

Datavetenskap

Ola Bull och Joakim Norlinder

**Utveckling av databasapplikation för
packningssatser**

Examensarbete, C-nivå

2004:28

Utveckling av databasapplikation för packningssatser

Ola Bull och Joakim Norlinder

Denna rapport är skriven som en del av det arbete som krävs för att erhålla en kandidatexamen i datavetenskap. Det material som inte är mitt eget i denna rapport har tydligt identifierats och inget material har inkluderats som tidigare använts för erhållande av annan examen.

Ola Bull

Joakim Norlinder

Godkänd, 040604

Handledare: Mari Göransson

Examinator: Martin Blom

Sammanfattning

Många företag har idag problem med redundant och inkonsistent information i form av papperskopior insatta i pärmar. Momentum är ett sådant företag. På uppdrag av dem kom detta arbete att ta sin början. Momentum önskade få hjälp med att skapa ett verktyg för att strukturera deras packningssatser. Vi inledde således ett samarbete med Momentum och beslut fattades att skapa någon form av central databas för lagring av packningssatser. Målet med arbetet var att skapa en användarvänlig produkt för hanteringen av information rörande packningssatser. Denna produkt skulle kunna användas från Momentums samtliga kontor. Resultatet av arbetet blev en databas med ett tillhörande webbaserat användargränssnitt. Uppsatsen behandlar de delar som produkten består av, samt en processbeskrivning av arbetsgången.

Development of a Database Application for Gasket Kits

Abstract

Redundant and inconsistent data in form of carbon copies in folders is a problem that many companies today struggle with. Momentum is one of these companies. They informed us of their problem and we started our cooperation. The task was to create some sort of central database for their gasket kits. The main goal was to create a user friendly system that handles information concerning their products. Momentum wanted this system to be available at all their offices. The result of this project was a database with a graphical user interface (GUI). This paper describes the different parts of the system and gives an overview of the creation process.

Innehållsförteckning

1	Inledning	1
2	Bakgrund	3
2.1	Företagspresentation.....	3
2.2	Problemställning.....	3
2.3	Syfte och mål.....	4
2.4	Kravspecifikation	4
2.5	Avgränsningar	6
2.6	Sammanfattning	6
3	Databas.....	8
3.1	Olika typer av databassystem.....	8
3.2	Olika typer av databashanteringssystem.....	9
3.2.1	Lotus Notes	
3.2.2	MySQL	
3.2.3	Access	
3.2.4	MSSQL	
3.3	Val av databashanteringssystem.....	10
3.4	Sammanfattning	11
4	Säkerhet	12
4.1	Överblick.....	12
4.1.1	Exponering av känslig information	
4.1.2	Förlorad eller förstörd information	
4.1.3	Manipulation av information	
4.1.4	Hindrande av access	
4.1.5	Felaktig mjukvara	
4.2	Ställda krav	14
4.3	Sammanfattning	14
5	Användargränssnitt.....	15
5.1	Olika typer av användargränssnitt.....	15
5.1.1	Textbaserade användargränssnitt	
5.1.2	Grafiska användargränssnitt	
5.1.3	HTML	
5.1.4	JavaScript	

5.1.5	PHP	
5.1.6	ASP	
5.1.7	JSP	
5.2	Val av skriptspråk.....	17
5.3	Sammanfattning	17
6	Testning.....	18
6.1	Anledning till testning	18
6.2	Olika typer av testmetoder	19
6.2.1	Black – box testning	
6.2.2	White – box testning	
6.2.3	Integrationstestning	
6.2.4	Acceptanstestning	
6.3	Val av testmetoder.....	20
6.4	Sammanfattning	20
7	Skapande av produkten.....	21
7.1	Prototyp.....	21
7.2	Databasdesign	22
7.3	Säkerhet	23
7.4	Användargränssnitt.....	24
7.5	Implementation av produkten	26
7.5.1	Databasen	
7.5.2	Gränssnittet	
7.6	Testning	30
7.7	Sammanfattning	31
8	Slutgiltiga kommentarer.....	32
	Referenser	33
A	Bilaga – Kravspecifikation	34
A.1	Inledning	37
A.2	Systembeskrivning	37
A.2.1	Befintlig system	
A.2.2	Planerat system	
A.2.3	Säkerhetsaspekter	
A.3	Omfattning	37
A.3.1	Utrustning	
A.3.2	Övriga uppgifter	
A.4	Krav..	38
A.4.1	Generella krav	
A.4.2	Specifika krav	
A.4.3	Teknisk miljö	

B	Bilaga – Användarmanual.....	39
B.1	Översikt.....	43
B.2	Att söka i databasen.....	44
	B.2.1 Märke/ Modell	
	B.2.2 Kombinerad sökning	
	B.2.3 Fritextsökning	
	B.2.4 Sökresultat	
B.3	Att avläsa statistik	48
	B.3.1 Statistik för kund	
	B.3.2 Statistik för packningssats	
B.4	Att lägga till packningssats	50
	B.4.1 Koppla ritning till artikel	
	B.4.2 Lägga till märke/modell	
	B.4.3 Skapa ny packningssats	
B.5	Att ändra eller ta bort.....	53
	B.5.1 Ändra i packningssats	
	B.5.2 Radera packningssats	
	B.5.3 Ändra i en artikel	
C	Bilaga – Databasdesign.....	57
C.1	Användare	58
	C.1.1 Användare	
C.2	Momentum	58
	C.2.1 Anställd	
	C.2.2 Artikel	
	C.2.3 Historik	
	C.2.4 Kund	
	C.2.5 Leverantör	
	C.2.6 Modell	
	C.2.7 Märke	
	C.2.8 Packningssats	
	C.2.9 Ritning	
	C.2.10 Uppsättning	
D	Bilaga – Testfall	64
D.1	Tesfall - Inloggning	65
	D.1.1 Testfall 1 – Korrekt inloggning	
	D.1.2 Testfall 2 – Felaktig inloggning	
D.2	Testfall - Sökning	66
	D.2.1 Testfall1 – Sökning på märke/modell	
	D.2.2 Testfall 2 – Kombinerad sökning	
	D.2.3 Testfall 3 – Fritextsökning	
	D.2.4 Testfall 4 – Ny kund	
	D.2.5 Testfall 5 – Lägg till statistik	
	D.2.6 Testfall 6 – Skapa plocksedel.	
D.3	Testfall – Skapa ny packningssats.....	68
	D.3.1 Testfall 1 – Lägg till packningssats	
	D.3.2 Testfall 2 – Lägg till märke/modell	
	D.3.3 Testfall 3 – Lägg till ytterligare artiklar	
D.4	Testfall – Ändra/ta bort packningssats	70
	D.4.1 Testfall 1 – Lägg till ytterligare artikel till packningssats	
	D.4.2 Testfall 2 – Ta bort packningssats	

D.4.3	Testfall 3 – Ändra artikel	
D.5	Testfall – Administrera	71
D.5.1	Testfall 1 – Lägg till användare	
D.5.2	Testfall 2 – Ta bort användare	
D.5.3	Testfall 3 – Ändra användaruppgifter	

Figurförteckning

Figur 2-1 - Användarfallsdiagram	5
Figur 3-1 - Exempel på trädstruktur	8
Figur 3-2 - Koppling relationsdatabas	9
Figur 4-1 - Säkerhetsaspekter.....	12
Figur 6-1 - Kostnaden för att åtgärda fel	18
Figur 6-2 - Black - box testning	19
Figur 7-1 - E/R diagram Användare	22
Figur 7-2 - E/R diagram Momentum.....	23
Figur 7-3 - Skärmdump från skapa packningssats	25
Figur 7-4 - Skärmdump sökresultat	25
Figur 7-5 - Filinkluderingsdiagram över gränssnittet.....	26
Figur 7-6 - Klient/Server arkitektur.....	27
Figur 7-7 - Anrop med GET.....	28

1 Inledning

På många företag är redundant och inkonsistent data ett problem. Företagen spar sin information på papperskopior som sätts in i pärmar. Momentum lider av det här problemet och därmed låg grunden för det här uppdraget. Momentum önskade få hjälp med att skapa ett verktyg för att strukturera deras packningssatser. Vi inledde därför ett samarbete med målet att skapa någon form av central databas för lagring av packningssatser. En packningssats är ett antal tätningar och packningar som är grupperade som en sammansatt artikel. Uppdraget Momentum gav oss var att skapa en användarvänlig produkt för hanteringen av information rörande packningssatser. Resultatet av arbetet blev en databas med tillhörande webbaserat användargränssnitt. Uppsatsen behandlar de delar som produkten består av, samt en processbeskrivning av arbetsgången.

Kapitel två behandlar bakgrunden till projektet samt de syften och mål som produkten och uppsatsen har. Här finns en kortfattad kravspecifikation som beskriver de delar som produkten skall innehålla.

I uppsatsens tredje kapitel finns en jämförelse mellan olika typer av databashanteringssystem och databaser. Några av databashanteringssystemen som behandlas är MySQL, MSSQL och Lotus Notes. Vi har till detta projekt valt att använda MySQL för produkten och motiveringen till detta val redogörs för i slutet av kapitel tre.

Kapitel fyra behandlar säkerhet och olika säkerhetshot som bör beaktas när en webbaserad lösning skapas. Exempel på säkerhetshot är: hindrande av access, exponering av känslig information samt förlust av data. Vidare beskrivs de krav på säkerhet som Momentum har på produkten.

Det femte kapitlet behandlar olika typer av användargränssnitt samt några av de verktyg som finns för att skapa gränssnitt för Internet t.ex. PHP, JSP och ASP. En jämförelse mellan de olika verktygen, finns dessutom med för att kunna motivera valet av verktyg, vilket blev PHP.

Då produkten måste testas har vi valt att ägna kapitel sex åt detta ämne. Kapitlet behandlar några av de testmetoder som finns t.ex. White – box testning och acceptanstestning. Vidare diskuteras vikten av testning för att skapa en produkt av hög kvalitet.

I kapitel sju beskrivs den process som använts för att färdigställa produkten. Kapitlet inleds med en genomgång av de olika sorters prototyper som vanligtvis används t.ex. T - prototyp,

bred prototyp samt djup prototyp. Dessutom ges en beskrivning av den prototyp som framställdes som ett led i skapandet av produkten. Det ges en inblick i designen av databasen. Vidare beskrivs hur gränssnittet skapades utifrån prototypen samt vilka riktlinjer vi valde att följa under utvecklingsprocessen.

I sista kapitlet, dvs. kapitel åtta Slutgiltiga kommentarer sammanfattas arbetet och en diskussion förs över produkten.

2 Bakgrund

I kapitel två kommer vi att beskriva de problemområden som föranledde denna uppsats och produkt. Syftet med uppsatsen och produkten kommer att klargöras samt de mål som eftersträvades.

2.1 Företagspresentation

1997 gick de fyra företagen Sesemic, Järn & Maskin, Kulan Väst och Kulan Syd samman och bildade Momentum, vilket betyder ”kraft satt i rörelse”. Momentum är idag en av Sveriges ledande leverantörer av industrikomponenter med verksamhet på cirka 20 orter runt om i landet. Sortimentet omfattar produkter inom affärsområden som lager, tätning, transmission och automation. Totalt omfattar produktsortimentet ca: 100 000 artiklar. Inom respektive område samarbetar de med tillverkare såsom SKF, Renold, Optibelt, Simrit och SMC.

2.2 Problemställning

Ett problem som finns på många företag är den ständigt växande högen av papper och pärmar. Uppdateringen av dokument blir lidande; nya papper sätts in men gamla tas inte bort. Olika avdelningar ligger inte i fas med varandra. Varje lokalkontor har sin egen uppsättning med pärmar och papper. Detta leder i många fall till redundant och inkonsistent information, något som kan vara förvirrande för både kunder och personal eftersom samma artikel kan ha olika benämning. Momentum är endast ett av många företag som brottas med denna problematik och det är på uppdrag av dem som detta projekt har arbetats fram.

På Momentum finns idag 23 lokalkontor runt om i landet. Problemet för Momentum är att samma packningssats finns på flera olika kontor med olika artikelnummer och benämning. Följden av detta är att det saknas kontroll på vad som egentligen finns på de olika kontoren. I och med att ingen vet vad som finns skapas nya packningssatser trots att de kanske redan finns.

2.3 Syfte och mål

Momentum har insett att något måste göras åt problemen med de växande högarna med papper och pärmar, samt den inkonsistens och redundans som finns. Målet är att skapa ett användarvänligt arbetsredskap som präglas av enhetlig benämning samt gemensamt artikelnummer för alla kontor. Momentum har kommit fram till att bästa lösningen vore att skapa någon form av central databas där informationen kan lagras, samt information kan hämtas och uppdateras.

