

Når skaderne opdages

Skadesregistrering, ordre/sagsstyring samt
behandlingsregistrering af arkivgenstande i
Københavns Stadsarkiv

Projektrapport

Projektgruppe

Niels Grove-Rasmussen

Martin Trnka

Vejleder

Flemming Bislev



Niels Grove-Rasmussen, 36 år.

Påbegyndte datamatikeruddannelsen åben uddannelse 1999.

Uddannet folkeskolelærer med linjefag i matematik og fysik/kemi.

Ansæt som systemkonstruktør i ATP-huset.

Gift og far til to børn på henholdsvis 3 og 9 år.



Martin Trnka, 46 år.

Påbegyndte datamatikeruddannelsen åben uddannelse 1998.

Uddannet håndbogbinder med restaurering som speciale.

Ansæt som håndbogbinder i Københavns Stadsarkiv.

Gift og far til et ”barn” på 18 år.

Når skaderne opdages

Skadesregistrering, ordre/sagsstyring samt
behandlingsregistrering af arkivgenstande i
Københavns Stadsarkiv

Projektrapport

Datamatiker uddannelsen • 2003.12.19

5. semester • Hovedopgave • Eksamensprojekt

Niels Brock • Nørre Voldgade • København

Indholdsfortegnelse

Indledning	9
1 Oplæg og indledende bemærkninger	9
1.1 Oplæg og målsætning.....	9
1.2 Indledende bemærkninger.....	9
Projekt	11
2 Projektgrundlag	11
2.1 Projektoplæg	11
2.2 Formål	11
2.3 Forudsætninger.....	11
2.3.1 Teknologi.....	11
2.3.2 Økonomi	11
2.4 Mål	11
2.5 Projektorganisering	12
2.6 Projektgruppetstyring og -kommunikation	13
2.7 Hovedopgaverapport og systemdokumentation.....	13
2.8 Projektstrategi	14
2.9 Projektmodel og metoder	15
2.10 Risikoanalyse	16
2.11 Projektopdeling	17
2.12 Tidsplan.....	17
2.13 Faseopdeling, milestones, succeskriterier og tidsestimering	18
2.14 Projektets realisering i forhold til projektplanen.....	27
2.15 Kvalitet.....	28
2.16 Programanvendelse og Standarder.....	28
2.16.1 Programanvendelse.....	28
2.16.2 Kodestandarder	28
2.17 Afgrænsning.....	28
Virksomhed	29
3 Virksomhedsanalyse	29
3.1 Virksomhedsbeskrivelse	29
3.2 Organisationsstruktur	29
3.3 Forandringer som følge af implementering af IT (Leavitt).....	30
3.4 Interessentanalyse	31
3.5 Nyttевærdi og procesforbedring som følge af IT-systemet	32
System - foranalyse	33
4 Foranalyse	33
4.1 Projektopgaven.....	33
4.1.1 Anvendelsesområdet - eksisterende arbejdsgange.....	33
4.1.2 Problemidentifikation	34
4.1.3 Fremtidigt anvendelsesområde - rigt billede.....	34
4.1.4 Problemområdet - fremtidige datakrav og begrebsforklaring.....	36
4.2 Forslag til fremtidig arbejdsgangsbeskrivelse.....	38

Indhold

System - slice 1	Skades- og aktionsregistrering	39
5	Objekt Orienteret Analyse	39
5.1	Anvendelsesområdet	39
5.1.1	Aktører	39
5.1.2	Brug (usecases og funktionalitet).....	41
5.1.3	Brugergrænseflade	47
5.2	Problemområdet	49
5.2.1	Klassediagram 1	50
5.2.2	Klassediagram 2	51
5.2.3	Klassebeskrivelse	53
6	Objekt Orienteret Design	55
6.1	Arkitektur	55
6.1.1	Komponentarkitektur	56
6.1.2	Systemarkitektur	58
6.2	Brugergrænsefladekomponenten	59
6.2.1	Struktur	59
6.3	Funktionskomponenten	59
6.3.1	Struktur	59
6.3.2	Standardmoduler	62
6.3.3	Valg af dataadgang	62
6.4	Model-/databasekomponenten	62
6.4.1	Transformering fra UML til logisk datamodel	62
6.4.2	Transformering fra logisk datamodel til fysisk datamodel	64
6.4.3	Normalisering	66
6.4.4	Håndtering af inddata	66
7	Konstruktion	67
7.1	MS Access applikation.....	67
7.1.1	Tabelstruktur	67
7.1.2	Datavalidering	69
7.1.3	Forespørgsler	70
7.1.4	Formularer	71
7.1.5	Navigationsknapper	75
7.1.6	Anden knapfunktionalitet	75
7.1.7	Menulinie	76
7.1.8	Styring af brugeradfærd	77
7.1.9	Sikkerhed	77
7.1.10	Hjælp.....	78
7.1.11	Standardmoduler	79
7.1.12	Programfærdiggørelse.....	80
8	Test	82
8.1	Testplan	82
8.2	Brugertest/systemtest	83
8.2.1	Test procedure.....	83
8.2.2	Test resultater.....	83
8.3	Accepttest.....	84
8.3.1	Test procedure.....	84
8.3.2	Test resultater.....	84

9	Implementering	85
9.1	Installation og afprøvning	85
System - slice 2 Handlingsplan		87
10	Objekt Orienteret Analyse	87
10.1	Anvendelsesområdet	87
10.1.1	Brug (aktører/usecases/scenarier)	87
10.1.2	Brugergrænseflade	89
10.2	Problemområdet	91
10.2.1	Klassediagram.....	91
10.2.2	Klassebeskrivelse.....	92
11	Objekt Orienteret Design	93
11.1	Brugergrænsefladekomponenten/funktionskomponenten	93
11.1.1	Struktur	93
11.1.2	Standardmoduler.....	94
11.2	Model-/databasekomponenten	95
11.2.1	Ændringer i fysisk datamodel	95
12	Konstruktion	97
12.1	MS Access applikation.....	97
12.1.1	Tabelstruktur tilføjelser.....	97
12.1.2	Forespørgsler	97
12.1.3	Formularer	98
12.1.4	Standardmoduler.....	99
System - slice 3 Behandlingsregistrering		101
13	Objekt Orienteret Analyse	101
13.1	Anvendelsesområdet	101
13.1.1	Brug (aktører/usecases/scenarier)	101
13.1.2	Brugergrænseflade	103
13.2	Problemområdet	104
13.2.1	Klassediagram.....	104
13.2.2	Klassebeskrivelse.....	105
14	Objekt Orienteret Design	106
14.1	Arkitektur	106
14.1.1	Deling i en front- og backend	106
14.1.2	Håndtering af samtidighed.....	107
14.2	Brugergrænsefladekomponenten	108
14.3	Funktionskomponenten.....	108
14.3.1	Struktur	108
14.4	Model-/databasekomponenten	109
14.4.1	Transformering til fysisk model.....	109
15	Konstruktion	110
15.1	MS Access applikation.....	110
15.1.1	Tabelstruktur ændringer.....	110
15.1.2	Formularer	110
15.1.3	Programfærdiggørelse.....	110

Indhold

16	Test	111
16.1.1	Test procedure.....	111
16.1.2	Test resultater.....	111
16.2	Accepttest.....	111
16.2.1	Test procedure.....	111
16.2.2	Test resultater.....	111
17	Implementering	112
17.1	Installation og afprøvning	112
Afrunding		113
18	Projekt konklusion og perspektivering	113
18.1	Diskussion og konklusion	113
18.1.1	Datamatiker hovedopgaven	113
18.1.2	Projekt opgaven for Københavns Stadsarkiv	115
18.2	Perspektivering.....	116
Bilag		117
19		117
Projektoplæg Bilag 19.1		119
	Introduktion.....	119
	Foreløbig problemidentifikation	120
	Foreløbig behovsanalyse	121
	Foreløbig afgrænsning.....	121
	Projektet set i forhold til kravene til et eksamensprojekt	121
	Projektaftale	123
	Formål	123
	Mål	123
	Projektorganisering	123
	Projektorganisations model.....	124
	Resurser	124
	Teknologi	124
	Tidsplan	124
	IT-projektets tilpasning til Datamatikerhovedopgave.....	125
Kravspecifikation Bilag 19.2		127
	Indhold	127
	Kort overordnet systembeskrivelse	127
	Detailkrav	128
	Fremtidig arbejdsgangsbeskrivelse og -opdeling	132
Testspecifikation Bilag 19.3		135
	Indhold	135
	Accepttest.....	135
	Brugertest af arbejdsganges sammenhæng med brugerinterfacet	140
Kodestandarder Bilag 19.4		145
	Kommentarer.....	145
	Navnestandard.....	145
	Procedurer og funktioner.....	145
	Fejlhåndtering	145
	Validering af inddata.....	145

Sikkerhed spike Bilag 19.5	147
Synopsis	147
Fysisk sikring	147
Netværkssikkerhed	147
Datasikkerhed ved multibrugerløsninger	147
Diskussion	151
Multibruger spike Bilag 19.6	153
Synopsis	153
Deling af databasen (multibruger).....	153
MS Access opsætningsmuligheder grafisk demonstreret.....	154
Diskussion	158
Arbejdsgangsforslag for udvikling til multibruger.....	159
Hvad er ADO spike Bilag 19.7	161
Synopsis	161
MDAC Arkitektur	161
OLE DB providers	162
ADO kontra ODBC.....	163
ADO kontra DAO	164
Diskussion	165
DML via ADO spike Bilag 19.8.....	167
Synopsis	167
ADO	167
DML	167
Diskussion	167
Transaktioner i MS Access spike Bilag 19.9.....	169
Synopsis	169
Problematik	169
SQL transaktion i MS Access	169
DAO transaktion i MS Access	170
Diskussion	170
Regulære Udtryk spike Bilag 19.10.....	171
Synopsis	171
Regulære Udtryk COM	171
Diskussion	172
Start script spike Bilag 19.11	173
Synopsis	173
Windows Scripting Host (WSH).....	173
Windows Shell	176
Kombineret løsning	176
Diskussion	177
Mail fra MS Access spike Bilag 19.12	179
Synopsis	179
Motivation.....	179
Notes mailklient	179
Interaktion mellem modtager og system	179
Er Notes startet?	180
CDO mailklient	180
Outlook mailklient.....	181
SMTP	181
Diskussion	181
Hjælp spike Bilag 19.13	183
Synopsis	183
Værktøjer.....	183
Diskussion	185

Indhold

Evaluering af milestones Bilag 19.14	187
1. milestone - Indledning og oplæg	187
2. milestone - Projektgrundlag	187
3. milestone - Virksomhedsanalyse.....	188
4. milestone - Foranalyse	188
5. milestone - Slice 1	189
6. milestone - Slice 2	191
7. milestone - Slice 3	192
8. milestone – Projektafslutning.....	193
Projektlog Bilag 19.15	195
Referencer	209
<hr/>	
20 Hjælpemidler	209
20.1 Bøger	209
20.1.1 Primær litteratur	209
20.1.2 Sekundær litteratur.....	209
20.2 Internet links	212
20.3 Software	212
Ord og begrebsforklaring	213
<hr/>	
21	213

Indledning

1 Oplæg og indledende bemærkninger

Formålet med oplægget og de indledende bemærkninger er kort, at beskrive hvad projektet drejer sig om, målsætte hovedopgaven, beskrive slutprodukterne og endelig skitsere hvad der kan forventes.

1.1 Oplæg og målsætning

Denne hovedopgave under datamatikerstudiet er udarbejdet som et projekt i samarbejde med Københavns Stadsarkiv. Her har der været et udtalt ønske om at videreudvikle og forbedre et eksisterende manuelt system til skadesregistrering af arkivalier og planlægning af ressourcer til deres behandling. Værkstedet (hvor Martin Trnka er ansat) havde tilsvarende et ønske om, at kunne dokumentere deres konserveringsbehandling af de arkivalier, der udvælges til behandling, og overordnet var der et ønske om, at kunne følge hele processen og dermed sikre fremdrift i forløbet.

Dette er detaljeret beskrevet i oplægget (se bilag 19.1 s. 119) og målsat i projektplanen (se Formål og Mål s. 11).

I forhold til det meget generelle datamatikerstudie, ser vi hovedopgaven som en mulighed for at fordybe os i en konkret brugervalgt problemstilling, der ønskes løst ved hjælp af et specifikt produkt. Opgaven er således ikke en ”ønskeopgave”, knyttet til vores kernekompetenceområder, ligesom program og programmeringssprog kun kendes overfladisk. Målet for hovedopgaven er at vise, vi kan anvende vores generelle viden til systematisk og analytisk at gennemføre en konkret problemløsning (inden for temmelig vide rammer), samt til at forbinde teori og praksis i et bredt sammenhængende fagligt perspektiv.

Da opgaven eksplicit må tilpasses begge ovenstående fokusområder (Datamatiker studiet og Københavns Stadsarkiv) er det færdige resultat af opgaven opdelt i:

- Projekt rapporten (denne rapport) med bilag, der er målrettet datamatikerstudiet.
- Systemdokumentationen der er målrettet Københavns Stadsarkiv, men skal ses som bilag til rapporten.
- MS Access applikationen (færdig eller som prototype).

1.2 Indledende bemærkninger

For opgaven som sådan har vi bevidst valgt en løs fortællestil, for at undgå at alle afsnit indeholder fraser som: Vi har valgt..., vi mener..., vi har tænkt..., vi har overvejet... osv. Arbejdsgange, forløb, ændringer i planer, succes og fiasko ses altid ikke direkte i rapporten, men reflekteres klart i evalueringerne af de individuelle milestones (se bilag 19.1 s. 187) og projektloggen (se bilag 19.15 s. 195).

Det skal endvidere bemærkes at navne på begreber, klasser, tabeller, forespørgsler, rapporter mm. ofte kan være sammenfaldende. Ligeledes kan navne skifte, sådan at tidligere navngivning på et senere tidspunkt i rapportforløbet, kan dække over ændret indhold eller funktionalitet. Denne navneforvirring er umulig at undgå i dynamiske udviklingsfaser, men kan vanskeliggøre pluklæsning. For rapporten som sådan håber vi dog, at navngivningen virker logisk og meningsfuld i den aktuelle kontekst.

Projekt

2 Projektgrundlag

Projektgrundlaget er projektgruppens formaliserede forudsætninger og rammer for afviklingen af projektet. Her fokuseres på hvorfor og hvordan det kommende projekt skal organiseres og styres, men det er ligeledes vigtigt at fastlægge værktøjer, standarder, kvalitet mm. Projektgruppens arbejde kan herefter allerede fra første færd foregå målrettet og bygge på fælles forståelse af opgaven, udviklingsmodellen og resurserne.

2.1 Projektoplæg

For oplæg, foreløbig problemlidentifikation, -behovsanalyse og -afgrænsning se bilag 19.1 s. 119.

2.2 Formål

Med udgangspunkt i projektoplægget og de eksisterende manuelle systemer for skadesregistrering/aktionsplanlægning samt registrering af konserveringsopgaver er det KSA's ønske at:

- Intensivere indsamlingen og behandlingen af data om skadede arkivalier.
- Forbedre grundlaget for en rationel beslutningsproces og kontrol omkring handlingsplanerne og deres udførelse.
- Skabe klarhed over arbejdsgange.
- Forbedre værkstedets muligheder for dokumentation af arbejdsopgaver og -processer.

2.3 Forudsætninger

2.3.1 Teknologi

- Microsoft Windows 2000 professional DK.
- Microsoft Office XP professional DK.
- Københavns kommunes LAN/WAN. Der er ikke adgang til en dedikeret databaseserve, ligesom der ikke er mulighed for installation af programmer på arbejdsstationerne.

2.3.2 Økonomi

Der er fra KSA's side ikke afsat økonomiske resurser til projektet.

2.4 Mål

- Udvikle og implementere et IT-system hvor skadesregistreringer af arkivalier og afledte aktionsplaner kan indtastes og danne grundlag for prioritering og planlægning af efterfølgende behandling, samt muliggøre kontrol med samme.
- Fastlægge og beskrive arbejdsrutiner, kompetence- og ansvarsområder i forbindelse med systemet og dets anvendelse.
- Udvikle et IT-system, der kan dokumentere arbejdsprocesser og evt. ressourceforbrug mm. for skadesregistreringer der behandles på værkstedet, og at disse kan ses i forbindelse med skadesregistreringssystemet.

Projekt

- Implementere IT-systemet så det kan anvendes fra centrale og decentrale PC'er med netopkobling, indenfor det eksisterende IT-miljø.
- Udarbejde systemdokumentation for IT-systemet.
- Projektet deadline 2003.12.19.

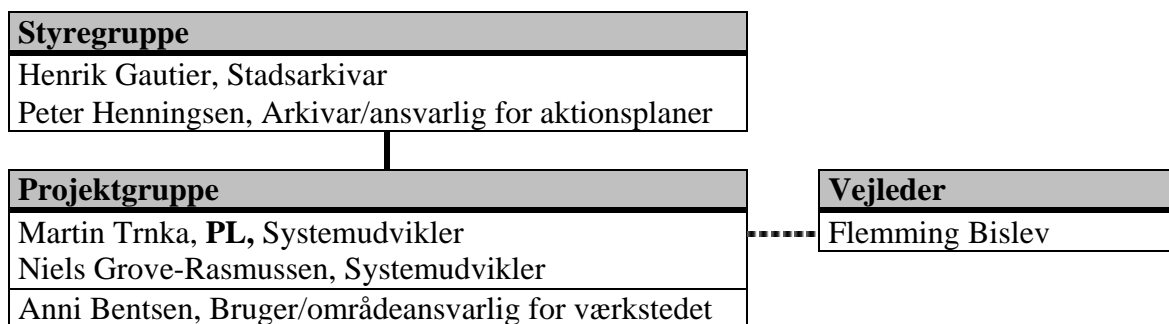
2.5 Projektorganisering

Projektet er forankret sådan, at Stadsarkivaren der har taget beslutning om og afstukket retningslinier for dets omfang, formål og de involveredes kompetencefordeling, samt den arkivar, der skal forestå arbejdet med aktionsplanerne, udgør en formel styregruppe.

Områdelederen for restaureringsværkstedet, Anni Bentsen (AB), der har ønsket en værkstedsdatabase, og er dens primære bruger, indgår sammen med systemudviklerne i projektgruppen. Martin Trnka (MT) har her status af Projektleder (PL) og det formelle ansvar for projektets kvalitet og gennemførelse ligger her. Martin Trnka har som PL reel bemyndigelse til selv at træffe overordnede beslutninger for projektet, uden først at sikre styregruppegodkendelse, da rammerne er velkendte. Projektet har endvidere bemyndigelse til at trække på interne ressourcer i forbindelse med planlægning, udvikling og testning.

Projektudviklerne har via Datamatikeruddannelsen fået tildelt Flemming Bislev (FB) som vejleder. Hans rolle ses primært som en ekstern sparringspartner i forhold til såvel projekt som eksamen.

Projektorganisations model



Figur 2-1. Da PL rollen varetages af en ansat fra Stadsarkivet med god kontakt til styregruppen har vi ikke fundet det nødvendigt, at indføre denne funktion som en selvstændig enhed mellem styre- og projektgruppe.

Til dette forholdsvis beskedne projekt er den skitserede opbygning ideel, da styregruppen udgør Stadsarkivets øverste myndighed, og der dermed i princippet er kort vej til at få afgjort principielle problemer. Projektgruppen har foruden de to systemudviklere deltagelse af en primær bruger for at sikre en hensigtsmæssig brugerrettet løsning af opgaven, opbygge en viden om opgavens løsning hos medarbejderen og dermed sikre optimal motivation, tilfredshed og følelsen af ejerskab af det nye system. Projektgruppens sammensætning og beskedne omfang er en fordel, da ændringer og nye ideer hurtigt kan implementeres og testes direkte. I enkelte specifikke faser (brainstorm, arkivpraksis mm.) kan det dog blive nødvendigt, at trække på eksterne resurser i forhold til projektgruppen.

2.6 Projektgruppestyring og -kommunikation

Der er til projektet oprettet et internet arkiv på Groupcare (www.groupcare.dk), hvor alle projektets relevante oplysninger, dokumentskabeloner, bilag, projektrapport, systemdokumentation og selve MS Access applikationen til en hver tid findes i sin nyeste version.

Med udgangspunkt i projektplanen er der indledningsvist oprettet hoveddokumenter med underdokumenter for de planlagte afsnit og afsnitsoverskrifter. Download, indskrivning eller tilretning samt upload behøver således kun at ske for en delmængde af den aktuelle rapport. Nye afsnitsdokumenter/bilag/spikes kan tilføjes via færdige dokumentskabeloner.

Kommunikation mellem gruppedeltagerne forgår fortrinsvis ved hjælp af Lotus Notes der anvendes af begge og placerer aftalte møder direkte i kalenderfunktionen, der også anvendes i arbejdssammenhæng.

Aftaler og korte referater af vigtige projekthændelser styres ved hjælp af en projektlog der opdateres, udsendes og hvis resultat evalueres på næstkommende møde eller arbejdsdag.

2.7 Hovedopgaverapport og systemdokumentation

Rapport og dokumentation opdateres løbende (se Projektgruppestyring og -kommunikation s. 13). Tilretning i åbne faser af hovedopgaverapporten sker direkte og erstatter tidligere dokumentation. Projektgrundlaget rettes direkte, men på en måde så det fremgår hvilke ændringer og tilføjelser der er sket. For systemdokumentationen gælder, at den til en hver tid afspejler systemet som forefindes.

Uregelmæssigheder og punkter til diskussion markeres i skrivefasen med gul.

Kommende krydshenvisning markeres i skrivefasen med cyklamen.

Kommende litteraturhenvisninger markeres i skrivefasen med grøn.

Tilføjelser i forhold til planen markeres i skrivefasen med turkis og efterfølgende med *.

~~Dele af originalplanen der er udgået, men som fortsat bør indgå markeres gennemstreget.~~

Projekt

2.8 Projektstrategi

Projektstrategien sammenfattes her på baggrund af en mini strategianalyse der forøger at afdække opgavens vilkår [*Strategisk Projektledelse, s. 70*].

Mål og vilkår	Særlige styrker	Særlige svagheder	Mulige beslutninger
Teknikken	MS Access er et tilgængeligt standardsoftware	Uegnet til store databaser	Planlæg database så senere flytning lettes
Udviklerne	Rimelige ressourcer Høj entusiasme	MS Access brugerniveau Ringe projekterfaring	Uddannelse Planlæg detaljeret
Resultatet	Overskueligt problemområde	Ønske om meget simpel løsning	Hold ambitioner og kompleksitet nede
Brugerne	"Bruger" i gruppen Adgang til testperson	Lav grad af EDB-kunnen hos brugere	Simpel GUI Grundig introduktion
Omgivelserne	PL ansat i KSA og kender arbejdsgangen	Risiko for midlertidig ændring af resurser	Planlæg realistisk og med lidt "luft"

Figur 2-2. Projektstrategien bliver til på baggrund af en gennemgang af styrker og svagheder der umiddelbart kan udtrages af viden om teknik, udviklere osv. der er relateret i forbindelse med opgaven.

Projektets strategi begrundet i projektets særlige mål og vilkår	
Sammenfatning	<p>Projektmodellen bør sammensættes så der trækkes på fordelene fra flere systemudviklingsmetoder (se afsnittet Projektmodel og metode s. 14).</p> <p>En modificeret "vandfaldsmodel" med overlappende parallelle faser vil give fleksibilitet, overskuelighed og gode muligheder for styring. Omhyggelig faseopdeling og realistisk tidsplan er helt nødvendig for at sikre og måle fremdrift.</p> <p>De centrale dele af udviklingen bør lægge op til en prototyping-konstruktionsfase på baggrund af en samlet tilbundsgående research der løbende justeres. Brugerinddragelse sikrer funktionalitet og styrer udviklingen.</p> <p>Fra Extreme Programming (fremover XP) kan der inddrages spikes, parkodning, kodestandarder og andre elementer der bidrager til en "lærende organisation".</p> <p>Delleverancer med overdragelse, implementering og instruktion.</p> <p>Sparring og direkte input under udviklingsfasen samt goodwill, indsigt og ansvar for det færdige produkt sikres via aktiv brugermedvirken.</p> <p>Projektet er en del af et uddannelsesforløb hvorfor det færdige produkt opdeles i to relaterede dele.</p> <p>Hovedopgave rapport der "uddannelsesstilpasses".</p> <p>Systemdokumentation, IT-systemet med indbygget hjælp og implementering der målrettes KSA.</p>

Figur 2-3. Styrker og svagheder opvejes og forsøges afbalanceret, hvorefter hovedlinierne for en arbejdsmodel udledes.

2.9 Projektmodel og metoder

Det ligger fra starten klart, at der er tale om et udviklings-/konstruktionsprojekt som skal afvikles i KSA's eget regi. Der er ikke umiddelbart afledte organisatoriske virkninger af det kommende systems implementering (i bedste fald bedre styring og øget effektivitet).

Ved en indledende gennemgang ses opgaven som delvist bunden, problemområdet synes overskueligt, de mulige løsningsmodeller for databasedelen virker relativt ukomplicerede. Brugerinterface og arbejdsrutiner kan virke komplekse, men der er umiddelbart mulighed for nedbrydning i flere dele (slices).

Projektmodellen kan med fordel foregå som en sammensmeltning af "vandfald" og vertikal prototyping, suppleret med et par elementer fra XP og styres på milestones.

Vandfaldsmodellen sikrer nedbrydning af projektet i overskuelige faser og at der inden for disse beregnes et "rimeligt præcist" tidsestimat for ressourceforbruget og dets ordning i tid - hvad der letter måling af fremdrift i forhold til estimater og deadlines. Styring og evaluering foretages på milestones der afslutter afgrænsede faser (se Faseopdeling, milestones, succeskriterier og tidsestimering s. 18). Grundmodellen er imidlertid modificeret sådan at der hele tiden kan arbejdes parallelt på flere forskudte faser, der afsluttes og evalueres i den rækkefølge de blev påbegyndt. Dette giver fleksibilitet undervejs og sikrer, at faser ikke fryses for hurtigt.

Prototyping inddrages for at nedbryde kompleksiteten omkring funktionalitet, brugerinterface og arbejds gange, der først endeligt kan afklares i samarbejde med brugerne. Hurtigt fremstillede delkomponenter letter brugerinddragelse/respons og sikrer at nye input og ændrede forudsætninger kan indpasses og at de overordnede mål indfries, til brugernes fulde tilfredshed. IT-systemet færdiggøres og implementeres således i vertikalt sammenhængende, men selvstændige bidder.

Fra XP anvendes spikes til hurtigt at afklare anvendelighed af de for os mange nye problemstillinger og sikrer en rimelig tidsestimering. Programudviklingen/kodning vil for en stor dels vedkommende foregå parvis efter aftalte kodenstandarder for at sikre kvalitet og fælles forståelse. Fælles læring og udvikling er ligeledes et vigtigt aspekt.

Projekt

2.10 Risikoanalyse

Simpel risikoanalyse der er afpasset efter opgaven [*Strategisk Projektledelse*, s. 76]. SWOT eller SBA der nedbryder risikomomenter i undergrupper som projektstørrelse, IT-erfaring, teknologi, organisation, miljø mm. kommer kun i spil ved større projekter.

Risiko	Sandsynlighed	Konsekvens	Håndtering
Fejlestimering af tidsforbrug på grund af ringe erfaring med udviklingsprojekter i MS Access og VBA	75 %	Kritisk	Hovedvægt på projektstyring, funktionalitet og dokumentation. Konstruktionen begrænses iht. prioritering
Ønsker om ændring af kravspecifikationen undervejs	75 %	Marginal	Mindre tilpasninger ønskes. Grundlæggende ændringer venter til version 2.
Vanskeligheder ved at arrangere styregruppemøder kan forsinke en stram projektplan	50 %	Kritisk	PL har fået bemyndigelse til egenhændigt at træffe nødvendige projektbeslutninger
Projektet er afhængig af ganske få udviklere der kan blive forhindret i at fortsætte	<5 %	Kritisk - katastrofal	Risikoen må accepteres
Andre opgaver kan kræve ressourcer så projekt må begrænses	10 %	Kritisk	Konstruktionen begrænses iht. Prioritering. Det manglende udvikles efterfølgende af KSA
Løsningsmodellens kompleksitet, ligger uden for projektgruppens viden og kompetence	5 %	Kritisk	Træk på følgende kilder til umiddelbar hjælp: Kolleger, lærere, medstuderende og litteratur
Tekniske problemer bl.a. i forbindelse med systemrettigheder/resurser	20 %	Kritisk	Tidlig systemtest. Konferer løsning med IT-afdelingen
Brugere yder modstand mod indførelsen - har lav EDB awareness	25 %	Kritisk	Introducer omhyggeligt og indbyg hjælp. Begræns, simplificer og test GUI. Inddrag brugere aktivt.
Brugere føler sig ikke forpligtiget/medinddraget	25 %	Kritisk	Hold møder med brugerne og kommuniker ofte. Øg synligheden ved ugentlige arbejdsdage i KSA
Systemudviklerne består ikke	<1 %	Katastrofalt	Brug det lærte. Følg planen. Opnå fælles forståelse. Fokus på problemløsning. Brug studievejlederen. Kommuniker og vær synlig. Dokumenter fyldestgørende.

Figur 2-4.. Risikoanalysen forsøger at afdække mulige negative planafvigelser fra planen.

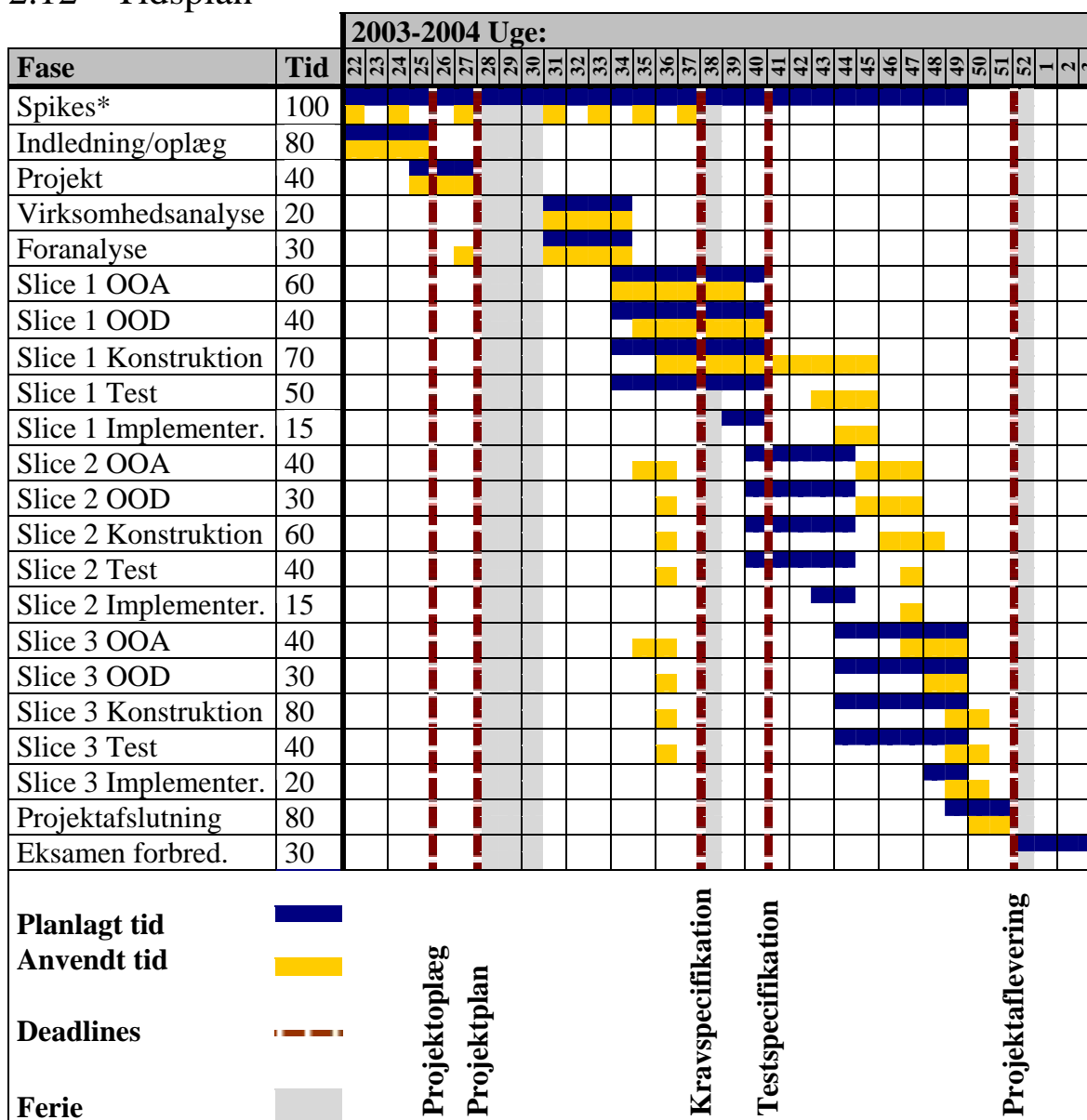
2.11 Projektopdeling

For at få projektplanen på plads tillod vi os at fremskynde første del af foranalysen, for ved hjælp af et op læg til et rigt billede, at få overblik over en mulig projektopdeling.

Vi har udvalgt følgende tre slices:

- Skades- og aktionsregistrering
- Handlingsplan
- Behandlingsregistrering

2.12 Tidsplan



Figur 2-5. Gantt diagram der grafisk viser projektaktiviteter og delmål samt deres ordning i tid. Milestones aflæses i nedenstående detaljerede faseopdeling af projektaktiviteterne.

Projekt

2.13 Faseopdeling, milestones, succeskriterier og tidsestimering

Alle formodede aktiviteter er forsøgt nedbrudt i bidder på max. 12 timer, for at forøge nøjagtigheden af tidsestimeringen. Derved fremkommer en løst "to do list", der ikke gør krav på at være komplet, men forhåbentlig sikrer at ingen større aktiviteter bliver glemt. Det er særdeles vigtigt, at udviklingsgruppen ikke fastlåses af listen som kun er vejledende, og at der undervejs reflekteres åbent men kritisk over mulige konkrete løsningsmodeller.

Styring og evaluering foretages på milestones der afslutter afgrænsede faser og er et af deres hoved succeskriterier.

De efterfølgende skemaer viser: De planlagte faser, fasernes individuelle aktiviteter, en samlet estimeret tid i timer for de to udvikleres arbejde, projektrapportens aktuelle afsnit og om der eventuelt er bindinger til foregående aktiviteter, der først skal færdiggøres.

Planens enkelte tekstlinier fungerer som pladsholdere, der kan omformuleres ved næsten identisk realisering. Udgåede og totalt fejlestimerede emner overstreges så de stadig kan læses, og nye emner tilføjes og markeret med en*. Forløb og ændringer skulle således kunne følges hele vejen fra planlægningsfasen til projektafslutningen.

Fase	Aktivitet	Tid/t	Afsnit	Forgå.
Spikes	Sikkerhed*	12*	19.5	
	Multibruger*	8*	19.6	
	Hvad er ADO*	30*	19.7	
	DML via ADO*	12*	19.8	
	Transaktioner i MS Access*	8*	19.9	
	Regulære udtryk*	12	19.10	
	Startscript*	8*	19.11	
	Mail fra MS Access*	12*	19.12	
	Hjælp i MS Access*	16*	19.13	

Fase	Aktivitet	Tid/t	Afsnit	Forgå.
Indledning og oplæg	Grundlæggende research KSA	12		
	Undersøg krav til hovedopgave	2		
	Møde - gennemgang af indhold og omfang af mulig projektopgave for KSA	4		
	Grundlæggende research teknologi	12		
	Indledende spikes for at lette tidsestimater	12		
	Opsætning af skabelon til projektrapport	12		
	Formøde med skole	2		
	Tilmelding til hovedopgave/vejleder	2		
	Indledning	2	1.1	
	Projektoplæg og -aftale		19.1	
	Introduktion		19.1	
	Aftalt grundlag og oplæg	2	19.1	
	Interview	2	19.1	
	Informationsindsamling	4	19.1	
	Formål	1	19.1	
	Mål	1	19.1	
	Projekt organisering	1	19.1	
	Foreløbig problemidentifikation	1	19.1	
	Foreløbig behovsanalyse	1	19.1	
	Foreløbig afgrænsning	1	19.1	
	Projektrapport mm.	3		
Succeskriterier	Hovedopgavetilmelding senest 2003.05.23	2		
	Projektoplæg og -aftale godkendes ved styregruppemøde senest 2003.06.20.	2		
	Evaluering af 1. milestone (Oplæg)	2		

Projekt

Fase	Aktivitet	Tid/t	Afsnit	Forgå.	
Projekt	Projektoplæg	¼	2.1	19.1	
	Formål (opdateret)	¼	2.2	19.1	
	Forudsætninger	¼	2.3	19.1	
	Mål (opdateret)	¼	2.4	19.1	
	Projekt organisering (opdateret)	¼	2.5	19.1	
	Projektgruppestyring og –kommunikation*	1*	2.6		
	Hovedopgaverapport/systemdokumentation*	1*	2.7		
	Projektstrategi	2	2.8	19.1	
	Projektmodel og metoder	2	2.9	2.8	
	Risikoanalyse	1	2.10	2.9	
	Projektetopdeling	2	2.11	2.10	
	Tidsplan	6	2.12	2.11	
	Faseopdeling, milestones, tidsestimering mm	10	2.13	2.12	
	Projektets realisering i forhold til planen*	1*	2.14		
	Kvalitet	¼	2.15	2.9	
	Programanvendelse og standarder	5	2.16	19.1	
	Afgrænsning	2	2.17	2.12	
		Projektrapport mm.	4		
	Succeskriterier	Grundlag afsluttes inden 2003.07.04			
		Intromøde møde med FB og projektgodkendelse inden 2003.07.04	2		
Evaluering af 2. milestone (Projekt).		2			
Virksomheds-analyse	Virksomhedsbeskrivelse	½	3.1	19.1	
	Organisationsstruktur	2	3.2	3.1	
	Interessenter	4	3.3	3.1	
	Nytteværdi og procesforbedring	4	3.4	3.1	
	Leavitts systemmodel	4	3.5	3.1	
		Projektrapport mm.	4		
	Succeskriterier	Analysen afsluttes inden 2003.08.15. Evaluering af 3. milestone (Virksomhed).	2		
Foranalyse	Projektopgaven			19.1	
	Eksisterende arbejdsgange	4	4.1.1		
	Problemidentifikation	2	4.1.2	4.1.1	
	Rigt billede	4	4.1.3	4.1.2	
	Krav til data og begrebsforklaring	4	4.1.4	4.1.3	
	Fremtidig arbejdsgangsbeskrivelse	6	4.2	4.1.4	
		Projektrapport mm.	4		
		System dokumentation	4		
	Succeskriterier	Foranalysen afsluttes senest 2003.08.22.			
		Evaluering af 4. milestone (Foranalyse).	2		

Fase	Aktivitet	Tid/t	Afsnit.	Forgå.
Slice 1 – OOAnalyse	Anvendelsesområdet			
	Aktører	2	5.1.1	4.2
	Brug (aktører/usecases/funktioner)	8	5.1.2	5.1.1
	Brugergrænseflade	12	5.1.3	5.1.2
	Problemområdet			
	Hændelser	4		
	Klassediagram1	4	5.2.1	4.2
	Tilstandsdiagrammer	4		
	Klassediagram 2	4	5.2.2	5.2.1
	Klassebeskrivelse*	4*	5.2.3	5.1.2
	Kravspecifikation (detaljer og arbejdsgang)	12	19.2	4.2
	Projektrapport mm.	4		
	System dokumentation	4		
	Succeskriterier	Kravspecifikation godkendes 2003.09.14. OOAnalysen afsluttes senest 2003.10.05.	2	
Slice 1 – OODesign	Arkitektur			
	Komponentarkitektur	2	6.1.1	
	Systemarkitektur	2	6.1.2	19.6
	Brugergrænsefladekomponenten			
	Struktur	4	6.2.1	5.1.3
	Funktionskomponenten			
	Valg af dataadgang*	4*	6.3.1	6.1.2
	Struktur	2		
	Diagram	4	6.3.2	6.3.1
	Database/modelkomponenten			
	Struktur	2		
	Transformering til logisk model	8	6.4.1	5.2.3
	Transformering til fysisk model	4	6.4.2	6.2.2
	Normalisering*	2	6.4.3	6.3.2
Håndtering af inddata (testkodet)*	40*	6.4.4	6.3.3	
Projektrapport mm.	4			
System dokumentation	4			
Succeskriterier	OODesignet afsluttes senest 2003.10.05.			

Projekt

Fase	Aktivitet	Tid/t	Afsnit	Forgå.
Slice 1 – Konstruktion	MS Access applikation			
	Tabelstruktur	8	7.1.1	
	Datavalidering*	8*	7.1.2	7.1.1
	Forespørgsler	8	7.1.3	7.1.2
	Formularer	12	7.1.4	7.1.3
	Rapporter (måske ikke endnu)	2		
	Switchboard eller andet interface	6		
	Navigationsknapper*	16	7.1.5	7.1.4
	Anden knapfunktionalitet*	8*	7.1.6	7.1.5
	Menulinie*	8*	7.1.7	7.1.6
	Styring af brugeradfærd*	32*	7.1.8	7.1.7
	Sikkerhed	8	7.1.9	
	Hjælp	8	7.1.10	
	Standardmoduler*	8	7.1.11	7.1.6
	Programfærdiggørelse	8	7.7.12	7.7.11
	Projektrapport mm.	4		
	System dokumentation	8		
Succeskriterier	Konstruktionen afsluttes senest 2003.10.05.			
Slice 1 – Test	(Oprindeligt kun et løst "timebox-estimat")	(25)		
	Test			
	Testplan	8	8.1	7
	Bruger/systemtest			
	Testprocedurer	8	8.2.1	
	Testresultater	4	8.2.2	
	Accepttest			
	Testprocedurer	8	8.3.1	
	Testresultater	4	8.3.2	
	Testspecifikation	12	19.3	
	Projektrapport mm.	4		
	System dokumentation	4		
Succeskriterier	Testspecifikation godkendes 2003.10.05. Testning afsluttes senest 2003.10.05.			
Slice 1 – Implementering	Installation og afprøvning	4	9.1	8
	Introduktion			
	Forberedelse	2		
	Præsentation	2		
	Projektrapport mm.	2		
	System dokumentation	2		
Succeskriterier	Implementering afsluttes senest 2003.10.05. Evaluering af 5. milestone (Slice1 samlet).			
		4		

Fase	Aktivitet	Tid/t	Afsnit	Forgå.
Slice 2 – OOAnalyse	Anvendelsesområdet			
	Brug (aktører/usecases/scenarier)	8	10.1.1	19.1
	Brugergrænseflade	4	10.1.2	10.1.1
	Problemområdet			
	Hændelser	4		
	Klassediagram	4	10.2.1	
	Klassebeskrivelse	2	10.2.2	10.2.1
	Tilstandsdiagrammer	4		
	Kravspecifikation (+ justering)		19.2	
	Projektrapport mm.	4		
Systemdokumentation	4			
Succeskriterier	Kravspecifikation godkendes 2003.11.02. OOAnalysen afsluttes senest 2003.11.02.	1		
Slice 2 – OODesign	Arkitektur			
	 Komponentarkitektur	±		
	 Systemarkitektur	±		
	Brugergrænseflade-/funktionskomponenten			
	Struktur	2	11.1.1	
	Standardmoduler	2	11.1.2	
	Model-/databasekomponenten			
	 Struktur	±		
	Ændringer i fysisk datamodel	8	11.2.1	
	Projektrapport mm.	4		
Systemdokumentation	4			
Succeskriterier	OODesignet afsluttes senest 2003.11.02.			

Projekt

Fase	Aktivitet	Tid/t	Afsnit	Forgå.
Slice 2 – Konstruktion	MS Access applikation			
	Tabelstruktur (tilføjelser)	4	12.1.1	
	Forespørgsler	8	12.1.2	12.1.1
	Formularer	12	12.1.3	12.1.2
	Rapporter	8		
	Switchboard eller andet interface	4		
	Sikkerhed	4		
	Hjælp	8		
	Standardmoduler	12	12.1.4	12.1.3
	Programfærdiggørelse	4		
	Projektrapport mm.	4		
	System dokumentation	8		
Succeskriterier	Konstruktionen afsluttes senest 2003.11.02.			
Slice 2 – Test	(Oprindeligt kun et løst "timebox-estimat")	(25)		
	Test			
	Accepttest			
	Testprocedurer	8		
	Testresultater	6		
	Systemtest			
	Testprocedurer	8		
	Testresultater	6		
	Testspecifikation (tilføjelser)	4		
	Projektrapport mm.	4		
System dokumentation	4			
Succeskriterier	Testspecifikation godkendes 2003.11.02. Testning afsluttes senest 2003.11.02.			
Slice 2 – Implementering	Installation af prototype	4		
	Introduktion			
	Forberedelse	2		
	Præsentation	2		
	Projektrapport mm.	2		
	System dokumentation	2		
	Succeskriterier	Implementering afsluttes senest 2003.11.02. Evaluering af 6. milestone (Slice 2 samlet).		
		4		

Fase	Aktivitet	Tid/t	Afsnit	Forgå.
Slice 3-OOAnalyse	Anvendelsesområdet			
	Brug (aktører/usecases/scenarier)	10	13.1.1	
	Brugergrænseflade	4	13.1.2	13.1.1
	Problemområdet			
	Hændelser	4		
	Klassediagram	4	13.2.1	
	Klassebeskrivelse	2	13.2.2	13.2.1
	Tilstandsdiagrammer	4		
	Kravspecifikation (+ justering)	4	19.2	
	Projektrapport mm.	4		
System dokumentation	4			
Succeskriterier	Kravspecifikation godkendes 2003.12.07. OOAnalysen afsluttes senest 2003.12.07.	1		
Slice 3-OODesign	Arkitektur			
	Deling i en front- og backend	4	14.1.1	
	Håndtering af samtidighed	2	14.1.2	
	Brugergrænsefladekomponenten	4	14.2	
	Struktur			
	Funktionskomponenten			
	Struktur	4	14.3.1	
	Model-/databasekomponenten			
	Struktur	2		
	Transformering til logisk model	4		
	Transformering til fysisk model	4	14.4.1	
	Normalisering	4		
	Projektrapport mm.	4		
System dokumentation	4			
Succeskriterier	OODesignet afsluttes senest 2003.12.07.			

Projekt

Fase	Aktivitet	Tid/t	Afsnit	Forgå.	
Slice 3- Konstruktion	MS Access applikation				
	Tabelstruktur (tilføjelser)	4	15.1.1		
	Forespørgsler	8			
	Formularer	12			
	Rapporter	12			
	Formularer	4	15.1.2		
	Sikkerhed	4			
	Hjælp	8			
	Programfærdiggørelse	16	15.1.3	15.1.2	
		Projektrapport mm.	4		
	System dokumentation	8			
Succeskriterier	Konstruktionen afsluttes senest 2003.12.07.				
Slice 3- Test	(Oprindeligt kun et løst "timebox-estimat")	(25)		15.1.5	
	Bruger/systemtest				
	Testprocedurer	8	16.1.1		
	Testresultater	6	16.1.2		
	Accepttest				
	Testprocedurer	8	16.2.1		
	Testresultater	6	16.2.2		
	Testspecifikation (+ justering)	4	19.3		
		Projektrapport mm.	4		
		System dokumentation	4		
Succeskriterier	Testspecifikation godkendes 2003.12.07. Testning afsluttes senest 2003.12.07.				
Slice 3- Implementering	Installation af prototype	4	17.1	16.2.2	
	Introduktion				
	Forberedelse	4			
	Presentation	2			
		Projektrapport mm.	2		
		System dokumentation	2		
Succeskriterier	Implementering afsluttes senest 2003.12.07. Evaluering af 7. milestone (Slice 3 samlet).				
		4			

Fase	Aktivitet	Tid/t	Afsnit	Forgå.
Projekt-afslutning	Konklusion (forløbs og resultat evaluering)	4	18.1	17
	Perspektivering	2	18.2	18.1
	Projektrapport mm.			
	Litteratur, internt og softwarehenvis	2		
	Ordforklaring	4		
	Projektlog	2		
	Evaluering af milestones	2		
	Hovedopgaveforside	2		
	Stave-, syntaks- og henvisningskont	12		
	Korrektur og rettelse	4		
	Udskrivning og efterbehandling	8		
	System dokumentation			
	System CD brænding og læsmig.txt	12		
	Stave-, syntaks- og henvisningskont	6		
Korrektur og rettelse	4			
Udskrivning og efterbehandling	6			
Succeskriterier	Afrunding afsluttes 2003.12.19.			
	Evaluering af 8. milestone (Afslutning).	4		
	Datamatikeropgave deadline 2003.12.19	2		
	Formel overdragelse af IT-systemet til KSA	2		

Estimeret tid		900		
Best case	(Estimeret tid – 25%)	675		
Worst case	(Estimeret tid + 25%)	1125		

Figur 2-6. Det har ikke været muligt helt at nedbryde alle aktiviteter på grund af manglende indsigt på planlægningstidspunktet. Et timebox-tidsestimat på baggrund af de forventede delresultater må derfor accepteres enkelte steder.

Planens enkelte tekstlinier fungerer som pladsholdere, der kan omformuleres ved næsten identisk realisering. Udgåede og totalt fejlestimerede emner overstreges så de stadig kan læses, og nye emner tilføjes og markeres med en*.

2.14 Projektets realisering i forhold til projektplanen

På baggrund af ovenstående, der for det totale projekt anslår 900 timer, er det klart at vores to semesternormeringer for hovedopgavens tidsmæssige udstrækning, på ca. 2 x 250 timer vil begrænse den realiserbare del.

For at kunne skelne mellem hvad der skal medtages og hvad der skal udelades i situationer med tidsnød, vil kravspecifikationen (se bilag 19.2 s.127) der udvikles som del af foranalysen og slice1, kunne ses som en prioriteret liste af krav der kan tjene som en første rettesnor. Endvidere kan der i en evt. kombination, blive tale om udeladelse af hele slices (vertikalt) og/eller udeladelse af kodning, test, implementering mm. (horisontalt).

For at nå projektet rundt vil vi dog som udgangspunkt bestræbe os på at gennemgå OOA & D for samtlige slices og konstruere databasedelen og hovedbrugerinterfaces således at kun sammenbinding af de to dele (kodning), test og implementering overlades til KSA.

Projekt

2.15 Kvalitet

Projektets kvalitet sikres ved:

- Styregruppe godkender/informerer om de nåede milestones.
- Vejlederfeedback efter forud aftalte milestones.
- Intern evaluering af milestones.
- Formelt review af vigtige faser.

2.16 Programanvendelse og Standarder

2.16.1 Programanvendelse

Til gennemførelse og dokumentation af projektførelsen og programudviklingen anvendes: Microsoft Access 2002 DK til database, tabeldiagrammer, formularer og rapporter.

Microsoft Access 2002 UK (Developer).

Microsoft Project 2002 standard DK til projektstyring.

Microsoft Visio 2002-3 DK til organisationsdiagram, rige billeder, klasse-, usecase-, tilstands-komponent- og arkitekturdiagrammer.

Microsoft Word 2002 DK til rapport, tabeller og mindre diagrammer.

2.16.2 Kodestandarder

For at sikre ensartethed samt lette samarbejde og senere vedligeholdelse af IT-systemet, er der udarbejdet et sæt standarder der forsøges fulgt gennem hele processen (se bilag 12.4 s. 145).

2.17 Afgrænsning

- Da opgaven formentlig ikke påvirker grundstrukturen i KSA, vil virksomhedsdelen begrænse sig til en gennemgang af organisationsopbygningen og de følger, som ændringer i arbejdsrutiner, ansvarsområder og teknologi kan have på omgivelserne.
- Der er ikke lagt en egentlig virksomhedsstrategi til grund for løsningsmodellen, der bygger på et simpelt ønske om rationalisering og styring af ressourcer til behandling af arkivalier.
- Økonomi/cost-benefit berøres ikke.
- Problemstillinger omkring datasikkerhed og transaktionsstyring berøres kun i begrænset omfang.
- Testprocedurer (accept-, system- og brugertest) lever af tidsmæssige årsager ikke op til kravene til en fuldstændig og formel softwaretest.
- Der bliver ikke som del af projektet udarbejdet en egentlig brugsvejledning - kun hjælpetekster og fejl-dialogbokse i MS Access databasen.
- Værkstedsdatabasen håndterer kun opgaver der er kommet i stand via skadesregistrering.
- Det virker oplagt at koble KSA's arkivdatabase til skadesregistreringen, men da arkivdatabasen p.t. er under omlægning vil det måske ikke kunne gennemføres.
- I tilfælde af tidsnød må der skæres i det færdige produkt og opgaveprioriteringen vil da ske i henhold til prioriteringen i kravspecifikationen (se bilag 19.2 s. 127).

Virksomhed

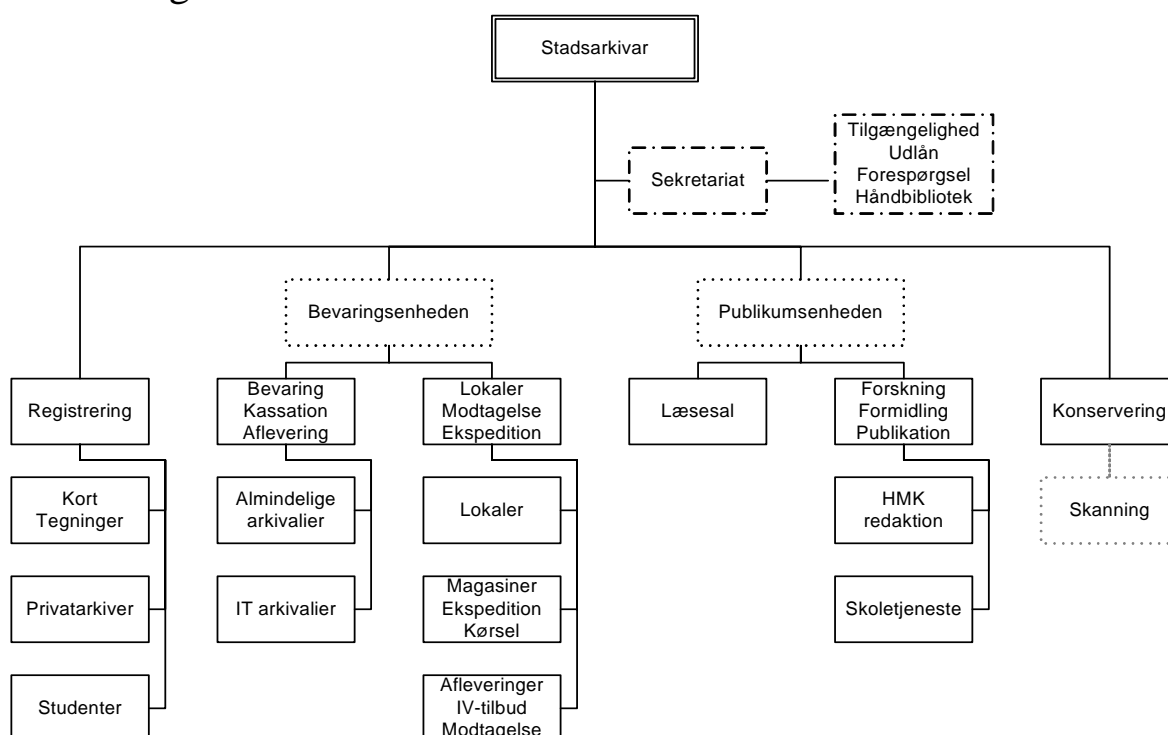
3 Virksomhedsanalyse

Formålet med virksomhedsanalysen er, at forstå omgivelserne og imødekomme de mulige interessekonflikter, indvirkninger og eventuelle ændringer i strukturen, der er nødvendige, hvis projektet skal blive succesfuldt.

3.1 Virksomhedsbeskrivelse

Se bilag 19.1 s. 119 for overordnet beskrivelse.

3.2 Organisationsstruktur



Figur 3-1. Stadsarkivets formelle organisationsstruktur 2003.

KSA er en mindre kontraktstyret del af Københavns Kommune og organisationen er opbygget hierarkisk efter linie-stabs-princippet. Der er dog en fladere struktur end det normalt ses i den bureaukratiske model, hvilket er i overensstemmelse med de nyere ledelsesprincipper og idealer for kommunens fremtidige værdigrundlag. Den overvejende del af de faste arbejdsopgaver, styres efter regler og aftaler med klare krav til ansvarsfordeling og passer fint overens med den hidtidige stærkt hierarkiske model. Nye tiltag skal imidlertid nu pga. kontraktstyringen forhandles, hvorfor projekter i dag er en større og vigtigere del af arbejdsformen. Projektarbejderne spænder fra mindre opgaver indenfor linierne til mere komplekse udfordringer med deltagelse fra større dele af organisationen. Medarbejdere arbejder således ofte på tværs (matrix) og opnår specialist kompetence. Den fladere struktur giver kortere kommandoveje, hurtigere beslutningsproces og mere medbestemmelse.

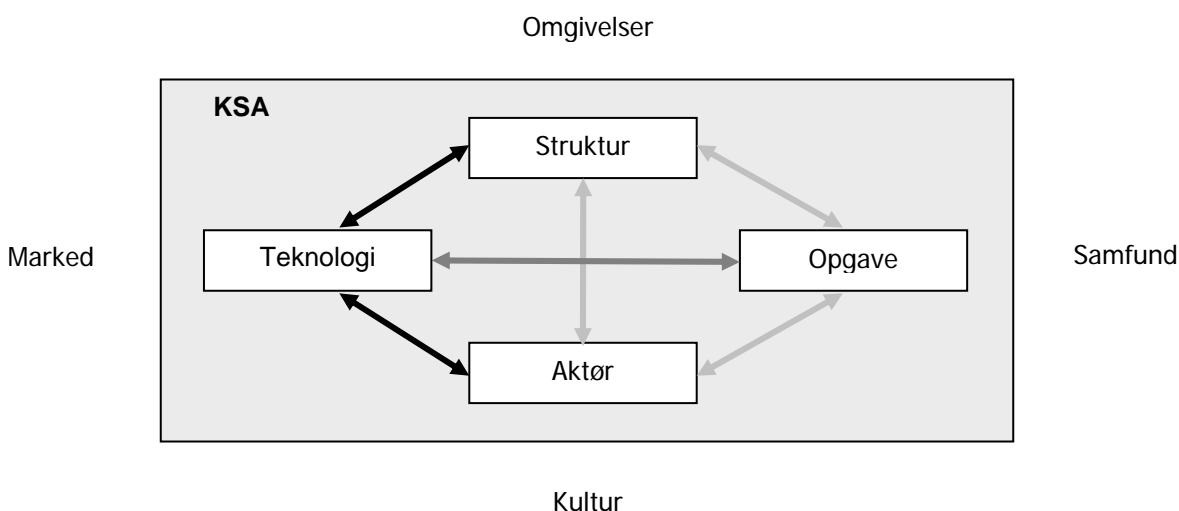
Virksomhed

3.3 Forandringer som følge af implementering af IT (Leavitt)

Indførelse af det ønskede IT-system vil kun ske med succes, hvis der tages behørigt hensyn til organisationsstrukturen, medarbejdernes uddannelse, motivation og opgavernes natur.

For at realisere sig, og dermed imødegå afledte problemer og påvirkninger i organisationen, betragter vi her det teknologiske delsystem i følge H. J. Leavitt's model.

Leavitts systemmodel



Figur 3-2. Ifølge Leavitt [Organisation og Logistik, s. 184] påvirker ændringer i ét delsystem alle andre delsystemer. Da der i dette projekt primært er tale om ændringer i teknologien, påvirkes (i teorien) i første række strukturen og aktørerne og i mindre grad opgaverne.

Teknologi

- KSA, der er en del af kommunens samlede IT løsning, har den fornødne indretning/hardware/software og dermed forpligtelse til at rationalisere arbejdsgange og rutiner for at leve op til egne mål om effektivitet.
- IT-systemet kan øge informationskvaliteten og sikre bedre beslutningsgrundlag.
- IT-systemet kan sikre bedre dokumentation.
- IT-systemet kan sikre bedre kontrol/opfølgning.
- IT-systemet kan rationalisere rutineprægede kontorarbejdsgange.

Aktører

- Der er som altid en fare for modvilje mod nye arbejdsformer, specielt hvis der er tale om jobudvidelse. Dette bør imødegås ved medinddragelse, grundig instruktion og klare arbejdsgange.
- Medarbejder tilfredsheden vil stige, hvis systemet er enkelt at bruge og lever op til forventningerne om bedre styring.
- Klarere arbejdsgange og kompetencefordeling giver bedre arbejdsklima.
- Mere indsigt og bedre intern kommunikation giver bedre arbejdsklima.
- Motivationen og arbejdsglæden ved at udføre behandlingsopgaver bliver større, hvis de er relevante, som resultat af omhyggelig udvælgelse og langsigtet planlægning.
- EDB "awareness" hos medarbejderne må gradvist højnes gennem kurser, da EDB værktøjer i høj grad er/bliver en del af arbejdsopgaverne for alle medarbejdere.

Struktur

- Strukturen berøres kun i begrænset omfang, da systemet er i brug i en manuel udgave.
- Klarere ansvarsfordeling.
- Mulighed for bedre opfølgning.
- Større åbenhed/kontrol da alle har mulighed for indsigt.

Opgaver

- Det vil være naivt at tro at det bliver lettere eller mere intuitivt at udfylde skadesregistrering på EDB, men andre fordele opvejer forhåbentlig besværet med at lære sig ny teknologi - et enkelt og intuitivt GUI vil lette tilpasningen.
- Overblik over skader forbedres.
- Udvælgelse og planlægning af behandling lettes.
- Kontrol med igangsatte behandlingsopgaver lettes.
- Udskrift af aktionsplanen kan, hvis det ønskes, følge arkivalierne.
- Lette dokumentation af arbejdsprocesser.
- Lette dokumentation af effektivitet.

3.4 Interessentanalyse

Interessent	Interesse i projektet	Systemkrav	Projektkrav
Ledelse (HG)	Fastlæggelse af rationelle rutiner samt effektivisering og kontrol med arbejdet	Anvendelighed, bedre beslutningsgrundlag og dokumentation af effektivitet	Kommunikation Omhyggelig implementering
Projektgruppen (ANB & MT)	Succes i arbejdet/ros	Et resultat i overensstemmelse med opgaven der har fokus på: Forståelighed, let vedligehold og udbygningsvenlighed	Godt og udviklende arbejdsklima
Projektlederen (MT)	Beståelse af eksamen, udvikling af kompetence og tilfredshed ved at bidrage aktivt til det fælles mål for KSA	Tilpas i højest mulig grad systemet til brugerønsker	Overhold plan og deadlines
System-administrator (MT)	Indflydelse på struktur, virkemåde, krav til teknologi og sikkerhed.	Simpel opsætning Simpelt vedligehold Mindst mulig kode (SQL) server forberedt Udbygningsmulighed Fuld dokumentation	Indflydelse
Slutbrugere	Fastlæggelse og forenkling af arbejdsgange vedrørende skadesregistrering	Simpel GUI Let registrering	Tages med på råd
Læsesalsansatte	Mere systematisk metode til skadesregistrering og dermed bedre planlægning og hurtigere reparation af vigtige arkivalier.	Ønske om mulighed for at aftale mindre "ad hoc" opgaver direkte med værksted	Tages med på råd

Virksomhed

Ansvarshavende arkivar (PH)	Overblik over skadesregistreringer og bedre mulighed for udvælgelse og kontrol af i gangsatning og afslutning af behandling	Simpel GUI Let udarbejdelse af aktionsplan Hurtigt overblik Nem søgning	Tages med på råd
Værkstedsleder (ANB)	Systematisering og dokumentering af konserveringsopgaver	Simpel GUI Let registrering Let udskrivning af data til rapporter	Indflydelse
Læsesalsgæster Forvaltning	Læsesalsarkivalier i bedre tilstand	-	-

Figur 3-3. For at sikre størst mulig tilfredshed og identificere interessekonflikter i tide, opstilles og analyseres projektets interessenter og deres indbyrdes interesser og succeskrav. Disse bør indarbejdes hvis målsætning og nytteværdien af projektet skal kunne realiseres.

3.5 Nyttевærdi og procesforbedring som følge af IT-systemet

Formålet med at udarbejde og indføre et IT-system er, at forbedre og tilpasse procesflowet sådan, at de nuværende identificerede virksomheds- og systemrelaterede problemer imødegås og målsætningen fra projektoplægget (se Forudsætninger s.11) opfyldes.

Procesflow fra observering til færdigbehandling af skadet arkivalie



Figur 3-4. Procesflowet ændres i princippet ikke ved indførsel af et nyt IT-system, men kortlagte problemer fra det tidligere manuelle system forsøges elimineret.

Et IT-system hvor interessenternes ønsker indpasses kan bidrage til:

- Klarlagte arbejdsgange og ansvar, der forhindrer usikkerhed og kompetencekonflikter.
- Bedre organiseret indsamling af skades- og aktionsregistreringsdata.
- Forbedret overblik og dokumentation for mængden af skadede arkivalier.
- Forbedret grundlag for udvælgelse (ressource styring) af arkivalier til behandling.
- Forbedret kontrol, styring og opfølgning af igangsatte handlingsplaner.
- Mere synlige arbejdsprocesser, der er tilgængelige for alle over nettet.
- Kobling mellem værkstedets behandling og skadesregistreringen.
- Forbedret og simplificeret dokumentation af værkstedets arbejdsopgaver.
- Decentral behandling i alle faser.

Man kan ikke regne med at et IT-system alene, vil motivere ansatte til at bakke op og udfylde skadesregistreringer på stribe. Alt dette kan kun ske ved samtidig, at vitalisere virksomhedskulturen, klarlægge værdien af arbejdet, fastslå arbejdsgangene samt definere den enkeltes rolle og betydning for projektets samlede succes.

System - foranalyse

4 Foranalyse

Formålet med foranalysen er at forstå situationen (problemområdet og anvendelsesområdet). Derudfra fastlægges overordnede krav til det nye systems indhold og de fremtidige arbejdsgange, der skal ligge til grund for dets anvendelse.

