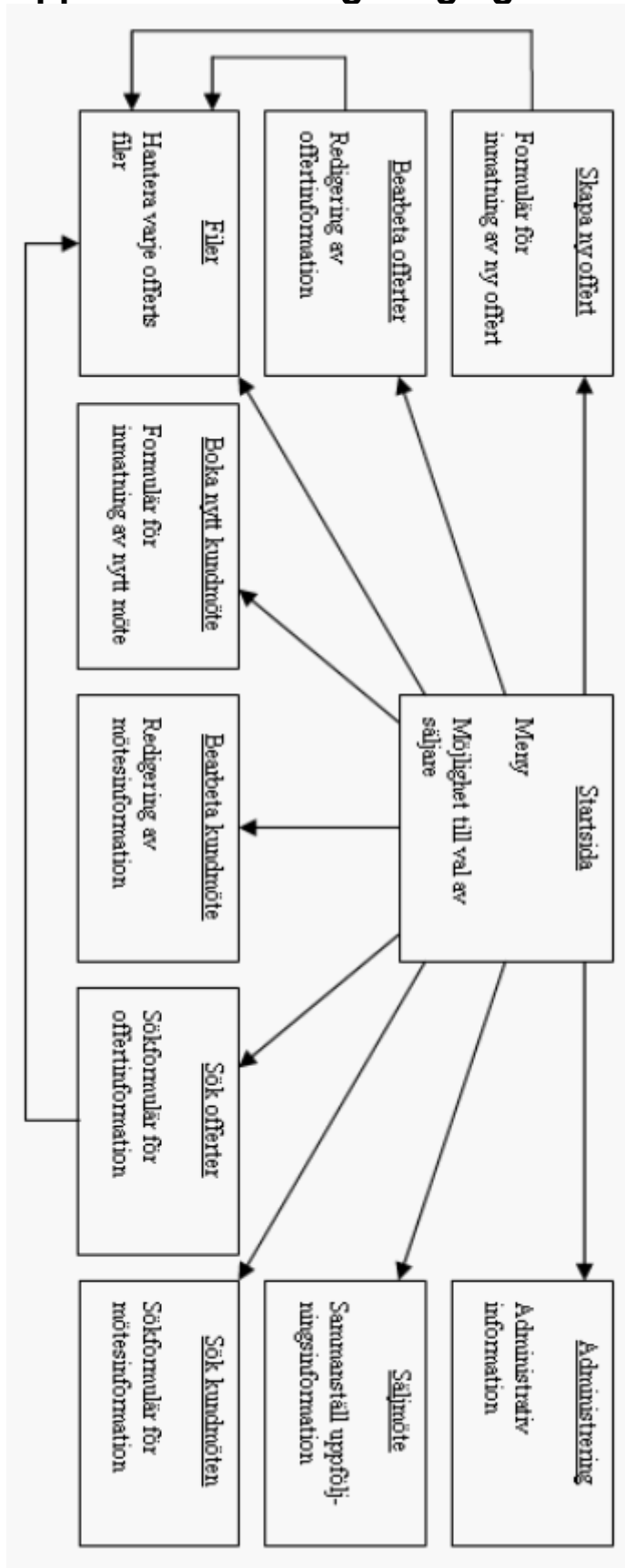
Objekt: Admin
Beskrivning: Innehåller systemets administrativa information
Poststorlek: 404 byte

Nyckel	Fältnamn	Datotyp	Storlek	Ej null	Beskrivning
Primär	id_admin	Int	4	X	Autouppräknande primärnyckel
	sokvag_underlag	Varchar	100		Vart underlagsmallar finns
	sokvag_offert	Varchar	100		Vart offertmallar finns
	sokvag_avtal	Varchar	100		Vart avtalsmallar finns
	sokvag_arkivering	Varchar	100		Vart filer ska arkiveras

Sammanräkning av databasens totala storlek efter ett år:

Tabellnamn	Antal poster (st)	Poststorlek (byte)	Tabellstorlek (byte)	Kommentar
Nykund	100	383	38 300	Antal nya kunder
Offert	200	294	58 800	Antal nya offerter
Filer	1 000	100 224	100 224 000	Fem filer per offert gånger antal offerter
Saljmote	24	64	1 536	Antal säljmöten
Admin	1	404	404	Administrativ information
Summa			100 323 040	

Appendix I Navigering i gränssnitt



Appendix J Namnstandard

En konsekvent namngivning är en viktig del vid programmering och ett antal olika namnstandarder har skapats för att underlätta detta.