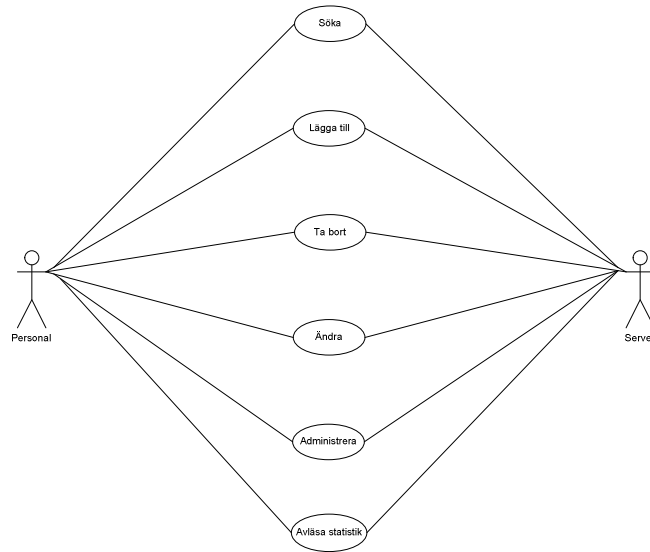
Målet med produkten är att skapa en informationsdatabas som i första hand vänder sig till Momentums anställda och följer kravspecifikationen, vilken tas upp i nästa kapitel. Databasen skall förberedas för att i framtiden även kunna användas som informationskälla åt företagets kunder. Informationsdatabasen skall kunna administreras och användas från ett grafiskt gränssnitt.

Den här uppsatsen fokuserar på två delar. Kapitel tre, fyra, fem och sex kommer i stora drag behandla de olika delar produkten består av. Kapitel sju kommer att behandla de steg som togs under framställandet av produkten. Målet med uppsatsen är att göra en processbeskrivning kring skapandet av produkten. Syftet är att dokumentera vilka delar produkten består av. En användarmanual kommer även att skrivas och bifogas i bilaga B.

2.4 Kravspecifikation

Det här avsnittet kommer att behandla de krav som kunden har på produkten. I bilaga A finns en utökad kravspecifikation som behandlar hela projektet. Denna kravspecifikation gäller både våra krav på kunden samt kundens krav på oss.

För att identifiera kundens önskemål med produkten har vi valt att använda användarfallsdiagram, se Figur 2-1. Ett användarfallsdiagram används för att ge kunden en översiktlig bild av systemet samt hur dess delar samverkar. Dessutom gav användarfallsdiagram oss en möjlighet att identifiera vilka aktörer som krävdes.



Figur 2-1 - Användarfallsdiagram

Enligt kravspecifikationen, se bilaga A, klargjordes följande krav på produkten.

Databasen skall innehålla följande information:

1. Artikelnummer
2. Leverantör
3. Benämning
4. Benämning ersättning (Momentums egen benämning)
5. Material
6. Antal
7. Datum upplagt i databasen
8. Vem som mätt upp packningssatsen
9. Packningssatsnummer
10. Benämning packningssats
11. Kund
12. Kundnummer
13. Ritningsnummer
14. Modell
15. Märke
16. Kommentar för artikel i packningssats

Databasen skall vara i princip underhållsfri dvs. nödvändiga operationer skall kunna skötas från gränssnittet. Nödvändiga operationer är:

1. Söka efter packningssats
2. Lägga till packningssats
3. Ändra i packningssats
4. Ta bort packningssats
5. Ändra i artikel
6. Kontrollera statistik
7. Lägga till användare
8. Ta bort användare
9. Administrera användarkonto (byta lösenord, namn och behörighetsgrad)

Dessa operationer är framtagna med hjälp av Figur 2-1.

Företaget ställer vissa krav på säkerheten. Det ska finnas tre olika behörighetsnivåer. En för sökning, en för att lägga till, ändra eller för att ta bort packningssatser samt en för att administrera användarkonton. För att hålla databasen så aktuell som möjligt har vi lagt till en extra funktion som rensar bort inaktuella artiklar dvs. artiklar som inte ingår i någon packningssats.

2.5 Avgränsningar

Detta arbete kommer enbart att bestå av design och implementation av databasen samt att skapa gränssnittet. Arbetet med att infoga information i databasen kommer Momentum själva att ombesörja. I kapitel sex – Testning, kommer vi att begränsa oss till att gå igenom de olika typer av testmetoder som ingår i kursen *Software Engineering (DAVC19)* Karlstads Universitet. Vi har valt att endast anpassa produktens gränssnitt efter Internet Explorer då det är den enda webbläsare som används på Momentum. Eftersom produkten kommer att finnas på Momentums intranät har vi inte tagit hänsyn till attacker utifrån utan det ombesörjer Momentum själva.

2.6 Sammanfattning

Momentum har idag likvärdig information lagrad i pärmar på flera olika kontor. De har upplevt det som ett problem och vill nu ha allt lagrat på ett och samma ställe. För att lösa detta

har Momentum bestämt sig för att upprätta en databas. De ansåg detta som ett projekt lämpat för ett examensarbete. I arbetet ingår design av databasen, skapande av ett användargränssnitt och en användarmanual.

3 Databas

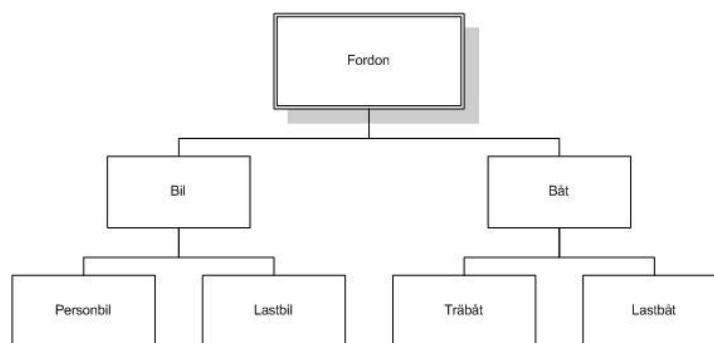
I dagligt tal när det pratas om databaser syftas det egentligen på databassystem (DBS). Ett databassystem består av databashanteringssystem (DBHS) och databas (DB). Ett databashanteringssystem är programvaran som används för att manipulera data som finns i databasen. En databas är en samling av relaterad data som representerar en del av verkligheten.

I det här kapitlet ger vi en introduktion till olika typer av databaser och databashanteringssystem. Vi kommer att beskriva de olika alternativ vi valde mellan samt skillnaden mellan dessa. Vilken typ av databassystem lämpar sig bäst för vårt arbete? Varför? Vad har vi valt?

3.1 Olika typer av databassystem

Den enklaste typen av databas är en ensam tabell och kallas *platt databas*. I dagligt tal kallas en sådan typ av databas för ett register. Sådana register kan skapas i exempelvis *Word* och *Excel*.

En annan typ är hierarkiska databaser, vilka bygger på en trädstruktur, se Figur 3-1 för ett hierarkiskt exempel.

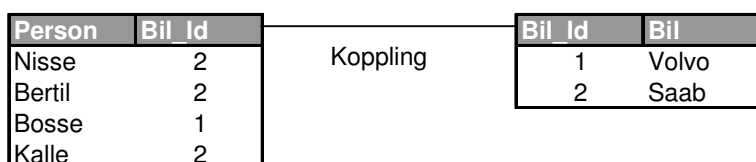


Figur 3-1 - Exempel på trädstruktur

Datorns filsystem är ytterligare ett exempel på en hierarkisk databas.

Den databas som är vanligast idag är relationsdatabasen. En relationsdatabas består av flera tabeller som står i relation till varandra genom gemensamma kopplingar, så kallade relationer,

se Figur 3-2. Dessa kopplingar gör att redundans undviks då data endast behöver lagras på ett ställe och inte på flera som var fallet vid platta och hierarkiska databaser. Dessutom är det till skillnad från de tidigare nämnda databaserna enkelt att söka, skriva och sortera i en relationsdatabas. Språket som används för att kommunicera med en relationsdatabas heter Standard Query Language (SQL).



Figur 3-2 - Koppling relationsdatabas

3.2 Olika typer av databashanteringssystem

Idag finns en mängd olika databashanteringssystem. Vi har valt att presentera några av de vanligaste här nedanför.

3.2.1 Lotus Notes

Lotus Notes är ett distribuerat, dokumentbaserat databasapplikationssystem och bygger på en hierarkisk databas. Dokumenten ligger lagrade enligt trädstruktur. Om en användare söker i Lotus Notes sker en fritextsökning i varje dokument och de dokument som innehöll sökkriteriet presenteras. Nackdelen med Lotus Notes är att det finns risk för inkonsistent och redundant data. Ändrar två användare i samma dokument kommer två versioner av dokumentet sparas.

3.2.2 MySQL

MySQL [7] är en relationsdatabas som baseras på så kallad *öppen källkod*, vilket betyder att vem som helst får använda produktens källkod och göra ändringar i den samma utan någon kostnad. MySQL utvecklas av företaget MySQL AB som ligger i Uppsala. Det här databashanteringssystemet är ett snabbt och driftsäkert system men saknar vissa avancerade funktioner som andra databashanterare har, t.ex. integritetsregler med främmandenycklar [1]. MySQL finns tillgänglig för de flesta operativsystem t.ex. Windows, Linux och Solaris. Några fördelar:

- Gratis
- Spridd – Många användare och installationer, därmed mycket dokumentation tillgänglig.
- Snabb
- Enkel att koppla till Internet
- Stabil, driftsäker
- Enkel att administrera och använda

Med hjälp av SQL ställs frågor till databasen som i sin tur returnerar resultatet av frågan.

3.2.3 Access

Access[8] är en relationsdatabas som är distribuerad av Microsoft. Då Access har ett grafiskt gränssnitt är det relativt enkelt och lättöverskådligt att arbeta med. Access kräver heller inga programmeringskunskaper av användaren, applikationen genererar koden utifrån de val användaren gjort. Språkets som används för att ställa frågor mot databasen kallas MS Jet SQL och är en variant av SQL.

3.2.4 MSSQL

Microsoft SQL Server (MSSQL) säljs av Microsoft och baseras även den på SQL. MSSQL fungerar ungefär som MySQL men med en skillnad, MSSQL kräver en licens.

3.3 Val av databashanteringssystem

Momentum hade ett önskemål att använda Lotus Notes som databas då företaget redan använder denna programvara. Eftersom databasen skulle förberedas för att i framtiden kunna användas av företagets kunder så var vi tvungna att titta på en webbaserad lösning. Lotus Notes har möjlighet att kopplas mot webben men denna är inte optimal. Ett annat problem med Lotus Notes är att den inte är en relationsdatabas utan är en hierarkisk databas, vilket skulle utgöra en risk för redundans. Dessa egenskaper gjorde att Lotus Notes föll bort ur resonemanget. Kvar blev MySQL, Access och MSSQL.

Access lämpar sig väl till mindre databaser, med det menas databaser som har få användare. Momentum har idag omkring 30 000 packningssatser, 120 000 artiklar samt 225 anställda som skall ha tillgång till databasen. Detta utesluter Access.

Valet stod då mellan MySQL och MSSQL. Vi har tidigare läst kursen databasteknik på Karlstads Universitet, där en del av kursen behandlade MySQL. MSSQL skulle säkert lämpa sig alldeles utmärkt för vår produkt, men eftersom vi känner till MySQL så blev det vårt val.

3.4 Sammanfattning

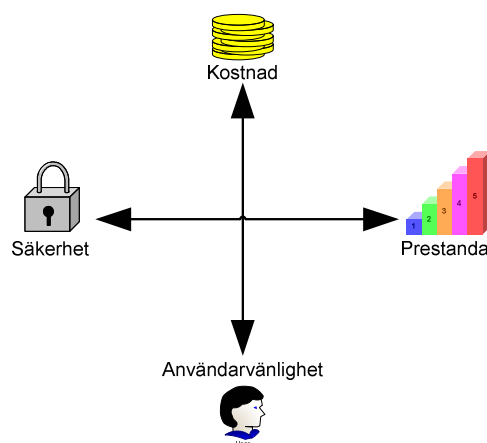
När det pratas om databaser menas vanligen databassystem (DBS). Ett databassystem består av databashanteringssystem (DBHS) samt en databas (DB). Det finns idag olika typer av databaser, varav den vanligaste är relationsdatabasen. Det finns även en mängd olika databashanteringssystem bland annat Access, MSSQL, MySQL samt Lotus Notes. Till det här arbetet valdes MySQL då vi tidigare hade erfarenhet av det samt att produkten krävde en webbaserad lösning.

4 Säkerhet

Säkerhetsaspekterna är något som är av betydande vikt för alla företag idag. I det här kapitlet kommer några av de aspekter som har betydelse för datasäkerheten i allmänhet att belysas. Hur säkerheten bör se ut beror på några olika faktorer, såsom hur viktig informationen är som finns lagrad. Kapitlet kommer att besvara frågorna: Vilka olika säkerhetshot finns det? Vad kan göras för att undvika dem? Vilka metoder har valts för att skydda produkten? Uppgifterna i detta kapitel är hämtat från *PHP and MySQL Web Development* [2].

4.1 Överblick

Första steget kring säkerheten av en produkt är att klargöra hur viktig informationen är. Hur långt skall utvecklaren sträcka sig för att skydda en produkt? Det är frestande att tro att högsta möjliga säkerhet krävs för en produkt, men att göra en produkt säker kostar både tid och pengar. Dessutom riskeras att prestandan blir lidande. Det finns ett förhållande mellan användarvänlighet, prestanda, kostnad och säkerhet se Figur 4-1. De här punkterna kan ses som konkurrerande mål och avkall måste oftast göras på någon för att öka en annan.



Figur 4-1 - Säkerhetsaspekter

Säkerhetsaspekterna skiljer sig gravt mellan olika typer av användare. En hemanvändare har inte samma behov av säkerhet som exempelvis en bank. Dessutom bör hänsyn tas till hur viktig informationen är för en obehörig användare.