4.1 Projekt opgaven

4.1.1 Anvendelsesområdet - eksisterende arbejdsgange

Skadesregistrering

For at kunne danne sig det ønskede overblik over tilstanden af arkivalier i de dele af KSA der bruges mest, er det meningen at alle medarbejdere, der observerer skadede arkivalier, skal bidrage til at registrere disse. Der er således ikke tale om en systematisk gennemgang af alle KSA's arkivalier, eller den totale kort- og tegningsamling, men mere en registrering af skadeomfanget i de afsnit, hvorfra der oftest ekspederes og rekvireres sager.

Medarbejderen skal med udgangspunkt i det fundne skadede arkivalie:

1. Udfylde forsiden på et grønt Skadesregistreringsskema.
2. Sendte skemaet videre til den ansvarlige arkivar der træffer endelig afgørelse om det videre forløb, og udfylder en Aktionsplan.
3. Udfylde en lysegrøn fortrykt PostIt, der findes i alle arkivafsnit, som indklæbes på forpermens inderside eller vedlægges sagen. Andre kan således se, at en registrering allerede har fundet sted.

Aktionsplan

Udfyldte skaderegistreringer samles og behandles af den ansvarlige arkivar der:

1. I nødvendigt omfang udfylder de data der danner grundlag for senere udvælgelse (klassificering, prioritering og evt. aktioner).
2. I flere tilfælde er et samarbejde med værkstedet eller andre personalegrupper nødvendig for at få klarlagt en hensigtsmæssig strategi.
3. Overblik over samlingens tilstand og dermed bredere beslutningsgrundlag for at samle aktionsplanerne i en pulje, sådan at ensartede prioriterede opgaver over tid kan sammensættes og behandles samtidigt, med en betydelig tidsgevinst som følge
4. Den ansvarlige arkivar aftaler for hvert parti omhyggeligt hvad der skal ske, og hvem der skal forestå arbejdet, samt medsender skaderegistreringen/aktionsplanen sammen med arkivalierne og nødvendig instruktion.
5. Medarbejderne udfører det pålagte arbejde.
6. Arkivalierne sendes retur/stilles på plads i henhold til aftale og skemaet returneres til den ansvarlige arkivar der afslutter sagen.

Konserveringsregistrering

1. For at kunne holde styr på arbejdsprocesser og kunne dokumentere arbejdsopgaver registrerer værkstedslederen alle faglige arbejdsopgaver manuelt på kartotekskort.

System - foranalyse

4.1.2 Problemidentifikation

Umiddelbart kan de identificerbare problemer (symptomer og årsager) der kan udledes af oplægget (se bilag 19.1 s. 119), gennemgang af arbejdsgangene og ved interview inddeles i to hovedgrupper.

Virksomhedsrelaterede problemer

- Manglende motivation fra medarbejdere på alle niveauer til at anvende eksisterende system.
- Uklare ansvarsområder og kompetencefordeling.
- Fejl i vejledningerne til det nuværende system.
- Ingen kontrol af igangsættelse og færdiggørelse af arbejdsopgaver.

Systemrelaterede problemer

- Begrænset overblik over skadede arkivalier.
- Begrænset opfølgning og prioritering af skaderegistreringer.
- Begrænset overblik over aktionsplaner for behandling af skadede arkivalier.
- Begrænset mulighed for at søge, sortere og udskrive i skaderegistreringer og aktionsplaner.
- Begrænset mulighed for styring og igangsættelse af opgaver.
- Begrænset mulighed for at kontrollere færdiggørelse af igangsat arbejde (aktionsplanen er fulgt med opgaven).
- Manglende motivation fra medarbejdere på alle niveauer til at anvende eksisterende system.
- Nuværende system måske for komplekst.
- Nuværende system består af et fysisk eksemplar og kan ikke ses fra andre lokaliteter.
- Registrering af værkstedsopgaver sker manuelt og er langsommeligt.
- Begrænset mulighed for at søge, sortere og udskrive registrerede værkstedsopgaver.
- Manglende kobling mellem skadesregistrering/aktionsplan og værkstedsopgaver.

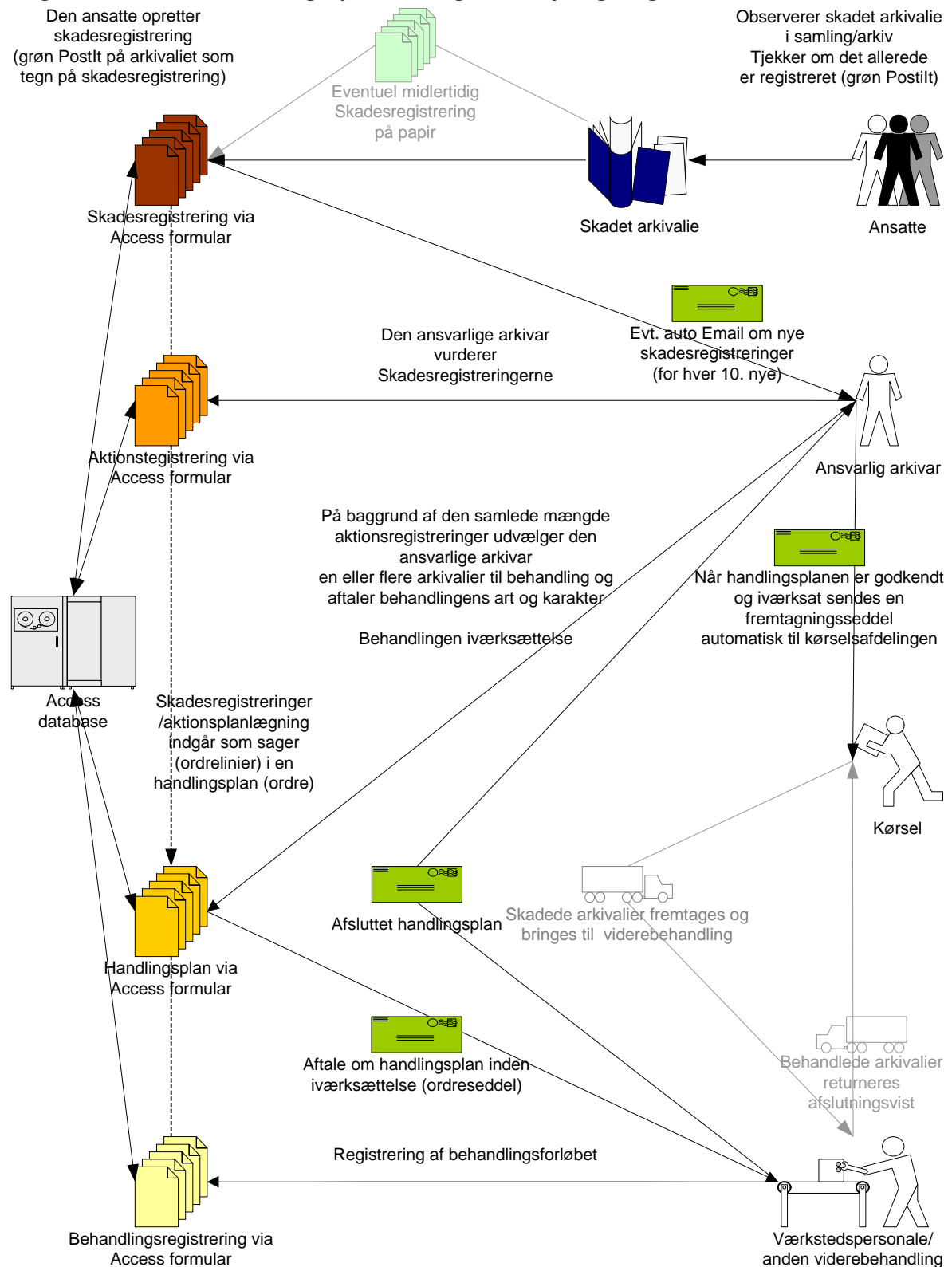
Et tydeligt symptom på at der er problemer omkring det eksisterende system, er den manglende medarbejderopbakning. Den kan bunde i de lokaliserede system- såvel som i de virksomhedsrelaterede problemer.

4.1.3 Fremtidigt anvendelsesområde - rigt billede

Rige billeder er en lidt påskønnet, men særdeles udbytterig metode til at visualisere endog meget komplekse problemstillinger.

Det har lettet dialogen med ikke "nørder" i projekt- og brugergruppen betydeligt, og sidst men ikke mindst inddraget kreativitet i denne fase, hvor vi på en åben måde har prøvet at skabe en nødvendig indsigt og forståelse som grundlag for idégenereringen.

Rigt billede over mulige fremtidige arbejdsgange



Figur 4-1. Forslaget til mulige fremtidige arbejdsgange visualiseret gennem en simpel letforståelig tegning, viser en nyttig opdeling i fire logiske dele: Skadesregistrering, aktionsregistrering, handlingsplan og behandlingsregistrering. Det rige billede danner grundlag for en arbejdsgangsbeskrivelse og senere for usecases. Grå linier og ikoner er arbejdsgange der ligger uden for problemområdet.

Aktionsregistrering (sag)	Bemærkning
Klassifikation A- Nøgle, læsesal, mm. M- Museal B- Andet	Hvilken klassifikationskategori befinder arkivaliet sig i? Journaler, registre, indholdsmæssigt vigtige arkivalier mm. Udstillingsbøger, gamle arkivalier, pergamenter mm. Alle almindelige arkivalier.
Prioritet Uopsættelig Snarest muligt Ved lejlighed Uudnyttede ressourcer Projekt arbejde	Hvor hurtigt kræver det behandling? Det kræver behandling straks. Det bør behandles indenfor kort tid. Det kan ordnes når der er tid. Det har lav prioritet. Det kan med fordel indgå i et særligt projekt.
Aktion KOV afgør behandlingen Nødreparation Konservering Udretning Indbinding Bindrestaurering Bindpleje Opklæbning Omslag Mappe Æske Ompakning Kopiering Skanning Utilgængeliggørelse Anden aktion	Her vælges eller beskrives, påkrævet behandling.
Status Fase Signatur Dato	Aktionsregistreringen anses for afsluttet. ”Aktionsregistreret”. Registrantens initialer. Dato for aktionsregistreringen.

Handlingsplan (ordre)	Bemærkning
Valg af arkivalier	Valg af en til flere skadesregistreringer.
Instruktion Beskrivelse	Uddybende instruktioner til aftalt behandling
Status Fase Signatur Dato	Indledningsvis sammensat handlingsplan. ”Preordre”. Initialer på den aftalen er kommet i stand med. Dato for sammensætning af preordre.
Status Fase Signatur Dato	Handlingsplanen er aftalt og iværksat. ”Ordre iværksat”. Initialer fra den ansvarlige arkivar. Dato for iværksættelse.
Status Fase Signatur Dato	Handlingsplanen anses for afsluttet. ”Ordre afsluttet”. Initialer fra den ansvarlige arkivar. Dato for afslutning.

System - foranalyse

Konservering	Bemærkning
Status Fase Signatur Dato	Konservering påbegyndes. "Konservering påbegyndt". Registrantens initialer. Dato for påbegyndelse af behandlingsregistrering.
Før beskrivelse Beskrivelse	Er der uddybende kommentarer?
Proces Adskillelse Tørrensning Vådrensning Desinficering Neutralisering Stabilisering Lapning Strimling Opklæbning Presning Forsats Hæftning Kapitæler Perme Ryg Overtræk Oppapning: Titelfelt Forgyldning Metalskinner Eftersyn Omslag Mappe Æske Paspapout Anden proces	Hvilke processer har arkivaliet gennemgået?
Efter beskrivelse Beskrivelse	Er der uddybende kommentarer?
Status Fase Signatur Dato	Konservering anses for afsluttet. "Konservering færdig". Registrantens initialer. Dato for afslutning af konservering.

Figur 4-2. På en logisk og begrebstro måde er data forsøgt opdelt, navngivet og forklaret. Resultatet anvendes senere som grundlag for udvælgelsen af hvilke data der skal indgå i IT-systemets enkelte deles funktionalitet samt et første bud på klassekandidater.

4.2 Forslag til fremtidig arbejdsgangsbeskrivelse

Forståelse af omgivelserne, interessenterne, problemområdet, anvendelsesområdet, systemområdet, rige billeder og ovenstående gennemgang af nødvendige systemdata, leder frem til udfærdigelse af en fremtidig arbejdsgangsbeskrivelse der er sendt til høring, og efter godkendelse er indgået som del af kravspecifikationen (se bilag 19.2 s. 127)

System - slice 1

Skades- og aktionsregistrering

Slice 1 er således bestemt til at indeholde IT-systemets skades- og aktionsregistreringsdele, dog uden besked delen. Arbejdet i denne slice danner grundlaget for de følgende to, og dens omfang svarer til mere end to tredjedel af de tre slices tilsammen. Den skal derfor ses som den helt centrale del i forhold til projekt og rapport og de efterfølgende slices som tillæg og tilføjelser til helheden. Arbejdet sker efter en relativ kort analyse og designfase, som en slags prototyping således, at erfaringer i konstruktions (og testfasen) straks implementeres og tilrettes i OOA & D delen, hvorefter iterationen fortsætter.

5 Objekt Orienteret Analyse

Formålet med OOA er overordnet at planlægge og beskrive, hvordan oplægget fra foranalysen kan realiseres under hensyntagen til anvendelsesområdet (aktører, brug, funktionalitet og GUI) og problemområdet (struktur, generalitet, modularitet, indkapsling, genbrug, kvalitet og overskuelighed).

I dette projekt er udarbejdelsen af en kravspecifikation et naturligt biprodukt af OOA fasen. Denne sammensættes af afdækkede detailkrav til systemet, og beskrivelsen af de fremtidige arbejds gange fra foranalysen.

Vi har for analysefaserne valgt, i modsætning til ”Ålborgmetoden” [Objekt Orienteret Analyse & Design], at tage udgangspunkt i anvendelsesområdet sådan som det foreslås af bl.a. Ivar Jacobson, der er parthaver i tilblivelsen af UML notationen og Rational Unified Process. Det virker for os mere intuitivt og rigtigt, først at forstå arbejds gange, brugerinterface og funktionalitet, inden arbejdet med problemområdet og udarbejdelsen af klassediagrammet påbegyndes.

5.1 Anvendelsesområdet

5.1.1 Aktører

Aktørbegrebet er en abstraktion over brugere, der interagerer med et IT-system. Der er foreløbig identificeret fem. Den ene (ansat der udfører anden behandling) er imidlertid ikke umiddelbart en del af oplægget, men tages med i betragtning for at sikre systemets fremtidige udvidelsesmuligheder. Ligeledes er der også for en god ordens skyld medtaget en administrator, der her har rettigheder til det totale anvendelsesområde.

Bruger

Alle ansatte i KSA fungerer i denne rolle.

Brugernes hovedopgave er at:

- Indtaste observationer om skadede arkivalier i KSA, og systemets succes afhænger for en stor del af deres indsats.

Blandt brugerne er der stor spredning på EDB erfaring.

System – slice 1

Arkivar

En ledelsesudvalgt person der besidder indgående kendskab til stadsarkivets samlinger og er sat til at prioritere de ressourcer, der er afsat til behandling.

Den ansvarlige arkivars rolle er delt i to hovedfunktioner:

- Prioritering af de skadesanmeldte arkivalier/sager og evt. foreslå en behandlingsform
- Sammensætning af sager til ordrer og aftale om deres viderebehandling med personalet.
- Sikre og kontrollere iværksættelse og færdiggørelse.

Konservator

Faguddannet person der forestår rådgivning, reparation, konservering og andre værkstedsrelaterede arbejdsopgaver (behandling af skadede arkivalier).

Ligesom arkivaren kan konservatoren:

- Prioritere de skadesanmeldte arkivalier/sager og evt. foreslå en behandlingsform, men det er ikke hans ansvar at det sker.
- Sammensætte dem til ordrer og aftale detaljer om deres viderebehandling med det personale der skal forestå viderebehandlingen, og sikre og kontrollere iværksættelse og færdiggørelse.

Konservatorens rolle er dog at:

- Acceptere/påbegynde og afslutte aftalte handlingsplaner, ligesom at registrere de enkelte sagers påbegyndelse, behandlingsforløb og afslutning.

Ansæt der udfører anden behandling

(Denne aktør er pt. ikke del af problemområdet, men det må formodes at flere roller af denne karakter senere kan tilkobles f.eks. skanning og ompakning)

Alle fast eller løst ansatte personer kan udvælges til at forestå behandlingsopgaver og i denne rolle er det deres ansvar at:

- Acceptere/påbegynde og afslutte en aftalt handlingsplan ligesom de enkelte sagers påbegyndelse, behandlingsforløb og afslutning registreres.

Administrator

Den person der bliver udvalgt til at varetage programmets vedligehold og evt. opdateringer.

Administratorens rolle er at:

- Vedligeholde IT-systemet og sikre at det løbende bliver tilpasset nye relevante brugerønsker.
- Tildele rettigheder/sikkerhed til systemets brugere (implicit og eksplicit)
- Tilpasse IT-systemet til løbende ændringer i teknologi (f.eks. ny mail klient 2004)

Administratoren har indgående kendskab til IT og MS Access.

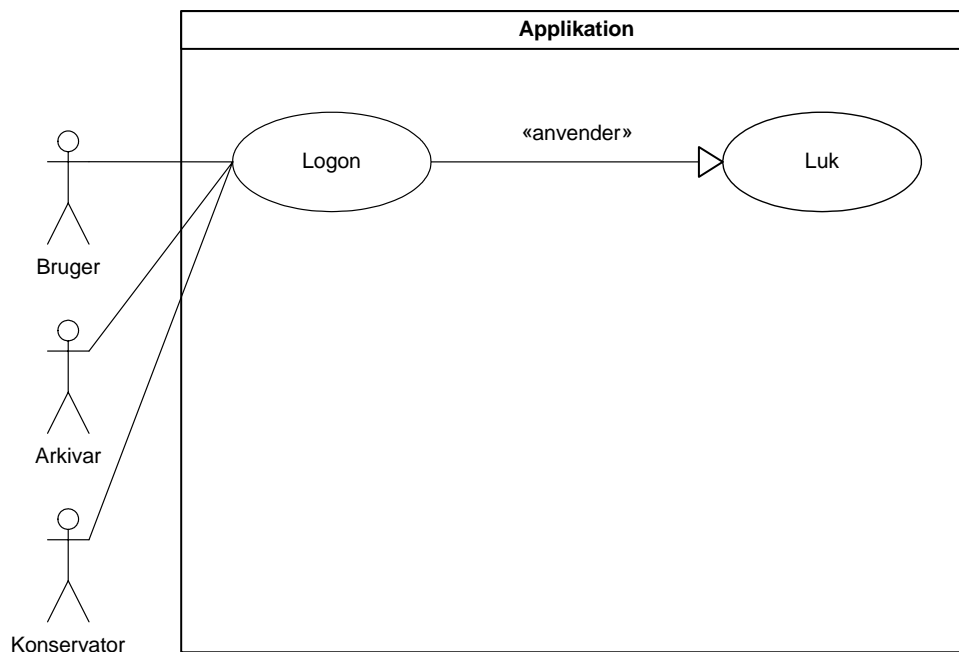
5.1.2 Brug (usecases og funktionalitet)

Usecasebegrebet (brugsmønster) er en abstraktion over anvendelsen af et IT-system. Til hver usecase er der tilknyttet en eller flere aktører/roller/andre IT-systemer, der kan interagere med systemet. Illustrationer og beskrivelser af IT-systemets enkelte dele kan efterhånden give et stadigt mere detaljeret indblik i arbejdsgange, brugeraccess og funktionalitet. Usecases samles i større sammenhængende scenarier der gensidigt tilpasses arbejdsgangsbeskrivelsen fra foranalysen og indgår i kravspecifikationen (se bilag 19.2 s. 127).

Udvælgelse af usecases er en iterativ proces, der er foregået i samarbejde med IT-systemets brugere, og bunder i deres overordnede organisation, ønsker til arbejdsform og fastlagte krav til funktionalitet.

Vi har i denne udviklingsopgave valgt at illustrere og beskrive usecases, samtidig med at krav til ikke-triviel tilhørende funktionalitet forsøges formuleret. Dette må naturødvendigt ske løseligt i prosaform, da der ved arbejdets påbegyndelse ikke var truffet endelige valg omkring datatyper/objekter, brugergrænseflade eller teknologi. Da usecasearbejdet imidlertid er sket iterativt inden for fasen, er ændringer efterfølgende rettet til, hvorfor beskrivelserne er mere konkrete end normalt. Vi har skønnet, at denne fremgangsmåde er fuld tilstrækkelig og i fin overensstemmelse med vores arbejdsform og det prototypearbejde som usecasene her er en vigtig del af.

Applikation

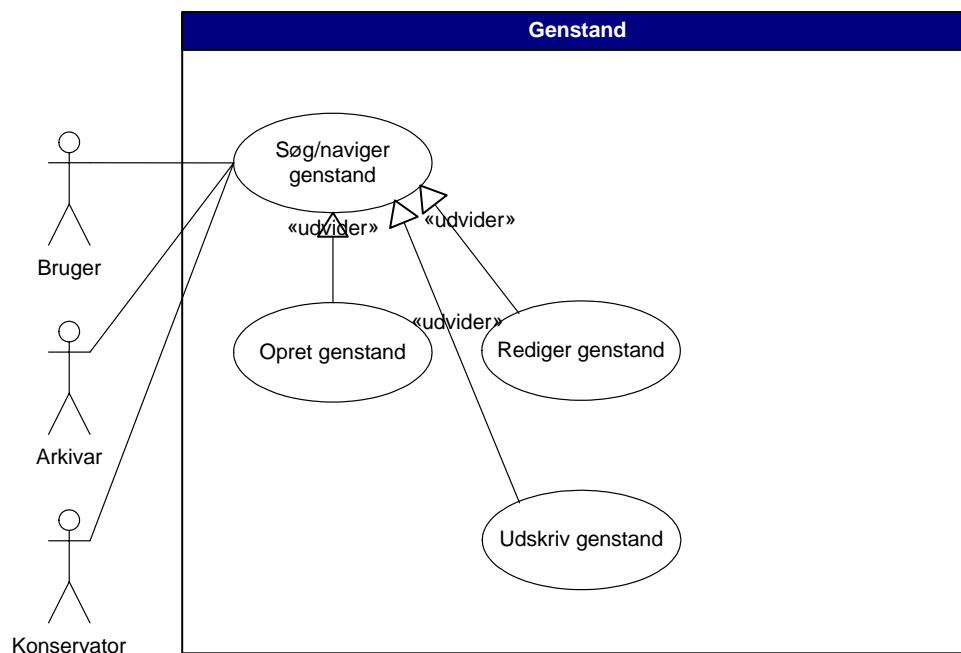


Figur 5-1. Usecasen for selve applikationen består kun af delene: Aktiveringen med logon og luk, hvor brugeren foranlediger at applikationen lukker. Mellem disse to yderpunkter kan usecasen gøre brug af andre usecases hvilket er forsøgt vist på det samlede oversigtsbillede over usecases for slice 1 (se Samlet usecasemodel s. 46)

OBS. Se usecasebeskrivelserne for applikationen i systemdokumentationen (se Systemdokumentation Applikation s. 13)

System – slice 1

Genstand



Figur 5-2. Usecasen for genstand er speciel i forhold til andre ved at den ikke kan slettes og redigering altid kan foretages.

OBS. Nedenstående gengiver kun et eksempel på en usecase. Se alle beskrivelserne i systemdokumentationen (se Systemdokumentation Genstand s. 14)

Opret genstand

Brugsmønster:

For at oprette en ny genstand aktiveres en ”ny” via indbygget funktionalitet.

Der indtastes/vælges data om: Kategori, materiale, genstand.

Funktionalitet:

Brugeren vælger først type og derefter materiale inden indtastningsfelterne for de relevante valg kan udfyldes.

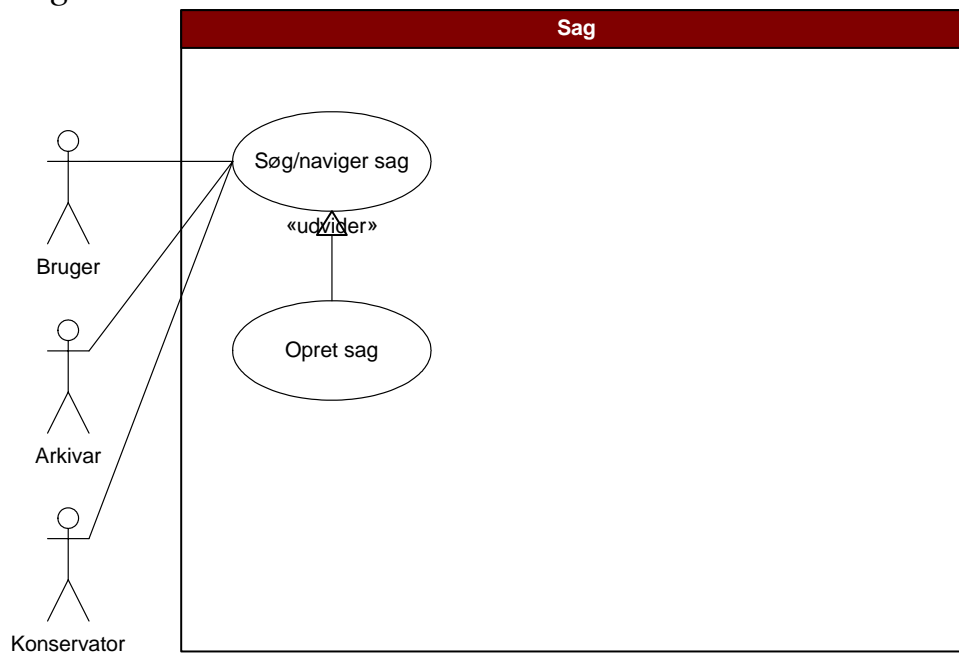
Typevalget er bestemmende for hvilke genstandsfelter der stilles til rådighed.

Ved indtastning i genstandsfelterne valideres disse direkte ved udgang i henhold til indtastningsmasker og valideringsregler.

Ved udgang af genstandsindtastningen kontrolleres om de nødvendige datafelter er udfyldt. I tilfælde af mangler vises en dialogboks med en oprensning af problemerne og mulighed for at fortsætte eller slette. Hvis brugeren vælger at slette, nulstilles alle datafelter i registreringen ellers sættes fokus på det første felt i rækken af manglende data og brugeren får mulighed for at rette og efterfølgende acceptere.

I tilfælde af at brugeren forlader formularen bør data ikke automatisk opdateres, men i stedet valideres hvorefter den tidligere dialogboks viser mulige fejl og giver mulighed for rettelser eller sletning.

Sag



Figur 5-3. Usecasen for sag begrænser sig til oprettelse og navigering, hvilket er helt i overensstemmelse med model og den virkelige verden, hvor sag blot fungerer som en container eller mappe.

Søg/naviger sag

Brugsmønster:

Standardnavigeringen foregår ved at der ”bladres” i resultatet af den aktuelle søgning med navigationsknapper.

Funktionalitet:

Navigering kan ske via indbygget eller konstrueret knapfunktionalitet.

OBS. Det er ikke nødvendigt med specialsøgninger for sag da denne i sig selv ikke indeholder andet end nøgleværdier. Søgning skal ske for skadesregistrering, aktionsregistrering og konserveringsregistrering, hvorved den aktuelle sag også automatisk vises.

Opret sag

Brugsmønster:

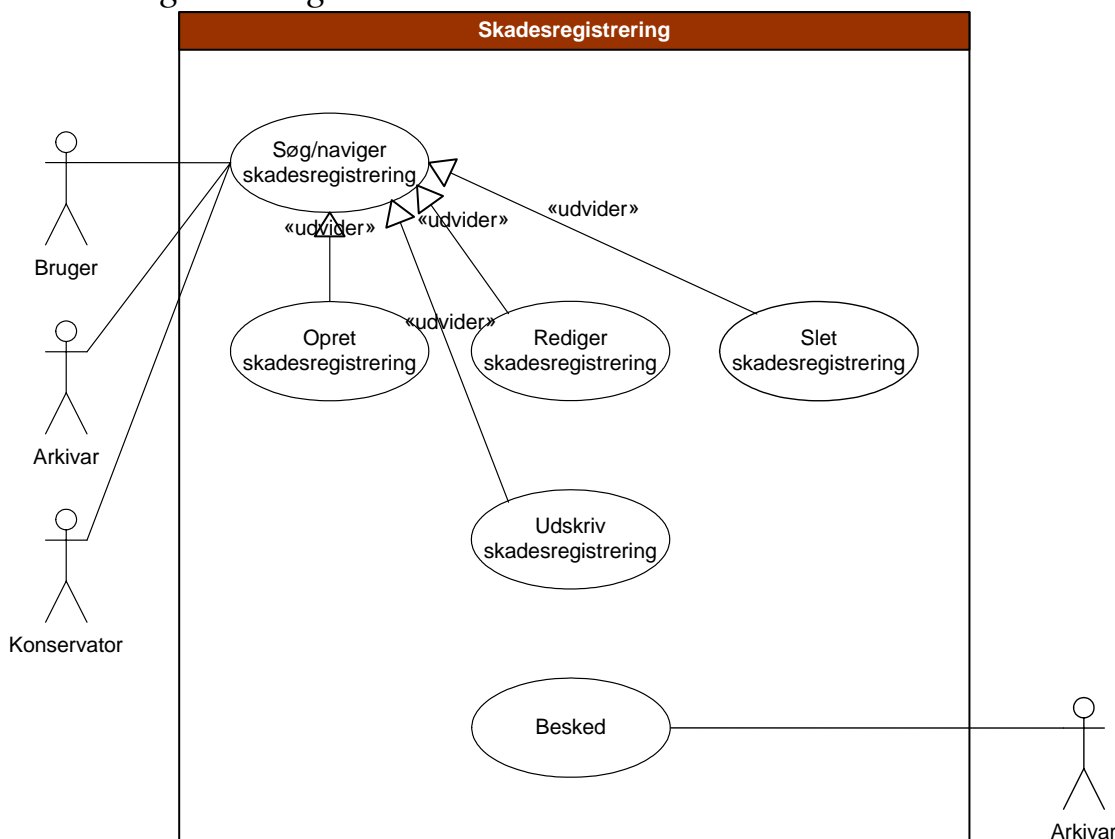
Brugeren aktiverer en ”ny” via indbygget funktionalitet.

Funktionalitet:

Oprettelse af ny sag kan ske via indbygget eller konstrueret knapfunktionalitet.

System – slice 1

Skadesregistrering



Figur 5-4. Frem til og med skadesregistrering har alle aktører samme rettigheder.

OBS. Nedenstående gengiver kun et eksempel på en usecase. Se alle beskrivelserne i systemdokumentationen (se Systemdokumentation Skadesregistrering s.18)

Søg/naviger skadesregistrering

Brugsmønster:

Standardnavigeringen foregår ved at der "bladres" i resultatet af den aktuelle søgning (usecase: Søg/naviger sag) med navigationsknapper.

Via valg kan der alternativt foretages søgninger, der viser almindelige delsæt af den samlede mængde genstande/sager/skadesregistreringer. De aktuelle skadesregistreringer vises med mindre der ikke findes et match i hvilket tilfælde brugeren adviseres og får mulighed for en ny søgning. Med filtrering kan der foretages selektion.

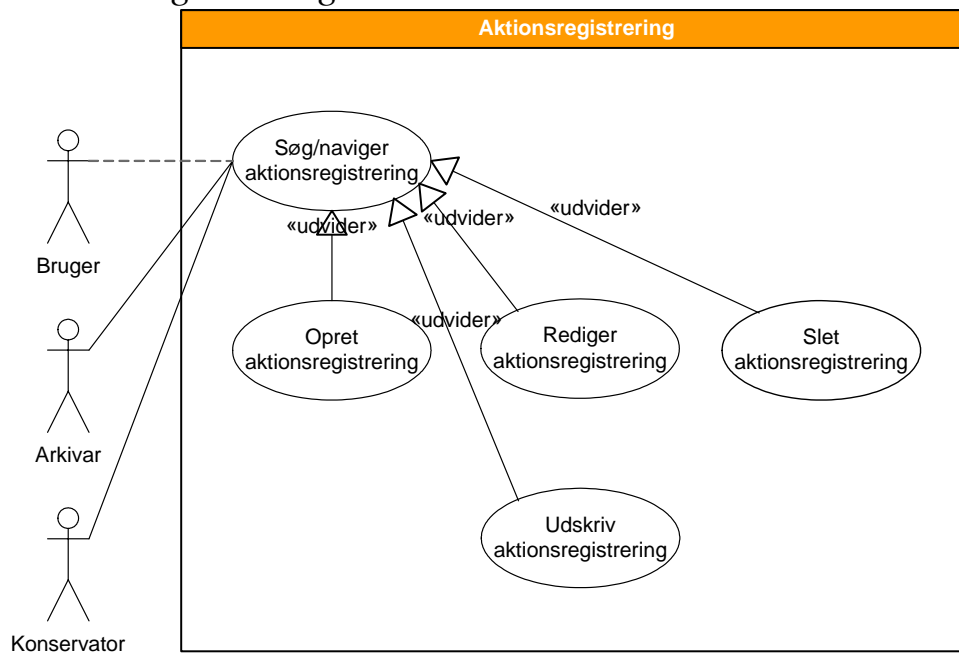
Funktionalitet:

Navigering kan ske via indbygget eller konstrueret knapfunktionalitet.

Søgning kan ske via valg. Der indbygges og vises en dialog i tilfælde af at der ikke findes et match. Hvis søgningen vises bør det fremgå hvilken søgning der p.t. ses.

En søgning på "Alle genstande" bringer brugeren tilbage til udgangspunktet.

Aktionsregistrering



Figur 5-5. Ved actionsregistrering har brugere kun læserettigheder i modsætning til arkivar og konservator der kan oprette, redigere og slette.

OBS. Nedenstående gengiver kun et eksempel på en uscase. Se alle beskrivelserne i systemdokumentation (se Systemdokumentation Aktionsregistrering s.21).

Rediger aktionsregistrering

Brugsmønster:

Aktionsregistreringer der endnu ikke indgår i en handlingsplan/ordre har kun arkivar og konservator ret til at redigere. Da værktødet ikke altid behøver at følge forslagene fra aktionsplanerne slavisk - de er kun forslag - og selv registrerer den reelle behandling i de enkelte sager, er der ikke brug for senere rettelser i aktionsregistreringen.

Annullering og godkendelse sker for de enkelte faser ved hjælp af et sæt knapper der aktiveres og deaktiveres afhængigt af rettigheder og forløb.

Ved redigering skal fasen på ny godkendes eller ændringerne annulleres.

Statusgodkendelsen "Aktionsregistreret" i sagshistorikken opdateres med hensyn til navn og timestamp.

Funktioner:

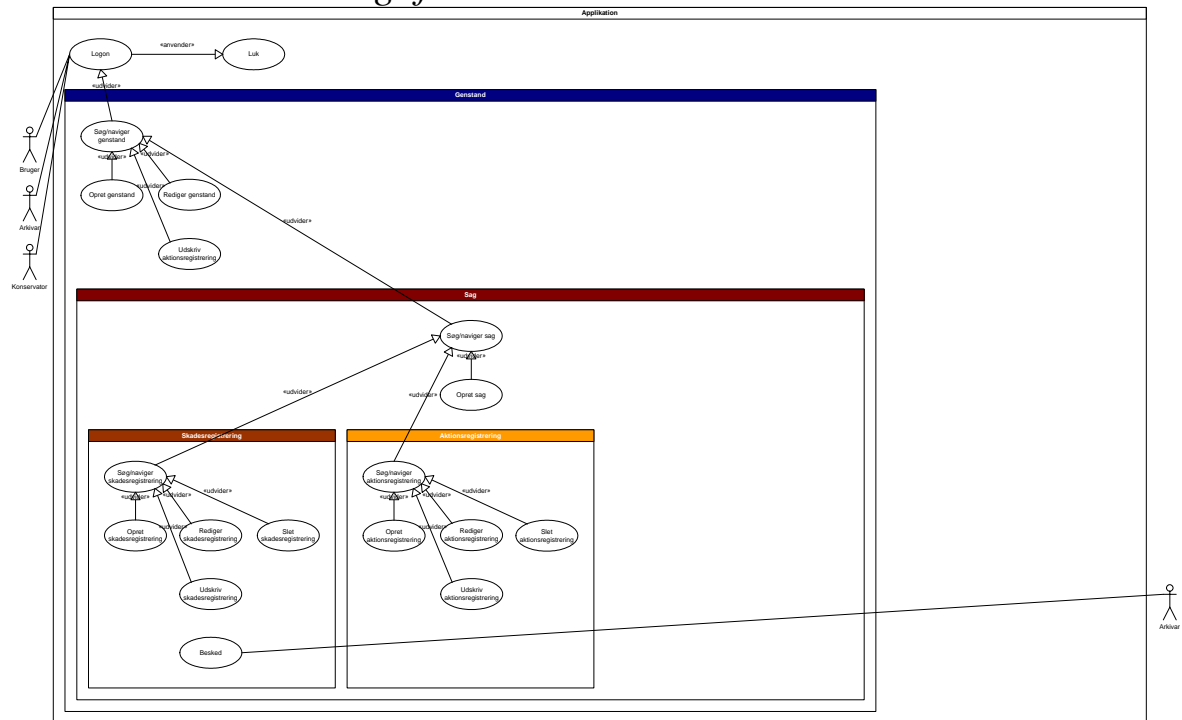
Aktionsregistreringen fremfindes evt. ved hjælp af en søgning og de berørte data skal kunne rettes/opdateres.

Annullering og godkendelse sker som ved oprettelse. Tilsvarende gælder at fasen ved godkendelse på ny kontrolleres.

Statusgodkendelsen "Aktionsregistreret" i sagshistorikken opdateres med hensyn til navn og timestamp.

System – slice 1

Samlet usecaseoversigt for slice 1



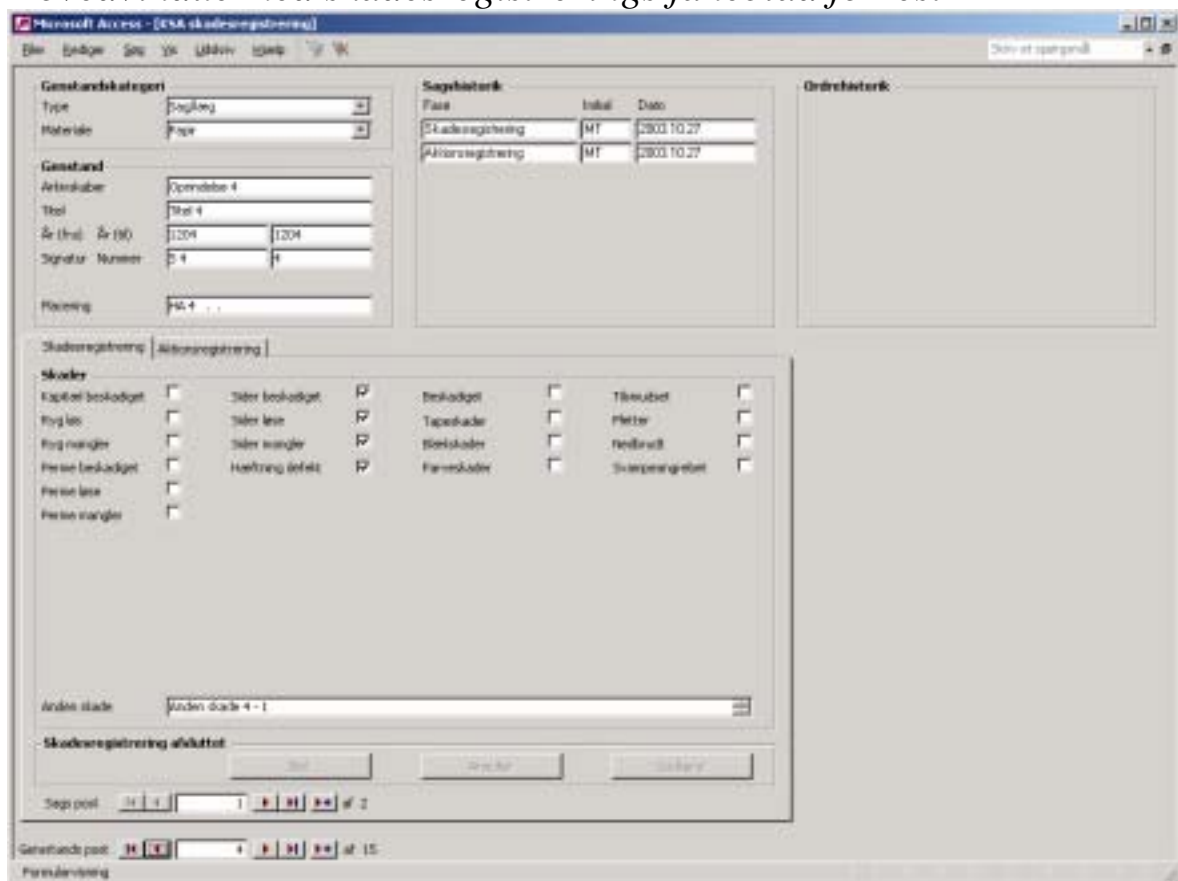
Figur 5-6. Som det ses, skal de enkelte usecases ikke betragtes isoleret, men må sættes ind i en overordnet sammenhæng, for at give et helt billede af IT-systemets totale anvendelsesområde.

5.1.3 Brugergrænseflade

Brugergrænsefladen er grafisk vinduesbaseret og hændelsesstyret. Det har været overvejet om der skulle anvendes det for MS Access meget brugte ”valgindgangsbillede” (switchboard) eller man som i de fleste større programmer starter direkte i applikationens hovedbillede. Vi forsøger her som udgangspunkt at undgå switchboardet og styrer den mest almindelige funktionalitet så intuitivt som muligt fra hovedvinduet, der bør være det vindue der skal anvendes af flertallet af brugerne.

Da applikationens GUI er grænsefladen mellem anvendelsesområdet og problemområdet og al interaktion med applikationen sker herfra, er det overordentligt vigtigt at det opdeles logisk og velordnet på baggrund af usecases og ønsket funktionalitet. Den meget visuelle kartotekskortstil er tilpasset brugerne der for den overvejende del er bibliotekarere og historieuddannede arkivarer.

Hovedvindue med skadesregistrerings faneblad forrest



Figur 5-7. Forslag til GUI for genstand, sag, skades- og aktionsregistrering. Øverst i hovedvinduet er der antydnet en tredeling til stamdata, sagshistorik og ordrehistorik (slice 2,) således at et samlet billede af den enkelte genstands skadesforløb og håndtering let kan overskues. Sagshistorikken viser her at der er tale om en sag der allerede er både skades- og aktivitetsregistreret.

Fanebladene vises ”sætvis” og forholder sig ligesom sagshistorikken til genstandens individuelle sager.

Bemærk at hovedformen har sine egne navigationsknapper og sagen med fanebladene har deres, således at der til en genstand kan knyttes mange sager over tid.

System – slice 1

Basisdelen i hovedvinduet viser overordnet data om genstanden.

En genstand kan have mange sager hvorfor der til hver sag hører et sæt faneblade med skadesregistreringen, aktionsregistreringen og den kommende konserveringsregistrering (slice 3). Der er skal lægges vægt på at det let kan lade sig gøre, at tilføje faneblade med fremtidige aktiviteter eller behandlinger.

Aktionsregistrering på sagsfaneblad

Figur 5-8. Aktionsregistreringsfaneblad med valgbokse til obligatoriske valg af klassifikation og prioritet, efterfulgt af mulige aktioner. Afslutningsvist knapper til sletning, annullering og godkendelse. Navigeringsknapperne i bunden tilhører sammen med fanebladene sagen.

Menulinie for hovedformularen



Figur 5-9. Menulinie der viser mulige søgevalg. Det bør tilsvarende være muligt herfra at kunne foretage udskrivning og navigere til kommende dele af applikationen.

For hver hovedformular med krav til specialfunktionalitet der skal kunne vælges fra menulinien, må der konstrueres en dedikeret menulinie. Standardmenulinien kan ikke anvendes da applikationen som et krav skal kunne afvikles via en runtime version og standardmenulinien understøttes ikke der.

OBS. Indholdsmæssigt er de enkelte formularers elementer beskrevet mere indgående i systemdokumentationen (se Systemdokumentation Formularer s. 50), hvor der også er gengivet et navigationsdiagram.

5.2 Problemområdet

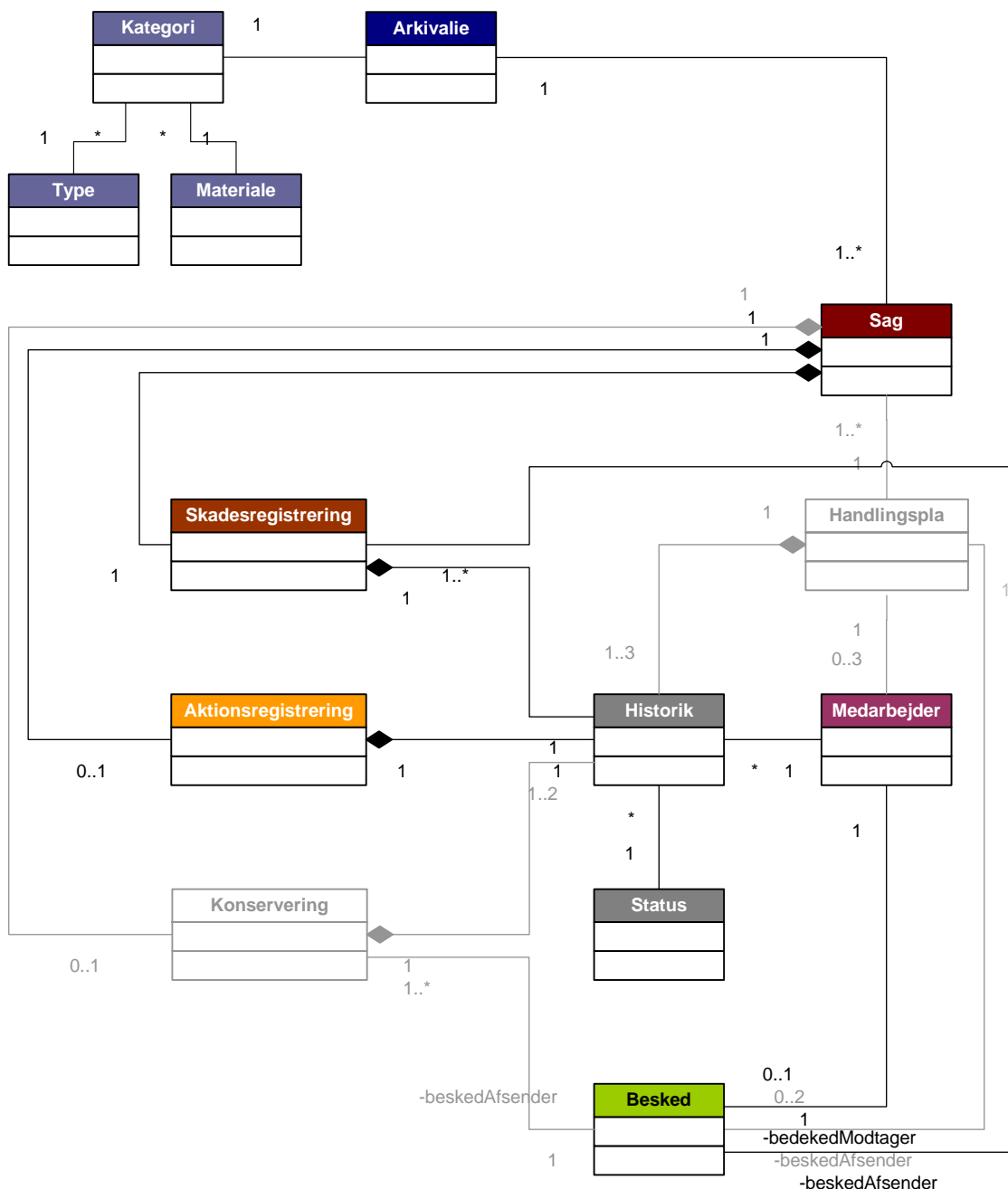
Som led i den systematiske vurdering og udvælgelse af klasser, forsøges deres sammenhænge kortlagt. I OOA fasen som beskrevet i ”Ålborgmetoden” [*Objekt Orienteret Analyse & Design, kapitel 3*] anvendes en hændelsestabel til at identificere objekter i problemområdet og lede frem mod kandidater til klasser (der er objekternes ”blueprint”).

Dette er ikke skønnet nødvendigt her, da foranalysens rige billede og afledte arbejdsgange samt begrebsforklaring og opstilling, som et biprodukt giver os de nødvendige klassekandidater.

System – slice 1

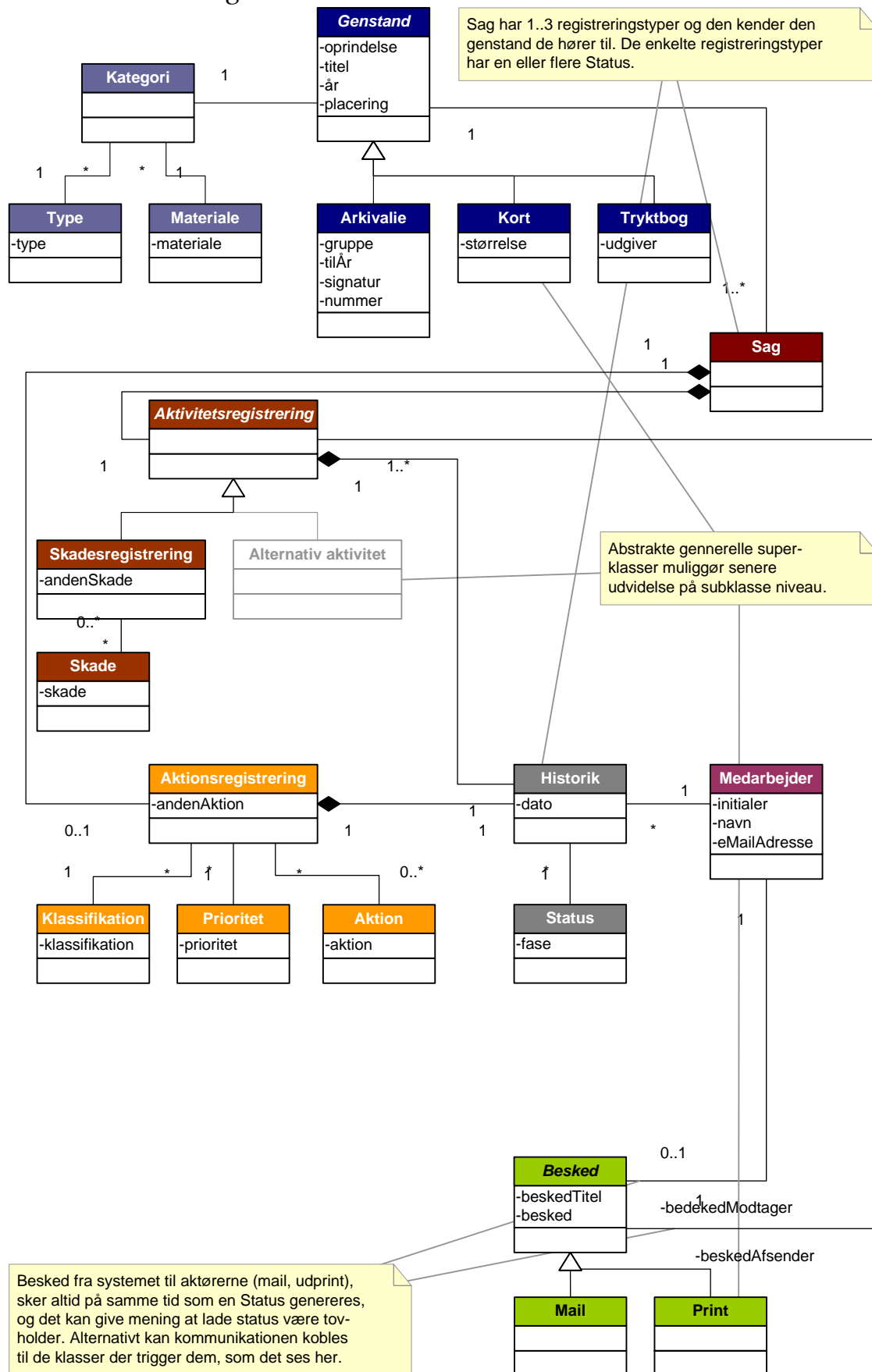
5.2.1 Klassediagram 1

På en åben, men kritisk måde bringes i første række orden på de centrale klasser med stærke indbyrdes bånd. Da denne første slice indeholder langt den overvejende del af klasserne i det samlede system, skabes der indledningsvist et overblik over hele systemet. Dette giver vished for, at udvidelser og tilretninger kan passes ind, og skulle dermed forhindre, at vi i slice 2 og 3 tvinges til radikalt at ændre på den grundlæggende model.



Figur 5-10. IT-systemet er i bund og grund et sags/ordresystem og den centrale klasse er som det også fremgår sag. En sag aggregerer (har en) skadesregistrering og aktionsregistrering og er associeret med (kender) det arkivalie der er den direkte årsag til sagens oprettelse. Et arkivalie kan være associeret med flere sager over tid. For at registrere fremdrift aggregerer skadesregistrering og aktionsregistrering historik der er associeret med status og medarbejder. Da vi ønsker at systemet skal kunne sende beskeder til aktørerne i forbindelse med udvalgte status er mail associeret til de berørte klasser og til medarbejder hvorfra beskedmodtageren hentes.

5.2.2 Klassediagram 2



Figur 5-11. Klassestrukturen efter tilpasning med af arv, aggregering og associeringer.

System – slice 1

Klasserne samler attributter og nødvendig funktionalitet, således at de (afledte objekter) under optimale forhold har en høj indre samhørighed, lav kobling og helst et begrænset veldefineret formål. Det er endvidere ønskeligt, at klasserne har en begrebslig overensstemmelse med det problemområde der forsøges modelleret.

Dette sidste tjener bl.a. til at lette forståelsen og dermed vedligehold, opdatering og udbygning af IT-systemet. Alt sammen kvaliteter der forlænger systemets levetid, holder driftsomkostningerne nede og retfærdiggør den muligvis højere udviklingsomkostning i systemudviklingens første faser.

I denne indledende fase er der naturligvis kun tale om et kvalificeret gæt da vi endnu ikke har styr på alle klassernes attributter, men efterfølgende prototypeaktiviteter sikrer, at udbygningen af en sammenhængende systematisk modelstruktur gradvist udkrystalliseres.

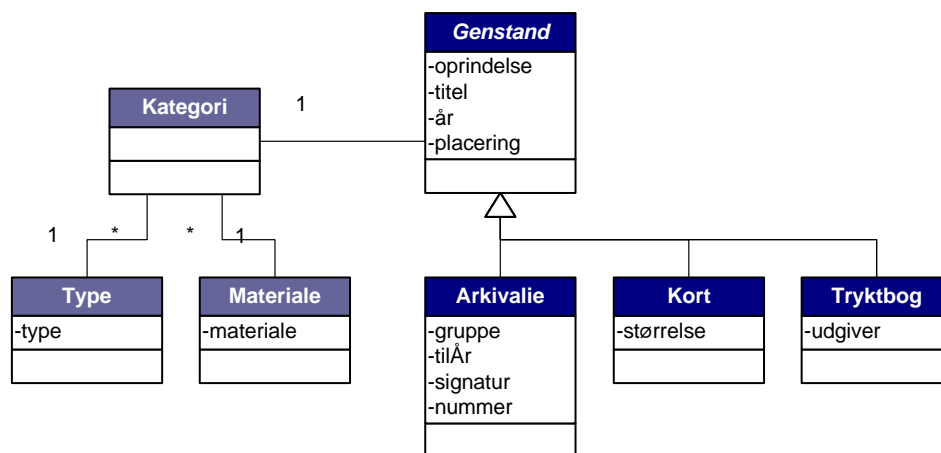
Med den overvejende del af attributterne fastlagt tillader vi os at se bort fra tilstandsdiagrammer der rigtigt udført kan være særdeles tidskrævende og komplekse. Manglende hjælpeattributter findes i konstruktionsfasen og modellen justeres inden fasen afsluttes.

5.2.3 Klassebeskrivelse

Klassebeskrivelserne fastlægger de enkelte klassers ansvarsområde i forhold til modellen. De er for overskuelighedens skyld behandlet blokvis og disse logisk sammenhørende grupper kan tages som udtryk for en form for klynge/pakke opdeling.

Vær opmærksom på at UML notationen implicit medtager aggregerede typer og at disse derfor kun ses i klassebeskrivelserne.

OBS. Nedenstående viser kun ét eksempel på detaljeret klassebeskrivelse. Se alle beskrivelserne i systemdokumentationen (se Systemdokumentation Klassebeskrivelse s. 36).



Figur 5-12. På baggrund af valg af først type og siden materiale der er associeret til kategori, er det muligt at beslutte hvilken konkret klasse af den abstrakte type genstand, der skal anvendes på et givet tidspunkt. Den abstrakte genstand sikrer fleksibilitet og udvidelsesmuligheder da der her let kan tilføjes nye genstandstyper.

Kategori

Klassebeskrivelse

Kategori samler informationer om en genstands type og materiale. Valget af type bør foretages som det første, da det derefter evt. vil være muligt at vise en tilpasset materialevalgsliste. På baggrund af kategori foretages instansiering af specifikke genstandsobjekter. Ændringer kan til stadighed ske, men kan medføre at oprettede specialiseringer af genstand må ændres. De oprindelige data gemmes eller det kan accepteres at tidligere data slettes.

Krav til attributter og deres multiplicitet

1 Type (reference) og 1 Materiale (reference).

Type

Klassebeskrivelse

Definerer hvilke typer en genstand kan have. Det vil være muligt at tilføje andre typer efterhånden som der opstår et behov.

Krav til attributter og deres multiplicitet

1 type.

System – slice 1

Materiale

Klassebeskrivelse

Definerer hvilket materiale en genstand overvejende har. Det vil være muligt at tilføje andre materialer efterhånden som der opstår et behov.

Krav til attributter og deres multiplicitet

1 materiale.

Genstand

Klassebeskrivelse

Genstand er superklasse for alle specialiseringer af genstande der kan tænkes at indgå i IT-systemet og indeholder sammen med kategori basale og generelle stamdata.

Krav til attributter og deres multiplicitet

1 oprindelse, 0..1 gruppe, 1 titel, 1 fraÅr, 1 placering, 0..* Sag (reference).

Arkivalie

Klassebeskrivelse

Definerer en specialisering af genstand. Arkivalie repræsenterer KSA's hovedforretningsområde og indeholder de specialdata der gør samlingen af stamdata komplet. Ændringer kan til stadighed ske og kun de seneste ændringer gemmes.

Krav til attributter og deres multiplicitet

0..1 tilÅr, 1 signatur, 0..1 nummer.

Kort

Klassebeskrivelse

Definerer en specialisering af genstand og indeholder de specialdata der gør samlingen af stamdata komplet. Ændringer kan til stadighed ske og kun de seneste ændringer gemmes.

Krav til attributter og deres multiplicitet

1 størrelse.

Tryktbog

Klassebeskrivelse

Definerer en specialisering af genstand og indeholder de specialdata der gør samlingen af stamdata komplet. Ændringer kan til stadighed ske og kun de seneste ændringer gemmes.

Krav til attributter og deres multiplicitet

1 udgiver.

6 Objekt Orienteret Design

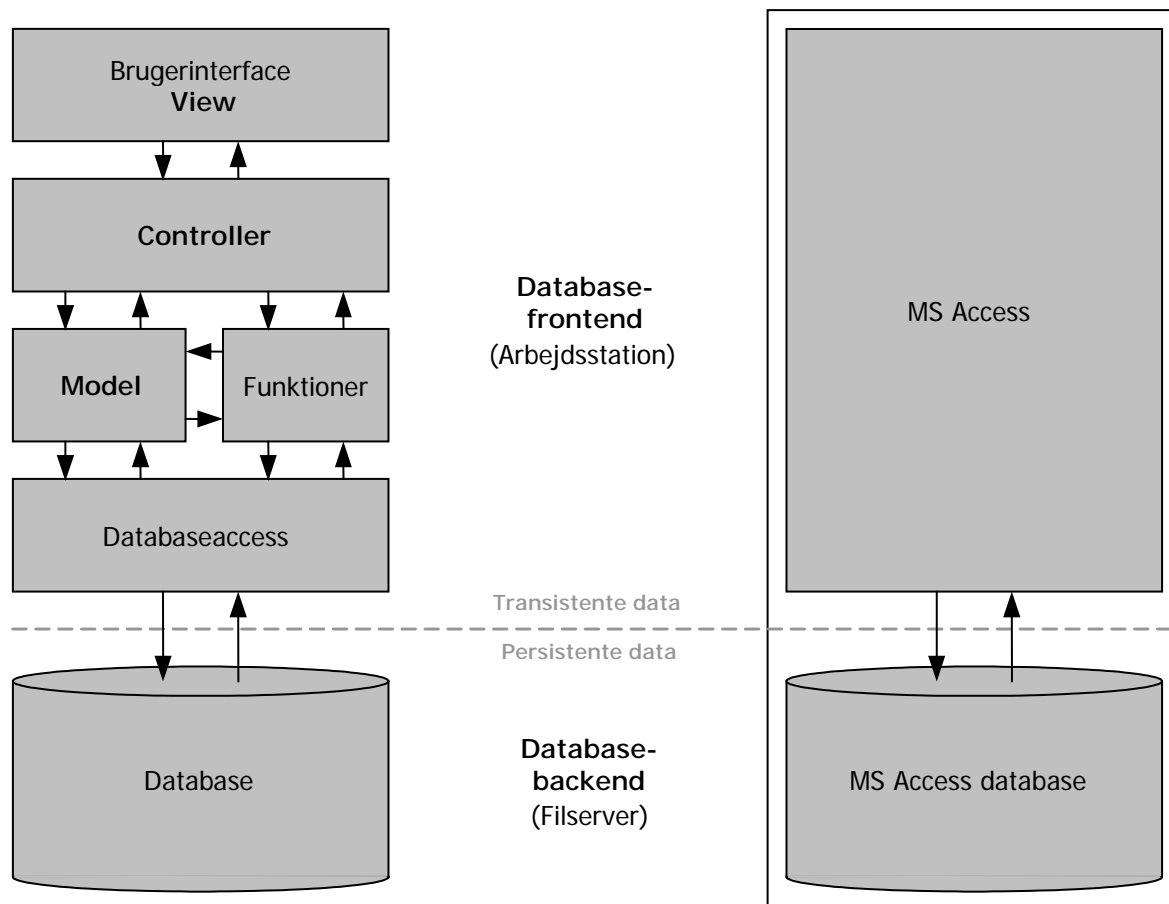
Formålet med OOD er detaljeret, at designe og udvikle hvordan oplægget fra analysefasen skal realiseres i arkitektur og komponenter - i overensstemmelse med kravspecifikationen.

Der er i det indledende analysearbejde gjort brug af OOA, og her i designfasen er der lagt op til at der arbejdes efter OOD principper. Dette er sket vel vidende, at den endelige konstruktion formodentlig ikke bygger på objektorienterede principper. Fordele som fleksibilitet og delegering vil dog forsøges overført i videst muligt omfang.

6.1 Arkitektur

Udarbejdelsen af en domainmodel som det indledningsvist er sket, kontra kravet om et færdigt resultat i form af en MS Access database (af standardtypen hvis en sådan eksisterer), lægger umiddelbart op til to væsensforskellige arkitekturer.

Multi-tier versus en/to-tier arkitektur



Figur 6-1. En multi-tier arkitektur som IT-systemet kunne besidde, hvis det skulle programmeres objektorienteret, viser en klar lag- og ansvarsfordeling for de enkelte komponenter. I den "simplere" MS Access arkitektur, er lignende komponenter sammensmeltet i forskellige grader under en samlet skal der dog kan opdeles i front- og backend. De enkelte komponenter kan i vid udstrækning tilgås, men det sker indenfor MS Access' restriktioner og fleksibiliteten, gennemsigueligheden, robustheden og skalerbarheden, er ringere end i multi-tier arkitekturen. Ligeledes skal man gøre sig klart, at resultatet ikke kan flyttes til platforme udenfor MS Access regi.

System – slice 1

Multi-tier arkitektur (objektorienteret - kodet)

Den illustrerede, gennemførte lagdeling ligger i forlængelse af den objekt-/komponentorienterede tanke om generalitet, modularitet, indkapsling, genbrug, høj indre samhørighed og indbyrdes lav binding mellem komponenter. Da vi her har fuld kontrol over arkitekturen bør den af hensyn til vedligehold, ændring og udbygning være lukket-streng, som vist på tegningen. De enkelte komponenter kan derefter let udskiftes med nye, der blot skal overholde kravene til de fælles grænseflader. Et skift til ny database der overholder andre standarder, vil f.eks. kun medføre udskiftning/tilretning af databaseaccess laget, resten af systemet forbliver uberørt af ændringen.

Konstruktionen her bliver let særdeles kompleks og al styring skal programmeres specifikt, herunder transaktionsstyringen, låsning mm.

En sådan applikation vil normalt være hændelsesstyret. Programmet startes i sit hoved-vindue hvorefter kommunikation og oprettelse af overordnede objekter i modellen sker fra controllerlaget gennem brugerens interaktion med skiftende vinduer.

En/to...-tier arkitektur (struktureret/objektorienteret – grafisk drag and drop/kodet)

Denne tætte sammenknytning af komponenterne i databasefrontenden er ikke kun kendt fra MS Access, også Oracle med sine Forms udgør en sådan konstruktion. I sin simpleste udgave kan der for MS Access vedkommende være tale om en en-tier model, da det er muligt at tilgå og vise indholdet af tabeller direkte i interfacet. Som oftest benyttes det sammensmeltede brugerinterface-, events- og funktionalitetslag med mulighed for tilføjelse af kodede elementer, og et egentligt modellag realiseres sjældent, men er muligt. I disse scenarier hentes recordset fra databasen direkte eller indirekte (forespørgsler/ADO/DAO) frem til brugerinterfacet for visning eller subsidiært til funktionslaget for manipulation inden visning.