Två ofta förekommande begrepp inom namnstandard är *Pascal Case* och *Camel Case* [1]. *Pascal Case* innebär att första bokstaven i identifieraren och första bokstaven i varje sammansatt efterföljande ord är skrivet med versaler. T.ex. NyttForm. Med *Camel Case* skrivs den första bokstaven i identifieraren med gemener och första bokstaven av efterföljande ord med versaler.

En frekvent använd standard är *Ungersk notation* [2], vilken hjälper till vid namngivning av variabler, funktioner, metoder, konstanter, m.m. Det innebär att man på ett enkelt sätt skall kunna bestämma typen och användningen av t.ex. en variabel genom att undersöka namnet. Ett eller flera tecken läggs till som ett prefix till identifierarens namn för att enkelt bestämma dess typ. Ett exempel är ett formulär med namnet *Sökning* vilket skulle få namnet *frmSearch*.

Den namnstandard det här projektet kommer att följa är baserad på följande riktlinjer:

Språk:

För att underlätta för andra programmerare kommer namngivning ske på engelska.

Kommentarer:

Koden skall kommenteras väl och alla kommentarer skall ske på svenska

Indentering:

All programkod skall indenteras väl för ökad läsbarhet

Variabler:

Använd förklarande namn med *Ungersk notation* och *Camel Case*. Specifikation av prefix:

- int – integer
- str – string
- bln – boolean
- arr – array
- ssn – session
- byt - byte

Instanser

Använd förklarande namn med *Ungersk notation* och *Camel Case*. Använd prefixet obj för följande instanser:

- SqlClient.SqlConnection
- SqlClient.SqlCommand
- Word.ApplicationClass
- Word.Document
- Data.Dataset
- Data.DataRow
- SqlClient.SqlParameter
- SqlClient.SqlDataReader

Komponenter (ASP.NET och HTML)

Använd förklarande namn med *Ungersk notation* och *Camel Case*. De komponenter som används i koden skall namnges. Komponenter som initieras via Visual Studio .NET gränssnittet namnges inte. Specifikation av prefix:

- txt – textbox
- dg – datagrid
- lbl – label
- btn – button
- ddl – dropdownlist
- lit – literal
- id – komponentens id-värde
- pnl - panel

Subrutiner och funktioner

Använd förklarande namn med *Pascal Case*.

Källförteckning (Namnstandard):

- [1] MSDN – Capitalization styles, (Microsoft),
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpgenref/html/cpconparameterNamingGuidelines.asp>, 2003-03-23.
- [2] MSDN – Hungarian Notation, (Charles Simonyi),
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnvsngen/html/hunganotat.asp>, 2003-03-23.

Appendix K SQL-skript

```
--- Ta bort innehållet i tabellerna ---
--- OBS! Finns inte tabellerna ska den här sektionen tas bort ---

DELETE FROM Saljmote
DELETE FROM Nykund
DELETE FROM Filer
DELETE FROM Offert
DELETE FROM Admin

--- Ta bort tabellerna ---
--- OBS! Finns inte tabellerna ska den här sektionen tas bort ---

DROP TABLE Saljmote
DROP TABLE Filer
DROP TABLE Nykund
DROP TABLE Offert
DROP TABLE Admin

--- Skapa tabellerna Nykund, Offert, Filer, Saljmote och Admin ---

CREATE TABLE Nykund (
  id_nykund    int          NOT NULL IDENTITY(1,1) PRIMARY KEY,
  kundnr      int          NOT NULL ,
  besok       smalldatetime NULL ,
  bokat       smalldatetime NULL ,
  bokat_av    varchar (30)  NULL ,
  bokat_for   varchar      (30) NULL ,
  avseende    varchar (100)  NULL ,
  minnesant   varchar (100)  NULL ,
  uppf_nar    varchar (5)    NULL ,
  uppf_text   varchar (100)  NULL ,
  uppfoljning bit          NULL ,
  saljmote    bit          NULL )

CREATE TABLE Offert (
  id_lopnr    int          NOT NULL IDENTITY(1000,1) PRIMARY KEY,
  kundnr      int          NOT NULL ,
  avseende    varchar (100)  NULL ,
  kommentar   varchar (100)  NULL ,
  saljare     varchar (30)    NULL ,
  referens    varchar (30)    NULL ,
  lamnad      smalldatetime  NULL ,
  giltig      smalldatetime  NULL ,
  besvarad    smalldatetime  NULL ,
  tagen       bit          NULL ,
  tappad      bit          NULL ,
  totalbelopp int          NULL ,
  tb          int          NULL ,
  tjanst      int          NULL )

CREATE TABLE Filer (
  id_fil      int          NOT NULL IDENTITY(1,1) PRIMARY KEY,
```