Det finns ett flertal olika säkerhetshot idag. Några av dessa är:

- Exponering av känslig information
- Förlorad eller förstörd information
- Manipulation av information
- Hindrande av access
- Felaktig mjukvara

4.1.1 Exponering av känslig information

Information som finns lagrad och skickas mellan olika datorer kan vara av känslig art, t.ex. kreditkortsnummer, lösenord och företagsinformation. Säkerhetsaspekterna är olika beroende på om mjukvaran befinner sig i ett lokalt nätverk, ett intranät eller på Internet.

En webbserver är en olämplig plats för att lagra känslig information eftersom webbservern kan nås av vem som helst när som helst. För att minska risken att känslig information läcker ut från en webbserver bör antalet användare som kan komma åt den specifika informationen minimeras. Dessutom bör antalet vägar att komma åt informationen minimeras, dvs. åtkomst till känslig information skall endast kunna komma åt genom en på förhand bestämd väg. Med detta menar vi att information inte skall kunna nås utan att först gå via en inloggningssida.

4.1.2 Förlorad eller förstörd information

Att förlora information eller att få den förstörd kan vara mer kostsamt än att få den exponerad. En databas kan ha fyllts med information under år av användning. Om all den informationen går förlorad skulle kostnaden i både tid och pengar bli stor vid återskapandet, om det överhuvudtaget är möjligt att återskapa allt.

Enda sättet att försäkra sig om att information inte går förlorad är att skapa en säkerhetskopia. I och med det finns alltid informationen tillgänglig för att återskapa allt. Sannolikheten att både säkerhetskopian och originaldata ska förstöras samtidigt torde ses som nästan obefintlig.

4.1.3 Manipulation av information

Även om förlust av information kan vara förödande så kan det vara värre med obehörig manipulation av information. Om någon manipulerar information i ditt system kan det ta lång tid innan detta upptäcks. Om t.ex. ett företag som säljer varor på nätet får sina

försäljningssiffror ändrade kan det få förödande konsekvenser när budgeten inte stämmer. Det går att använda digitala signaturer eller kryptera data för att hindra manipulation när information skickas mellan datorer.

4.1.4 Hindrande av access

Hindrande av access (Denial of Service, DoS) är en av de svåraste hot att skydda sig mot. Hindrande av access sker när någon vidtar åtgärder för att hindra eller göra det omöjligt för vanliga användare att utnyttja en service, eller skapa fördröjningar i ett tidskritiskt system. Detta kan förhindras genom att stänga de portar som vanligtvis används vid DoS attacker.

4.1.5 Felaktig mjukvara

Det är möjligt att den mjukvara som du köpt, använder eller skrivit har allvarliga inbyggda fel. Då utvecklingstiden för webbprojekt i regel är väldigt kort ökar risken för att fel uppstår. Ett företag som är beroende av sin mjukvara blir väldigt utsatt om denna innehåller en massa fel. Fel i mjukvara kan leda till alla möjliga typer av oväntade fel t.ex. säkerhetshål, ekonomiska förluster och dålig service till kunder. De felen kan bero på dåligt specificerad programvara, felaktiga antaganden av programmeraren eller dåligt genomförd testning. Det bästa sättet att undvika fel i mjukvaran är att testa grundligt.

4.2 Ställda krav

Momentums enda säkerhetskrav på produkten var att det skulle finnas en inloggningssida med tre behörighetsnivåer. Eftersom produkten är tänkt att ligga på Momentums intranät så behövs inte hänsyn tas till vare sig DoS, exponering av känslig information eller manipulering av data. Däremot har tester gjorts för att försöka förhindra felaktig programvara, se kapitel 7.6 Testning.

4.3 Sammanfattning

Nivån på säkerheten styrs av hur viktig informationen som skall skyddas är. Det finns tre kriterier för att balansera säkerhet mot produktens övriga krav. Kostnad, prestanda samt användarvänlighet.

Det finns ett antal olika säkerhetshot, vi har diskuterat kring följande: exponering av känslig information, förlorad eller förstörd information, hindrande av access samt felaktig mjukvara. Vår produkt krävde endast inloggningssida med tre olika behörighetsnivåer.

5 Användargränssnitt

I kapitel fem kommer vi att diskutera allmänt kring användargränssnitt, dvs. den del av produkten som presenteras på skärmen för användaren. Vi kommer även att diskutera kring de verktyg som finns för att bygga ett gränssnitt. Med verktyg avses olika typer av språk med vilka ett gränssnitt kan skapas. Vad finns det för olika typer av användargränssnitt? Vilka olika typer av verktyg finns det? Vilka kriterier skall vårt användargränssnitt uppfylla? Vilket väljer vi att använda och varför?

5.1 Olika typer av användargränssnitt

5.1.1 Textbaserade användargränssnitt

Den här typen av gränssnitt baserar sig på text. Dessa gränssnitt är idag ganska ovanliga, främst för att de ger en sämre överblick och därmed svårare att använda. Vi har valt att inte använda detta eftersom ett textbaserat gränssnitt blir alldeles för komplext för en normalanvändare. Detta gör att det blir svårt att förbereda produkten för framtida kundbruk.

5.1.2 Grafiska användargränssnitt

I dag är så gott som alla användargränssnitt grafiska. Fördelen med dessa är att de är lättanvända och inte kräver så mycket förkunskaper av användaren, dessutom ger de en bättre överblick än textbaserade. Då databasen är tänkt att i framtiden kunna användas av Momentums kunder anser vi att den bästa lösningen vore att skapa ett webbaserat användargränssnitt. Fördelen med den typen av gränssnitt är att de är plattformsoberoende dvs. de fungerar på alla typer av grafiska webbläsare oberoende av operativsystem eller dator. De vanligaste typerna av verktyg för att skapa ett gränssnitt för webben är:

- HTML
- JavaScript
- PHP
- ASP
- JSP

Dessa kommer att beskrivas kortfattat i efterföljande avsnitt.

5.1.3 HTML

HTML betyder *Hypertext Meta Language*. Webbsidor som skrivs med enbart HTML blir statiska dvs. de presenteras alltid på samma sätt. T.ex. om det finns en klocka som har kodats för att visa 13:45 så kommer den alltid att visa den tiden. HTML är också klientbaserat vilket innebär att webbsidan laddas ner till klientdatorn (användaren) innan den kan visas.

5.1.4 JavaScript

JavaScript är precis som HTML klientbaserat. Det är ett programmeringsspråk som lämpar sig för webbprogrammering. JavaScript kombineras tillsammans med HTML för att skapa avancerade och dynamiska webbsidor. Med dynamiska menas att webbsidan ändrar utseende beroende på vad användaren väljer att göra på sidan.

5.1.5 PHP

PHP står för *PHP: Hypertext Preprocessor* och är ett serverbaserat HTML-inbakat skriftspråk. Serverbaserat betyder att all PHP kod exekverar på serversidan och inte på klientsidan. PHP är så kallad *öppen källkod*. Med detta menas att vem som helst får använda produktens källkod och göra ändringar i den samma utan någon kostnad. Att det är HTML-inbakat betyder att den PHP kod som finns på serversidan tolkas som HTML av användarens webbläsare.

PHP kan användas för att göra webbsidor dynamiska.

5.1.6 ASP

ASP står för *Active Server Pages* och är en teknologi utvecklad av Microsoft för att baka in programmeringskod i HTML koden. Fördelen med detta är att webbsidan kan göras dynamisk. ASP fungerar i stora drag ungefär som PHP men är inte *öppen källkod*. För att använda ASP krävs en licens.

5.1.7 JSP

Betyder Java Server Pages och fungerar i stort på samma sätt som ASP och PHP. JSP använder programspråket Java. Nackdelen med JSP är att den inte stöds av så många webbhotell.

5.2 Val av skriptspråk

Det första kriteriet som ställs på gränssnittet är att det skall kunna vara dynamiskt, vilket utesluter enbart HTML som möjlig kandidat. Då gränssnittet även skall kopplas till en databasserver så krävs en serverbaserad lösning. Detta utesluter även en applikation skriven helt i JavaScript. Eftersom vi har valt att använda en MySQL databas så är PHP den bästa lösningen, då dessa är anpassade att användas tillsammans. Dessutom är PHP gratis att använda. Det finns troligtvis andra lösningar som är lika bra, men den här lösningen passade oss och Momentum bra.

5.3 Sammanfattning

I det här kapitlet har vi behandlat olika typer av användargränssnitt. Det finns två huvudsakliga typer, textbaserade och grafiska. Vi har valt att använda ett grafiskt användargränssnitt. Vårt gränssnitt måste uppfylla två kriterier: det ska kunna presenteras dynamiskt samt att det skall kopplas mot en databas. Det finns idag ett flertal olika verktyg för att bygga webbaserade gränssnitt. PHP tillsammans med HTML och javascript har valts för att bygga gränssnittet.

6 Testning

I samband med framställandet av system är det viktigt att utföra testning. Detta för att minimera riskerna för fel.

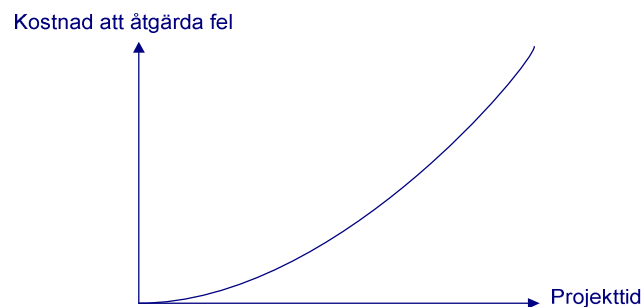
I kapitel sex kommer vi att diskutera kring olika typer av testmetoder och följderna av dessa. Varför skall testning användas? Vad finns det för olika testmetoder? Vad ger testningen för garantier? Vilka testmetoder kommer vi att använda?

6.1 Anledning till testning

Anledningen till att testning förekommer är främst för att spara tid och pengar. Det är enklare att rätta till ett fel om det upptäcks tidigare i utvecklingsprocessen. Testning har följande fördelar:

- Hitta fel snabbt
- Upprätthålla kvaliteten på en produkt
- Verifiera att produkten uppfyller kundens krav
- Spara tid och pengar

Här nedan presenteras ett diagram som påvisar hur kostnaden ökar exponentiellt över tiden. Detta visar hur viktigt det är att hitta fel tidigt i utvecklingsprocessen. Diagrammet är hämtat ur boken *Extreme Programming Explained* [3]



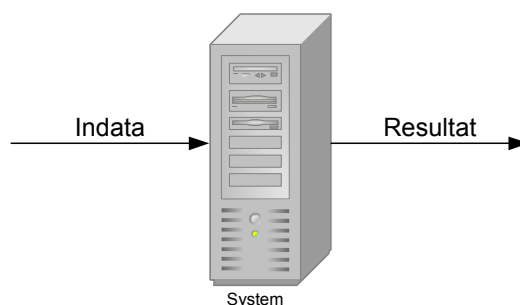
Figur 6-1 - Kostnaden för att åtgärda fel

Det faktum att ett program genomgår en mängd tester är ingen garanti för att programmet är felfritt. Däremot har risken för fel i programvaran minskat avsevärt.

6.2 Olika typer av testmetoder

6.2.1 Black – box testning

Principen för Black – box testning är att implementationsdetaljerna är oväsentliga för testet. Testfallen baseras på de krav som systemet har. Black – box testning koncentrerar sig på systemets externt synliga delar. Testpersonen (den person som utför testningen) matar in förutbestämda värden och erhåller ett resultat från systemet. Detta resultat jämförs mot ett förväntat resultat och kan på så vis kontrollera systemets pålitlighet.



Figur 6-2 - Black - box testning

6.2.2 White – box testning

Principerna för White – box testning är att testpersonen har kännedom om implementationsdetaljerna och kan på så sätt testa specifika delar av systemet. Testfallen baseras på programmets struktur.

6.2.3 Integrationstestning

Principen för Integrationstestning är att en del skapad för systemet först testas separat för att sedan läggas till systemet och testas igen. På så sätt minskar risken för att felaktiga delar implementeras i systemet vilket gör systemet mer driftsäkert.

6.2.4 Acceptanstestning

Den sista delen av testningen är acceptanstestning. Detta innebär att systemet som skapats skall integreras hos kunden och att denna testar systemet för att se om det motsvarar dennes förväntningar.

6.3 Val av testmetoder

Till denna produkt har samtliga testmetoder nämnda i texten ovan valts. Black – box testning krävs för att kontrollera att resultat till en inmatning blir riktigt utförd, medan White – box testning behövs för att testa specifika delar såsom SQL – frågor och specifika funktioner. Integrationstestning krävs för att kontrollera att systemets olika delar fungerar bra tillsammans. Acceptanstest är viktigt då det är där kunden testar och godkänner produkten.

6.4 Sammanfattning

Det finns ett antal olika testmetoder, bland annat Black – box testning, White – box testning, integrationstestning samt acceptanstestning. Genom att använda testning upptäcks fel i ett tidigare skede i utvecklingen och spar därmed både tid och pengar. Att test utförts lämnar inga garantier om ett felfritt system, däremot minskar risken för fel samt ökar systemets pålitlighet och kvalitet.

7 Skapande av produkten

I kapitel sju kommer vi att diskutera kring det tillvägagångssätt som valts i skapandet av produkten. I 7.1 Prototyp behandlar vi skapandet av prototypen. I kapitel 7.2 Databasdesign beskriver vi designen av databasen. I 7.3 Säkerhet redogör vi för de säkerhetsaspekter som finns på produkten. I kapitel 7.4 Användargränssnitt redogörs för skapandet av gränssnittet. I kapitel 7.5 Implementation av produkten beskriver vi mer ingående hur vi skapat vissa delar av produkten rent tekniskt. Slutligen i kapitel 7.6 Testning så behandlar vi de testmetoder och testfall som vi använt.