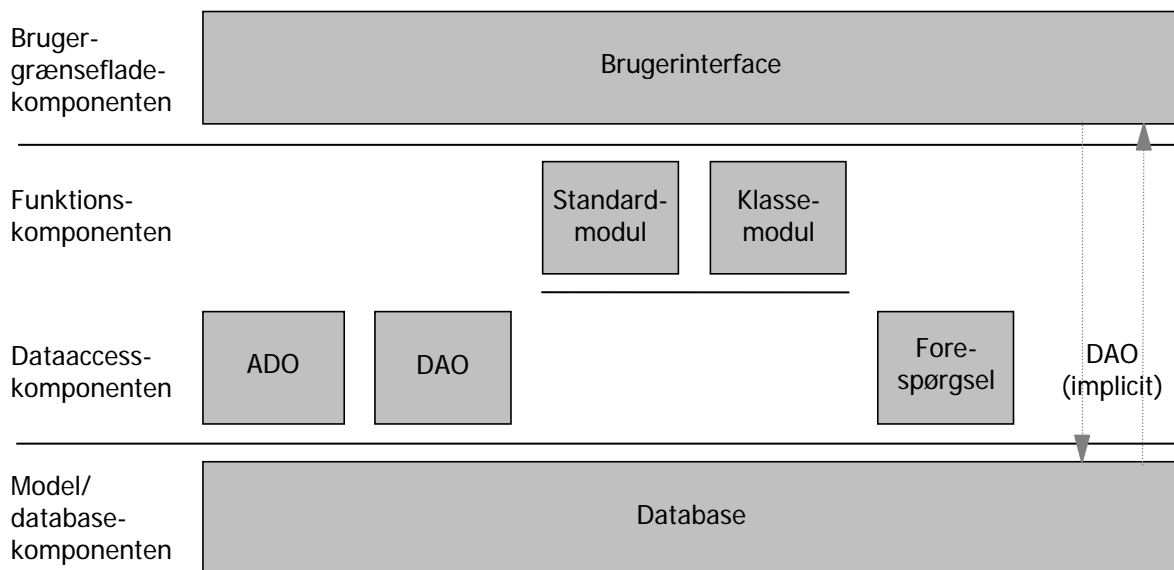
Udviklingsmiljøer af denne type gør lagdelingen delvist transparent for udvikleren og skaber grundlag for langt hurtigere programudvikling (RAD). Styringsmekanismer der normalt omhyggeligt må programmeres findes ofte som indbyggede features, og f.eks. låsningsniveauer håndteres automatisk eller kan let kontrolleres gennem dialogbokse.

Fleksibilitet, gennemskuelse, sikkerhed, portabilitet, skalerbarhed, programkontrol, og -udvikling bliver på den anden side lagt helt i hænderne på produktudbyderen.

6.1.1 Komponentarkitektur

Kravspecifikationen dikterer direkte en MS Access 2002 løsning, og vi har derfor, jævnfør den generelle ovenstående beskrivelse, begrænsede muligheder vi må forsøge at forstå og udnytte bedst muligt.

Forslag til komponent struktur



Figur 6-2. Komponentmodellen for MS Access løsninger gengiver bevidst kun den lille del af den komplette struktur, der er af interesse for vores projekt. Både ADO og DAO er medtaget for at lade det endelige valg stå åbent. Al kommunikation med databasen foregår gennem dataaccesskomponenten, og selvom det fra MS Access kan se ud som om det er muligt at binde tabeller direkte til formularer, sker kommunikation i realiteten via DAO.

Brugergrænsefladekomponenten håndterer interaktion mellem aktører og funktionalitet, og kommer i vores tilfælde til at være synonym med formularer, der er MS Access' grafiske brugerinterface. Der indgår endvidere MS Access' rapporter, der er formateret printet output fra tabellerne, makroer mm.

Funktionskomponenten består af funktionalitet der bl.a. styrer, manipulerer, validerer og viser data. Jævnfør definitionen fra "Ålborgmetoden" [*Objekt Orienteret Analyse & Design*, s. 199] tilføres denne funktionalitet til "modellen" om end det ofte sker indirekte (åben struktur). Også i forholdet til brugerinterfacet, har styring og kontrol af visning og brugeradfærd sin naturlige plads her. Kommunikation vil oftest foregå af to veje:

- Fra brugerinterfacet gennem dataaccesskomponenten til model-/databasekomponenten og direkte tilbage til visning i brugerinterfacet. Eventuelt med behandling i funktionslaget efter visning.
- Fra brugerinterfacet gennem funktionslaget og dataaccesskomponenten til model-/databasekomponenten og ofte samme vej tilbage

Foruden de til formularerne meget stærkt bundne klassemoduler, er det muligt selv at konstruere standardmoduler med generelt anvendelig funktionalitet.

Al kommunikation med databasen foregår gennem dataaccesskomponenten, der består af teknologier og åbne standarder stillet til rådighed af Microsoft. Komponentens er reelt særdeles kompleks, og er her fortolket meget frit.

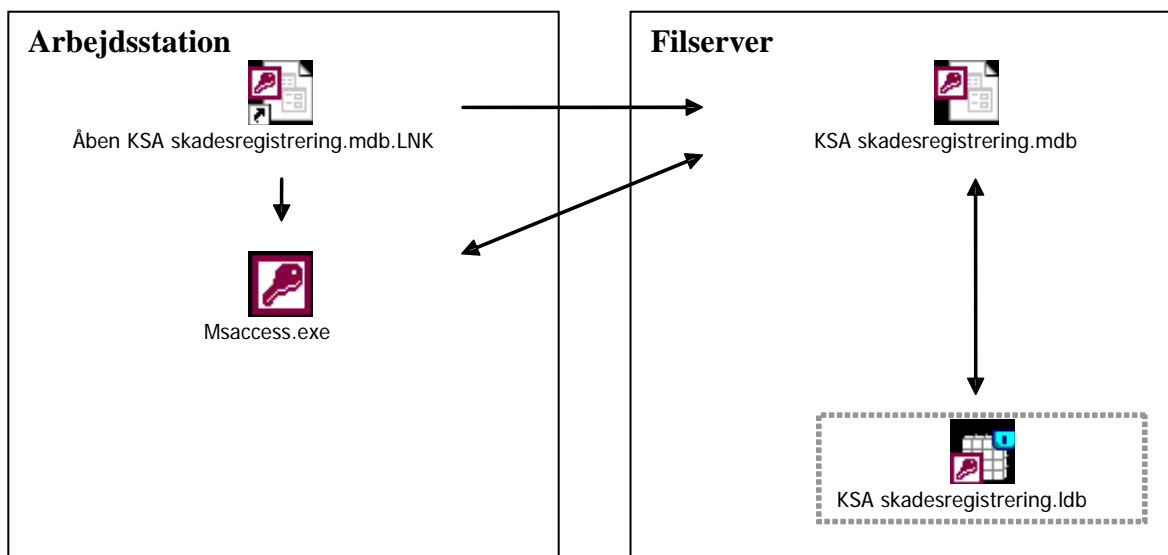
Model-/databasekomponenten sættes lig modellen fra analysefasen, der transformeres til fysiske tabeller og indkapsles af selve databasen eller mere præcist DataBaseManagementSystemet (Jet 4.0). Det er komponentens opgave at holde styr på de objekter/entiteter der er repræsenteret i problemområdet.

System – slice 1

6.1.2 Systemarkitektur

IT-systemet afvikles på arbejdsstationer der over LAN og WAN er koblet til en filserver. Databasen installeres på filserveren og applikationen åbnes lokalt i en runtime version ved hjælp af en genvej.

Det hele på filserveren



Figur 6-3. Jævnfør Multibruger spiken (se bilag 19.6 s. 153) har vi for denne første slice valgt at opbygge MS Access løsningen så simpelt som overhovedet muligt.

Motivationen her er, at anbringelse af hele MS Accessdatabasen på en server eller i en delt mappe er den letteste måde at dele data på. Alle deler data og bruger de samme formularer, rapporter, forespørgsler, makroer og moduler. Denne strategi kan bruges, da alle (udviklere og testere) i denne indledende fase skal bruge MS Accessdatabasen på samme måde.

Differentiering af brugerrettigheder kan styres ved hjælp af sikkerhed på brugerniveau (se bilag 19.6 s. 147). Risikoen for at samtidig brug af statiske elementer i "frontenden" skal nedsætte ydelsen er ringe. Alle medarbejdere befinder sig på et hurtigt netværk (100 Mbit). Generelt håndteres og overføres kun meget små datamængder. Sikkerhed er et relativt lille problem hvorfor konvertering til en .mde fil udelades og til en hvis grad overtages af den nedsatte funktionalitet runtime versionen yder. Samtidig administreres indledningsvist med optimistisk låsning. Udskrivning og netværksadgang styres gennem operativsystemer som p.t. er Windows 2000.

Evaluerings af resultatet kan medføre ændringer i opsætningen ved de efterfølgende slices og en deling af databasen i front- og backend er overordenlig tænkelig.

6.2 Brugergrænsefladekomponenten

6.2.1 Struktur

Brugergrænsefladekomponenten er i dette tilfælde en integreret del af MS Access "frontenden" (se bilag 19.6 s. 147). Formularer er fra Microsofts side koblet til klassemoduler der via peg og klik binder GUI objekter til kode via events.

Gennem forespørgsler er langt de fleste tabeller repræsenteret i formularer eller underformularer, en til en. Der er herved opstået et hierarki på linie med vores objektmodel, hvor vi har valgt at placere aktuel funktionalitet.

Den planlagte struktur for komponenten ses af fællesdiagrammet med funktionskomponent (se Plan B s. 61).

6.3 Funktionskomponenten

Funktionskomponenten er på flere punkter vanskelig at skille fra brugergrænsefladekomponenten da der er en særdeles stærk binding mellem formularerne og deres tilhørende klassemoduler.

MS Access tilbyder flere forskellige steder at placere sin egenudviklet funktionalitet:

- Som løse rutiner i et standardmodul. Dette er den traditionelle strukturerede løsning.
- Som egne klasser i klassemoduler.
- I klassemodulet til formularen, rapporten eller lignende. Dette er oftest anvendt.

Vores system benytter mest muligheden for at lægge funktionaliteten direkte på formularen. Dog har vi lavet enkelte simple standardmoduler, som er nærmere beskrevet i systemdokumentationen (se Systemdokumentation Standard moduler s. 63).

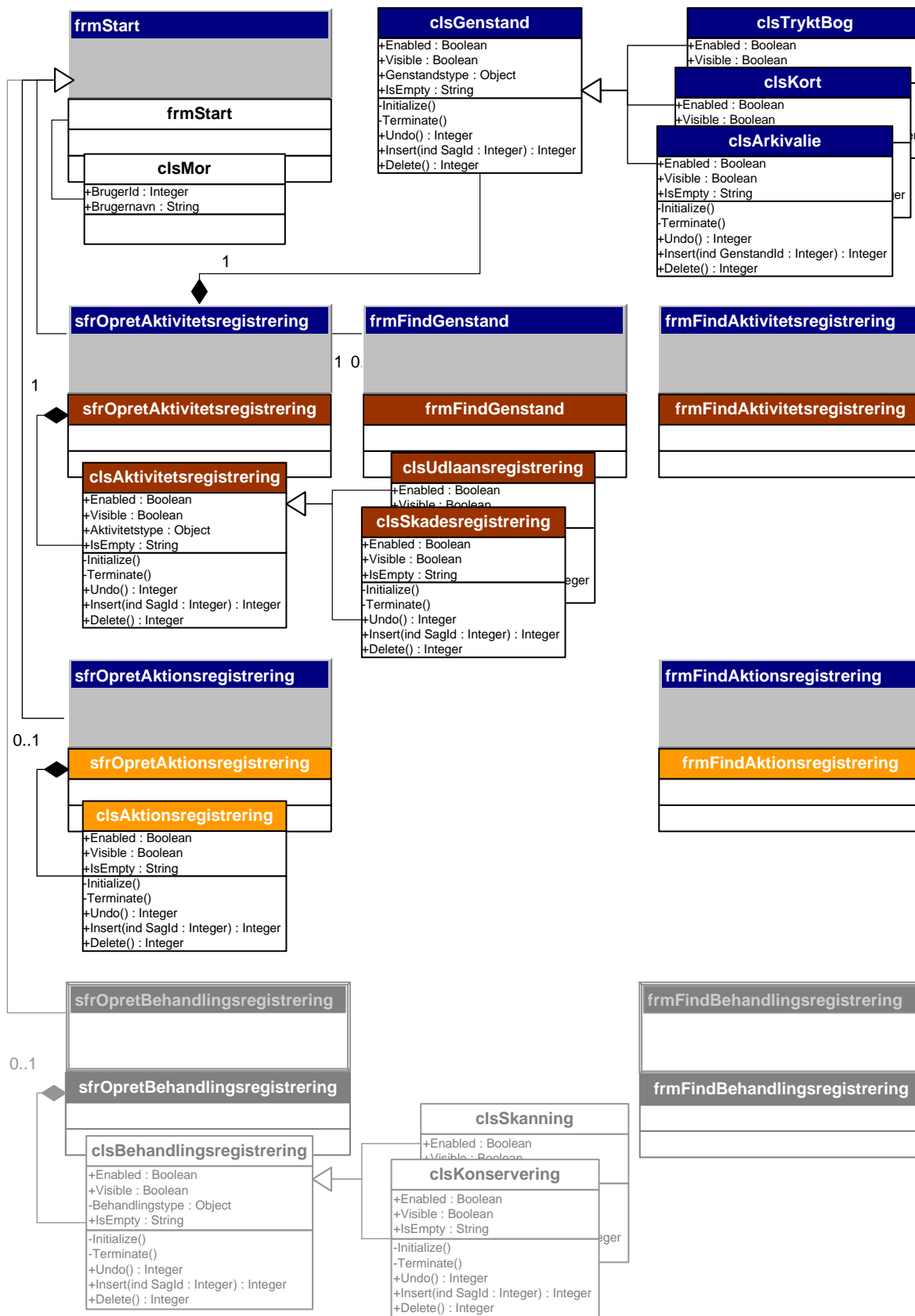
6.3.1 Struktur

I vores første forsøg på at forstå og realisere en struktur for modelkomponenten arbejdede vi på at få trukket funktionaliteten ud af formularerne. I lang tid blev der kodet på en løsningsmodel, der bestod af separate indtastningsvinduer til de tre registreringsformer (aktivitets-, aktions- og behandlingsregistrering) (se Figur 6-4). Der blev imidlertid set for snævert på problematikken, og det viste sig, at indtastning, visning og søgning måtte foregå i selvstændige vinduer. Meget af funktionaliteten og al databehandlingen lå imidlertid uden for det skitserede klassehierarki og kompleksiteten steg voldsomt uden de ønskede visninger af relevante sammenhænge kom i stand.

Efter en reevaluering af projektoplæg og løsningsmuligheder, gik vi tilbage til et tidligere løsningsforslag (se bilag 19.14 s. 189), der bygger på underformularer og direkte dataadgang via DAO.

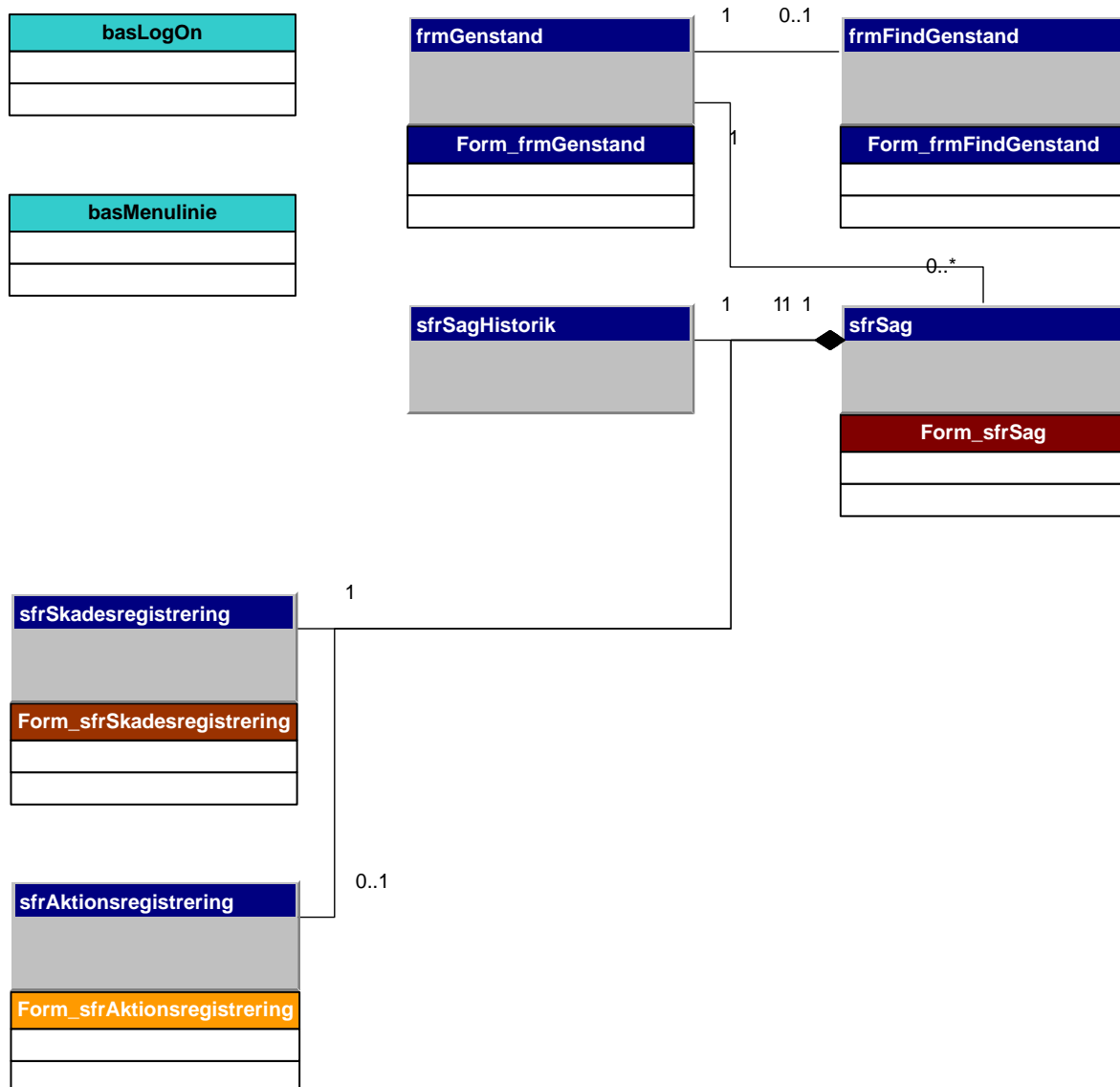
System – slice 1

Plan A



Figur 6-4. Plan A viser brugergrænseflade- funktionskomponenterne og var tænkt som en direkte kodning af modellens funktionalitet i VBA klasserne. Indtastning, søgning og visning skulle foregå i andre vinduer med egne styringsfunktioner - en meget kompleks sag.

Plan B



Figur 6-5. Med plan B startede vi forfra med vores oprindelige koncept, og fokuserede på prototyping og simple løsninger, hvorfor brugerinterface- og funktionskomponentens struktur her kun er løseligt skitseret. Den tætte sammenknytning mellem formularer og klassemoduler (via hændelser) er forsøgt gengivet. Som det ses er der endvidere oprettet to standardmoduler til generel funktionalitet.

Denne version bygger på, at brugerinterfacet og funktionalitet samles ”a la MS Access”. Felterne i brugergrænsefladen bindes gennem forespørgsler til tabellernes data. Ved hjælp af DAO vises recordset direkte i interfacet. Der er ikke tale om objektorienteret programmering i gængs forstand selv om funktionsklassehierarkiet ligner vores model.

Forsøgsvist sikrede vi os at data i formularerne blot kunne vises. Dette lykkedes hurtigt med en vis grad af succes, hvorefter der blev konstrueret opdatering, sletning, brugerstyring og anden funktionalitet i henhold til usecasene. Denne tilgang gav den rigtige følelse af ”prototyping”, og funktionaliteten kunne bedre testes direkte. Slutresultatet blev en mere rigtig løsning, med færre vinduer og et mindre komplekst klassehierarki (se Figur 6-5).

System – slice 1

6.3.2 Standardmoduler

Der er til slicen blevet planlagt og konstrueret to standardmoduler med funktionalitet der skal kunne anvendes af flere klassemoduler og derfor placeres bedst på denne måde.

LogOn

basLogOn

Figur 6-6. Standardmodul rummer kontrol af om brugeren der er logget på arbejdsstationen og så findes i medarbejder tabellen.

Menulinie

basMenulinie

Figur 6-7. Her samles funktionalitet der er fælles for flere menuliner f.eks. standardsøgninger.

6.3.3 Valg af dataadgang

Af hensyn til systemets skalerings- og opdateringsmuligheder havde vi på et meget tidligt tidspunkt omfattende overvejelser om hvilken dataadgang vi skulle vælge. Primært undersøgte vi DAO og ADO, hvor vi ser DAO som den etablerede teknologi i Office og MS Access, mens ADO er den tidssvarende og bredere dataadgang, der også i den nærmeste fremtid vil blive videreudviklet af Microsoft.

En del af disse overvejelser og tilhørende undersøgelser kan ses ud af Hvad er ADO spiken (se bilag 19.7 s. 161) og DML via ADO spiken (se bilag 19.8 s. 167).

På grund af DAO's store integration med MS Access, at der ikke er et umiddelbart relationelt databasesystem der skal skaleres til og at vores projekt har en meget begrænset tidsramme, har vi valgt at benytte DAO som dataadgang. Dette er primært sket ud fra de umiddelbare integrationsfordele med MS Access vi får, så vi ikke skal kode alle tingene selv fra grunden – selv om resultatet sikkert vil være mere i overensstemmelse med best practice.

6.4 Model-/databasekomponenten

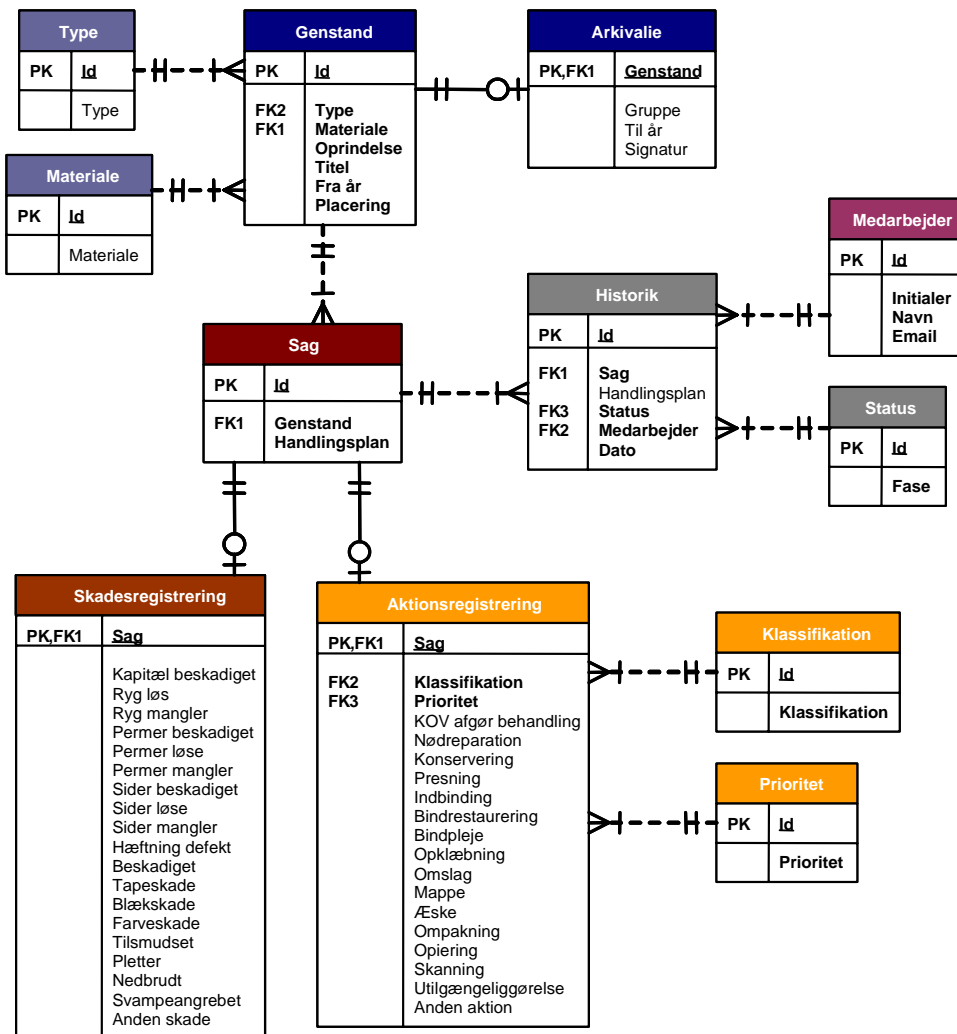
6.4.1 *Transformerer fra UML til logisk datamodel*

Generelt har vi givet alle entiteter en ikke-databærende ("dum") primærnøgle i en separat attribut. Undtagelserne er child-entiteter i en en-til-en relation. Disse vil blive nævnt i det enkelte tilfælde. Entiteternes attributter er navngivet som attributterne i UML diagrammets klasser.

”Dumme” primære nøgler, også kaldet surrogatnøgler er beskrevet nærmere af Dates [*An Introduction to Database Systems, s. 434*] og Elmasri [*Fundamentals of database systems, s. 749*]. Surrogatnøgler er indført i alle tabeller for at lette join og gøre syntaksen letforståelig og ensartet.

Transformeringen fra modelkomponentens klassediagram er foretaget efter disse grundregler:

- 1 - 0...1 relationer (aggregering eller associering) mellem klasser, medfører 1 - 1 relationer mellem to tabeller eller muligvis en sammenlægning til én tabel.
- 1 - 1 relationer (aggregering eller associering) mellem klasser, medfører 1 - 1 relationer mellem to tabeller eller endnu bedre en sammenlægning til én tabel.
- 1 - 0...* relationer (aggregering eller associering) mellem klasser, medfører 1 - * relationer mellem to tabeller.
- * - * relationer mellem (associering) klasser, medfører en indskudt tabel med 1 - * forhold til de deltagende tabeller.
- Nedarvning af konkrete eller abstrakte klasser, medfører en 1 - 1 relation mellem tabeller eller samling af data i en tabel.



Figur 6-8. ER diagram for logisk datamodel. Forkortelser: [PK]: Primær nøgle (eng. Primary Key), [FK<n>]: Fremmednøgle (eng. Foreign Key), [fed]: Nødvendig værdi.

System – slice 1

Klassen 'Kategori' er ikke databærende, så klasserne 'Type', 'Materiale' og 'Genstand' transformeres til de tilsvarende entiteter. Entiteterne 'Type' og 'Materiale' relateres til entiteten 'Genstand' med fremmednøgler i attributterne 'Type' og 'Materiale'. Klassen 'Arkivalie' transformeres til entiteten 'Arkivalie', og relateres til entiteten 'Genstand' i en en-til-en relation med nøgle i attributten 'Genstand' i entiteten 'Arkivalie'.

Klassen 'Sag' transformeres til entiteten 'Sag', der ud over egen primærnøgle i attributten 'Id' indeholder fremmednøgler til entiteten 'Genstand' og (den kommende entitet) 'Handlingsplan' i attributterne 'Genstand' og 'Handlingsplan'.

Klasserne 'Aktivitetsregistrering' og 'Behandlingsregistrering' er ikke databærende, så de transformeres ikke. Det vil sige at klassen 'Skadesregistrering' transformeres til entiteten af samme navne, og relateres til entiteten 'Genstand' i en en-til-en relation med primær nøgle i attributten 'Sag'.

Klassen 'Skade' denormaliseres ind i entiteten 'Skadesregistrering' for at forenkle modellen – vi ser i praksis ikke nogen grund til at have fleksibiliteten i separate entiteter, og det er et udtrykt ønske fra brugerne at have en checkbox for hver eneste attribut i brugergrænsefladen.

Klassen 'Aktion' denormaliseres ind i entiteten 'Aktionsregistrering', der er en transformering fra klassen 'Aktionsregistrering'. Klasserne 'Klassifikation' og 'Prioritet' transformeres til entiteterne 'Klassifikation' og 'Prioritet' og relateres til entiteten 'Aktionsregistrering' med fremmednøgler i attributterne 'Klassifikation' og 'Prioritet'.

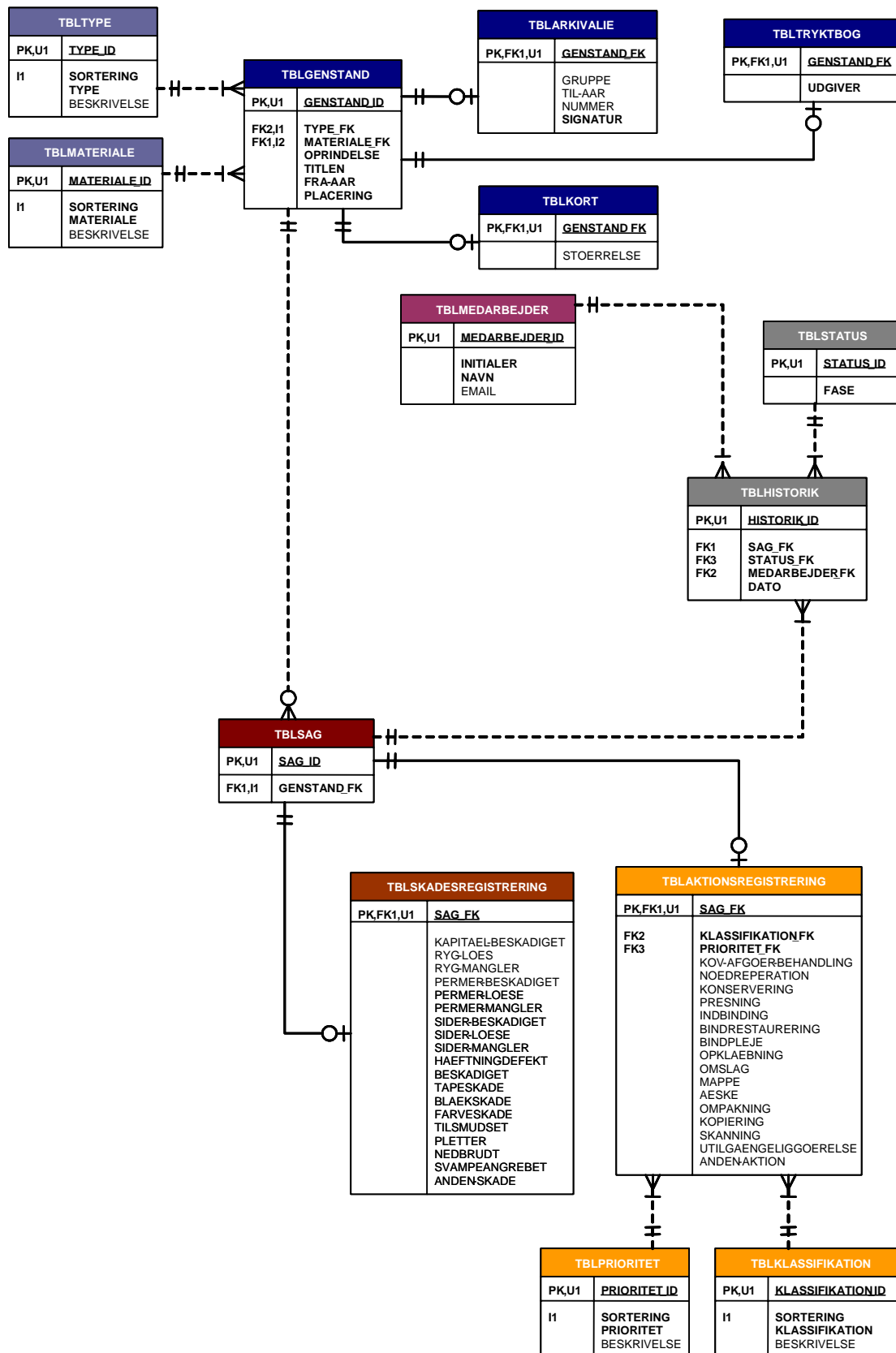
En sags historik i klasserne 'Status' og 'Fase' med en associering til klassen 'Medarbejder' har vi transformeret til entiteterne 'Historik', 'Status' og 'Medarbejder', hvor de ændrede og rokerede navne giver mere mening nu vi tænker i data. Entiteterne 'Status' og 'Medarbejder' relateres til entiteten 'Historik' med fremmednøgler i attributterne 'Status' og 'Medarbejder', og endelig er entiteten 'Sag' relateret til entiteten 'Historik' selv med fremmednøgle i attributten 'Sag'. Derud over har vi oprettet attributten 'Handlingsplan' som skal indeholde fremmednøgle til (den kommende) entitet 'Handlingsplan'.

6.4.2 *Transformerung fra logisk datamodel til fysisk datamodel*

Generelt er alle navne skrevet med versaler, uden mellemrum og uden nationale (diakritiske) tegn. En nærmere beskrivelse af navnestandarden for fysiske data findes i systemdokumentationen (se Systemdokumentation Figur 6-2).

I denne fase har vi set på datatyper, indeksering, referentiel integritet og andre begrænsninger/afhængigheder (eng. Constraints). Desværre er det kun nogle af disse overvejelser der fremgår af diagrammet.

Indekseringen er sket ud fra hvilke SQL udtryk vi har på nuværende tidspunkt, hvor vi primært ser på JOIN, WHERE-klausuler og ORDER BY. Da vi ikke har et særligt godt billede af datamængderne i systemet, har det ikke været muligt at tage hensyn til disse i indekseringen eller forespørgslerne. En nærmere beskrivelse af indekseringer og begrænsninger findes i systemdokumentationen samt i selve databasen. Vi (desværre) bundet af MS Access' begrænsede mulighed for indeksering af en tabel. En fuldstændig indeksanalyse skal derfor foretages forfra hvis systemet flyttes til et rigtigt RDBMS.



Figur 6-9. ER diagram for fysisk datamodel. Forkortelser: [PK] = Primær nøgle (eng. Primary Key), [FK<n>] = Fremmednøgle (eng. Foreign Key), [fed] = Nødvendig værdi, [U<n>] = et indeks på unikke værdier og [I<n>] angiver et indeks på ikke-unikke værdier.

System – slice 1

Transformationen af entiteterne og deres attributter i den logiske model er i praksis en-til-en med lidt anderledes navne som tidligere angivet. Dog har vi set lidt nærmere på om det er nødvendigt at have en værdi for den enkelte attribut.

Der er tilføjet tabeller til genstandstyperne Kort og Trykt bog, da det viste sig nødvendigt.

Det viste sig i forbindelse med Plan A (se Figur 6-4) nødvendigt at have implementerede flertydigheder for aktivitetsregistrering og behandlingsregistrering. Derfor blev der tilføjet tabeller for at systemet kan rumme flertydigheder i aktivitetsregistrering ved udlånsregistrering og i behandlingsregistrering ved skanning.

6.4.3 Normalisering

For at sikre databasens konsistens og forhindre unødigt redundans, er tabelstrukturen kontrolleret med hensyn til normalisering i forhold til de første tre normalformer (ingen tabeller har sammensatte nøgler) [*Fundamentals of database systems –Third Edition, s. 483-493*].

1NF sikrer at kolonner kun indeholder atomare værdier.

2NF sikrer at der ikke forekommer afhængighed af andet end den samlede primærnøgle.

3NF sikrer at der ikke forekommer transitiv afhængighed af ikke primærnøgle kolonner.

At dette ikke har givet anledning til ændring i tabelstrukturen har vi valgt at tage som et udtryk for en hensigtsmæssig opdeling af data på modelniveau og efterfølgende transformering.

6.4.4 Håndtering af inddata

I forbindelse med oprettelsen af en skadesregistrering på en ny genstand har vi brugt en del tid på at oprette en ny Aktivitetsregistrering på en ny Genstand korrekt med automatisk fejlhåndtering. Problematikken er nærmere beskrevet i Transaktioner i MS Access spiken (se bilag 19.9 s. 169).

Validering af inddata sker i realiteten to steder:

- Validering af data i tabellen ved hjælp af MS Access' mulighed for at indsætte en valideringsregel på en kolonne.
- Validering af Nullable kolonner formularen ved hjælp af en MS Access inputmaske på kontrolelementet. Hvis kolonnen er Not Null valideres ved hjælp af en MS Access valideringsregel på kontrolelementet.

Dette er i ret god overensstemmelse med Best Practice.

Reglen for validering af inddata er:

- Indtastede data valideres når feltet forlades.
- Data der tilsammen skal danne en entitet valideres når entiteten skal oprettes, typisk når brugeren klikker på en knap i en formular.

Vi har arbejdet mere detaljeret med inddata end MS Access' inputmasker eller valideringsregler kan håndtere. Tilgangen har været en validering ved hjælp af regulære udtryk sådan som det fremgår af Regulære Udtryk spiken (se bilag 19.10 s. 171).

7 Konstruktion

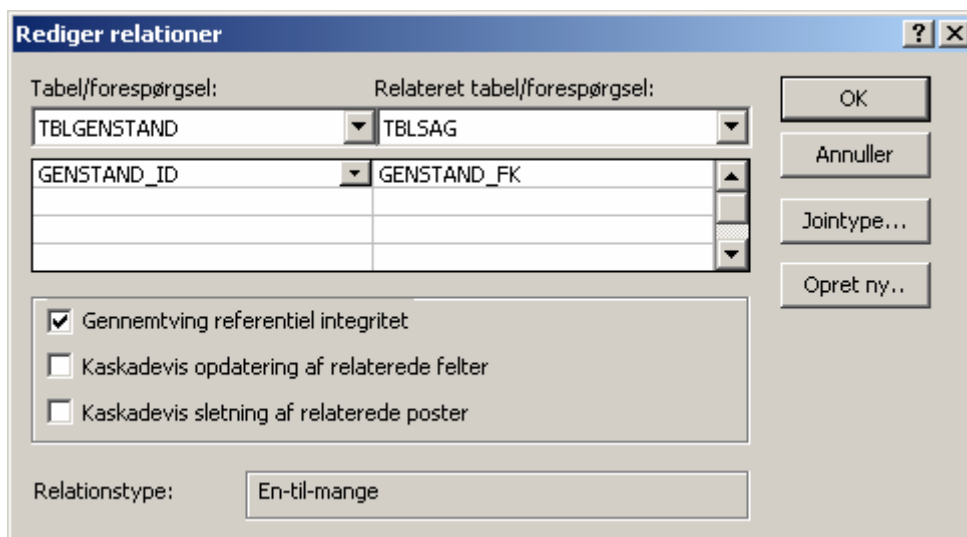
Det er konstruktionsfasens formål, at realisere det analyserede og designede IT-system.

7.1 MS Access applikation

7.1.1 Tabelstruktur

I tidligere afsnit belyses tilblivelsen af datastrukturen (se Model-/Databasekomponenten s.62) og den fysiske realisering følger direkte disse retningslinier. Den fysiske MS Access tabelstruktur ses på Figur 7-2.

Beslutninger om tabelrelationernes egenskaber sættes i et dedikeret vindue (se Figur 7-1) og er for de fleste sat til referentiel integritet. Dette betyder i korthed at der ikke kan oprettes poster på mange-siden (child), hvis der ikke allerede eksisterer en post en-siden (parent) og er dermed en sikring af sammenhængen mellem oprettede poster. Kaskadevis opdatering er ikke relevant i denne applikation og kaskadevis sletning skal man undlade med mindre det er gennemtænkt særdeles grundigt. Det er allerede her muligt at sætte jointypen, men da manipulation og visning af data i denne applikation foregår gennem forespørgsler, er det bedst at differentiere mellem jointyper på forespørgselsniveauet.

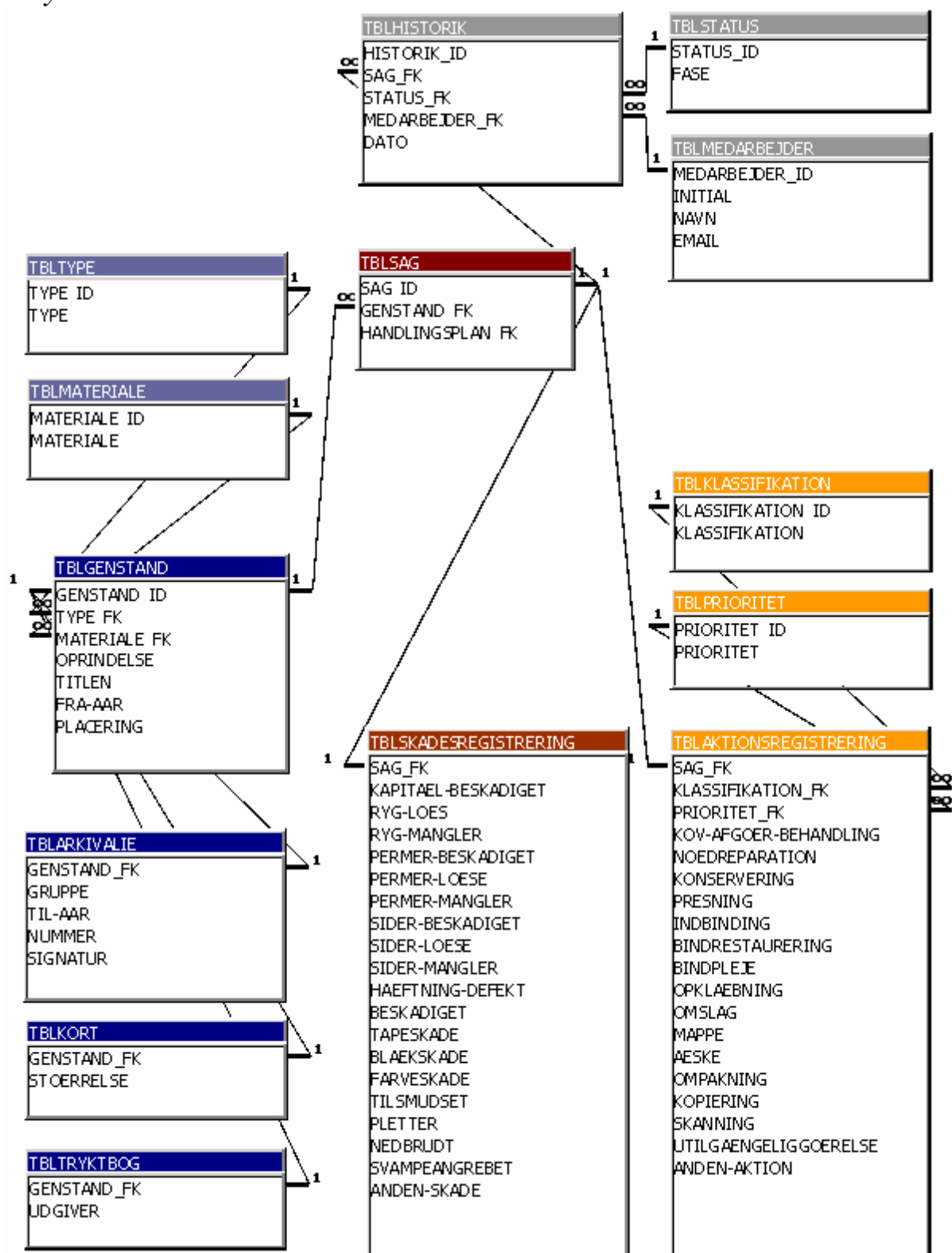


Figur 7-1. Relationsvinduet sættes i forbindelse med den fysiske tabelkonstruktion og specielt referentiel integritet er vigtig at tage stilling til for applikationens enkelte dele.

Surrogatnøgler dannes ved hjælp af MS Access datatypen 'Autonummer', som fortløbende lange positive heltal. Hvis det ønskes er det også muligt at danne unikke tilfældige heltal med Autonummer funktionen. Hvor brugeren gives mulighed for at indtaste prosa tekst, er datatypen Memo, hvilket giver mulighed for op til 65.536 tegn. Dette er gjort for ikke at blive begrænset af datatypen Text, der kun giver mulighed for 256 tegn. Med datatypen OLE objekt, kunne man have givet brugeren mulighed for at indtaste "uendelige" tekster.

Tabeller og relationer er her oprettet og ændret ved hjælp af direkte indtastning, som det formodentligt er normen for MS Access udvikling. Det er dog muligt at gøre dette ved hjælp af DDL scripts hvilket normalt anses for Best Practice blandt professionelle DBA'er.

Fysisk MS Access tabelstruktur



Figur 7-2. I relationsvinduet trækkes relationerne grafisk mellem tabellernes nøgleværdier. Samtidigt bestemmes om der skal være tale om referentiel integritet, kaskadevis opdatering, kaskadevis sletning og hvilken jointype der er tale om.

Datavalidering

Validering af data der indskrives i databasen følger i første række sammen med oprettelsen af den fysiske tabelstruktur.

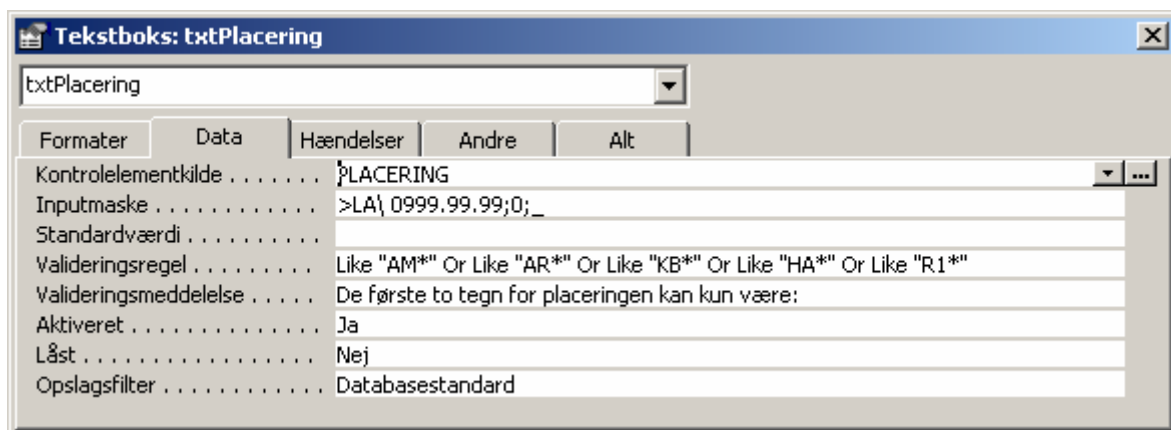
Brug af kombinationsboks med valgmuligheder er første valg, hvor der skal opsamles data der kan udvælges af en overskuelig række af muligheder. Dette kan i MS Access gøres allerede på tabelniveau og sikrer således også valide data i konstruktionsfasen og situationer hvor en udvalgt administrator arbejder direkte på tabellerne.

Indtastningsmasker for felter hvor indtastningsformatet kan fastlægges, hjælper brugeren gennem indtastningen, formaterer data og sender fejlmeddelelser som følge af indtastningsovertrædelser.

I tilfælde hvor der ikke kan fastlægges en korrekt indtastningsmaske, er der mulighed for efterfølgende at validere indtastningen i kode. Accepteres det indtastede ikke, kan der sendes en brugertilpasset fejlmeddelelse der hjælper brugeren frem til en korrekt indtastning (se Hjælp s.78).

Indtastningsmasker, datavalidering og evt. fejlmeddelelser bør naturligvis altid ligge på tabellerne. Det er i første række RDBM systemet der har ansvar for sikring af valide data som over tid kan tænkes at komme fra mange kilder, og ikke kun fra den brugergrænseflade databasen p.t. er forsynet med. Desuden er det formålstjenligt at validere data ved indtastning (se Figur 7-3). Dette giver kortere svartid for brugeren, forhindrer overflødig datatrafik over nettet og unødvendig anvendelse af servertid.

I forbindelse med valgfrie felter med både indtastningsmaske og datavalidering kan det dog vise sig vanskeligt/umuligt at slette en allerede foretaget brugerindtastning, og her må data valideringen udelades på formularniveau eller ske på anden vis.



Figur 7-3. Eksempel på indtastningsmaske, datavalidering og starten af valideringsmeddelelsen der fortsætter yderligere nogle linier.

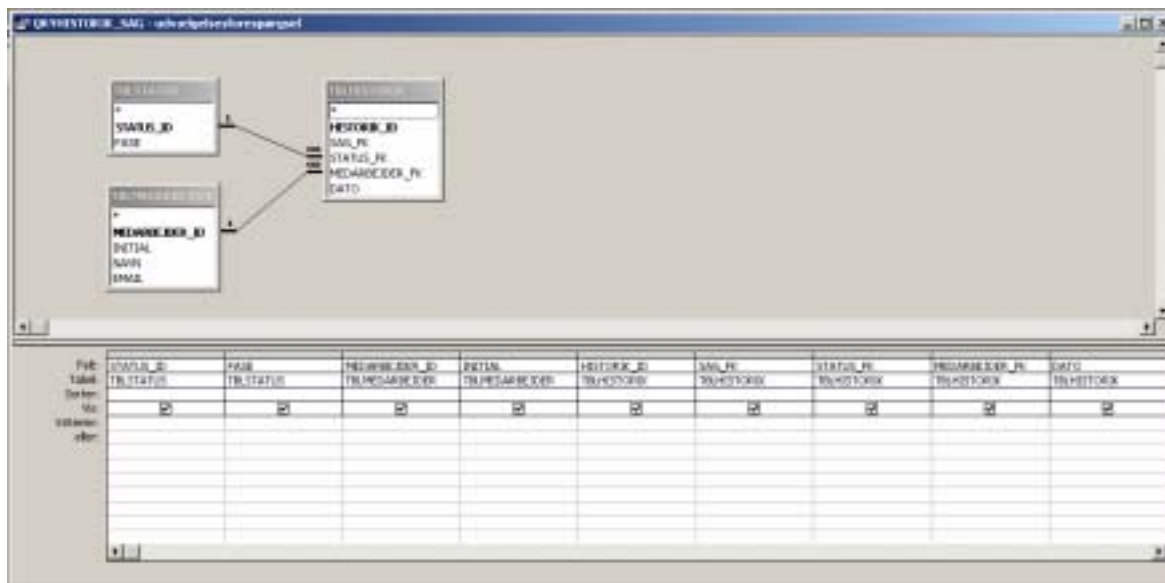
I tilknytning til brugergrænsefladens feltbaserede datavalidering kontrolleres det at samlede datagrupper er til stede før en post afsluttes/opdateres (se Håndtering af inddata s. 66). Dermed mødes brugeren kun af en samlet hjælpdialog med anvisning på hvilke felter der nødvendigvis skal udfyldes før der kan fortsættes med en anden aktivitet. Alternativt vil MS Access selv advare for hvert obligatorisk felt der ikke er udfyldt, hvad der kan virke trættende i længden.

System – slice 1

7.1.2 Forespørgsler

Gennem forespørgsler kan forenings- eller delmængder af data findes frem ligesom der kan slettes, rettes og oprettes poster i tabellerne. Som en særlig feature kan der endvidere konstrueres fleksible krydstableringsforespørgsler i datasammenstillings øjemed (sådanne vil dog ikke efterfølgende kunne migreres til en SQL Server klient/server løsning).

Forespørgsler genereres her i en MS variant af SQL kaldet Jet SQL og kan skrives direkte via DML. Alternativt stilles en grafisk QBE (query by example) editor (se Figur 7-4) til rådighed for både simple og ganske komplekse forespørgsler. Union forespørgsler skal dog altid kodes direkte.



```
SELECT
    TBLSTATUS.STATUS_ID, TBLSTATUS.FASE, TBLMEDARBEJDER.MEDARBEJDER_ID, TBLMEDARBEJDER.INITIAL,
    TBLHISTORIK.HISTORIK_ID, TBLHISTORIK.SAG_FK, TBLHISTORIK.STATUS_FK, TBLHISTORIK.MEDARBEJDER_FK,
    TBLHISTORIK.DATO
FROM
    TBLSTATUS INNER JOIN (TBLMEDARBEJDER INNER JOIN TBLHISTORIK ON TBLMEDARBEJDER.MEDARBEJDER_ID =
    TBLHISTORIK.MEDARBEJDER_FK) ON TBLSTATUS.STATUS_ID = TBLHISTORIK.STATUS_FK;
```

Figur 7-4. Simple forespørgsel i MS Access QBE editor og den resulterende SQL forespørgsel. Det er muligt at fremstille særdeles komplekse sammensatte og/eller indlejrede forespørgsler der endvidere kan gøres parametriserede.

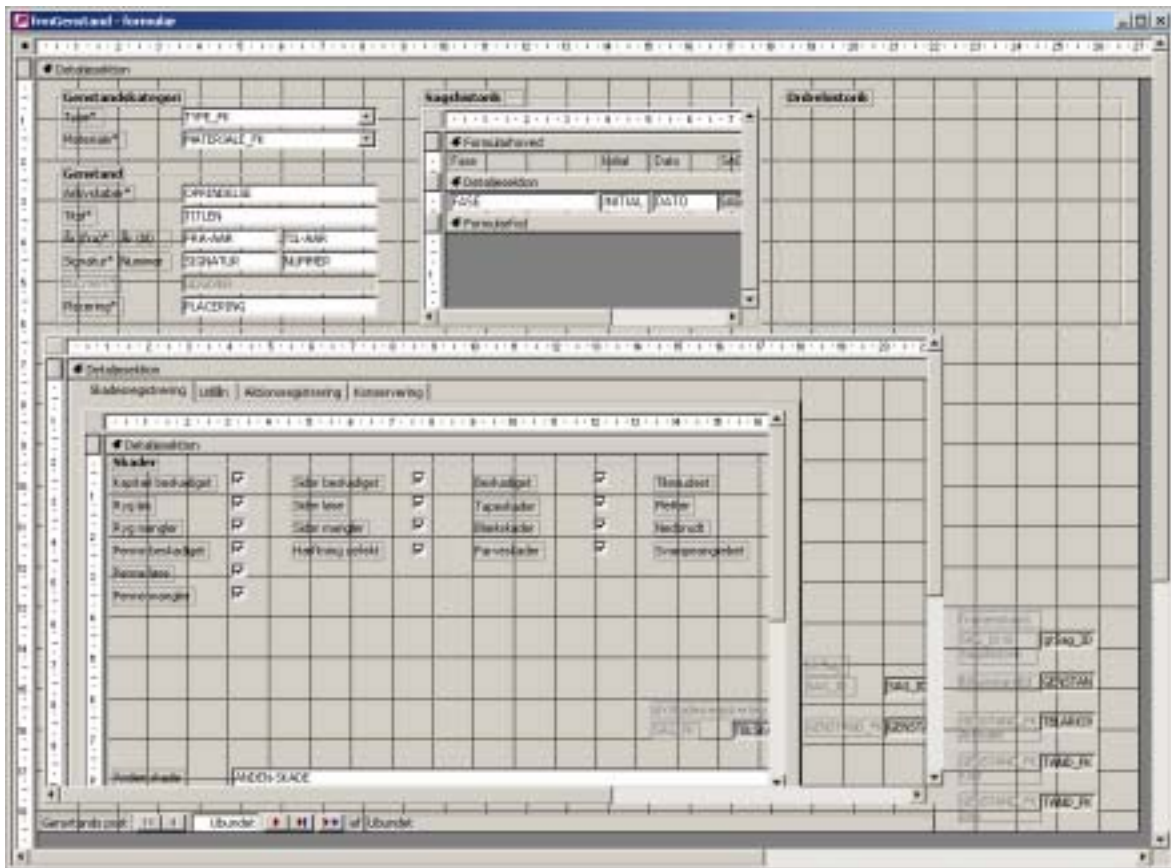
Det er vores mål at have flest mulige forespørgsler implementeret som det MS Access kalder stored queries. Det vil sige forespørgsler der ligger for sig selv på niveau med formularer og rapporter. Dette gør vi for at drage nytte af at MS Access prækompilerer disse udtryk, og derfor vil de blive afviklet hurtigere og ensartet. Forespørgsler der er indskrevet i egenskaberne for de enkelte objekter gemmes ligeledes i prækompileret tilstand, selv om de ikke navngives eksplicit eller ses i det grafiske visningsvindue for forespørgsler.

Specielt i find-formularerne kan vi dog ikke undgå dynamisk eller indlejret (embedded) SQL, med hvad det medfører af tekniske u hensigtsmæssigheder – for eksempel skal udtrykket bygges og oversættes ved hvert kald.

7.1.3 Formularer

I MS Access er det muligt at arbejde direkte på tabeller eller opdaterbare forespørgsler, men denne mulighed låses her for den almindelige bruger og formularer bliver den eneste grænseflade til interaktion og dialog med applikationen. Man kan sige, at opdelingen og normaliseringen af data i tabeller gøres transparent tilgængelig for brugerne, der kun ser én samlet applikation. Fra formularerne oprettes, fremfindes, rettes, slettes og udskrives data.

Opbygning af formularer i MS Access er en simpel drag and drop handling. Efterfølgende kan de enkelte indtastningsfelter, valgbokse og checkbokse (kontroller) bindes til de data fra den underliggende forespørgsel de er udtrykt gennem (tabeller bør aldrig tilgås direkte fra brugergrænsefladen). Simple formularer kan autogenereres og tilrettes, også selv om det er på baggrund af forespørgsler.



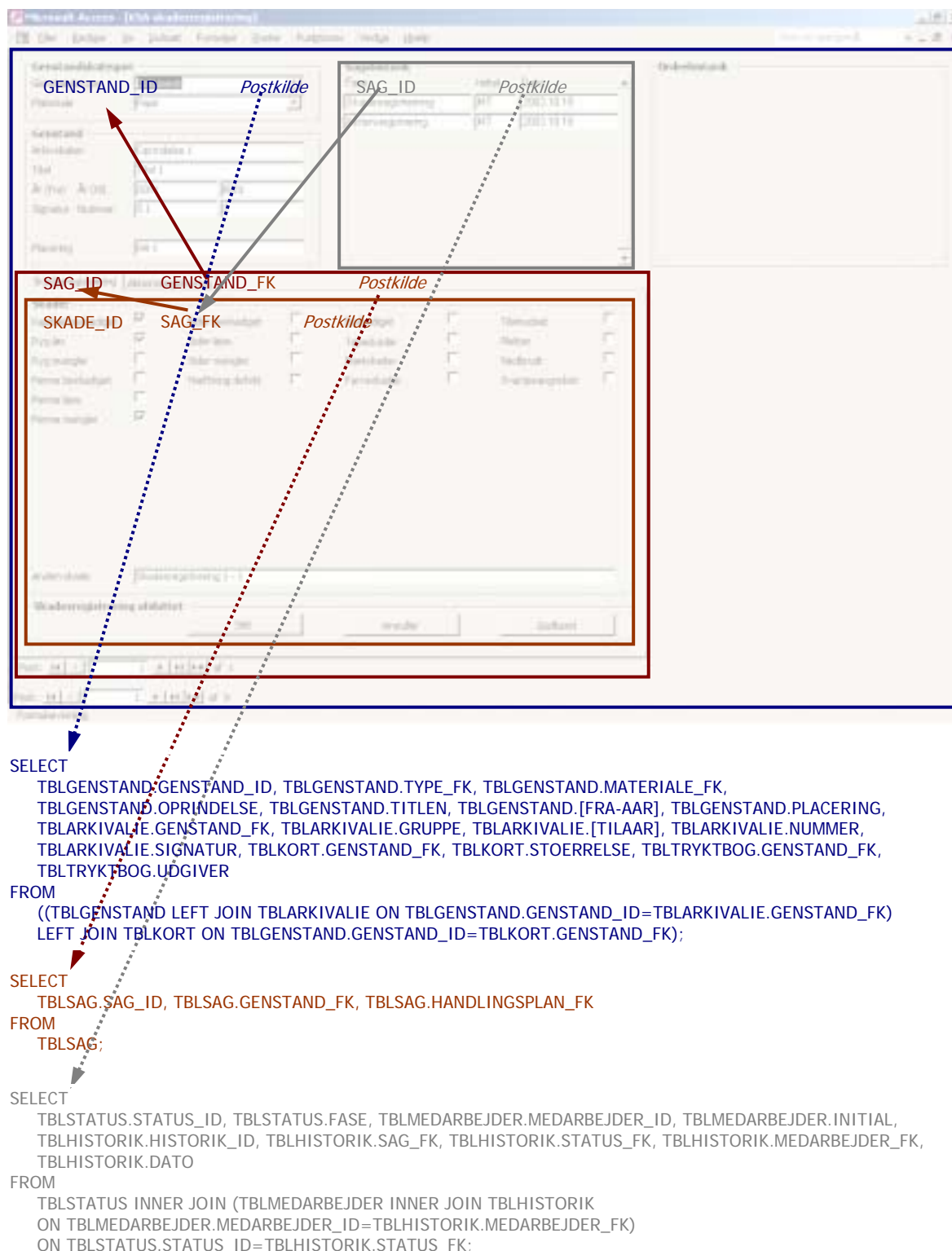
Figur 7-5. Hoved- og underformularer med faneblade samt nøglefelter.

Vi har efterfølgende erfaret, at rigtigt udført tabeldesign med relationsdefinering, indtastningsmasker, datavalidering, hjælpebeskrivelser, datanavne og evt. definering af enkelte tabellers relationer til opslagstabeller betyder, at datafelter, kombobokse, korrekt navngivning og datavalidering trækkes med direkte over i formularen – også som her via forespørgsler. En kæmpe lettelse hvis man ellers har gjort sig fordelene klar.

Underformularer er en enkelt og effektiv måde at opdele en hovedformular modulært (se Figur 7-5) og anvendelserne rækker langt længere end blot visning af 1-* forhold. De mange bindinger mellem formularerne kan naturligvis skade ydelsen og ved fanebladene kan det være en mulighed kun at binde dem der er aktiveret.

System – slice 1

Gennem den enkelte formulars datakilde bindes de til deres respektive forespørgsler (se Figur 7-6), der er prækompileret for hurtig afvikling.



Figur 7-6. Skematisk visning af bindingsmekanismen mellem hoved og underformularer. Ved opdatering af Genstand bindes sag via GENSTAND_ID/ GENSTAND_FK, skadesregistrering bindes via SAG_ID/SAG_FK og sagshistorikken bindes via SAG_ID/SAG_FK. Alle faneblade har deres egen underformular der bindes til SAG_ID.

Til specielle søgninger har vi arbejdet en del på formularen frmFindGenstand. Grundtanken er at brugeren indtaster og vælger kriterierne for søgningen, og når der klikkes på knappen 'Find' dannes en dynamisk SQL forespørgsel.



Figur 7-7. Formularen frmFindGenstand vist i design.

Den dynamiske forespørgsel dannes i to tempi (se detaljer i koden for den private property SQLudtryk)

1. SELECT og FROM forudsætninger
2. WHERE klausuler

Inden selve forespørgslen dannes bestemmer funktionaliteten tabelnavnet på den tabel der indeholder den aktuelle genstandstype.

SELECT og FROM forudsætningerne dannes ud fra genstandstypen så der kan joinses korrekt.

WHERE klausuler indsættes ud fra indholdet af bestemte kontroller på formularen. Her har vi stor nytte af vores kodestandard.

```

For Each objKlausul In Me.Controls
    Select Case Left(objKlausul.Name, 3)
        Case "cbo"
            If (IsNull(objKlausul.Value) = False) Then
                If (strWhere = "") Then
                    strWhere = _
                        "WHERE" & vbCrLf & vbTab
                Else
                    strWhere = _
                        strWhere & "AND" & _
                        vbCrLf & vbTab
                End If
                strWhere = _
                    strWhere & objKlausul.Tag & " = "
    
```

Forespørgslen danner et DAO recordset

```
Set rstGenstand = CurrentDb.OpenRecordset(SQLudtryk)
```

System – slice 1

Indholdet af recortsettet sættes ind i listboksen nederst i formularen

```
With rstGenstand
  Do Until .EOF
    lstSoegeresultat.AddItem Item:=_
      .Fields(0).Value & ";" & _
      .Fields(1).Value & ";" & _
      .Fields(2).Value
    .MoveNext
```

Desværre virker denne formular ikke ved projektets afslutning, men det er én af de ting KSA vil arbejde videre med.

7.1.4 Navigationsknapper

Standard navigationsknapper lader meget tilbage at ønske med hensyn til virkemåde. F.eks. er det muligt ved kontinuerlig bladren fremad at fortsætte direkte ind på en ny post, hvad der er u hensigtsmæssigt og skaber forvirring. Ligeledes virker det ulogisk, at man i recordsættets yderpunkter fortsat får stillet muligheden for at gå til første eller sidste til rådighed. Sidst, men ikke mindst kan indledende tekst, placering, størrelse og knapsymbolerne ikke tilpasses.

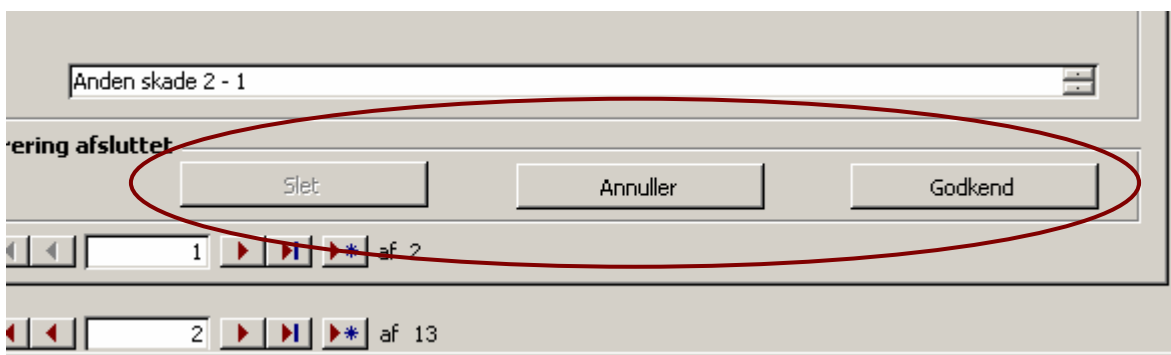
Denne funktionalitet er derfor konstrueret fra bunden. Knappernes størrelse og symboler er gjort en anelse større/tydeligere og det har været muligt at placere dem mere hensigtsmæssigt og forsyne dem med relevant tekst. Derudover vises der i tilknytning til navigationsknapperne, om der er tale om en oprettelse, hvilken søgningsvisning der ses og om der evt. er tilknyttet filtrering (se Figur 7-8). Det er vores håb at disse forbedringer sammen og hver for sig giver tilstrækkelig mulighed for at navigere og orientere sig hensigtsmæssigt i forenings- eller delmængder af poster.



Figur 7-8. Der er her foretaget en standardsøgning på "alle arkivalier" og yderligere er der filtreret på et felt i genstandsformularen hvorved 11 arkivalier er fremfundet. Endvidere er der foretaget en oprettelse af en ny sag/skadesregistrering.