Systemintegration i praktiken
Utveckling av ett offert- och kundhanteringssystem i ASP.NET

```
fid_lopnr int NOT NULL FOREIGN KEY REFERENCES Offert (id_lopnr),
benamning varchar (50) NULL ,
skapad smalldatETIME NULL ,
underlag bit NULL ,
offert bit NULL ,
avtal bit NULL ,
ovrig bit NULL ,
version smallint NULL ,
fil_uppladdad bit NULL ,
filnamn varchar (100) NULL ,
arkiverad bit NULL ,
fil image NULL ,
contenttyp varchar (50) NULL ,
fillangd int NULL )
```

```
CREATE TABLE Saljmote (
  id_saljmote int NOT NULL IDENTITY(1,1) PRIMARY KEY,
  fid_nykund int NOT NULL ,
  fid_offert int NOT NULL ,
  mote smalldatETIME NOT NULL ,
  sum_totalbelopp int NOT NULL ,
  sum_tb int NOT NULL ,
  sum_tjanst int NOT NULL ,
  ant_tagna smallint NOT NULL ,
  ant_tappade smallint NOT NULL ,
  ant_lamnade smallint NOT NULL ,
  ant_gen_moten smallint NOT NULL ,
  ant_upp_moten smallint NOT NULL ,
  ant_kom_moten smallint NOT NULL ,
  ant_bok_moten smallint NOT NULL ,
  mal_totalbelopp int NULL ,
  mal_tb int NULL ,
  mal_tjanst int NULL ,
  mal_bokade int NULL )
```

--- Ska finnas för att säljmötesfunktionen ska fungera första gången ---

```
INSERT INTO saljmote (fid_nykund, fid_offert, mote) VALUES (0, 0, '2000-01-01')
```

```
CREATE TABLE Admin (
  id_admin int NOT NULL IDENTITY(1,1) PRIMARY KEY,
  sokvag_underlag varchar (100) NULL ,
  sokvag_offert varchar (100) NULL ,
  sokvag_avtal varchar (100) NULL ,
  sokvag_arkivering varchar (100) NULL )
```

Appendix L Datagrid

Löppnr: 1005	Kund: 100592 A+A Corporate Design AB
Säljare: Rune Wallgren	Referens: Anders Andersson
Avsende: Serverköp	
Kommentar: Är nog också intresserad av Säker Internet	
Lämnad: 2003-06-01	Giltig: 2003-06-21
Tagen: <input checked="" type="checkbox"/> Tappad: <input type="checkbox"/>	Besvarad: 2003-06-15
Totalbelopp: 24000	Täckningsbidrag: 4500 Tjänst:

Löppnr: 1006	Kund: 100003 Zimmer Scandinavia AB
Säljare: <input type="text" value="Rune Wallgren"/> <input type="button" value="▼"/>	Referens: <input type="text" value="Per Persson"/>
Avsende: <input type="text" value="Säker Internet"/>	
Kommentar: <input type="text"/>	
Lämnad (YYYY-MM-DD): <input type="text" value="2003-07-15"/>	Giltig (YYYY-MM-DD): <input type="text" value="2003-09-05"/>
Tagen: <input type="checkbox"/> Tappad: <input type="checkbox"/>	Besvarad: <input type="text"/>
Totalbelopp: <input type="text"/>	Täckningsbidrag: <input type="text"/> Tjänst: <input type="text"/>

<input type="button" value="Uppdatera"/>	<input type="button" value="Avbryt"/>	<input type="button" value="Ta bort"/>	<input type="button" value="Filer"/>
--	---------------------------------------	--	--------------------------------------

<input type="button" value="Ändra"/>	<input type="button" value="Ta bort"/>	<input type="button" value="Filer"/>
--------------------------------------	--	--------------------------------------

Appendix M Systemdokumentation

Systemdokumentationen är indelad i olika delar. Projektets namn är Exjobb och här specificerar vi nu de filer som ingår i projektet samt alla de funktioner som finns i dessa filer tillsammans med en beskrivning.

Installationsanvisning

För att Systemet skall fungera finns en del krav på programvara som måste vara installerad på webbservern.