7.1 Prototyp

Det finns i huvudsak tre sorters prototyper vad gäller gränssnitt; bred, djup och T-prototyp. Bred prototyp saknar funktionalitet men är så gott som fullständig utseendemässigt. En djup prototyp har ett begränsat utseende men full funktionalitet. T-prototypen är en kombination av de båda tidigare nämnda.

Vi har valt att använda en bred prototyp för vår produkt. Användandet av den breda prototypen gav oss möjligheten att fastställa vilka olika delar som gränssnittet skulle innehålla. Vi byggde prototypen utifrån den kravspecifikation (se bilaga A) som utarbetats tillsammans med Momentum samt den information som framkommit vid samtal med företagets anställda. Prototypen som vi skapade åt Momentum består av sju stycken grafiska sidor utan funktionalitet. Momentum godkände vår prototyp och denna har sedan fungerat som en mall vid skapandet av den slutgiltiga produkten.

7.2 Databasdesign

Det första vi gjorde när vi skulle designa databaserna var att utgå från kravspecifikationen. Vi använde de entiteter som vi hade fått fram i utvecklingsprocessen. Entiteterna strukturerade vi med hjälp av E/R – diagram. Detta för att skapa ett databeroende mellan de olika delarna. Vi bestämde oss för att skapa två stycken databaser, en för användarkonton som används vid inloggning, samt en för packningssatserna. Anledningen till att vi skapade två databaser var att det saknas en naturlig relation mellan Användare och Packningssatser.

De tabeller som skapats finns beskrivna i bilaga C – Databasdesign. För att beskriva dessa tabeller har följande punkter använts.

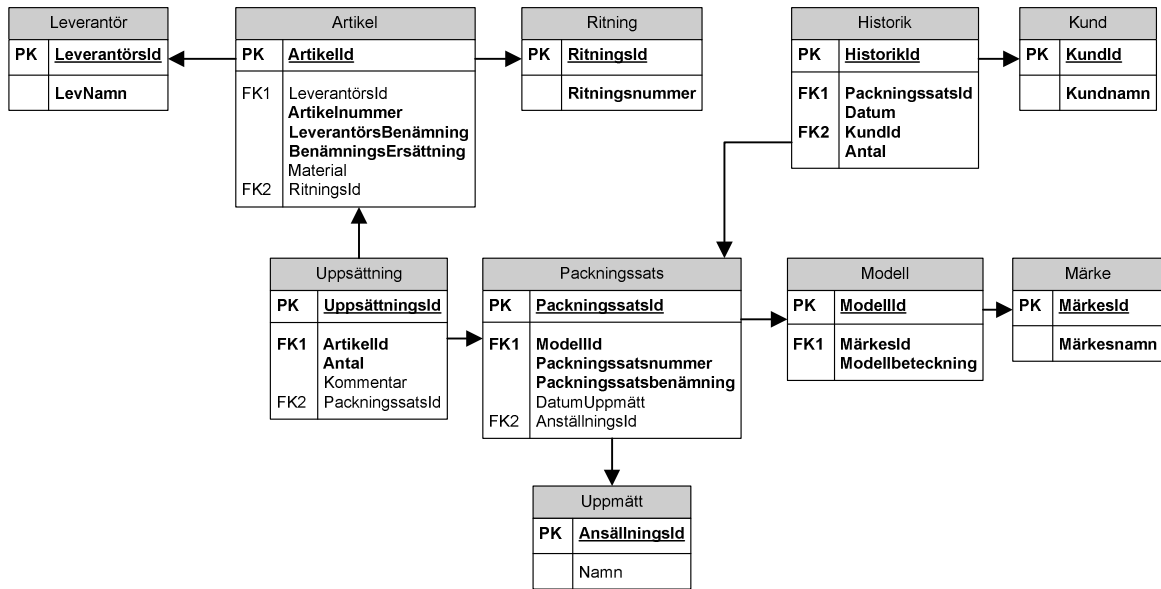
1. Beskrivning – Beskriver attributet översiktligt.
2. Datatyp – Vilken typ av data som attributet har
3. Verifieringsuttryck – Beskriver attributets speciella egenskaper
4. Obligatorisk – Anger om värde måste finnas.
5. Främmandenyckel – Anger relationen mellan två attribut i två olika tabeller. Denna punkt finns endast med om en sådan relation existerar.

Databasen användare består av en enda tabell, som också kallas Användare se Figur 7-1 - E/R diagram Användare.

Användare	
PK	<u>Användarnamn</u>
	Lösenord Förnamn Efternamn Nivå

Figur 7-1 - E/R diagram Användare

Databasen Momentum består av tio tabeller enligt Figur 7-2 - E/R diagram Momentum.



Figur 7-2 - E/R diagram Momentum

7.3 Säkerhet

Enligt kravspecifikationen skulle produkten ha en inloggningssida samt tre behörighetsnivåer. För att höja säkerheten har vi valt att lagra alla användarkonton i en separat databas. Dessutom har vi valt att kryptera lösenordet med 128-bitars kryptering. Varje användare lagras även med ett heltal som representerar användarens behörighetsnivå. Inloggningen går till så att användarnamn och lösenord matchas mot databasen och därefter erhålles en behörighetsnivå. På varje webbsida i produkten finns det även en kontroll av behörigheten. Detta för att hindra obehöriga användare att ta sig in i produkten någon annan väg än den tänkta, se efterföljande kodexempel .

```

include 'checkuser.php';
if(isset($_SESSION['user'])) // Kontrollerar att användare är inloggad
{
    if($_SESSION['level']<1) // Kontrollerar behörighetsnivå
    {
        $_SESSION['check']=1;
        header("location: index.php"); //Felaktig behörighet, kastas ut.
        exit;
    }
}
    
```

```

    }
}

else
{
    $_SESSION['check']=1;
    header("location: index.php");//Ej inloggad, åtkomst nekad.
    exit;
}

```

7.4 Användargränssnitt

När användargränssnittet skulle skapas utgick vi från prototypens utseende. Det primära kravet på gränssnittet var att all hantering av databasen skulle gå via gränssnittet. Produkten kommer att användas av många olika personer med olika förkunskaper, därför är det av stor vikt att produkten är lättmanövrerad med ett enkelt utseende. För att uppnå kraven som vi satt på gränssnittet, skapade vi några riktlinjer som vi följde under arbetets gång. Dessa var:

- Enhetligt – Alla sidor skall ha samma design.
- Enkel färgsättning – Använda färger som är behagliga för användaren. Dessutom skall färgerna matcha Momentums färger.
- Enkelhet – Designen skall vara enkel men ändå funktionell
- Balans – Sidorna skall vara balanserade.

På nästa sida visas två exempel hämtade från gränssnittet.

Skapa ny packningssats

Valj Märke

Packningssatsnummer Packningssatsbenämning Uppmätt av Datum

Artikelnummer Leverantörsbenämning Benämning ersättning Antal

Leverantör Riftningsnummer Material

Kommentar

Figur 7-3 - Skärmdump från skapa packningssats

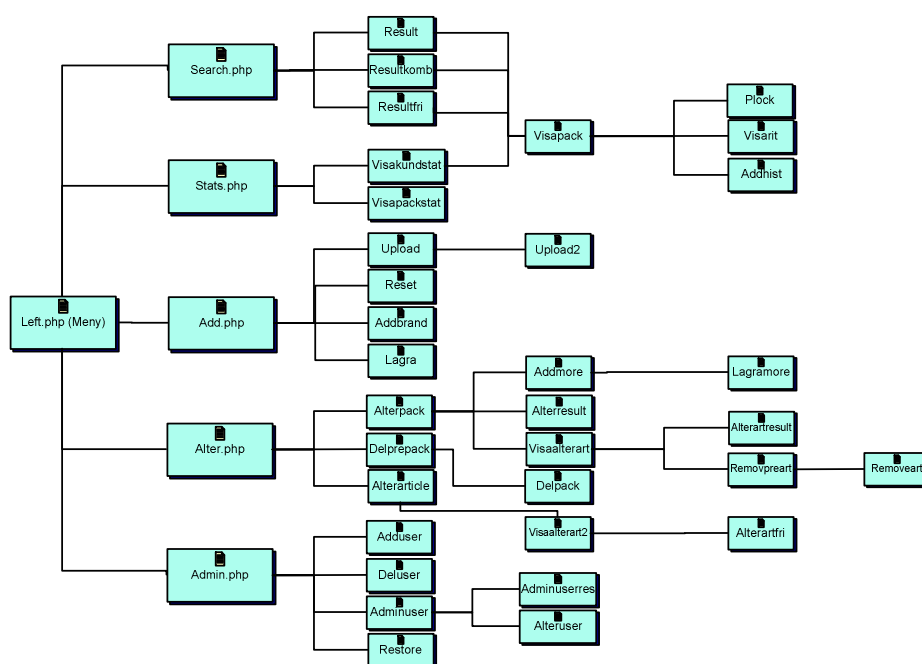
Välj artikel att ändra

Artnr	Benämning	Benämning/Ersättning	
AS 35/45x7/10			<input type="button" value="Ändra"/>
AS 40/50x7/10			<input type="button" value="Ändra"/>
AS 60/74x5/8			<input type="button" value="Ändra"/>
AS 45/53x5/7 P38			<input type="button" value="Ändra"/>
AS 60/68x5/7 P38			<input type="button" value="Ändra"/>
AS 75/83x5/7 P38			<input type="button" value="Ändra"/>
AS 90/98x5/7 P38			<input type="button" value="Ändra"/>
AS 105/113x5/7 P38			<input type="button" value="Ändra"/>
AS 120/128x5/7 P38			<input type="button" value="Ändra"/>
AS 140/148x5/7 P38			<input type="button" value="Ändra"/>

Figur 7-4 - Skärmdump sökresultat

Eftersom vårt gränssnitt skulle vara webbaserat så fanns det en del saker som vi var tvungna att ta hänsyn till. Ett problem var att olika användare har olika skärmapplösning, vilket riskerar att gränssnittet ser olika ut på olika arbetsstationer. Detta löste vi genom att skapa en dynamisk ram på högersidan som ändrar storlek beroende på användarens upplösning, detta möjliggör ett bibehållande av utseendet. Eftersom Momentum endast använder Internet Explorer som webbläsare så har vi valt att enbart anpassa produkten efter den och inte ta hänsyn till andra webbläsare.

De komponenter som ingår i gränssnittet samt hur de samverkar kan avläsas i Figur 7-5 - Filinkluderingsdiagram över gränssnittet.



Figur 7-5 - Filinkluderingsdiagram över gränssnittet

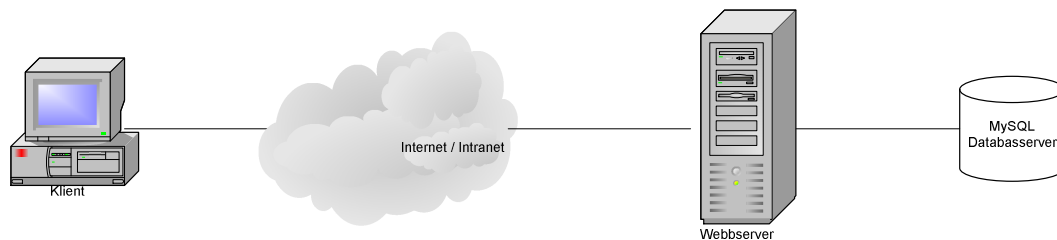
7.5 Implementation av produkten

Vi har valt att använda en klient/server arkitektur för produkten. Med detta menas att produkten ligger på en webserver som i sin tur är kopplad till en MySQL databasserver se Figur 7-6 - Klient/Server arkitektur. Momentum kommer sedan att ansluta sig till webbservern från sina datorer (klienter) via företagets intranät. Genom att använda den här lösningen så uppnås flera fördelar t.ex.

- Säkerhet – Säkerhet i form av central kontroll över all kommunikation i systemet.
- Uppgradering – Lätt att uppgradera programvaran då den är central på ett ställe.
- Oberoende klienter – Alla delade resurser finns på servern och klienter kan läggas till eller tas bort utan att det påverkar andra klienter.
- Inga belastningar på klienterna.

När vi skapade produkten tog vi även hänsyn till de nackdelar som finns med den här typen av lösning. Exempel på nackdelar är

- Single point of failure – Om ett fel uppstår i servern påverkas hela systemet.
- Hög belastning – Då all kommunikation sker via en ensam server skulle den kunna bli överbelastad om många klienter ansluter sig samtidigt.
- Det krävs en systemadministratör som tilldelar nya klienter rättigheter till systemet.



Figur 7-6 - Klient/Server arkitektur

7.5.1 Databasen

Vid skapandet av databasen använde vi ett verktyg vid namn PHP MyAdmin[9]. Detta är ett grafiskt verktyg skapat i PHP. PHP MyAdmin används för att skapa och administrera MySQL databaser. Genom att använda detta verktyg sparade vi avsevärt mycket tid jämfört med om vi använt det i MySQL inbyggda textbaserade gränssnittet.