7.1.5 Anden knapfunktionalitet

For hver status godkendelse gælder, at et tilhørende sæt knapper giver en visuel kontrol med sletning, annullering og godkendelse (se Figur 7-9). Dette er tænkt som en hjælp til brugeren, da MS Access normalt blot gemmer en valid post ved udgang af den formular/underformular hvor der er foretaget en indtastning.



Figur 7-9. Knapfunktionaliteten viser her efter en indtastning at der er mulighed for enten at annullere eller gemme. Efter valget ændres knapmulighederne i overensstemmelse med forholdene.

System – slice 1

7.1.6 Menulinie

Udvikling af en brugertilpasset menulinie i denne applikation er ikke kun en hensigtsmæssighed, men en direkte nødvendighed da der er krav om en MS Access løsning der kan afvikles som runtime version. Runtime versioner viser ikke standardmenuliner.

Brugerdefinerede menuliner kan endvidere lette brugerens interaktion med applikationen og forhindre, at de kommer galt af sted ved at klikke på ”forkert” funktionalitet. Samtidigt nedsættes kompleksiteten da kun relevante valgmuligheder vises.

Design af brugermenuliner i MS Access bryder lidt med andre indbyggede konstruktionsmetoder og højreklikks menuen (genvejsmenu) er direkte umulig at finde ved hjælp af intuition. En fordel er dog, at den kan specialiseres til hvert enkelt felt og dermed give reelle kontekstuelle valg. Ligeledes kan menu- og værktøjslinier tilknyttes den enkelte formular.

I denne fase er der udviklet en menulinie med et par enkelte værktøjs elementer for filtrering til og fra. Disse kunne hentes via menuvalg, men for let at kunne se om filtrering er slået til eller om der arbejdes i det fulde recordset er den visuelle tilstedeværelse nødvendig.

En hovedaktivitet fra menulinen er visning af gemte forespørgsler der hentes via kald til et kodet menuliniemodul (se basMenulinie s. 79). Ligeledes vil der i en senere version blive givet mulighed for udskrivning og overførsel af sager og anden interaktion med den kommende handlingsplansformular.



Figur 7-10. Hovedformens brugermenulinie rummer kun et reduceret specialtilpasset funktions sæt. Der vælges her en søgning på alle genstande der er vinduets default og som man ofte må vende tilbage til efter mere specialiserede søgninger.

7.1.7 Styring af brugeradfærd

Som det fremgår af det ovenstående er der taget betydelige skridt til at styre, rydde vejen og gøre brugerens interaktion med applikationen så intuitiv og enkel som muligt.

Den endelige styring kommer til at bestå af fire lag:

1. Styring og visning af navigationsmuligheder, søgning og filtrering.
2. Hel eller delvis styring af brugeren gennem slette-, annullerings- og oprettelsesfaser.
3. Låsning af mulig interaktion med fanebladene på baggrund af status er en overordnet regel der sikrer rigtig rækkefølge for tilgang af funktionaliteten på fanebladene.
4. Låsning og tildeling af rettigheder til applikationens enkelte tabeller, forespørgsler og formularer i henhold til aktørbeskrivelsen (se Aktører s. 39).

OBS. Dette er planlagt til at foregå i slice 3 hvis konstruktionen er så fremskredent at databasen kan tages i brug af flere brugere end Anni Bentsen.

Fordelt på aktører ser mulighederne for interaktion sådan ud:

Brugere (alle)

- Fri adgang via navigationsmuligheder, søgning, filtrering til læsning og udskrivning (når det implementeres) i alle vinduer.
- Oprettelse af genstande hvor det sikres at der kun kan startes med valg af type og materiale. Herefter bliver de generelle og specielle genstandsdatafelter vist og åbnet. Genstanden valideres undervejs og totalt før udgang.
- Genstande er herefter altid åbne for rettelser.
- Mulighed for oprettelse og udfyldning af skadesregistreringer. Skadesregistreringen valideres undervejs og totalt før udgang.
- Der er mulighed for at rette eller slette indtil der oprettes en aktionsregistrering.

Ansvarshavende arkivar, værkstedsansat og administrator

- Ved oprettelse af aktionsregistrering lukkes skadesregistreringen for rettelser. Aktionsregistreringen valideres undervejs og totalt før udgang.
- Der er mulighed for at rette eller slette en aktionsregistrering indtil den overliggende status (Preordre) nås.

7.1.8 Sikkerhed

Jævnfør Sikkerhed spike (se bilag 19.5 s. 147) opfattes sikkerhed i forbindelse med denne applikation primært som fysisk-, netværks- og som datasikkerhed i multibrugersammenhæng. Ingen af disse former tages op i denne slice, men sikkerhed på brugerniveau er gennemgået i spiken for fremtidig anvendelse.

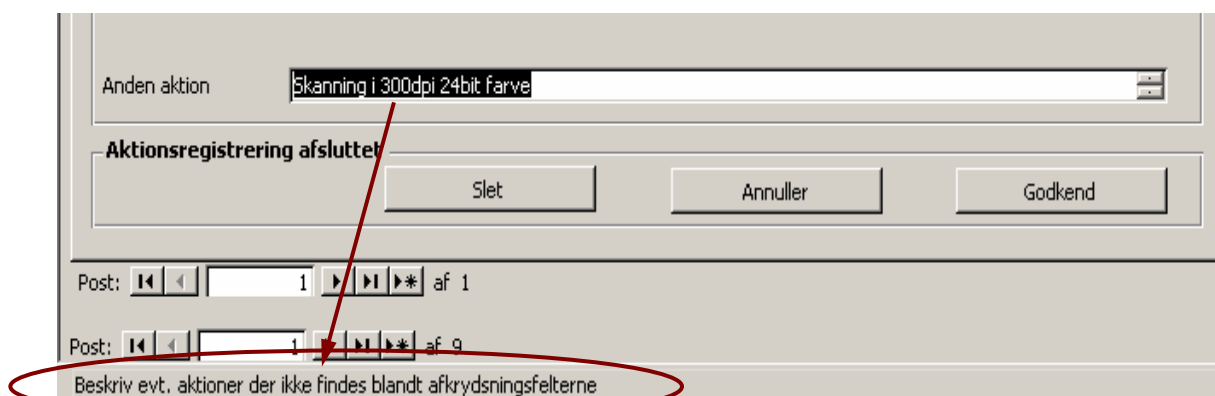
System – slice 1

7.1.9 Hjælp

Hjælp i MS Access er inddelt i flere typer der hver især og sammen kan bringes til at yde brugeren den nødvendige hjælp i tilfælde hvor brugergrænsefladen eller funktionaliteten ikke er fuldt så intuitiv som ønsket.

Statusliniehjælp

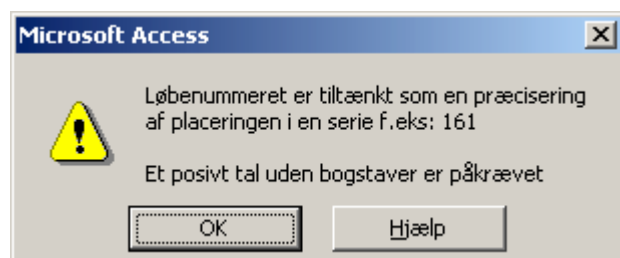
I denne applikation har vi i første række valgt at vise statuslinehjælp. Denne ses i applikationens underkant, og viser en hjælpelinie tekst for hvert objekt i brugergrænsefladen, der kan have fokus (se Figur 7-11). Det er en enkelt og altid tilstedeværende hjælp der kort husker brugeren på indhold eller formål med et objekt uden at kræve opmærksomhed når den ikke bruges. Den er endvidere enkel at aktivere og ændre.



Figur 7-11. Statusliniehjælp i funktion.

Tekstdialog som følge af fejlindtastning eller -håndtering af applikationen

Ved validering af feltindtastning eller fejlbetjening af applikationen hjælpes brugeren til at korrigere fejlen ved hjælp af dialogbokse, der mere detaljeret end statuslinieteksten, viser hvordan indtastningsregler overholdes.



Figur 7-12. Det bør ikke være muligt at indtaste ugyldige datatyper i databasen. Valideringsregler giver mulighed for at sikre dataintegritet og bruges til om nødvendigt at udløse vejledende fejlmeddelelser til hjælp for brugeren.

Tekst tips

Kan forekomme som hjælp til knapfunktionalitet.

MS Access hjælp (F1)

Ikke planlagt som del af dette projekt, men en nødvendighed for det færdige IT-system. Emnet er behandlet i Hjælp spiken (se bilag 19.13 s.183) og strukturen til MS Access lidt kluntede kontekstuelle hjælp er forsøgsvist realiseret, men ikke indbygget.

7.1.10 Standardmoduler

basLogOn

basLogOn

Figur 7-13. Standardmodul der rummer kontrol af om brugeren der er logget på maskiner står i TBLMEDARRBEJDER og dermed har rettigheder til at slette, ændre eller oprette samt hvilke initialer der skal anvendes til historikken.

Der er behov for at vide hvem der er den aktuelle bruger af systemet i den kørende proces, derfor gemmes brugerens primærnøgle MEDARBEJDER_ID i en for modulet privat variabel.

Hvis denne variabel er tom – hvilket den jo er til en start – hentes brugeres initialer fra Windows miljøvariablen %USERNAME%

```
LogonNavn = UCase(Environ("USERNAME"))
```

og så findes brugerens nøgle

```
lngBrugerID = DLookup("medarbejder_id", "tblMedarbejder", _
    "INITIAL='" & LogonNavn & "'")
```

basMenulinie

basMenulinie

Figur 7-14. Her samles funktionalitet der er fælles for flere menulinier f.eks. standardsøgninger.

Til hvert menupunkt er der en rutine, som typisk har denne opbygning:

```
With Form_frmGenstand
    If (.OpretNyGenstand = "" And _
        Form_sfrSag.OpretNySag = "") Then
        .RecordSource = "QRYGENSTAND_ARKIVALIE_KORT_BOG"
        .AktuelGenstandSoegning = ""
        .Requery
    End If
End With
```

Grundidéen er at formularens datakilde skiftes ud med en anden forespørgsel (Stored Query), og så opdateres formularens indhold med metoden 'Requery'.

System – slice 1

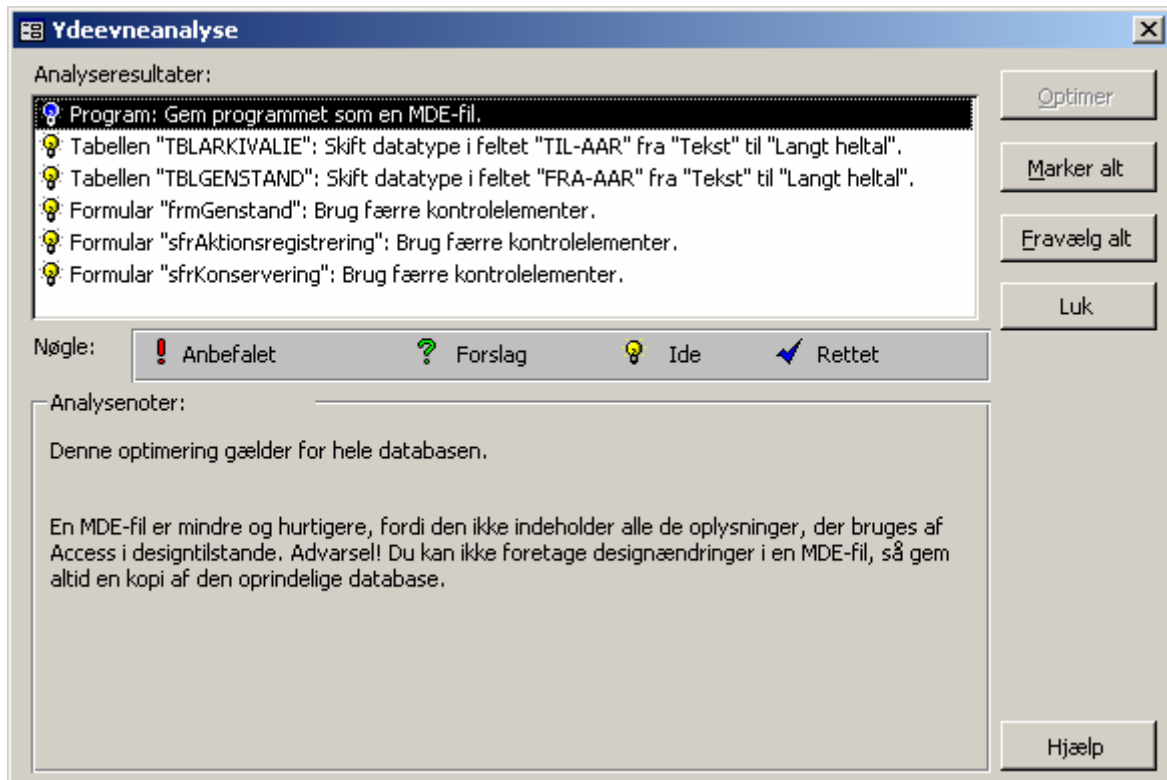
7.1.11 Programfærdiggørelse

Afslutningsvist gennemgås indstillingspanelet (se Figur 7-15) og det sikres bl.a. at databasen åbnes som delt og efter endt brug komprimeres og repareres. I start panelet stilles opstartparametre og alle unødvendige indstillingsmuligheder der kan skabe brugerforvirring slås fra.



Figur 7-15. Programmet får her titel, ikon og andre opstartparametre sat eller begrænset. OBS det er stadig muligt at åbne databasevinduet hvis SHIFT holdes nede under opstart i det fulde MS Access program. Denne mulighed eksisterer ikke for brugerne, der ser applikationen gennem runtime versionens restriktioner.

MS Access har flere værktøjer til analyse af tabeller, ydeevne og strukturvisning. I denne fase har vi gennemført en ydeevneanalyse. Heldigvis var der ikke meget at komme efter.



Figur 7-16. MS Access indbyggede analyseværktøj kan være en stor hjælp og automatisk finde problemområder der bør kontrolleres yderligere (her er alt dog OK).

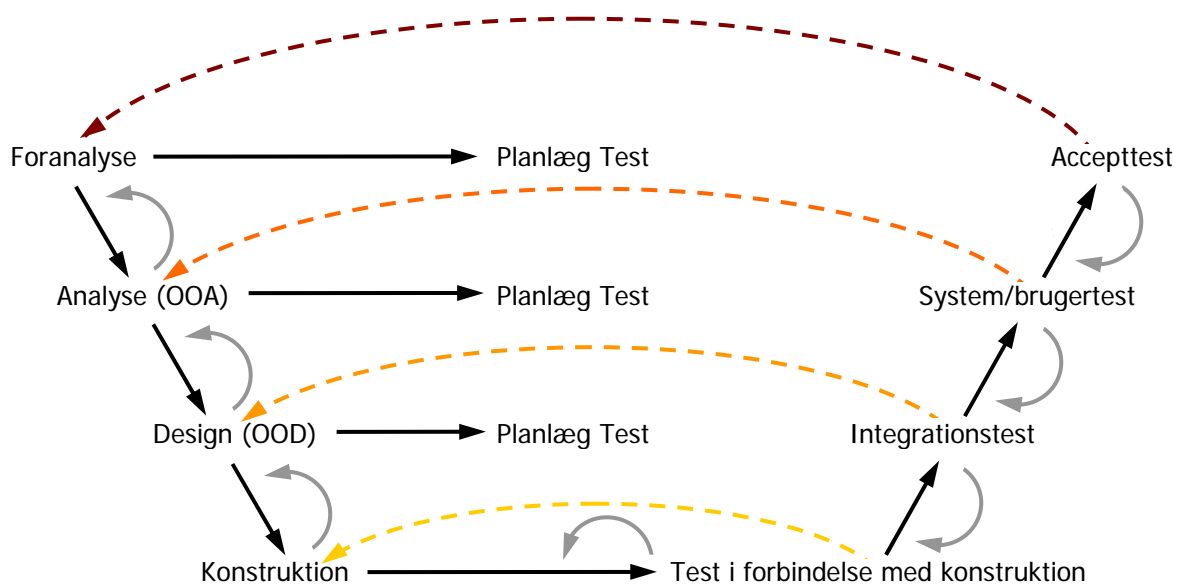
8 Test

Det er testfasens formål, i henhold til testspecifikationen (se bilag 19.3 s. 135), at afvikle planlagte test for de enkelte slices. Mængden af fejl skulle herved begrænses til et minimum og sikre, at IT-systemets anvendelighed under realistiske forhold overholder kravspecifikationen inden implementering og overdragelse.

8.1 Testplan

For dette projekt er der af tidsmæssige grunde ikke planlagt et fuldt testforløb for de enkelte slices. Det har alligevel været vores ambition at planlægge og teste efter en reduceret V-model for at vise modellen i anvendelse og selv gøre os praktiske erfaringer.

V-testmodel



Figur 8-1. V-testmodel der her er gengivet frit efter Poul Staal Vinje [Projektledelse af systemudvikling, s. 141], viser at planlægning af test sker under udviklingen (ned langs venstre ben) og at de efterfølgende udføres i modsat rækkefølge efter konstruktionen (op langs højre ben). De mindre pile illustrerer at der gennemføres reviews og at deres resultat kan få projektet til at returnere til en tidligere fase. Vi er uenige i relevansen af at kunne gå tilbage fra en testfase til en tidligere, og mener i stedet at hvis en brugertest fejler fordi der ikke er overensstemmelse mellem hvad brugerne ønsker og det udviklede produkt, må man jo naturligvis helt tilbage til den fase hvor fejlen opstod (det har vi vist med farvede stiplede linier).

V-modellen er valgt til dette projekt, fordi planlægning af test tidligt kan medvirke til at objekter/data/usecasemodeller er forståelige og konsistente. Det er endvidere uvurderligt at man under konstruktionsfasen har faste veldefinerede mål at programmere op mod.

Der er for denne projektopgave medtaget accept- og system/brugertest. Integrationstest er derimod ikke beskrevet separat, men en del er medtaget i accepttesten.

Under konstruktionen der er forløbet som prototyping er der ikke eksplicit beskrevet testmål for funktionaliteten, men fasen som sådan kan opfattes som et konstant testforløb. Kodet test (Unittestning) af kode er en særlig effektiv kvalitetssikring, der dog ligger helt uden for dette korte projekts rammer.

8.2 Brugertest/systemtest

8.2.1 Test procedure

Annie Bentsen (AB) som ikke tidligere havde arbejdet med udviklingsprototyperne fik installeret en genvej på sin arbejdsstation og efter en kort instruktion i brugergrænseflade og virkemåde blev hun bedt om at indtaste ti tidligere indberettede skadesregistreringer. Forløbet blev holdt op mod testspecifikationens brugertestsценarie for skadesregistrering (se bilag 19.3 s. 140) og hun kommenterede og spurgte til funktionalitet undervejs.

Da AB også har adgang til at registrere aktionsregistreringer agerede hun ligeledes testbruger for denne del, der blev gennemført efter samme principper som ovenstående.

8.2.2 Test resultater

Skadesregistrering

Ønsker til rettelser:

- Markering af obligatoriske felter, evt. med *.
- Mulighed for at påvirke valget af genstandsmaterialet på baggrund af typevalget. For indbundne genstande gælder f.eks. at det her er mere interessant at vide om indbindingsmaterialet tilhører en af de forskellige indbindinger: Skindbind, pergamentbind, papirbind eller subsidiært om bindet evt. helt mangler.
- Direkte kombinationsboksvalg af kort/tegningens størrelse - gerne med valgbeskrivelse direkte i kombinationsboksmenuen.
- Mulighed for på forhånd at fastlægge rækkefølgen i kombinationsbokse.
- Bedre tilpassede labels i forbindelse med de enkelte genstandstyper, f.eks. skal der for bøger ikke stå: År (fra) men År (udgivelse) og for kort/tegninger blot År.
- Interfacet er lidt for stort til korrekt visning på skærm.

De ovennævnte indsigelser, der er både relevante og forholdsvis små, forsøges indpasset i slice 2.

Aktionsregistrering

Efter gennemgangen af skadesregistreringen er aktionsregistreringen nærmest selvforklarende og der var ingen umiddelbare uklarheder eller ønsker til forbedringer.

Observationer

- Det blev klart at der er brug for ganske meget instruktion af de enkelte brugere og at en indbygget hjælpefunktionalitet er nødvendig inden en endelig version implementeres. Selvom begreberne er velkendte virker de faste/obligatoriske formater snærende, fordi man tidligere af dovenskab kunne undlade at indføre data der måske først skulle findes frem. Det er tydeligt at informationen på de oprindelige skadesregistreringer netop er ukorrekt, indforstået og mangelfuld. En senere kobling til den reviderede arkivdatabase kan afhjælpe dette problem ved at overflødiggøre indtastning for allerede registrerede genstande.

System – slice 1

8.3 Accepttest

8.3.1 *Test procedure*

Test skemaet for de tests der er relevante for denne fase blev gennemløbet og de enkelte punkter blev taget op endnu en gang - et for et.

8.3.2 *Test resultater*

For de dele der er indeholdt i slice 1 var der kun ganske få mangler eller fejl.

Test af hard- og software

- 1.5 Interfacet er lidt for stort i både højde og bredde.

Test af generelle krav til applikationen

Alt ok.

Test af skadesregistrering (faneblad)

- 3.9 Grundlæggende søgning er ok, men avanceret søgning mangler.

Test af aktionsregistrering (faneblad)

- 4.8 Grundlæggende søgning er ok, men avanceret søgning mangler.

Tilpasning af brugergrænsefladen der også var nævnt ved brugertesten, forsøges indpasset i slice 2. Vinduerne (formularerne) til specielle eller avancerede søgninger må vente til en senere version.

9 Implementering

Det er implementeringsfasens formål at installere, afprøve, introducere IT-systemet til rekvirenten og dennes brugere.

9.1 Installation og afprøvning

Det er indledningsvist aftalt at slice 1 konstruktionen ikke implementeres for andre end udvalgte testpersoner hvorfor følgende oprindelige punkter udgår fra fasen:

- Introduktion
 - Forberedelse
 - Præsentation

I forbindelse med accept- og system/brugertest blev der foretaget en udokumenteret integrationstestning af slice 1 applikationens virkemåde ved installation på filserveren og eventuel forandring som følge af runtimeversionens begrænsninger.

Der blev konstrueret genveje som efterfølgende kan kopieres og bringe MS Access programmet til at starte på lokale arbejdsstationer. I den forbindelse var der lidt komplikationer med indskrivning af genvejsargumenter.

Vi ønsker som udgangspunkt at alle brugere, uanset om de har installeret en version af MS Access, skal anvende runtimeversionen. For enkelte brugere kan det ellers betyde at programmet åbnes i en mindre restriktiv version, indtil sikkerhed på brugerniveau gennemføres.

System - slice 2

Handlingsplan

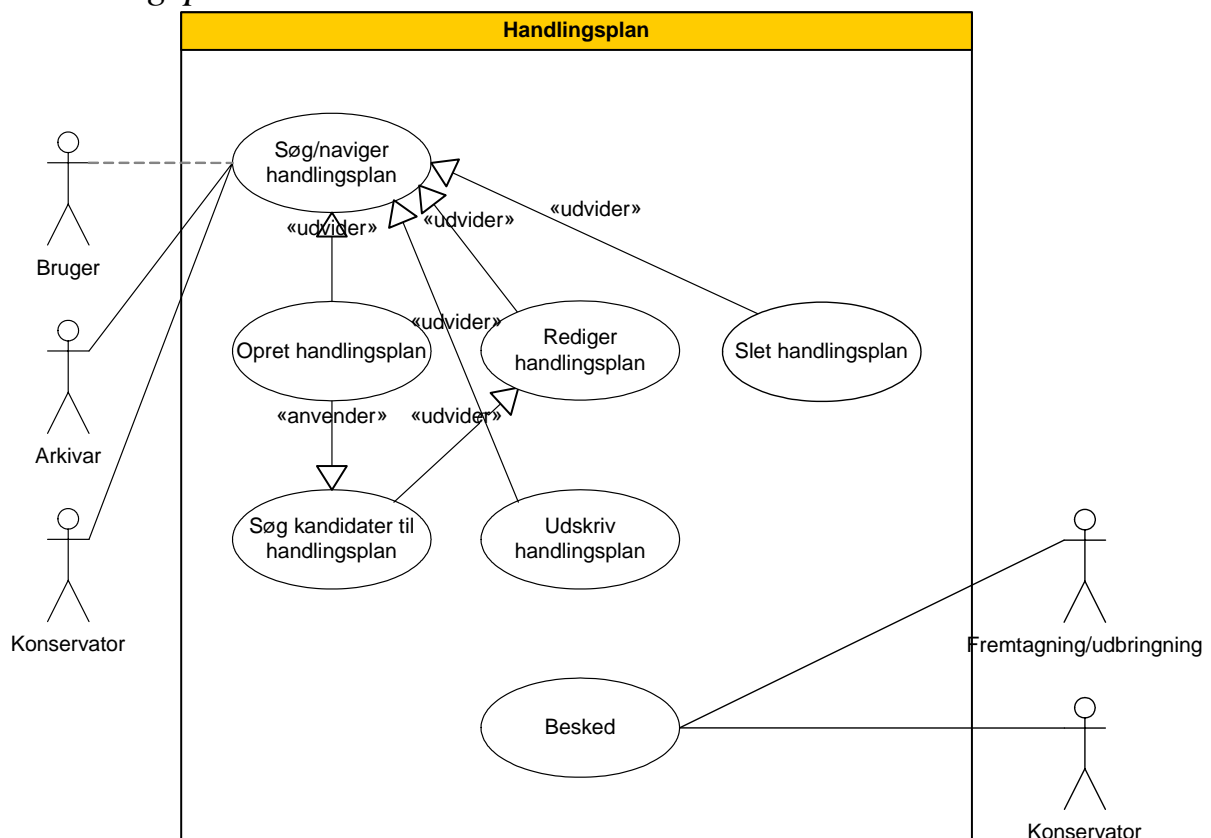
Slice 2 fokuserer primært på handlingsplandelen, og i overensstemmelse med de tilbageværende resurser, projektplanen og prioriteringen fra kravspecifikationen, er det for denne slice besluttet at lægge hovedvægten på OOA, OOD og konstruktion af tabelstrukturen, således at slicens top og bund færdiggøres helt. Det overlades herefter til KSA at konstruere den foreslåede funktionalitet. Der medtages og konstrueres ønsker og rettelser fra slice 1, sådan at denne bliver så færdig som mulig til projektoverdragelsen, men der er ikke planlagt aktiviteter i form af formaliseret test eller implementering.

10 Objekt Orienteret Analyse

10.1 Anvendelsesområdet

10.1.1 Brug (aktører/usecases/scenarier)

Handlingsplan



Figur 10-1. Man kommer frem til usecasen for handlingsplan gemmen logon og søg/naviger genstand usecasene der er default ved åbning af applikationen. Herefter åbnes Søg/naviger handlingsplan, og det bliver muligt at læse, oprette, redigere eller slette afhængigt af rettighederne beskrevet i aktørlisten

OBS. Efterfølgende gengiver kun et eksempel på en usecase. Se alle beskrivelserne i systemdokumentationen (se Systemdokumentation Handlingsplan s. 24)

Besked (fremtagning/transport)

Brugsmønster:

I forbindelse med godkendelse af en iværksat status udskrives en fælles besked til medarbejderne, der skal forestå fremtagning og transport af genstandene i handlingsplanen.

Denne seddel skal følge med ordren frem til behandlingen, hvor genstandene kontrolleres.

Funktionalitet:

Der er mulighed for udskrivning i prædefineret format hvor følgende skal indgå:

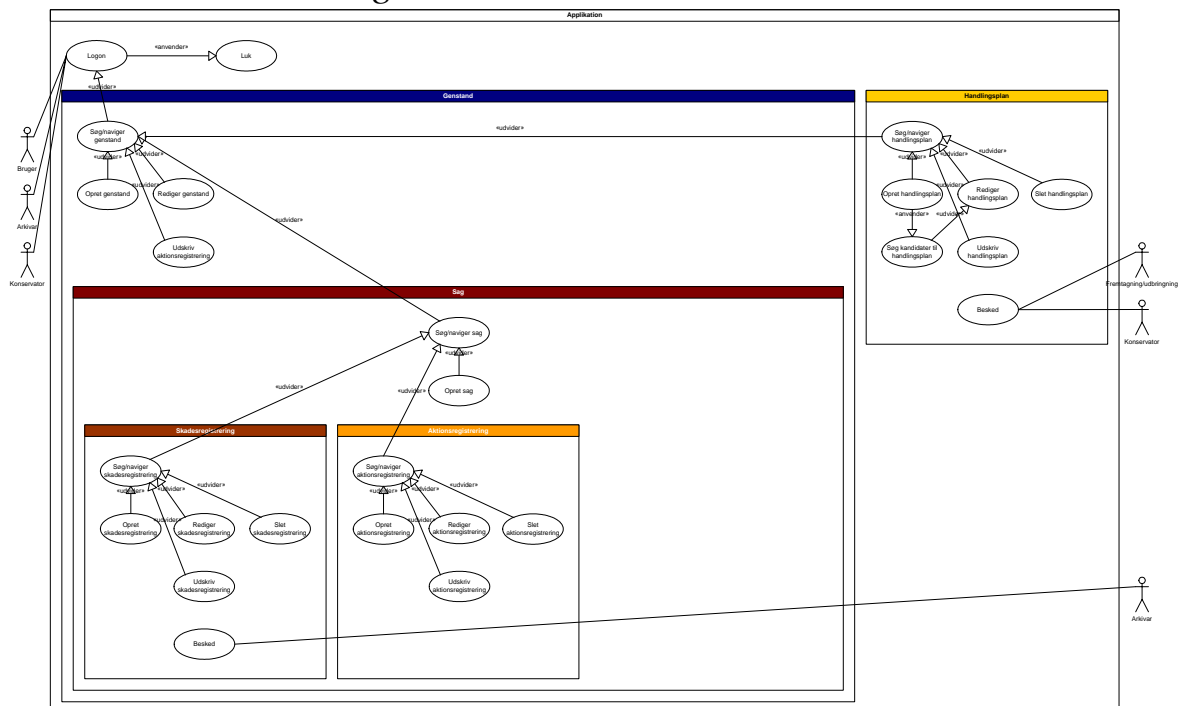
- Medarbejderen der skal forestå fremtagningen er dokumentets første adressat
- Emneoverskrift og timestamp.
- Instruktion i forbindelse med aftalen eller fremtagningen.
- Liste med alle genstande i handlingsplanen og deres placering i arkivet.
- Medarbejderen der skal forestå transporten er dokumentets anden adressat.
- Medarbejderen behandlingen er aftalt med.
- Instruktion i forbindelse med aftalen eller transporten.
- Udskriveren af handlingsplanen.

Fremtagningssedelen udskrives automatisk ved godkendelse af iværksat status, og der skal gives besked til udskriveren via en dialogboks.

Det skal være muligt, efterfølgende at udskrive en ny kopi af fremtagnings-/transportsedelen.

Denne kommunikation kan med fordel foregå via mail i forbindelse med kommunens overgang til Outlook i 2004.

Samlet usecaseoversigt

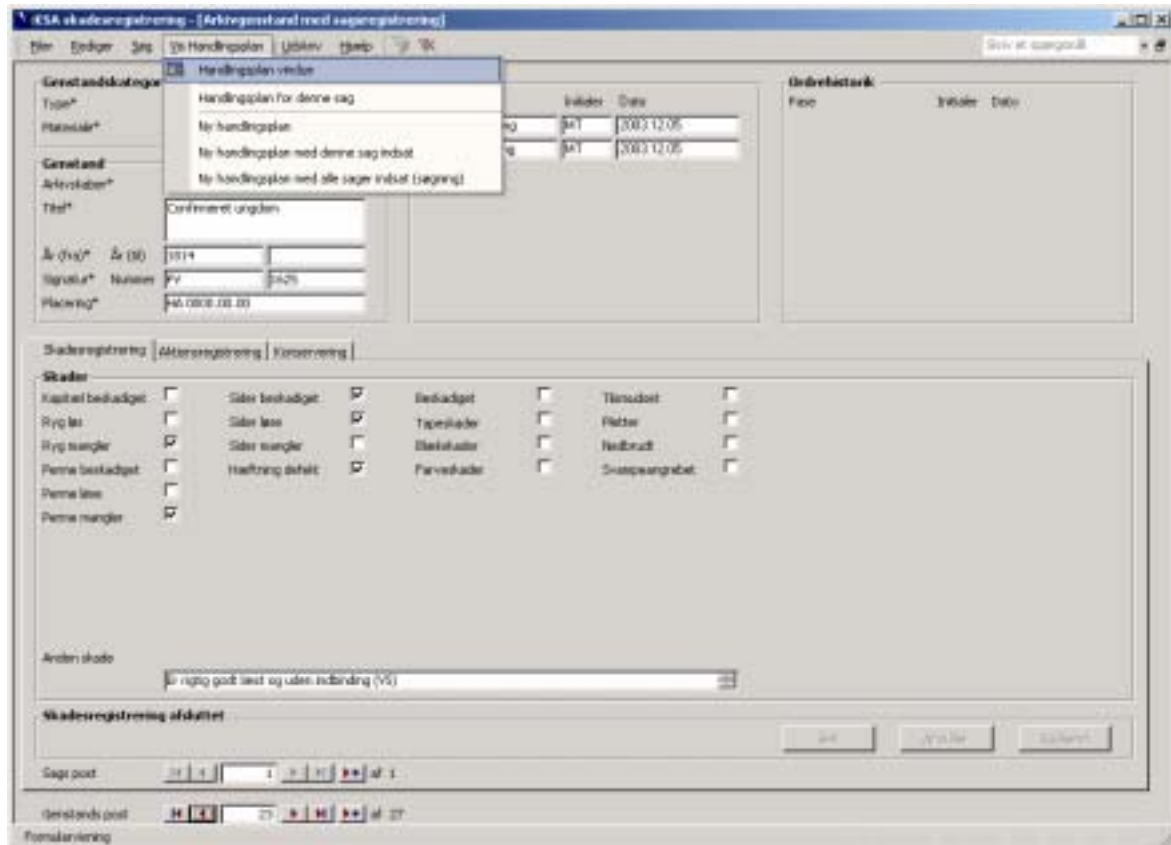


Figur 10-2. Handlingsplanen fungerer som en udvidelse til logon og navigation i genstand

10.1.2 Brugergrænseflade

Med udgangspunkt i problemerne fra den seneste testfase med at overholde størrelsen 1024 x 768 punkter (det var bl.a. glemmt at mange brugere har fast programbjælke nederst på skærmen) og ønsker om at udvide titelfeltet til flere linier blev der foretaget en mindre revision af layoutet. Som det fremgår, fylder fanebladene nu hele bredden og der er gjort plads til vertikal scrollbar. Designmæssigt er den horisontale tredeling ført tydeligere igennem.

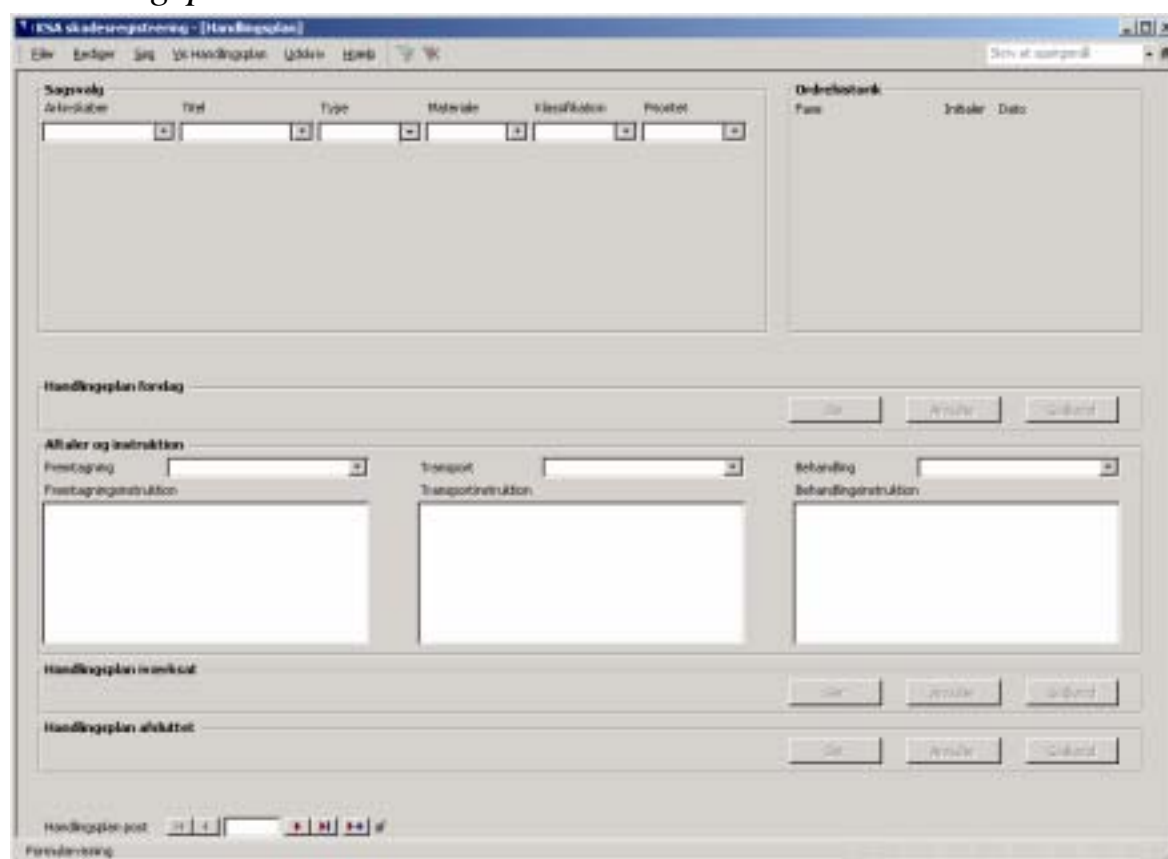
Nyt design af hovedvinduet



Figur 10-3. Genstandsformen er blevet redesignet til at passe bedre til arkivets standartskærme. Der er endvidere indført et menupunkt til navigation og interaktion med handlingsplanvinduet.

I forholdet til denne slice, er menulinien klargjort til samspillet med handlingsplanvinduet og der er tilføjet en ordre historik, der i listeform viser en sags evt. deltagelse i handlingsplaner (status, medarbejder og dato).

Handlingsplan



Figur 10-4. Handlingsplanen har samme tredeling som genstandsvinduet og ensartet funktionalitet. Designelementer har samme placering for at sikre et rolig vinduesskift.

I handlingsplanvinduet sammensættes ordre bestående af udvalgte sager. Denne udvælgelse bør kunne ske gennem flere kanaler:

- I genstandsvinduet fremfindes en eller flere sager gennem søgning og filtrering. Via menuvalg overføres kun den aktive sag eller alle sagerne til en eksisterende eller en ny handlingsplan.
- De genstandssager der på et givent tidspunkt er valgt i genstandsvinduet kan ses ved at aktivere kombinationsboksknapperne i sagsvalg feltet. Man kan forestille sig at de enkelte felters valgbokse sortere på deres eget felt sådan at der af den vej er mulighed for, at undersøge forskellige sammenstillinger og grupperinger.
- Via et søgevindue kan der foretages specialiseret søgning på alle genstands, skadesregistrerings og aktionsregistreringsdata og resultatet vises i listeform. Herfra kan en eller flere udvælgtes og overføres til enten en eksisterende eller en ny handlingsplan.

Iværksættelse af ordre sker efter medarbejdervalg og udfyldning af instruktioner i forbindelse med fremtagning, transport og behandling.

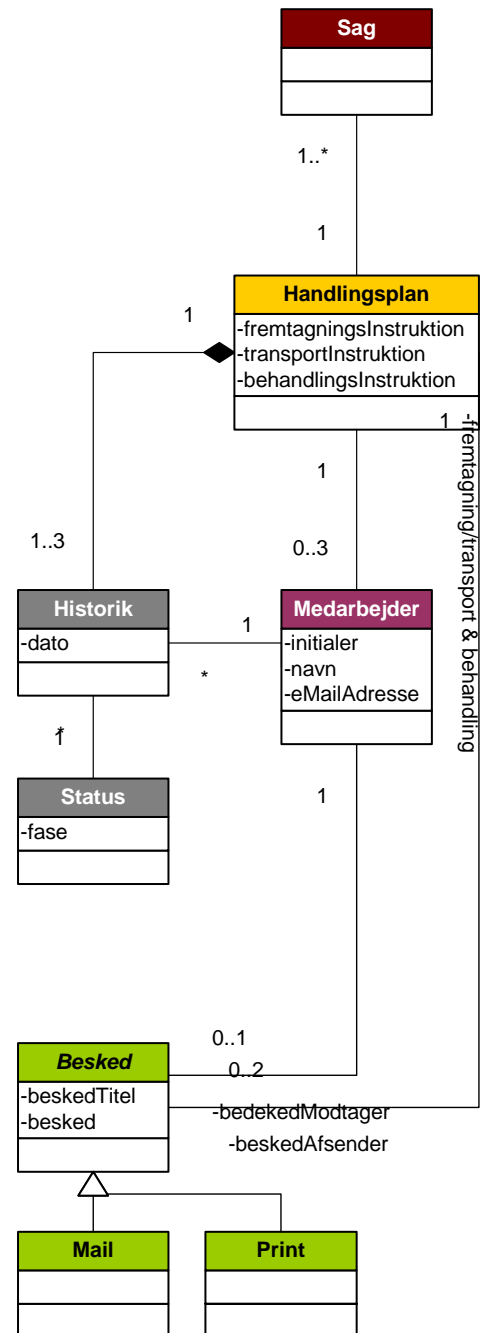
Ved afslutning af alle sager i ordren er der mulighed for at afslutte ordren.

I forbindelse med alle de tre faser er der mulighed for at annullere, slette og godkende.

Ordrehistorikdelen viser undervejs status, initiale og dato for hver af de afsluttede faser.

10.2 Problemområdet

10.2.1 Klassediagram



Figur 10-5. Handlingsplan udgør her en meget beskeden, men vigtig tilføjelse til vores samlede model. Der er her kun medtaget handlingsplanens nærmiljø. For det fulde diagram henvises til systemdokumentationen (se Systemdokumentation Figur3-1).

10.2.2 Klassebeskrivelse

Handlingsplan
-fremtagningsInstruktion
-transportInstruktion
-behandlingsInstruktion

Figur 10-6. Handlingsplan er en ordre der samler flere sager og tilknytter aftaler/instrukser med de medarbejdere der skal forestå fremtagning, transport og behandling. Der registreres status for forløbet således at fremdrift til enhver tid kan måles.

Handlingsplan

Klassebeskrivelse

Modellerer en "ordre" på en eller flere sager/genstande der er udvalgt til behandling. Handlingsplanen skal sikre rationel udvælgelse af ensartede ordrer, og muliggøre kontrol med forløbet. Der er tilkoblet tre medarbejdere der hver især har ansvar for en af de følgende handlinger: Fremtagning, transport og behandling. Handlingsplan har tre status og ændringer for de enkelte kan forekomme så længe en højere status ikke er opnået. Kun de seneste ændringer gemmes.

Krav til attributter og deres multiplicitet

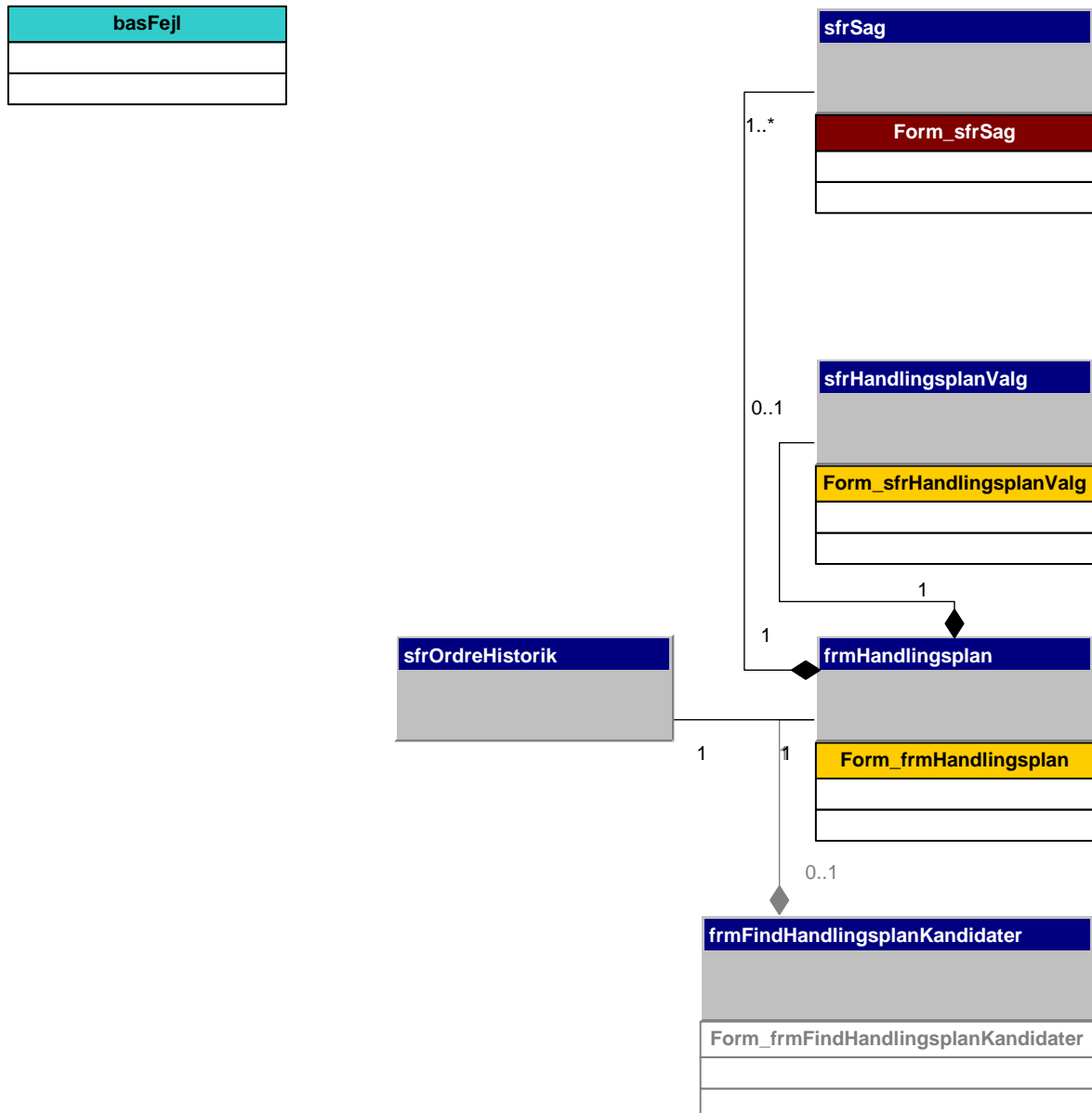
0..3 instruktion, 1..3 Historik (reference), 1..* Sag (reference) og 3 Medarbejder (reference).

11 Objekt Orienteret Design

11.1 Brugergrænsefladekomponenten/funktionskomponenten

Brugergrænseflade- og funktionskomponenterne er ikke detaljeret planlagt, men i oversigtsform indsat i de relevante sammenhænge.

11.1.1 Struktur



Figur 11-1. Brugergrænseflade-/funktionskomponenten udbygges gradvist og der er medtaget en planlagt, men ikke konstrueret søgningsformular til handlingsplan. For det fulde diagram henvises til systemdokumentationen (se Systemdokumentation Figur 5-1).

11.1.2 Standardmoduler

MS Access tilbyder flere forskellige steder at placere sin egenudviklet funktionalitet:

- Som løse rutiner i et standardmodul. Dette er den traditionelle løsning.
- Som egne klasser i klassemoduler.
- I klassemodulet til formularen, rapporten eller lignende. Dette er oftest anvendt

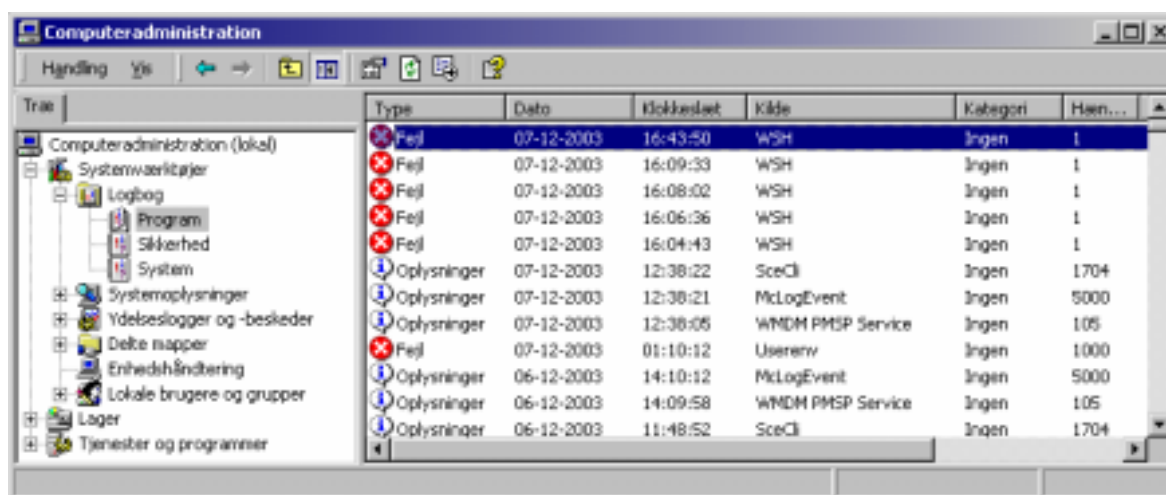
Vores system benytter mest muligheden for at lægge funktionaliteten direkte på formularen. Dog har vi lavet enkelte simple standardmoduler, som er nærmere beskrevet i systemdokumentationen (se Systemdokumentation Standard moduler s. 63).

Fejl

basFejl

Figur 11-2. Der er til almindelig fejlrapportering lavet et standardmodul, som benyttes i hver rutines fejlhåndtering

Modulet sørger for at logge fejl i en central logfil, samt give brugeren en sigende meddelelse.

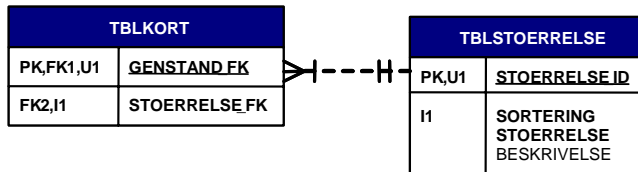


Figur 11-3. En fejl i systemet optræder også i arbejdsstationens programlog. Desværre har brugerne ikke selv mulighed for at se disse, da den implementerede politik i Københavns Kommunes IT-miljø hindrer dette.

11.2 Model-/databasekomponenten

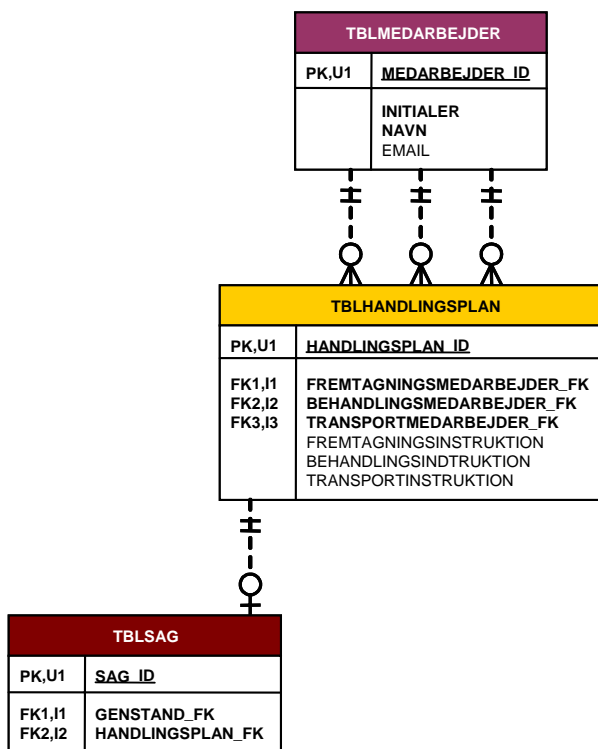
11.2.1 Ændringer i fysisk datamodel

Der er i denne slice fem primære ændringer:



Figur 11-4. Kolonnen STOERRELSE i tabellen TBLKORT er normaliseret ud i den nyoprettede tabel TBLSTOERRELSE.

Da det viste sig at der er ganske få og standardiserede størrelser af kort er det naturligt at vi har normaliseret størrelsen ud i sin egen tabel med nøgler og relationer svarende til de tidligere tabeller. Tabellen TBLKORT har uændret relation til tabellen TBLGENSTAND.

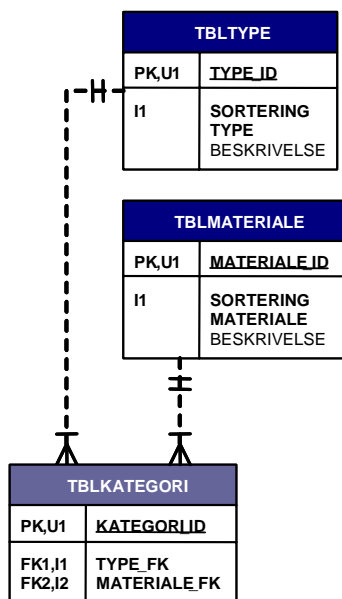


Figur 11-5. Tabellen TBLHANDLINGSPLAN er tilføjet.

Ifølge planerne er der tilføjet en tabel med relationer til at indeholde systemets handlingsplaner.

Der er sat kolonnen BESKRIVELSE på til beskrivelse i tabellerne TBLTYPE, TBLMATERIALE, TBLPRIORITET og TBLKLASSIFIKATION.

Der er sat kolonnen SORTERING til styring af visningsrækkefølge i tabellerne TBLTYPE, TBLMATERIALE, TBLSTOERRELSE, TBLPRIORITET og TBLKLASSIFIKATION



Figur 11-6. Tabellen TBLKATEGORI er tilføjet.

Indførelsen af kategori som mellemtabel for type og materiale skyldes ønsket om at materialevalget kan tilpasse specielt efter typevalget der foretages først (se Formularer s. 98). Dette har der oprindeligt været taget højde for i modellen (se Figur 10-5), men kategoriklassen der ikke havde egne attributter blev ”denormaliseret” ved transformeringen til tabelstrukturen allerede i slice 1. Dette bunder i at det ikke var klart for værkstedet at sådanne muligheder forelå og derfor naturligtvis heller ikke var fremsatte som et krav eller ønske i den indledende fase.

Denne og mange andre småfejl og -forbedringer der er opdaget undervejs i faserne og ved formel og uformel brugerafprøvning er forholdsvis enkle at tilrette under prototypearbejdet.

Dette er netop en af metodens store styrker.

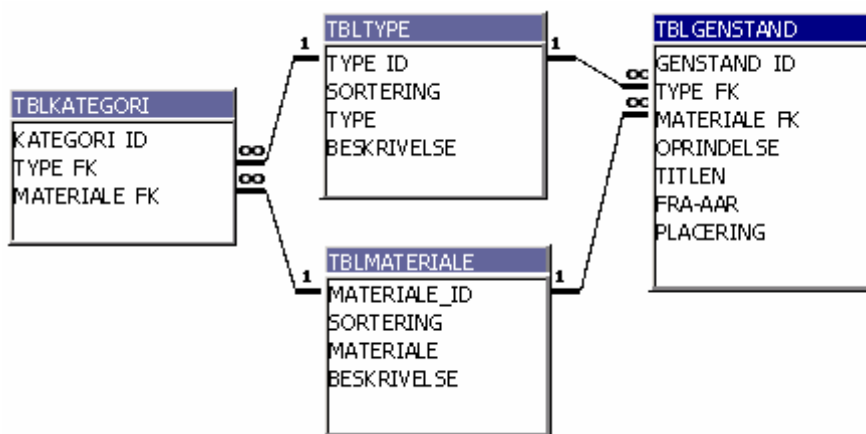
12 Konstruktion

Konstruktionsfasen er som tidligere nævnt blevet indskrænket til ønskede småtilretninger fra slice 1, jævnfør dennes testfase (se Testresultater s. 83). Oplægget fra OOA fasen til designændringer i brugerintergrænsefladen er konstrueret, og vi vil kun kort beskrive et par andre synlige ændringer.

12.1 MS Access applikation

12.1.1 Tabelstruktur tilføjelser

Konstruktionen af tabeltilføjelse har fulgt designet, bortset fra tabellen: TBLKATEGORI der på grund af vanskelighederne med at konstruere den optimale løsning (se Figur 5-11) er bundet anderledes end ønsket. Designet er efterfølgende tilrettet.



Figur 12-1. Tabelstrukturændringerne viser at TBLKATEGORI ikke som planlagt er knyttet til TBLGENSTAND, men fungerer som mellemtabel der blot leverer data til materialekombinationsboksen.

Som en lille særhed for MS Access, vises flere en-til-mange relationer mellem to tabeller ved hjælp af skyggetabeller (se Systemdokumentation Bilag B s. 89).

12.1.2 Forespørgsler

For at muliggøre en koordineringen af type og materiale valgboksene, er der gjort brug af MS Access mulighed for at konstruerer parameteriserede forespørgsler i QBE editoren.

```

SELECT
  TBLMATERIALE.MATERIALE_ID, TBLMATERIALE.MATERIALE, TBLMATERIALE.BESKRIVELSE
FROM
  TBLMATERIALE INNER JOIN TBLKATEGORI ON TBLMATERIALE.MATERIALE_ID = TBLKATEGORI.MATERIALE_FK
WHERE
  TBLKATEGORI.TYPE_FK=[cboGenstandstype];
  
```

Figur 12-2. Kriteriet i SQL forespørgslen er sat til [cboGenstandstype] der ikke er kendt i tabelstrukturen hvorfor den søges i objektstrukturen og efterfølgende aflæses og anvendes. Findes kriteriet ikke fremkommer i stedet en indtastningsboks til brugerinteraktion, hvilket kan anvendes i mange andre forespørgselssammenhænge.

12.1.3 Formularer

Koordinerede valgbokse

Med konstruktionen af mellemtabellen TBLKATEGORI og bindingen af materiale valgboksen gennem den parameteriserede forespørgsel QRYMATERIALE_KATEGORI, er koordination mellem de to valgbokse for type og materiale blevet mulig (et par hændelser, kode og andre objektindstillinger er ikke medtaget).

Genstandskategori		Sagshistorik	
Type*	Dokument	Fase	Initialer Dato
Materiale*	Papir		
Genstand	Papir	evt. papir beskrivelse	
	Pergament	evt. pergament beskrivelse	

Genstandskategori		Sagshistorik	
Type*	Protokol	Fase	Initialer Dato
Materiale*	Skind ryg		
Genstand	Skind ryg	evt. skind ryg eskrivelse	
	Skind helbind	evt. skind helbind beskrivelse	
Arkivskaber*	Pergament ryg	evt. pergament ryg beskrivelse	
	Pergament helbind	evt. pergament helbind beskrivelse	
Titel*	Lærred ryg	evt. lærred ryg beskrivelse	
	Lærred helbind	evt. lærred helbind beskrivelse	
År (fra)*	Papir helbind	evt. papir papir helbind beskrivelse	
År (til)	Bindet mangler helt	Umuligt at kategorisere præcist	

Figur 12-3. Resultatet af forskellige type/materiale valg kan nu mere præcist tilpasses de reelle ønsker, og det er overladt til værkstedet at skrive meningsfulde beskrivelser.

Beskrivelser, sortering og markering af obligatoriske felter

For at give brugeren mulighed for at se og gøre brug af de overordnede tabelrettelser er valgboksene overalt i applikationen blevet opdateret. Beskrivelsesfelterne vises og den ønskede rækkefølge kan fremover bestemmes og justeres helt frit på tabelniveau ved hjælp af sorteringskolonnen.

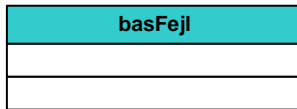
År (fra)*	1761
Størrelse*	Lille
Placering*	Mindre end A2 (29,7 x 42,0 cm)
	Større end A2, men mindre end A1 (59,4 x 42,0)
	Større end A1, men mindre end A0 (59,4 x 84)
	Større end A0
Skadesregistrering	
Skader	
Kapitæl beskadiget	<input type="checkbox"/>
Sider beskadiget	<input type="checkbox"/>
Beskadiget	<input type="checkbox"/>
Tilsmudset	<input type="checkbox"/>

Figur 12-4. Valgboksen for størrelse ses her aktiveret og det er nu, ved hjælp af beskrivelsesfelterne, lettere at tage stilling til den relative størrelsesangivelse der er ønsket for kort og tegninger.

For at skabe mere klarhed over hvilke felter der er obligatoriske, er der tilføjet en diskret stjerne markering.

12.1.4 Standardmoduler

basFejl



Figur 12-5. Der er til almindelig fejlrapportering lavet et standardmodul (basFejl), som lægges i hver rutines fejlhåndtering.

```
Private Sub Form_Current()
On Error GoTo Fejl_Form_Current
...
Fejl_Form_Current:
    Fejl (Me.Name & ".Form_Current")
    Resume Next
```

Desuden er det i hver rutine muligt at sætte og læse systemets nuværende fejlkode

```
Fejlkode = IDVisible(Not Form_frmGenstand.GENSTAND_ID.Visible)
IDVisible = Fejlkode
```

Fejlkode er implementeret som en property i basFejl:

```
Private intFejlkode As Integer

Public Property Let Fejlkode(ByVal intFejlnummer As Integer)
    intFejlkode = intFejlnummer
End Property

Public Property Get Fejlkode() As Integer
    Fejlkode = intFejlkode
End Property
```

Den generelle fejlrapportering skriver for hver fejl en meddelelse i en central logfil

```
strFilnavn =
"J:\Arkiv\Fælles\Skadesregistreringsdatabase\System\Prod\Fejl.log"
Set objFile = objFileSystem.OpenTextFile(strFilnavn, _
    FORAPPENDING, True)
objFile.WriteLine Now & vbTab & strHaendelse
En meddelelse i logfilen kan for eksempel være
07-12-2003 16:43:50    Fejl (0) i frmFejl - TEST.cmdFejl_Click:
```

Der skrives også en meddelelse i den lokale programlog

```
Set objShell = CreateObject("WScript.Shell")
objShell.LogEvent 1, strHaendelse
```


System - slice 3

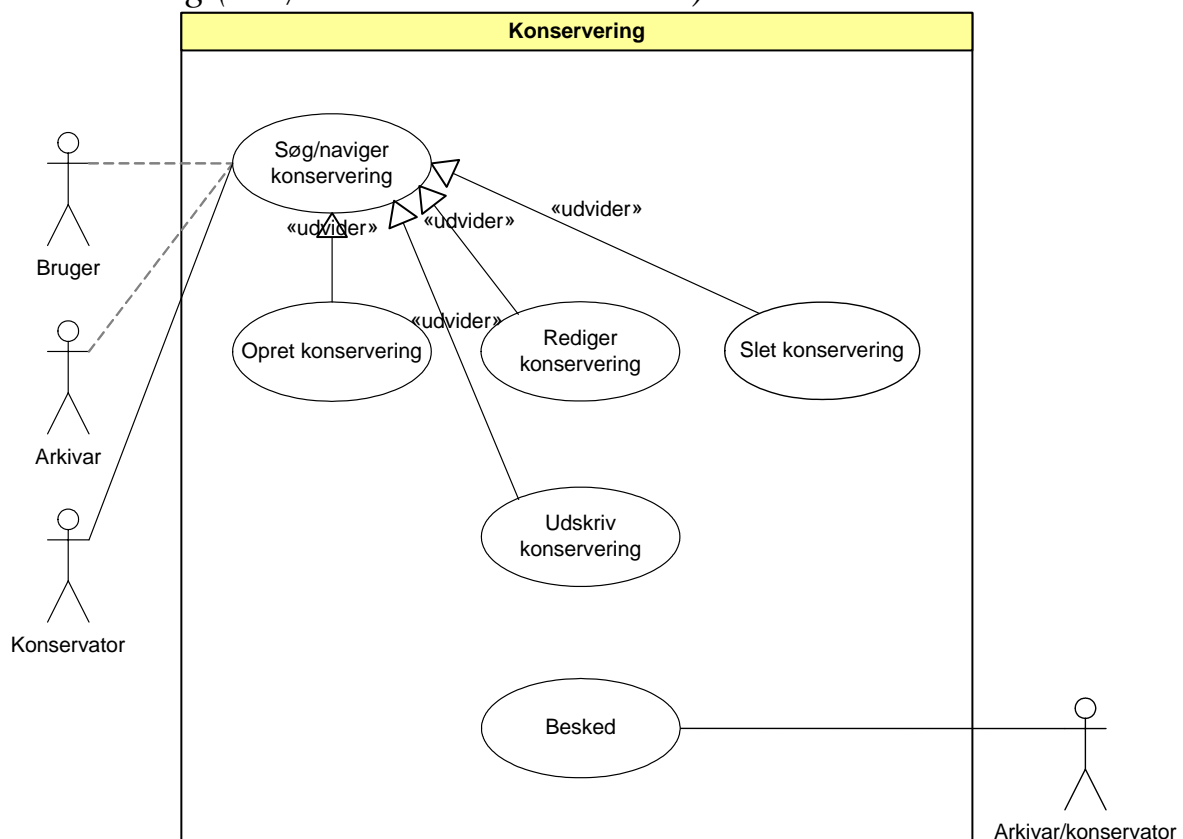
Behandlingsregistrering

Slice 3 indeholder hovedsagligt udviklingen af behandlingsdelens konkrete aktivitet ”konservering”. Som ved foregående slice fokuseres der primært på OOA, OOD og tabelkonstruktion således at det er op til KSA at konstruere den foreslåede funktionalitet. Slicen indeholder endvidere enkelte rettelser og forbedringer til tidligere kode. Det har været planlagt at aktivere sikkerhed på brugerniveau (se bilag 19.5 s. 147), men set i forhold til færdiggørelsesgraden vil det være til mere besvær end gavn under det resterende udviklingsarbejde og er derfor udeladt. Databasen opdeles i frontend/backend for forbedret fleksibilitet i forbindelse med det fortsatte udviklingsarbejde. Test foretages i overensstemmelse med rettelserne og forsøges afrundet. Implementeringsfasen gennemgås omhyggeligt ligesom der foreslås en hensigtsmæssig fremtidig struktur for produktions og testmiljø.