- IIS 5.0
- Microsoft .NET Framework >= 1.0
<http://asp.net/download-1.1.aspx?tabindex=0&tabid=1>
- Microsoft Word (Office XP)
- MS SQL Server 2000

Försäkra er om att .NET Framework är korrekt registrerat genom att köra kommandot "**C:\WINNT\Microsoft.NET\Framework\version\aspnet_regiis.exe**" -i

En mindre modifiering av IIS är dessutom nödvändig, välj Default Web Site Properties - > Home Directory och ändra Application Protection till Low (IIS Process). Anledningen till detta är att skapandet av Word dokument skall fungera problemfritt.

Vid installationen av .NET Framework, skapas en användare vid namn ASP.NET. För att interaktionen mellan Säljman och Word skall fungera, krävs att ASP.NET användaren har tillräckliga rättigheter. En lösning är att skapa en grupp med dessa rättigheter för användaren.

En databas vid namn Exjobb behöver skapas i SQL Server 2000, därefter skall det bifogade SQL-skriptet Exjobb_sql_skript.sql köras. Skriptet skapar automatiskt nödvändiga tabeller och poster.

För att större filer än 4MB skall kunna laddas upp i databasen, krävs att man ändrar största tillåtna filstorlek på variabeln `maxRequestLength` i `C:\WINNT\Microsoft.NET\Framework\version\CONFIG\machine.config`

Systemet använder sig av mallar (underlags-, offert- och avtalsmallar), och förslagsvis bör tre olika kataloger skapas på webbservern innehållande dessa tre olika typer av mallar. En avtalsmall och en underlagsmall är helt enkelt en Word- och en Excel-fil som kan utnyttjas som grund för offerter. Gällande offertmallar så krävs det förberedande arbete. En offertmall är en Word mall (.dot) som i princip kan se ut hur som helst, t.ex. med Optinets logotyp m.m. Dock måste mallen innehålla följande bokmärken: företagsnamn, referens, säljare, offertdatum, och slutdatum. Skapa ett bokmärke på följande sätt: placera markören där bokmärket skall finnas i dokumentet, välj sedan Tools -> Bookmark och skriv in namnet på bokmärket (se ovan) och välj Add. Ett bokmärke har nu skapats med angivet namn. En mall kan innehålla samma bokmärke flera gånger.

Med systemet medföljer en simpel mall, template1.dot, vilken kan användas för referens om så önskas.

Slutligen kopieras mappen Exjobb vilken innehåller alla nödvändiga filer av Systemet till lämplig plats på webbservern. Det första som bör göras när allt är installerat är att ändra sökvägarna för de olika mallarna. Detta görs via Administreringssidan. Som standard är alla sökvägar satta till C:\, men det skall ändras till katalogerna där mallarna finns. Dessutom bör sökvägen för arkiveringskatalog ändras, vilken är där filer kommer att flyttas till från databasen när arkivering önskas.

Filer inom projektet Exjobb

Alla sidor i projektet där kodning av gränssnitt och funktionalitet sker består av två filer, en .aspx fil där all HTML och ASP.NET-kod finns och en .aspx.vb fil där all VB.NET kod finns. Följande sidor ingår i projektet Exjobb (Där extension ej är angiven finns både .aspx och .aspx.vb filer. Namnet inom parentes anger namnet på länken från menyn):

- **Admin** (Administrering) – Innehåller enklare administrering såsom arkivering och inställningar av sökvägar för underlags-, offert- och avtalsmallar.
- **Center** (Startsida) – Startsidan där användaren har möjlighet att logga in i systemet.
- **File_update** – Sköter uppladdning av filer till databasen. Hanterar underlag, offerter och avtal. Endast de filer som är skapade via "tender_files", nedladdade till användarens dator och post är skapad i databasen. För att komma till den här sidan finns länk från "Tender_files"
- **File_upload_ovrig** – Sköter uppladdningen av övriga filer till databasen. För att komma till den här sidan finns länk från "Tender_files"
- **Functions** – Innehåller de funktioner systemet använder sig av som inte måste ligga på den sidan de anropas ifrån (Hanterar funktionen komponenter finns funktionen på respektive sida där komponenten finns. .
- **Index.htm** – Startsida för projektet Exjobb. Laddar in ett ramverk bestående av tre ramar där sidorna left, top och center laddas in initialt.
- **Left** – Innehåller menyn för navigering inom systemet och finns alltid framme. Här visas också namnet på den säljare som är inloggad i systemet.
- **Meeting_info** (Bearbeta nykundsmöten) – Innehåller formulär för det dagliga arbetet med att hantera och redigera information rörande kundmöten.
- **New_meeting** (Boka nykundsmöte) – Innehåller formulär för bokningar av kundmöten.