Vi har valt att använda datatyperna INT, VARCHAR och DATE i produkten. INT står för heltal och används i produkten som identifierare för varje unik rad. Värdeområdet för INT i MySQL är mellan -2 147 483 648 till 2 147 483 647. VARCHAR lagrar tecken dynamiskt, dvs. den storlek som teckensträngen är, vilken kan ligga mellan 0 till 255 tecken. Vi använder denna datatyp för att lagra namn och övriga textsträngar. DATE används för att lagra datum enligt formen ÅÅÅÅ-MM-DD. Intervallet för DATE är från 1000-01-01 till 9999-12-31. För

att kontrollera att ett giltigt datum matats in använder vi oss av funktionen checkdate(MM, DD, ÅÅÅÅ). För att kunna använda checkdate var vi tvungna att kasta om ordningen på den inmatade strängen då den inte var av samma format. Se exemplet nedan.

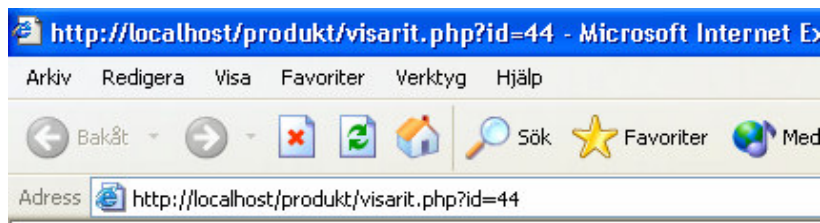
```
if($_POST['date']!=NULL) //om något inmatat i fältet datum
{
    $dat=chunk_split($_POST['date'], 2, ':'); //Delar i sär sträng och skapar datum
    $yy=strtok($dat,':');
    $mm=strtok(':');
    $dd=strtok(':');

    if(!checkdate($mm,$dd,$yy)) //Kontrollerar datum
    {
        $_POST['date']="";
        ?>
        <script>
            window.alert("Felaktigt datum\n Mata in enligt \nÅÅMMDD ex: 031231");
        </script>
        <?
    }
}
```

7.5.2 Gränssnittet

När vi påbörjade arbetet med produkten ställdes vi inför några olika problem. Bland annat visste vi inte hur variabler skulle skickas mellan gränssnittets olika sidor. Efter att ha läst på om variabler och sökt på olika PHP – forum [4], [5] konstaterade vi att det fanns tre sätt att skicka variabler mellan sidor; GET, POST och SESSION. Men det finns några viktiga skillnader dem emellan.

GET skickar variabeln via länkar. Dessa länkar visas i URL: en som i Figur 7-7 - Anrop med GET. I detta fall är variabelns namn *id* och har värdet 44.



Figur 7-7 - Anrop med GET

Nackdelen med GET är att variabeln visas i klartext vilket kan vara en säkerhetsrisk då all information är tillgänglig för användaren.

POST lägger variabeln i en fördefinierad array som följer med mellan sidorna. POST är det säkraste mellan de två nämnda då variabeln inte syns för användaren.

SESSION lagrar variabeln globalt på servern med hjälp av cookies. Denna variabel är då tillgänglig på alla sidor till dess att den avrefereras. Detta görs med kommandot unset.

En annan del vi funderade över var sökningen. Vi ville matcha delar av en sträng mot inmatad sökparameter. Även här sökte vi i PHP – forum [4], [5] på nätet för att få information. Vi valde att lösa problemet på följande sätt.

Det finns tre stycken sökfunktioner i produkten; sök märke/modell, kombinerad sökning samt fritextsökning. Sökfunktionerna fungerar i grunden på samma sätt. Användaren får mata in sökparametrar, dessa lagras i ett formulär som skickas vidare till sidan där resultatet skall visas. Där tas parametrarna emot med POST och används sedan för att ställa en SQL – fråga. Kodexemplet nedan är hämtat från den kombinerade sökningen.

```
$komb= "SELECT * FROM kund, historik, packningssats, anstalld, uppsattning, artikel, leverantör, ritning
      WHERE(((kund.KundId=historik.KundId)
      AND (historik.PackningssatsId=packningssats.PackningssatsId)
      AND (packningssats.AnställningsId=anstalld.AnställningsId)
      AND (packningssats.PackningssatsId=uppsattning.PackningssatsId)
      AND (uppsattning.ArtikelId=artikel.ArtikelId)
      AND (artikel.RitningsId=ritning.RitningsId)
      AND (artikel.LeverantörsId=leverantör.LeverantörsId)
      AND ((packningssats.Packningssatsnummer LIKE '{$_POST['pnummer']}')
      AND (leverantör.LeverantörsNamn LIKE '{$_POST['Levmeny']}')
      AND (kund.KundNamn LIKE '{$_POST['kundmeny']}')
      AND (artikel.Artikelnummer LIKE '{$_POST['Artnum']}')
      AND (packningssats.Packningssatsbenämning LIKE '{$_POST['pben']}')
      AND (artikel.LeverantörsBenämning LIKE '{$_POST['Benämning']}')
      AND (artikel.BenämningErsättning LIKE '{$_POST['BenErs']}')
      AND (anstalld.namn LIKE '{$_POST['Measure']}')
      AND (packningssats.DatumUppmätt LIKE '$fixeddate'))
      GROUP BY packningssats.PackningssatsId ORDER BY Packningssatsnummer";
```

I exemplet har vi använt LIKE för att matcha sökkriterierna. Detta för att tillåta användaren att använda Wildcards i sökningen dvs. använda reguljära uttryck tex. % som matchas mot vilka tecken som helst. Sökning efter packning% ger träffar på alla ord som börjar med packning.

Vi har valt att använda dynamiska rullgardinsmenyer i vår produkt. Med detta menar vi att innehållet på menyn presenteras olika beroende på vad användaren gör för några val samt vad som finns i databasen. Nedan visas ett kodexempel på en sådan meny.

```
echo '<select name="Brand" onChange="this.form.submit()">';
echo "<option >,$_POST['Brand'],'</option>";
if($_POST['Modell']==NULL)
{
    for($i=0; $i< $numrows; $i++)
    {
        $row=mysql_fetch_array($brandresult);
        if($row['Brandname'] != 'saknas')
        {
            echo '<option >,$row['Brandname'],'</option>';
        }
    }
}
echo '</select>';
```

7.6 Testning

Vårt mål är att kunna leverera en produkt med minimalt antal fel. Detta görs lämpligast genom ett flertal tester. Vi anser att lättaste sättet att konstruera testfall är genom att studera flödesscheman för produkten och därmed få en bra överblick över systemet samt se till att alla grenar testas. I bilaga D – Testfall, presenteras de testfall som gjorts på produkten samt de flödesscheman som konstruerats. Samtliga testfall har genomförts med godkänt resultat. Till vår hjälp har vi använt boken *The Unified Software Development Process* [6]. Följande punkter används för att beskriva varje testfall.

1. Syfte – Varför vi har skapat testfallet
2. Input – Vad som skall matas in.
3. Resultat – Förväntat resultat från testfallet.
4. Beskrivning – Testets tillvägagångssätt
5. Villkor – Finns endast om testet har några speciella villkor som måste uppfyllas

Det är det här som kallas Black – box testning. Vi har även gjort White – box testning under skapandet av produkten för att testa specifika delar av systemet t.ex. SQL frågor. Dessutom

har vi testat vissa delar separat för att sedan integrera dem med övriga komponenter, så kallad integrationstest. Vi har även låtit Momentum testa och komma med synpunkter på produkten, dvs. acceptanstest.

7.7 Sammanfattning

När vi skapade produkten gick vi igenom olika utvecklingsfaser. Först skapade vi prototypen genom att utgå från kravspecifikationen, detta för att förfina kundens krav samt strukturera uppgiften. Efter godkännande från kund påbörjade vi designen av databasen med hjälp av E/R – diagram. Detta resulterade i två databaser samt totalt 11 tabeller.

Till vår produkt har vi valt att använda en inloggningssida för att identifiera användare samt tre stycken behörighetsnivåer för dessa.

Användargränssnittet skapades utifrån resultatet från prototypen. Vi ställde fyra krav på gränssnittets utseende; enhetlighet, enkel färgsättning, enkel design samt balans.

För att öka produktens tillförlitlighet har vi använt oss av testning. De testmetoder vi använt är: White – box testning, Black – box testning, acceptanstestning och integrationstestning.

8 Slutgiltiga kommentarer

Det här projektet är nu till ända och Momentum har godkänt produkten. Målet med produkten var att skapa en informationsdatabas med ett användarvänligt gränssnitt. För att kunna skapa ett användargränssnitt var vi tvungna att ta reda på vilka delar som skulle ingå. Vi skapade en prototyp som fick ligga till grund för utvecklingen av användargränssnittet. Användargränssnittet blev webbaserat med krav på inloggning för användarna. För att höja säkerheten valde vi att skapa tre behörighetsnivåer. Varje behörighetsnivå har fått sin egen vy. Användargränssnittet är kopplat till ett databashanteringssystem som heter MySQL. Kommunikationen mellan gränssnittet och databashanteringssystemet sker med hjälp av skriptspråket PHP.

För att minimera risken för att produkten innehåller fel har ett flertal tester utförts. De fel som hittades vid testningen har rättats till och därmed har risken för framtida fel minskat.

Det enda kravet som fanns på produkten vad gäller säkerhet var som tidigare nämnts inloggningsfunktionen. Vi anser att detta räcker då produkten endast kommer användas internt inom Momentum, och alltså inte utsätts för hot utifrån.

I en förlängning är det tänkt att en viss del av databasen skall öppnas upp mot företagets kunder och då måste ett nytt användargränssnitt skapas men med begränsad åtkomst.

Referenser

- [1] <http://www.lida.liu.se/~tompa/databaser/mysql.html>, 2004-04-16
- [2] Welling, Thomson, PHP and MySQL Web Development 2nd Edition, Sams 2003
- [3] Beck, Extreme Programming Explained: embrace change, Pearson Education 1999
- [4] <http://www.php.net>, 2004-04-15
- [5] <http://www.phpportalen.net>, 2004-04-15
- [6] Booch, Jacobson, Rumbaugh, The Unified Software Development Process, Addison-Wesley 1999
- [7] <http://www.mysql.com>, 2004-04-16
- [8] <http://www.microsoft.com/products/info/>, 2004-04-17
- [9] <http://www.phpmyadmin.net>, 2004-05-04

A Bilaga – Kravspecifikation

Kravspecifikation

Informationsdatabas

Innehållsförteckning

A.1 Inledning	37
A.2 Systembeskrivning.....	37
A.2.1 Befintlig system.....	37
A.2.2 Planerat system.....	37
A.2.3 Säkerhetsaspekter	37
A.3 Omfattning.....	37
A.3.1 Utrustning	37
A.3.2 Övriga uppgifter	37
A.4 Krav	38
A.4.1 Generella krav	38
A.4.2 Specifika krav.....	38
A.4.3 Teknisk miljö	38

A.1 Inledning

Tanken med detta exjobb är att skapa en informationsdatabas plus användargränssnitt åt Momentum. Databasen riktar sig i första skedet till Momentums anställda men i en förlängning även till deras kunder.

A.2 Systembeskrivning

A.2.1 Befintlig system

Eventuellt finns en webbserver att tillgå. Annars måste det ordnas med ytterligare en extern server.

A.2.2 Planerat system

En informationsdatabas skall implementeras på en server. Till denna skall det skapas ett webbaserat gränssnitt.

Databasen skall innehålla information om packningssatser som Momentum tillhandahåller.

A.2.3 Säkerhetsaspekter

Det ställs krav på att det finns säkerhet i form av en inloggningssida för att få access till databasen. Olika användare kan komma att tilldelas olika rättigheter, vilket måste tas i beaktande. Databasen skall även förberedas på att användas av kunder, därför måste vissa uppgifter kunna döljas.

A.3 Omfattning

I arbetet ingår att designa relationsdatabasen, samt att skapa ett tillhörande användargränssnitt och inloggningsfunktion. Momentum kommer själva stå för att fylla databasen med information.

A.3.1 Utrustning

Momentum tillhandahåller all nödvändig utrustning samt står för eventuella omkostnader.

A.3.2 Övriga uppgifter

Ej klart med vilken utrustning som skall användas. SQL och PHP eller motsvarande kommer att användas för implementationen.

A.4 Krav

A.4.1 Generella krav

Databasen skall innehålla följande fält:

1. Artikelnummer
2. Leverantör
3. Benämning
4. Benämning Ersättning (Momentums egen benämning)
5. Material
6. Antal
7. Datum upplagt
8. Vem som mätt upp
9. Packningssatsnummer
10. Benämning packningssats
11. Kund
12. Kundnummer

A.4.1.1 Användargränssnitt

I användargränssnittet skall det finnas följande operationer.

- Lägga till packningssats
- Söka efter packningssats, packningssatsnummer.
- Söka på Märke, artikelnummer, benämning, benämning ersättning, kund, kundnummer.
- Ta bort packningssats.
- Ändra i packningssats.

A.4.1.2 Säkerhet

Det kommer finnas minst tre behörighetsnivåer. En för sökning, en för borttagning och ändring samt en för att lägga till uppgifter. Detta för att hindra obehöriga att ta del av information som finns lagrat i databasen.

A.4.2 Specifika krav

Databasen skall vara i princip underhållsfri. Allt som behöver göras skall kunna skötas från användargränssnittet.

A.4.3 Teknisk miljö

Tillgång till nödvändig utrustning dvs. Dator och Programvara beroende på vilken typ av databas som skall användas.