13 Objekt Orienteret Analyse

13.1 Anvendelsesområdet

13.1.1 Brug (aktører/usecases/scenarier)



Figur 13-1. Usecasen konservering er den p.t. eneste realiserede form for behandling. Flere muligheder kan tænkes i fremtiden f.eks. sikkerhedsfilmning, skanning, ompakning, kassesion. Bemærk at det kun er konservator der har adgang til at oprette, redigere og nedlægge registreringer

System – slice 3

OBS. Nedenstående gengiver kun et eksempel på en usecase. Se alle beskrivelserne i systemdokumentationen (se Systemdokumentation Konservering s.30)

Opret konservering

Brugsmønster:

Først aktiveres indtastningsdelen for behandling, og via et valg foretages en søgning der viser alle skadesregistreringer i den handlingsplan der ønskes påbegyndt.

Kontroller de udbragte genstande og konferer med den medfølgende ”Besked til behandling”.

For hver sag der er til stede i handlingsplanen og påbegyndes, godkendes behandlingsregistreringen inden programmet lukkes eller der foretages et andet valg.

Statusgodkendelsen ”Konservering påbegyndt” vises i sagshistorikken sammen med navn og timestamp.

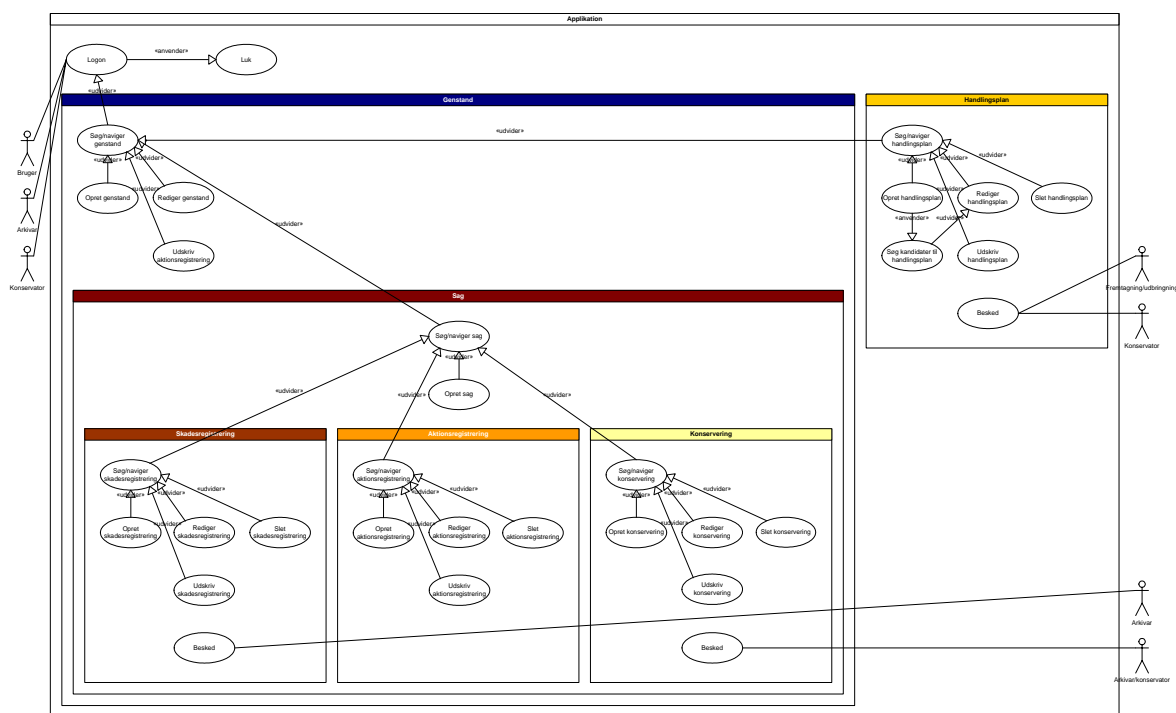
Funktionalitet:

Annullering og godkendelse, kan foretages under hele forløbet, ved hjælp af et sæt knapper der aktiveres og deaktiveres afhængigt af rettigheder og forløb.

Ved godkendelse og validering vises statusgodkendelsen ”Konservering påbegyndt” i sagshistorikken sammen med navn og timestamp.

Denne status låser for redigering og sletning af handlingsplanfasen iværksat for alle de berørte sager og åbner for næste aktivitet for den enkelte sag der er faset frem til behandling anses for afsluttet.

Samlet oversigt



Figur 13-2. Konservering afslutter diagrammet der nu viser det samlede usecasesenarie.

13.1.2 Brugergrænseflade

Opgaven for brugergrænsefladefasen er planlægning af en behandlingsregistrering for konservering, der ligesom skades- og aktionsregistreringen anbringes på et sagsfaneblad.

Konserveringsregistrering på sagsfaneblad

Figur 13-3. Oplæget til konserveringsregistreringen følger usecasen (se Figur 13-1) for konservering, og indledes med knapper til sletning, annullering og godkendelse af påbegyndelse af behandlingen. Det er derefter muligt at indtast førbeskrivelse, markere almindeligt forekommende behandlinger eller beskrive andre processer. Efterfølgende kan bemærkninger om proces eller resultat indføres som en efterbeskrivelse. Knapper til sletning, annullering og godkendelse afslutter funktionaliteten. Navigationsknapperne i bunden tilhører sammen med fanebladene sagen.

Informationsformular

Figur 13-4. Indledningsinformation der dukker ved programstart, og kan afstilles af brugeren.

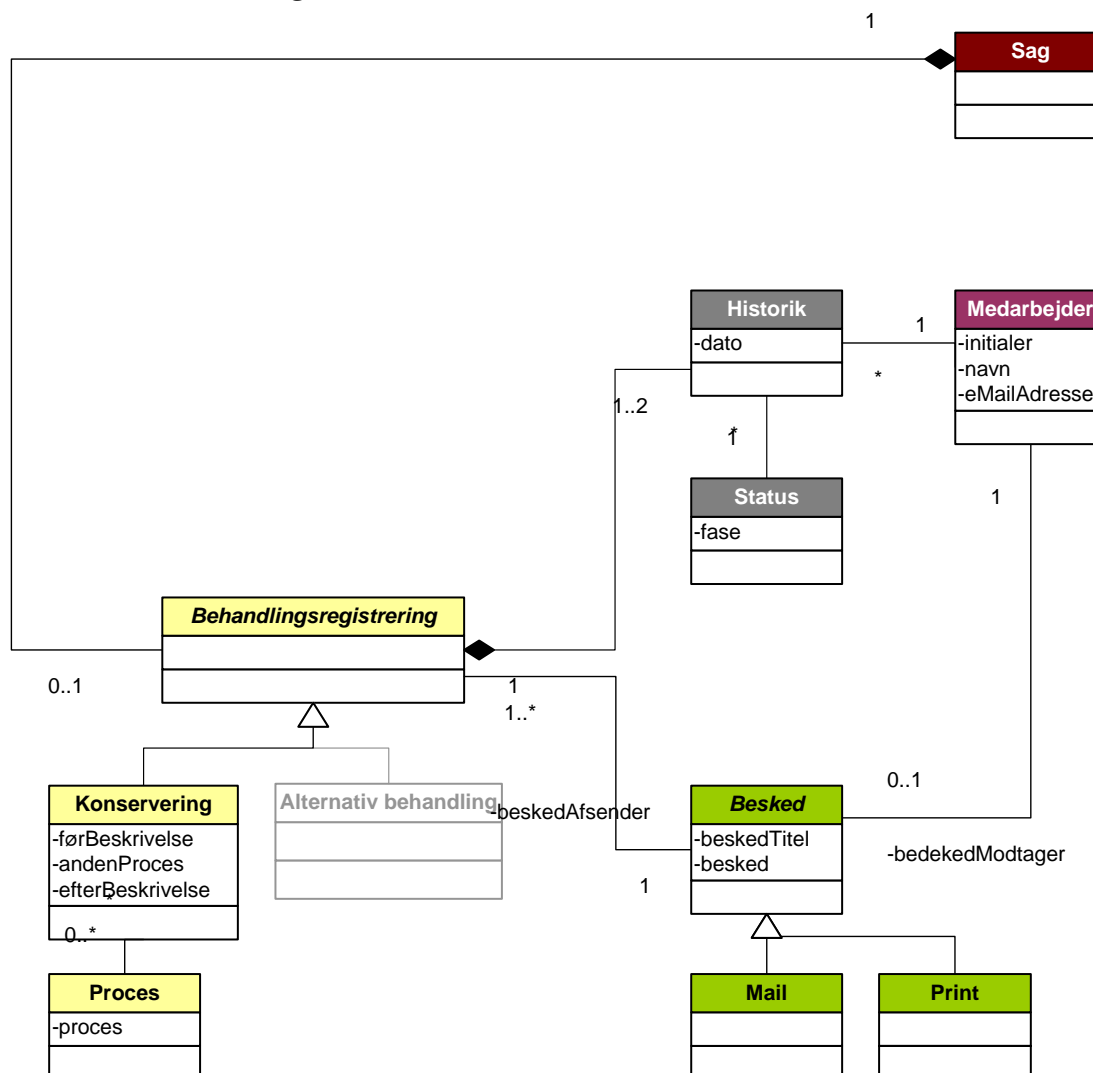
For indledningsvis at informere brugeren om hvilke tanker der ligger til grund for applikationen og dens tilblivelse, samt give udviklerne mulighed for at indskrive deres navne med flammeskrift i brugernes bevidsthed, blev der lidt spontant tegnet en "splashscreen".

Ved nærmere overvejelse viser det sig imidlertid, at det er en fortrinlig mulighed, for at kommunikere ændringer og opdateringer til de reelle brugere - den bør videreudvikles.

System – slice 3

13.2 Problemområdet

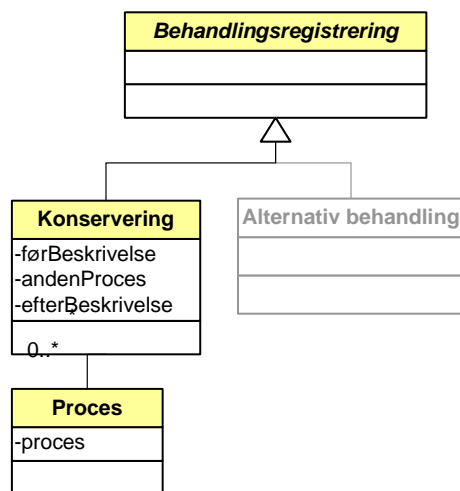
13.2.1 Klassediagram



Figur 13-5. Der er her kun medtaget behandlingsregistreringens nærmiljø. For det fulde diagram henvises til systemdokumentationen (se Systemdokumentation Figur 3-1).

Behandlingsstrukturen afslutter modellen for det planlagte projekt. Som det var tilfældet for aktivitetsregistrering er der også her taget behørigt højde for fleksibilitet gennem mulighed for fremtidige konkrete tilføjelser af f.eks. skanning, ompakning eller lignende.

13.2.2 Klassebeskrivelse



Figur 13-6. Jævnfør aktivitetsregistrering er behandlingsregistrering abstrakt superklasse for alle former for behandlinger der kan tænkes at blive iværksat. De eneste fællestræk for disse behandlinger er p.t. status for påbegyndelse og afslutning af behandlingen. Den abstrakte behandlingsregistrering sikrer fleksibilitet og udvidelsesmuligheder, da der her let kan tilføjes nye behandlinger på et senere tidspunkt. Konservering, kender via valg brugeren gør sig forskellige generelle processer der anvendes.

Behandlingsregistrering

Klassebeskrivelse

Behandlingsregistrering er en superklasse for alle former for behandlinger der kan tænkes for arkivets genstande. Klassen sikrer systemfleksibilitet ved at være indsætningspunkt for fremtidige specialiseringer. Eneste fællesnævner for de p.t. kendte aktiviteter er status for påbegyndelse og afslutning af registreringen.

Krav til attributter og deres multiplicitet

1..2 Historik (reference).

Konservering

Klassebeskrivelse

Modellerer en genstands behandling (her på konserveringsværkstedet) med før-beskrivelse, processer og efter-beskrivelse. Ændringer kan forekomme så længe det ikke har status "Færdigbehandlet" og kun de seneste ændringer gemmes.

Krav til attributter og deres multiplicitet

0..* Proces (reference), 0..1 andenProces [Mindst en af de to første attributter], 0..1 førBeskrivelse og 0..1 efterBeskrivelse.

Proces

Klassebeskrivelse

Proces definerer en specifik behandling en genstand modtager på værkstedet.

Krav til attributter og deres multiplicitet

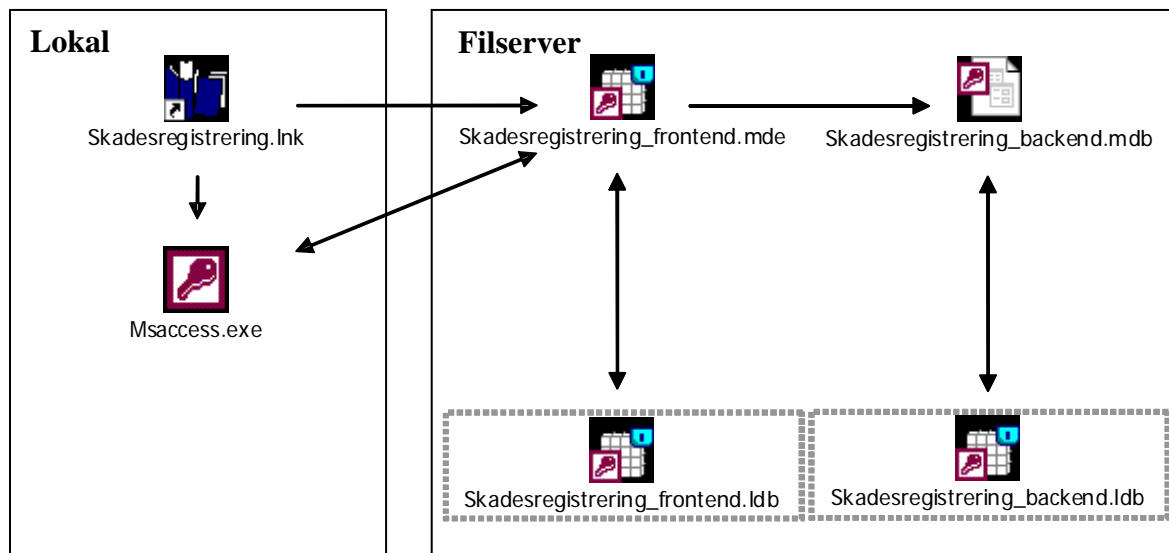
1 proces.

14 Objekt Orienteret Design

14.1 Arkitektur

14.1.1 Deling i en front- og backend

I forbindelse med projektafslutningen er det besluttet at dele databasen i en front- og en backend der begge er placeret på filserveren.



Figur 14-1. Databaseløsningen opsplittet i en front- og en backend på filserveren. Frontenden rummer bl.a. applikationens statiske objekter, formularer, (makroer) og prækompilerede forespørgsler. Backendens derimod indeholder tabeller og relationer. Genvejen starter som før den lokale version af MS Access eller subsidiært en MS Access runtimeversion.

Frontenden er efter forsøg med afviklingshastighed blevet foreslået konverteret til mde fil. Selve tabeldelen anses for værende færdigudviklet, og den foreslåede deling have følgende fordele for vores projekt, frem for den tidligere alt-i-en-version:

- mde-filen sikrer aflukning fra ændringer i frontenden, koden kan ikke tilgås, mindre størrelse og hurtigere åbning over nettet.
- Mere fleksibilitet i forhold til udviklingsarbejde på frontenden, der ved færdiggørelse blot substitueres med den eksisterende frontend i produktion.
- Databasen lades uberørt af ændringer i frontenden.

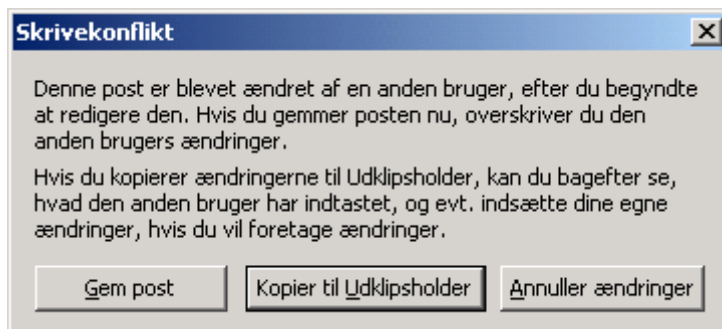
14.1.2 Håndtering af samtidighed

Samtidighed opstår i databaser når to eller flere brugere forsøger at redigere den samme post samtidig. MS Access opererer for .mdb filer med tre hovedtyper af samtidighedshåndtering:

- **Ingen låse ("optimistisk låsning")**
(Standard) I formularer kan to eller flere brugere redigere den samme post samtidig. Dette kaldes også begrænset låsning. Hvis to brugere forsøger at gemme ændringer af den samme post, får den bruger, der forsøger at gemme posten sidst, en meddelelse (se Figur 14-2). Denne bruger kan derefter slette posten, kopiere posten til Udklipsholder eller erstatte de ændringer, den anden bruger har foretaget. Denne indstilling bruges typisk i skrivebeskyttede formularer eller i enkeltbrugerdatabaser. Den bruges også i flerbrugerdatabaser, hvis mere end én bruger skal kunne foretage ændringer i den samme post samtidig.
- **Alle poster (tabel)**
Alle poster i den underliggende tabel eller forespørgsel låses, mens formularen er åben i formularvisning eller dataarkvisning, mens rapporten vises eller udskrives, eller mens forespørgslen køres. Brugere kan læse posterne, men ikke redigere, tilføje eller slette poster, før formularen lukkes, rapporten er udskrevet, eller forespørgslen er kørt.
- **Redigeret post (side)**
(Kun formularer og forespørgsler) En side med poster låses, når en bruger begynder at redigere et felt i posten, og forbliver låst, indtil brugeren flytter til en anden post. En post kan således kun redigeres af én bruger ad gangen. Dette kaldes også fuldstændig låsning.

Låsning uanset niveau er resursekrævende da oprettelse, redigering og sletning for den enkelte bruger skal registreres i låsefilen. Denne ligger på et netværksdrev og der bruges unødigt båndbredde og serverressurser, om end i begrænset mængde. Det er vores skøn at der i daglig drift er meget få situationer hvor der vil forekomme samtidig anvendelse af databasen, og meget sjældent i de samme poster. Et optimistisk "workloadgæt" er 500 sager om året, men allerede ved 200 er systemet en succes. Den almindelige anvendelse vil være indtastning af skadesregistreringer, hvilket foregår i forbindelse med oprettelse af genstandsposter og der vil derfor ikke forekomme samtidighed i denne del af anvendelsessituationerne.

Indledningsvist er det derfor besluttet, at der ikke bliver foretaget låsning, men at det er op til brugerne i de meget sjældne tilfælde at håndtere samtidighedsproblemerne jævnt før det ovenstående afsnit (se Ingen Låse ovenfor). Den overordnede låsningsstrategi kan altid senere ændres til en mere restriktiv form, hvis det viser sig nødvendigt.



Figur 14-2. Dialogboksen der viser at et samtidighedsproblem er opstået og anviser tre mulige løsningsmodeller.

System – slice 3

14.2 Brugergrænsefladekomponenten

Som tidligere nævnt er brugerfladekomponenten en realisering af analysens oplæg til brugergrænsefladen, der er fremstillet direkte i MS Access, i stedet for først at blive tegnet. Den er derfor allerede konstrueret, og der er her kun tilbage at kontrollere om der er i overensstemmelse med funktionskomponenten og sikre at funktionaliteten kan placeres hensigtsmæssigt.

14.3 Funktionskomponenten

Funktionskomponenten indeholder for denne slice kun begrænset selvstændig nykonstrueret funktionalitet. For frmInfo vil der imidlertid indgå hændelser der sikre visning og eventuelt afstilning.

14.3.1 Struktur



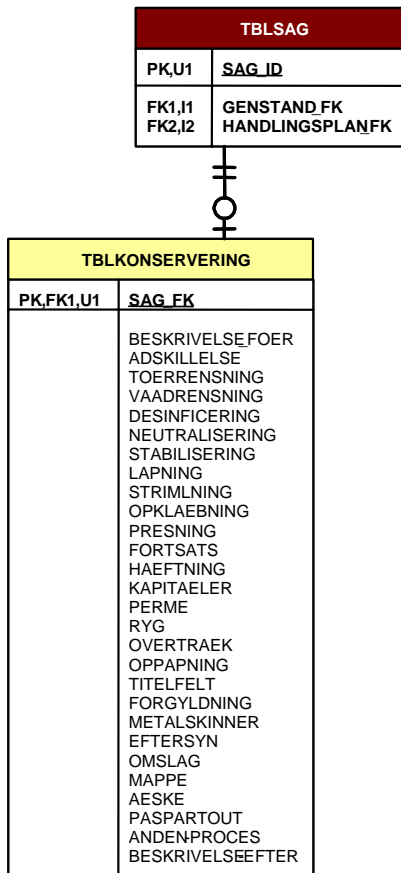
Figur 14-3. Tilføjelserne er sfrKonsivering, og frmInfo der kan udvikles til at vise brugerinformation

For det fulde diagram henvises til systemdokumentationen (se Systemdokumentation Figur 5-1).

14.4 Model-/databasekomponenten

14.4.1 Transformering til fysisk model

I denne slice er der en enkelt tilføjelse til den fysiske model:



Figur 14-4. Tabellen TBLKONSERVERING tilføjes modellen.

Konsivering er et eksempel på en behandlingsregistrering, og entiteten er denormaliseret på samme vis som skadesregistrering i slice 1, det vil sige at der ikke er en generel tabel for behandlingsregistrering og at alle processerne er lagt ind i selve konserveringstabellen.

15 Konstruktion

15.1 MS Access applikation

15.1.1 Tabelstruktur ændringer

Tabellen TBLKONSERVERING er indsat som beskrevet ved modellen.

15.1.2 Formularer

Informationsformularen (se Figur 13-4) har som et par andre tiltag deres rod i konstruktionsfasen hvor der jo arbejdes på det laveste niveau med programmet og ikke er langt mellem ide og realisation.

Det er her som et forsøg konstrueret i sin simpleste form med en billedbaggrund komprimeret i JPG (8k) og tekstfelt til lidt programindledning. Formularen er konstrueret som modal dialog og er koblet direkte til startupsekvensen. OK knappen kalder hovedformularen og fra en afstillingscheckboks bringes koden til at sætte en ”startup property” til fremover at pege på hovedvinduet. For at forskellige vinduer ikke skal have forskellige proportioner og hoppe ved vinduesskift er der tilkoblet en tom menulinie der kun har visuel effekt.

Der har således ikke været behov for ændringer i tabelstrukturen, men til gengæld virker info dialogen kun til den afstilles første gang.

Efterfølgende er konceptet taget op i en iteration af OOA fasen og det vil være nærliggende at videreudvikle informationsvinduet således at det virker individuelt for hver bruger, ligesom frastillingen bør kunne annulleres for alle brugere når ny information om applikationen skal viderebringes.

15.1.3 Programfærdiggørelse

Der er oprettet scripts og genveje svarende til Startscript spiken (se bilag 19.11 s. 173). Den specifikke løsning kan ses i systemdokumentationen.

16 Test

16.1.1 Test procedure

Annie Bentsen (AB) som sids vores testperson og testen blev afviklet som i slice 1 (se Test Procedure s. 83) Målet var at følge op på tilføjelser i skades- og aktionsdelen.

16.1.2 Test resultater

Konstruerede ændringer der var ønsket i slice 1 (se Test resultater s. 84) blev gennemgået og godkendt.

Skadesregistrering

Registreringen virker nu bedre tilpasset.

Ønsker til rettelse:

- Genstande hvor materialet fokuserer på indbindingen, tilføjes et valg for: Indbinding ukendt eller blot Ukendt.

Aktionsregistrering

Ingen problemer.

Observationer

Det blev i forbindelse med brugertesten klart at der er funktionalitet der skal tydeliggøres eller tilføjes:

- Indbygget hjælp er nødvendig inden applikationen implementeres endeligt.
- Oprettning af både ny genstand og ny sag fremtræder ikke tydeligt nok og må også kunne ske fra menulininen.
- Der skal være en standardmåde at komme videre på, selv om enkelte data ikke findes.
- Det er tydeligt at der ikke kan testes eller ændres i historikvinduerne.

16.2 Accepttest

16.2.1 Test procedure

Test skemaet for de tests der er relevante for denne fase blev gennemløbet og de enkelte punkter blev taget op endnu en gang - et for et.

16.2.2 Test resultater

For de dele der er indeholdt i slice 1 var der kun ganske få mangler eller fejl.

Test af skadesregistrering (faneblad)

- 3.9 Grundlæggende søgning er ok, men avanceret søgning mangler.

Test af aktionsregistrering (faneblad)

- 4.8 Grundlæggende søgning er ok, men avanceret søgning mangler.

Vinduerne (formularerne) til specielle eller avancerede søgninger må vente til en senere version.

17 Implementering

17.1 Installation og afprøvning

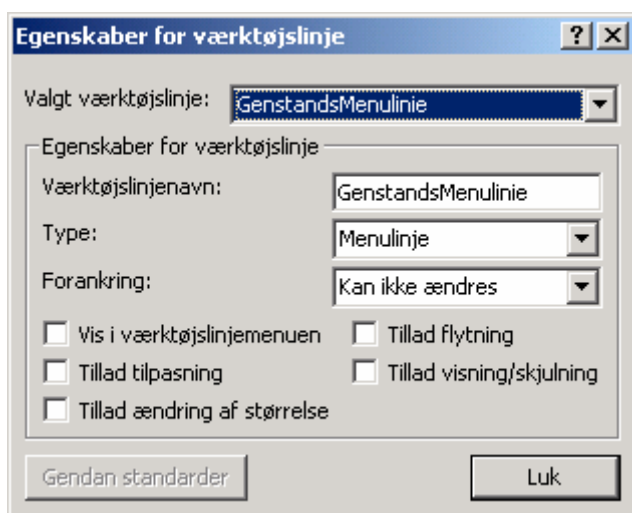
Systemet blev installeret som angivet i Start Script spike (se bilag 19.11 s.173).

Det viste sig ved en hurtig test på en almindelig arbejdsstation at systemet starter hurtigst når det kører i en mde-fil.

Dette blev implementeret og systemdokumentationen opdateret.

En kort diskussion om change management som serverinstallationen i

Systemdokumentationen lægger op til (se Installation på server s. 82) bekræftede os i at det var en rigtig struktur, men at der er for mange manuelle operationer.



Figur 17-1. Som det sidste i forbindelse med brugergrænsefladen er menulinierne fastlåst på deres position øverst i vinduet og al overflødig funktionalitet spærret

Afrunding

18 Projekt konklusion og perspektivering

Det er formålet med konklusion og perspektivering, at opsummere i hvilken grad målene for projektet er opfyldt og hvilke erfaringer, der er gjort undervejs samt, at se resultatet i en større fremtidig sammenhæng.

18.1 Diskussion og konklusion

De to for skellige hovedmålgrupper må skelnes og evalueres hver for sig:

- Hovedopgaven for Datamatikerstudiet (projektrapporten)
- Projektopgaven for Københavns Stadsarkiv (produkt og systemdokumentation)

18.1.1 Datamatiker hovedopgaven

Diskussion – Produkt (projektrapport)

Målet med hovedopgaven jævnfør oplæg og målsætningen var at vise, at vi i forhold til det meget generelle datamatikerstudie, kan anvende vores viden til systematisk og analytisk at gennemføre en konkret problemløsning (inden for et brugervalgt emne), samt at kunne forbinde teori og praksis i et bredt sammenhængende fagligt perspektiv.

- Der er her tale om et MS Access databaseprojekt hvor vi i opgaveløsningen spænder over alle pensumgrupper fra datamatikerstudiet, med hovedvægt på: Systemudvikling, Projektledelse, Organisation og Logistik, Objekt Orienteret Analyse & Design, Distribuerede systemer, Relationsdatabaser og Programmering.
- Der er således arbejdet meget bredt, og vi har været glade for at kunne følge projektet hele vejen fra brugerønsker til færdigt produkt. Processen er forløbet meget velstruktureret og systematisk gennem hele projektet uden at det har fastlåst forløbet. De indlagte prototypefaser er i den forbindelse en god metode, der opfordrer til at indarbejde nye ideer og sent tilkomne brugerønsker.
- Som det fremgår, har vi anvendt hensigtsmæssige metoder og modeller til løsning af specifikke faser, og undervejs justeret metodevalget efter situationen. Der er således kun medtaget arbejdsmetoder der har været nødvendige.
- Systemkrav og brugerønsker er omhyggeligt blevet analyseret og vi har her udvidet vores palette af værktøjer, ved praktisk at anvende UML til usecases og modellering.
- Da stoffet med hensyn til MS Access og VBA var uprøvet og på flere punkter nyt, gav de indlagte spikes rig mulighed for mere dyb teknisk indsigt i velafgrænsede områder, der har ligget helt uden for skolepensummet. I alle tilfælde har spikene dannet et vigtigt grundlag for beslutningsprocesser primært i designfaserne.
- Databasestoffet, der teoretisk har været gennemgået på studiet, er blevet repeteret, uddybet og sat endeligt på plads gennem praktisk erfaring. Ligesom med modellen lever resultatet op til vores forhåbninger og det er lykkedes at overføre noget af den objektorienterede models fleksibilitet til relationsdatabasen.

Konklusion

Vi har levet op til vores egen og uddannelsens målsætning ved, med udgangspunkt i det generelle og teoretiske vi har lært, systematisk og analytisk at have gennemført en konkret praktisk problemløsning. Ved anvendelse af hensigtsmæssige metoder, udvikling af modeller og analyse af mulige teknologier, er opgaven under hensyntagen til Københavns Stadsarkivs mål, gennearbejdet og løst med den påkrævede faglige bredde og dybe.

Afrunding

Diskussion - Proces

Der har for arbejdsprocesserne som sådan ikke været formaliserede mål, men ved aftalen om at indgå i et samarbejde omkring hovedopgaven, har vi naturligvis haft nogle forventninger - ud over blot det at bestå vores afsluttende eksamen.

- Det har været spændende og resursekrævende, at gennemarbejde dette umiddelbart overskuelige projekt, der dog under overfladen har været komplekst nok til at udgøre en reel udfordring.
- Arbejdet i tomandsgruppen har været præget af stor entusiasme, høj arbejdsmoral, dynamik, kreativitet, gensidig sparring og respekt. Projektet igennem er der ”kørt på to motorer” og det har ikke været vanskelig at finde sammen om hverken arbejdsfordeling eller -form. Morsomt nok viser det sig efterfølgende, at vi i dette projekt har byttet om på vores normale grupperoller. Vores store faglige forskelligheder til trods er det lykkedes os at bygge bro med den ”røde bog” [*Objekt Orienteret Analyse & Design*] som en fælles reference. Vi har gennem en meget visuel arbejdsproces tegnet os gennem alle opståede problemer, hvad der afspejles i både rapport og systemdokumentation.
- Forløbet har været understøttet af et omhyggeligt udformet projektgrundlag og en tidlig veldefineret projektplan med udførlige standarder for arbejdet, kvalitet, metoder, dokumentation og værktøjer. Vi har hér sikret fælles mål og næsten helt undgået senere uoverensstemmelser på grund af individuelle fremgangsmåder. Dokumentation er skrevet undervejs og derfor blevet ganske fyldig, ligesom faserne løbende er blevet evalueret og erfaringer ekstraheret.
- Ugentlige arbejdsdage bl.a. på KSA hvor vi har haft samarbejde og møder med andre medarbejdere, men også har haft tid til at finde fælles forståelse for problemstillingerne, samt de uddelegerede opgaver der har kunnet løses i fred og ro i hjemmet, har været en god blanding. Det indledende ønske om at kode parvist (XP) kunne desværre ikke realiseres på grund af geografisk afstand, arbejde og familie. Dette er ærgerligt da det ville have højnet kvaliteten af det samlede projekt.
- Hurtigt udsendt projektlog med velbeskrevne aftaler efter hvert møde, er brugt som huskeliste. Resultaterne er altid efterfølgende blevet evalueret, hvilket har betydet høj effektivitet og lovede arbejder er stort set aldrig løbet ud i sandet.
- Til trods for projektets meget høje grad af strukturering, har vi oplevet næsten alle stadier fra total kontrol, 100 % succes, rolig afklarethed, stress, næsten komplet kuldsejling, midlertidig rådvildhed og fremad igen til fuld ”recovery”. Dette har bekræftet os i, at man ikke kan planlægge sig ud af alle problemer, og at fejltagelser kan undgås i et sådan projekt – det vigtigste er at man er klar til at agere og hele tiden se muligheder for at få det bedste ud af også mindre favorable situationer. Et velforberedt projekt hjælper helt sikkert og har været et godt grundlag for succesen.

Konklusion

En helt igennem prima og lærerig oplevelse, hvor vi har fået erfaring i vigtigheden af god planlægning, faste aftaler, klare mål, ansvarsfordeling og efter vores ”kuldsejling” også ydmyghed for opgaven. I forbindelse med processen er det der har været givende og som vi nu bringer med os videre til kommende projekter: Åbenhed i diskussioner, ligeværdighed, respekt og ikke mindst viljen til at gennemføre sammen.

Det er godt at afslutte Datamatikerstudiet, med en følelse af at vi gennem hele uddannelsesforløbet, opgave for opgave, har kunnet bygge videre på tidligere erfaringer og nu kan selv.

18.1.2 Projektopgaven for Københavns Stadsarkiv

Diskussion - Produkt (MS Access applikation og systemdokumentation)

Formål og målet for selve projektet er formuleret i samarbejde med Københavns Stadsarkiv og beskrevet i projektgrundlaget og sammenholdes nedenstående med det afleverede produkt:

- Det er lykkedes at udvikle og implementere en MS Access applikation der opfylder Københavns Stadsarkiv's ønske om, at kunne intensivere indsamlingen og behandlingen af data om skadede arkivalier. I forlængelse af selve skadesregistreringen er der udviklet en aktionsregistrering der danner grundlag for en rationel beslutningsproces i forbindelse med udvælgelsen af sager til behandling. Der er endvidere foretaget analyse, design og konstruktion af brugergrænseflade og database til at styre den efterfølgende sammensætning og ordre/sagsstyring af behandlingsforløbet, samt sikre kontrol af aftaler og fremdriften for disse. Behandlingsdelen er håndteret og udviklet på tilsvarende måde.
- Der er fastlagt og beskrevet arbejdsrutiner, kompetence- og ansvarsområder i forbindelse med IT-systemet og dets anvendelse, hvorfor den endelige ibrugtagning ikke skulle give anledning til uklarheder medarbejderne imellem. Dette vil med konstruktionen af sikkerhed på brugerniveau yderligere minimeres da de enkelte aktører kun får rettigheder, der stemmer overens med deres arbejdsopgaver og kompetence.
- Der er foretaget analyse, design og konstruktion af brugergrænseflade og database, til registrering af værkstedets arbejdsprocesser, behandlingsforløb og indirekte resurseforbrug for skadede arkivgenstande. Disse indgår i det realiserede system, men er ikke konstrueret med hensyn til funktionalitet jævnfør projektplanens prioritering. Ligeledes mangler der at blive konstrueret udskrivningsrapporter og forespørgsler for deres sammensætning, før værkstedets ønsker om mulighed for dokumentation af arbejdsopgaver og -processer helt er opfyldt.
- IT systemet er implementeret så det kan anvendes fra alle Københavns Stadsarkiv's arbejdsstationer på netværket og fungerer inden for det eksisterende IT-miljø.
- Der er udarbejdet en fyldestgørende systemdokumentation der sammen med applikationen sikre forståelse af hensigterne bag systemets opbygning og virkemåde.
- Projektdeadline 2003.12.19 er overholdt.

Konklusion

Det kan således konkluderes, at de samlede formål og mål for projektet er indfriet, selv om der ikke er tale om et færdigt og fuldt produktionsklart produkt. Det har der på intet tidspunkt været lovet - selv om det må indrømmes at vi som systemudviklere, indledningsvist havde skyhøje ambitioner og gerne havde set en endnu mere afsluttet løsning. På mange måder er arbejdet dog gået betydeligt videre end Københavns Stadsarkiv's konkrete ønsker, for produktet er bevidst fleksibelt planlagt og har stort potentiale med henblik på videre udvikling.

Afrunding

18.2 Perspektivering

I forbindelse med analyse og realisering af projektet er det blevet klart, at denne konkrete opgave meget vel kan ønskes udbygget til at omfatte styring og registrering af andre aktiviteter og behandlingsformer, end Københavns Stadsarkivs ønsker udtrykker. Det har derfor været essentielt for os, at medtage og indbygge denne fleksibilitet i strukturen allerede fra starten. Der er derfor allerede nu mulighed for:

- Sammenkobling til Københavns Stadsarkivs eksisterende arkivdatabase.
- Let tilføjelse af nye arkivgenstande f.eks. mikrofilm, billeder.
- Let tilføjelse af nye aktivitetsregistreringer på linie med skadesregistreringer f.eks. udlånsregistreringer. Sådanne vil kunne indpasses i et tilpasset og styret ordresagsforløb med værkstedet som omdrejningspunkt for indledende og efterfølgende kontrol.
- Let tilføjelse af nye behandlingsregistreringer på linie med konservering f.eks. skanning og ompakning.

Endvidere er der indgående set på mulige forbedringer i forhold til sikkerhed, mail, avanceret datavalidering med regulære udtryk og ADO der vil kunne lette en migrering til en regulær klient/serverløsning. Dette og meget andet der ikke direkte er indgået i det realiserede IT-system, vil både umiddelbart og i tiden fremover danne et uvurderligt grundlag for, at projektet let kan videreudvikles af Martin Trnka i Københavns Stadsarkivs eget regi. En sidegevinst der ikke kan overvurderes.

Københavns kommunes har for 2004 planer om at overgå til Exchange Server, Active Directory og Outlook som en del af et Elektronisk Sags og Databehandlingssystem. Dette giver nye muligheder for at benytte persondata fra et kommende Active Directory frem for den nuværende løsning med egne persondata i systemet. Endvidere kan det give mulighed for at realiser automatiseret e-mail kommunikation frem for papirarbejds gange.

Der er derfor alle muligheder for, at IT-systemet med årene kan videreudvikles til et mere skalerbart og helstøbt "Asset management" værktøj, der sikre rational styring og resursefordeling i forbindelse med Københavns Stadsarkivs håndtering af aktiviteter og behandling af arkivgenstande.

Bilag

19

Efterfølgende bringes alle enkeltstående dokumenter samt projektbilag der har haft betydning for projektets gennemførelse, men som ikke direkte har plads i selve rapporten.

En væsentlig del af bilagene udgøres af spikes, der i vores version er korte eksperimenter ("grundforskning") der giver nødvendig indsigt til at afgøre om det testede kan og skal realiseres og evt. hvordan. Det er således helt bevidst, at der ikke er tale om en tilbunds gående og udtømmende udforskning af emnerne. En sideeffekt har været at spikene har givet en fælles forståelse og referenceramme der har været til stor hjælp

Introduktion

Københavns Stadsarkiv (KSA) er et kulturhistorisk arkiv med 18 fastansatte medarbejdere. Det overordnede formål er, at modtage skriftligt kildemateriale, der har betydning for københavnernes og deres bys historie, samt at gøre det tilgængeligt for myndighederne, offentligheden og forskningen. Desuden skal arkivet bidrage til udforskning, dokumentation og formidling af Københavns historie og arkivforhold.

Størstedelen af KSA's samlinger stammer fra Københavns forvaltning og institutioner. Arkivet modtager også arkivalier, dvs. dokumenter og protokoller fra privatpersoner, firmaer og foreninger med tilknytning til København. De ældste af de arkivalier, der opbevares i Stadsarkivet, er fra slutningen af 1200-tallet, men hovedparten stammer dog fra tiden efter Københavns brand i 1728. I dag omfatter samlingerne ca. 30 hyldekilometer arkivalier og omkring 100.000 kort- og tegninger.

På trods af KSA's bestræbelser for på samvittighedsfuld vis, at overholde sine bevaringsforpligtelser gennem veldisponerede arkivforhold og skånsom håndtering, er det ikke muligt at undgå skader på samlingerne. Disse kan opstå som følge af forkert opbevaring (lys, temperatur og fugt), slid, forkert håndtering, skadedyr eller i form af hærværk, brandskade, vandskade mm.

KSA's konserveringsværksted er kun bemandet med to deltidsansatte, og da reparation, nyindbinding og konserveringsopgaver af natur er særdeles tidskrævende, er det umuligt at reparere alle skader, der opstår over tid. Metoder som nedpakning i æsker, utilgængeliggørelse, mikrofilmning, affotografering og i fremtiden skanning (der ses som arkivernes redningsplanke) må anvendes, og kan gøres til genstand for projektarbejder af veldefineret karakter og varighed. Det er derfor meget vigtigt, at registrering og prioritering af arbejdsopgaver kan foregå på et velovervejede grundlag, hvor der er overblik over omfanget af skader, ligesom det skal være muligt at kontrollere om iværksatte opgaver også afsluttes.

Konserveringsopgaver kom tidligere lidt tilfældigt i stand ved, at medarbejdere foreslog eller sendte skadede arkivalier (og andre opgaver) direkte til værkstedet uden "særlig" overordnet styring. For at introducere en sådan styring og prioritering blev der i 1999 udarbejdet et skaderegistreringsskema, men brugen har været sparsom, og det er svært at danne sig et bedre overblik via disse forstørrede kartotekskort. Det har også vist sig svært at motivere medarbejdere på alle niveauer til at bruge systemet. Der mangler desuden en klar ansvarsfordeling for hvem, der opdaterer ændringer i procedurer og vejledninger, så disse indbyrdes er i overensstemmelse. Grundideen er dog god nok, og man kunne forestille sig, at det ved hjælp af EDB kan gøres simplere at registrere og prioritere skader, samt nemmere at søge, skabe overblik, udvælge og styre både konserveringsopgaver og andre relaterede bevaringsforbedrende tiltag.

Da værkstedet ofte bliver konsulteret ved fastlæggelse af aktionsplaner for enkeltarkivalier eller arkivaliegrupper kræver det nu, at skadeanmeldelserne rundsendes med intern post, eller at der holdes fælles møder. Begge dele er tidskrævende og upraktisk, da KSA's medarbejdere er spredte over flere fysiske lokaliteter.

Bilag

Værkstedet foretager sin egen udvidede registrering af alle typer arbejdsopgaver, samt de processer de gennemløber. Dette sker på kartotekskort og er et tidskrævende arbejde, der ikke giver de ønskede muligheder for søgning og dokumentation af effektiviteten.

Skadesregistrering/aktionsplaner og afledte værkstedsopgaver befinder sig i to forskellige systemer uden en egentlig kobling. En sådan kobling kunne være praktisk.

Foreløbig problemidentifikation

Problemliste
<ol style="list-style-type: none">1. Begrænset overblik over skadede arkivalier.2. Begrænset opfølgning og prioritering af skaderegistreringer.3. Begrænset overblik over aktionsplaner for behandling af skadede arkivalier.4. Begrænset mulighed for at søge, sortere og udskrive i skaderegistreringer og aktionsplaner.5. Begrænset mulighed for styring og igangsættelse af opgaver.6. Begrænset mulighed for at kontrollere færdiggørelse af igangsat arbejde (aktionsplanen er fulgt med opgaven).7. Manglende motivation fra medarbejdere på alle niveauer til at anvende det eksisterende system.8. Nuværende system måske for komplekst.9. Uklare ansvarsområder og kompetencefordeling.10. Fejl i vejledningerne til det nuværende system.11. Nuværende system består af et fysisk eksemplar og kan ikke ses fra andre lokaliteter.12. Registrering af værkstedsopgaver sker manuelt og er langsommeligt.13. Begrænset mulighed for at søge, sortere og udskrive registrerede værkstedsopgaver.14. Manglende kobling mellem skadesregistrering/aktionsplan og værkstedsopgaver.

Symptomer	Mulige årsager
<ol style="list-style-type: none">2. Begrænset opfølgning og prioritering af skadesregistreringer.7. Manglende motivation fra medarbejdere på alle niveauer til at anvende det eksisterende system.	<ol style="list-style-type: none">8. Nuværende system måske for komplekst.9. Uklare ansvarsområder og kompetencefordeling.10. Fejl i vejledningerne til det nuværende system.* Begrænset anvendelighed i almindelighed.

Figur 19-1. Indledningsvis liste over umiddelbare problemområder og deres mulige sammenhæng.

Foreløbig behovsanalyse

Med udgangspunkt i de eksisterende manuelle systemer for skadesregistrering, aktionsplanlægning samt registrering af konserveringsopgaver, er der i KSA behov for at udvikle og dokumentere et let tilgængeligt EDB-system med følgende karakteristika: Skadesregistrering af arkivalier og afledte aktionsplaner skal kunne indtastes og danne grundlag for prioritering, udvælgelse og behandling.

Indbyggede faciliteter skal skabe overblik og muliggøre planlægning af KSA's begrænsede resurser og styrke kontrollen med den efterfølgende behandling.

På en hensigtsmæssig måde sikre dokumentation af arbejdsprocesser, ressourceforbrug osv. for de forskelligartede konserveringsopgaver, der løses på værkstedet.

Kunne anvendes fra alle centrale og decentrale arbejdspladser med netopkobling.

Klarlægge og understøtte de enkelte medarbejderes arbejdsrutiner, kompetence- og ansvarsområder i forbindelse med systemet.

Foreløbig afgrænsning

Da opgaven formentlig ikke påvirker grundstrukturen i KSA, vil virksomhedsdelen begrænse sig til en gennemgang af organisationsopbygningen og de følger, som ændringer i arbejdsrutiner, ansvarsområder og teknologi kan have på omgivelserne.

Hvis tiden tillader det, kan det være interessant, at inddrage det aktuelle aspekt omkring det kommunale bureaukratis delvise transformering til en mere projektorienteret struktur, der styres via kontrakter - en spændende, men ikke helt problemløs sammensmeltning af to modsatrettede virksomhedskulturer.

Opgaven kan vise sig for stor, hvorfor implementering af værkstedsdatabasen må udelades. Problemstillinger omkring datasikkerhed og transaktionsstyring berøres kun i begrænset grad.

Det virker oplagt at koble KSA's arkivdatabase til både skaderegistreringen og værkstedsdatabasen, men søgefunktionen kan være en så komplicerende faktor, at det må udelades i det aktuelle projekt.

Projektet set i forhold til kravene til et eksamensprojekt

Selv om der her formodentligt primært bliver tale om en MS Access databaseløsning, mener vi bestemt, at opgaven kan inddrage de vigtigste pensumgrupper fra både Systemudvikling, Strategisk Projektledelse, Organisation og Logistik, Objekt Orienteret Analyse & Design, Distribuerede systemer, Relationsbase stof og programmering.

Der skulle således blive rig mulighed for at bevise, at vi kan anvende vores meget generelle viden til at opfylde målsætningen fra ministeriets bekendtgørelse og Niels Brocks eksamensvejledning til 5. semester om:

At den studerende både skriftligt og mundtligt kan dokumenterer evnen til systematisk og analytisk at gennemføre en problemløsning samt til at forbinde teori og praksis i et bredt sammenhængende fagligt perspektiv.

Projektaftale

Formål

Med udgangspunkt i projektoplægget og de eksisterende manuelle systemer for skadesregistrering/aktionsplanlægning samt registrering af konserveringsopgaver, er det KSA's ønske at:

- Intensivere indsamlingen og behandlingen af data om skadede arkivalier.
- Forbedre grundlaget for en rationel beslutningsproces og kontrol omkring handlingsplanerne og udførelse.
- Skabe klarhed over arbejdsgange.
- Forbedre værkstedets muligheder for dokumentation af arbejdsopgaver og -processer.

- Systemudviklerne bruger projektet til deres afsluttende hovedopgave under Datamatikeruddannelsen

Mål

- Udvikle og implementere et IT-system hvor skadesregistreringer af arkivalier og afledte aktionsplaner kan indtastes og danne grundlag for prioritering og planlægning af efterfølgende behandling, samt muliggøre kontrol med samme.
- Fastlægge og beskrive arbejdsrutiner, kompetence- og ansvarsområder i forbindelse med systemet og dets anvendelse.
- Udvikle et IT-system, der kan dokumentere arbejdsprocesser og evt. ressourceforbrug mm. for skadesregistreringer der behandles på værkstedet, og at disse kan ses i forbindelse med skadesregistreringssystemet.
- Implementere IT-systemet så det kan anvendes fra centrale og decentrale PC'er med netopkobling indenfor det eksisterende IT-miljø.
- Udarbejde systemdokumentation for IT-systemet.
- Projektet deadline 2003.12.19

- Systemudviklerne består deres hovedopgave eksamen under datamatikeruddannelsen.

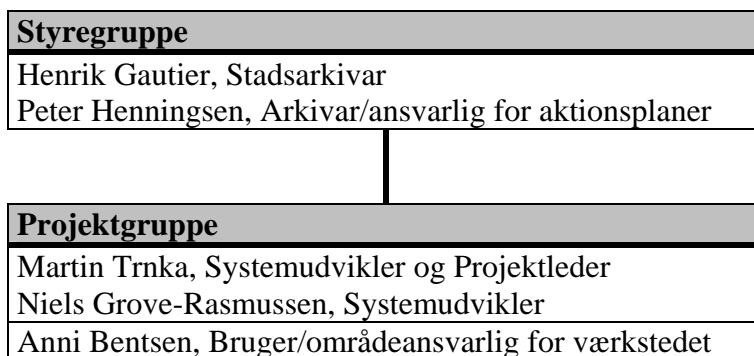
Projektorganisering

Projektet forankres sådan, at Stadsarkivaren der har taget beslutning om og afstukket retningslinier for dets omfang, formål og de involveredes kompetencefordeling, samt at den arkivar, der skal forestå arbejdet med aktionsplanerne, udgør en formel styregruppe.

Områdelederen for restaureringsværkstedet, der har ønsket en værkstedsdatabase, og er dens primære bruger, indgår sammen med 2 systemudviklerne i projektgruppen. MT får status af Projektleder (PL) og det formelle ansvar for projektets kvalitet og gennemførelse ligger her. PL har reel bemyndigelse til selv at træffe overordnede beslutninger for projektet, uden først at sikre styregruppegodkendelse, da rammerne er velkendte. Projektet har endvidere bemyndigelse til at trække på interne ressourcer i forbindelse med planlægning, udvikling og testning.

Bilag

Projektorganisations model



Figur 19-2. Da gruppen er så begrænset, er der ikke behov for at indføje et selvstændigt koordineringsled mellem styre- og projektgruppe.

Til dette forholdsvis beskedne projekt er opbygningen ideel, da styregruppen indbefatter Stadsarkivets øverste myndighed, og der dermed i princippet er kort vej til at få afgjort principielle og økonomiske problemer. Projektgruppen har deltagelse af en primær bruger for derigennem at sikre en hensigtsmæssig brugerrettet løsning af opgaven, opbygge en viden om opgavens løsning hos medarbejderen og dermed sikre optimal motivation og tilfredshed med det nye system. Projektgruppens sammensætning og beskedne omfang kan være en fordel, da ændringer og nye ideer hurtigt kan implementeres og testes direkte. I mere overordnede brainstormfaser må der trækkes på interne medarbejdere, men opgavens natur og beskrivelse har på forhånd fastlagt store dele af strukturen.

Resurser

Store dele af projektarbejdet vil foregå uden for normal arbejdstid, men projektgruppen tilstræber en ugentlig arbejdsdag i KSA. Der skal endvidere være mulighed for at foretage system- og brugertest i virksomheden undervejs i forløbet.

Der er fra KSA's side ikke afsat økonomiske resurser til projektet.

Teknologi

Opgaven skal løses inden for rammerne af KSA's nuværende IT-konfiguration.

- Microsoft Windows 2000 professional DK.
- Microsoft Office XP professional DK.
- Københavns kommunes LAN/WAN, der er ikke adgang til en dedikeret databaseserver.

Tidsplan

Tidsplanen følger hovedopgaveforløbet med start primo maj 2003 og forventet afslutning medio december 2003.

IT-projektets tilpasning til Datamatikerhovedopgave

Det skal bemærkes, at projektet i relation til datamatikerhovedopgaven har en fast deadline og at forløbet ikke kan forskydes eller strækkes.

Systemet installeres, demonstreres og overdrages ”as is” inden projektafslutning, hvorefter projektgruppens forpligtigelser i forhold til IT-systemet ophører.

Rettigheder til programkoden tilhører systemudviklerne, men Københavns Stadsarkiv har fuld brugsret og tilladelse til selv at videreudvikle systemet efterfølgende.

Ledelsen for KSA udfærdiger afslutningsvist en skriftlig udtalelse om projekt og forløb.

Kravspecifikation

Bilag 19.2

Formålet med kravspecifikationen er, at formulere en bindende kontrakt for produktet af projektarbejdet, som der efterfølgende kan testes op imod. Det kan være en fordel at udarbejde kravene allerede under OOA fasen, så de hurtigt kan komme til høring (evt. udbud) inden der er taget for omfattende afgørelser angående struktur mm. Omvendt kan udskydelse til senere i forløbet sikre større præcision.

Indhold

Kort overordnet systembeskrivelse

Detaljeret specificering af målbare detailkrav til:

- Krav til tilpasning til hard- og software (arkitektur).
- Krav generelt for applikationen.
- Krav til skadesregistrering.
- Krav til aktionsregistrering.
- Krav til handlingsplan.
- Krav til behandlingsregistrering.
- Krav til projekt deadlines.

Fremtidig arbejdsgangsbeskrivelse og -opdeling ("sunny day cases") for:

- Skadesregistrering.
- Aktionsregistrering.
- Handlingsplan.
- Behandlingsregistrering.

Kort overordnet systembeskrivelse

Det ønskede IT-system, der skal kunne håndtere skaderegistrering af Københavns Stadsarkivs genstande, skal overordnet kunne håndtere selve registreringen af de observerede skadede genstande og kunne betjenes af alle ansatte.

I umiddelbar tilknytning og på baggrund af udtræk af skaderegistreringerne, skal det være muligt for den ansvarshavende arkivar i første omgang, at prioritere genstandene og foreslå aktionsplaner, der siden udvælges enkeltvist eller i større partier (handlingsplaner/ordre) til behandling. Det skal her være muligt at se og kontrollere de igangsatte behandlingsopgaver og den samlede ordrestatus. Der er behov for udprint af enkelte genstandes status og udskrift over indholdet af partier til behandling.

Langt den overvejende del af behandlingsopgaverne ender på Københavns Stadsarkivs Konserveringsværksted. Der er her behov for udvidet registrering af gennemførte arbejdsprocesser for de opgaver, der kommer i stand gennem skaderegistrering. Ligeledes er der til brug for rapporter, behov for udprintet dokumentation af konserveringsopgaverne.

Bilag

Detailkrav

Detailkravene er her prioriteret groft efter følgende skala: 1-3 er absolutte krav, 4-6 er krav der helst skal opfyldes og endelig 7-9 er krav der gerne må opfyldes.

Nr.	Krav til hard- og software tilpasning	Prio.	Bemærkning
1.1	Databasen, med forespørgsler, formularer, rapporter og søge muligheder skal oprettes så den kan afvikles under MS MS Access 2002 der også bruges til KSA's overordnede arkivaliedatabase.	1	
1.2	Databasen skal kunne afvikles, fra en centralt placeret filserver over LAN på centrale arbejdsstationer og WAN (100 Mbit) på de decentrale arbejdsstationer, under styresystemet Windows 2000. Der er ikke mulighed for adgang til en SQL Server eller lignende.	1	
1.3	Applikationen skal kunne afvikles som en runtimeversion da kun et fåtal af arbejdsstationerne har MS Access 2002.	1	
1.4	Arbejdsstationerne i KSA er for tiden bestykket med processorer på mellem 450 og 2000 MHz og med min 256 MB ram. Programmet skal kunne afvikles på selv de ringeste af de nuværende maskiner.	1	
1.5	GUI tilpasses til 17" CRT skærme med opløsning 1024 x 868 der er standard (husk at tage højde for at brugerene normalt har vist proceslinien).	1	
1.6	Der er ikke særlige krav til datasikkerhed, kryptering mm da data ikke indeholder personinformationer og værktøjet "kun" er et hjælpeværktøj til styring af processflow. Backup håndteres automatisk på linie med IT-afdelingens backuppolicy for almindelige filer.		Se sikkerhed spike bilag 19.5 s. 147

Nr.	Krav generelt for applikationen	Prio.	Bemærkning
2.1	Samlet hovedapplikationsbillede for hovedparten af funktionaliteten (ikke valg via switchboard)	1	
2.2	Navigation mellem skærbilleder (evt. via menu valg).	1	
2.3	Statuslinie hjælpetekst i alle indtastningsfelter.	1	
2.4	Hjælpe-menu med mere omfattende hjælpetekster	8	
2.5	Statusliste der viser sagens øjeblikkelige status.	1	jf. 4.1.4 s. 36
2.6	Rettighedsstyring og sikkerhed (evt. password beskyttelse for all andre end almindelige brugere).	5	Se sikkerhed spike bilag 19.5 s. 147
2.7	Brugergrænsefladen skal fremstå i Windows2000 stil.	1	

Nr.	Krav til skadesregistrering	Prio.	Bemærkning
3.1	Tilgængelig for alle. Redigerbar for ansatte der står opført i systemet.	5	Se sikkerhed spike bilag 19.5 s. 147
3.2	Registrering af stamdata om relevante genstande (arkivalier, kort, bøger mm.) i overensstemmelse med den eksisterende arkivdatabase.	1	jf. 4.1.4 s. 36
3.3	Kobling til arkivdatabase og dennes søgefunktion.	9	Under omlægning
3.4	Registrering af arkivalietypen og materiale.	1	jf. 4.1.4 s. 36
3.5	Registrering af observerede skader og evt. andre skader (note).	1	jf. 4.1.4 s. 36
3.6	Godkendelse og validering af skadesregistreringen. Sagshistorik med registranten, status og dato.	2	jf. 4.1.4 s. 36
3.7	Redigering af skadesregistreringer så længe de endnu ikke er aktionsregistrerede.	5	
3.8	Sletning af skadesregistreringer så længe de endnu ikke er aktionsregistrerede.	5	
3.9	Søgning/visning af skadesregistreringer (på baggrund af genstand, type, materiale og/eller skader).	4	
3.10	Udskrivning af udvalgt skadesregistrering.	7	
3.11	Udskrivning af alle skadesregistreringer (søgning).	7	
3.12	Automatisk notifikation af den ansvarlige arkivar via e-mail eller rapportudskrift (evt. for hver tiende ny ikke aktionsbehandlede skadesregistrering).	8	

Nr.	Krav til aktionsregistrering	Prio.	Bemærkning
4.1	Tilgængelig for alle, men kan kun redigeres af Arkivar og Konservator (password).	5	Se sikkerhed spike bilag 19.5 s. 147
4.2	Registrering af klassifikation.	1	jf. 4.1.4 s. 36
4.3	Registrering af prioritet.	1	jf. 4.1.4 s. 36
4.4	Registrering af aktioner og evt. anden aktion (note).	1	jf. 4.1.4 s. 36
4.5	Godkendelse og validering af aktionsregistreringen. Sagshistorik med registranten, status og dato.	2	jf. 4.1.4 s. 36
4.6	Redigering af egne aktionsregistreringer så længe de endnu ikke indgår i en handlingsplan.	3	
4.7	Sletning af egne aktionsregistreringer så længe de endnu ikke indgår i en handlingsplan.	3	
4.8	Søgning/visning af aktionsregistreringer (på baggrund af genstand, klassifikation, prioritet, aktion mm.).	4	
4.9	Udskrivning af udvalgt skades-/aktionsregistrering.	7	
4.10	Udskrivning af alle skades-/aktionsregistreringer (søgning).	7	

Bilag

Nr.	Krav til handlingsplan	Prio.	Bemærkning
5.1	Tilgængelig for alle, men kan kun redigeres af Arkivar og Konservator (password).	5	Se sikkerhed spike bilag 19.5 s. 147
5.2	Valg og fravalg af en eller flere skadesregistrerede genstande til behandling på grundlag af: Oprindelse, titel, type, materiale, klassificering, prioritet mm.	1	
5.3	Godkendelse og validering af preordre. Ordrehistorik med registranten, status og dato.	2	jf. 4.1.4 s. 36
5.4	Registrering af aftale med viderebehandler.	1	jf. 4.1.4 s. 36
5.5	Registrering af instruktioner i forbindelse med viderebehandlingen.	1	jf. 4.1.4 s. 36
5.6	Registrering af hvem der skal fremtage ordren til transport.	1	Ny fra slice 3
5.7	Registrering af instruktion i forbindelse med fremtagningen.	1	Ny fra slice 3
5.8	Registrering af hvem der skal transportere ordren til behandlingen.	1	Ny fra slice 3
5.9	Registrering af instruktion i forbindelse med transport	1	Ny fra slice3
5.10	Godkendelse og validering af iværksættelse. Ordrehistorik med registranten, status og dato.	2	jf. 4.1.4 s. 36
5.11	Godkendelse og validering af ordreaflutning. Ordrehistorik med registranten, status og dato.	2	jf. 4.1.4 s. 36
5.12	Redigering af egne handlingsplaner så længe de ikke er iværksat.	3	
5.13	Sletning af egne handlingsplaner så længe de ikke er iværksat.	3	
5.14	Søgning/visning af handlingsplaner (på baggrund af genstand, status mm.).	4	
5.15	Udskrivning af udvalgt handlingsplan (arbejdsseddel/ordre) med felt til færdigmelding af arbejdet.	7	
5.16	Udskrivning af alle handlingsplaner (søg).	7	
5.17	Udskrivning af udvalgt skaderegistrering/aktionsregistrering der indgår i partiet.	7	
5.18	Udskrivning af alle skaderegistreringer/aktionsplaner der indgår i partiet.	7	
5.19	Automatisk e-mail notifikation eller mulighed for udskrivning af rapport til viderebehandler.	8	
5.20	Automatisk e-mail notifikation eller mulighed for udskrivning af rapport til fremtagning og transport.	8	

Nr.	Krav til behandlingsregistrering (konservering)	Prio.	Bemærkning
6.1	Tilgængelig for alle, men kan kun redigeres af Konservator.	5	Se sikkerhed spike bilag 19.5 s. 147
6.2	Godkendelse af påbegyndelse. Sagshistorik med registranten, status og dato.	2	jf. 4.1.4 s. 36
6.3	Registrering af arkivaliebeskrivelse før behandling.	1	jf. 4.1.4 s. 36
6.4	Registrering af arbejdsprocesser og evt. anden arbejdsproces (note).	1	jf. 4.1.4 s. 36
6.5	Registrering af arkivaliebeskrivelse efter behandling.	1	jf. 4.1.4 s. 36
6.6	Godkendelse af færdiggørelse. Sagshistorik med registranten, status og dato.	2	jf. 4.1.4 s. 36
6.7	Redigering af egne behandlingsregistreringer så længe behandlingen ikke er meldt afsluttet.	3	
6.8	Sletning af egne behandlingsregistreringer så længe behandlingen ikke er meldt afsluttet.	3	
6.9	Søgning/visning af behandlingsregistreringer (på baggrund af genstand, type, processer, status, år, inter.v).	4	
6.10	Udskrivning af udvalgt behandlingsregistrering.	7	
6.11	Udskrivnings af alle behandlingsregistreringer (søgning).	7	
6.12	Automatisk e-mail notifikation eller rapportudskrift til ansvarlig arkivar ved færdigmelding af samtlige arkivalier i et parti (muligvis auto ordreafslutning).	8	

Nr.	Krav til projekt deadlines	Prio.	Bemærkning
7.1	Seneste dato for implementering, demonstration og overdragelse af prototype eller 1.version til KSA.	1	2003.12.19
	Seneste dato for aflevering af hovedopgave (rapport, systemdokumentation og program).	1	2003.12.19 kl. 12:00

Figur 19-3. Detailkravene er her prioriteret groft efter følgende skala: 1-3 er absolutte krav, 4-6 er krav der helst skal opfyldes og endelig 7-9 er krav der gerne må opfyldes.

Bilag

Fremtidig arbejdsgangsbeskrivelse og -opdeling

Nedenstående gennemgås kun de mest almindelige "sunny day" brugsmønstre og arbejds gange.