- **New_tender** (Skapa offert) - Innehåller formulär där offerter kan skapas. Då offert är skapad kan användaren via knappen "Skapa fil" navigera till "tender_files".
- **Sale_meeting** (Säljmöte) – Innehåller funktioner för att beräkna verksamhetens nyckeltal och bestämma mål.
- **Seek_meeting** (Sök kundsmöten) – Innehåller sökformulär för framtagning av nykundsmöten. Används främst under säljmötet.
- **Seek_tender** (Sök offerter) – Innehåller sökformulär för framtagning av offerter. Används främst under säljmötet.
- **Styles.css** – Är "Cascading Style Sheet" och styr utseendet på systemet så att det blir enhetligt.
- **Tender_files** (Offertfiler) – Här sker skapandet av offertfiler och visning av de filer som är skapade för en specifik offert. Härifrån kan användaren via knappen "Ladda upp fil" navigera till sidan "File_update" för att ladda upp en fil. Samt via knappen "Skapa övrig fil" navigera till sidan "File_upload_ovrig" där övriga filer kan laddas upp.
- **Tender_info** (Bearbeta offerter) – Innehåller formulär för det dagliga arbetet med att hantera och redigera information rörande offerter. Härifrån kan användaren navigera till "Tender_files" via knappen "Filer".
- **Top** – Översta sidan i ramverket. Innehåller Optinets logo.

Subrutiner och funktioner inom varje fil (aspx.vb)

Admin

- Page_Load – Tar fram lagrade sökvägar ur databasen.
- btnSaveSettings_Click – Anropar SavePath med sökvägar som argument.
- SavePath – Lagrar sökvägarna till databasen.

Center

- Page_Load – Lagrar sökvägar till mallar som sessionsvariabler. Hämtar säljares namn och visar till en rullgardinsmeny.
- ddlSeller_SelectedIndexChanged – Då säljare loggar in lagras namnet som sessionsvariabel och sidan "Left" laddas in igen.

File_update

- Page_Load – Sätter sessionsvariabler inför filuppladdning.

- DoUpload – Sköter filuppladdning till databasen.

File_upload_ovrig

- Page_Load – Visar text angående övrig fil på skärmen
- DoUpload – Sköter filuppladdning till databasen.

Functions.resx

- AddFile2DB – Laddar upp underlag, offerter och avtal till databasen "Filer". Anropas från "Tender_Files".
- DownloadFromDB – Laddar ner alla typer av filer från databasen till användaren. Anropas från "Tender_files".
- DownloadFile – Laddar ner underlags-, offert och avtalsmallar från webbservern till användaren. Anropas från "Tender_files".
- UpdateFile2DB – Laddar upp filer av typen övriga till databasen. Anropas från "File_update".
- CreateName – Hanterar automatisk namngivning och returnerar en sträng. Anropas från "Tender_files" och "File_upload_ovrig".
- Database_Select_Query_Visma – Skickar förfrågningar till Vismas databas och returnerar ett dataset. Anropas från "Center", "Meeting_info", "New_meeting", "New_tender", "Seek_meeting", "Seek_tender" och "Tender_info".
- Database_Select_Query_Exjobb – Skickar förfrågningar till databasen Exjobb och returnerar ett dataset. Anropas från "Meeting_info", "New_meeting", "Seek_meeting", "Seek_tender" och "Tender_info".
- FileExists – Kontrollerar om en fil är uppladdad till databasen och returnerar ett heltal. Anropas från "tender_files".
- GetHighestVersion – Tar fram det högsta versionsnumret av de filer som finns i databasen och returnerar ett heltal. Anropas från "Tender_files".
- GetDirPath – Tar fram sökvägar till alla mallar och returnerar en sträng. Anropas från "Tender_files".
- sSplit – Tar fram filnamn ur hela sökvägen och returnerar en sträng. Anropas från "File_update" och "File_upload_ovrig".
- VersionExist – Kontrollerar om filnamn finns i databasen och returnerar ett heltal. Anropas från "Tender_files".

Left

- Page_Load – Skriver ut inloggat namn.