B Bilaga – Användarmanual

Del

1

MOMENTUM

ANVÄNDARMANUAL

Databas för packningssatser

MOMENTUM

Databas för packningssatser

Innehåll

B1. Översikt.....	43
B2. Att söka i databasen	44
B2.1 Märke/ Modell.....	44
B2.2 Kombinerad sökning.....	44
B2.3 Fritextsökning.....	45
B2.4 Sökresultat.....	45
B3. Att avläsa statistik	48
B3.1 Statistik för kund	48
B3.2 Statistik för packningssats.....	48
B4. Att lägga till packningssats.....	50
B4.1 Koppla ritning till artikel.....	50
B4.2 Lägga till märke/modell.....	51
B4.3 Skapa ny packningssats.....	52
B5. Att ändra eller ta bort	53
B5.1 Ändra i packningssats	53
B5.1.1 Ändra uppgifter för packningssatsen.....	54
B5.1.2 Lägg till fler artiklar till en packningssats.....	54
B5.1.3 Radera eller ändra artikel i packningssats	54
B5.2 Radera packningssats.....	55
B5.3 Ändra i en artikel	55

B.1 Översikt

Här nedan syns en översikt över gränssnittet. För att ta sig vidare används menyn på vänster sida.



Beroende på vilken behörighetsnivå du har kommer du presenteras olika val. Har du behörighetsnivå ett kan du bara söka, samt se och lägga till i statistik. Vid behörighetsgrad två kan du även skapa nya packningssatser samt ändra och ta bort befintliga. Behörighetsgrad tre gäller bara administratören och tas ej upp i denna manual.

B.2 Att söka i databasen

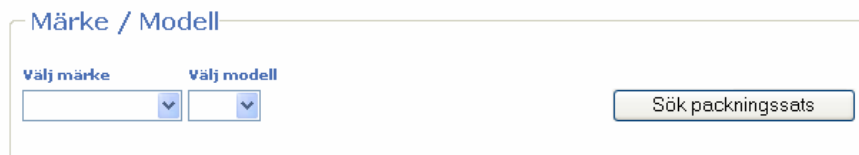
Det finns tre sätt att söka i databasen. Sökning på Märke och Modell, kombinerad sökning samt fritextsökning.

B.2.1 Märke/ Modell

För att söka på märke/modell gör följande:

- Välj märke i rullgardinsmenyn för märke.
- Välj modell i rullgardinsmenyn för modell
- Klicka på *sök packningssats*.

Om du önskar att söka på alla packningssatser för ett specifikt märke. Välj endast *märke* i rullgardinsmenyn för märke. Klicka på *sök packningssats*.



The screenshot shows a search form titled "Märke / Modell". It contains two dropdown menus: "Välj märke" and "Välj modell". To the right of these menus is a button labeled "Sök packningssats".

B.2.2 Kombinerad sökning

För att få ett så exakt sökresultat som möjligt kan man använda kombinerad sökning. Med den kombinerade sökningen gör man sökningar med en eller flera kombinationer. För att göra en kombinerad sökning, fyll i de fält du önskar att söka på och klicka på *sök packningssats*

Kombinerad sökning

Packningsattsnummer	Leverantör
<input type="text"/>	Alla <input type="button" value="v"/>
Benämning packningsatts	Kund
<input type="text"/>	Alla <input type="button" value="v"/>
Artikel Nummer	
<input type="text"/>	
Benämning	Uppmätt (ÅÅMMDD)
<input type="text"/>	<input type="text"/>
Ersättning	Uppmätt av
<input type="text"/>	<input type="text"/>

Det finns även möjlighet att söka med hjälp av tecknet %. Detta används främst när du bara vet delar av en söksträng.

T ex. sökning på artikelnummer %125% matchar alla packningsattsar som innehåller någon artikel med 125 någonstans i artikelnumret.

Exempel:

K1 125/10 x 9,5

OR 125,6 x 3

% - tecknet motsvarar alltså vilka teckenkombinationer som helst.

B.2.3 Fritextsökning

Med fritextsökning kan man söka på vilken sträng som helst i en packningsatts. Även här kan man använda sig av % - tecknet för att söka. Detta beskrivs i *kombinerad sökning*.

Fritextsökning

Fritext(max 50 tecken)

B.2.4 Sökresultat

Efter att sökningen är gjord så kommer ett sökresultat att presenteras. Detta resultat är klickbart. Om du klickar på packningsattsens nummer så kommer du att kunna se vilka artiklar som ingår i satsen.

Sökresultat

Packningssatsnummer	Benämning	Uppmätt av	Datum
1030451	125/60	KNN	2002-02-22

Gör så här:

- Välj packningssats ur listan med sökresultatet genom att klicka på packningssatsens nummer.
- Nu bör du se packningssatsen i detalj.

Packningssats 1030451

Packningssatsnummer	Benämning	Uppmätt av	Datum	Märke	Modell
1030451	125/60	KNN	2002-02-22	MacGREGOR	

Artnr	Benämning	Ersättning	Leverantör	Ritningsnr	Material	Antal
	K1 125/103x9,5		MOMSEAL	macgregor557	PUR-H	2
		<i>Kommentär:</i>				
NI300 60 80 12						1
		<i>Kommentär:</i>				
	A2 60/73x6/9,5		MOMSEAL	macgregor145	PUR-H	1
		<i>Kommentär:</i>				
	F3 125/103x5		MOMSEAL	macgregor557	POM	2
		<i>Kommentär:</i>				
OR 39,69 x 3,53						1
		<i>Kommentär:</i>				
OR 129,2 x 5,7						1
		<i>Kommentär:</i>				
OR 134,5 x 3						1
		<i>Kommentär:</i>				

Skriv ut plocksedel

Fältet för ritningsnummer till en artikel är klickbar, detta under förutsättning att en ritning är kopplad till artikeln. Detta märks genom att fältet *ritningsnr* är ifyllt. För att titta på en ritning klicka på ritningsnumret.

1. Önskas en plocksedel klicka på *skriv ut plocksedel*.

2. Om en kund väljer att köpa en packningssats, ska denna läggas till i statistiken. Detta görs genom att välja kund ur rullgardinsmenyn *Välj kund* och därefter skriva in antal köpta packningssatser i fältet *Antal*. Klicka sedan på *Lägg till statistik*.

Lägg till i statistik

Välj Kund

1234567 VOLVO

Antal

Lägg till statistik

Lägg till kund

Kundnr

Kundnamn

Lägg till ny kund

3. Om den aktuella kunden inte finns i rullgardinsmenyn, måste kunden läggas till. Detta görs i fältet *lägg till kund*. Mata in kunduppgifter och klicka sedan på *Lägg till ny kund*. Upprepa sedan steg 2.

B.3 Att avläsa statistik

Det finns två metoder för att avläsa statistik i den här databasen, för en specifik kund eller för en specifik packningssats.

B.3.1 Statistik för kund

Om du vill avläsa statistik för en specifik kund, väljer du kundens namn i rullgardinsmenyn *Välj kund* och klickar *Visa statistik*.

Visa statistik för kund

Välj kund

1234567 VOLVO ▾

Visa statistik

Om kunden någon gång har köpt packningssatser kommer ett resultat för kunden att presenteras innehållande:

- Totala antalet packningssatser kunden köpt
- Vilka packningssatser kunden köpt med datum och antal

Statistik för kund 1234567 VOLVO

Kund	Totalt antal
1234567 VOLVO	13

Packningssatsnummer	Datum	Antal
1030451	2004-04-23	4
1030451	2004-04-23	9

Skriv ut resultat

Klickar du på packningssatsnumret hamnar du i sökresultatet för *sökning* på packningssats och kan se packningssatsens ingående artiklar. De val du nu kan göra kan du läsa om i kapitel 2.4.

B.3.2 Statistik för packningssats

Du kan avläsa statistik för en bestämd packningssats genom att mata in packningssatsnumret i rutan *Mata in packningssatsnummer* och klicka på *Visa statistik*.

Visa statistik för packningssats

Mata in packningssatsnummer

Visa statistik

Om du gör det och packningssatsen finns med i databasen, kommer nu ett resultat visas för dig. Det du kan avläsa av resultatet är:

- Hur många av packningssatsen som sålts totalt
- Vilka kunder som köpt packningssatsen med datum och antal

Statistik för packningssats 1030451

Packningssatsnummer	Totalt antal
1030451	18

Kund	Datum	Antal
1234567 VOLVO	2004-04-23	4
4805756 SAAB	2004-04-23	5
1234567 VOLVO	2004-04-23	9

Skriv ut resultat

B.4 Att lägga till packningssats

Om du har behörighetsnivå två, kan du lägga till packningssatser i databasen. Detta gör du genom att klicka på *Skapa ny* i menyn.

Det första du bör tänka på när du skall skapa en ny packningssats är att fundera på om du vill koppla en ritning till någon artikel i packningssatsen.

Skapa ny packningssats

Välj Märke	<input type="button" value="Lägg till nytt märke/modell"/>			
<input type="text"/>	Packningssatsnummer	Packningssatsbenämning	Uppmätt av	Datum
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Artikelnummer	Leverantörsbenämning	Ersättning	Antal	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Leverantör	Ritningsnummer	Material		
<input type="text"/>	<input type="text"/>	<input type="text"/>		
Kommentar	<input type="text"/>			
<input type="button" value="Ladda upp ritning"/>	<input type="button" value="Avbryt"/>	<input type="button" value="Lägg till ytterligare artiklar"/>		

B.4.1 Koppla ritning till artikel

För att koppla en ritning till en artikel måste ritningen finnas med i databasen. Gör den inte det måste den laddas upp till databasen från din egen dator, vilket innebär att ritningen måste finnas på din dator och att den är sparad i formatet JPG.

För att lägga till en ritning i databasen klickar du på *Ladda upp ritning*. Du kommer nu till en ny sida. Här klickar du på knappen *Bläddra...* Ett nytt fönster öppnas och du kan nu leta upp ritningen där du sparad den i din dator. Klicka sedan *OK* för att stänga ner fönstret. Klicka nu på *Lägg till*, så laddas ritningen upp till databasen.

Ladda upp ritningar

Välj fil:

Befintliga ritningar:

Filnamn	Storlek	
inbmomsealavstrykare.jpg	31 kB	Radera
macgregor145.jpg	34 kB	Radera
macgregor557.jpg	89 kB	Radera

Att ritningen laddats upp kan du se genom att den finns med i tabellen *Befintliga ritningar* under *Filnamn*.

Efter att du har laddat upp de ritningar du vill ha med i packningssatsen, kan du återgå till *Skapa ny*.

B.4.2 Lägga till märke/modell

Om du vill koppla ett märke eller en modell till din packningssats väljer du först märke ur rullgardinsmenyn *Märke*. Efter det dyker en ny rullgardinsmeny upp där du kan välja *Modell*.

Skulle det saknas märke eller modell i rullgardinsmenyerna, måste dessa läggas till. Detta görs genom att man klickar på *Lägg till nytt märke/modell*. Genom att klicka på knappen tas du till en ny sida där du kan skriva in det nya märket i fältet *Mata in märke*. Klicka därefter på *Lägg till märke*, så läggs det nya märket in i databasen.

Lägg till märke

Mata in märke

Lägg till ny modell till märke

Välj märke Mata in ny modell

Om du önskar lägga till en ny modell under ett märke, gör du så här:

- Välj önskat märke i rullgardinsmenyn *Välj märke*
- Skriv in namnet på den nya modellen i textfältet *Mata in ny modell*
- Klicka *Lägg till ny modell*

För att återgå till att skapa nya packningssatser klickar du på *Återgå till skapa packningssats*.

B.4.3 Skapa ny packningssats

Nu är du redo att skapa en ny packningssats. Det finns vissa fält som måste fyllas i för att man skall kunna lagra packningssatsen i databasen. Dessa fält är:

- Packningssatsnummer
- Minst ett av fälten Artikelnummer, Benämning eller Ersättning

Övriga fält är valfria.

Skapa ny packningssats

Välj Märke	Välj Modell	Lägg till nytt märke/modell	
MacGREGOR			
Packningssatsnummer	Packningssatsbenämning	Uppmätt av	Datum
12345	stängtätning	AFS	040423
Artikelnummer	Leverantörsbenämning	Ersättning	Antal
OR 11,91x2,62			3
Leverantör	Ritningsnummer	Material	
MOMSEAL	1334	NBR 70	
Kommentar			
Ladda upp ritning	Avbryt	Lägg till ytterligare artiklar	

Packningssatsnumret måste vara unikt för varje packningssats. Skulle du råka skriva in ett nummer som redan finns i databasen, kommer en varningstext att poppa upp och du får välja ett nytt nummer.

OBS: Om du vill koppla en ritning till en artikel, måste du skriva in samma namn i fältet *Ritningsnummer* som du döpte ritningen till databasen, fast utan *.jpg*. För att ladda upp en ritning i databasen, se kapitel 4.1.

Antag att det finns en ritning med namnet macgregor557.jpg lagrat i databasen och att man vill koppla den till en artikel. Då skriver man in macgregor557 i fältet *Ritningsnummer*.

Mata nu in de värden och namn du vill ha med i packningssatsen och klicka därefter på *Lägg till ytterligare artikel*. Det du matat in presenteras nu längst ner på sidan. Vill du lägga till fler artiklar till samma packningssats fortsätter du att mata in värden i de fält du vill ha med och trycker på *Lägg till ytterligare artikel*. När du är färdig med de artiklar som ingår i packningssatsen klickar du på *Bekräfta* för att lagra uppgifterna i databasen. *Bekräfta* finner du längst ned på sidan.