Fremtidig arbejdsgangsbeskrivelse

Skadesregistrering	Bemærkning
<p>Alle medarbejdere skal fra en vilkårlig PC kunne se og registre observerede skadede genstande samt se deres øjeblikkelige sagsstatus. Medarbejder skal både kunne redigere egne og andres skadesregistreringer.</p> <p>OBS. Det kan i nogle tilfælde være nemmere eller den eneste mulighed at udfylde en "grøn skadesregistrering" lokalt, og på et senere tidspunkt oprette en elektronisk registrering på baggrund heraf. Det er naturligvis et personligt valg.</p> <p>Skadet genstand registreres.</p> <ol style="list-style-type: none">1. Åben programmet til skadesregistrering (evt. password).2. Find det aktuelle skærbillede.3. Registrer typen.4. Registrer materialet.5. Registrer arkivaliets stamdata.6. Registrer synlige skader.7. Godkend eller annuller skadesregistreret-status. <i>(Auto Email til arkivar ved hver10. ubehandlede skadesregistrering).</i> <i>Gentag evt. processen for flere arkivalier eller:</i>8. Luk programmet til skadesregistrering. <p><i>Udfyld en registreringsseddel for hver registrering (lysegrøn fortrykt PostIt der findes i alle arkivafsnit) der indklæbes på forpermens inderside eller vedlægges sagen. Så kan andre umiddelbart se, at en registrering har fundet sted.</i></p>	<p>jf. 4.1.4 s. 36 jf. 4.1.4 s. 36 jf. 4.1.4 s. 36 jf. 4.1.4 s. 36</p>

Aktionsregistrering	Bemærkning
<p>Alle medarbejdere skal kunne se aktionsdelen og sagsstatus, men kun den ansvarlige arkivar og eventuelt værkstedet skal kunne oprette og redigere data.</p> <p>Der er skadesregistreret nye arkivalier.</p> <ol style="list-style-type: none">1. Åben programmet til skadesregistrering (evt. password).2. Find det aktuelle skærbillede.3. Gennemse skadesregistreringerne eller søg efter nye.4. Registrer klassificeringen.5. Registrer prioriteten.6. Registrer relevant aktion.7. Godkend eller annuller aktionsregistreret-status.8. Luk programmet til skadesregistrering. <p><i>Gentag evt. processen for flere skadesregistreringer.</i></p>	<p>jf. 4.1.4 s. 36 jf. 4.1.4 s. 36 jf. 4.1.4 s. 36 jf. 4.1.4 s. 36</p>

Handlingsplan	Bemærkning
<p>Alle medarbejdere skal kunne se hvornår og hvilke genstande, der er udtaget til behandling samt deres øjeblikkelige ordre- og sagsstatus. Kun den ansvarlige arkivar og eventuelt værkstedet skal kunne oprette og redigere i data.</p> <ol style="list-style-type: none"> Der er behov for at sammensætte et rationelt parti til behandling. Åben programmet til skadesregistrering (evt. password). Find det aktuelle skærmbillede. Udvælg en eller flere aktionsregistreringer til behandling Godkend eller annuller preordre-status. <p><i>Aftal for hvert parti hvad der skal ske, og hvem der skal forestå arbejdet.</i></p> <ol style="list-style-type: none"> Registrer initialerne for hvem fremtagningen er aftalt med. Registrer evt. uddybende instruktion i forbindelse med fremtagningen. Registrer initialerne for hvem transport er aftalt med. Registrer evt. uddybende instruktion i forbindelse med transporten. Registrer initialerne for hvem behandlingen er aftalt med. Registrer evt. uddybende instruktion i forbindelse med behandlingen. Godkend eller annuller igangsæt-status. <p><i>(Auto E-mail eller rapportudskrift af ordreliste med nødvendig instruktion for partiet til viderebehandler.</i> <i>Auto E-mail eller rapportudskrift af ordreliste med nødvendig instruktion om fremtagning og afsendelse.).</i> <i>Gentag evt. processen for flere partier.</i></p> <ol style="list-style-type: none"> Luk programmet til skadesregistrering. 	<p>jf. 4.1.4 s. 36</p> <p>Ny fra slice 3</p> <p>Ny fra slice 3</p> <p>Ny fra slice 3</p> <p>jf. 4.1.4 s. 36</p> <p>jf. 4.1.4 s. 36</p>

Behandlingsregistrering (konservering)	Bemærkning
<p>Alle medarbejdere skal kunne se behandlingsdelen og øjeblikkelig sags- og ordrestatus, men kun ansatte på værkstedet skal kunne oprette og redigere i data.</p> <p>Ved påbegyndelse af de enkelte sager:</p> <ol style="list-style-type: none"> Åben programmet til skadesregistrering (evt. password). Find det aktuelle skærmbillede. Find det aktuelle arkivalie/sag. Godkend eller annuller påbegyndt-status. Beskriv evt. arkivaliet før behandlingen. Registrer anvendte arbejdsprocesser. Beskriv evt. arkivaliet efter behandlingen. Godkend eller annuller afsluttet-status. <p><i>Gentag evt. processen for flere arkivalier.</i></p> <ol style="list-style-type: none"> Luk programmet til skadesregistrering. <p><i>Hvis alle sager i en ordre er færdigbehandlet vises det på skærmen og det sender en færdigmelding til ordreiværksætteren i form af en autogenereret E-mail eller rapportudskrift af ordren (muligvis sættes ordrens afsluttetstatus automatisk).</i> <i>Arkivalierne sendes tilbage med kørslen.</i></p>	<p>jf. 4.1.4 s. 36</p> <p>jf. 4.1.4 s. 36</p> <p>jf. 4.1.4 s. 36</p> <p>jf. 4.1.4 s. 36</p> <p>jf. 4.1.4 s. 36</p>

Figur 19-4. Forslag til fremtidige arbejdsgangsbeskrivelser

Testspecifikation

Bilag 19.3

Formålet med Testspecifikationen er at sikre IT-systemets kvalitet lever op til kravspecifikationen, og det aftalemæssige grundlag dermed er overholdt ved overdragelsen. Der er til denne opgave kun planlagt accepttest af overordnet funktionalitet og en rudimentær brugertest der tager udgangspunkt i arbejdsgangsscenerne fra kravspecifikationen.

De relevante tests gennemløbes for hver slice, men er jævnfør projektbeskrivelsen kun realiseret fuldt ud for slice 1.

Indhold

Accepttest:

- Test af hard- og software.
- Test af generelle krav til applikationen.
- Test af skadesregistrering (faneblad).
- Test af aktionsregistrering (faneblad).
- Test af handlingsplan.
- Test af behandlingsregistrering (faneblad).
- Test af projekt deadlines overholdelse.

Brugertest (systemtest) af arbejdsganges sammenhæng med brugerinterfacet:

- Skadesregistrering.
- Aktionsregistrering.
- Handlingsplan.
- Behandlingsregistrering (konsivering).

Accepttest

Accepttest specifikation

Nr.	Test af hard- og software	Fase	Test - ok
1.1	Kan IT-systemet installeres på filserveren ifølge kravspecifikationen og kan det ses fra arbejdsstationerne.	Slice 1	2003.11.10
1.2	Kan IT-systemet afvikles fra arbejdsstationer KSA.	Slice 1	2003.11.10
1.3	Fungerer applikationen afviklet som runtimeversion (menulinier og hjælp).	Slice 1	2003.11.10
1.4	Test at IT-systemet virker på de svageste computere der er placeret på lokationen Artillerivej 126.	Slice 1	2003.11.10
1.5	Passer GUI'et ved maksimering til 17" CTR skærme – også når brugere har en fast proceslinie vist nederst.	Slice 3	2003.12.16
1.6			

Bilag

Nr.	Test af generelle krav til applikationen	Fase	Test - ok
2.1	Kan programmet startes fra et ikon. Leder aktivering af knapper til de ønskede programdele.	Slice 1	2003.11.10
2.2	Kan der navigeres mellem skærbilleder på en intuitiv måde.		
2.3	Eksisterer og vises statuslinieliniehjælp for alle indtastningsfelter.	Slice 1	2003.11.10
2.4	Er der konstrueret en hjælpemenu og virker den for de enkelte felter.		
2.5	Vises status korrekt i den rækkefølge de godkendes. Indtast et normalt gennemløb med alle statusposter.		
2.6	Fungerer sikkerhed på brugerniveau/rettighedsstyring		Se sikkerhed spike bilag 19.5 s. 147
2.7	Fremstår brugergrænsefladen i Windows 2000 stil.	Slice 2	2003.13.16

Nr.	Test af skadesregistrering (faneblad)	Fase	Test - ok
3.1	Kan alle ansatte åbne og redigere skadesregistreringer.	Slice 1	2003.11.10
3.2	Er der overensstemmelse mellem datatyperne i stamdata om arkivaliet i IT-systemet og den eksisterende arkivdatabase.	Slice 1	2003.11.10
3.3	Kan et arkivalie søges/kobles direkte i Arkivdatabase.		
3.4	Kan arkivalie typen og materialet vælges som det første.	Slice 1	2003.11.10
3.5	Kan skade(r) vælges/fravælges og/eller kan der skrives en note om anden skade. Valg af skade(r) og/eller note om anden skade skal give anledning til at "Skadesregistreret" status kan godkendes, hvis type, materiale og arkivalie eksisterer.	Slice 1	2003.11.10
3.6	Kan der foretages en godkendelse der vil medføre en datavalidering og efterfølgende at initialerne, handlingens fase "Skadesregistreret" og dagsdato vises i sagshistorikken og gemmes korrekt.	Slice 1	2003.11.10
3.7	Kan skadesregistreringer rettes og gen-godkendes så længe de ikke er Aktionsregistrerede.	Slice 1	2003.11.10
3.8	Kan sletning foregå så længe skadesregistreringen ikke er Aktionsregistrerede.	Slice 1	2003.11.10
3.9	Kan der søges på alle p.t. kendte datatyper og kombinationer heraf, og vises de i fortløbende rækkefølge i formularvinduet. Virker forespørgslen korrekt. Kan der bladres frem og tilbage i de søgte eller filtrerede skadesregistreringer.	Slice 1	2003.11.10 (avanceret søgning mangler)
3.10	Udskrives den aktuelle genstand/skadesregistrering korrekt.		
3.11	Udskrives alle genstand/skaderegistreringer fra en søgning korrekt.		
3.12	Genereres og afsendes der automatisk en notifikations e-mail til den ansvarshavende arkivar for hver tiende ny skadesregistrering, der ikke er aktionsregistreret.		

Nr.	Test af aktionsregistrering (faneblad)	Fase	Test - ok
4.1	Er aktionsregistreringen synlig for alle og kan data kun indtastes, ændres og godkendes af Arkivar og Konservator.		
4.2	Kan klassifikation vælges og gemmes den siden korrekt.	Slice 1	2003.11.10
4.3	Kan prioritet vælges og gemmes den siden korrekt.	Slice 1	2003.11.10
4.4	Kan aktione(er) vælges/fravælges og/eller kan der skrives en note om anden aktion. Valg af aktion(er) og/eller note om anden aktion skal give anledning til at "Aktionsregistreret" status kan godkendes, hvis klassificering og prioritet eksisterer.	Slice 1	2003.11.10
4.5	Kan der foretages en godkendelse der medfører en datavalidering og efterfølgende at initialerne, handlingens fase "Aktionsregistreret" og dagsdato vises i sagshistorikken og gemmes korrekt.	Slice 1	2003.11.10
4.6	Kan aktionsregistreringer rettes og gen-godkendes så længe de ikke indgår i en handlingsplan.	Slice 1	2003.11.10
4.7	Kan sletning foregå så længe de ikke indgår i en handlingsplan.	Slice 1	2003.11.10
4.8	Kan der søges på alle p.t. kendte datatyper og kombinationer heraf, og vises de i fortløbende rækkefølge i formularvinduet. Virker forespørgslen korrekt. Kan der bladres frem og tilbage i de søgte eller filtrerede aktionsregistreringer.	Slice 1	2003.11.10 (avanceret søgning mangler)
4.9	Udskrives den aktuelle skades- aktionsregistrering korrekt.		
4.10	Udskrives skades- aktionsregistreringer fra søgning korrekt.		

Nr.	Test af handlingsplan	Fase	Test – ok
5.1	Er handlingsplaner synlig for alle og kan data kun indtastes, ændres og godkendes af Arkivar og Konservator.		
5.2	Kan en eller flere arkivalier vælges fra alle aktionsregistreringer der endnu ikke indgår i en handlingsplan efter: Arkivskaber, titel, type, materiale, klassificering og prioritet. Det samme arkivalie må ikke kunne vælges flere gange Kan valgte arkivalier fravælges.		
5.3	Kan der foretages godkendelse der medfører en datavalidering og efterfølgende at initialerne, handlingens fase "Preordre" og dagsdato vises i ordrehistorikken og gemmes korrekt.		
5.4	Er det muligt at vælge og ændre valget af hvilken medarbejder viderebehandling er aftalt med.		
5.5	Kan mulig instruktion i forbindelse med viderebehandlingen, indtastes, ændres og slettes. Gemmes den korrekt.		
5.6	Er det muligt at vælge og ændre valget af hvilken medarbejder fremtagning er aftalt med.		
5.7	Kan mulig instruktion i forbindelse med fremtagning, indtastes, ændres og slettes. Gemmes den korrekt.		
5.8	Er det muligt at vælge og ændre valget af hvilken medarbejder transport er aftalt med.		
5.9	Kan mulig instruktion i forbindelse med transport, indtastes, ændres og slettes. Gemmes den korrekt.		

Bilag

5.10	Kan der foretages godkendelse der medfører en datavalidering og efterfølgende at initialerne, handlingens fase "Iværksat" og dagsdato vises i ordrehistorikken og gemmes korrekt.		
5.11	Kan der foretages godkendelse der medfører en datavalidering og efterfølgende at initialerne, handlingens fase "Afsluttet" og dagsdato vises i ordrehistorikken og gemmes korrekt. Denne fase kan evt. auto-godkendes når den samlede ordres enkelte genstande alle har opnået status "Færdig".		
5.12	Kan handlingsplaner rettes og gen-godkendes så længe de ikke er "Iværksat".		
5.13	Kan handlingsplaner slettes så længe de ikke er "Iværksat".		
5.14	Kan der søges efter ordre på genstand- og statusdata. Virker forespørgslen korrekt. Kan der bladres frem og tilbage i handlingsplanerne, hvis der er flere ordrer med de søgte data.		
5.15	Udskrives den aktuelle handlingsplan korrekt. Er der et felt til færdigmelding af arbejdet.		
5.16	Udskrives alle handlingsplaner fra en søgning korrekt.		
5.17	Udskrives en udvalgt skaderegistrering/aktionsregistrering der indgår i partiet korrekt.		
5.18	Udskrives alle skaderegistreringer/aktionsplaner der indgår i partiet korrekt.		
5.19	E-mail notificeres medarbejderen der skal forestå viderebehandlingen automatisk når "Iværksat" status for en handlingsplan er godkendt, eller kan der udskrives en rapport der kan sendes med intern post.		
5.20	E-mail notificeres medarbejderen der skal forestå fremtagning og udbringning automatisk når "Iværksat" status for en handlingsplan er godkendt, eller kan der udskrives en rapport der kan sendes med intern post.		

Nr.	Test af behandlingsregistrering (faneblad)	Fase	Test – ok
6.1	Er behandlingsregistreringen synlig for alle og kan data kun indtastes, ændres og godkendes af Konservator.		
6.2	Kan der foretages godkendelse der medfører at initialerne, handlingens fase "Påbegyndt" og dagsdato vises i sagshistorikken og gemmes korrekt.		
6.3	Kan før-beskrivelse (valgfri) af arkivaliet indtastes, ændres, og slettes. Gemmes den korrekt.		
6.4	Kan proces(er) vælges/fravælges og/eller kan der skrives en note om anden proces. Valg af proces(er) og/eller note om anden proces skal give anledning til at "Færdig" status kan godkendes.		
6.5	Kan efter-beskrivelse (valgfri) af arkivaliet indtastes, ændres, og slettes. Gemmes den korrekt.		
6.6	Kan der foretages godkendelse der medføre at initialerne, handlingens fase "Færdig" og dagsdato vises i sagshistorikken og gemmes korrekt.		
6.7	Kan behandlingsregistreringer rettes og evt. gen-godkendes så længe den handlingsplan de indgår i ikke har status "Færdig".		
6.8	Kan behandlingsregistreringer slettes så længe den handlingsplan de indgår i ikke har status "Færdig".		
6.9	Kan der søges på alle pt. kendte datatyper og kombinationer heraf, og vises behandlingsplanerne i fortløbende rækkefølge i formularvinduet. Virker forespørgslen korrekt. Kan der evt. bladres frem og tilbage i de søgte og/eller filtrerede behandlingsregistreringer.		
6.10	Udskrives den aktuelle behandlingsregistrering korrekt.		
6.11	Udskrives alle behandlingsregistreringer fra en søgning korrekt.		
6.12	Notificeres den ansvarlige arkivar (eller iværksætter af handlingsplanen) automatisk når "Færdig" status for alle genstande i en handlingsplan er opnået.		

Nr.	Test af projekt deadlines overholdelse	Fase	Test – ok
7.1	Blev seneste dato for implementering, demonstration og overdragelse af prototype overholdt.		

Figur 19-5. Acceptestene er mappet direkte fra kravspecifikationen og nummereringen passer til denne. Der beskrives kort hvad og evt. hvordan der skal testes for at kravet kan anses for opfyldt. Yderligere findes to kolonner med mulighed for at beskrive i hvilken fase funktionaliteten er indeholdt og et datofelt for succesfuld gennemførelse (så længe testresultatet er negativ indføres et NEJ).

Bilag

Brugertest af arbejdsganges sammenhæng med brugerinterfacet

Med udgangspunkt i beskrivelsen af brugerinterfacet og arbejdsbeskrivelsen fra kravspecifikationen brugertestes som et minimum rækkefølgen i de beskrevne arbejdsgange for at se, om der er sammenhæng i den ønskede funktionalitet.

Brugertesten er endvidere et glimrende forum for diskussion af tilføjelser, applikationsforbedringer eller reelle rettelser i den realiserede såvel som fremtidige funktionalitet. Disse opsamles og forsøges konstrueret som del af den efterfølgende slice.

Brugertest specifikation

Skadesregistrering	Fase	Test – ok
1. Åbner programmet til skadesregistrering ved dobbeltklik på ikonet.	Slice 1	2003.11.10
2. Startes applikationen i hovedvinduet hvor genstande og deres evt. sager ses og kan der navigeres til indtastningstilstanden for en ny genstand.	Slice 1	2003.11.10
3. Er typen som det eneste klar og kan der foretages et valg.	Slice 1	2003.11.10
4. Ledes brugeren automatisk til materialet og kan der foretages et valg.	Slice 1	2003.11.10
5. Fremkommer felterne til indtastning af obligatoriske og valgfrie genstandsstamdata og konkrete data i overensstemmelse med det første typevalg.	Slice 1	2003.11.10
6. Udløser korrekt indtastning af konkrete genstandsdata mulighed for indtastning af skader eller anden skadesbeskrivelse (mindst en er obligatorisk).	Slice 1	2003.11.10
7. Kan skadesregistreringsdelen annulleres med ”Annuller”. Kan skadesregistreringsdelen godkendes med ”Godkend”, og vises status i sagshistorikken.	Slice 1	2003.11.10
Kan skadesregistreringsdelen slettes med ”Slet” og slettes status fra sagshistorikken..	Slice 1	2003.11.10
Kan skadesregistreringsprocessen gentages ved at oprette en ny sag/genstand.	Slice 1	2003.11.10
8. Kan programmet lukkes ved at klikke i lukkeboks eller ved valg af luk.	Slice 1	2003.11.10
Brugerrefleksion: <ul style="list-style-type: none">• Felter der er obligatoriske bør afmærkes.• Mulighed for at påvirke valget af genstandsmaterialet på baggrund af typevalget.• Direkte kombinationsboksvalg af kort/tegnings størrelse - gerne med valgbeskrivelse direkte i kombinationsboksmenuen.• Bedre tilpassede labels i forbindelse med de enkelte genstandstyper.• Interfacet er lidt for stort til korrekt visning på AB skærm.• Indbinding ukendt tilføjes via den nye kategoritabel.		
Udviklerrefleksion: <ul style="list-style-type: none">• Overordnet virker alt efter hensigten, men der skal mere instruktion til i den indledende fase end forventet.• Der er enkelte steder tvivl om mulighederne – ændring i historik.		

Aktionsregistrering	Fase	Test – ok
1. Åbner programmet til skadesregistrering ved dobbeltklik på ikonet.	Slice 1	2003.11.10
2. Startes applikationen i hovedvinduet hvor genstande og deres evt. sager ses og kan der navigeres en udvalgt genstand/sag.	Slice 1	2003.11.10
3. Er det muligt at navigere mellem de endnu ikke aktionsregistrerede skadesregistreringer med eller kan de søges.	Slice 1	2003.11.10
4. Er klassificering klar til valg (obligatorisk).	Slice 1	2003.11.10
5. Er prioritet klar til valg (obligatorisk).	Slice 1	2003.11.10
6. Er der mulighed for valg af aktioner eller anden aktionsbeskrivelse (mindst en er obligatorisk).	Slice 1	2003.11.10
7. Kan aktionsregistreringsdelen annulleres med "Annuller".	Slice 1	2003.11.10
Kan aktionsregistreringsdelen godkendes med "Godkend", og vises status i sagshistorikken.	Slice 1	2003.11.10
Kan aktionsregistreringsdelen slettes med "Slet" og slettes status fra sagshistorikken.	Slice 1	2003.11.10
Kan aktionsregistreringsprocessen gentages ved at skifte til en ny sag eller genstand der allerede er skadesregistreret.	Slice 1	2003.11.10
8. Kan programmet lukkes ved at klikke i lukkeboks eller ved valg af luk.	Slice 1	2003.11.10
<p>Brugerrefleksion: Nemt!</p> <p>Udviklerrefleksion: Der er intet særligt at bemærke. Når først grundprincippet fra skadesregistreringen er lært er aktionsregistreringen stort set af samme type.</p>		

Bilag

Handlingsplan	Fase	Test - ok
<ol style="list-style-type: none">1. Åbner programmet til skadesregistrering ved dobbeltklik på ikonet.2. Startes applikationen i hovedvinduet hvor genstande og deres evt. sager ses og kan der let navigeres til handlingsplanvinduet.3. Er det muligt at vælge/søge genstande på baggrund af skades- og aktionsregistreringer og fremkommer der efterfølgende en valgliste eller andet, sådan at der ud fra søgningen kan vælges en eller flere genstande. Kontroller om en valgt aktiveret genstand kan bringes frem i skadesregistreringsvinduet ved at aktivere menupunktet "Vis i reg.". Kan man komme tilbage ved at aktivere "Vis i ordre".4. Kan preordredelen annulleres med "Annuller". Kan preordredelen godkendes med "Godkend", og vises status i ordrehistorikken. Kan preordredelen slettes med "Slet" og slettes status fra ordrehistorikken.5. Kan medarbejderen der skal forestå viderebehandlingen vælges og ændres.6. Kan uddybende instruktion til behandlingen indtastes, ændres og slettes i et notesfelt.7. Kan medarbejderen der skal forestå fremtagningen vælges og ændres.8. Kan uddybende instruktion til fremtagningen indtastes, ændres og slettes i et notesfelt.9. Kan medarbejderen der skal forestå transporten vælges og ændres.10. Kan uddybende instruktion til transporten indtastes, ændres og slettes i et notesfelt.11. Kan iværksatdelen annulleres med "Annuller". Kan iværksatdelen godkendes med "Godkend", og vises status i ordrehistorikken. Kan iværksatdelen slettes med "Slet" og slettes status fra ordrehistorikken. Kan der oprettes en ny handlingsplan efter godkendelse af en vilkårlig ordrestatus.12. Kan programmet lukkes ved at klikke i lukkeboks eller ved valg.		
Brugerrefleksion:		
Udviklerrefleksion:		

Dette er en gennemgang af vores overvejelser for en kodestandard. Resultatet af disse overvejelser kan ses i systemdokumentationen.

Kommentarer

Også den endelige kode vil indeholde kommentarer. Vi tilstræber at kategorisere kommentarerne for at fremme overblikket over implementeringen.

Navnestandard

I selve databasen forsøger vi at navngive sigende, så for eksempel tabeller kan skelnes fra forespørgsler i implementeringen.

Navnestandarden for VBA koden tager udgangspunkt i den Ungarske navnestandard (opkaldt efter dens Ungarske opfinder Charles Simonyi) [*Access 2002 VBA Handbook*].

Procedurer og funktioner

Generelt foretrækker vi at implementere ved funktioner frem for procedurer. I de mange tilfælde hvor MS Access vælger for os at der er tale om en procedure, forsøger vi at håndtere procedures afvikling så kontrolleret som muligt.

Den primære årsag til at vi helst vil implementere funktioner frem for procedurer er at det giver os mulighed for en tilstandskode ved funktionens afslutning. Dette er velkendt fra andre miljøer som for eksempel SQL eller JCL. Tilstandskoderne er mere detaljeret beskrevet i systemdokumentationen.

Parametre sendes til funktionen som værdioverførsler og returverdier modtages fra funktionen via referenceoverførsler.

Fejlhåndtering

Hvis den enkelte funktion eller procedure af en ekstern årsag fejler, for eksempel ved databaseadgang, har vi forsøgt at benytte VBA's (begrænsede) muligheder for fejlhåndtering, så funktionen kan afslutte på den pæneste mulige måde.

Validering af inddata

- Værdi valideres ved Exit af tekstfelt.
- Ved generel 'Godkend' valideres om nødvendige felter er udfyldt (har et indhold / en værdi).

Dette gøres typisk med metoden 'IsEmpty' i relevante moduler. Metoden (funktionen) returnerer en streng med feltnavne svarende til tekstfeltets tilknyttet labelfelts indhold uden præfikset "lbl", _erne eller lignende.

Sikkerhed spike

Bilag 19.5

Synopsis

Kort gennemgang af sikkerhed i relation til IT-systemet og dets realisering under MS Access.

Fysisk sikring

Lokaler, materiel, brand, tyveri mm. er i første række IT-afdelingens område og deres løfte om regelmæssige sikkerhedsbackup må anses som tilstrækkeligt.

Netværkssikkerhed

Den almindelige netværks sikkerhed i Københavns Kommunes forvaltninger er generelt meget høj efter udrulning af Windows 2000 medio 2003. Logon af godkendt hardware inddeling, brugergrupper og password sikring nedbringer risikoen for uautoriseret adgang.

Datasikkerhed ved multibrugerløsninger

Datasikkerheden i MS Access styres via en bred vifte af muligheder der repræsenterer forskellige niveauer af sikkerhed:

- Kryptering.
- Fjernelse af objekter i databasevinduet.
- Brug af opstartparametre.
- Adgangskode.
- Sikkerhed på brugerniveau.
- Sikring af VBA-kode (MDE).

Til dette projekt bør en kombination af flere metoder overvejes.

Backup og opdatering kommer også på tale under denne del.

Sikkerhed på brugerniveau

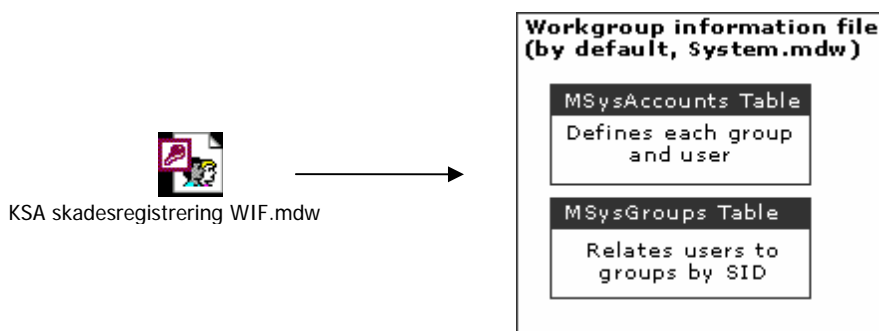
MS Access sikkerhed er ifølge Microsoft [*Running Microsoft Access 2000*, s. 928] ”state-of-the-art for personal desktop systems”, og uden at gå i rette med udsagnet skal vi blot være opmærksomme på at meget få programmer er helt sikre. Ved hjælp af adgangskoder og tilladelser, kan adgangen for enkeltpersoner eller grupper af enkeltpersoner styres. Sikkerhedskonti identificerer de brugere og grupper af brugere, der har adgangstilladelse til databasen. Disse oplysninger om ”arbejdsgruppen” er lagret i en arbejdsgruppeoplysningsfil.



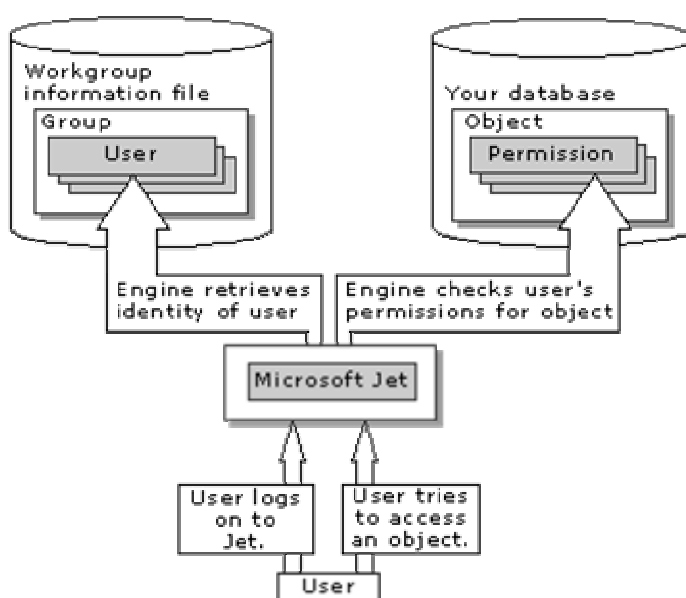
KSA skadesregistrering.mdb

Figur 19-7. Med hensyn til sikkerhed og arbejdsgrupper indeholder databasefilen information om brugernes aktuelle rettigheder og tilladelse til objekter.

Bilag



Figur 19-8. Workgroupfilen/arbejdsgruppefilen indeholder information om grupper og brugere inklusiv deres individuelle gruppe-medlemskaber og password.



Figur 19-9. Grafisk visning af sammenhænge mellem sikkerhedsdelen i databasen og i workgroupe filen.

Da brugerkontoen Administrator er nøjagtig den samme for hver kopi af MS Access, er de første trin i sikring af databasen at definere administrator- og ejerbrugerkonti (eller at anvende en enkelt brugerkonto som både administrator- og ejerkonto) og derefter fjerne brugerkontoen Administrator fra gruppen Administratorer. I modsat fald kan alle med en kopi af MS Access logge sig på arbejdsgruppen ved hjælp af kontoen Administrator og få alle tilladelser til arbejdsgruppens tabeller, forespørgsler, formularer, rapporter og makroer.

MDE

MS Access frontenden kan med fordel konverteres fra en .mdb til en .mde fil. Herved fjernes al redigerbar kode og der kan ikke foretages ændringer i databasens formularer, rapporter og moduler. Dette sikrer applikationen mod både tilsigtet og utilsigtede brugerændringer. Filen bliver som en sidegevinst mindre end den tilsvarende .mdb fil hvorved anvendelse af hukommelsen optimeres, og ydeevnen forbedres. Der opnås ingen datasikkerhed, men som distributionsfil er MDE en oplagt mulighed.

Runtime-versions restriktioner

Vær opmærksom på følgende når en MS Access applikation afvikles som runtimeversion:

- Databasevinduet, makroer, VBA miljøet, design view og filtermuligheder er skjult.
- Indbyggede værktøjslinier er ikke suporterede, men man kan fremstille sine egne.
- Kun ens egen "hjælp" er tilgængelig.
- En del Windows menuer og short cuts er udeladt.
- VBA error handling er påkrævet da fejl der ikke håndteres bringer programmet ned (undlad af samme grund makroer).
- Enkelte tastatur-kombinationer er ikke tilgængelige.

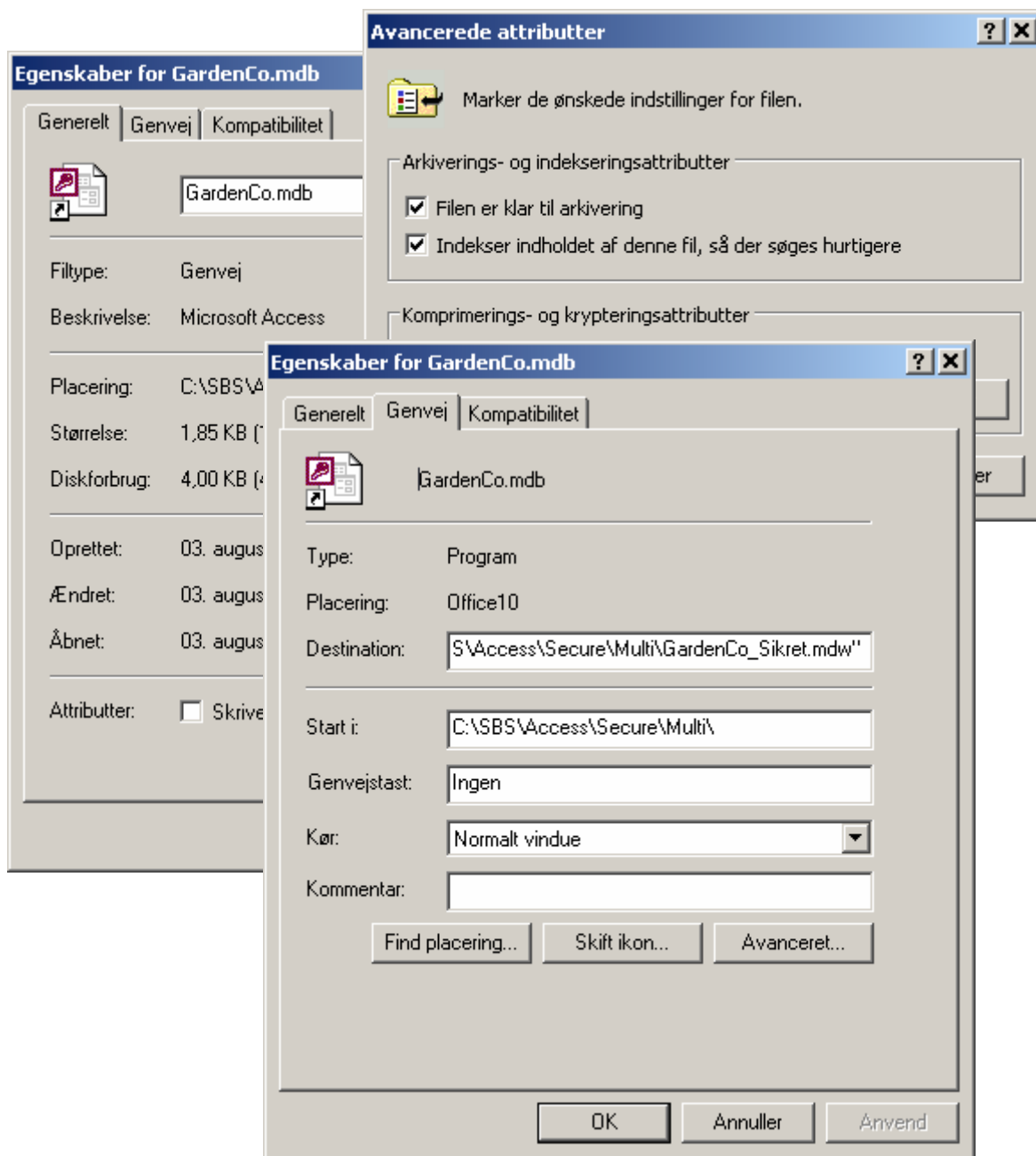
Dette kan måske gøre runtime versionen mere attraktiv end MDE hvis der kun er tale om at sikre mod utilsigtet brugerændringer, men de to kan også kombineres. Ydelsen er formentlig bedre end ved konvertering til MDE da der ikke automatisk anvendes kryptering.

Alle brugere kan bringes til at anvende runtimeversionen hvis den eksisterer på deres workstation ved at tilføje /RUNTIME som opstartparameter.

Bilag

Genvej der automatisk åbner den rigtige Workgroup

Denne opsætning åbner MS Access med den rigtige Workgroup fil, men ændrer ikke indstillingerne for fremtidig adgang til andre MS Access applikationer, hvad der er et krav for ikke at skabe brugerforvirring.



Figur 19-10. Destination evt tilføjet opstartparametre:

```
"C:\Programmer\Microsoft Office\Office10\MSACCESS.EXE"  
"C:\SBS\Access\Secure\Multi\GardenCo.mdb"  
/WRKGRP "C:\SBS\Access\Secure\Multi\GardenCo_Sikret.mdw"  
/USER "Trnka"  
/PWD "pw"  
/RUNTIME
```

Diskussion

Indledningsvist er det nødvendigt at se på vigtigheden af systemets data og krav til rigtighed og hastighed inden en sikkerhedsstrategi fastlægges endeligt.

I dette tilfælde er der tale om:

- Ikke personfølsomme data.
- Ikke data der betyder liv og død for os selv eller tredje part.
- Fejlagtige data kan ikke medføre skader eller krav om erstatning fra involverede eller tredje part.
- Systemet er et resursestyrings værktøj og hvis data ikke er helt præcise begrænser skaden sig til at reducere systemets nytteværdi.
- Der er ikke krav til performance (realtime o.lign.).
- Alle genstandsdata er offentlig tilgængelige og stiller ingen krav til netværkssikkerhed.
- Systemets workload er begrænset. Et optimistisk gæt er 500 "sager" om året, men allerede ved 100 er det en succes.

Fysisk sikkerhed og backup er IT-funktionens ansvar, så set i den ovennævnte kontekst kan sikkerhed for vores del begrænses til at omfatte:

- Forholdsregler der sikrer mod til- og utilsigtet ødelæggelse af IT-systemet.
- Styring af tilladelser og rettigheder til systemet.

For at sikre mod utilsigtet ødelæggelse af IT-systemet er MDE filer absolut en mulighed, men da ganske meget funktionalitet alligevel er aflåst fra brugere der afvikler applikationen via en runtime version er denne løsning at foretrække på alle lokale arbejdsstationer. Runtime versionen er i modsætning til MDE filen ikke krypteret, hvad der heller ikke er behov for.

For at sikre mod bevidst hærværk er det naturligvis vigtigt at alt befinder sig indenfor Kultur og Fritidsforvaltningens (KFF) beskyttede netværk og at IT-funktionen dagligt tager backup. Men alle ansatte i KSA har via nettet i princippet mulighed for at slette applikationen fra filserveren.

Styring af rettigheder og tilladelser via sikkerhed på brugerniveau er en fortræffelig løsning til at differentiere hvad brugergrupper og enkeltpersoner kan se og anvende i IT-systemet. Dette kan på samme tid sikre en opdeling af funktionaliteten og reducere kompleksiteten specielt for brugere på det laveste niveau. Vi forestiller os at almindelige brugere der har logget på deres lokale arbejdsstation og er underlagt KFFs IT-sikkerhedsregler, ved start af applikation får overført brugernavnet fra systemlogonet og kan åbne applikationen uden indtastning af password (default brugerpassword overføres automatisk med mindre den enkelte selv ønsker at anvende et dedikeret password). Arkivar, konservator og administrator taster et selvvalgt password ved start.

Multibruger spike

Synopsis

Da IT-projektet formodentligt er tænkt som en flerbrugerløsning prøver vi her at se på både hvilke muligheder MS Access indeholder for at dele/anvende den samlede databaseløsning af flere samtidige brugere og evt. opdele selve databasen så den samlede løsning bliver så fleksibel og veltilpasset som muligt.

Deling af databasen (multibruger)

Der er flere måder at dele databasen mellem flere brugere. Fælles er dog at databasen skal åbnes som delt (hvilket forringer ydeevnen i forhold til en udelt løsning, da en del resurser bruges til gensidigt at holde de åbne udgaver opdateret og sikre låsning eller i det mindste rapportere en evt. "samtidig" opdatering).

Der er således mulighed for at:

- **Dele hele MS Access-databasen**
Den letteste måde at dele data på er anbringe hele MS Access-databasen på en netværksserver eller i en delt mappe. Alle deler dataene og bruger de samme formularer, rapporter, forespørgsler, makroer og moduler. Denne strategi kan bruges, hvis alle skal bruge MS Access-databasen på samme måde, da man ellers ikke kan understøtte brugere, der opretter deres egne objekter.
- **Kun dele tabellerne i MS Access-databasen**
Det er også muligt at dele databasen sådan at tabellerne findes på netværksserveren og alle andre databaseobjekter kører på brugernes computere. I dette tilfælde er MS Access-databasens ydeevne hurtigere, da det kun er data, der sendes på tværs af netværket. Når man opdeler en database i en back-end (tabeller) og en front-end, kan brugerne tilpasse formularer, rapporter og andre objekter i deres front-end-databaser uden at påvirke andre brugere.
- **Dele MS Access-databaseobjekter eller dataadgangssider på Internettet**
Man kan endvidere skrive et eller flere databaseobjekter til statisk HTML- eller servergenereret HTML-format eller oprette dataadgangssider og derefter vise dem i en browser, f.eks. Microsoft Internet Explorer, på World Wide Web.
- **Replikere MS Access-databasen**
Hvis der anvendes flere computere, f.eks. en computer på kontoret og en bærbar computer, kan man bruge Microsoft Windows Rejsetaske til at oprette replikaer af MS Access-databasen og bevare disse replikaer synkroniseret. Flere brugere, der arbejder forskellige steder, kan også arbejde på deres egen kopi samtidigt og derefter synkronisere dem over netværket, enten via en Netværk via modem-forbindelse eller på Internettet.
- **Oprette et klient/server-program**
Ultimativt kan der arbejdes i et klient/server-miljø, hvorved ydeevnen og sikkerheden ved at oprette et klient/server-program forbedres betragteligt. Dataene gemmes i tabeller på en databaseserver, f.eks. Microsoft SQL Server, i stedet for i lokale tabeller i MS Access. MS Access-programmet (klienten) henter de data, der er brug for, på serveren. Serveren bevarer dataintegriteten og kører forespørgsler, den kan evaluere.

Bilag

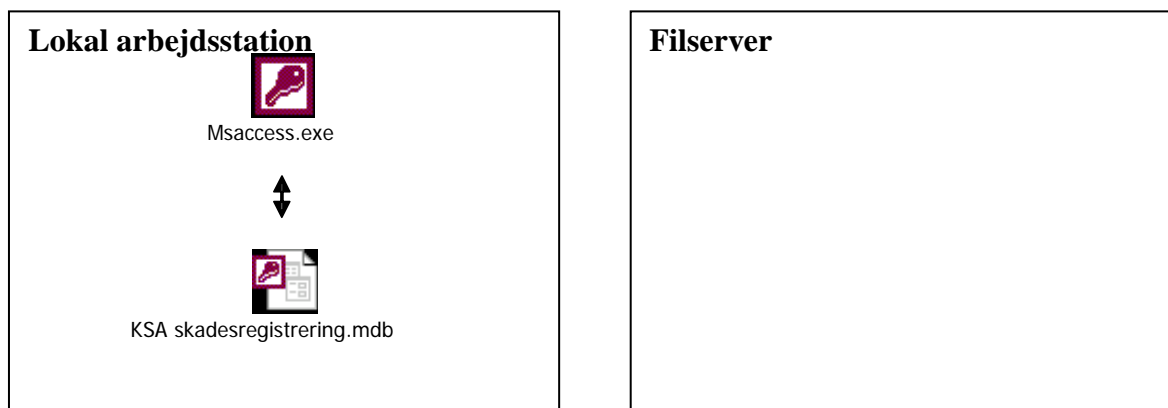
I et flerbrugermiljø kan mere end én person arbejde med de samme poster på samme tid. Da andre brugere kan ændre eller ligefrem slette de samme data, som man forsøger at redigere, kan der undertiden opstå en konflikt med andre, under arbejdet.

I MS Access arbejdes der med flere former for låsning hvis opløsningsniveau kan justeres i opsætningen. Man får hjælp til at holde styr på posternes status, mens de redigeres, og der bliver sørget for, at der hele tiden anvendes de seneste data. Når to eller flere brugere forsøger at redigere den samme post, vises der meddelelser, som hjælper med at løse konflikterne.

MS Access opsætningsmuligheder grafisk demonstreret

Efterfølgende demonstreres opsætningsmuligheder der indeholder kombinationer af de ovennævnte elementer til at opnå forskellige grader af fleksibilitet og sikkerhed. Eksempellisten gør langtfra krav på at være komplet, men medtager MS Access løsningsmodeller for hele spekteret.

Det hele på den lokale arbejdsstation

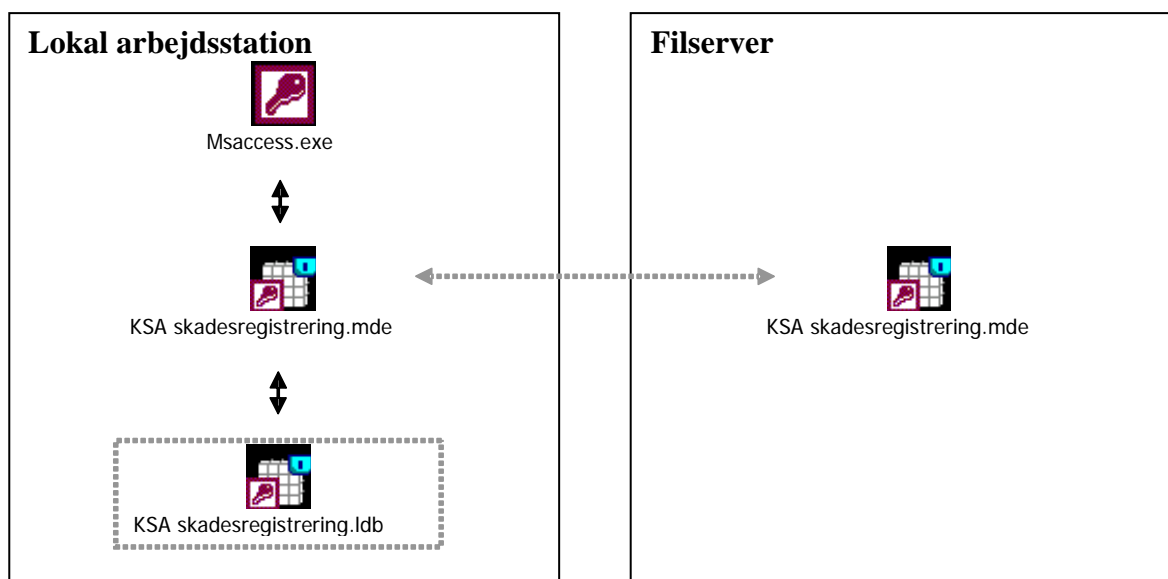


Figur 19-11. Udgangspunktet for mange mindre databaseløsninger. Enkeltbrugersystem hvor der kun er brug for enkelt (password) eller slet ingen sikkerhed og brugeren muligvis endog selv er udvikler. Der oprettes ikke en .ldb fil hvis basen åbnes som udelt.

Fordele: Nem og enkel løsning ved smådatabaser i enkeltmands eller mindre virksomheder.

Ulemper: Ikke en flerbruger løsning.

Det meste på den lokale arbejdsstation via datareplikering



Figur 19-12. I dette scenario er hele databasen placeret på filserveren og udnævnt til master. Herfra kan replikerede kopier distribueres ud på de lokale arbejdsstationer. MS Access starter databasen (her konverteret til en sikret og krypteret MDE fil) og opdateringer sker i første række på den lokale kopi. Denne synkroniseres senere med masteren og eventuelle konflikter med andre opdateringer løses ved hjælp af en "Replication Manager".

Fordele:

Databasen kan anvendes på lokale arbejdsstationer uden der er forbindelse til filserveren og er derved en mulighed for laptopbrugere og på lokationer med usikker LAN/WAN forbindelse. Decentrale databasekopier kan operere med bedre performance som stand alone og opdateres om natten til en fælles master hvorfra konflikter kan løses centralt og resultatet replikeres tilbage.

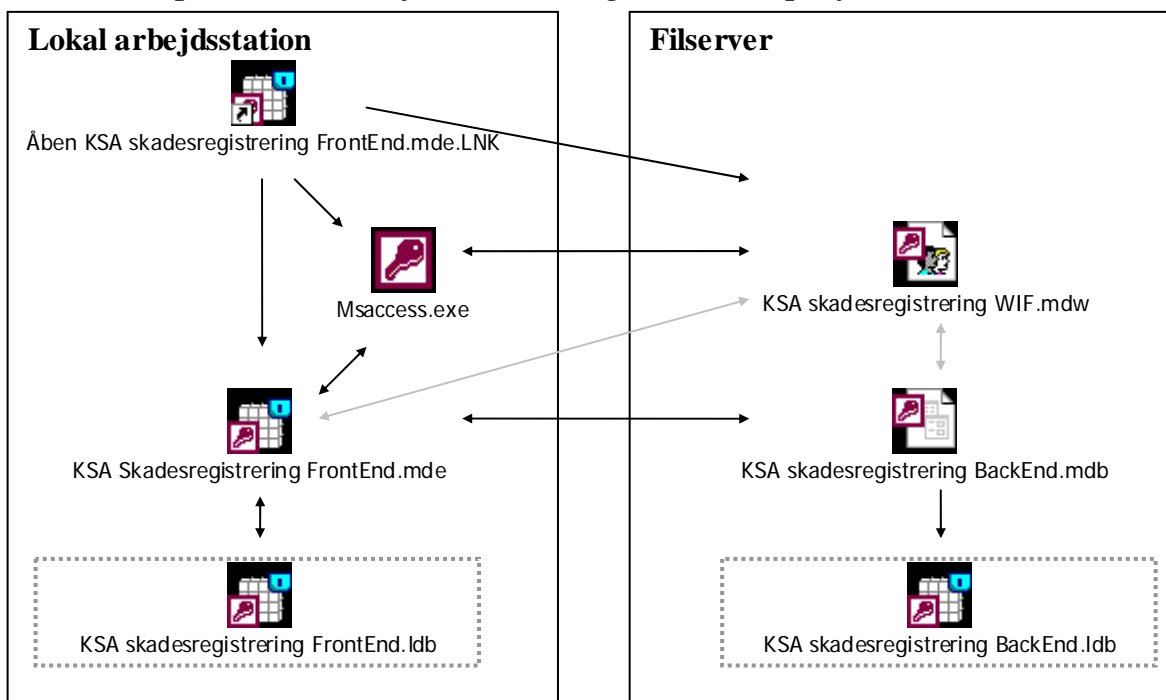
Ulemper:

Ved mange samtidige data opdateringer kan der blive et betydeligt udredningsarbejde for at bestemme hvilke data der har prioritet (dette kan dog til en vis grænse automatiseres). Mere kompliceret opsætning, sikkerhed og vedligehold.

Al data findes i kopi på hver lokal arbejdsstation der anvender datareplikering

Bilag

Frontend på lokal arbejdsstation og backend på filserver



Figur 19-13. Databaseløsningen er her opsplittet i en frontend på de lokale arbejdsstationer, og en backend på filserveren. Genvejen starter den lokale version af MS Access eller subsidiært en MS Access runtimeversion, frontenden (her konverteret til en sikret og krypteret MDE fil) og sikre at workgroupen på filserveren gøres til den aktive workgroup mens programmet afvikles. Workgroupen og backenden ligger på en server hvor de kan tilgås af alle potentielle brugere.

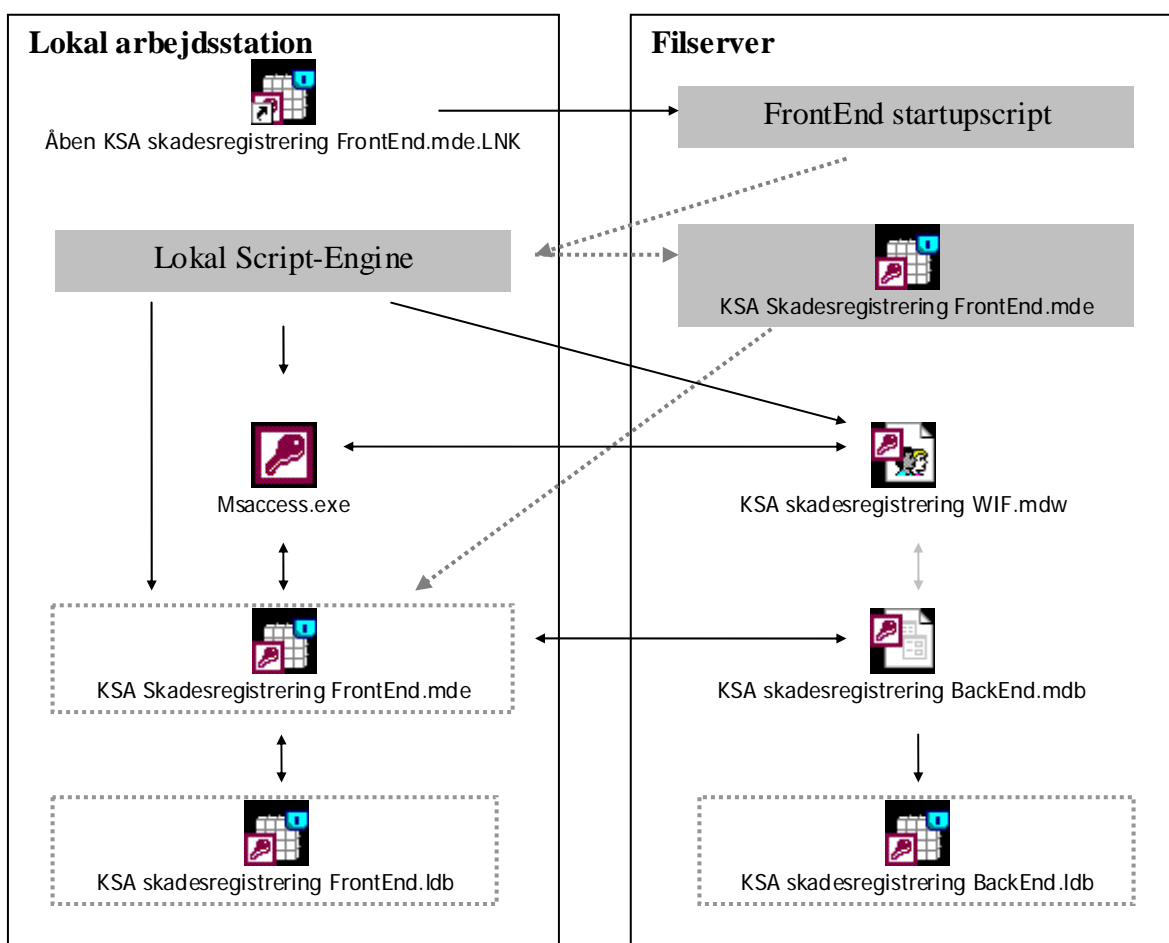
Fordele:

MDE-frontend skjuler kode, mindsker chancerne for utilsigtet at komme til at ødelægge databasen og er størrelsesmæssigt mindre. Delingen reducerer netværkstrafik og separerer data fra andre objekter i applikationen. Det bliver endvidere muligt at individualisere frontenden (forskellig funktionalitet) de forskellige lokaliteter, der alle gør brug af samme data. Workgroupen på filserveren muliggør sikkerhed på bruger niveau og differentiering af tilladelser til databasens objekter.

Ulemper:

MDE filen er krypteret hvilket har indvirkning på ydelse. Frontenden skal installeres og vedligeholdes på alle lokale arbejdsstationer.

Frontend via script og backend på filserver



Figur 19-14. Denne løsning er næsten identisk med den foregående med en lille tilføjelse. Genvejen på de lokale arbejdsstationer aktiverer et script der afvikles lokalt. Scriptet kopierer frontenden fra serveren og starter MS Access eller en runtimeversion på arbejdsstationen med den kopierede frontend og sikrer at workgroupefilen på serveren anvendes.

OBS der er ikke tale om datareplikering da tabellerne kun findes og kan tilgås på serveren.

Fordele i forhold til foregående løsning:

Frontenden skal ikke manuelt installeres lokalt og er altid nyeste version. Uforsættelig ændringer af frontenden overskrives ved næste åbning.

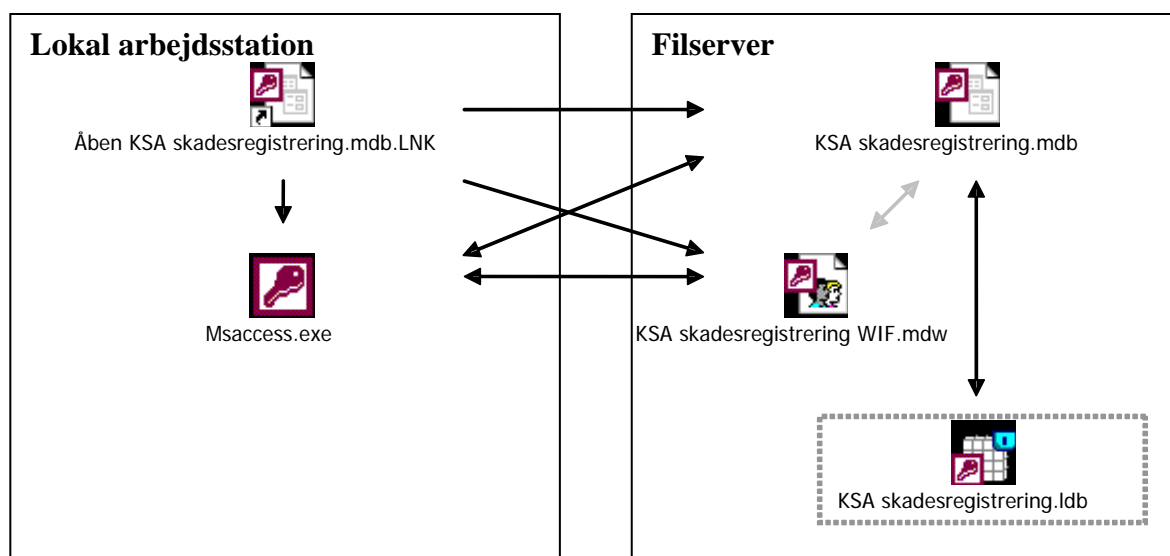
Simpel opdatering af frontenden da den gamle version blot substitueres med en opdateret version på filserveren og derefter automatisk downloades når programmet igen åbnes lokalt.

Ulemper i forhold til foregående løsning:

Forskellig funktionalitet på de forskellige lokale arbejdsstationer er kun mulig via tilladelser og grupper. Downloadtid under opstart. Scriptet er en ekstra komponent der skal udvikles og vedligeholdes om end den er ret begrænset i størrelse.

Bilag

Det hele på filserveren



Figur 19-15. Genvejen på de lokale arbejdsstationer åbner som tidligere MS Access eller en runtimeversion på klienten og sikrer at workgroupefilen på filserveren anvendes. Løsningen kan forbedres og gøres mere fleksibel ved at opsplitte databasen på filserveren.

Fordele: Den umiddelbart mest simple flerbrugerløsning. Der skal ikke installeres andet på de lokale arbejdsstationer end en genvej der er ens for alle.

Ulemper: Da hele applikationen skal afvikles over nettet er der både en tidpenalty såvel som ulemperne ved øget nettrafik. Hvis applikations "frontend" skades kan ingen bruge den berørte funktionalitet. Alle åbner den samme "frontend" hvorfor adgang til funktionaliteten skal styres via tilladelser.

OBS. Ved alle de ovennævnte multibrugereksempler sker databehandlingen på den lokale arbejdsstation og løsningen må ikke forveksles med en ægte klient/server løsning hvor stored procedures, views, triggers mm. afvikles direkte på serveren.

Diskussion

Da det fra start ligger klart at løsningen skal ligge inden for lokal arbejdsstation/filserver løsninger vil vi i første række pege på den simple model med alt på serveren fordi:

- Det er en simpel og tilstrækkelig løsning.
- Der er et minimalt vedligehold.
- Tabellinks kommer ikke ud af synk ved flytning.
- Alle er altid på nettet.
- Der rådes over en hurtig forbindelse på 100 Mbit.
- Applikationen afvikles via runtimeversionen på alle maskiner.
- Det er sjældent der er flere samtidige brugere (se bort fra låsning til en start).

Det er dog klart at en deling vil give fleksibilitet og placere statiske elementer på de lokale maskiner med en forbedret ydelse til følge. Netop denne del sammen med valget af kommunikation (DAO/ADO) må afvejes grundigt i designfasen og sammenholdes med merarbejdet samt KSA's mulighed for selv at vedligeholde og opdatere applikationen.

Arbejdsgangsforslag for udvikling til multibruger

- Opret eller konverter den ny database 2002 formatet (for at sikre mulighed for MDE mm.).
- Byg databasen med funktionalitet.
- Del evt. databasen for forbedret fleksibilitet
- Tag højde for begrænsning ved afvikling under runtimeversionen (indgangsformular, menulinie og hjælp).
- Opret kun MDE distribueringsfil hvis det skønnes nødvendigt med den ekstra sikkerhed eller det kan forbedre afviklingshastighed.
- Implementer sikkerhed tidligst muligt under hensyntagen til besværet i udviklingsfasen:
 - Fælles Workgroupe fil for både Front- og BackEnd.
 - Opsæt grupper og tildel tilladelser.

Kontroller ydelsen ved:

- Frontenden på klienten (evt. downloadet via script).
- Frontenden på serveren.

Microsoft Office XP Developer har tilsyneladende relevante hjælpeprogrammer (bl.a. Package and development Wizard) der bør undersøges nøjere før endelig implementering.

Hvad er ADO spike

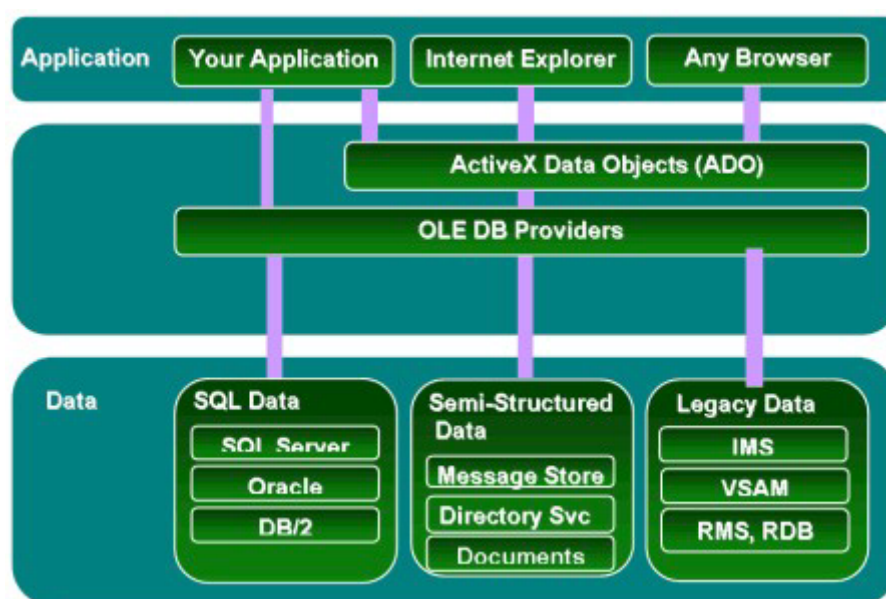
Bilag 19.7

Synopsis

ADO er en del af MDAC. MDAC er en samling af teknologier fra Microsoft som giver udviklere mulighed for at implementere UDA. UDA er i sig selv ikke en teknologi, snarere en strategi fra Microsoft for hvordan data skal til tilgås uafhængigt af programmeringsproget.

MDAC Arkitektur

ADO er en del af Microsofts MDAC arkitektur, der ud over ADO blandt andet indeholder ODBC, OLE DB og RDS.



Figur 19-16. Oversigt over MDAC arkitekturen

[MSDN Home > MSDN Library > Data Access > Microsoft Data Access Technical Articles Overview > MDAC Road Map]

ADO har adgang til data via en OLE DB provider.

Med Windows 2003 Server blev frigivet version 2.8. Når næste version af SQL Server bliver frigivet – sikkert sommeren 2004 – frigives også version 9.0 af MDAC.

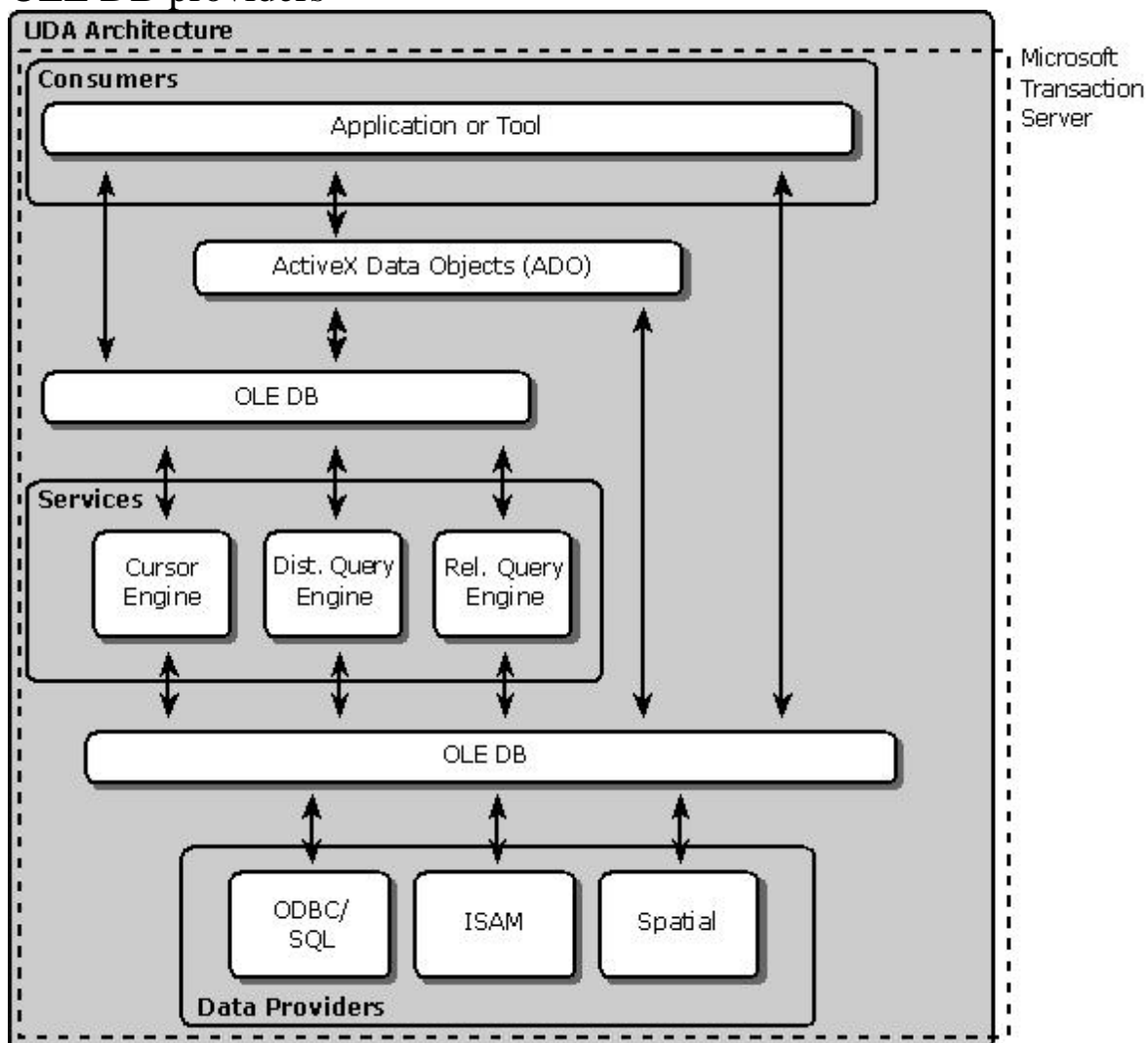
Jet

Microsoft Jet Database Engine er udviklet til MS Access og frigivet med version 1.1 af MS Access i 1993. Jet er standarden for databasehåndtering i MS Access og meget integreret i alle dele af MS Access. COM interfacet til Jet er DAO, som behandles senere.