Meeting_info

- Page_Load – Initierar formuläret.
- txtStartCustomerName_TextChanged – Hämtar nya kundnamn då kundens begynnelsebokstäver anges.
- txtCustomerNumber_TextChanged – Hämtar nytt kundnamn då kundnr anges.
- Fill_ddlCustomerName – Kopplar kundnamn och kundnummer till rullgardinsmeny.
- btnShow_Click – Kontrollerar inmatningen i formuläret och hämtar eftersökt information från databasen och kopplar till en datagrid.
- GetCustomerName – Hämtar kundnamnet från Visma för visning i datagriden. Det är endast kundnummer som lagras i databasen Exjobb.
- dgMeetingEdit – Gör den rad användaren klickade på i datagriden editierbar. Anropar datagridens inbyggda funktion "EditItemIndex". Anropar sedan funktionen BindData.
- dgMeetingCancel – Ångrar editering av rad i datagriden. Anropar sedan funktionen BindData.
- dgMeetingDeleteRow – Tar bort en rad ur datagriden. Anropar sedan funktionen BindData.
- dgMeetingUpdate – Uppdaterar den ändrade raden i datagriden till databasen. Anropar sedan funktionen BindData.
- Check – Anropas då poster listas i datagriden för felkontroll. Retrunerar en sträng tillbaka som skrivs ut.
- BindData – Binder data till datagriden då användaren klickar på "ändra", "uppdatera", "avbryt" eller "ta bort"
- GetSellers – Hämtar säljarnamn från Visma för visning i rullgardinsmeny i datagridens editläge.
- GetSelectedSeller – Hämtar den säljare som är vald för visning i datagrid.
- GetSourceUppf_nar – Skapar en array med värden för visning i rullgardinsmeny vid editering av datagriden.
- GetSelectedUppf_nar – Hämtar det värde för när uppföljning skall ske för att välja rätt index i rullgardinsmenyn.

- `cbxUpcoming_CheckedChanged` – Om checkboxen kommande markeras avmarkeras checkboxen för utförda möten.
- `cbxDone_CheckedChanged` - Om checkboxen utförda markeras avmarkeras checkboxen för kommande möten.
- `ComputeSum` – Beräknar summan av nyckelvärden för varje post och redovisas sist i datagriden.

New_meeting

- `Page_Load` – Initierar formuläret.
- `btnBook_Click` – Skapa mötespost i databasen för bokad kund.
- `calBookedDate_SelectionChanged` – Skriver valt datum till osynlig textbox för jämförelse med mötesdatum.
- `calMeetingDate_SelectionChanged` – Skriver valt datum till osynlig textbox för jämförelse med bokningsdatum.
- `txtStartCustomerName_TextChanged` – Hämtar nya kundnamn då kundens begynnelsebokstäver anges.
- `txtCustomerNumber_TextChanged` – Hämtar nytt kundnamn då kundnr anges.
- `Fill_ddlCustomerName` – Kopplar kundnamn och kundnummer till rullgardinsmeny.

New_tender

- `Page_Load` – Initierar formuläret.
- `btnCreateTender_Click` – Skapa offertpost i databasen offerter.
- `txtStartCustomerName_TextChanged` – Hämtar nya kundnamn då kundens begynnelsebokstäver anges.
- `txtCustomerNumber_TextChanged` – Hämtar nytt kundnamn då kundnr anges.
- `GetReference` – Hämtar namn på kundens referens beroende på vald kund i rullgardinsmenyn.
- `ddlCustomerName_SelectedIndexChanged` – Anropar `Getreference` varje gång ett namn väljs i rullgardinsmenyn.
- `Fill_ddlCustomerName` – Kopplar kundnamn och kundnummer till rullgardinsmeny.
- `btnCreateFile_Click` – Förflyttar användaren till finlen ”Tender_files”.

Sale_meeting

- Page_Load – Anropar ShowHistory för visning av datagrid.
- btnSalesMeeting_Click – Sammanräknar nyckeltal från det senaste mötet och visar på skärmen.
- btnSave_Click – Lagrar beräknade nyckeltal och inmatade mål till databasen.
- ShowHistory – Hämtar data från tabellen säljmöte och binder till en grid.
- ShowIndexedGrid – Hämtar dataset från tabellen säljmöte och returnerar till dgSalePagerChange.
- dgSalePagerChange – Anropas från datagriden och hanterar paging, d.v.s. visning av poster från tabellen säljmöte tio åt gången.

Seek_meeting

- Page_Load – Initierar formuläret.
- GetCustomerName – Hämtar kundnamnet från Visma för visning i datagriden. Det är endast kundnummer som lagras i databasen Exjobb.
- dgMeetingEdit – Gör den rad användaren klickade på i datagriden editierbar. Anropar datagridens inbyggda funktion "EditItemIndex". Anropar sedan funktionen BindData.
- dgMeetingCancel – Ångrar editering av rad i datagriden. Anropar sedan funktionen BindData.
- dgMeetingDeleteRow – Tar bort en rad ur datagriden. Anropar sedan funktionen BindData.
- dgMeetingUpdate – Uppdaterar den ändrade raden i datagriden till databasen. Anropar sedan funktionen BindData.
- Check – Anropas då poster listas i datagriden för felkontroll. Retrunerar en sträng tillbaka som skrivs ut.
- BindData – Binder data till datagriden då användaren klickar på "ändra", "uppdatera", "avbryt" eller "ta bort"
- GetSellers – Hämtar säljarnamn från Visma för visning i rullgardinsmeny i datagridens editläge.
- GetSelectedSeller – Hämtar den säljare som är vald för visning i datagrid.
- GetSourceUppf_nar – Skapar en array med värden för visning i rullgardinsmeny vid editering av datagriden.