B.5 Att ändra eller ta bort

Om du har behörighetsnivå två kan du ändra och ta bort artiklar och packningssatser. Det finns tre olika val du kan göra när du klickar på *Ändra/Ta bort* i menyn. Dessa val är:

- Ändra i packningssats
- Ta bort packningssats
- Ändra i artikel

Sök packningssats att ändra

Mata in packningssatsnummer

Sök packningssats

Radera packningssats

Mata in packningssatsnummer

Radera packningssats

Sök artikel att ändra

Mata in artikelnummer

Mata in benämning

Mata in ersättning

Sök artikel att ändra

B.5.1 Ändra i packningssats

För att ändra i en packningssats måste du mata in packningssatsnumret i fältet *Mata in packningssatsnummer*. Klicka sedan på *Sök packningssats*. Om packningssatsen du matade in finns med i databasen kommer den att presenteras.

Ändra i packningssats

Märke	Modell	Lägg till ytterligare artikel	
MacGREGOR			
Packningssatsnummer	Packningssatsbenämning	Datum	Uppmätt av
1030451	125/60	2002-02-22	KNN
Ändra			

Artnr	Benämning	Ersättning	Antal	
	K1 125/103x9,5		2	Ändra
NI300 60 80 12			1	Ändra
	A2 60/73x6/9,5		1	Ändra
	F3 125/103x5		2	Ändra
OR 39,69 x 3,53			1	Ändra
OR 129,2 x 5,7			1	Ändra
OR 134,5 x 3			1	Ändra

B.5.1.1 Ändra uppgifter för packningssatsen

Här kan du nu mata in nya värden i textrutorna om det är något du vill ändra. Efter att du ändrat klickar du *Ändra*, så lagras uppgifterna i databasen.

B.5.1.2 Lägg till fler artiklar till en packningssats

Vill du lägga till fler artiklar till packningssatsen, klickar du på *Lägg till ytterligare artikel*. När du matat in värden för artikeln, klickar du på *Lägg till artikel*.

Lägg till artikel i packningssats

Artikelnummer	Leverantörsbenämning	Ersättning	Antal
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Leverantör	Ritningsnummer	Material	
<input type="text"/>	<input type="text"/>	<input type="text"/>	
Kommentar	<input type="text"/>		
Avbryt Lägg till artikel			

Upprepa nu förfarandet om du vill lägga till fler artiklar. När du är nöjd klickar du på *Bekräfta* och packningssatsen uppdateras med det nya innehållet.

B.5.1.3 Radera eller ändra artikel i packningssats

Efter att du klickat på *Ändra*/*Ta bort* i menyn och matat in packningssatsen du vill ändra i, kommer ett resultat att presenteras. Längst ner kommer en tabell att visas

med artiklarna som ingår i packningssatsen. Om du vill ändra antal eller kommentar för artikeln eller om du vill radera den ur packningssatsen, klickar du på *Ändra* på den rad du avser att ändra.

Ändra i artikel

Artikelnummer	Benämning	Ersättning	
	K1 125/103x9,5		
Leverantör	Ritningsnummer	Material	Antal
MOMSEAL	macgregor557	PUR-H	<input type="text" value="2"/>
Kommentar			
<input type="text"/>	<input type="button" value="Ändra artikel i packningssats"/>		
	<input type="button" value="Radera artikel i packningssats"/>		

Du kommer nu till en ny sida där du kan välja att antingen radera artikeln ur packningssatsen eller att ändra antal eller skriva i kommentarfältet. Vill du radera artikeln ur packningssatsen klickar du på *Radera artikel i packningssats*. Om du istället vill ändra antal eller kommentar, skriver du in de nya värdena och klickar på *Ändra artikel i packningssats*, så lagras de nya uppgifterna i databasen.

B.5.2 Radera packningssats

Efter att ha klickat på *Ändra/Ta bort* i menyn, kan du välja att radera en packningssats ur databasen. För att radera en packningssats skriver du in packningssatsnumret i *Mata in packningssatsnummer* och klickar *Radera packningssats*. Du kommer nu till en ny sida där du måste bekräfta att du vill radera packningssatsen. Klicka *Ja, radera* för att radera eller *Nej, återgå* om du ändrat dig.

B.5.3 Ändra i en artikel

I *Ändra/Ta bort* finns möjlighet att ändra i en artikel. Detta gör du genom att först söka upp den artikel du vill ändra i. Här finns även möjlighet att söka med hjälp av %-tecknet. Läs vidare i kapitel 2.2.

- Mata in sökord i minst ett av de trefälten *Artikelnummer*, *Benämning* eller *Ersättning*
- Klicka på *Sök artikel att ändra*

Nu kommer ett sökresultat att visas. För att ändra i en artikel klickar du på *Ändra* för den artikel det motsvarar.

Välj artikel att ändra

Artnr	Benämning	Ersättning	
	K1 125/103x9,5		Ändra
NI300 60 80 12			Ändra
	A2 60/73x6/9,5		Ändra
	F3 125/103x5		Ändra
OR 39,69 x 3,53			Ändra
OR 129,2 x 5,7			Ändra
OR 134,5 x 3			Ändra

Du kommer nu till en ny sida där du kan skriva in nya uppgifter för artikeln. Klicka sedan *Ändra artikel* för att spara de nya ändringarna.

Ändra i artikel

Artikelnummer	Benämning	Ersättning	
<input type="text" value="OR 39,69 x 3,53"/>	<input type="text"/>	<input type="text"/>	
Leverantör	Ritningsnummer	Material	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Ändra artikel"/>

OBS: Var uppmärksam på att om du ändrar i en artikel så ändrar du i alla packningssatser där artikeln finns.

C Bilaga – Databasdesign

C.1 Användare

Databasen Användare består av en enda tabell och är uppbyggd på följande sätt:

C.1.1 Användare

Tabellen Användare lagrar användarkonton för Momentums anställda. Den innehåller följande fält.

Användarnamn: **beskrivning:** Unikt användarnamn för varje användare.
datatyp: varchar(20)
verifieringsuttryck: not null
obligatorisk: ja

Lösenord: **beskrivning:** Krypterat lösenord för varje användare.
datatyp: varchar(32)
verifieringsuttryck: not null
obligatorisk: ja

Förnamn: **beskrivning:** Användarens förnamn.
datatyp: varchar(20)
verifieringsuttryck: not null
obligatorisk: ja

Efternamn: **beskrivning:** Användarens efternamn
datatyp: varchar(20)
verifieringsuttryck: not null
obligatorisk: ja

Nivå: **beskrivning:** Användarens behörighetsnivå.
datatyp: integer
verifieringsuttryck: 1,2 eller 3
obligatorisk: ja

C.2 Momentum

Databasen Momentum består av tio tabeller och är uppbyggd på följande sätt:

C.2.1 Anställd

I Anställd lagras de anställda som lägger upp packningssatser i databasen.

AnställningsId: **beskrivning:** Identifierare för Momentums personal.
datatyp: integer
verifieringsuttryck: > 0

obligatorisk: ja

Namn: **beskrivning:** Den anställdes för- och efternamn.
datatyp: varchar(40)
verifieringsuttryck: not null
obligatorisk: ja

C.2.2 Artikel

I Artikel lagras de artiklar som ingår i packningssatserna.

ArtikelId: **beskrivning:** Identifierare för Momentums artiklar.
datatyp: integer
verifieringsuttryck: > 0
obligatorisk: ja

RitningsId: **beskrivning:** Identifierare för ritning tillhörande en artikel.
datatyp: integer
verifieringsuttryck: > 0
obligatorisk: ja
Främmandenyckel: Refererar till RitningsId i Ritning

LeverantörsId: **beskrivning:** Identifierare för leverantör av artikel.
datatyp: integer
verifieringsuttryck: > 0
obligatorisk: ja
Främmandenyckel: Refererar till LeverantörsId i Leverantör

Artikelnummer: **beskrivning:** Unikt nummer för artikel.
datatyp: varchar(50)
verifieringsuttryck: -
obligatorisk: nej¹

LeverantörsBenämning: **beskrivning:** Leverantörens artikelbenämning.
datatyp: varchar(50)
verifieringsuttryck: -
obligatorisk: nej¹

BenämningsErsättning: **beskrivning:** Alternativ artikelbenämning.
datatyp: varchar(50)
verifieringsuttryck: -
obligatorisk: nej¹

Material: **beskrivning:** Artikelns material.
datatyp: varchar(30)
verifieringsuttryck: -
obligatorisk: nej

¹ Ett av fälten Artikelnummer, LeverantörsBenämning eller BenämningsErsättning måste vara ifyllt.

C.2.3 Historik

I Historik lagras statistik om de packningssatser som Momentum säljer, samt vilka som köpt packningssatser.

HistorikId:	beskrivning: Identifierare för statistik. datatyp: integer verifieringsuttryck: >0 obligatorisk: ja
KundId:	beskrivning: Identifierare för kund. datatyp: integer verifieringsuttryck: >0 obligatorisk: ja främmandenyckel: Refererar till KundId i Kund.
PackningssatsId:	beskrivning: Identifierare för packningssats. datatyp: integer verifieringsuttryck: >0 obligatorisk: ja främmandenyckel: Refererar till PackningssatsId i Packningssats.
Datum:	beskrivning: Datum för köp av packningssats. datatyp: date verifieringsuttryck: not null obligatorisk: ja
Antal:	beskrivning: Antal packningssatser som köps. datatyp: integer verifieringsuttryck: >0 obligatorisk: ja

C.2.4 Kund

I Kund lagras de kunder som köper packningssatser av Momentum.

KundId: **beskrivning:** Identifierare för kund.
datatyp: integer
verifieringsuttryck: >0
obligatorisk: ja

Kundnamn: **beskrivning:** Kundens namn.
datatyp: varchar(50)
verifieringsuttryck: not null
obligatorisk: ja

C.2.5 Leverantör

I Leverantör lagras de företag som levererar artiklar till Momentum.

LeverantörsId: **beskrivning:** Identifierare för leverantör.
datatyp: integer
verifieringsuttryck: >0
obligatorisk: ja

Leverantörsnamn: **beskrivning:** Leverantörens namn.
datatyp: varchar(30)
verifieringsuttryck: not null
obligatorisk: ja

C.2.6 Modell

I Modell lagras de modeller som varje märke har.

ModellId: **beskrivning:** Identifierare för modell.
datatyp: integer
verifieringsuttryck: >0
obligatorisk: ja

Modellbeteckning: **beskrivning:** Modellens beteckning.
datatyp: varchar(30)
verifieringsuttryck: not null
obligatorisk: ja

MärkesId: **beskrivning:** Identifierare för märke.
datatyp: integer
verifieringsuttryck: >0
obligatorisk: ja
främmandenyckel: Refererar till MärkesId i märke.

C.2.7 Märke

I Märke lagras de olika märken som ingår i någon packningssats.

MärkesId: **beskrivning:** Identifierare för märke.
datatyp: integer
verifieringsuttryck: >0
obligatorisk: ja

Märkesnamn: **beskrivning:** Märkets namn.
datatyp: varchar(30)
verifieringsuttryck: not null
obligatorisk: ja

C.2.8 Packningssats

I Packningssats lagras information om de packningssatser som Momentum har tillgängliga.

PackningssatsId: **beskrivning:** Identifierare för packningssats.
datatyp: integer
verifieringsuttryck: >0
obligatorisk: ja

Packningssatsnummer: **beskrivning:** Nummer på packningssats.
datatyp: varchar(30)
verifieringsuttryck: not null
obligatorisk: ja

Packningssatsbenämning: **beskrivning:** Benämning för packningssats.
datatyp: varchar(30)
verifieringsuttryck: -
obligatorisk: nej

DatumUppmätt: **beskrivning:** Datum för skapande av packningssats.
datatyp: date
verifieringsuttryck: ÅÅMMDD
obligatorisk: Nej

AnställningsId: **beskrivning:** Identifierare för anställd.
datatyp: integer
verifieringsuttryck: >0
obligatorisk: ja
främmandenyckel: Refererar till AnställningsId i Anställd

ModellId: **beskrivning:** Identifierare för modell.
datatyp: integer
verifieringsuttryck: >0
obligatorisk: ja

främmandenyckel: Refererar till ModellId i Modell

C.2.9 Ritning

I Ritning lagras ritningar som finns kopplade till en artikel.

RitningsId: **beskrivning:** Identifierare för ritning.
datatyp: integer
verifieringsuttryck: >0
obligatorisk: ja

Ritningsnummer: **beskrivning:** Ritningens nummer.
datatyp: varchar(30)
verifieringsuttryck: not null
obligatorisk: ja

C.2.10 Uppsättning

I Uppsättning lagras antalet artiklar i varje packningssats samt kommentarer till artiklarna.

UppsättningsId: **beskrivning:** Identifierare för uppsättning.
datatyp: integer
verifieringsuttryck: >0
obligatorisk: ja

ArtikelId: **beskrivning:** Identifierare för artikel.
datatyp: integer
verifieringsuttryck: >0
obligatorisk: ja
främmandenyckel: Refererar till ArtikelId i Artikel

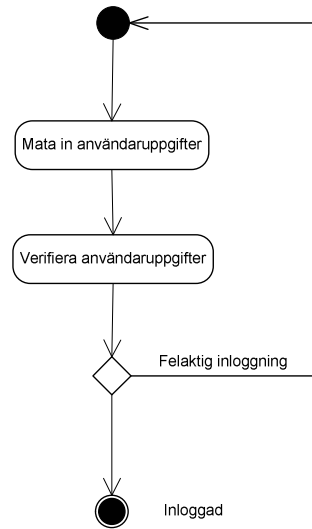
PackningssatsId: **beskrivning:** Identifierare för packningssats.
datatyp: integer
verifieringsuttryck: >0
obligatorisk: ja
främmandenyckel: Refererar till PackningssatsId i Packningssats.