Med version 2.6 indeholder MDAC ikke længere Jet komponenter, så version 4.0 af Jet er den sidste. Desuden vil der ikke blive udviklet udvidelser til Jet. Dog bliver der stadig udsendt kritiske sikkerhedsrettelser til Jet.

Bilag

OLE DB providers



Figur 19-17. Oversigt over hvilken placering OLE DB Providers har i det samlede hieraki. [MSDN Home > MSDN Library > Data Access > Microsoft OLE DB > OLE DB Providers]

Provideres til det enkelte databasesystem udvikles enten af Microsoft, som for eksempel til SQL Server, eller af leverandøren af databasesystemet, som for eksempel IBM DB2.

Common Object Model

Der findes to former for implementering af en COM-server:

- Out-of-process: Typisk en .exe-fil som kører i en anden proces end scriptet, for eksempel Word i Winword.exe. Dette gør det også muligt at gemme data ved hjælp af Data Access Objects (DAO), der giver adgang til data i MSAccess databaser.

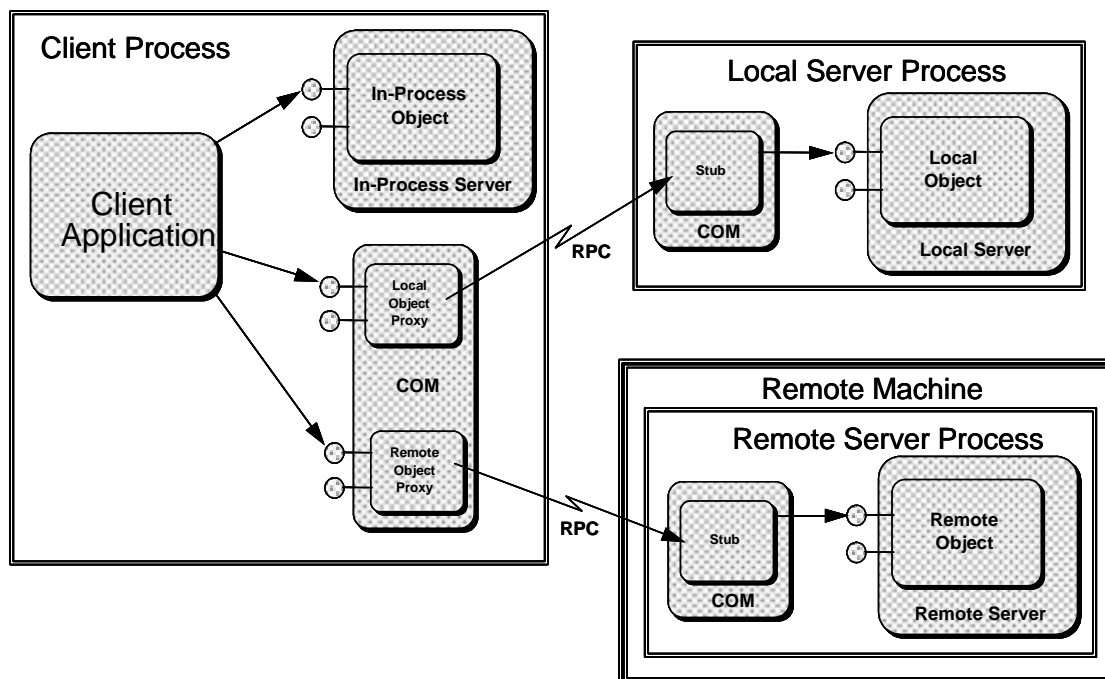
In-process: Typisk en .dll- eller .ocx-fil som kører i samme proces som scriptet (se

- Figur 19-18), for eksempel FileSystemObject i Scrrun.dll eller WshNetwork i wshom.ocx

Selve tilgangen sker som regel med et Intelligent Name, også kaldet en moniker, via interfacet IMoniker:

```
Dim objExcel
Set objExcel = CreateObject("Excel.Application")
```

På jævnt dansk oprettes der et midlertidigt objekt.



Figur 19-18. Klienter og objekter kalder altid In-process. Det underliggende gennemsigtlige tilgås (transparent) med RPC ved hjælp af COM interfaces [The Component Object Model Specification Version 0.9]

ADO kontra ODBC

Beskrivelse af ODBC

ODBC er et meget udbredt API til database adgang. Grundlaget er CLI som defineret af X/Open ("Data Management: SQL Call-Level Interface (CLI)") og ISO/IEC ("9075-3: 1995 (E) Call-Level Interface (SQL/CLI)") til database API med SQL som Database adgangs sproget. Udgangspunktet for ODBC er at give adgang til et databasesystems muligheder – ikke udvide mulighederne. Undtagen hvis det er for at implementere et minimum af SQL funktionalitet, som databasesystemet selv ikke har.

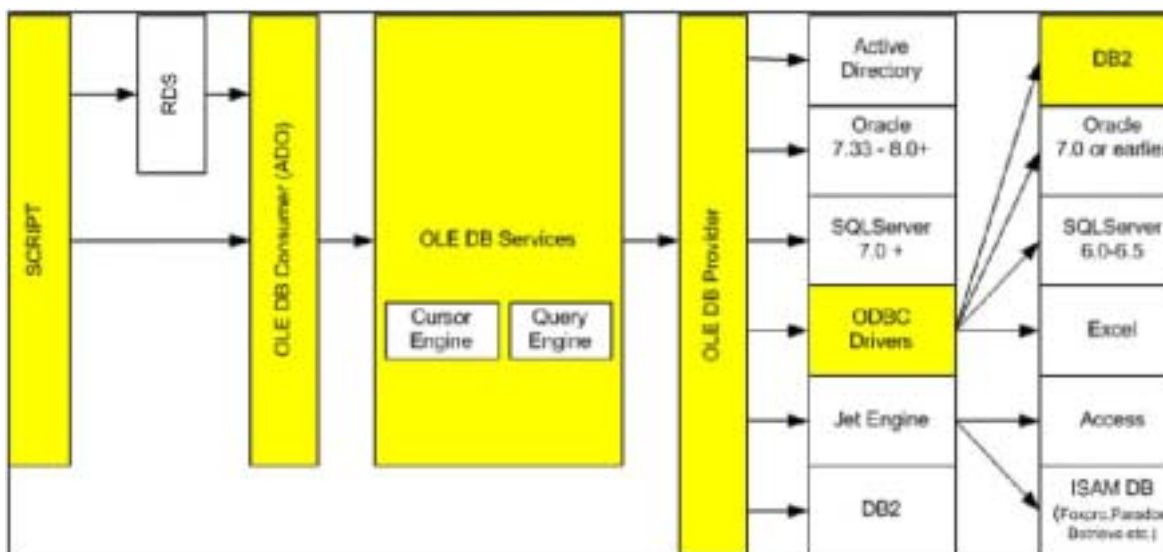
ODBC er af oprindelse et C interface til de relationelle data – kun. Dog vil ODBC blive udgivet til 64-bits Windows, men i en funktionelt uændret udgave – det vil sige at version 3.0 er den sidste fra Microsoft. ODBC findes ikke kun til Windows, men for eksempel også MacIntosh og forskellige UNIX-arter.

Historisk har ODBC været opfattet som en langsom forbindelse, men det passer ikke længere – i praksis er der ingen mærkbar forskel på ODBC og OLE DB hastighedsmæssigt.

Bilag

ADO i forhold til ODBC

Denne illustration fra IBM giver et meget godt billede af at der er flere veje fra til data gennem en blanding af providere.



Figur 19-19. Adgang til RDBM via enten "native" OLE DB driver eller ODBC.
[DB2 UDB Exploitation of the Windows Environment]

Modsat ODBC tilbyder ADO hændelser (events) direkte fra databasesystemet, hvilket er et udtryk for at ODBC er lidt ældre end ADO. Af samme årsag opfattes ADO generelt som fremtidens teknologi.

ADO kontra DAO

DAO er et rammeverk (framework) til at kode dannelse og manipulation af databaser med udgangspunkt i Microsoft Jet Database Engine. DAO tilbyder en hierarkisk objektsamling som benytter Microsofts Jet databasekerne til at tilgå data og databasestrukturer i

- Microsoft Jet databaser (.mdb).
- ODBC data kilder ved hjælp af en ODBC driver.
- Installérbare ISAM databaser som for eksempel Excel, Paradox eller dBase, som database kernen kan læse direkte.

DAO's objekt model er meget mere mangfoldig og har større funktionalitet end ADO's objekt model. Dette lyder umiddelbart besnærende, men i praksis er det overvældende. DAO er i version 3.6 kommet til vejs ende. DAO er inkluderet i Office XP, men vil ikke blive udgivet til 64-bits Windows.

Diskussion

ADO er fremtiden fra Microsoft, og hvis der er tale om samarbejde med en SQL Server installation er det måden at gøre tingene på i dag.

Desværre er ADO endnu ikke fuldt integreret i MS Access, så alle forbindelser skal kodes i VBA, mens DAO forbindelser kan ske grafisk med sammenkædede tabeller. Det ser ud som om Office og specielt MS Access står midt i et teknologiskifte; Jet/DAO med alle dens afledte teknologier er standard, men denne platform bliver ikke videreudviklet af Microsoft, og kun understøttet i begrænset omfang i fremtiden.

Med hensyn til ydelse er ADO en smule langsommere end ODBC og DAO, men med dagens teknologier og vores begrænsede system vurderes det i praksis til ikke at være noget problem.

DML via ADO spike

Synopsis

Hvordan håndteres DML via ADO? Hvilke opgaver og problemer er der ved at gå via ADO?

ADO

ADO er en samling af COM interfaces til data af forskellig art. Vores udgangspunkt er adgang til data i en Jet database, men ADO kan også give adgang til for eksempel filer, ADSI, web-data eller Exchange / Outlook data.

I dette tilfælde har vi nøjes med at arbejde på små og simple tabeller med få data. Vi har også begrænset brugergrænsefladen og fejlhåndteringen til noget nær et minimum ud fra at dette primært var en ADO/DML-spike.

DML

DML er klassisk defineret som SQL kald til en relationel database for at læse eller ændre indholdet. Udtrykkene er generelt

- INSERT
- SELECT
- UPDATE
- DELETE

Diskussion

Umiddelbart var det ret nemt at afvikle DML scripts via ADO, men der er nogle ting omkring flere brugere og opdatering, som skal kodes.

Transaktioner i MS Access spike

Bilag 19.9

Synopsis

I forbindelse med indsættelse af en skadesregistrering i databasen har vi fået et konkret behov for at indsætte og læse data i en given rækkefølge i én sammenhængende operation. Programmet til denne spike ligger i 'spike – Transaktioner i Access.mdb'.

Problematik

Helt konkret drejer det sig om når en ny skadesregistrering skal oprettes, for dér er andre data til systemet før og efter oprettelsen af selve skadesregistreringen i databasen:

- En gangstand med oplysninger om type og materiale oprettes i databasen.
- Denne genstand primærnøgle skal bruges til oprettelsen af en sag og en arkivalie eller lignende i databasen.
- Sagens primære nøgle bruges til at oprette en skadesregistrering og en historik med status.

Dette giver mindst fem trin, som vi gerne vil samle i én transaktion, så hvis for eksempel oprettelsen af historik fejler vil der ikke være ufuldstændige data i databasen:

- Opret en komplet genstand i databasen.
- Læs den oprettede genstands primære nøgle fra databasen.
- Opret en komplet sag og en komplet arkivalie eller lignende i databasen. Dette punkt kan måske kun gennemføres i to separate trin i MS Access.
- Læs den oprettede sags primære nøgle fra databasen.
- Opret en komplet skadesregistrering og en komplet historik i databasen. Dette punkt kan måske kun gennemføres i to separate trin i MS Access.

SQL transaktion i MS Access

Med Jet version 4.x kom muligheden for (primitive) transaktioner.

Statement	Handling
BEGIN TRANSACTION	Starter transaktion
COMMIT [TRANSACTION WORK]	Afsender transaktionen og skriver data i de fysiske tabeller
ROLLBACK [TRANSACTION WORK]	Annulerer transaktionen

Figur 19-20. En transaktion kan for eksempel bygges op som et DAO recordset, men det er desværre ikke muligt at lægge en Jet-transaktion ind i en MS Access forespørgsel.

Bilag

DAO transaktion i MS Access

Metode	Handling
<arbejdsområde>.BeginTrans	Starter transaktion
<arbejdsområde>.CommitTrans	Afsender transaktionen og skriver opdatering til de permanente databasers objekter
<arbejdsområde>.Rollback	Annulerer transaktionen

*Figur 19-21. <arbejdsområde> er typisk det almindelige sat med kommandoen:
Set wrkCurrent = DBEngine.Workspaces(0)*

Diskussion

Det er ikke umiddelbart muligt af lave en transaktion med flere commits i Jet/DAO så delresultater kan anvendes inde i transaktionen selv.

Regulære Udtryk spike

Bilag 19.10

Synopsis

I forbindelse med validering af inddata vil vi gerne kunne validere data så nøjagtige som muligt, for eksempel en e-mail adresse.

De muligheder der er med valideringsregler og inputmasker i MS Access synes vi ikke altid er nøjagtige nok. Derfor har vi set nærmere på Regulære Udtryk, som er kommet ind i Windows miljøet med Windows Scripting Host (WSH) version 5.

Denne spike skal kun afklare om og hvordan Regulære Udtryk kan formuleres i MS Access VBA.

Regulære Udtryk COM

MS Access får adgang til COM modulet RegExp via ActiveX referencen "Microsoft VBScript Regular Expressions 5.5" som er implementeret i filen "vbscript.dll\3".

Programmet til denne spike er placeret i formularen "frmSpike-RegulæreUdtryk", men de centrale linjer er gengivet her:

Da COM modulet ikke er en del af en normal MS Access installation, men forudsætter WSH version 5, bliver der til en start forsøgt at oprette en reference, hvis referencen allerede findes er alt jo OK, men hvis der fejler for eksempel hvis der ikke findes WSH version 5 bliver det så konstateret

```
References.AddFromFile ("vbscript.dll\3")
```

Når formularen lukkes forsøger programmet at rydde op ved at fjerne referencen

```
References.Remove (References!REFREGEXP)
```

Dette fejler dog med fejlnummer 9 (Subscript out of range). Det ligger uden for denne spike at rette fejlen, og den er heller ikke afgørende, så vi lader det ligge til en senere lejlighed...

Mailadressen inddeles først i understreng (substrings) ved hjælp af udtrykket

```
objRegExp.Pattern = "(.+)@(.+)\.([a-z]{2,3})$"
objRegExp.IgnoreCase = True
```

Udtrykket betyder (bagfra)

- Er der tale om en topdomænenavn på to til tre alfabetiske og ikke diakritiske karakterer? Dette er den tredje understreng.
- Kommer der et punktum (dot)
- Kommer der mindst ét tegn, som ikke er et linjeskift? Dette er den anden understreng.
- Er der et snabel-a (at)
- Kommer der mindst ét tegn, som ikke er et linjeskift? Dette er den første understreng.

Kan der svares nej til ét af punkterne er svaret generelt nej (False)

Dernæst kontrolleres hver understreng yderligere. Brugeren før og subdomænet efter '@' et kontrolleres i samme rutine ved hjælp af udtrykket

```
objRegExp.Pattern = "[a-z0-9]+"
objRegExp.IgnoreCase = True
```

Dette udtryks muligheder er dog ret begrænsede, da det kun ser om er mindst ét tegn intervallerne a-z sammen med 0-9. Det vil sige at hvis der er et ulovligt tegn som for eksempel "£" inder mellem nogle lovlige tegn er udtrykket stadig sandt (True). Dette er klart et punkt til videre bearbejdelse...

Topdomænet kontrolleres yderligere ved hjælp af udtrykket

```
objRegExp.Pattern = "com|dk|edu|org"
```

Bilag

```
objRegExp.IgnoreCase = True
```

Udtrykket er en liste over godkendte domænenavne, det vil sige at det for eksempel ikke er muligt at sende en mail til Norge (no).

Diskussion

Det lykkedes at fremstille et fungerende og anvendeligt Regulært Udtryk i VBA, så vi tror på at det kan anvendes til en mere nøjagtig validering af inddata.

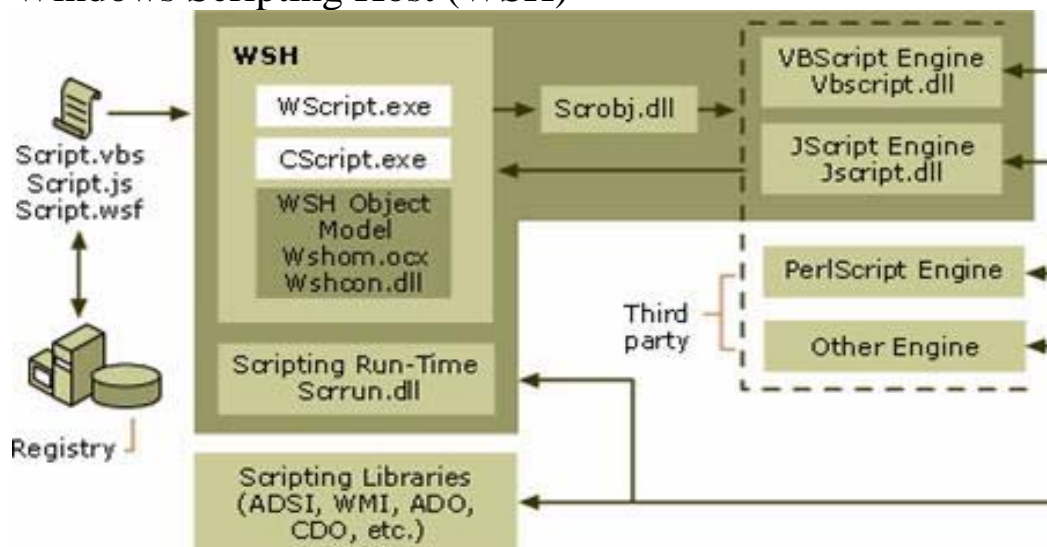
Start script spike

Bilag 19.11

Synopsis

Afhængig af den ønskede funktionalitet og dennes kompleksitet er der flere muligheder for at afvikle et script til start af en applikation.

Windows Scripting Host (WSH)



Figur 19-22. Oversigt over WSH miljøet [Microsoft Windows 2000 Scripting Guide]

Et WSH script kan have en af flere filtyper. Afhængigt af hvilken oversætter og hvilket miljø man ønsker at afvikle scriptet i.

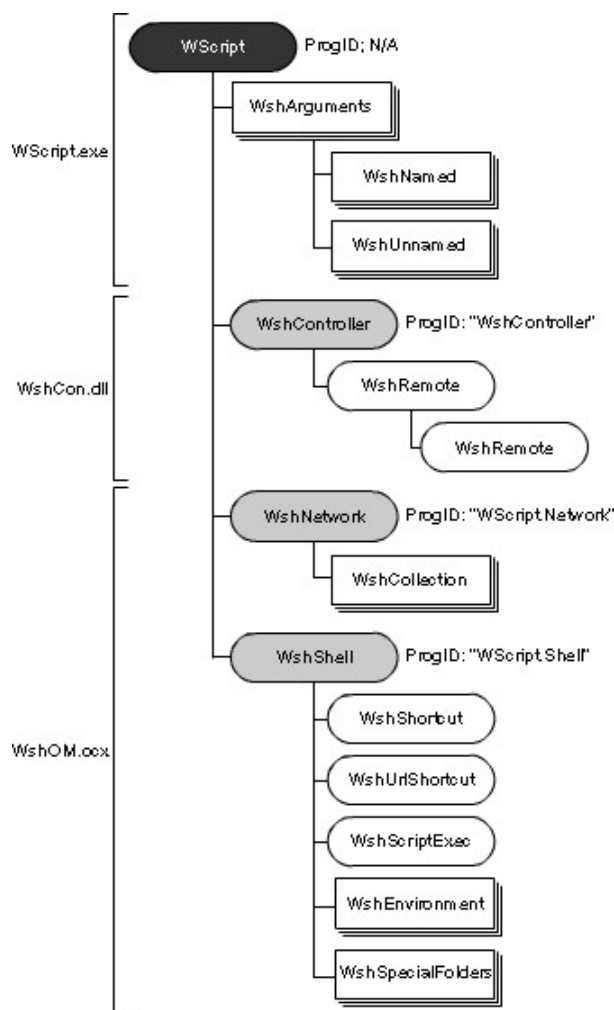
- .vbs = enkelt VBScript script uden import.
- .js = enkelt Jscript eller JavaScript (EMEA-script) script uden import.
- .wsh = egenskaber for en script fil.
- .wsf = Windows Script File. Script project formateret med XML med mulighed for import, flere script engines, type libraries og flere jobs i samme fil.
- .inc = import-fil (include).
- .wsc = Windows Script Component. COM-komponent skrevet i XML-format.
- .tlb = Script Component Type Library. Indeholde oplysninger om et script komponent interfaces og medlemmer (members). Et Type library er oftest nødvendigt for at kunne tilføje hændelser (events) til et script komponent.
- .hta = HTML Applikation. Script formateret med XML til kørsel med IE men i en anden proces (Mshta.exe) som ligger uden for IE's sikkerhed. Det giver nogle muligheder, for eksempel at afvikle WMI kald uden at blive spurgt om det er i orden.

Hvis ikke andet er nævnt afvikles scriptet som regel med én af disse to oversættere:

- CScript.exe = afvikling af script i kommandolinje-miljø.
- Wscript.exe = afvikling af script i Windows-miljø.

Objekt modellen for selve WSH er ganske kort:

Bilag



Figur 19-23. Hvert af de tre objekter *WshController*, *WshNetwork* og *WshShell* har deres egen objekt model med objekter, metoder, attributter (properties) og hændelser (events). [Microsoft Windows 2000 Scripting Guide]

Når en applikation skal startes fra WSH kan de ske med metoden 'Exec' fra *WshShell*, hvilket giver mulighed for at samle meldinger fra applikationen op med *StdIn*, *StdOut* og *StdErr*, da applikationen kører i en child proces.

```

Set objExec = objShell.Exec(strAccessProgramfil & _
    " " & Chr(34) & strFrontendfil & Chr(34))
Do Until objExec.StdOut.AtEndOfStream
    document.writeln objExec.StdOut.ReadLine()
Loop
  
```

```

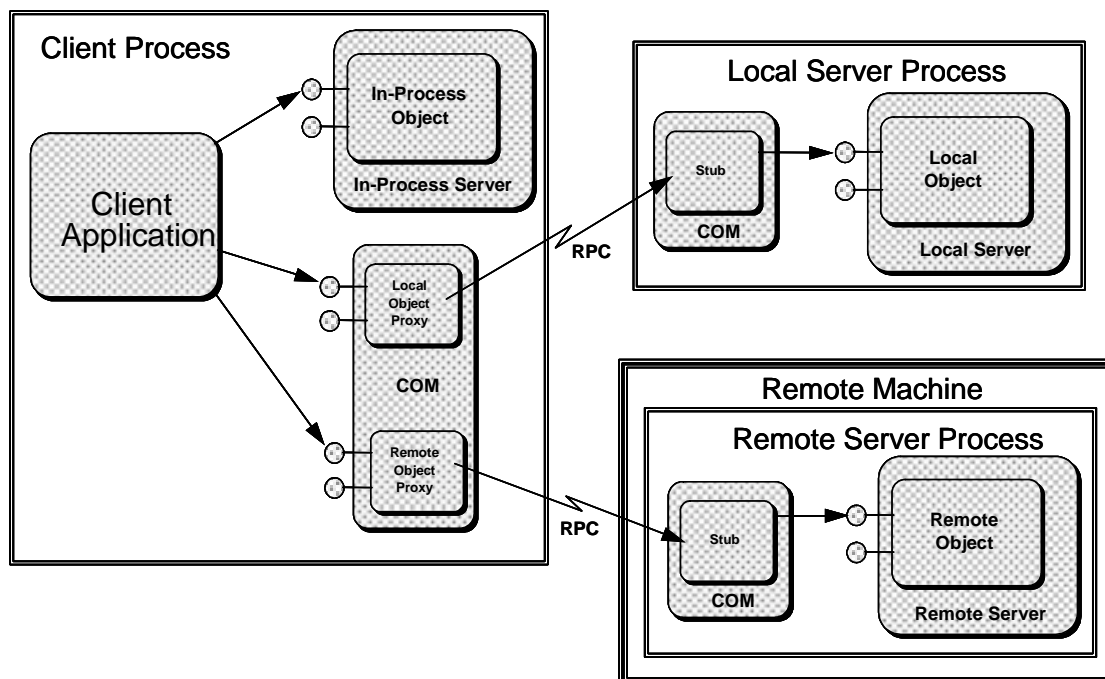
intFejlkode = objExec.ExitCode
  
```

En mere simpel måde er at starte applikationen med 'Run' metoden fra *WshShell*, hvor applikationen startes i en separat proces uden mulighed for at læse meldinger.

```

Const KSAPGM = "J:\Klient\Prod\Pgm\KSA.mdb "
Dim objShell
Set objShell = WScript.CreateObject("WScript.Shell")
objShell.Run KSAPGM, 3
  
```

Det er muligt at anvende COM objekter fra WSH. Dette sker som regel med funktionen 'CreateObject'. Dette benyttes også internt af WSH til at binde de enkelte superklasser sammen, for eksempel *Wscript* med *WshShell*.



Figur 19-24. COM sørger for forbindelsen mellem de enkelte objekter [The Component Object Model Specification]

HTML Applikation

Ved at afvikle et script skrevet i for eksempel VBScript eller Jscript som en HTA, gives der fuld adgang til det lokale filsystem samt netværksressurser med brugerens rettigheder. Desuden er der mulighed for at formatere meddelelser løbende til brugeren ved hjælp af DOM.

Starten på et HTA scripts kode kan for eksempel se sådan ud:

```
<?xml version = "1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" _
    "http://www.w3.org/TR/xhtml/DTD/xhtml-strict.dtd">

<!-- Skadesregistrering.hta -->
<!-- 2003, Martin Trnka og Niels Grove-Rasmussen -->

<!-- Installerer og starter Skadesregistreringssystem -->

<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
    <title>KSA Skadesregistrering</title>
    <HTA:Application
        ID = "objSkadesregistrering"
        ApplicationName = "Skadesregistrering"
        Scroll = "no"
        SingleInstance = "yes"
        WindowsState = "normal"
    />
    <link rel = "stylesheet" type = "text/css" _
        href = "./Skadesregistrering.css" />
    <script type = "text/VBScript">
        Option Explicit
```

...

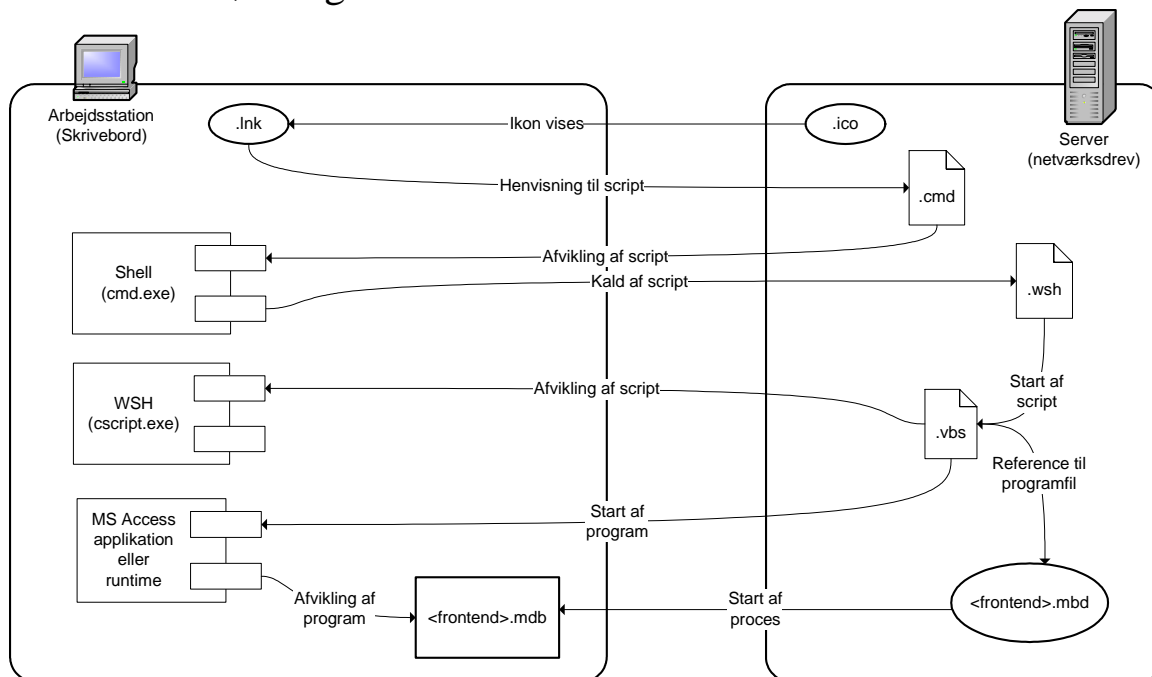
Bilag

Windows Shell

Til og med Windows Me var kommandolinje-miljøet `command.com`, hvor man enten afvikler kommandoer og programmer direkte eller i `.bat`-filer. Med Windows NT 3.51 kom muligheden for at afvikle `.bat`-filer og `command.com` via Virtual Dos Machine (VDM) eller afvikle `.cmd`-scripts direkte til Windows NT ved hjælp af `cmd.exe` – der er fuldt kompatibel bagud med `command.com`. Til og med Windows NT 4.0 indeholdt standardinstallationen en komplet filsamling fra MS-DOS 6.22. Som regel blev systemet sat op til at afvikle `.bat`-filer med Windows miljøvariablen `%ComSpec%` sat til `"C:\WINNT\System32\cmd.exe"` på grund af mere stabil afvikling af scripts og højere hastighed.

Med Windows 2000 og den nye Windows NT 5.0 kerne forsvandt `command.com` og antallet af kommandoer i Windows Shell blev udvidet markant. Men stadig er der meget langt op til mulighederne i WSH – på den anden side har mange Windows-miljøer mange Shell scripts, der på fuldt tilfredsstillende vis løser konkrete opgaver. Disse Shell scripts kan stadig benyttes, også med WSH ved hjælp af `WshShell`-metoden `'Exec'`.

Kombineret løsning



Figur 19-25. Oversigt over et kombineret startscript

Der er flere fordele ved at kombinere Windows forskellige scriptmiljøer. Dette bliver forhåbentligt klart ved at beskrive figuren på foregående side:

Genvejen (.lnk) på brugerens Skrivebord med ikonet (.ico) på netværksdrevet kalder Shell scriptet (.cmd) minimeret. Scriptet afvikles lokalt med `cmd.exe` i en separat proces og starter WSH scriptet (.vbs) ved at kalde egenskabsfilen (.wsh).

I egenskabsfilen (.wsh) sættes WSH scriptet blandt andet til at starte usynligt, så det eneste brugeren ser, er Shell scriptet (.cmd) i proceslinjen med vores ikon. Når WSH scriptet er startet slutter Shell scriptet.

Denne omvej sker primært for at kunne vise en valgt ikon på brugerens Skrivebord, men den giver også mulighed for at vælge et andet striptmiljø fra en tredjeparts leverandør som for eksempel IBM.

WSH scriptet indeholder den egentlige startfunktionalitet, da det finder ud af om der findes en MS Access installation på arbejdsstationen. Afhængigt af dette startes programmet (.mdb) i MS Access applikationen eller runtime. Når programmet er startet slutter WSH scriptet.

Diskussion

En kombineret løsning giver flest muligheder, men umiddelbart kan den hurtigste løsning laves som et Windows Shell script. Det lykkedes både at lave et startscript med acceptabel funktionalitet i HTA med ekstern formattering i CSS samt en kombineret løsning.

Mail fra MS Access spike

Bilag 19.12

Synopsis

Det er et af vores store ønsker til dette system at vi kan sende en mail fra systemet til en specifik person. Allerhelst vil vi gerne lægge en interaktion mellem modtageren og systemet i en mail, for eksempel ved hjælp af en trykknop.

I den sammenhæng kigger vi hér på forskellige teknikker og løsninger for at se om vores forhåbninger kan opfyldes.

Motivation

I vores rige billede er angivet flere steder at nogle aktører skal have en besked. Dette kunne vi godt tænke os kunne ske med mail. I den sammenhæng arbejder vi med to forskellige slags mail besked:

- *Besked.* Der sendes en besked om for eksempel en bestilling af transport. Hvis der skal registreres noget om besked bliver det gjort når beskeden sendes fra systemet.
- *Notifikation.* En aktør med arkivarrolle får besked om udestående opgaver/aktiviteter. Det er vores ønske at der i denne besked er funktionalitet til direkte at håndtere de i beskeden nævnte udestående aktiviteter.

Notes mailklient

Notes' COM interface (provider) tilgås via OLE Automation, hvilket blandt andet medfører at Notes skal være installeret i miljøet, og at adgang gives med sen binding (late bind). Dette er ikke noget problem for vores projekt, da arbejdsstationerne i Københavns Kommune har installeret Notes 5.0.10 og vores applikation afvikles som noget der kan kaldes et script oversat til pseudokode (VBA).

For at mail-funktionaliteten virker fra VBA, skal en lokalt installeret Notes være startet og brugeren skal være logget på Domino (mail gateway etcetera). Det vil sige at afsendelsen ellers vil fejle, også hvis brugeren har en Notes session startet i for eksempel en Citrix klient.

Det skal bemærkes at programmet til denne spike er skrevet og afviklet i MS Access 2000 installeret på Windows NT 4.0 Workstation og Windows XP Professional.

Interaktion mellem modtager og system

Disse muligheder vil idag typisk blive implementeret i LotusScript, men med version 6 af Notes er LotusScript kommet til vejs ende, og IBM satser fuldt og helt på Java som miljøet til udvikling af funktionalitet i Domino/Notes. Små ting vil skulle skrives i JavaScript, der allerede nu er rimeligt implementeret i Notes, dog er der lidt til at DOM er fuldt implementeret, men det kommer (også) med version 7.

Vi kunne egentlig godt allerede nu implementere interaktionen i Java, men det vil tage megen tid – og så er Københavns Kommune ved at køre Outlook i stilling som Notes' afløser, og Outlook vil hellere tale VBA end Java. Det ser ud som om en Java implementering vil være spild af tid.

Bilag

Klassen 'Button'

Klassen 'Button' kan ikke anvendes via COM – det er én af forskellene mellem COM og LotusScript. Det var vores håb at vi kunne lægge en SQL forespørgsel i sådan en knap sammen med en ADO tilgang til vores database, men det kan ikke lade sig gøre p.t. Det vil sige at en notifikation kun er envejs.

Agent

En anden mulighed er at lave en agent som udsendes til de relevante brugere. Agenten kunne for eksempel reagere på en bestemt emnestreng, som så startede en session med modtageren hvor forskellige muligheder vælges og endelig opdateres systemets database. Agenten distribueres ved hjælp af en Notes-mail, hvor koden til agenten lægges i en trykknop, så brugeren selv kan bestemme om agenten skal installeres.

Er Notes startet?

For at håndtere de forskellige Notes situationer ville vi gerne kunne lade vores system selv undersøge om Notes er startet lokalt. Desværre indeholder VBA ikke funktionalitet til selv at kunne undersøge software på en Windows arbejdsstation, så derfor er det nødvendigt at benytte WMI til at få de ønskede oplysninger om kørende processer og deres tilstand.

CDO mailklient

CDO er implementeret i flere forskellige versioner til forskellige installationer:

- *Active Messaging* (olemsg32.dll): Dette bibliotek installeres med Exchange 5.0.
- *CDO 1.2.1* (cdo.dll): Biblioteket installeres med Exchange 5.5, Outlook 98 og 2000 og Exchange 2000 Server.
- *CDO 1.2 for Windows NT Server* (cdonts.dll): Dette bibliotek installeres med Internet Information Server 4.0 og Windows 2000 Server.
- *CDO for Windows 2000* (cdosys.dll): Dette bibliotek installeres kun med Windows 2000 Server. Denne udgave af CDO kaldes også *CDO 2.0*.
- *CDO for Exchange 2000 Server* (cdoex.dll): Dette bibliotek installeres med Exchange 2000 Server.
- *CDO for Exchange Management* (cdoexm.dll): Dette bibliotek installeres med Exchange 2000 Server og benyttes til at administrere postboks og servere. I daglig tale kaldes denne udgave af CDO for *CDOEX*.
- *CDO Workflow Objects for Microsoft Exchange* (cdowf.dll): Dette bibliotek installeres med Exchange 2000 Server og benyttes til at lave workflow applikationer.

CDO 2.0 forudsætter CDO 1.2 for Windows NT Server, der er implementeret ved cdonts.dll. CDO 2.0 baserer sig på SMTP og NNTP, men objekt modellen er udvidet til også at kunne håndtere MIME beskeder. CDO 2.0 er ud over NNTP understøttelsen kendetegnet ved at kunne håndtere protokol hændelser. Dette giver muligheder for at kunne håndtere ind- og udgående mail som for eksempel en virusscanner.

Det ser ud som om at cdonts.dll og cdosys.dll ved en senere service pack (3?) til Windows 2000 er lagt ud på Windows 2000 Professional. Derfor vil denne spise forsøge at implementere så meget funktionalitet som muligt. Dog er der spørgsmål, som sikkert kun kan besvares ved en test på en normal arbejdsstation på KSA. For eksempel: Findes filerne – eller bare nogle af dem – på en normal arbejdsstation på KSA?

Outlook mailklient

For at få adgang til funktionaliteten i Outlook benyttes COM objektet mapi32.dll. Dette COM objekt kommunikerer med COM objektet cdo.dll, som klient applikationen (vores) så kommunikerer med.

SMTP

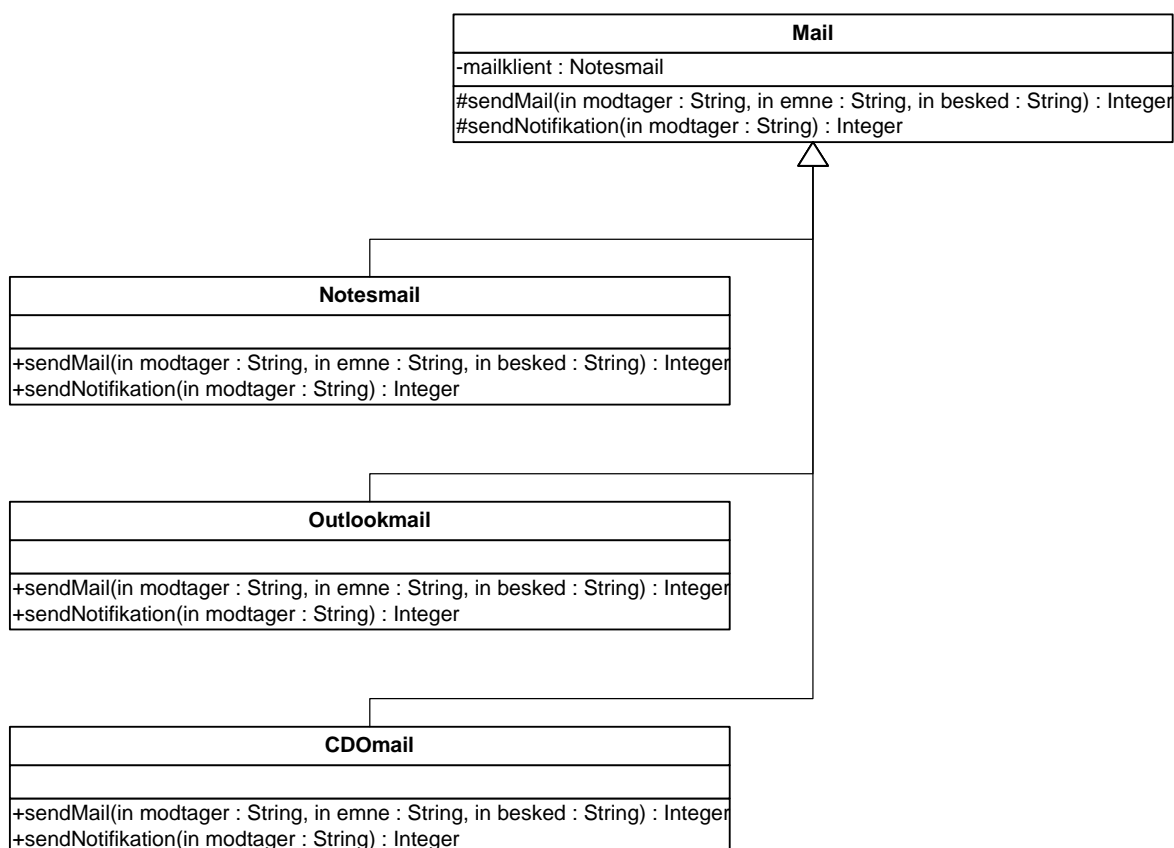
SMTP kan kun anvendes fra Office (MS Access) ved hjælp af CDO, som sørger for at SMTP kører lokalt. Eller hvis der er installeret en anden SMTP klient lokalt som for eksempel Notes.

Vi har ikke mulighed for at installere på arbejdsstationerne, så der bliver ikke arbejdet videre på denne del af spiken.

Diskussion

Det er lykkedes at lave en næsten generisk løsning til afsendelse af mail fra Notes. Med hensyn til distribution af den ønskede funktionalitet fra vores program er det blevet anbefalet at lave en agent eller se på QueryOpen i Notes.

Løsningen kan nemt udvides til også at kunne håndtere mail fra Outlook. Løsningen er skitseret hér.



Figur 19-26. Hvis/når Københavns Kommune går over til Outlook behøver administratoren blot at ændre typen på attributten 'mailklient' fra 'Notesmail' til 'Outlookmail'.

Af tidsmæssige er kun subclasses 'Notesmail' implementeret i denne spike.

Bilag

Det virker som om CDO er afhængig af at tingene sker i et Outlook / Exchange miljø, hvilket Københavns Kommune ikke har i projektperioden. Derfor vil der ikke blive arbejdet videre med denne løsning.

Denne spike indeholder ikke andre beskedmuligheder, selv om de er med i en Windows installation. Det vil sige at der for eksempel ikke er arbejdet med Outlook Express og Windows Messenger.

Hjælp spike

Bilag 19.13

Synopsis

Vi vil gerne lægge brugerdokumentationen ind i systemet (applikationen), og det skal gøre på en måde der bedst muligt udnytter mulighederne i Office. Det vil sige at vi gerne vil lave vores egen lille lilla bog – måske flere.

Værktøjer

Office XP applikationer kan vise hjælp skrevet og oversat med enten HTML Help (.chm fil) eller WinHelp 4.0 (.hlp fil). Kontekst afhængig hjælp til Office XP skal laves med WinHelp 4.0. Dette sker ved hjælp af Microsoft Help Workshop eller et andet WinHelp 4.0 forfatterværktøj.

Installation af Microsoft Help Workshop 1.32

- Hent filerne 'HelpDocs.zip' og 'htmlhelp.exe' fra www.microsoft.com/downloads
- Pak filen 'HelpDocs.zip' ud i folderen hvor filen 'htmlhelp.exe' ligger
- Start filen 'htmlhelp.exe'
- Klik 'Yes'
- Klik 'Ja'
- Klik 'OK'
- Hvis der kommer et vindue om at computeren allerede har en nyere version af HTML Help klikkes 'OK'
- Klik 'OK'

Programmet startes via 'Start > Programmer > HTML Help Workshop > HTML Help Workshop'.

Komponenter

WinHelp kan samarbejde med disse filer:

- Hjælp (.hlp) filer
- Filer til indholdsfortegnelse – eng: content - (.cnt) filer (ren ASCII)
- Oversatte (.gid) .cnt filer og indeks nøgleord trukket ud fra flere .hlp filer
- Fuldtekst søgning (.fts) filer
- Fuldtekst gruppering (.ftg) filer
- Bemærkning - eng: annotation - (.ann) filer (samme filstruktur som .hlp-filer)
- Bogmærke (.bmk) filer (samme filstruktur som .hlp-filer)

WinHelp kan starte makroer som for eksempel Shell execute, men det kræver at 32-bits udgaven WinHlp32.exe bruges. Hvis man bruger 16-bits udgaven WinHelp.exe er makroer ikke en mulighed.

Desuden giver WinHlp32.exe mulighed for fuldtekst søgning ved hjælp af FTsrch.dll, som interagerer direkte med fts- og ftg-filer.

Både WinHelp.exe og WinHlp32.exe er en del af en normal Windows 2000 installation.

Bilag

Oprettelse af WinHelp hjælpefil

- a. WinHelp kan dannes ved hjælp af et Word-dokument gemt i en rtf-fil, men for at der er kontekstafhængig hjælp skal der
 - a. Indsættes et tvunget sideskift mellem hvert emne.
 - b. Indsættes en fodnote, hvor Custom Mark er sat til # som fodnote symbolet. Selve fodnoten indeholder en unik streng, som senere kaldes "topic ID". Der er bedst at strengen består af diakritiske tegn og ikke indeholder mellemrum.
- b. Projektfilen (.hpi) indeholder alle navnene på rtf-filerne i projektet.
- c. Hvis man ønsker 32-bits hjælp (Windows 95) skal der laves en fil til indholdsfortegnelse (.cnt), for eksempel ved hjælp af Help Workshop.
- d. Endelig skal rtf-filerne oversættes ind i en WinHelp-fil (.hlp), for eksempel ved hjælp af Help Workshop. Dette sker med hcw.exe, som er installeret i for eksempel Visual Studio .Net 2003 i folderen %ProgramFiles%\Microsoft Visual Studio .NET 2003\Common7\Tools.
 - a. Åbn projektfilen (.hpi)
 - b. Tilføj referencer til hvert emne med knappen Map... Referencerne får hér en numerisk kontekst ID mappet til det emne id (topic ID) de fik i .rtf-filen.
 - c. Når alle referencer (og andet) er på plads oversættes filen. Bemærk at det ser ud som om at MS Access er case-sensitiv i denne sammenhæng, og da Microsoft Help Workshop danner en .hlp-fil med filnavnet i versaler, skal angivelsen også være skrevet i versaler i MS Access.
 - d. .hlp-filen kan dog let omdøbes med et Windows Shell script (.cmd) der også står for oversættelsen til .hlp med hcw.exe. Dette letter vedligeholdelsen når indholdsfortegnelsen (.cnt) og projektet (.hpi) er dannet første gang.

WinHelp 4.0 oversætter

Filerne hcw.exe og hcrtf.exe er kommandolinje værktøjer, som er velegnet til opbygning af mange hjælpefiler.

Bemærk at hcw.exe kræver hcrtf.exe, og at filerne skal ligge i samme folder.

Syntaksen for oversætterne er:

hcw [/a] [/c] [/e] [/m] [/n] [/r] [/t] *projektfilnavn*

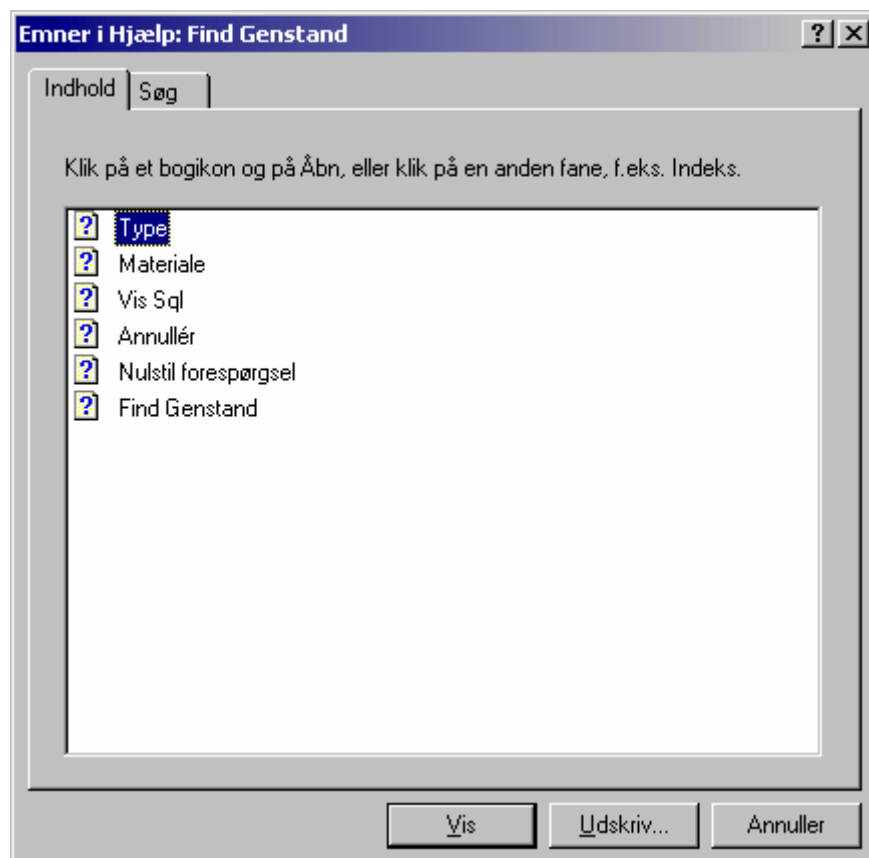
- /a: Gør kontekst ID og kildefil tilgængelig fra kontekst (Context) menu på en hjælp (Help) side. Dette svarer til muligheden 'Include .rtf filename and topic in Help File' i oversætterens 'Help File' dialog.
- /c Oversæt .hpi og .hmk filerne i kommandolinjen.
- /e Lukker Help Workshop når oversættelsen er færdig
- /m Minimerer Help Workshop mens der oversættes. Dette svarer til muligheden 'Minimize window while compiling' i oversætterens 'Help File' dialog.
- /n Slår kompression fra, hvilket giver en hurtigere oversættelse. Dette svarer til muligheden 'Turn off compression' i oversætterens 'Help File' dialog.
- /r Viser den oversatte hjælpefil. Dette svarer til muligheden 'Automatically display Help file in WinHelp when done' i oversætterens 'Help File' dialog.
- /t Sætter oversættelses tilstand (Translation mode), hvilket gør at .hpi og .cnt filer ikke bliver editert uden varsel. Dette svarer til menupunktet 'File' | 'Translation' i Help Workshop.

- hcrft [/a] [/b] [/o] [/nc] [/x] filnavn.rtf filnavn.hpj
- /a Gør kontekst ID og kildefil tilgængelig fra kontekst (Context) menu på en hjælp (Help) side. Dette svarer til muligheden 'Include .rtf filename and topic in Help File' i oversætterens 'Help File' dialog.
 - /b *sti* Giver sti-oplysninger til bitmaps som projektet indeholder. Denne mulighed svarer til BMROOT muligheden i sektionen [OPTIONS] i .hpj filen. Hvis der er flere stier skal der bruges flere /b'er
 - /fc *filnavn.hlp output_filnavn*
Generer en liste af kontekst ID'er.
 - /fd *filnavn.hlp output_filnavn*
Generer en liste af al teksten i .rtf filerne.
 - /ff *filnavn.hlp output_filnavn*
Generer en liste af hash ekvilaventer i alle kontekst ID'er.
 - /fg *filnavn.hlp output_filnavn*
Generer en liste af .hpj muligheder (options).
 - /fh *filnavn.hlp output_filnavn*
Generer en liste af alle K-nøgleord.
 - /fi *filnavn.hlp output_filnavn*
Generer en liste af alle A-nøgleord.
 - /o *filnavn.hlp*
Navnet på .hlp målfil. Denne mulighed bruges til at oversætte en .rtf fil uden en .hpj fil.
 - /nc Slår kompression fra, hvilket giver en hurtigere oversættelse. Dette svarer til muligheden 'Turn off compression' i oversætterens 'Help File' dialog.
 - /x Giver flere muligheder. Det er muligt at bruge flere /x'er, men der må ikke sættes mellemrum mellem mulighederne.
 - /xc Oversætter den RTF test som ligger på klippebordet (Clipboard). Brug ikke denne mulighed sammen med /xr.
 - /xh Viser den oversatte hjælpefil. Dette svarer til muligheden 'Automatically display Help file in WinHelp when done' i oversætterens 'Help File' dialog.
 - /xn Skjuler vinduet der viser forløbet af oversættelsen mens der oversættes.
 - /xr Oversætter en .rtf fil i stedet for en .hpj fil. Brug ikke denne mulighed sammen med /xc.
 - /xt Slår fejlkontrol fra for en .rtf fil.

Diskussion

Winhelp er en Windows 3.x-ting som fulgte med op til Windows 95, men som ikke rigtig har udviklet sig siden. Det ser ud som om det kun er Office der holder liv i miljøet – Resten af Microsofts miljøer er gået over til HTML Help (.chm). Dette kan for eksempel ses af at mange af Googles henvisninger til WinHelp ikke findes.

Bilag



Figur 19-27. Hjælp vinduet hvor de aktuelle hjælpefiler kan vælges.

Det lykkedes at danne en fornuftig hjælp til en enkelt formular, men der er foruden skrivearbejdet mange manuelle operationer i aktiviteten med at opbygge og vedligeholde en anvendelig kontekst afhængig hjælp.

Evaluering af milestones

Bilag 19.14

Evalueringen af milestones er stedet vor vi kort gennemgår arbejdsgange, forløb, beslutninger om ændringer i planer, succes og fiasko samt uddrager resultatet af det lært.

1. milestone - Indledning og oplæg

På baggrund af research i KSA samt interview med rekvirenten af systemet er der udfærdiget et Projektoplæg og -aftale dokument der blev godkendt på et indledende styregruppemøde.

Generelt

Oplæget giver et godt udgangspunkt og vi føler vi hermed er godt i gang!

Tidsestimatet på 80 timer rummer ikke spikes og forsøg på at skaffe sig teknisk indsigt i MS Access, Visio, Projekt mm.

Tidsplanen blev overholdt (evalueringen blev dog skubbet til efter sommerferien).

Hovedopgavetilmelding blev afleveret 2003.05.22

Projektoplæg og -aftale blev godkendt ved styregruppemøde 2003.06.17.

Review og evaluering af 1. milestone foregik 2003.07.30.

2. milestone - Projektgrundlag

Projektgrundlaget bygger på opgavens formål, mål, og den aftalte projektstruktur.

Risikoanalysen er indskrænket til det mest nødvendige og strategi er tillempet opgavens art og udviklernes resurser.

Tidsestimeringen kan formodentlig finpudses, men er rigeligt god som den er nu.

Skabeloner for rapporter og bilag er klar og vi har et godt fundament at starte selve projektet på.

Generelt

Gruppearbejdet fungerer til begges tilfredshed, men den tid vi bruger sammen skal fremover intensiveres (1-2 gange ugentligt) for at sikrer fælles forståelse – vi har vidt forskellig baggrund.

Tidsestimatet på 40 timer blev planlagt som del af denne fase og passede derfor helt perfekt.

Tidsplanen blev overholdt (review dog skubbet til efter sommerferien).

Projekt grundlaget blev afsluttet 2003.07.04

Det har endnu ikke været muligt at komme i kontakt med vejleder FB (sommerferie?)

Review og evaluering af 2. milestone foregik 2003.07.30.

Bilag

3. milestone - Virksomhedsanalyse

Virksomhedsanalysen er holdt på et absolut minimum, men er nødvendig for at forstå omgivelserne, interessenterne og imødegå teknologiindførsels mulige negative indvirkninger. Samtidigt gøres der her, sammen med gennemgangen af anvendelsesområdet og det rige billede fra foranalysen, klar til at beskrive arbejdsgangene for det fremtidige IT-system (de skal dog løbende samordnes med de kommende usecases).

Tidsestimatet på 20 timer passede fint til den lidt rudimentære fremstilling.

Tidsplanen blev overholdt.

Virksomhedsanalysen blev afsluttet 2003.08.15.

Review og evaluering af 3. milestone foregik 2003.08.17

4. milestone - Foranalyse

Der er her foretaget en gennemgang af anvendelsesområdets hidtidige arbejdsgange, problemidentifikation og problemområdets krav til data samt en begrebsforklaring. Det rige billede har eksisteret i en del forskellige udgaver, men der er nu konsensus om arbejdsgangene og vi har en god forståelse af hvad der ønskes af IT-systemet. Den grundige gennemgang af krav til data (og begrebsforklaring) og giver en solidt baggrund forud for analysen.

Generelt

Tidsregistrering i MS Project ser flot ud, men er utrolig tidskrævende at vedligeholde. I vores projekt med blot to deltagere har det vist sig at være et overkill. Det er oplagt til større projekter med mange parallelle arbejdsopgaver og mere end tre deltagere, men vi mener at kunne anvende tiden bedre på andre opgaver og har valgt at nøjes med omhyggelig opdatering af vores Gantt diagram og meget detaljerede projektplan.

Tidsestimatet på 30 timer blev overholdt.

Foranalysen blev afsluttet 2003.08.21 med det forbehold at det ikke har været muligt at afholde møde med den ansvarlige arkivar PH. Mødet bør afholdes inden slice2 og evt. ændringer til det rige billede og arbejdsgange må justeres efterfølgende.

Review og evaluering af 4. milestone foregik 2003.08.21.

5. milestone - Slice 1

OOA

Det blev besluttet at anvende UML og usecase, i stedet for hændelsestabeller og brugsmønstre, til at fastlægge og udlede funktionaliteten fra arbejdsgangene, fordi det bedre afspejlede vores meget visuelle måde at illustrere ideer og problemstillinger i gruppearbejdet.

Brugergrænsefladen frembød ikke de store diskussioner og blev fremstillet direkte i databaseprogrammet hvorfor den ikke detaljeret er beskrevet i konstruktionsfasen. Fasen igennem var der derimod problemer med at få klar besked om krav og anvendelsen af de enkelte datafelter for genstande.

Der blev forholdsvis hurtigt opnået enighed omkring klassediagrammer og modelkonstruktionen, ligesom det blev besluttet at afstå fra anvendelsen af tilstandsdiagrammer bl.a. fordi hovedparten af grunddatatyperne på forhånd var kendte.

OOD

Fasen blev indledt med ganske meget fokus på databasekomponenten og arkitektur generelt, men hvilken reel komponentstruktur vi ønskede for vores MS Access applikation blev ikke i tilstrækkelig grad fastlagt indledningsvist, hvilket skulle vise sig helt ødelæggende for projektplanen. Ligeledes tog det os det meste af slicen at gennemskue opståede problemer mellem den objektorienterede model vi fulgte indledningsvist og MS Access standardmåde at håndtere data og muligheder i VBA kodemiljøet der lægger op til en noget afvigende arkitektur. Først da forståelsen omkring dette kom på plads og projektet efterfølgende blev tilrettet, kunne der igen fortsættes fra en fælles platform. På baggrund af to spikes fravalgte vi ADO til fordel for den traditionelle DAO forbindelse til databasekomponenten, ligesom vi afslutningsvist valgte overalt at binde brugerinterfacet via forespørgsler, for at sikre afgrænsning og fleksibilitet.

Konstruktion

Efter en smertefri realisering af tabelstrukturen, standardforespørgsler og hovedformular, stødte vi på vores første rigtige problemer, jævnfør det ovenstående.

Det alt for løst aftalte designforløb omkring komponentstrukturen, gjorde at vi talte forbi hinanden og den manglende fortrolighed med MS Access og de metoder der ofte anvendes i dette miljø, gjorde det ikke muligt at opnå tilstrækkelig enighed om en helhedsløsning. På forsøgsbasis blev der fokuseret på indtastningsfunktionen der forstilledes realiseret via kode og ubundne felter. Dette førte gradvist og nærmest umærkeligt frem mod en realisering af hele modellen i VBA kode, en mulighed vi oprindeligt havde afskrevet os som værende alt for kompleks til dette projekt og hverken var i overensstemmelse med oplæg eller ønsket om simplicitet. Det blev undervejs klart at indskrivningsformularerne ikke så let kunne bruges til datavisning og at søgning ligeledes krævede dedikerede formularer. Alt i alt en betydelig mere kompleks struktur end der fra starten var lagt op til.

Med hensyn til at opnå fælles forståelse af problemstillinger har det netop her vist sig, at vores forskellige baggrunde har skabt et lille misforhold. Firmaarbejdsgange, MS Accessdesign og -konstruktionsmekanismer mm. kan ikke umiddelbart gennemskues ved blot at nedskrive hensigtserklæringer og retningslinier – det tager tid at snakke på plads.

Efter 2 – 3 ugers kodning og en overskridelse af tidsplanen uden at en realisering var inden for umiddelbar rækkevidde, blev der iværksat en plan B. Fokus blev fordelt mellem det igangværende og en visningsversion der byggede på formularer bundet til data gennem forespørgsler.

Bilag

Denne løsning kom relativt hurtigt på benene og det blev aftalt at fortsætte hele projektet af denne rute - der uden nævneværdig tilføjelse kunne håndtere både opdatering og oprettelse. Styringsfunktionalitet og grundlag for søgefunktionalitet fra den oprindelige kode blev migreret og ny funktionalitet kodet jævnfør konstruktionsfasebeskrivelsen.

Det var på mange måder en hård fase at komme igennem for den endelige beslutning handlede ikke kun om et rationelt valg, men også om iblandet personlige følelser. Vi var dog afslutningsvis enige og afklarede om beslutningen.

Det må i denne sammenhæng nævnes, at vi generelt har undervurderet tidsforbruget af denne fase, der kom til at rumme gennemlæsning af flere bøger og rigtig meget ”prøv og fejl” med MS Access programkonstruktion der var os fremmed.

Test

Testspecifikationen er prima at arbejde ud fra og virker efter hensigten. Selve testningen foregik med Anni Bentsen som testperson, og var en lærerig oplevelse. Det blev her klart for os, at en sådan applikation ikke kan stå alene, men at der skal ganske meget instruktion til før brugerne opnår den fornødne tryghed og rutine ved anvendelsen. Der kom desuden flere gode forslag for dagen, der er noteret og medtages i næste slice.

Implementering

Applikationen blev installeret på filserveren i simplest tænkelige version og genveje tilrettet i henhold til Multibruger spike (se bilag 19.6 s. 153). Runtime miljøet fungerede efter hensigten.

Generelt

Vores totale kuldsejling og efterfølgende ”recovery” i denne slice, har været særdeles lærerig og det er naivt at tro, at man kan forhindre alle fejl og afsporinger et projekt kan komme ud for – det handler i høj grad om at kunne agere og få det bedste ud af også disse situationer. En yderligere styrke i projektgruppen viste sig ved, at der på intet tidspunkt har været dårlig stemning, selvom humøret ved et par lejligheder nærmede sig ”nulpunktet”.

Sicen igennem har vi haft stor glæde af de spikes der er foretaget og de har udgjort grundlaget for mange af vores beslutningsprocesser omkring teknologi.

Tidsestimatet på 235 timer blev overskredet med ca. fire uger og gav anledning til begrænsning i realiseringen af det resterende projekt, i overensstemmelse med projektplanen (se Projektets realisering i forhold til projektplanen s. 27) og prioriteringen fra kravspecifikationen (se bilag 19.2 s. 127). Dette er således sket inden for projektrammerne, da det som det fremgår aldrig har været meningen at projekt kunne realiseres fuldt ud inde for denne opgave (bortset fra i udviklernes helt urealistiske drømmevisioner).

Slice 1 blev afsluttet 2003.11.13.

Review og evaluering af 5. milestone foregik 2003.11.13.

6. milestone - Slice 2

OOA

Der har her været en del arbejde med udarbejdelse/genskrivning af usecase og modeltegninger. Brugrinterfacet er redesignet i forbindelse med handlingsplanvinduet og der er nu en fin overensstemmelse mellem de to hovedvinduer og menulinie-funktionaliteten. Vinduesstørrelsen er blevet tilpasset i henhold til slice 1 testfasen.

OOD

Design fasen viser kun de enkelte tilføjelser til slicen samt de ændringer ønsket til den første prototype. Jævnfør Annis test af systemet efter slice 1 kom der værdifulde input, specielt kategori er medtaget i denne designfase.

Konstruktion

Konstruktionsfasen er her indskrænket til rettelser af slice 1 og et par tilføjelser omkring standartmoduler, fejl, interface og tabeller

Test

Testfasen er sprunget over da der ikke er konstrueret nævneværdig ny funktionalitet de ikke kan vente.

Implementering

Den tidligere implementering er skiftet ud med den nye.

Generelt

Afslutningsvist evaluerede vi på kommunikation og opgavedisciplin.

Der var enighed om, at de fælles standarder der blev aftalt i forbindelse med projektgrundlaget har været stor lettelse for hele det efterfølgende arbejde.

Wordskabelonerne har været en vigtig ingrediens til fremstilling af en homogen rapport, og en uvurderlig hjælp under rapportskrivningen. Der kunne dog med fordel have været taget beslutninger om anvendte billedformater (.wmf) og deres max. størrelse. Med hensyn til anvendelse af Visio burde der ligeledes have været udviklet en skabelon så tegninger herfra altid overholdt max. størrelsen for skabelonklummen. Ideen med en opdeling i underdokumenter har været hensigtsmæssig da al dokumentation befinder sig på internettet (Groupecare) og vi herfra henter og uploader de nyeste versioner. Dette system har fungeret helt upåklageligt og der har kun i den indledende fase været problemer. Mail, telefon og ugentlige møder har, den fysiske afstand mellem os til trods, sikret en tilstrækkelig kommunikation. Aftale disciplinen har været god og er blevet lettet væsentligt ved udsendelse af en udskrift af projektloggen umiddelbart efter afsluttede møder. Denne har kunnet bruges som huskeliste, og er altid fulgt til dørs på det efterfølgende møde. Arbejdet på filerne er blevet styret godt og kun i få tilfælde har været nødvendigt at migrere kode eller tekst fra en version til en anden.

Tidsestimatet på de oprindelige 185 timer blev reduceret jævnfør planlægningsændringer efter slice 1. Bortset fra at konstruktionsestimater generelt i dette projekt er blevet helt fejlbedømt, ligger de fleste estimater tæt på den tid vi har anvendt på realiseringen.

Der har været en del tidspres og vi har måttet afpasse opgaverne efter resurserne og prioriteringen fra kravspecifikationen, og kravspecifikationens arbejdsprioritering guld værd. Vi er helt tilfredse med fremskridtet og det vi lærer undervejs i processen.

Bilag

Slice 2 blev afsluttet 2003.11.28.

Review og evaluering af 6. milestone foregik 2003.11.28.

7. milestone - Slice 3

OOA

Der er iforhold til planen kommet en infoformular til, der kun er hurtigt konstrueret. Der er imidlertid gode muligheder for at den fremover kan ændres til "talerør" fra administratoren eller sustemudvikleren og brugerne.