- GetSelectedUppf_nar – Hämtar det värde för när uppföljning skall ske för att välja rätt index i rullgardinsmenyn.
- btnShow_Click – Kontrollerar inmatningen i formuläret och hämtar eftersökt information från databasen och kopplar till en datagrid.
- ComputeSum – Beräknar summan av nyckelvärden för varje post och redovisas sist i datagriden.

Seek_tender

- Page_Load – Initierar formuläret.
- Fill_ddlCustomerName – Kopplar kundnamn och kundnummer till rullgardinsmeny.
- txtStartCustomerName_TextChanged – Hämtar nya kundnamn då kundens begynnelsebokstäver anges.
- cbxOpen_CheckedChanged – Markerar och avmarkerar checkboxar i formuläret.
- cbxTaken_CheckedChanged – Markerar och avmarkerar checkboxar i formuläret.
- cbxNotTaken_CheckedChanged – Markerar och avmarkerar checkboxar i formuläret.
- cbxGiven_ Markerar och avmarkerar checkboxar i formuläret.CheckedChanged –
- cbxNotGiven_CheckedChanged – Markerar och avmarkerar checkboxar i formuläret.
- cbxClosed_CheckedChanged – Markerar och avmarkerar checkboxar i formuläret.
- cbxAll_CheckedChanged – Markerar och avmarkerar checkboxar i formuläret.
- btnShow_Click – Kontrollerar inmatningen i formuläret och hämtar eftersökt information från databasen och kopplar till en datagrid.
- Check – Anropas då poster listas i datagriden för felkontroll. Retrunerar en sträng tillbaka som skrivs ut.
- dgTenderEdit – Gör den rad användaren klickade på i datagriden editierbar. Anropar datagridens inbyggda funktion "EditItemIndex". Anropar sedan funktionen BindData.

- dgTenderCancel – Ångrar editering av rad i datagriden. Anropar sedan funktionen BindData.
- dgTenderDeleteRow – Tar bort en rad ur datagriden. Anropar sedan funktionen BindData.
- dgTenderUpdate – Uppdaterar den ändrade raden i datagriden till databasen. Anropar sedan funktionen BindData.
- BindData – Binder data till datagriden då användaren klickar på ”ändra”, ”uppdatera”, ”avbryt” eller ”ta bort”
- GetSellers – Hämtar säljarnamn från Visma för visning i rullgardinsmeny i datagridens editläge.
- GetSelectedSeller – Hämtar den säljare som är vald för visning i datagrid.
- GetCustomerName – Hämtar kundnamnet från Visma för visning i datagriden. Det är endast kundnummer som lagras i databasen Exjobb.
- ComputeSum – Beräknar summan av nyckelvärden för varje post och redovisas sist i datagriden.
- Files – Lagrar offertens löpnummer i en sessionsvariabel och skickar användaren till ”Tender_files”.

Tender_files

- GetLopNr – Hämtar och returnerar det senaste löpnumret från databasen offert.
- CreateOffert – Skapar offertfil utifrån mall.
- Page_Load – Läser in sökvägar till mallar.
- btnCreateOffert_Click – Skapar filnamn, kopplar den till vald offertmall och laddar ner den till användarens dator.
- btnCreateUnderlag_Click - Skapar filnamn, kopplar den till vald underlagsmall och laddar ner den till användarens dator.
- btnCreateAvtal_Click - Skapar filnamn, kopplar den till vald avtalsmall och laddar ner den till användarens dator.
- dgFilesDeleteRow – Tar bort en rad ur datagriden. Anropar sedan funktionen BindData.
- BindData – Binder data till datagriden då användaren klickar på ”ändra”, ”uppdatera”, ”avbryt” eller ”ta bort”
- GetFileType - Hämtar vilken typ av fil det är från Visma för visning i datagrid.