Antal: **beskrivning:** Antal av en artikel i en packningssats.
datatyp: integer
verifieringsuttryck: >0
obligatorisk: ja

Kommentar: **beskrivning:** Kommentarer om en artikel.
datatyp: varchar(50)
verifieringsuttryck: -
obligatorisk: nej

D Bilaga – Testfall

D.1 Testfall - Inloggning



Figur D.1 – Flödesschema för att logga in

D.1.1 Testfall 1 – Korrekt inloggning

Syfte: Kontrollera att en giltig användare kan logga in på systemet.

Input: Giltigt användarnamn och lösenord.

Resultat: Användare inloggad.

Beskrivning: Mata in ett giltigt användarnamn och lösenord. Klicka sedan på logga in.

D.1.2 Testfall 2 – Felaktig inloggning

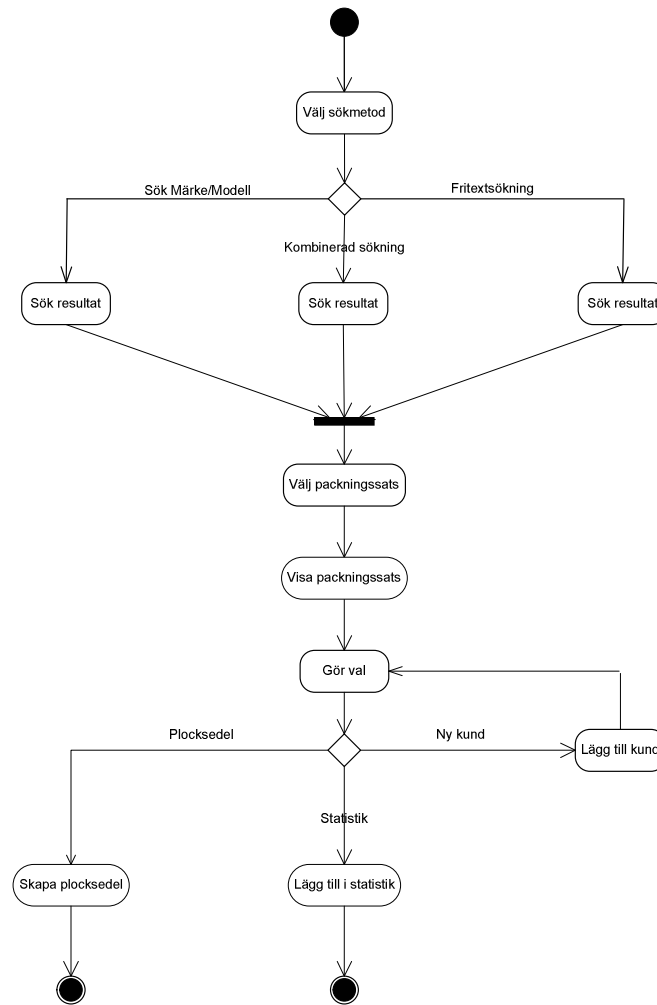
Syfte: Kontrollera att en ogiltig användare inte kan logga in på systemet.

Input: Ogiltigt användarnamn och lösenord.

Resultat: Användare ej inloggad.

Beskrivning: Mata in ett ogiltigt användarnamn och lösenord. Klicka sedan på logga in.

D.2 Testfall - Sökning



Figur D.2 - Flödesschema för att söka

D.2.1 Testfall1 – Sökning på märke/modell

Syfte: Kontrollera att sökning på märke/modell lyckas.

Input: Välj märke och modell i rullgardinsmenyn.

Resultat: Samtliga packningssatser som är refererade till valt märke och modell listas.

Beskrivning: Välj märke. Välj modell. Klicka på *Sök packningssats*.

D.2.2 Testfall 2 – Kombinerad sökning

Syfte: Kontrollera att kombinerad sökning efter packningssats fungerar.

Input: Mata in packningssatsnummer och packningssatsbenämning i respektive textruta.

Resultat: Samtliga packningssatser som är refererade till valt packningssatsnummer och packningssatsbenämning listas.

Beskrivning: Fyll i fälten packningssatsnummer samt packningssatsbenämning. Klicka på *Sök packningssats*.

D.2.3 Testfall 3 – Fritextsökning

Syfte: Kontrollera att fritextsökning efter packningssats fungerar.

Input: Mata in valfritt sökord som finns i minst en packningssats.

Resultat: Samtliga packningssatser som innehåller sökordet listas.

Beskrivning: Fyll i fältet för fritextsökning med valt sökord. Klicka sedan på *Sök packningssats*.

D.2.4 Testfall 4 – Ny kund

Syfte: Kontrollera att ny kund läggs till vid köp av packningssats.

Input: Resultat från Testfall 3 – Fritextsökning. Mata in kundnamn i avsett textfält.

Resultat: Ny kund har lagts till.

Villkor: Testfall 3 – Fritextsökning lyckades

Beskrivning: Fyll i fältet för ny kund med kundnamn. Klicka sedan på *Lägg till ny kund*.

D.2.5 Testfall 5 – Lägg till statistik

Syfte: Kontrollera att statistik läggs till vid köp av ett godtyckligt antal packningssatser.

Input: Resultat från Testfall 3 – Fritextsökning. Valt kundnamn ur rullgardinsmenyn.

Resultat: Statistik har lagts till för kund och packningssats.

Villkor: Testfall 3 – Fritextsökning lyckades.

Beskrivning: Välj kund ur rullgardinsmenyn. Fyll i fältet för antal packningssatser med godtycklig siffra. Klicka sedan på *Lägg till statistik*

D.2.6 Testfall 6 – Skapa plocksedel.

Syfte: Kontrollera att plocksedel skapas.

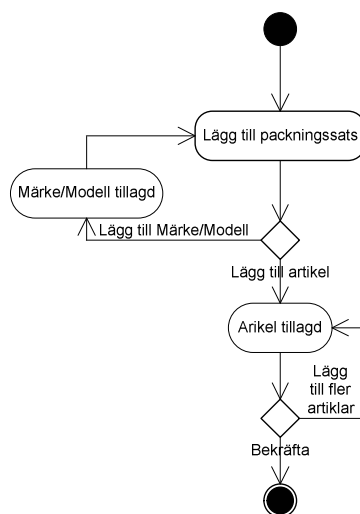
Input: Resultat från Testfall 3 – Fritextsökning.

Resultat: Plocklista skapats för packningssats.

Villkor: Testfall 3 – Fritextsökning lyckades.

Beskrivning: Klicka på *Skapa plocklista*

D.3 Testfall – Skapa ny packningssats



Figur D.3 – Flödesschema för att lägga till packningssatser

D.3.1 Testfall 1 – Lägg till packningssats

Syfte: Kontrollera att packningssats läggs till i databasen.

Input: Godtyckligt packningssatsnummer samt artikelnummer.

Resultat: Packningssats har lagts till.

Beskrivning: Mata in ett godtyckligt packningssatsnummer samt ett godtyckligt artikelnummer.
Klicka på *Lägg till ytterligare artiklar*. Klicka på *Bekräfta*.

D.3.2 Testfall 2 – Lägg till märke/modell

Syfte: Kontrollera att märke och eller modell lagts till.

Input: Namn på märke och eller modell.

Resultat: Märke och eller modell lagts till.

Beskrivning: Mata in ett godtyckligt namn i fältet för märke. Klicka på *Lägg till märke*. Mata in godtycklig text i fältet för modell. Klicka på *Lägg till modell*.

D.3.3 Testfall 3 – Lägg till ytterligare artiklar

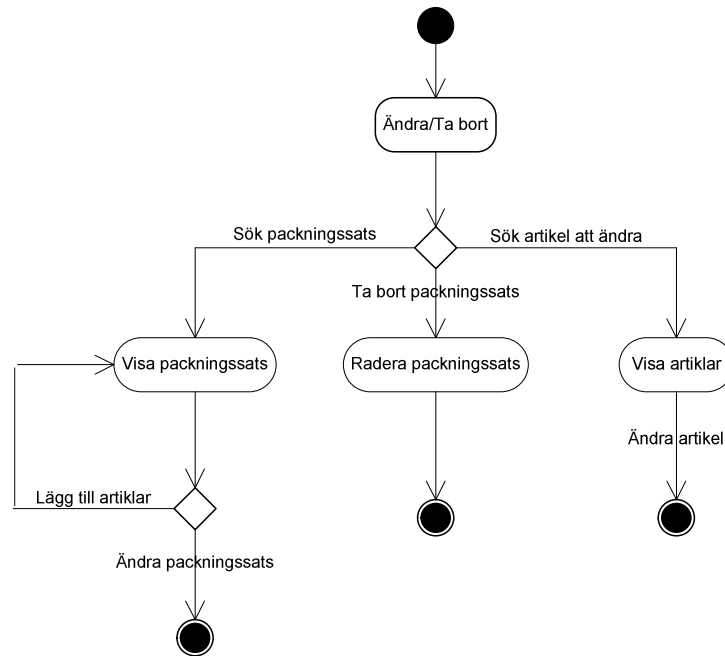
Syfte: Kontrollera att flera artiklar kan läggas i en packningssats.

Input: Godtyckligt packningssatsnummer samt godtyckliga artikelnummer.

Resultat: Flera artiklar har lagts till i en packningssats.

Beskrivning: Mata in ett godtyckligt packningssatsnummer. Klicka på *Lägg till ytterligare artiklar*. Mata in godtyckligt artikelnummer i fältet för artikelnummer. Klicka på *Lägg till ytterligare artiklar*. Klicka på *Bekräfta*.

D.4 Testfall – Ändra/ta bort packningssats



Figur D.4 – Flödesschema för att ändra/ta bort

D.4.1 Testfall 1 – Lägg till ytterligare artikel till packningssats

Syfte: Kontrollera att ytterligare artikel kan läggas i en befintlig packningssats.

Input: Packningssatsnummer som finns lagrat i databasen.

Resultat: Artikel har lagts till i en befintlig packningssats.

Beskrivning: Mata in ett existerande packningssatsnummer. Klicka på *Sök packningssats*. Klicka på *Lägg till ytterligare artikel*. Mata in godtyckligt artikelnummer i fältet för artikelnummer. Klicka på *Lägg till ytterligare artiklar*.

D.4.2 Testfall 2 – Ta bort packningssats

Syfte: Kontrollera att packningssats kan tas bort.

Input: Packningssatsnummer som finns lagrat i databasen.

Resultat: Packningssats har blivit borttagen ur databasen.

Beskrivning: Mata in ett existerande packningssatsnummer. Klicka på *Radera packningssats*.

D.4.3 Testfall 3 – Ändra artikel

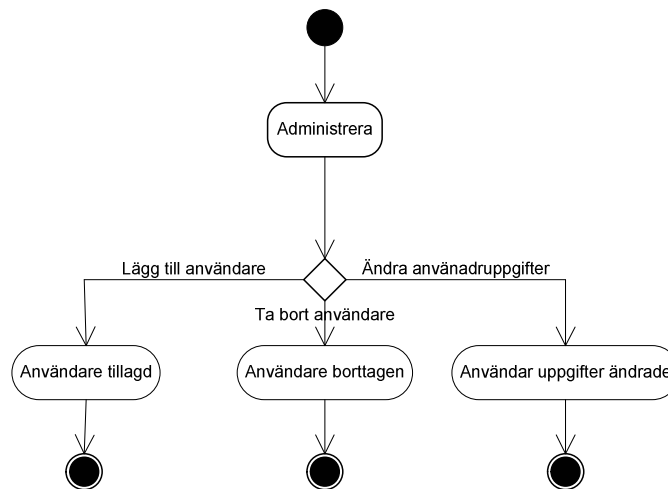
Syfte: Kontrollera att artikel kan ändras.

Input: Artikelnummer som finns lagrat i databasen.

Resultat: Artikel har blivit ändrad.

Beskrivning: Mata in ett existerande artikelnummer. Klicka på *Sök artikel att ändra*. Klicka på *Ändra*. Mata in nytt artikelnummer i fältet för artikelnummer.

D.5 Testfall – Administrera



Figur D.5 - Flödesschema för att administrera

D.5.1 Testfall 1 – Lägg till användare

Syfte: Kontrollera att användare kan läggas till.

Input: Godtyckliga användaruppgifter.

Resultat: Användare har blivit tillagd i databasen *användare*.

Beskrivning: Mata in godtyckliga användaruppgifter i fälten för användaruppgifter. Klicka på *Lägg till användare*.

D.5.2 Testfall 2 – Ta bort användare

Syfte: Kontrollera att användare kan tas bort.

Input: Användarnamn som finns lagrad i databasen *användare*.

Resultat: Användare har blivit borttagen ur databasen *användare*.

Beskrivning: Välj ett användarnamn ur rullgardinsmenyn. Klicka *Ta bort*.

D.5.3 Testfall 3 – Ändra användaruppgifter

Syfte: Kontrollera att användaruppgifter för en användare kan ändras.

Input: Användarkonto som finns lagrat i databasen.

Resultat: Användarkonto har ändrats.

Beskrivning: Mata in förnamn samt efternamn för existerande användare. Klicka på *Sök*.
Klicka på *Ändra*. Mata in godtyckliga nya uppgifter. Klicka på *Ändra*.