OOD

Designet har behandlet delingen af databasen og låsning der indledningsvis foretages optimistisk.

Konstruktion

Fasens eneste reelle konstruktion har handlet om startscript.

Test

Der er foretaget en udokumenteret systemtest der viser at applikationen kan fungere med flere samtidige brugere på forskellige arbejdsstationer. Hastigheden var ikke imponerende så det blev besluttet at frontenden fremover skal konverteres til mde. Forbedringen var betydelig. Den vigtige læring her er, at det er vigtigt regelmæssigt at teste i det rigtige miljø.

Implementering

Den oprettede filstruktur er en glimrende hjælp til at overholde datadisciplin. Startløsningen i sin endelige for giver fleksibilitet til ændringer uden dette påvirker brugeren, der fremover kan benytte samme startikon.

Generelt

Afslutningsvist har vi igen haft et tættere samarbejde med sammenhængene dages fællesarbejde hvad der letter kommunikation, beslutningsproces og gensidig læring betydeligt.

Med et sidste blik bagud ville det have været hensigtsmæssigt med en mere formaliseret versionsstyring af både dokumentation og applikation – det bliver næste gang.

Tidsestimatet på de oprindelige 210 timer blev reduceret jævnt før planlægningsændringer og vi er nu hvor vi skal være før afslutningen.

Slice 3 blev afsluttet 2003.12.16.

Review og evaluering af 7. milestone foregik 2003.12.16.

8. milestone – Projektafslutning

Der er blevet evalueret på hele projektet og vi er enige om, at vi har nået vores mål over en bred front. Vi er kommet omkring mange problemstillinger der vil være til gavn for vores videre udvikling. Det har været godt at afslutte Datamatikerstudiet, med en følelse af at vi gennem hele uddannelsesforløbet, opgave for opgave, har kunnet bygge videre på tidligere erfaringer og nu kan selv.

Groupecare har specielt i denne fase med mange opdateringer i alle filer været en stor hjælp fordi hér har alle filer ligget i nyeste udgave.

Rapportskrivning for et projekt der er opdelt i faser er en utaknemmelig opgave, da der på intet tidspunkt kan gives en fornuftig perspektiveret sammenfatning. Efter den første fase er der endvidere for det meste tale om tilføjelser, men hele strukturen må gennemløbes selv om der ikke er meget nyt og kan derfor syntes ”tynd”. Også fra et læse synspunkt skal information hentes fra mange afsnit for at give overblik over det samlede forløb.

Korrekturlæsning af bekendte uden for projektet har været guld værd.

Efterfølgende har der været mange fejl der skulle rettes og vi er enige om at det ville have været en fordel hvis der i forbindelse med evalueringerne af f.eks. milestones var blevet læst korrektur af os begge. Det ville have bidraget til fælles læring og forståelse.

I forbindelse med den endelige afslutning af dokumentationen er vi blevet opmærksomme på at vores bilagsplacering -overskrifter kunne have været mere hensigtsmæssig, ligesom faste regler for henvisninger og billedtekster kunne have skabt mere klarhed.

Denne fase er ligesom konstruktionsfaserne blevet underestimeret på planlægningstidspunktet.

Tidsestimatet på de oprindelige 80 timer blev overskredet, men tidsplanen overholdt.

Projektafslutningen blev endeligt afsluttet 2003.12.18.

Review og evaluering af 8. milestone foregik 2003.12.16.

Projektlog

Bilag 19.15

Projektloggen tjener til at fastholde hovedpunkter omkring aftaler, møder og andre større aktiviteter, men gør ikke krav på at være komplet, ligesom mindre arbejdsopgaver bevidst udelades

- 2002.ultimo** **Hovedopgavesamarbejde aftales NGR MT**
 Projekt tanker gennemdrøftes og det bliver aftalt at hovedopgaveprojektet afvikles i 2. semester 2003 da NGR først skal bestå sit valgfag.
- 2003.02.06** **Indledende projekt research** **MT**
 Research og uformel aftale med KSA (HG ANB) om Skadesregistreringsprojekt.
- 2003.02.13** **Planlægnings møde** **NGR MT**
 Diskussion omkring KSA ønsker og mulige arbejds- og løsningsmodeller.
 Gennemdrøftelse af NGR's liste over mulige problemområder og forslag til indledende spikes.
- MT aftaler møde med KSA.
- 2003.05.05** **Formøde med skole** **MT & A. Tjørnmark**
 Projekt tanker og formalia blev gennemgået.
- MT håndterer tilmelding.
 - Valg af studievejleder.
- 2003.05.22** **Tilmelding til hovedopgave** **MT**
 Tilmeldingsblanket afleveret til Jane Petersen. Giroopkrævning udsendes efter ferien.
- 2003.06.primo** **Tildeling af studievejleder**
 Anette Tjørnmark har tildelt os Flemming Bislev som studievejleder.
- 2003.06.17** **Fællesarbejdsaften** **NGR MT**
 MT's oplæg til: Projekt oplæg og -aftale blev gennemdrøftet og justeret.
- MT skriver oplæget rent til styregruppemødet.
- 2003.06.18** **Styregruppe møde** **HG NGR MT**
 Præsentation og rundvisning i KSA.
 Projekt oplæg og -aftale blev fremlagt og godkendt – Vi går nu i gang.

Bilag

2003.07.03

Fælles arbejdsdag

NGR MT

Projektgrundlaget blev aftalt og revideret i forhold til oplæget.

- NGR Ser nærmere på DAO/ADO (spike).
- NGR Færdiggør oplæg til kodenstandard for: VBA, database og klassediagrammer mm.
- NGR Leger lidt med modeller i Visio (spike).
- NGR Overvejer struktur for systemdokumentation.
- MT Skriver Projektgrundlaget rent og lægger det i Groupcare (Gantt og faseopdeling mangler noget).
- MT Påbegynder projektlog.
- MT Skriver til vejleder FB igen efter ferien, men inden næste møde.
- MT Færdiggør rapportskabelonen og opsætning som hoved-/ underdokumenter i Word

2003.07.30

Fælles arbejdsdag

NGR MT

Gennemgang af arbejdet siden sidst. Evaluering af indledning og projektfasen – Milestone et og to. Fælles programmering og MS Access sikkerhed spikes.

- NGR Kontrollerer projektloggen og indsætter evt. manglende overordnede opgaver.
- NGR Prøver selv at estimere projektopgaverne.
- NGR Fortsætter med ADO (spike).
- NGR Arbejder videre med sikkerhed og logon.
- MT Arbejder videre med sikkerhed og logon – og ser på .mde dristribueringsfiler
- MT aftaler møde 2003.08.07 med AB (formiddag) og PH (eftermiddag).

Næste fælles arbejdsdag torsdag 2003.08.07 på KSA.

2003.08.08

Fælles arbejdsdag/besøg på KUA

NGR MT (AB)

Gennemgang af arbejdet siden sidst. Foranalyse i samarbejde med AB: Eksisterende arbejdsgange, problemidentifikation, fremtidigt anvendelsesområde (rigt billede) og indledende gennemgang af datakrav. PH havde ferie hvorfor MT havde arrangeret en udflugt til Det Arnamagnæanske Institut (Islandske håndskrifter) hvor Konservator Mette Jacobsen fremviste en databaseløsning der har fællestræk med en del af vores problemstilling. Afslutningsvist blev fremtidige opgaver og spikes gennemgået således at forskellige teknologier ikke først skal kortlægges i designfasen.

- NGR Kontrollerer projektloggen og indsætter evt. manglende overordnede opgaver.
- NGR Prøver selv at estimere projektopgaverne.
- NGR Fortsætter med at ADO (update-spike).
- NGR Dynamisk opdatering af tabeller/formularer og sikkerhed (låsnings)?
- NGR Er al indbygget funktionalitet DAO?
- NGR Fremstiller simpelt startscript i forbindelse med genvej.
- NGR Tegner ADO opsætning (front- & backend).
- MT Tegner de forskellige DAO opsætningsmuligheder og workgroupe sammenhænge
- MT Delt/udelt adgang – kan en bruger komme til at udelukke de andre uforsætteligt?
- MT Undersøger MS Access installation, start og afvikling i runtime miljø.
- MT Rentegner det rige billede i Visio.
- MT Sammenskriver dagens arbejde til et foranalyseudokumentoplæg der gennemdrøftes på førstkomende møde.

Næste fælles arbejdsdag foreslås i weekenden 2003.08.16 -17, tid og sted efter aftale.

2003.08.17**Fælles arbejdsdag****NGR MT**

Gennemgang af arbejdet siden sidst og MT's oplæg til minivirksomhedsanalyse og foranalyse. Efterfølgende blev omfanget for 1. slice planlagt til at omfatte selve skadesanmeldelsen og aktionsregistreringen. Det bliver dog nødvendigt overalt at se bredere på problemstillingen her i første iteration. Metodevalget for problemområdet blev i første omgang aftalt til at bygge på usecases (i modsætning til en brugsmønstertabel som ved Ålborgmetoden) der sammen med funktionalitet og GUI er emnet for næste fælles arbejdsdag. Selve GUIet gav anledning til en omfangsrig og helt central diskussion om IT-systemets virkemåde og bør sammen med grundlæggende UML forberedes til næste møde

- NGR Fortsætter med at ADO igen igen... (update-spike)
- NGR Kan man koble til tabeller og bruge grafisk genererede forespørgsler ved ADO?
- NGR Fremskaffer nogle screendumps af ATPs ordre-/sagssystem.
- MT Korrigerer den sidste del af oplægget til nye arbejdsgange.
- MT Sender de færdige rapportafsnit (virksomhed/foranalyse) til FB hurtigst muligt.
- MT Aftaler med TB at der pakkes og brændes en MS Access runtimeversion til de maskiner der enten ikke har den installeret eller fordi de har selve MS Access installeret.
- MT Undersøger hvilken Notes version der anvendes og hvornår der skiftes til Outlook.
- MT Omdanner det tegnede logo for skadede bøger til et brugbart program icon.
- MT Udarbejder en aktørkarakteristik

Næste fælles arbejdsdag er planlagt til torsdag morgen 2003.08.21. Om eftermiddagen er der aftalt vejledningsmøde med FB kl. 13:00.

2003.08.21**Fælles arbejdsdag/vejlednings møde****NGR MT (FB)**

Gennemgang af arbejdet siden sidst – ”alt afklaret”. Der blev derefter foretaget en evaluering af virksomhedsanalysen. Specielt interessentanalysen er god at have for øje når der skal prioriteres ønsker for applikationen. Usecases blev påbegyndt for slice 1. Første møde med vejleder FB bekræftede vores eget indtryk af at vi er på rette spor og godt i gang. FB satte fokus på at opgaven skulle have et hovedtema, vi skulle forfølge gennem hele arbejdet og at baggrunden/motivationen for vores valg var vigtigere end det færdige produkt. Da opgaven var vægtet klart højere end mundtlig fremlæggelse og karakteren for begge studerende ens bør den overvejende indsats lægges i dokumentationen så den rigtige karakter allerede nås her. Resultatet af vores spikes vedlægges som bilag og der henvises til den gennemgang af mulige løsninger og teknik vi her har foretaget. Rammerne for samarbejdet blev aftalt til at vi sender kapitlerne efterhånden som de bliver færdige og FB kommenterer pr. mail. Møder ved behov og i hvert fald efter hver slice. Efterfølgende evaluering af nødvendige kursændringer som følge af møde og vejleder. Afslutningsvist blev usecasene for skadesregistreringen færdiggjort.

- NGR Fortsætter med at ADO multibruger opdatering (spike).
- NGR Dokumenterer tidligere spikes.
- NGR Notes/Outlook integration – kan der sendes en dataopdatering direkte fra en mail.
- NGR & MT Begge gennemlæser Sam-data dokumentet der er grundlaget for datamatikerhovedopgaver (muligvis bør ministeriets retningslinier også nærlæses).
- MT Aftaler møde med henholdsvis PH og TB i uge 36.
- MT Indpasser og uploader Spikes.doc som del af hoveddokumentet.
- MT Tegner videre på usecase diagrammet for aktionsregistreringen (Visio).

Næste fælles arbejdsdag er planlagt til søndag formiddag 2003.08.24 og målet er at frembringe et kvalificeret bud på fasens GUI samt evaluering af foranalyse fasen.

Bilag

2003.08.24

Fælles arbejdsdag

NGR MT

Gennemgang af arbejdet siden sidst. NGR havde en mail klassespike klar til at kommunikere fra databasen – den anden vej er sværere, men ikke opgivet. MT havde uscecasene for slice 1 på plads. Selvom vi stadig mangler et møde med PH (ferie) som del af foranalysenplanen, valgte vi at afslutte fasen og evaluerede forløb og resultat. Første layout til GUI blev snakket på plads. I stedet for at tegne det i Visio som først planlagt bygges det nu direkte i MS Access hvilket sikrer at vi er fortrolige med formularmiljøet og at layoutet er realiserbart.

- NGR Afslutter ADO (spike).
- NGR Fortætter med at dokumentere tidligere spikes – Husk konklusion!
- NGR Notes/Outlook integration – dataopdatering direkte fra en mail (CDO).
- NGR Fremskaffer ministeriets retningslinier for hovedopgave.
- NGR Gennemlæser Sam-data dokumentet der er grundlaget for hovedopgaven.
- MT Aftaler møde med henholdsvis PH og TB i uge 36.
- MT Skriver spikeskonklusion for sikkerhed og multibruger.
- MT Realiserer det aftalte GUI for slice1.

Næste arbejds møde er onsdag eftermiddag kl. 12:30 2003.08.27 på KSA hvor vi starter på problemområdet med klassediagrammet.

2003.08.27

Fælles arbejdsdag

NGR MT

Gennemgang af arbejdet siden sidst. NGR havde endvidere forslag til ændringer i opsætningen af dokumentationen af de individuelle pikes, nu hvor de skal med i rapporten. Klassediagrammet for første slice blev udarbejdet og gav anledning til megen frugtbar diskussion. Et resultat er at vi nok bør se lidt bredere (mere generelt) på opgaven for at sikre fremtidige udvidelsesmuligheder.

- NGR Fortætter med at dokumentere de sidste spikes.
- NGR Arbejder hen mod en ”generisk” mail klient klasse (CDO).
- NGR & MT Har begge læst både sam-data og ministeriets vejledning til næste møde.
- MT Tilretter spikesopsætning for sikkerhed og multibruger.
- MT Rentegner klassediagram i Visio.

Næste arbejds møde er torsdag formiddag 2003.09.04 på KSA hvor vi starter på designfasen (database tabeller) og har aftalt møde med PH kl. 13:00 og TB kl. ca. 14:00

2003.09.01

Ekstraordinært aftenmøde

NGR MT

Da det viste sig hensigtsmæssigt at se bredere på projektoplægget og sikre fleksibilitet og udvidelsesmuligheder i modelstrukturen, har vi nu løseligt analyseret videre end slice 1 for at imødegå større revisioner af modellen i de senere faser. Opstilling og indhold af Systemdokumentationen blev afklaret, men bør diskuteres med FB.

- NGR & MT Prøver at finde en anden titel på såvel rapport som applikation, der afspejler de nye mere rummelige anvendelsesmuligheder.

Næste arbejds møde er som aftalt tidligere torsdag formiddag 2003.09.04 på KSA hvor vi starter på designfasen (database tabeller) og har aftalt møde med TB kl. ca. 14:00. det planlagte møde med PH kl. 13:00 er udskudt.

2003.09.04**Fælles arbejdsdag****NGR MT (TB)**

Gennemgang af arbejdet siden sidst. Det er stadig et problem at finde en løsning på mailproblematikken. Måske er der andre muligheder end CDO? Database komponenten blev gennemdrøftet og UML modellen blev forsøgt mappet til en fysisk tabelstruktur. Der mangler stadig et hjørne og maildelens tilknytning til status må gennemtænkes nøjere. På mødet med TB (KSAs IT-arkivar) blev genereringen af en MS Access-runtime-installations-CD gennemgået og Stadsarkivets arkiv på Rådhuset blev efterfølgende fremvist.

- NGR Arbejder energisk videre mod en ”generisk” mail klient klasse (CDO / SMTP?).
- NGR & MT Overvejer om implementationen/koden skal ske på dansk eller engelsk.
- MT Færdiggør hoveddelen af denne slices systemdokumentation.
- MT Tilretter bogreferencer og tilføjer ISBN .
- MT Opretter en spørgsmålsliste til FB på Groupecare.
- MT Rentegner tabelstrukturen frem til nu.

Næste arbejds møde er som aftalt tidligere torsdag formiddag 2003.09.11 på KSA hvor tabelstrukturen skal være færdig til implementering.

2003.09.11-12 Fælles arbejdsdag**NGR MT**

Gennemgang af arbejdet siden sidst. Logisk tabelstruktur blev afsluttet og den fysiske blev indtastet og bundet sammen med relationer. Liste tabellerne blev indledningsvist udfyldt og prøveindtastning direkte i tabelstrukturen virker sammenhængende og gennemtænkt. SQL forespørgsler og programmeringsdelen med den logik der skal binde brugergrænseflade og tabelstruktur blev indledt med succes – og afstedkom en behagelig optimisme.

- NGR Fortsætter med programmering af slice1 (uden ophør og til alt virker ;-)).
- NGR Tester om underformularer er så oplagte som MT påstår.
- NGR Opdaterer kodestandarderne i henhold til hans eget forslag.
- NGR Beskriver fremkomsten af den logiske model.
- NGR Motiverer fravalget af ADO.
- MT Opdaterer hjemmesiden med manglende dokumentation.
- MT Holder ferie 7 dage, men forpligter sig til momentvist at tænke på projektet.

Næste møde evt. onsdag 2003.09.24

Bilag

2003.09.24

Fælles arbejdsdag

NGR MT

Niels forklarede bl.a. hvorfor dele af klassemodellen alligevel med fordel kunne anvendes /kodes i MS Access. Dagen blev sat af til programmering af valideringsregler og -rutiner og efter en succesfuld for- og eftermiddag fortsatte vi med at konstruere en søgeformular til skadesregistreringer. Der opstod problemer med at indsætte recordsæt i formularen ligesom selve håndteringen af søgnings fremvisningen kræver lidt overvejelse.

- NGR Fortsætter med programmering af slice1 (søgning, visning af søgeresultater og visning af første status).
- NGR Justerer indtastningsformularen så felterne for arkivalie og skadesregistrering altid ses på skærmen (disabled før de kan anvendes forstås).
- NGR Motiverer fravalget af ADO.
- NGR Ser på anvendelse af regulærer udtryk.
- MT Formaterer NGR's forslag til kodestandarder.
- MT Kontrollerer mulige placeringsnøgler (PS).
- MT Ordner spikeslaout så de kan indgå i rapporten.
- MT Indsætter konverterer og manglende billeder i rapporten.
- MT Fjerner E-mail adresser i modellens mail klasse.

Næste møde evt. onsdag 2003.10.01.

2003.10.01

Fælles arbejdsdag

NGR MT

Brugerinterfacet og dets virkemåde har skabt problemer og for ikke at arbejde os ind i en blindgyde må vi gå lidt på kompromis med det første oplæg. Der konstrueres nu et visningsvindue der bindes direkte til stored queries. Indholdet skal ikke kunne manipuleres (i første omgang!). Fremfindning foregår på basis af specieltilpassede søgeformularer der generer de nødvendige SQL kald. Indtastning og opdatering foregår stadig i separate vinduer for skadesregistrering, aktionsregistrering osv.

- NGR Raffinerer indskrivningsformularen for skadesregistrering (update?) og søgeformularen.
- NGR Koder en find genstand søgeformular.
- NGR Påbegynder kodning af indskrivningsformularen for aktionsregistrering (update?) og søgeformularen.
- NGR Ser på anvendelse af regulærer udtryk i forhold til validering af inddata i tekstfelterne (kan sagtens vente til senere).
- NGR Motiverer fravalget af ADO (kan sagtens vente til senere).
- MT Konstruerer basis vis vinduet med genstand, skadesregistrering, aktionsregistrering, sagsstatus og ordrestatus (kun læs).
- MT Sætter sig ind i parameteriserede forespørgsler der skal binde vores søgning med basis vis vinduet.
- MT forsøger om store dokumenter Word kan håndterer flere indholdsfortegnelser.
- MT foreslår et møde til FB en dag i uge 42 (- mandag) efter kl. 16:00.

2003.10.07**Aftenmøde****NGR MT**

NGR havde udbygget klassemodulerne til at omfatte de abstrakte klasser og arbejdede på at hæve abstraktionsniveauet sådan at de kunne bruges fra flere forskellige formularer. De afledte objekter er påtænkt til at indeholde funktionalitet i form af styring af visning og opdatering i den ”flade” ubundne struktur der blev startet med. Arbejdet var stadig en hel del fra målet og der blev ligeledes arbejdet på ”find” forme til søgning.

Da vi p.t. er omkring tre uger forsinket og ikke umiddelbart er i nærheden af konstruktionsmålet for slice 1, blev det aftalt at iværksætte en plan-B. Indledningsvist udbygges det netop fremstillede visningsvindue til også at omfatte indtastning og redigering. Efterfølgende må vi så evaluere om denne meget simple og mere MS Access agtige vej giver os tilstrækkeligt kontrol over brugerinterfacet.

- NGR Fortsætter med at raffinere indskrivningsformularen for skadesregistrering (update?) og søgeformularen.
- NGR Fortsætter med at koder en find genstand søgeformular.
- NGR Ser på anvendelse af regulære udtryk i forhold til validering af inddata i tekstfelterne (kan sagtens vente til senere).
- NGR Motiverer fravalget af ADO (kan sagtens vente til senere).
- MT Udbygger basis vis vinduet til også at omfatte indtastning og opdatering.
- MT Arbejder videre med parameteriserede forespørgsler således at vis-vinduet kan vise forespørgsler fra find formene.

Næste møde evt. torsdag 2003.10.14 (eftermiddag-aften).

2003.10.16**Fælles arbejdsdag****NGR MT**

Projektplan og kravspecifikation blev evalueret for at se hvordan vores nye plan passede sammen med tidsplan og prioriteter – ingen problemer, der er stadig den nødvendige overensstemmelse selv om vores originale plan er skredet. MT gennemgik de aftalte ændringer til plan-B applikationen og bortset fra den manglende styring af brugeren ved nyoprettelse og validering, der jo er konstrueret tidligere, tegner det absolut lovende. En sidegevinst ved at konstruere underforms til de fleste tabeller er, at det ønskede klassehierarki til funktionalitet, automatisk kobles til formene. Næste arbejdsdags mål bliver derfor at sammenlægge kode fra de to eksperimenter, samt gennemlæse og justere rapporten og dokumentationen frem til nu, inden den udskrives og afleveres til FB.

- NGR Dokumenterer (se tidligere aftaler).
- NGR Fortsætter med at koder find søgeformular.
- MT Dokumenterer og indsætter billede af fysisk tabelstruktur fra MS Access.
- MT Rydder op i koden til plan-B applikationen.

Næste møde søndag 2003.10.19 kl. 9:00 (morgen-eftermiddag).

Bilag

2003.10.20

Fælles arbejdsdag

NGR MT

Gennemgang og organisering af rapport og systemdokumentation for mødet med FB i næste uge. Det blev aftalt at kodning af brugerstyring og samlet validering af de enkelte registreringer merges med "visningen" direkte i formularkoden eller i generelle moduler, ligesom feltvalideringen og den udestående styring af hvilke ændringer og registreringer der på et givent tidspunkt kan foretages af en bruger. NGRs smukt kodede klassehierarki medtages og forklares i rapporten som et eksempel på en del af et mere helstøbt "objektorienteret" løsningsmodel.

- NGR Koder brugerstyring via funktionalitet i formularer eller evt. moduler hvor genbrug kan komme på tale..
- MT Ser på menulinje funktionalitet.
- MT Udskriver og afleverer rapport og systemdokumentation tirsdag formiddag til FB.

Til mandag aften:

- NGR Motiverer fravalget af ADO.
 - NGR Spikes justeres mht. litteratur og forkortelser der skal forklares i begrebsforklaringen.
 - NGR Systemdokumentationen tilrettes til nuværende status.
 - MT Beskriver den nuværende hjælp i rapporten.
 - MT Skriver et afsnit om formularer til rapporten.
 - MT Indsætter et dump af den justerede fysiske tabelstruktur (A3 eller 2 x A4).
 - MT Frembringer et hoveddokument til systemdokumentationens afsnit om systemdesign og implementering.
 - MT Justere projektplanen og indfører spikes i tidsestimeringen.
 - MT Skriver et afsnit om projektstyring, kommunikation, mail, skabeloner mm.
- Næste møde onsdag 2003.10.22 eftermiddag kl. 12:00 på Bjørg's (Møde med FB kl. 13:00).

2003.10.22

Eftermiddagsmøde

NGR MT (FB)

Gennemgang af rettelserne til rapport og systemdokumentation før mødet med FB.

Møde med FB hvor hovedessensen var, at vi var godt organiseret og over halvvejs i fin stil. Vi aftalte, at det var fornuftigt at se bort fra konstruktion, test og implementering af de to efterfølgende faser, men fuldfører OOA & D samt selve databasen. FB manglede som det eneste lidt mere præcision i indledningens formuleringen, om hvad vi ønskede at få ud af projektet. Endvidere påpegede han at meget store tekniske/massive forklaringer sikkert ikke vil blive læst af hverken sensor eller vejleder. FB gik ind i en diskussion omkring arkitektur og det kan måske give et hint om mulige samtaleemner i en eksamenssituation. Der var enighed om at de vil være fornuftigt at sprede de individuelle fremlæggelser, så de dækker opgaven bedst muligt – og ikke trætter censor og FB. Afslutningsvist blev det aftalt at vi som minimum mødes medio november og umiddelbart før afleveringen for at sikre en endelig afpudsning af opgaven og demonstrere det færdige program for FB.

- MT Opdaterer vores studievejlednings log.

Næste arbejdsdag fredag 2003.10.24 kl. 9:00 til ca.17:00.

2003.10.24**Fælles arbejdsdag****NGR MT**

Gennemgang af programmet og merge af kode fra de to prototyper. Dagen gik med at udvikle indtastningsmasker, valideringsregler og meddelelser for relevante felter samt sikre styring af feltvisning.

- NGR Fortsætter med at kode styringen af hvad der kan tillades rettelse i ved en aktuel status – jævnfør usecase'ne.
- NGR Fortsætter med "find genstand" søgeformularen.
- NGR Tilretter den del af MT's kode der ikke er i overensstemmelse med standarden eller NG's best practice ;-).
- MT Færdiggør brugertilrettet menulinien til genstandsformularen, der bl.a. skal indeholde kald til gemte søgeforespørgsler og filtreringsmuligheder.
- MT Koder navigering-, opretknapper og visningsstyring til både genstand og sag.
- MT Koder knapfunktionalitet og visningsstyring for slet, annuller og godkend til både genstand og sag.
- MT Undersøger krav til data og evt. valideringsregler med PS så der er overensstemmelse med arkivdatabasen.

Næste arbejdsdag – i løbet af uge 44 – efter nærmere aftale.

2003.11.06**Fælles aftenmøde****NGR MT**

Udeståender blev gennemgået og hele opgaven blev set igennem med et lidt bredere view. Programmet blev set igennem og det meste af funktionaliteten er nu til stede i mere eller mindre afsluttet form.

- NGR Fortsætter med "find genstand" søgeformularen.
- NGR Tilretter den del af MT's kode der ikke er i overensstemmelse med standarden eller NG's best practice ;-).
- NGR Bilag om kodestandarder opdateres.
- NGR Tjekker "kilderne" for gode argumenter - hvorfor må nøgler ikke være databærende?
- NGR Retter kode – alle genstandsfelter skal enables efter oprettelse.
- NGR Retter kode – efter typeskift skal det ikke være nødvendigt at genindtaste materiale hvis det allerede eksisterer.
- NGR Tilretter kode – holdes navigationsknapperne for frem og tilbage nede skal der bladres indtil knappen slippes eller enden nås.
- MT Beskriver nuværende struktur – modelkomponenten.
- MT formulerer at vi er tilhængere af et forholdsvis statisk interface der dog kun giver adgang til relevant funktionalitet i den enkelte situation.
- MT Justere og gennemgår datavalidering i formularer og tabeller.
- MT Arbejder på at afpudse Slice 1 programmet så vi kan teste i weekenden.

Næste arbejdsdag er aftalt til søndag eftermiddag 2003.11.09 på KSA.

Bilag

2003.11.09

Fælles arbejdsdag

NGR MT

Vi var ikke på omgangshøjde med vores egne aftaler og planer. Manglende kode for genstandens korrekte visning mm. blev konstrueret og der blev fundet et par fejl som følge af forskellig opsætning af dato. Kodningen af den del af slice 1 vi har planlagt er nu 90 % færdig. Vi blev desværre ikke klar til test og må igen skyde deadline for slice 1.

- NGR Tilretter kode – holdes navigationsknapperne for frem og tilbage nede skal der bladres indtil knappen slippes eller enden nås.
- NGR Bilag om kodestandarder opdateres.
- NGR Tilretter den del af MT's kode der ikke er i overensstemmelse med standarden eller NGR's best practice ;-).
- MT Beskriver nuværende struktur – modelkomponenten.
- MT Konstruerer sletning af Genstands specialiseringer i tilfælde af at der skiftes fra f. eks kort til arkivalie.
- MT afrunder testdokumentation og tester.

Næste arbejdsdag er aftalt til torsdag eftermiddag kl. 12:00 2003.11.13 på KSA. Slice 1 evalueres kort og der startes på slice 2 som maksimalt må tage 14 dage.

2003.11.13

Fælles arbejdsdag

NGR MT

Slice 1 blev formelt afsluttet og et par sidste kodedetaljer blev diskuteret og rettet. Et par ændringer fundet ved test og bruger afprøvning bliver overført og rettet efterfølgende. Der blev foretaget en evaluering og forsøgt samlet op på hvad der gik godt/galt. Slice 2 blev igangsat med en hurtig gennemgang af den oprindelige plan og den nødvendige tilretning i forhold til tid/resurser og prioriteringen i kravspecifikationen og projektbeskrivelsen.

- NGR Tilretter kode – holdes navigationsknapperne for frem og tilbage nede skal der bladres indtil knappen slippes eller enden nås.
- NGR Fanger dialogboks efter sletning af skades- eller aktionsregistrering.
- NGR Sikre at en tegning opfattes og behandles (vises) som et kort.
- NGR Prøver om der kan indsættes tegn i fanebladsteksten.
- NGR Opretter Tabellen TBLSTOERRELSE med felterne STOERRELSE og BESKRIVELSE, tabellen bindes med relationer og tabelfeltet for dens fremmednøgle STOERRELSE_FK sættes til opslag som ved andre tilsvarende opslag. Nødvendig ændring og binding i frmGenstand fortages ligeledes.
- NGR Opretter et ekstra instruktionsfelt i TBLHANDLINGSPLAN (FREMTAGNINGSSINSTRUKTION og BEHANDLINGSINSTRUKTION). Ingen af instruktionsfelter er obligatoriske.
- NGR Tilretter den del af MT's kode der ikke er i overensstemmelse med standarden eller NGR's best practice ;-).
- MT Beskriver nuværende struktur – modelkomponenten.
- MT Beskriver testforløbet og indsætter "V" tegning.
- MT Beskriver implementering.
- MT Tegner usecase.
- MT Realisere oplæg til GUI.
- MT Prøver sig lidt frem med hvilken markering af obligatoriske felter der virker bedst
- MT Fremstiller en lille test hvor to kombinationsbokse koordineres.
- MT Aftaler de sidste to møder med FB henholdsvis 2003.12.01 eller 02 og 2003.12.15 eller 16.

Næste arbejdsdag er aftalt til onsdag 2003.11.19 kl. 8:00-18:00 (eller der omkring).

2003.11.24**Fælles Aftenmøde****NGR MT**

Slice 2 er i realiteten færdig, men dokumentationen mangler meget bl.a. omkring brugerinterfacet, der blev besluttet ændret. Vi fortsætter derfor arbejdet ugen ud og så er det slut. Forsøg med at kode "autobladrings" som vi tænkte skulle fungere ved at holde navigationsknapperne for frem og tilbage nede, blev opgivet. Efterhånden er meget af rapporten på plads og der tegner sig et rigtigt fint billede af resultatet. Kombinationsboksen for størrelse drillede og koden må ses nøjere efter – det skal virke.

- NGR Fanger dialogboks efter sletning af skades- eller aktionsregistrering.
- NGR Sikrer at en tegning opfattes og behandles (vises) som et kort.
- NGR Prøver om der kan indsættes tegn i fanebladsteksten.
- NGR Den oprettede størrelseskombinationsboks bringes til at virke eller der returneres til den oprindelige tekstboks. Herefter gennemtestes al kode der har været genstand for ændring.
- NGR Beskriver databasekomponent afsnittet i slice 2.
- NGR Beskriver den logfunktionalitet han har konstrueret i slice 2.
- MT Tilpasser GUIoplægget for handlingsplan til den ekstra medarbejder.
- MT Tilpasser GUI for genstand så fanebladene bruger hele bredden ligesom handlingsplanvinduet. Samtidigt tilrettes fejlen med de for store dimensioner.
- MT Opdaterer modelkomponenten til slice 2 (design).
- MT Opdaterer tegninger af model og klassediagram til at omfatte de ny ændringer.
- MT Opdaterer uscasene for handlingsplan.
- MT Markerer obligatoriske felter i henhold til det godkendte oplæg.
- MT Prøver at implementere de koordinerede kombinationsbokse (helst så det ligner vores model).
- MT Sikrer at det er det rigtige kodebilag der vises i rapporten.
- MT Sender en kopi af mødeaftalesvar fra FB.

Næste arbejdsdag er aftalt til fredag 2003.11.28, tidspunktet ikke aftalt.

2003.11.28**Fælles Aftenmøde****NGR MT**

NGR gennemgik sin seneste opdatering af spiken om regulære udtryk. MT demonstrerede det nye GUI layout og gennemgik de seneste rettelser til rapporten. Der var lidt snak om PDF dokumenters velsignelser og evt. anvendelse i vores projekt. Slice 2 blev evalueret. Slice 3 er blæst i gang og indeholder lidt om samtidighed, deling af databasen, implementeringsstruktur, sikkerhed på brugerniveau samt selvfølgelig de tegninger og forklaringer der er nødvendige. Altså kun lidt OOA, lidt OOD, ingen konstruktion, kun lidt test og den afsluttende implementering.

- NGR Følger den tidligere spike om hjælp helt til dørs og konstruere frameworket til hjælp – "Hej fra Hjælp" er fuldt acceptabelt, blot der er klar til senere indtastning.
- NGR Retter i afsnittet om håndtering af uddata.
- NGR Skriver et afsnit i slice 2 om standardmoduler eksemplificeret ved fejlmodulet – er det her der også vises lidt kode?
- NGR Beskriver og afslutter den logfunktionalitet han har konstrueret i slice 2.
- NGR Bringer orden i de sidste selvreferencer i ordlisten.
- MT medtager moduler i modelkomponenttegningen til slice 1, 2 og 3 (design).
- MT Opdaterer tegninger af model og klassediagram til at omfatte de ny ændringer
- MT Forsøger at konstruere de koordinerede kombinationsbokse som i modelbeskrivelsen – subsidiært som de allerede er indsat i tabellerne.
- MT Skriver et afsnit til slice 3 om samtidighed og låsestrategi.

Bilag

- MT Beskriver at også den forskellige baggrund har bidraget til problemer omkring design.
- MT Beskriver at der i tilknytning til dokumentskabelonen burde have været taget stilling til billedformater (wmf) og størrelser evt. skabeloner til Visio. Dette kunne have lettet arbejdet omkring de første illustrationer de nu må laves om.
- MT Beskriver at Groupcare har været en stor hjælp.
- MT Udskriver rapporten som den er for nærværende og beder en skrivekyndig om hjælp med fejretning og sprog.
- MT Færdiggør applikationen med hensyn til formularer og foretager deling i front- og backend.
- MT Går til møde med FB mandag 2003.12.01 kl. 14:00.
Næste arbejdsdag er aftalt til fredag 2003.12.05, eftermiddag.

2003.12.05

Fælles arbejdsdag

NGR MT

MT demonstrerede en lille infoformular, i simpleste udgave, der med tiden kan udvikles til at blive en kommunikationsmulighed fra udviklere til brugere. De koordinerede kombinationsbokse virker ligeledes om end ikke som de er fremstillet p modellen – det blev for kompliceret på nuværende tidspunkt. NGR viste at det var lykkedes at konstruere den lidt kluntede hjælp MS Access stiller til rådighed. Mangler og småfejl i koden blev vendt. Der var konsensus om de foretagne strukturændringer i rapporten og de sidste arbejdsopgaver for slice 3 blev fordelt. System dokumentationen blev samlet til et dokument og omstruktureret så vi nu er endog meget tilfredse med resultatet – der mangler dog ganske meget dokumentation. Smertensbarnet: ”opgave titel” rykkede sig nu efter en fornøjelig brainstorm på om vi i stedet for det åbenlyst nørdede, i stedet kunne finde en hookline med lidt bid i. Det var lidt grænseoverskridende og blandt de mange vilde forslag er vi foreløbig landet på noget i retning af: ”Stadsarkivets forfald”, ”Forfald i Stadsarkivet” eller muligvis blot ”Forfald i arkivet”. Vi tænker videre, men er mere på sporet nu.

- NGR Skriver et afsnit i slice 2 om standardmoduler eksemplificeret ved fejlmodulet – er det her der også vises lidt kode?
- NGR Beskriver og afslutter den logfunktionalitet han har konstrueret i slice 2 (simplest mulige konstruktion).
- NGR Udskriver rapport og systemdokumentation søndag, som afleveres til gennemlæsning for fejl i forståelsen eller strukturen, hos bror og svigerinde.
- NGR Fremstiller screendump af det lille helpfilvindue der viser de forskellige mulige filer (ikke browseren med eksemplet).
- NGR Fra år etiketten ændres til at vise Udgivelsesår for bog og blot År for kort og tegning.
- NGR Retter aftalte fejl og tester.
- NGR Kan scrollhjulet slås fra? (laveste prioritet).
- NGR Opdaterer Startscript spike.
- NGR Opretter filstruktur til server og tager screendump.
- NGR Systemdokumentationen opdateres mht. handlingsplan.
- NGR Tegninger af fysiske modeller opdateres (evt. i samarbejde når vi ses).
- NGR Skriver alle de afsnit der blev aftalt på mødet og indskrevet i hans rapport.
- MT Indføre to forskellige titler på de to dokumenter, og husker at opdatere dem senest lørdag aften kl. 24:00, så NGR kan skrive dem ud.
- MT Finder og retter relationsfejl ved oprettelse af bl.a. kort.
- MT Fjerner filtreringsmulighed på typekombinationsboksen.
- MT Tænker over visning af de tre fysiske tabeldiagrammer fra MS Access.

- MT Tegner streger ind på komponentdiagrammet og tilretter de to versioner.
 - MT Ændre tegningen af plan A så den endnu bedre giver udtryk for kompleksiteten ved at påføre formularernes klassemoduler (husk at ændre navnene) (lav prioritet).
 - MT Skriver oplæg til konklusion og perspektivering.
 - MT Omorganisere systemdokumentationen så den er opdelt gruppevis efter hvad der høre logisk sammen og opretter derefter et register så der kan slå op alfabetisk.
 - MT Overvejer om der er behov for et lille interaktionsdiagram for applikationens vinduer.
 - MT Skriver alle de afsnit der blev aftalt på mødet og indskrevet i hans rapport.
- Næste arbejdsdag er aftalt til en dag i løbet af uge 50, når vi har nået alle de ovenstående opgaver.

2003.12.11

Fælles arbejdsdag

NGR MT

Gennemgang af systemdokumentation og rapport, der på alle måder lever op til forventningerne, samt alle de afsluttede punkter. Det var ikke muligt for MT at afstille filtrering på et felt – det kan kun gøres på en formular. Ligeledes var det umuligt at reproducere en periodisk fejl – der dog senere på aftenen viste sit grimme fjæs igen en enkelt gang (muligvis en navngivnings eller kilde fejl). Drøftelse af oplæget til konklusion og perspektivering og fælles tilretning. Installation på server og test af startscript og hastighed. Dette førte til at det blev besluttet at konvertere frontenden til MDE for forbedret sikkerhed og afviklingshastighed. Titlen blev endeligt fastlagt til ”Når skaderne opdages” hvad der jo er den udløsende faktor for anvendelse af systemet og den første handling i det rige billede.

Følgende arbejdsplan forsøges afsluttet til mandag aften.

- NGR Skriver lidt overordnet tekst i slice 2 om standardmoduler?
 - NGR Retter aftalte fejl og tester.
 - NGR Tilretter startscript så de passer til forholdene.
 - NGR Retter screendump: Faeles og MDE.
 - NGR Fejllog mangler bruger og maskine nr.
 - NGR Afslutter og farver ER diagrammerne (komprimerede i slice 2 og 3)
 - NGR Skriver de sidste afsnit der er markeret i hans rapport.
 - MT Ser om han kan nå at generere en rapport for også at få den ”pind” med.
 - MT Rette to ikoner i delingstegningen for databasen og skriver vi nu anvender MDE.
 - MT Afslutter forslag til konklusion og perspektivering.
 - MT Skriver forord til systemdokumentationen.
 - MT Pepper fixer find genstand så den ligner de andre formularer (SQL knappen må godt skjules). Screendump og indsætning i systemdok og navigationsdiagram.
 - MT Komprimere diagrammer i slice 2 og 3.
 - MT Kodestandarder sidder på den gale side.
 - MT Skriver de sidste afsnit der er markeret i hans rapport.
 - MT Retter de mange hundrede fejl Marianne har fundet (søndag hvor der er fred og ro)
- Arbejdsdag lørdag 2003.12.13 kl. 10:00-16:00 på KSA. Mål: Afsluttende installation.
- Arbejdsdag mandag 2003.12.15 kl. 12:30 på Bjørk. Kl. 14:00 møde med FB og derefter på KSA. Mål: Rapport samles til et dokument og tilrettes så det stort set er klar til udskrivning onsdag.
- Arbejdsdag tirsdag 2003.12.16 morgen - 16:00 på KSA. Mål: Afsluttende test med Anni og rapport afslutning.

Bilag

2003.12.13

Fælles arbejdsdag

NGR MT

Gennemgang af og oprydning i rapport for slice1, 2 og 3 der mangler nu kun et par afsnit, lidt finpudsning og så endelig gennemlæsning og fejlretning. Der blev foretaget evaluering af slice 3 samt konklusion og perspektivering. NGR demonstrerede sine evner med scripting og den endelige struktur på serveren blev færdig og virker nu. Rapporten blev samlet til et dokument.

Arbejdsdagen blev afsluttet med: Julegløg æbelskiver og teatertur med ”konerne”.

Følgende arbejdsplan forsøges afsluttet til mandag aften.

- NGR Skriver tekst i slice 1 om standardmoduler?
- NGR Fejllog mangler bruger og maskine nr.
- MT Indsætter to diagrammer af den fysiske tabelstruktur (ikke fold-ud)
- MT Skriver forord til systemdokumentationen.
- MT Retter de mange hundrede fejl Marianne har fundet (søndag hvor der er fred og ro).
- MT Ser om han kan nå at generere en rapport for også at få den ”pind” med.
- MT Fixer findGenstand så den ligner de andre formularer (SQL knappen må godt skjules). Screenshot og indsætning i systemdokumentation og navigationsdiagram.
- MT Alle referencer til bilag skal gås efter da to hovedoverskrifter er faldet ud..
- MT Retter farvefejl i alle handlingsplan diagrammer – sort tekst.
- MT Justerer Kravspec og Testspec.
- MT Justerer projektfasen i henhold til astrofysiske standarder ;-).
- MT Indsætter nummerering på alle billedundertekster.
- MT udskriver rapport og systemdokumentation der afleveres til FB mandag morgen.

Arbejdsdag mandag 2003.12.15 kl. 12:30 på Bjørk. Kl. 14:00 møde med FB og derefter på KSA. Mål: Rapport og systemdokumentation tilrettes så det stort set er klar til udskrivning onsdag.

2003.12.15

Fælles arbejdsdag

NGR MT

Møde med FB og snak om projekteksamen og fremvisning af program – ingen negative kommentarer. Bagefter opgave tilretning af fejl i opgaven – en opgave i sig selv. Det skulle have været sket meget før.

- INGEN NY LISTE – den seneste projektlog afsluttes.

Arbejdsdag tirsdag 2003.12.16 morgen - eftermiddag på KSA. Mål: Afsluttende test med Anni, evaluering og rapport afslutning.

2003.12.16

Fælles arbejdsdag

NGR MT

Afsluttende installation og opsætning. Test med Anni Bentsen – det begynder at ligne noget. Mere tilretning i systemdokumentationen.

- MT Tilretter rapport og systemdokumentationen endeligt.
- MT Indkøber kontor artikler og printer opgave og CD.
- MT Printer og færdiggør hovedopgaverne og CD (7 stk) Husk forside!

Fælles aflevering få Niels Brock og Københavns Stadsarkiv 2003.12.19 formiddag.

Referencer

20 Hjælpemidler

20.1 Bøger

20.1.1 Primær litteratur

Ditmer, Morten (2001)

Introduktion til applikationsudvikling – med Access 2000, København V: Ingeniøren|bøger
ISBN 87-571-2402-7

Mathiassen, Lars et. al (2001)

Objekt Orienteret Analyse & Design, 3. udgave, Aalborg: Forlaget Marko Aps
ISBN 87-7751-153-0

Novalis, Susann & Jones, Dana (2001)

Access 2002 VBA Handbook, Sybex
ISBN 0-7821-4013-0

Shank, David & Roberts, Mark & Myers, Tamra (1999)

Microsoft Office 2000/Visual Basic Programmer's Guide, Redmond: Microsoft Press

Smith, Robert & Sussman, Dave et. al (2003)

Beginning Access 2002 VBA, Birmingham: Wrox Press
ISBN 1-861000-821-X

Bekendtgørelse om datamatikeruddannelsen

(BEK nr 248 af 29/03/1996, LOV Nr. 418 af 06/06/2002)

<http://www.retsinfo.dk/DELFIN/HTML/B1996/0024805.htm>

Datamatikeruddannelsen, Niels Brock (2002)

Eksamensvejledning, 5. semester

20.1.2 Sekundær litteratur

Birchall, Graeme (2003)

DB2 UDB V8.1 SQL Cookbook, NYC:

http://ourworld.compuserve.com/homepages/Graeme_Brichall

Childs, Matt & Lomax, Paul & Petruscha, Ron (2000)

VBScript in a Nutshell - First Edition, Sebastopol (CA): O'Reilly & Associates
ISBN 1-56592-720-6

Christiansen, Poul Erik & Trojel, Thomas (1996)

Organisation og Logistik, Nykøbing F: Trojka
ISBN 87-89830-31-8

Referencer

- Dobson, Rick (2001)
Programming Microsoft Access Version 2002 Core Reference, Redmond: Microsoft Press
ISBN 0-7356-1405-9
- Eaglestone, Barry & Ridley, Mick (2001)
Web Database Systems, Berkshire: McGraw-Hill Book Company Europe
ISBN 0-07-709600-2
- Elmasri, Ramez & Navathe, Shamkant B. (2000)
Fundamentals of database systems –Third Edition, Addison-Wesley, kapitel 16
ISBN 0-8053-1755-4
- Groff, James R. & Weinberg, Paul N. (1994)
LAN Times Guide to SQL, Osborne McGraw-Hill, kapitel 12
ISBN 0-07-882026-X
- IBM Redbook (2000)
COM together – with Domino, Armonk: IBM International Technical Support
Organization
SG24-5670-00
- IBM Redbook (2003)
DB2 UDB Exploitation of the Windows Environment, Armonk: IBM International
Technical Support Organization
ISBN 0738427985
- Lammers, Don & O’Rear, Paul (2000)
WinHelp Programmer’s Reference
PrgRefWH.zip (fundet vha. Google)
- Microsoft Corporation
Microsoft ActiveX Data Objects 2.5, Redmond: Microsoft Corporation
ADO210.CHM
- Microsoft Corporation
Microsoft DAO 3.60, Redmond: Microsoft Corporation
DAO360.CHM
- Microsoft Corporation
Microsoft Jet Error Message Reference, Redmond: Microsoft Corporation
JETERR40.CHM
- Microsoft Corporation
Microsoft Jet SQL Reference, Redmond: Microsoft Corporation
JETSQL40.CHM
- Microsoft Corporation (1995)
The Component Object Model Specification Version 0.9, Redmond: Microsoft Corporation
DRAFT: October 24, 1995

Microsoft Corporation (1997)
Applikationsudvikling med Microsoft Access 97, Redmond: Microsoft Corporation
Artikelnr. DB84909, p 310ff

Microsoft Corporation (1999)
Windows Management Instrumentation White Paper – Background and overview,
Redmond: Microsoft Corporation

Microsoft Corporation (2000?)
SQL Server Books Online, Redmond: Microsoft Corporation

Microsoft Windows Resource Kit Scripting Team, The (2002)
Microsoft Windows 2000 Scripting Guide, Redmond: Microsoft Press
ISBN 0-7356-1867-4

Munk Madsen, Andreas (1996)
Strategisk Projektledelse, Aalborg: Forlaget Marko Aps
ISBN 87-7751-115-8

Online Training Solutions, Inc (2001)
Microsoft Access Version 2002 Step by Step, Redmond: Microsoft Press
ISBN 0-7356-1299-4

Resources Online (2001)
Microsoft Visio Version 2002 Step by Step, Redmond: Microsoft Press
ISBN 0-7356-1301-X

Roff, Jason T. (2001)
ADO – ActiveX Data Objects, Sebastopol (CA): O'Reilly & Associates
ISBN 1-56592-415-0

Rolland, F. D. (1999)
Start på SQL, Valby: IDG Forlag
ISBN 87-7843-169-7

Scott, Kendall (2001)
UML Explained, New Jersey: Addison-Wesley
ISBN 0-201-72182-1

Vinje, Poul Staal (2000)
Projektledelse af systemudvikling, 2. udgave: Ingenøren|bøger
ISBN 87-571-2059-5

WinHelp Survival Kit
<http://www.geocities.com/area51/6793/helpsurv.htm>

Viescas, John L. (1999)
Running Microsoft Access 2000, Redmond: Microsoft Press
ISBN 1-57231-934-8

Referencer

20.2 Internet links

www.google.com	Meget givtig søgemaskine.
www.rational.com	UML og projektstyring.
www.rational.dk	UML og projektstyring.
msdn.microsoft.com/library	Essentiel udviklingsside for MS produkter.
www.microsoft.com/technet	Essentiel teknikside for MS produkter.
www.microsoft.com/office/access	MS Access introductions side.
www.microsoft.com/office/project	MS Project introductions side.
www.microsoft.com/office/visio	MS Visio introductions side.
www.microsoft.com/office/word	MS Word introductions side.
www.notes.net	

20.3 Software

Microsoft Access 2002 DK/EN.
Microsoft Project 2002 standard DK.
Microsoft Visio 2002 pro DK og Professional 2003 DA.
Microsoft Word 2002 DK/EN.
Microsoft Visual Studio .NET 2003 Professional.
MSDN Subscriptions Library (July 2003).
IBM Lotus Notes 5.0.6 og .11 DK (inklusive Designer).

Ord og begrebsforklaring

21

Active Directory Services Interfaces (ADSI): Katalog service til Microsoft Windows, som giver adgang til et netværks resurser.

ActiveX Data Objects (ADO): Højniveau interface til OLE DB. Begge dele er indeholdt i MDAC.

ADO – se ActiveX Data Objects.

ADSI – se Active Directory Services Interfaces.

American National Standards Institute (ANSI): USA repræsentant I ISO. Se også www.ansi.org.

ANSI – se American National Standards Institute

Anvendelsesområdet: En organisation, der administreres, overvåges eller styres ved hjælp af et system.

API – se Application Programming Interface

Application Programming Interface (API): Programmerings interface på lavt niveau til et system som for eksempel Windows. Et API bliver ofte pakket ind i et framework som .NET, Microsoft Foundation Class (MFC) eller Borlands Visual Component Library (VCL). Et API kan også være pakket ind i en samling af middleware komponenter.

arbejdsgruppe: Gruppe af brugere i et flerbrugermiljø, der er fælles om data og en arbejdsgruppeoplysningsfil.

arbejdsgruppeoplysningsfil: Fil, som MS Access læser ved starten, der indeholder oplysninger om brugerne i en arbejdsgruppe. Disse oplysninger omfatter f.eks. brugernes kononavn, deres adgangskode og angivelse af de grupper, de er medlem af.

arbejdsgruppe-id: Alfamerisk streng på 4 til 20 tegn, der skelner mellem store og små bogstaver, som du indtaster, når du opretter en ny arbejdsgruppeoplysningsfil ved hjælp af Workgroup Administrator. Denne identificerer entydigt administratorgruppen til denne arbejdsgruppefil.

Brugerkonto: En konto, der er identificeret ved et brugernavn og en personlig identifikation (PID), der er oprettet for at kunne kontrollere brugerens adgang til databaseobjekter i en MS Access-arbejdsgruppe.

Call-Level Interface (CLI): Tillader adgang til database uden at bruge SQL-kald. Bygger på Microsofts ODBC og blev føjet til SQL standarden i 1995.

Ordforklaring

Cascading Style Sheets (CSS): Samling af stil-definitioner til HTML dokument.

CDO – se Collaboration Data Objects.

CLI – se Call-Level Interface.

Collaboration Data Objects (CDO): Objekt bibliotek med COM interface til sende og modtage beskeder (mail) i et Microsoft miljø.

COM – se Common Object Model.

Common Object Model (COM): Grundlæggende objektmodel som for eksempel ActiveX og OLE bygger på. Objekter er kendetegnet ved deres interfaces.

CSS – se Cascading Style Sheets.

CSS-fil: Fil indeholdeholde en CSS implementation.

DAO – se Data Access Objects.

Data Access Objects (DAO): En dataadgangsbrugerflade, der kommunikerer med Microsoft Jet, ODBC og ISAM kompatible datakilder for at forbinde til, hente, manipulere og opdatere data og databasestrukturen. Se spiken ”Hvad er ADO?”.

Data Manipulation Language (DML): Den del af SQL som benyttes til håndtering af data, for eksempel ved kommandoerne INSERT, SELECT, UPDATE eller DELETE.

Database Administrator (DBA): Person som er ansvarlig for administrationen af databasesystemers resurser.

DBA – se Database Administrator

DLL – se Dynamic Link Library.

DML – se Data Manipulation Language.

Document Object Model (DOM): Objekt model for et HTML dokument.

DOM – se Document Object Model.

Dynamic Data Exchange (DDE): Protokol til udveksling af data mellem Microsoft Windows-baserede programmer.

Dynamic Link Library (DLL): En komponent som en applikation kæder sig sammen med på kørselstidspunktet.

ECMA – se European Computer Manufacturers Association.

EDB – se Elektronisk Data Behandling.

Elektronisk Data Behandling (EDB): 70'ers begreb om aktiviteter der afvikles på computer. Forkortelsen anvendes stadig.

Entitet-Relation (ER): Højniveau konceptuel datamodel, mest kendt fra ER-diagrammer.

ER – Se Entitet-Relation

European Computer Manufacturers Association (ECMA): Organisation som varetager standarderne for blandt andet ECMAScript C# og CLI. ECMAScript er implementeret af Microsoft i JScript og af Mozilla i JavaScript.

EXtensible Markup Language (XML): Sprog til beskrivelse af strukturerede data ved hjælp af tag sæt. Koden ligner HTML, med XML er stringent og entydigt.

EXtreme Programming (XP): Metode til systemudvikling som blandt andet tager udgangspunkt i historier (stories), enkelt design og korte faser med fokus på anvendelighed for kunden.

Forespørgsel: Et spørgsmål om de data, der er gemt i dine tabeller, eller en anmodning om at udføre en opdatering af nogle data. En forespørgsel kan bringe data fra flere forskellige tabeller sammen, så de tilsammen kan danne en kilde af data til en formular, rapport eller en dataadgangsside.

gemt procedure (Stored Procedure): Et prækompileret SQL udtryk der er gemt i databasen under et navn og behandles som en enhed. De opbevares i en SQL-database, og afvikles via ét opkald fra et program.

global menulinje: En speciel brugerdefineret menulinje, der erstatter den indbyggede menulinje i alle vinduerne i MS Access programmet, undtagen i de tilfælde, hvor der er angivet en brugerdefineret menulinje til en formular eller rapport.

Graphical User Interface (GUI): Grafisk brugergrænseflade til et IT system. For eksempel Microsoft Windows eller Motif.

gruppekonto: En samling af brugerkonti i en arbejdsgruppe, der er identificeret ved gruppenavn og personlig id (PID). De tilladelser, der er tildelt en gruppe, gælder for alle brugere i gruppen.

GUI – se Graphical User Interface.

HTA: HTML Application. Script der er XML formatteret og bliver afviklet med Mshta.exe, det vil sige at der er adgang til DOM som i en webbrowser, men der er ingen begrænsninger på det lokale system.

HTML – se Hypertext Markup Language

Hypertext Markup Language (HTML): Sprog til beskrivelse af et dokument formatering ved hjælp af tags. Sproget er ikke stringent.

IE: Microsoft Internet Explorer

Ordforklaring

Indexed Sequential Access Method (ISAM): Generelt databaseformat som blandt andet omfatter dBASE, Borland Paradox og Microsoft Excel.

Informations Teknologi (IT): Betegnelse for en teknologi, der kan anvendes til informationsbehandling eller -håndtering i el eller anden grad.

International Organization for Standardization (ISO): Se www.iso.org

ISAM – se Indexed Sequential Access Method.

ISO – se International Organization for Standardization.

IT – se Informations Teknologi

JCL – Se Job Control Language

Job Control Language (JCL): Scriptsprog kendt fra IBM OS /400, OS/390 og Z/OS til styring af batchkørsler og –jobs.

kolonne: En placering i en databasetabel, der indeholder en bestemt type data. Kolonnen er også den visuelle præsentation af et felt i et dataark, og i en MS Access-database er det forespørgslens designgitter eller filterets designgitter.

kombinationsboks: Et kontrolelement, der anvendes i en formular, der tildeler kombinerede funktioner til en liste eller tekstboks. Indtast en værdi, eller klik på kontrolelementet for at få vist en liste, og marker et element i listen.

KSA – se Københavns Stadsarkiv

Københavns Stadsarkiv (KSA): Institution i Københavns Kommune.

LAN – se Local Area Network.

LDB-fil: Når man åbner en MS Access-databasefil (.mdb) med delt adgang, oprettes der en locking information-fil (.ldb) med det samme filnavn i den samme mappe som databasefilen. I denne locking information-fil lagres computernavnet og sikkerhedsnavnet for hver delt bruger af databasen. Disse oplysninger bruges til at kontrollere samtidighed.

Local Area Network (LAN): Computernetværk i privat eje og med begrænset geografisk udstrækning.

makro: Handling eller sæt af handlinger, der kan bruges til automatisering af opgaver.

MDAC – se Microsoft Data Access Components.

MDB-fil: Fil indeholdende en MS Access database.

MDE-fil: MS Access-databasefil, hvor alle moduler er prækompileret, og al redigérbar kildekode er fjernet.

MDW-fil: Arbejdsgruppe informationsfil til MS Access applikation. Indeholder blandt andet brugere og brugergrupper.

Microsoft Access-database: En samling af data eller objekter, f.eks. tabeller, forespørgsler eller formularer, der henviser til specielle emner eller formål. Microsoft Jet-databaseprogrammet styrer dataene.

Microsoft Data Access Components (MDAC): Samling af teknologier fra Microsoft som giver udviklere mulighed for at implementerer UDA.

Microsoft Data Engine (MSDE): Databasekerne der er fuldt kompatibel med Microsoft SQL Server. MSDN skal installeres separat på hver maskine da det ikke er en standard del af Microsoft Office.

Microsoft Developers Network (MSDN): www.msdn.microsoft.com

Microsoft Jet: Microsoft databasekerne der blandt andet anvendes af MS Access-databasesystemet.

Microsoft SQL Server database: En database i Microsoft SQL Server, der blandt andet består af tabeller, visninger, indekser, gemte procedurer, funktioner og udløbere.

MIME – se Multipurpose Internet Mail Extensions.

Model: En beskrivelse af klasser, objekter, strukturer og adfærd i et problemområde.

MSDE – se Microsoft Data Engine.

MSDN – se Microsoft Developers Network.

Multipurpose Internet Mail Extension (MIME): Standard for internet mail, der er kendetegnet ved en enkel header der giver mulighed for håndtering af mail med mange forskelle former for indhold. Se også RFC 1521.

Network News Transfer Protocol (NNTP): NNTP har to formål

- Tillade nyheder (artikler) at flyde fra en maskine til en anden over en troværdig forbindelse
- Give brugere hvis computer ikke kan modtage nyheder mulighed for at ”fjernlæse” nyheder

Protokollen er kendt som grundlaget for nyhedsgrupper (newsgroups) på internettet. Se også RFC 977.

NNTP – se Network News Transfer Protocol.

Object Linking and Embedding (OLE): En mekanisme fra Microsoft som giver mulighed for at arbejde med dokumenter der indeholder elementer (”objekter”) fra flere applikationer.

Ordforklaring

Object Linking and Embedding Databases (OLE DB): COM interface fra Microsoft der giver adgang til tabel datakilder. OLE DB er nogle gange betegnet som objekt orienteret ODBC, men der er temmelig forkert da OLE DB tilbyder en meget stærkere funktionalitet og adgang til datakilder som ODBC ikke kan tilgå. Se eventuelt spiken "Hvad er ADO?".

Object: En helhed med identitet, tilstand, og adfærd

Objekt Orienteret Analyse (OOA): Analysefasen i objekt orienteret systemudvikling.

Objekt Oreinteret Design (OOD): Designfasen i objekt orienteret systemudvikling.

ODBC – se Open Database Connectivity.

OLE – se Object Linking and Embedding.

OLE DB – se Object Linking and Embedding Databases.

OOA – se Objekt Orienteret Analyse

OOD – se Objekt Orienteret Design

Open Database Connectivity (ODBC): Et API til databaseadgang, der er blevet ophøjet til at være formel CLI standard af ANSI. Udviklingen af ODBC er stoppet, da Microsoft vil have standarden afløst af OLE DB og ADO.

Problemområdet: Den del af omgivelserne der administreres, overvåges eller styres ved hjælp af et system.

RAD – se Rapid Application Development.

Rapid Application Development (RAD): 80'er begreb for applikationsudvikling med grafiske værktøjer, ofte i et mindre formelt (struktureret) forløb.

RDBMS – se Relational Database Management System.

RDS – se Remote Data Service.

Relational Database Management System (RDBMS): Samling af programmer til oprette og vedligeholde en relationel database.

Remote Data Service (RDS): Giver ADO mulighed for at tilgå fjerne data, for eksempel i et web miljø.

Remote Procedure Call (RPC): Teknologi til at skabe klient/server applikationer. Første gang beskrevet 1984 af Birrel og Nelson i artiklen "Implementing Remote Procedure Calls" udgivet i ACM Trans on Computer Systems vol. 2 (feb. 1984).

replika: En replika er et medlem af et replikasæt og kan synkroniseres med andre replikaer i sættet. Ændringer i data i en replikeret tabel i en replika sendes til og anvendes i de andre replikaer.

Rich Text Format (RTF): Flytbart dataformat, der i princippet burde kunne bruges til at flytte formaterede data mellem forskellige platforme, men for eksempel Microsoft overholder ikke standarden fuldstændigt, hvorfor problemer vil opstå.

RPC – se Remote Procedure Call.

RTF – se Rich Text Format.

SBA – se Sårbarheds analys.

Simple Mail Transfer Protocol (SMTP): En simpel ASCII protokol til simple beskeder over TCP port 25. Se også RFC 821.

SMTP – se Simple Mail Transfer Protocol.

SQL – se Structured Query Language.

Strengths, Weaknesses, Opportunities and Threats: Udvidet risikoanalyse der giver flere muligheder for at identificere trusler og handlinger til at imødegå dem.

Structured Query Language (SQL): Forespørgselsprog til relationelle databaser.

SWOT – se Strengths, Weaknesses, Opportunities and Threats

synkronisering: Opdatering af to medlemmer af et replikasæt ved at udveksle alle opdaterede poster og objekter i hvert medlem. To replikasætmedlemmer synkroniseres, når ændringerne i hvert medlem er anvendt i det andet.

tilladelser: Et sæt attributter, der angiver, hvilken type adgang en bruger har til data eller objekter i en database.

Sårbarheds analys: Særdeles grundig svensk risikoanalyse hvor alle forhold et projekt kan komme ud for ”rates” og der udregnes en samlet risikofaktor.

UDA – se Universal Data Access.

udelt adgang: En adgangstype til data i en database, der deles i et netværk. Når en database åbnes med udelt adgang, forhindres andre i at åbne databasen.

UML – se Unified Modeling Language.

Unified Modeling Language (UML): En meget udbredt notationsform der adskiller proces og notation.

Universal Data Access (UDA): En samling af teknologier fra Microsoft til at tilgå data uafhængigt af hvor data er opbevaret. Typisk er der tale om anvendelse af flere COM interfaces.

Ordforklaring

usecase: En serie af handlinger der udføres af en aktør inden for et system for at opnå et konkret resultat.

VBA – se Visual Basic for Applications.

Visual Basic for Applications (VBA): En version af Microsoft Visual Basic, der bruges til at programmere Windows-programmer.

WAN – se Wide Area Network.

Wide Area Network (WAN): Netværk med større geografisk udstrækning, ofte et land eller et kontinent.

Windows Management Instrumentation (WMI): Microsofts implementering af den web-baserede managementplatform Web-Based Enterprise Management (WBEM) fra Distributed Management Task Force (DMTF).

Grundlæggende består WMI af tre ting:

- Common Information Model Object Manager (CIMOM), i Windows kaldet WMI Service.
- Common Information Model repository (CIM), i Windows kaldet WMI repository.
- WMI Provider

Se mere på MSDN og www.dmtf.org.

Windows Scripting Host (WSH): Administrativt miljø fra Microsoft der giver mulighed for hosting scripts. Det enkelte script kan være skrevet i forskellige sprog som for eksempel Visual Basic Scripting Edition (VBScript), JScript eller Perl.

WMI – se Windows Management Instrumentation.

WSH – se Windows Scripting Host.

XML – se EXtensible Markup Language

XP – se EXtreme Programming.