- GetUploaded – Kontrollerar om fil är uppladdad till databas för visning av ja eller nej i datagrid.
- GetArchived – Kontrollera om arkiveringsbiten är satt, returnera ja eller nej till datagriden.
- CheckUploaded – Kontrollera om fil är uppladdad till databas.
- GetName – Hämtar kundnamn från Visma för att kunna skriva det i offertmallen.
- GetTenderDate – Hämtar datumet då offerten lämnades för att kunna skriva det i offertmallen.
- GetReference – Hämtar kundens referens för att kunna skriva det i offertmallen.
- CheckUpOrDown – Kontrollerar om fil skall laddas upp till databasen eller ner till användaren. Överför antingen användaren till "File_update" eller kallar på funktionen DownloadFormDB.
- btnShow_Click – Visar filer för en viss offert i en datagrid.
- btnChooseAvtal_Click – Visar respektive döljer textbox och knapp för val av avtalsmall.
- btnChooseOffert_Click – Visar respektive döljer textbox och knapp för val av offertmall.
- btnChooseUnderlag_Click – Visar respektive döljer textbox och knapp för val av underlagsmall.
- btnCreateOvrig_Click – Laddar upp en övrig fil till databasen.

Tender_info

- Page_Load – Initierar formuläret.
- ddlCustomerName_SelectedIndexChanged – Anropar Getreference varje gång ett namn väljs i rullgardinsmenyn.
- ddlSeller_SelectedIndexChanged – Då säljare loggar in lagras namnet som sessionsvariabel och sidan "Left" laddas in igen.
- txtInvoiceNumber_TextChanged – Hämtar kundnamn och kundnummer som binds till en rullgardinsmeny då en offerts löpnummer skrivs in.
- cbxAll_CheckedChanged – Markerar och avmarkerar checkboxar i formuläret.
- cbxOpen_CheckedChanged – Markerar och avmarkerar checkboxar i formuläret.

- `cbxTaken_CheckedChanged` – Markerar och avmarkerar checkboxar i formuläret.
- `cbxNotTaken_CheckedChanged` – Markerar och avmarkerar checkboxar i formuläret.
- `cbxGiven_Markerar och avmarkerar checkboxar i formuläret.CheckedChanged` –
- `cbxNotGiven_CheckedChanged` – Markerar och avmarkerar checkboxar i formuläret.
- `cbxClosed_CheckedChanged` – Markerar och avmarkerar checkboxar i formuläret.
- `txtStartCustomerName_TextChanged` – Hämtar nya kundnamn då kundens begynnelsebokstäver anges.
- `txtCustomerNumber_TextChanged` – Hämtar nytt kundnamn då kundnr anges.
- `Fill_ddlCustomerName` – Kopplar kundnamn och kundnummer till rullgardinsmeny.
- `btnShow_Click` – Kontrollerar inmatningen i formuläret och hämtar eftersökt information från databasen och kopplar till en datagrid.
- `dgTenderEdit` – Gör den rad användaren klickade på i datagriden editierbar. Anropar datagridens inbyggda funktion `”EditItemIndex”`. Anropar sedan funktionen `BindData`.
- `dgTenderCancel` – Ångrar editering av rad i datagriden. Anropar sedan funktionen `BindData`.
- `dgTenderDeleteRow` – Tar bort en rad ur datagriden. Anropar sedan funktionen `BindData`.
- `Check` – Anropas då poster listas i datagriden för felkontroll. Retrunerar en sträng tillbaka som skrivs ut.
- `dgTenderUpdate` – Uppdaterar den ändrade raden i datagriden till databasen. Anropar sedan funktionen `BindData`.
- `BindData` – Binder data till datagriden då användaren klickar på `”ändra”`, `”uppdatera”`, `”avbryt”` eller `”ta bort”`
- `GetSellers` – Hämtar säljarnamn från Visma för visning i rullgardinsmeny i datagridens editläge.
- `GetSelectedSeller` – Hämtar den säljare som är vald för visning i datagrid.

- `GetCustomerName` – Hämtar kundnamnet från Visma för visning i datagriden. Det är endast kundnummer som lagras i databasen Exjobb.
- `ComputeSum` – Beräknar summan av nyckelvärden för varje post och redovisas sist i datagriden.
- `Files` – Lagrar offertens löpnummer i en sessionsvariabel och skickar användaren till "Tender_files".

Appendix N Programkod

Programkoden tillhörande projektet skickas endast med vid inlämning av rapporten till examinator och Optinet. Den skickas inte med till andra instanser eller får på annat vis publiceras eller distribueras. Anledningen till detta är för att skydda Optinet samt för att projektdeltagarna i framtiden skall kunna fortsätta utveckla och eventuellt sälja produkten.